applied sciences

MDPI

Article

Formal Analysis and Design of Supervisor and User Interface Allowing for Non-Deterministic Choices Using Weak Bi-Simulation

Shazada Muhammad Umair Khan [1,*] and Wenlong He [2]

[1] Department of Industrial and Information Engineering, Hanyang University, ERICA Campus, Ansan 15588, Korea
[2] Department of Mechanical Engineering, Hanyang University, ERICA Campus, Ansan 15588, Korea; hewenlong@hanyang.ac.kr
* Correspondence: umairkhan@hanyang.ac.kr; Tel.: +82-031-8063-7533

Received: 13 November 2017; Accepted: 22 January 2018; Published: 31 January 2018

Abstract: In human machine systems, a user display should contain sufficient information to encapsulate expressive and normative human operator behavior. Failure in such system that is commanded by supervisor can be difficult to anticipate because of unexpected interactions between the different users and machines. Currently, most interfaces have non-deterministic choices at state of machine. Inspired by the theories of single user of an interface established on discrete event system, we present a formal model of multiple users, multiple machines, a supervisor and a supervisor machine. The syntax and semantics of these models are based on the system specification using timed automata that adheres to desirable specification properties conducive to solving the non-deterministic choices for usability properties of the supervisor and user interface. Further, the succinct interface developed by applying the weak bi-simulation relation, where large classes of potentially equivalent states are refined into a smaller one, enables the supervisor and user to perform specified task correctly. Finally, the proposed approach is applied to a model of a manufacturing system with several users interacting with their machines, a supervisor with several users and a supervisor with a supervisor machine to illustrate the design procedure of human–machine systems. The formal specification is validated by z-eves toolset.

Keywords: user interface; machine interface; supervisor interface; formal model; composite interface

1. Introduction

My specific target in this article is to preserve non-deterministic choices in the context of the fundamental feature of a user and supervisor machine interaction. At a more comprehensive level, the aim of this article is to give confidence to the analytic development of the concept of a formal model of supervisor and supervisor machine. The traditional use of theory has been to evaluate the supervisor interaction under different operating and environmental conditions [1]. In formal model, the formal specification uses the variables that describe the system set of states to develop the proposition with a transition in between them [2]. The process of formal model verification will satisfy the system model and system specification properties [3]. The behavioral equivalences are used to verify a property of a system by assessing the equivalence of the observed system with a system which is known to possess that property and whether the two systems cannot be illustrated by an invader. These formal models manifest the mental and physical activities incorporating the user and supervisor with machine operation to achieve their objectives. Interaction between the user and machine may be brittle; the purpose of interface is only to provide the pre-enumerated condition for which it was planned [4]. The brittle interaction of a complex and safety critical system due to unexpected deficiencies in communication and coordination between the human and the machine [5] encompasses manufacturing systems [6,7]. Likewise, automated systems may be enabled to upscale their potential after being deployed [8]. However, difficult situations can still emerge because of functional conditions or machine behaviors which were not expected by the designer; the automation design having been oversimplified because the embedded machine limitations, machine automation and user interface were not executed in agreement with the design.

Basic formalism is enough to model the functionality of systems, and hence to capture the qualitative behavior, but if one wants to also capture quantitative aspects, such as time or frequency-dependent properties, formalisms must be extended with real-time features [9]. In systems that model quantitative processes, steps are associated with a given quantity, such as the resources (e.g., time or cost) needed to perform that step. Timed automaton has potential to allow for non-deterministic behavior to be solved while the weak bi-simulation perpetuate the co-reachability after its abstraction. The augmented interface was generated through the weak bi-simulation modelling technique that specifically addresses current modelling issues with several users and machines formally represented by timed automata. The concise, complete, unambiguous and comprehensible specification construction that unable the supervisor to make sense out of a rather complex implement. Our objective to develop a formal specification in such a way that it leads in the direction of an appropriate implementation and the process of progress called the refinement. We used Z notation for analyzing and validating the formal specification by z-eves toolset [10,11].

Initially, the formal representation of user and machine model is extended with a discrete event system and its further extension with event-based analysis as a means of representing the activities of the user and machine. These models show goal level of behavior in terms of the user and machine triggering transition. According to the principle of timed automaton, each transition will need time constraints that describe under what conditions a single transition among several possible transitions from the same state will be activated. This phenomenon will help us to understand the system characteristics in real time. Each user interacts with their machine and generate the interaction behavior by combining all users and machines inside the system that links with the supervisor to achieve the system goal under the time constraint value for each transition. However, non-deterministic choice in interface is not only in user but also in supervisor control which is a big safety concern regarding the manufacturing system and can contribute to unforeseen problems. For example, failure of part of the manufacturing system due to poor user interaction consisting of the interface of several machines being poorly defined can cause part supply delay and product recall [12,13] from the market. We first present the formal semantics and syntax of supervisor and supervisor machine model. We show how the interface can be generated through a weak bi-simulation that preserves the co-reachability. We then present a part of a manufacturing system that consists of several users and machines which are controlled by the supervisor. In conclusion, we present the limitations of this technique and the future direction for its model development.

2. Literature Review

2.1. *Interface Generating Models of Human Automation Interaction*

Scientists [14] modelled a user interface established on modes, error and pattern of interaction. Traditionally, most human factors research on interface design has emphasis on perceptual and cognitive compatibility between the human and the interface structure [15]. Much fewer studies have been conducted on the correspondence between the interface [16] and the machine being controlled [17]. Fewer researcher [18] modelled both the machine's behavior and user's operation as discrete event systems and put forward a formal approach for verifying their interface. Further, the scientists [19] modelled the detection of automation surprise in human machine system operated through multiple operation by user. Researchers [20] discussed an interface generating model based on user observable vs. unobservable and controllable vs. uncontrollable events provoked by the user decision through the interface model. Few scientists used logics and theory to represent the user and machine interaction [21].

In their article, the user can be regarded as a human intervention which is an extension of supervisory control with some expedite actions and avoid mode confusion [22] accompany through the behavior of underlying machine by supervisor. To be clearer, these models explicitly used to develop training manuals to anticipate the underlying machine behavior and avoiding undeveloped accidents due to mode confusion and automation surprises. To execute this ultimate concern, it is important for manufacturing system for understanding of user, supervisor and machine model in realistic term. Moreover, I need to manifest the interface constitute of following different notions:

1. User and Supervisor action-based interfaces, which distinguish between controllable vs. uncontrollable, observable vs. un-observable by the user and supervisor and internal transitions.
2. The operating modes that characterize the user, supervisor and machine states that the user or supervisor needs to be able to distinguish.

2.2. Formal Verification with Interface Generating Models

Model checking is a computerized formal method practiced verifying system contents based on a formal model a set of anticipated characteristics in the form of formal specification [23]. The formal model of a system defines with respect to the variable in the form of set and shifts among state of variable. Verification is the procedure of verifying that the system encounters the properties lying under specification. Model checking achieves this procedure automatically by comprehensively identifying a system's state space to control if these measures hold. If there is an against of rule, then counterexample will generate. The counterexample will show the clear description of rule violation for each state and specification including with next state of model that led up to the violation.

The scientist [24] integrates the user interface model into the discrete event system. This will allow the scientist to verify the state matching between the user and the machine [25] under the umbrella of user knowledge and their expectations. Also, it will verify that the interface will able to satisfy the user requirements and the need for updates to fulfil the system requirements. The first concept regarding formal verification of a user machine interaction model was introduced by [26]. In this composition, the scientist combines the user and machine model states into a state duplet and estimates their matched march with respect to the identical specification classes. Few scientist [20] explores the verification of a user interface model by simulation relation. They [5] generate the user interface constructed on formal model and verifies it systematically maneuvering the formal specifications. In response of such needs, the non-deterministic choice at any state of interface is still insufficient for these interfaces. Further, the interface generating model can be used to include a single user and machine-based interaction. In case of several users that are controlled by the supervisor in a formal system models along with the necessary system elements such as the user machine interface and supervisor machine interface have still not been considered by the research community.

2.3. Limitations on Current Techniques

The prospective verification analysis is limited in extent by the potential user machine interaction model to establish the correct user behavior. Works such as these researcher [18] express the user machine interaction with an account of a formal description and they are not well reinforce the relation with user machine interaction. Correspondingly, they [20] established awareness of the link to unobservable and uncontrollable events for user but it had some weak points. Further, they [5] investigated and analyzed the human machine interaction through a formal model representation using predicate and proposition. All the provided practices do not have the non-deterministic behavior or choices lie at any state.

Moreover, currently we cannot generate the interface when the system has more than one user and machine all of which are interlinked with the supervisor as we described the limitation in Table 1. We are also interested to investigate the supervisor interface. $(\sum_M^{com})_{choice=1}$, $(\sum^{obs})$ and $(\sum^{int})$ have already evaluated in [19] the user domain while the supervisor's perspective still needs to be considered. Further, $(\sum^{com})_{choice>1}$ and $(\sum^{com'})_{choice>1}$ events are still not being investigated by scientists.

Table 1. Event execution based on criteria of observability and controllability through user and machine operation.

#User and Machine	User	Choice	Event	Controllability	Observability	Supervisor Model	Supervisor Machine Model	Publication
Only 1	$M1 \overset{\eta_{1,1}*}{\rightarrow} L1$ User operation	1	$(\Sigma_M^{com})_{choice=1}$	Yes	Yes	No	No	[26]
Only 1	$L2 \overset{\beta_{1,1}*}{\rightarrow} M1$ Machine operation	-	Σ_M^{obs}	No	Yes	No	No	[20]
Only 1	$L1 \overset{\gamma_{1,1}*}{\rightarrow} L2$ $L2 \overset{\gamma_{1,2}*}{\rightarrow} L1$ $M1 \overset{\gamma_{1,3}*}{\rightarrow} M2$ $M2 \overset{\gamma_{1,5}*}{\rightarrow} M3$ $M2 \overset{\gamma_{1,4}*}{\rightarrow} M1$ $M3 \overset{\gamma_{1,6}*}{\rightarrow} M2$ Machine operation	-	Σ_M^{int}	No	No	No	No	[20,26]
>1 can be used	$L2 \overset{\alpha_{1,1}*}{\rightarrow} M1$ $L2 \overset{\alpha_{1,2}*}{\rightarrow} M2$ User operation	>1	$(\Sigma_{M,i}^{com})_{choice>1}$	Yes	Yes	No	No	***
>1 can be used	$N_{S0} \overset{\alpha_{SU1}}{\rightarrow} L1$ Supervisor operation	1 **	$(\Sigma_S^{com'})_{choice>1}$	Yes	Yes	Yes	Yes	***

Note: * Previously model can support only one user and machine so we used η_1* representation. Now, we changed the representation because of more than one user and machine from η_1* to $\eta_{1,1}$ (user or machine = 1, operation = 1). ** Supervisor will deliver the instruction to user for proceeding the user operations. Also, depend upon supervisor model; supervisor may have one or more than one choice to execute her/his task. *** Contribution.

3. Formal Semantics of the Supervisor Machine Interaction

3.1. Formal Semantics of the Supervisor Model

We are assuming that we are considering only one supervisor in our system. This supervisor must merge the several users who are operating their machines. The functions and responsibility of supervisor is that she/he must inform the users of their tasks initially. After that the supervisor will also play their role by modifying the product through different user and machine interactions and obtain information through the supervisor interface [27]. Therefore, the supervisor interface should be correct [28] and meaningful otherwise the outcome of product will not be as per the requirement of customer. We used the four-machine cell $N_{S1},\ N_{S2},\ N_{S3}$ and N_{S4} operated by the four different users. At each cell we have states for operation for example in N_{S1} we have four states like wise $L_1,\ L_2$ represent the low having the same class while $M_1,\ M_2$ represent the medium having same class as shown in Figure 1. In between of these states we have transitions and these transitions are labelled as per our definitions. The dotted line represents the machine transitions while the dark line represents the user transition within the cell or if the transition is incoming or outgoing the cell then it termed as supervisor transition.

The supervisor has also the same character as the individual user, as shown in Table 2. The observable and controllable events $\alpha_{SUi} \in (\sum_S^{com'})_{choice>1}$ is formed due to the execution of a supervisor task. The $\alpha_{SUi} \in (\sum_S^{com'})_{choice>1}$ is the communication of the supervisor with the user and the supervisor has more than one choice in their state. All of them are observable and controllable. The machine transition which is also observable but uncontrollable $\beta_{si} \in \sum_S^{obs}$ likewise moving the lathe to milling operation. If there is change inside the system without a user task, then the event must fall into the unobservable and uncontrollable category $\gamma_{si} \in \sum_S^{int}$ for the supervisor.

$$\sum_S = (\sum_S^{com})_{choice=1} \dot{\cup} (\sum_S^{com'})_{choice>1} \dot{\cup} \sum_S^{obs} \dot{\cup} \sum_S^{int} \tag{1}$$

The supervisor model can be represented as the timed automata;

$M_S = \langle N_S, n_{OS}, E_S, I_S \rangle$ Where,
N_S : Set of supervisor states,
$n_{OS} \in N_S$: Initial (starting) state in the supervisor,
$E_S \subseteq N_S \times B(C) \times \sum_S \times 2^C \times N_S$: Set of edges termed as transition among the system states,
$I_S : N_S \rightarrow B(C)$ Assigns invariants to locations.

$B(C)$ is the clock constraint where $x \sim n$ or $x - y \sim n$ for $x, y \in C$, $\sim \in [\leq, <, =, >, \geq]$ and $n \in$. In our modelling, We used T_{Si} for supervisor modelling state to state time constraint while, T_{xi} is used for user station. The x is representing the station, we used as a, b, c and d station respectively while the $i \in \mathbb{N}$.

According to the above definition we can write $n_S :\overset{g,a,r}{\rightarrow} n'_S$ if and only if $\langle n_S, g_S, a_S, r_S, n'_S \rangle \in E_S$.

g_S : is the guard of $e_s = \langle n_S, g_S, a_S, r_S, n'_S \rangle \in E_S$, a_S : is the action of e_s, r_S: is the set of clocks that is reset by e_s, $\sum_S$: Set of events among the supervisor states.

Definition 1. *The appearance of observable and controllable event* $\eta_{S1} \in (\sum_s^{com})_{choice=1}$ *will change the state of the supervisor interface through supervisor action. The formation of this event* $\eta_{i,j}$ *by any state will yield the only choice of user operations. When it appears in a state then it will become* $(N_{SM},\ N_S) \in B_{RSM}$ *or* $((N_{SM0}),\ (N_{S0})) \in B_{RSM}$. *The change of machine state* $n'_{SM} \in N_{SM}$ *or* $(N_{SM1}) \in (N_{SM})$ *because of the observable and controllable event* $\eta_{S1} \in (\sum_S^{com})_{choice=1}$.

Definition 2. *The observable but uncontrollable event* $(\beta_{S1}, \beta_{S2}) \in (\sum_s^{obs})$ *will change the state of user interface through the occurrence of a machine transition. When it appears in a state then* $((N_{SM}, N_S) \in B_{RSM}$

or $(N_{SM2}),(N_{S2})) \in B_{RSM}$ as in the supervisor model of Figure 1. The change of machine state $n'_{SM} \in N_{SM}$ or $(N_{SM3}) \in (Station2)_M$ because of the observable and controllable event $(\beta_{S1},\beta_{S2}) \in (\sum_s^{obs})$ that change the state $(N_{SM2}) \overset{\beta_{S1}}{\rightarrow} (N_{SM3})',(N_{SM3}) \overset{\beta_{S2}}{\rightarrow} (N_{SM2})'$ having the interaction of $(n'_{SM},n'_S) \in B_{RSM}$ as per the supervisor model described in Figure 1 $((N_{SM2})^{\#x2032;},(N_{S2})') \in B_{RSM}$ if and only if $\exists\beta_{S1}$ and $((N_{SM3})',(N_{S3})') \in B_{RSM}$ if and only if $\exists\beta_{S2}$.

Definition 3. *The unobservable and uncontrollable event $(\gamma_{S1},\gamma_{S2},\gamma_{S3},\gamma_{S4}) \in \sum_S^{int}$ will not change the state of the supervisor. It is the internally change of machine state is and unobservable because there is no supervisor action either before or after the formation of the machine state. It may cause the uncontrollable event $(\gamma_{S1},\gamma_{S2},\gamma_{S3},\gamma_{S4}) \in \sum_S^{int}$. There is no binary relation that exists and the change of machine state $n'_{SM} \in N_{SM}$ or $(N_{SM1}) \in (Station1)_M,(N_{SM2}) \in (Station2)_M$ and $(N_{SM4}) \in (Station4)_M$ because of we have no supervisor action and the unobservable and uncontrollable event $(\gamma_{S1},\gamma_{S2},\gamma_{S3},\gamma_{S4}) \in \sum_S^{int}$ that change the state of machine $(N_{SM1}) \overset{\gamma_{S1}}{\rightarrow} (N_{SM2})$ and $(N_{SM2}) \overset{\gamma_{S2}}{\rightarrow} (N_{SM1})$. Similarly, $(N_{SM2}) \overset{\gamma_{S3}}{\rightarrow} (N_{SM4})$ and $(N_{SM4}) \overset{\gamma_{S4}}{\rightarrow} (N_{SM2})$ that is internally triggered by machine having no interaction with supervisor.*

Definition 4. *The observable and controllable supervisor event $(\alpha_{SU1},\alpha_{SU2},\alpha_{SU3},\alpha_{SU4}) \in (\sum_S^{com})_{choice>1}$ has more than one choice at the starting state of supervisor for the user to perform the machine interaction. Similarly, $(\alpha_{S1},\alpha_{S2}) \in (\sum_S^{com'})_{choice>1}$ it has also more than one choice at N_{S1} state of supervisor to execute the task by the supervisor operations. The time transition will handle this choice easily to allow user to perform their operation safely and correctly. When it appears in a state then $(N_{SM},N_S) \in B_{RSM}$ or $(N_{SM1},N_{S1}) \in B_{RSM}$ as in the supervisor model of Figure 1. The change of machine state $n'_{SM} \in N_{SM}$ or $(N_{SM1}) \in (Station1a)_{SM},(N_{SM2}) \in (Station1b)_{SM},(N_{SM3}) \in (Station2)_{SM}$ and $(N_{SM4}) \in (Station1c)_{SM}$ because of the observable and controllable event that change the state $(\alpha_{SU1},\alpha_{SU2},\alpha_{SU3},\alpha_{SU4}) \in (\sum_S^{com})_{choice>1}$ and in the Form of transition $(N_{SM0}) \overset{\alpha_{SU1}}{\rightarrow} (N_{SM1}),(N_{SM0}) \overset{\alpha_{SU2}}{\rightarrow} (N_{SM2}),(N_{SM0}) \overset{\alpha_{SU3}}{\rightarrow} (N_{SM3}),(N_{SM0}) \overset{\alpha_{SU4}}{\rightarrow} (N_{SM4}),(N_{SM1}) \overset{\alpha_{SU6}}{\rightarrow} (N_{SM3})$ and $(N_{SM1}) \overset{\alpha_{SU5}}{\rightarrow} (N_{SM2})$ having the interaction of $(n'_{SM},n'_S) \in B_{RSM}$ or $(N_{SM0},N_{S0}) \in B_{RSM}$ has four choices and $(N_{SM1},N_{S1}) \in B_{RSM}$ has two choices for the interaction.*

According to all the above definitions, the all supervisor operation is observable and controllable. The event that is unobservable and uncontrollable $|\mathrm{E}_S(n_S,e_{com}| \leq 1$ for any supervisor state $n_S \in N_S$ is uncontrollable to the supervisor. We also incorporated the deterministic and non-deterministic choices into the supervisor model. Moreover, we used the Z notation for analyzing and validating the supervisor model using z-eves toolset. The information is well structured and presented at appropriate abstraction using z notation. The snapshot for specification validation of supervisor model is given in the appendix section in Figure A1.

Table 2. Event execution based on criteria of observability and controllability through supervisor and supervisor machine operation.

#User and Machine	Supervisor	Choice	Event	Controllability	Observability	Publication
More than 1	$N_{SM0} \xrightarrow{\eta_{S1}} N_{SM0}$ Supervisor operation	1	$(\sum_{SM}^{com})_{choice=1}$	Yes	Yes	***
More than 1	$N_{SM2} \xrightarrow{\beta_{S1}} N_{SM3}$ $N_{SM3} \xrightarrow{\beta_{S2}} N_{SM2}$ Machine operation	**	$\sum_{SM}^{obs}$	No	Yes	***
More than 1	$N_{SM1} \xrightarrow{\gamma_{S1}} N_{SM2}$ $N_{SM2} \xrightarrow{\gamma_{S3}} N_{SM4}$ $N_{SM2} \xrightarrow{\gamma_{S2}} N_{SM1}$ $N_{SM4} \xrightarrow{\gamma_{S4}} N_{SM2}$ Machine operation	**	$\sum_{SM}^{int}$	No	No	***
More than 1	$N_{SM0} \xrightarrow{\alpha_{SU1}} N_{SM1}$ $N_{SM0} \xrightarrow{\alpha_{SU2}} N_{SM2}$ $N_{SM0} \xrightarrow{\alpha_{SU3}} N_{SM3}$ $N_{SM0} \xrightarrow{\alpha_{SU4}} N_{SM4}$ $N_{SM1} \xrightarrow{\alpha_{SU5}} N_{SM2}$ $N_{SM1} \xrightarrow{\alpha_{SU6}} N_{SM3}$ Supervisor operation	>1 *	$\alpha_{SUi} \in (\sum_{SM}^{com'})_{choice>1}$	Yes	Yes	***

Note: * Supervisor will deliver the instruction to user for preceding the user operations. Also, depend upon supervisor model; supervisor may have one or more than one choice to execute her/his task. ** The machine transition could be one or more but activate as per time transition. *** Contribution.

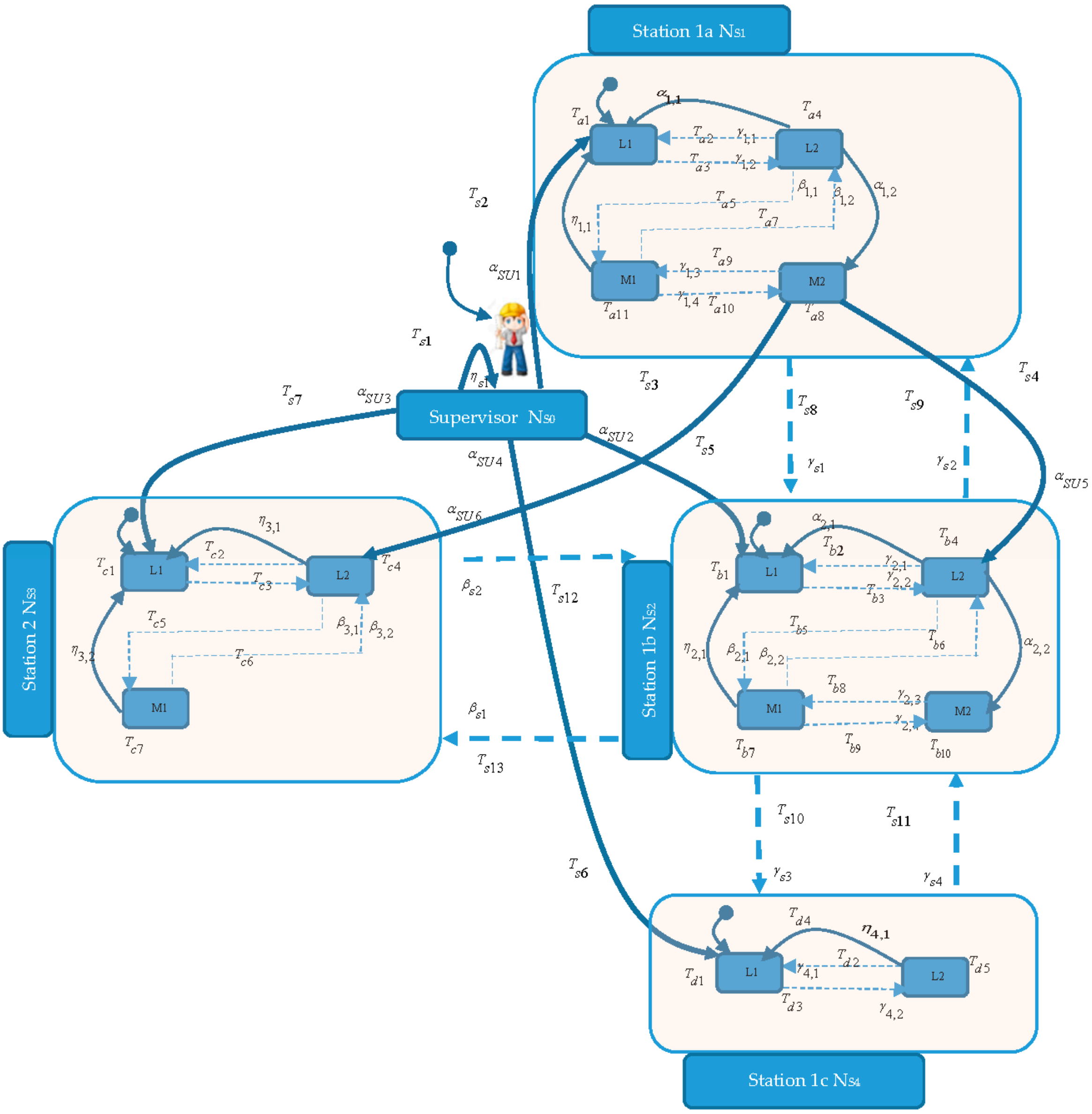


Figure 1. An example of a supervisor model.

3.2. *Formal Semantics of the Supervisor Machine Model*

The supervisor machine model consists of several interactions of users with respect to their machines to reach the desired goal. Each work cell consisting of a user and machine would be considered as a supervisor state. Moving the product from one work cell to another is considered as a machine transition. The state is observable vs controllable and uncontrollable vs unobservable if and only if the criteria as mentioned in the Table 2 are true.

The supervisor machine model as shown in Figure 2 can be represented in terms of timed automata;

$M_{SM} = \langle N_{SM}, n_{OSM}, E_{SM}, l_{SM} \rangle$ Where,

N_{SM}: Set of supervisor machine states,

$n_{OSM} \in N_{SM}$: Initial (Starting) state of supervisor machine,

$E_{SM} \subseteq N_{SM} \to N_{SM} \times B(C) \times N_{SM} \times 2^C \times \sum_{SM}$ Set of edges termed the transition among the system states, $I_{SM} : N_{SM} \to B(C)$ assigns invariants to locations, $B(C)$ is the clock constraints where

$x \sim n$ or $x \sim y \sim n$ for $x, y \in C$, $\sim \in [\leq, <, =, >, \geq]$ and $n \in \mathbb{N}$. According to the above definition we can write $n_{SM} \overset{g,a,r}{\to} n'_{SM}$ when $\langle n_{SM}, g_{SM}, a_{SM}, r_{SM}, n'{}_{SM} \rangle \in E_{SM}$. We used similar technique for defining the time constraint as we described in supervisor model.

g_{SM} : is the guard of $e = \langle n_{SM}, g_{SM}, a_{SM}, r_{SM}, n'_{SM} \rangle \in E_{SM}$,
a_{SM} : is the action of e_{SM},
r_{SM} : is the set of clocks that is reset by e_{SM},
$\sum_{SM}$: Set of events among the supervisor machine states.

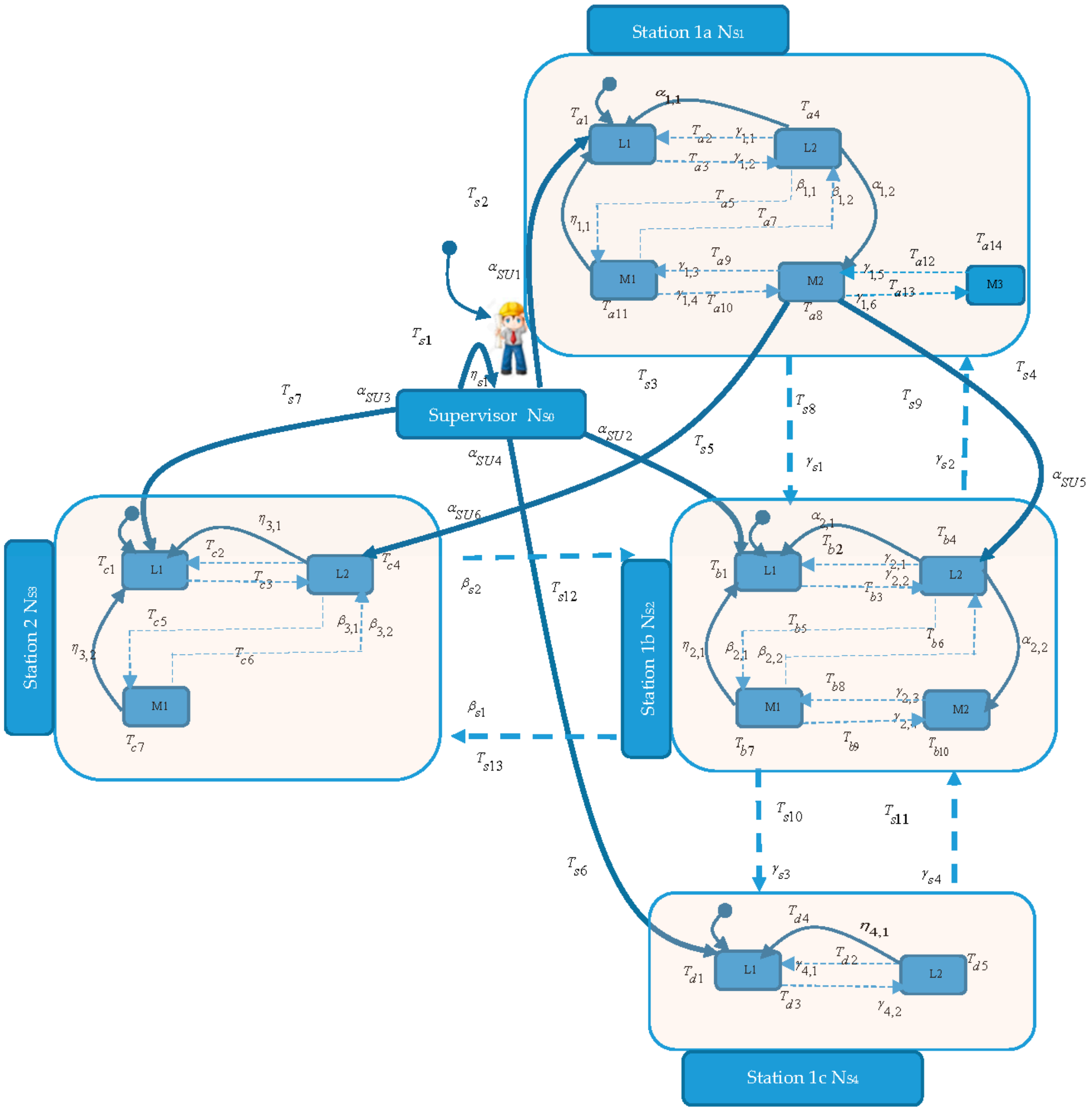


Figure 2. An example of a supervisor machine model.

The set of supervisory action that consists of $\sum_{SM} = \{(\sum_{SM}^{com})_{choice=1}\} \dot{\cup} (\sum_{SM}^{com'})_{choice>1}\} \dot{\cup} \sum_{SM}^{obs} \dot{\cup} \sum_{M}^{int}$ has three disjoint subsets. These subsets are only workable for $(\sum_{SM}^{com})_{choice=1}$: an observable and controllable event having only one choice for supervisor operation, $(\sum_{SM}^{com'})_{choice>1}$: an observable and controllable event having more than one choice for supervisor operation, $\sum_{SM}^{obs}$: an observable and uncontrollable event and $\sum_{SM}^{int}$: an unobservable and uncontrollable event for user. These subsets are based on a discrete event system using finite state machines. In our case, we represent the semantics of the machine model by time automata because we can easily include more than one choices of user operation at any stage of machine state as shown in Figure 2. Now the updated equation will be as follows;

$$\sum_{SM} = \Bigg\{ (\sum_{SM}^{com})_{choice=1} \Bigg\} \dot{\cup} \, (\sum_{SM}^{com'})_{choice>1} \Bigg\} \dot{\cup} \sum_{SM}^{obs} \dot{\cup} \sum_{SM}^{int} \tag{2}$$

According to the Figure 2 $\eta_S = \{\eta_{S1}\}$ are the observable and controllable events that exists in N_{SM0} state of machine such that $\forall \eta_S \in (\sum_{SM}^{com})_{choice=1}$. They $\beta_S = \{\beta_{S1}, \beta_{S2}\}$ are the observable but uncontrollable events that exists in the N_{S2} and N_{S3} state of machine such that $\forall \beta_S \in \sum_{SM}^{obs}$. They $\gamma_S = \{\gamma_{S1}, \gamma_{S2}, \gamma_{S3}, \gamma_{S4}\}$ are the unobservable and uncontrollable events in N_{SM2} and N_{SM4} such that $\forall \gamma_S \in \sum_{SM}^{int}$. They $\alpha_{SU} = \{\alpha_{SU5}, \alpha_{SU6}\}$ are the observable and controllable events that exists in the $M1$ machine state in which the user has only one choice to execute their operation such that $\forall \alpha_{SU} \in (\sum_{SM}^{com'})_{choice>1}$. These $\alpha_{SU} = \{\alpha_{SU1}, \alpha_{SU2}, \alpha_{SU3}, \alpha_{SU4}\}$ are the events in which the supervisor will give the task to the user to execute their operation in the N_{SM0} state such that $\forall \alpha_{SU} \in (\sum_{SM}^{com'})_{choice>1}$. The time constraint at N_{SM0} is T_{S1}. While the outgoing transition has four different choices of state N_{SM1}, N_{SM2}, N_{SM3} and N_{SM4} with time constraints of N_{SM0} state is T_{S1}. N_{SM1} the state is T_{S2}, N_{SM2} state is T_{S5} and T_{S4}, N_{SM3} state is T_{S7}, T_{S3}, N_{SM4} state is T_{S12}. In addition, the formal specification of supervisor machine model is analyzed and validated using a-eves toolset. The snapshot for formal specification validation of supervisor machine model is given in appendix section in Figure A2.

3.3. Supervisor Interface Model

The details regarding the controllable and observable events, number of user and machine, non-deterministic choices, which lie or not in the user and supervisor model are mentioned in Table 2. The events $\alpha_{SU5}, \alpha_{SU6} \in (\sum_{SM}^{com'})_{choice>1}$ are supervisor observable and controllable having more than one choice at a single state and two choices at N_{S1}. The event $\eta_{S1} \in (\sum_{SM}^{com})_{choice=1}$ is uncontrollable but observable for user and has only one choice for the supervisor perform their operation at N_{S0} described in the supervisor interface model and illustrated in Figure 3. The event $\gamma_{S1}, \gamma_{S2}, \gamma_{S3}, \gamma_{S4} \in \sum_{SM}^{int}$ are uncontrollable and unobservable for user.

Finally, the event $(\alpha_{SU1}, \alpha_{SU2}, \alpha_{SU3}, \alpha_{SU4}) \in (\sum_{SM}^{com'})_{choice>1}$ are observable and controllable for supervisor that provides information to the user to perform their operation using the supervisor interface as shown in Figure 3. Accordingly, this event defines the supervisor and supervisor machine model as shown in the Table 2, the interaction between the $n_S \in N_S$ supervisor and $n_{SM} \in N_{SM}$ machine, $n_{SM} \in N_{SM}$ supervisor and $n_S \in N_S$ user described with binary relation. $B_R \subseteq N_S \times N_U$ implies that the interaction $(Supervisor, N_U)$ between the supervisor and user to proceed with the user operations $B_R \subseteq N_S \times N_{SM}$ implies that the interaction between $(Supervisor, N_{SM0})$, $(Station1a, N_{SM1})$, $(Station1b, N_{SM2})$, $(Station1c, N_{SM4})$, $(Station2, N_{SM3})$ the supervisor gives the command to the machine with the help of the supervisor interface. According to the supervisor machine model shown in Figure 2, it always shows the important transitions and state that describe the behavior of the machine operated by the supervisor according to the guideline of the supervisor interface model.

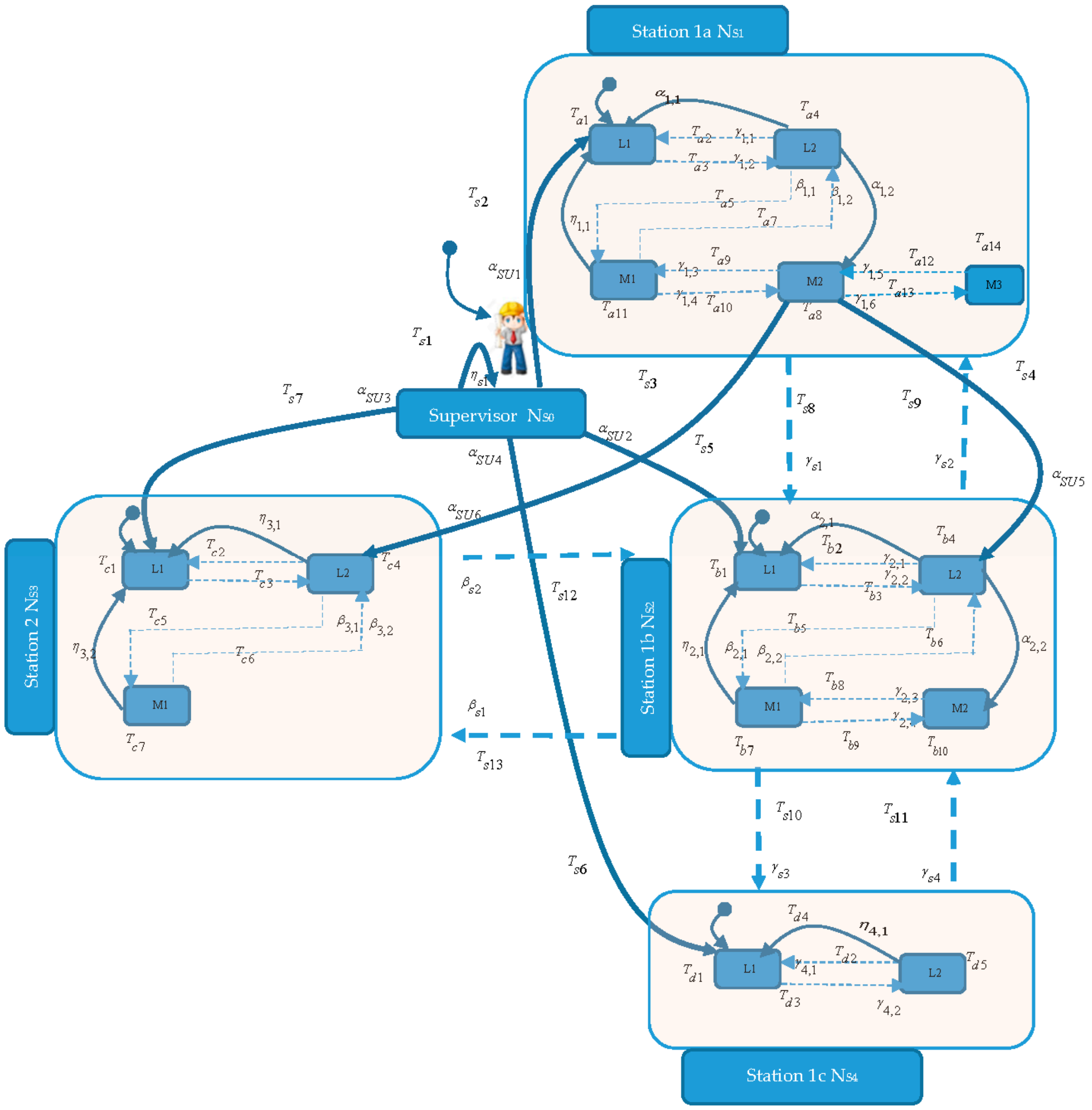

Station 1a Ns1
L1
L2
M1
M2
M3
Supervisor Ns0
Station 2 Ns3
Station 1b Ns2
Station 1c Ns4
T_{a1} T_{a2} T_{a3} T_{a4} T_{a5} T_{a7} T_{a8} T_{a9} T_{a10} T_{a11} T_{a12} T_{a13} T_{a14}
$\alpha_{1,1}$ $\alpha_{1,2}$ $\beta_{1,1}$ $\beta_{1,2}$ $\gamma_{1,1}$ $\gamma_{1,2}$ $\gamma_{1,3}$ $\gamma_{1,4}$ $\gamma_{1,5}$ $\gamma_{1,6}$ $\eta_{1,1}$
T_{s1} T_{s2} T_{s3} T_{s4} T_{s5} T_{s6} T_{s7} T_{s8} T_{s9} T_{s10} T_{s11} T_{s12} T_{s13}
α_{SU1} α_{SU2} α_{SU3} α_{SU4} α_{SU5} α_{SU6} η_{s1}
γ_{s1} γ_{s2} γ_{s3} γ_{s4} β_{s1} β_{s2}
T_{c1} T_{c2} T_{c3} T_{c4} T_{c5} T_{c6} T_{c7}
$\eta_{3,1}$ $\eta_{3,2}$ $\beta_{3,1}$ $\beta_{3,2}$
T_{b1} T_{b2} T_{b3} T_{b4} T_{b5} T_{b6} T_{b7} T_{b8} T_{b9} T_{b10}
$\alpha_{2,1}$ $\alpha_{2,2}$ $\beta_{2,1}$ $\beta_{2,2}$ $\gamma_{2,1}$ $\gamma_{2,2}$ $\gamma_{2,3}$ $\gamma_{2,4}$ $\eta_{2,1}$
T_{d1} T_{d2} T_{d3} T_{d4} T_{d5}
$\eta_{4,1}$ $\gamma_{4,1}$ $\gamma_{4,2}$

Figure 3. An example of supervisor interface model.

4. Interface Generation Using the Weak Bi-Simulation

To generate the interface, we need to consider the timed automaton, which can be either a machine model of user or a machine model of supervisor but as a machine model in the form of tuple can be describe as $M_M = \langle N_M, n_{OM}, E_M, l_M, v_M, N_m \rangle$ shown in Figure 4. The $v_M : N_m \times \sum_M \rightarrow 2^X$ is a partial transition map, N_m is the marker sate. Now, we are considering here the $\sum_{SM}^{int} \sum_M^{cor} \subseteq \sum_M$ and $\sum_M^{rch} \subseteq \sum_M$ while $\sum_M^{cor}$ is showing that all states are reachable but there is no illegal state [29] and $\sum_M^{rch} \subseteq \sum_M$ are the reachable states during the user and machine interaction. For co-reachability [30] the events are observable and controllable with respect to user $\sum_M^{cor} = \sum_{choice=1}^{com} \dot{\cup} \sum_{choice>1}^{com}$ while in reachability, the $\sum_M^{rch} = \sum_{choice=1}^{com} \dot{\cup} \sum_{choice>1}^{com} \dot{\cup} \sum^{obs}$ events are not only $\sum_{choice=1}^{com} \dot{\cup} \sum_{choice>1}^{com}$ observable and controllable but also $\sum^{obs}$ observable and uncontrollable with respect to user therefore $P : \sum_M \rightarrow \sum_M^{rch}$ is in the form of natural projection. The relation of weak bi-simulation to N_M according to the $\sum_M^{rch}$ is the equivalence relation $\kappa \subseteq N_M \times N_M \mid$ for each $(n_M, n_M{}') \in \kappa$ and every $e_M^{cor} \in \sum_M^{cor}$. If $v_M(n_M, e_M^{cor})!$ then $\exists (e_M^{cor})' \in \sum_M^{cor} \Big| v_M(n_M, (e_M^{cor})')!$ having the following.

Definition 5. *The co-reachable event e_M^{cor} that is executed by user $e_M^{cor'}$ is the natural projection $P : (e_M^{cor}) \to P(e_M^{cor'})$ while if the events are the same before and after the state then it will fall under the equivalence relation for each machine state $v_M(n_M, (e_M^{cor})')$ that is reachable by machine state $n_M' \in v_M(N_M', (e_M^{cor})')$ with the equivalence relation $(n_U, n_U') \in \kappa \wedge [n_U \in N_M \Leftrightarrow n_U' \in N_M]$ the events of a system using machine are the same $n_M \in v_M(n_M, (e_M^{cor})')\exists n_M' \in v_M(N_M', (e_M^{cor})')$ before and after the machine state as formally represented if and only if $(n_U, n_U') \in \kappa \wedge [n_U \in N_m \Leftrightarrow n_U' \in N_m]$.*

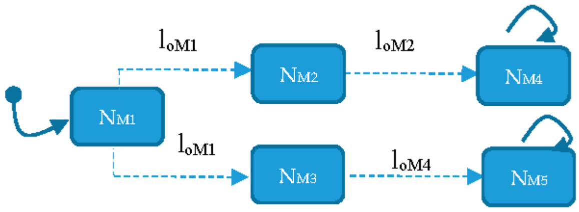


Figure 4. An example of machine model.

We used the abstract idea of [31] and explain this more clearly with the help of the machine model such that, where $\sum_M = \{l_{oM1}, l_{oM2}, l_{oM3}, l_{oM4}\}$ and $\sum_M^{rch} = \{l_{oM2}, l_{oM4}\}$. The equivalence relation will be as a weak bi-simulation relation $\sum_M = \{(N_{M1}, N_{M1}), (N_{M2}, N_{M3})\}$. Hence, the l_{oM2} and l_{oM4} are different so they cannot be executed and will not fulfil the above definition criteria. The κ is the weak bi simulation relation to N_M. According to the $\sum_M^{rch}$ thus the weak bi similarity will be as follows; $\sim \sum_M \sum_M^{rch} = \{(N_{M1}, N_{M2}), (N_{M1}, N_{M3}), (N_{M2}, N_{M1}), (N_{M3}, N_{M4})\}$ for each $n_M \in N_M$ assuming that $[n_M]$ will be represent as class of equivalence of n_M under $\sim \sum_M \sum_M^{rch}$ namely the set of all elements $n'_M \in N_M \mid (n_M, n'_M) \in \sim \sum_M \sum_M^{rch}$. The quotient set of N_M by equivalence relation is $N_M / \sim \sum_M \sum_M^{rch} = \{[n_M] \subseteq N_M \Big| n_M \in N_M\}$ according to $\sim \sum_M \sum_M^{rch}$.

Definition 6. *The timed automaton $M_M = \langle N_M, l_{OM}, E_M, I_M, v_M, N_m \rangle$ assume that $\sum_M^{rch} \subseteq \sum_M$. The degree of timed automaton according to $\sim \sum_M \sum_M^{rch}$ is an automaton $M_M / \sim \sum_M \sum_M^{rch} := \langle N_{MR}, \sum_{MR}, v_{MR}, n_{oMR}, N_m \rangle$ where N_{MR}: is the reduced state as shown in Figure 5, $\sum_{MR}$: a common user action represents a single action in the reduced model in Figure 5, $v_{MR} : N_{MR} \times \sum_{MR} \to 2^X$ and n_{oMR}: the reduced initial state as specified by,*

- $N_{MR} := N_M / \sim \sum_M \sum_M^{rch}$.
- $n_{oMR} := [n_{oM}]$
- $N_{mR} := \{n_R \in N_{MR} | n_R \subseteq N_m\}$
- $\sum_{N_{MR}} := \sum^{rch} \cup \{e \in \sum - \sum^{rch} \Big| (\exists n_M \in N_{MR}) v_M(n_M, e) - [n_M] \neq \{\}\}$
- $v_{MR} : N_{MR} \times \sum_{MR} \to 2^R | (n_{MR}, e) \in N_{MR} \times \sum_{MR}, v_{MR}(n_{MR}, e) =$ $\{n'_{MR} \in N_{MR} | (\exists n_M \in n_{MR}) v_M(n_M, e) \cap n'_{MR} \neq \{\}\}$

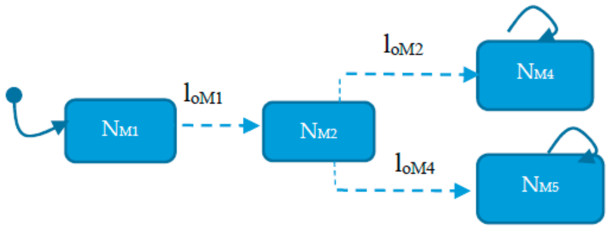


Figure 5. An example of reduced machine model.

We normally describe $M_M \ / \sim \sum_M \sum_M^{rch}$ as the reduced form of M_M by using the techniques of weak bi-simulation relation $\sim \sum_M \sum_M^{rch}$. As per the consideration of our example the final version of the automaton will be as in Figure 5. The partition on the set of state is M_M. As per the easy understandable we have $N_{MR} = \{[N_{M1}], [N_{M2}], [N_{M4}], [N_{M5}]\}$ while the $[N_{M2}] = \{N_{M2}, N_{M3}\}$. Hence the above automaton is in the form of the reduced automaton.

We can define the product of two systems $||sys \times sys \rightarrow sys : (M_M, U_M) \mapsto M_M||U_M := M_M U_M$. The symbol is $||$ used to represents the product of two entities. We normally compare to obtain the product of the machine and user model with respect to transition architectures. According to the [12] user and machine model can be defined as M_M and U_M where timed automaton M_M a set map to another set $N_{MU} : N_M \rightarrow U_M$ if the following conditions hold. The first condition is $N_{MU} : N_M \rightarrow U_M$. The second condition is $N_{MU}(n_{M,0}) = n_{U,0}$ and $N_{MU}(N_{M,m}) = N_{U,m}$. There is a third condition; in the third condition, it is for every $n_M \in N_M$ and $e_M \in \sum_M, \zeta_M(n_M, e_M)! \Rightarrow \zeta_U(N_{MU}(n_U), e_M)! \& \zeta_U(N_{MU}(n_U), e_M) = N_{MU}(- \ \zeta_M(n_M, e_M))$ where this condition holds here $N_{MU}(\zeta_M(n_M, e_M)) := \{N_{MU}(n'_M) | (n'_M) \in \zeta_M(n_U, e_M)\}$. The final condition is for every $n_U \in N_U$ and $e_M \in \sum_M$ therefore $\zeta_U(n_M, e_M)! \Rightarrow \exists n'_M \in N_M) \zeta_M(n'_M, e_M)! \ \& N_{MU}(n'_M) - = n_M$, if and only if $\zeta_U : N_U \times B(C) \times \sum_U \rightarrow 2^C, \zeta_M : N_M \times B(C) \times \sum_M \rightarrow 2^C$. In addition, the formal specification is presented here is analyzed and validated using z notation through z-eves toolset. We used the iteration-based approach to validate the interactive systems by using weak bi-simulation through checking of two systems simultaneously using z-eves, as a snapshot presented in appendix in Figure A3. The iteration 2 and 3 are solved based on source code using java. In iteration 1: we specify the system using the formal specification likewise in our case, the development of formal specification of supervisor, interface and machine model with transition definition. In iteration 2: we can identify the relevant interaction between the supervisor and machine through interface. In relevant interaction there is no blocking, error and illegal state [18]. If there is irrelevant interaction then there must be blocking, error and illegal state then label it. Iteration 3: In this iteration, the identified irrelevant interaction should not be a part of the interaction. It means the supervisor and machine model interaction is free from blocking, error and illegal state. To make sure there is no irrelevant interaction, we apply the above described interface correction using weak bi-simulation method in Section 4. Iteration 4: In this iteration, we check the weak bi-simulation relation between the supervisor and machine model. If the supervisor and machine model are bi similar then the interaction has no illegal, error and blocking state. Hence, the correct interface is ready for the operational mode.

To apply the operation of a model checking verification, formal semantics of supervisor machine interaction model should be interpreted in the language of model checking. We apply the formal semantics to translate supervisor machine interaction model into the symbolic analysis laboratory (SAL) language [32,33]. The formal semantics of supervisor machine interaction model to SAL translation is computerized by our practice constructed in java program which practices the document object model [34] to parse the supervisor and interaction model's extensible markup language (XML) code.

A diversity of examinations was course to authenticate that the translator was producing a SAL code that observed the formal semantics of supervisor machine interaction model. To estimate the complexity and scalability of the formal semantics of supervisor machine interaction models, we produced their models and examined the translated models' in terms of state spaces and runtimes by means of the SAL. The formal semantics supervisor model and formal semantics of supervisor machine interaction models comprise only supervisor who interact with different users and machine in a manufacturing system environment using the supervisor interface. Further, the different users also interact with their machine after getting the information from the supervisor by using their interface. The part manufacturing process were used to understand for both the users and supervisor operations. To guarantee that the verification method would examination a model's complete state space, we formed the specification that would not generate a counterexample. We also took help from the Z-eves tool to validate and verify our presented model using the analysis of counter example.

5. A Case Study of a Part Manufacturing System

To illustrate how this technique can be applied to find solutions to the problems in a parts manufacturing system, we present the following case study. In our case study the transmission does not shift into the higher gear when the driver wants to drive her/his vehicle at more than 160 km per hour. At the start we checked with the scan tool which point in the engine and ECT has the fault. After performing the code test, the malfunctioning in the shift solenoid control circuit was found to be high. There are three causes of these problems; the first one is open circuit, second one is a wiring problem and finally the solenoid valve malfunctioning. As per the manual instructions, first we checked the resistance of several solenoids. Fortunately, we traced the malfunctioning to solenoid valve having low resistance. We checked the underbody of the vehicle; the solenoid valve was completely dipped into oil and this was creating the malfunction. Further after analysis, we saw that the pipe set radiator was damaged and that it was draining the oil into this solenoid valve. After new parts were fitted, we faced same problem within a month. Moreover, we checked the engine noise and heating temperature inside the engine.

We identified that the engine noise and temperature as measured were not as per the standard. So, a high amount of gas was leaking from the exhaust manifold that caused the damage to the pipe resulting in the oil draining into the solenoid valve. We investigated the manufacturing process of exhaust manifold to identify the cause of the gas leakage. Now we will describe its current user and supervisor model and the user and supervisor machine model with the current user and supervisor interface. The data were obtained from the vendor of the car manufacturing company. We investigated the series of extensive production of exhaust manifold manufacturing using the automated manufacturing system. We selected the problematic fragment of the entire manufacturing process of the exhaust manifold and this analysis can be performed using our defined approach. As far as the verification is concerned, we start to describe the control panel by how the user interact with machine and several users are assisted and controlled by the supervisor. Also, the supervisor will interact with the machine and display inform to the user about their task and tells the supervisor the behavior of the machine.

6. Current Interface Description

The relevant elements of the computer-numerically controlled (CNC), Robot (low weight material transfer) control panel and the electronic attitude display mode in a CNC machine are shown in Figures 6 and 7. In the robot controller we will discuss about the function of the on/off switch and in the user controller we will discuss about the five switches controlling both control panel and display mode as shown in Figure 7 that are interest of us. These five switches are (1) tool change for machining operation; (2) coordinate setting for part manufacturing; (3) start cycle button; (4) move to bin; (5) take part from bin and are easily operated through the display mode. These five modes for user and one mode for supervisor can be engaged by pressing the respective buttons on the CNC and robot control panel as shown in Figures 6 and 7.

In the CNC machine the selection of the five operations is made on the top portion of the control panel operated by user having a small window as the display mode, indicating the tool change, start operation, movement of parts into the bin and dimension set by the user. The user can change the dimension by entering the values through the x, y and z button at the panel side. We can also adjust the speed of the spindle by the feed rate switch and edit the program as per the user or customer requirements as shown in Figure 7. The procedure for iteration 1 is given in the subsections.

Figure 6. (**a**) Computer-numerically controlled (CNC) milling control panel; (**b**) Controller switch.

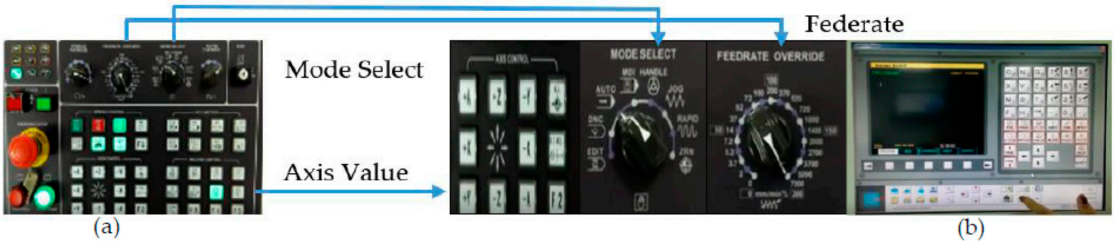


Figure 7. (**a**) Switches of computer-numerically controlled (CNC) milling control; (**b**) Display mode.

6.1. Modeling the Rest of the System

The exhaust manifold manufacturing process is operated simultaneously in two ways; the first part is highly automated with no human involvement while the other part is operated by several users linked with the supervisor interacting with the machine. This two-way manufacturing process will be able to manufacture the exhaust manifold for the Corolla (EM1), Cuore (EM2) and innovative international multi-purpose vehicle (IMV) Hilux (EM3) car variants. We are considering here only the human involvement with machine. To complete the formal system model, for the system operational environment we created at the formal representation of a three-user model, three machine models, a supervisor model and supervisor machine model for exhaust manifold manufacturing system. The model representation is for more readable, expressive and includes the choices for the handling of non-determinism, so we can use easily the timed automaton transition systems.

6.2. Transition Definition

α_{SU1}: Supervisor transmit information to user 1 to perform the operations of turning {take first EM1 part then EM2 then EM3 from bin 1 to perform turning operations}, Facing {take first EM1 part then EM2 then EM3 part to perform the facing operation}and then perform the reaming and boring operation on EM1, EM2 and EM3 parts. After performing the turning operation on EM1, EM2 and EM3 user will move the few EM1 part into the bin2 similarly, user will move the few EM2 part into the bin 2 after finishing the facing operation. The few EM3 parts will receive from the bin 3 to perform the reaming operation and then part EM1, EM2 and EM3 will move to the bin 4 after finishing the boring operation.

α_{SU2}: Supervisor transmit the information to user 2 to perform the operations of indexing, knurling, and taping on EM1, EM2 and EM3 then few EM1 and EM2 parts move to bin 6 after indexing operation and few EM3 parts move to bin 7 after knurling operation. After performing the taping operation on EM1, EM2 and EM3 parts will move to bin 8.

$(\alpha_{SU2})^m$: Supervisor transmit the information to user 2 to perform the operations of indexing1 and 2, knurling, and taping on EM1, EM2 and EM3 then few EM1 and EM2 parts move to bin 6 after indexing 2 operations. Few EM3 parts move to bin 7 after knurling operation. After performing the taping operation on EM1, EM2 and EM3 parts will move to bin 8.

α_{SU3}: Supervisor transmits the information to user 3 to perform the operations of boring on EM1, EM2 and EM3 one by one or their availability. After performing the boring operation, the user will move the EM1, EM2 and EM3 parts for performing the reaming operation and move to bin 11. After this operation the user will perform the threading operation on EM1, EM2 and EM3. While few EM3 parts were received from bin 10 to perform the threading operation. Finally, the reaming operations were performed on EM1, EM2 and EM3 then move to bin 11.

$(\alpha_{SU3})^m$: Supervisor transmits the information to user 3 to perform the operations of boring on EM1, EM2 and EM3 one by one or their availability. After performing the boring 1 operation, the user will move the EM1, EM2 and EM3 parts for performing the reaming operation. After that the user will perform the boring 2 operations on EM1, EM2 and EM3. While few EM3 parts were received from bin 10 to perform the threading operation on all parts. Finally, the reaming operations were performed on EM1, EM2 and EM3 then move to bin 11.

α_{SU4}: Parts are moved after facing operation to perform Indexing operation.

$(\alpha_{SU4})^m$: Parts are moved after facing operation to perform Indexing 1operation.

α_{SU5}: Parts are moved after facing operation to perform boring operation.

$(\alpha_{SU5})^m$: Parts are moved after facing operation to perform boring1 operation.

α_{SU6}: Parts are moved after boring operation to perform reaming operation.

$(\alpha_{SU6})^m$: After finishing the boring 2 operations to perform the reaming operation {Reaming = 1 mm}.

α_{SU7}: Parts are moved after indexing operation to perform reaming operation.

$(\alpha_{SU7})^m$: Parts are moved after indexing 2 operations to perform reaming operation.

$\eta_{1,1}$: Take the part from bin1 one by one in ordering of first EM1, EM2 and EM3 or if and only if one of them is exists then perform the turning operation. The depth of cut is 3 mm.

$\eta_{1,2}$: The few EM1 part will move to bin 2.

$\eta_{1,3}$: The few EM3 part will move to reaming operation.

$\eta_{1,4}$: The EM1, EM2 and EM3 part will move to bin 4.

$\beta_{1,1}$: Part is moving into the facing operation {Reduced length = 3 mm}.

$\beta_{1,2}$: Part is moving into the reaming operation {R = 1 mm}.

$\beta_{1,3}$: Part is moving into the boring operation {Depth of cut = 7 mm}.

β_{S1}: Parts are moved through conveyor to bin 2 from bin 6.

$\alpha_{1,1}$: The few EM2 part will move to bin 2.

$\alpha_{1,2}$: Take the part after performing the facing operation then go to the boring operation.

$\eta_{2,1}$: Take the part from bin 5 one by one in the ordering of first EM1, EM2 and EM3 or if and only if one of them is exists there then perform the indexing operation {Index = 6}.

$\eta_{2,2}$: Part will move to bin 6.

$\eta_{2,3}$: The Part will move to bin 6 at machine and bin 7 for supervisor model.

$\eta_{2,4}$: Part will move to bin 7 at machine and part will move for bin 8 for supervisor model.

$\eta_{2,5}$: Part will move to Bin 8.

$\beta_{2,1}$: Part is moving into the knurling operation {Length = 18 mm}.

$\beta_{2,2}$: Part is moving into the taping operation {Length = 8 mm}.

$\beta_{2,1b}$: Part is moving into the indexing 2 operation {Index = 6}.

β_{S2}: Parts are moved through conveyor to bin 6 from bin 2.

$\eta_{3,1}$: Take the part from bin 9 one by one in the ordering of first EM1, EM2 and EM3 or if and only if one of them is exists there then perform the boring operation {Doc = 3 mm}.

$\eta_{3,2}$: After finishing the boring operation to perform the reaming operation {Reaming = 1 mm}.

$\eta_{3,3}$: Part will be moved for threading operation {Thread Length = 8 mm}.

$\eta_{3,4}$: EM1, EM2 and EM3 parts will be placed into the bin11.

$\beta_{3,1}$: Part is moving to the threading operation {Thread Length = 8 mm}.

$\beta_{3,2}$: Part is moving to reaming operation {R = 1 mm for finishing}.

$\beta_{3,1b}$: Part is moving to boring 2 operations {Depth of cut = 3 mm}.

β_{S3}: Parts are moved through conveyor to bin 7 from bin 3.

β_{S4}: Parts are moved through conveyor to bin 10 from bin 7.

6.3. Supervisor Model

In the supervisor model, we have only one supervisor and the supervisor will provide information to all users to perform the needed task as per the production schedule. The supervisor switches on the robot to transfer the part to its dedicated place. The parts will be moved by robot after the indexing operation to perform the reaming operation. Similarly, this operation is needed on those parts that have already under- gone the boring operation of user model 3.

While, after the facing operation the parts will move to indexing and boring operation. The supervisor events are $(\sum_S^{com})_{choice=1} = \{\}$, $(\sum_S^{com'})_{choice>1}\{\alpha_{SU1}, \alpha_{SU2}, \alpha_{SU3}, \alpha_{SU4}, \alpha_{SU5}, \alpha_{SU6}, \alpha_{SU7}\}$.

6.4. Supervisor Machine Model

The supervisor will switch on the conveyor to transfer the part to its dedicated place and the parts will move from bin 6 to bin 2 and vice versa. Similarly, the parts move from bin 3 to bin 7 and bin 7 to bin 10. The user only places the part on to the conveyor and all the parts are moved through the conveyor which is controlled by the supervisor as per their company production plan. The dashed line is used for the machine transition and the dark lines represent the supervisor task as shown in Figures 8 and 9. The supervisor machine events are $\sum_S^{obs} = \{\beta_{S1}, \beta_{S2}, \beta_{S3}, \beta_{S4}\}$ and $\sum_S^{int} = \{\}$. The procedure of iteration 2 is given in the Section 7 and their subsections.

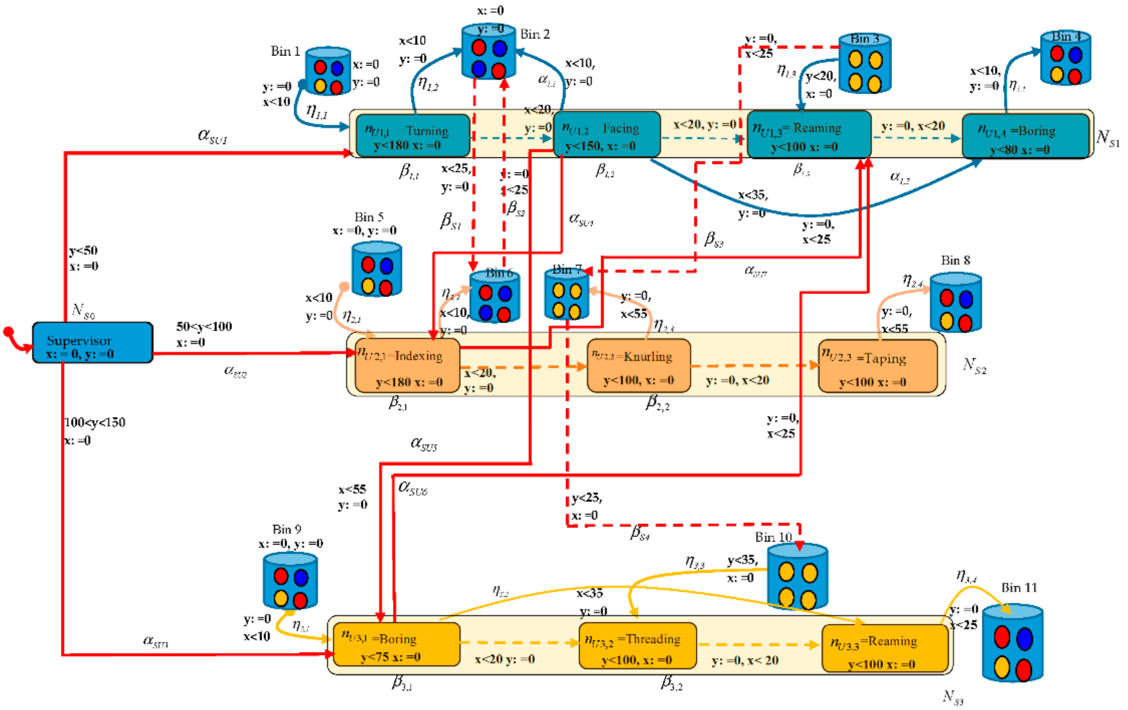


Figure 8. Supervisor Model.

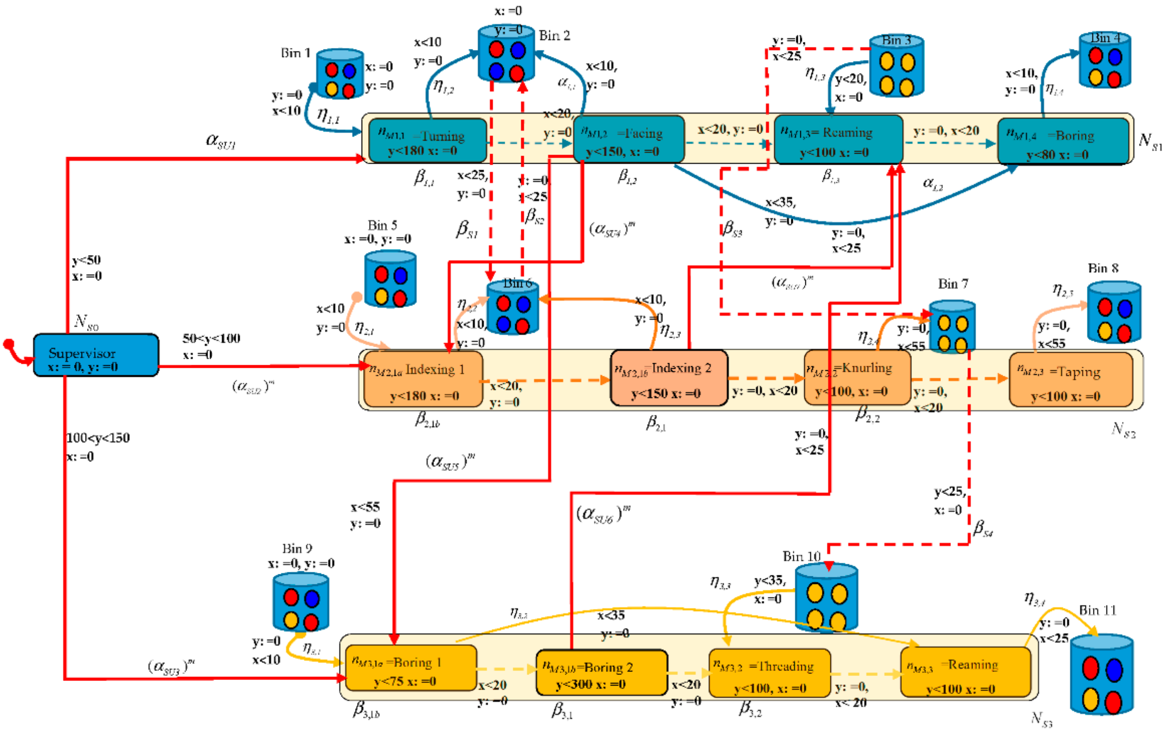


Figure 9. Supervisor Machine Model.

7. Discussion

Formal verification was performed using the model composition of user with machine and supervisor with a machine. For all user models, two of the timed automaton-based formalisms indicated that the acts of user model 2 and user model 3 reset-ability, act and skip-ability, every part of these user model as shown in Figure 8 and its associated time transitions between execution states was not reachable. Similarly, n_{S6} and n_{S7} in the supervisor model the supervisory task was unable to reach their target state as per customer requirements. Indicating that a conflicting mode arises which does not fulfil the user and the supervisor demand. Thus the complete-ability [35] will not be evaluated correctly, indicating that the manufacturing process of the exhaust manifold observed by user with machine and supervisor with the supervisor machine model composition. This could go for the exhaust gas leakage inside the engine compartment to create malfunction during driving of the car. In the user model the event sets $\eta_{2,3}$ and $\eta_{3,2}$ have an issue with the interaction of the machine due to their unmatched state compositions. Similarly, supervisor has also an unmatched state with their machine during the execution of α_{SU6} and α_{SU7} task. This information was not mentioned inside the user manual. In that case the interface should be correct for the user and supervisor and there is a need to update the user and supervisor manual to execute the task by user and supervisor as per customer needs.

7.1. Formal Verification Results

Formal verification was done on a LG computer in succession with Linux Mint 18 with an intel core i7, and 64 giga bytes of RAM, Ansan, Korea with the aid of SAL's symbolic model checker open source. It acquired 19.86 s of entire completing time to validate all 68 obtained properties with 1496 numbered as the determined staying at states for verification. We also used the Z-eves tool for model verification using proof. Further, there is no counterexample formed.

7.2. Scalability

We have no observation for substantial growth during the verification times and states pace magnitudes among the normal behavior of model holding at 2 min 38 s and 3,062,072 states producing scheme obtainable here does scale fit. Therefore, the scheme may be suitable for the examination of considerable bigger systems. Forthcoming work should explore how this technique should implement for different scenario. The procedure for iteration 3 and 4 are given in Section 8 and their subsection.

8. Interface of User Model and Supervisor Model

We rectified the supervisor interface as shown in Figure 10, interface of user model 2 as shown in Figure 11 and the user model 3 interfaces as shown in Figure 12 using the weak bi-simulation approach. This interface of user and the supervisor will be co-reachable because the weak bi-simulation preserves this property. We already implemented this interface after we received feedback from the customer in favor of the product. Further, this interface allows the user and supervisor to handle the non-deterministic choices with the help of time transition in user and the supervisor model as shown in Figures 10–12.

Operational incidents occur when the part is fitted into the vehicle and then after few weeks, leakage is identified by the customer. Normally this type of complain would not appear on the plant and dealer side. However, the customer feedback motivates us to improve the quality of product inside the plant to make the product as per the standard criteria. Similarly, the set pipe hose has no leakage after the improvement in the interface model of user and the supervisor and further the malfunctioning of the solenoid valve was not observed, and it performs well.

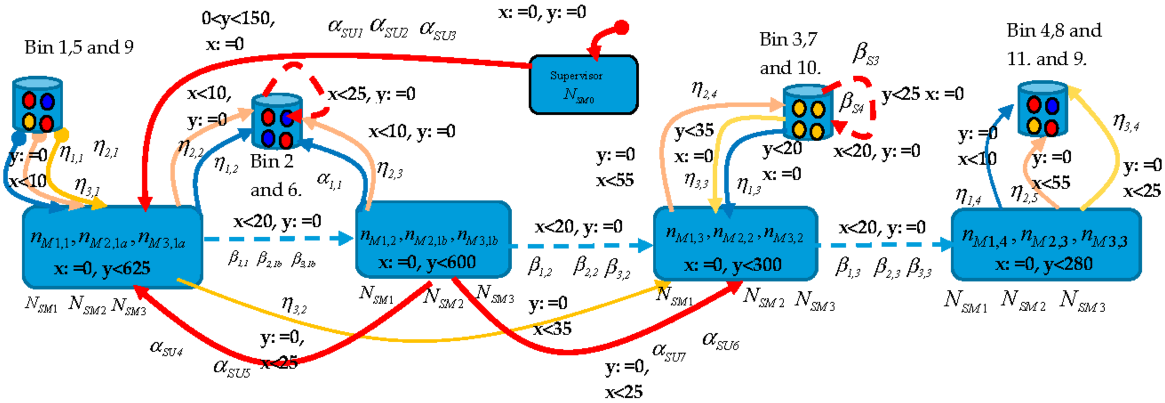


Figure 10. Supervisor Interface.

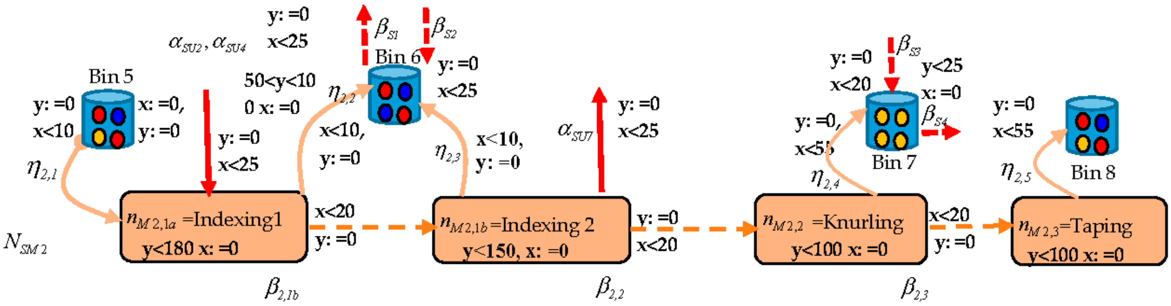


Figure 11. Interface Model of User #2.

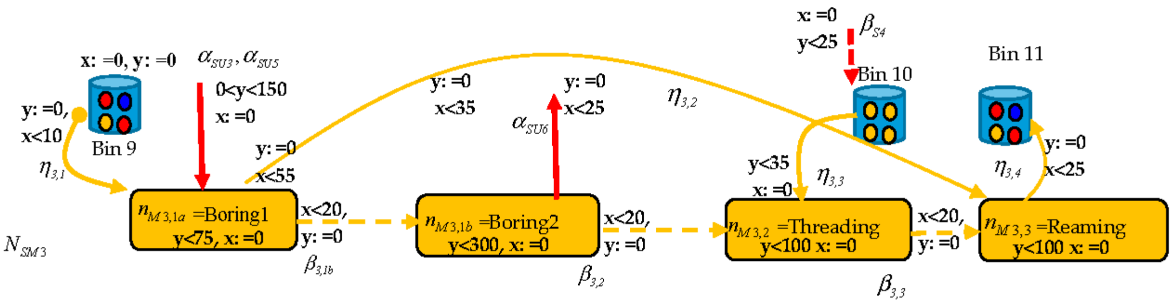


Figure 12. Interface Model of User#3.

9. Conclusions

The design of an interface based on user understanding about the systems is not as stable as manufacturing systems and product quality demand, due to several user-machine interactions aligned with the supervisor when the process of manufacturing is ongoing. However, a general observation can still be drawn about their relative performance. To compare with those needs, we developed a formal framework for the analysis of human-computer interaction systems modelling based on time automaton. Compared with other models [36], our novel approach describes the formal models of user and supervisor activities with their machine behavior by adopting the modelling techniques of time automaton with full control and mode preserving. Also, we propose a technique to generate the supervisor interface based on multiple users with a machine and user interface through weak bi-simulation. Moreover, our technique has the potential to evaluate the interaction in real time and we also discuss how these techniques can be adapted to consider information about the machine and user states to solve for non-deterministic choices. We used z-eves for analyze and validate the formal specification of supervisor, machine and weak bi-simulation relation. We used the iteration-based approach to validate the interactive systems by using weak bi-simulation through checking of two systems simultaneously using z-eves to generate the correct interface. We implemented our technique on case study of a transmission gear not shifting at more than 189 km/h. Each treatment-created specification property designed the estimated consequence on behalf of that, for all its related transitions among finishing states was accessible. Further, there is no counterexample formed. For future perspectives, a possible extension is to add information about the environment and a cognitive model of the system and user. Such information constitutes a user and supervisor state-based interface. The supervisor and user models for such systems can be advanced in the sense that their transitions should be well defined. This will raise some issues related with observation of the user and supervisor state-interface meaning that both should know the previous interface observation. Defining the generation of such an interface is a possible extension of this work. Finally, with this method the interfaces are more expressive and understandable, and improvements in product quality and customer response have also been achieved.

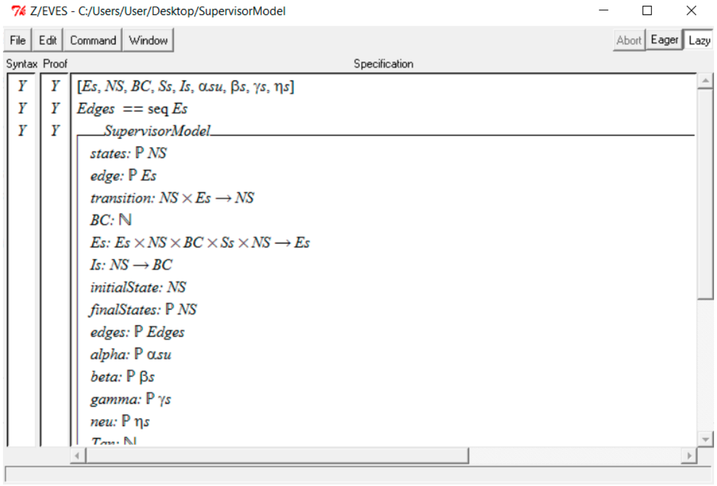
Acknowledgments: The work described in this paper was funded by HEC Pakistan and supported by Hanyang University, South Korea. This basic research and technology are the effort on human-automation interaction as well as the Customer Satisfaction with high quality products. Scientists wrote the first version of the formal composite model of single user and machine [18]. Scientists described the concept of user based generated events and modify the concept of single user interface by building up mathematical tool using simulation technique [20]. Finally, we give thanks to reviewers for providing us the nice suggestions, time and effort to improve my papers.

Author Contributions: The mathematical models were developed by Shazada Muhammad Umair Khan. Further he also executes the case study. Wenlong He Analyze the literature review and finalized the formatting of this manuscript. Wenlong He also helps to understand the logic of case study and provide information for its better solution.

Conflicts of Interest: The author declares that there is no conflict of interest regarding the publication of this paper.

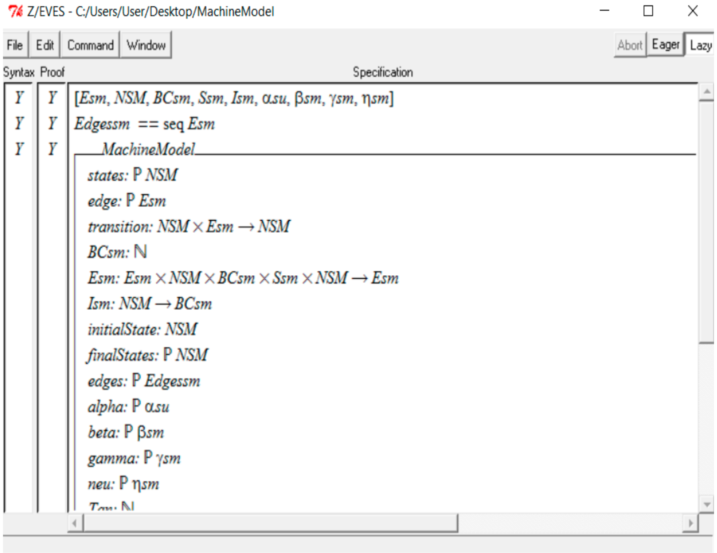
Appendix A

The validation of formal specification for supervisor model, machine model and weak bi-simulation are given as a snapshot in Figures A1–A3 respectively. In these figures, there is limitation in the symbol representation using z editor, likewise in supervisor model we used the symbol N_S for supervisor state but in z notation we used N_S. Similar approach is applied for all other symbols. The symbol which are different from their model representation are also explained clearly. The subscript we used for supervisor model is 's' and for machine model we used 'sm'. In z representation the symbol $\sum_S$ used as Ss, $(\sum_S^{com})_{choice=1}$ used as Sscomch, $(\sum_S^{com'})_{choice>1}$ used as Sscomchgo for machine we used Ssmcomchgo. The symbol $\sum_S^{obs}$ used as Ssobs for machine we used Ssmobs. The symbol $\sum_S^{int}$ used as Ssint for machine we used Ssmint. Similarly, the symbol $\sum_M^{cor}$ used as Sscor for machine we used Ssmcor, $\sum_M^{rch}$ used as Ssrch for machine we used Ssmrch. Finally, the symbol $\sum_{SM}^{int}$ used as Ssint for machine we used Ssmint and e_M^{cor} used as emcor respectively.



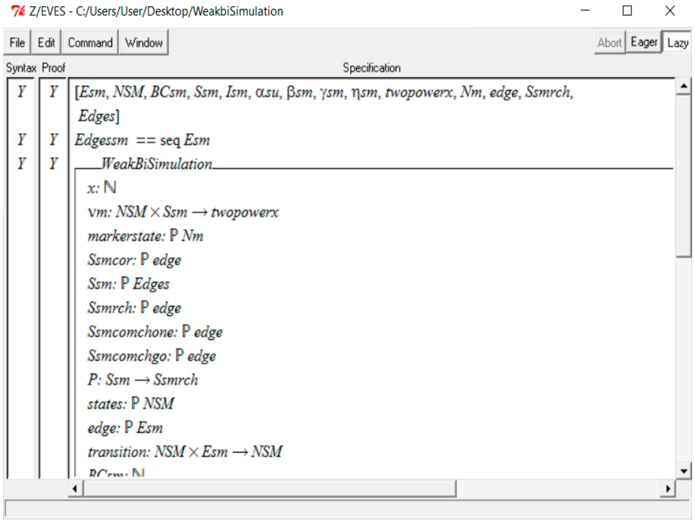
```
Y  Y  [Es, NS, BC, Ss, Is, αsu, βs, γs, ηs]
Y  Y  Edges == seq Es
Y  Y  SupervisorModel
        states: ℙ NS
        edge: ℙ Es
        transition: NS × Es → NS
        BC: ℕ
        Es: Es × NS × BC × Ss × NS → Es
        Is: NS → BC
        initialState: NS
        finalStates: ℙ NS
        edges: ℙ Edges
        alpha: ℙ αsu
        beta: ℙ βs
        gamma: ℙ γs
        neu: ℙ ηs
        Tau: ℕ
```

Figure A1. The snapshot of the formal specification analysis of supervisor model.



```
Y  Y  [Esm, NSM, BCsm, Ssm, Ism, αsu, βsm, γsm, ηsm]
Y  Y  Edgessm == seq Esm
Y  Y  MachineModel
        states: ℙ NSM
        edge: ℙ Esm
        transition: NSM × Esm → NSM
        BCsm: ℕ
        Esm: Esm × NSM × BCsm × Ssm × NSM → Esm
        Ism: NSM → BCsm
        initialState: NSM
        finalStates: ℙ NSM
        edges: ℙ Edgessm
        alpha: ℙ αsu
        beta: ℙ βsm
        gamma: ℙ γsm
        neu: ℙ ηsm
        Tau: ℕ
```

Figure A2. The snapshot of the formal specification analysis of machine model.



```
Y  Y  [Esm, NSM, BCsm, Ssm, Ism, αsu, βsm, γsm, ηsm, twopowerx, Nm, edge, Ssmrch,
       Edges]
Y  Y  Edgessm  == seq Esm
Y  Y  ──WeakBiSimulation──────────────
       x: ℕ
       vm: NSM × Ssm → twopowerx
       markerstate: ℙ Nm
       Ssmcor: ℙ edge
       Ssm: ℙ Edges
       Ssmrch: ℙ edge
       Ssmcomchone: ℙ edge
       Ssmcomchgo: ℙ edge
       P: Ssm → Ssmrch
       states: ℙ NSM
       edge: ℙ Esm
       transition: NSM × Esm → NSM
       BCsm: ℕ
```

Figure A3. The snapshot of the formal specification analysis of weak bi-simulation model.

References

1. Saraph, J.V.; Sebastian, R.J. Human resource strategies for effective introduction of advanced manufacturing technologies (AMT). *Prod. Inventory Manag. J.* **1992**, *33*, 64.
2. Weyers, B.; Bowen, J.; Dix, A.; Palanque, P. *The Handbook of Formal Methods in Human-Computer Interaction*; Springer: Berlin, Germany, 2017.
3. Cheng, R.; Zhou, J.; Chen, D.; Song, Y. Model-based verification method for solving the parameter uncertainty in the train control system. *Reliab. Eng. Syst. Saf.* **2016**, *145*, 169–182. [CrossRef]
4. Rezazadeh, I.M.; Wang, X.; Firoozabadi, M.; Golpayegani, M.R.H. Using affective human–machine interface to increase the operation performance in virtual construction crane training system: A novel approach. *Autom. Constr.* **2011**, *20*, 289–298. [CrossRef]
5. Bolton, M.L.; Siminiceanu, R.I.; Bass, E.J. A systematic approach to model checking human–automation interaction using task analytic models. *IEEE Trans. Syst. Man Cybern. Part A* **2011**, *41*, 961–976. [CrossRef]
6. Wang, X.V.; Wang, L.; Mohammed, A.; Givehchi, M. Ubiquitous manufacturing system based on Cloud: A robotics application. *Robot. Comput. Integr. Manuf.* **2017**, *45*, 116–125. [CrossRef]
7. Wan, J.; Tang, S.; Li, D.; Wang, S.; Liu, C.; Abbas, H.; Vasilakos, A.V. A Manufacturing Big Data Solution for Active Preventive Maintenance. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2039–2047. [CrossRef]
8. Thurman, D.A.; Mitchell, C.M. An apprenticeship approach for the development of operations automation knowledge bases. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* **2000**, *44*, 231–234. [CrossRef]
9. Oishi, M.M.; Tilbury, D.; Tomlin, C.J. Guest Editorial Special Section on Human-Centered Automation. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 4–6. [CrossRef]
10. Meisels, I.; Saaltink, M. *The Z/EVES Reference Manual (for Version 1.5)*; Reference Manual TR-97-5493-03; ORA: Ottawa, ON, Canada, 1997.
11. Woodcock, J.; Davies, J. *Using Z: Specification, Refinement, and Proof*; Prentice Hall Englewood Cliffs: Oxford, UK, 1996.
12. Maynard, M. *Issue: Global Manufacturing Global Manufacturing*; Sage Businessresearcher: New York, NY, USA, 2015.
13. Wu, Y.; Boyle, L.N.; McGehee, D.; Roe, C.A.; Ebe, K.; Foley, J. Foot placement during error and pedal applications in naturalistic driving. *Accid. Anal. Prev.* **2017**, *99*, 102–109. [CrossRef] [PubMed]
14. Degani, A.; Kirlik, A. Describing the design contributors to mode error. In Proceedings of the Fourth Annual Symposium on Human Interaction with Complex Systems, Dayton, OH, USA, 22–25 March 1998; pp. 112–115.
15. Wickens, C.D. Automation in air traffic control: The human performance issues. In *Automation Technology and Human Performance*; Lawrence Erlbaum Associates: Mahwah, NJ, USA, 1999; pp. 2–10.

16. Hilbert, D.M.; Redmiles, D.F. Extracting usability information from user interface events. *ACM Comput. Surv.* **2000**, *32*, 384–421. [CrossRef]
17. Torney, H.; O'Hare, P.; Davis, L.; Delafont, B.; Bond, R.; McReynolds, H.; McLister, A.; McCartney, B.; Di Maio, R.; McEneaney, D. A usability study of a critical man–machine interface: Can layperson responders perform optimal compression rates when using a public access defibrillator with automated real-time feedback during cardiopulmonary resuscitation? *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 749–754. [CrossRef]
18. Degani, A.; Heymann, M. Formal verification of human-automation interaction. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2002**, *44*, 28–43. [CrossRef] [PubMed]
19. Suzuki, A.; Ushio, T.; Adachi, M. Detection of automation surprises in discrete event systems operated by multiple users. In Proceedings of the International Joint Conference, SICE-ICASE, Busan, Korea, 18–21 October 2006; pp. 1115–1118.
20. Adachi, M.; Ushio, T.; Ukawa, Y. Design of user-interface without automation surprises for discrete event systems. *Control Eng. Pract.* **2006**, *14*, 1249–1258. [CrossRef]
21. Nachreiner, F.; Nickel, P.; Meyer, I. Human factors in process control systems: The design of human–machine interfaces. *Saf. Sci.* **2006**, *44*, 5–26. [CrossRef]
22. Cummings, M.L.; Clare, A.; Hart, C. The role of human-automation consensus in multiple unmanned vehicle scheduling. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2010**, *52*, 17–27. [CrossRef] [PubMed]
23. Clarke, E.M.; Grumberg, O.; Peled, D. *Model Checking*; MIT Press: Cambridge, MA, USA, 1999.
24. Parasuraman, R.; Sheridan, T.B.; Wickens, C.D. A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cybern. Part A* **2000**, *30*, 286–297. [CrossRef]
25. Saleh, L.; Chevrel, P.; Claveau, F.; Lafay, J.-F.; Mars, F. Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 974–983. [CrossRef]
26. Heymann, M.; Degani, A. Formal analysis and automatic generation of user interfaces: Approach, methodology, and an algorithm. *Hum. Factors* **2007**, *49*, 311–330. [CrossRef] [PubMed]
27. Pérez, A.; García, M.I.; Nieto, M.; Pedraza, J.L.; Rodríguez, S.; Zamorano, J. Argos: An advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 463–473. [CrossRef]
28. Wei, Z.; Zhuang, D.; Wanyan, X.; Liu, C.; Zhuang, H. A model for discrimination and prediction of mental workload of aircraft cockpit display interface. *Chin. J. Aeronaut.* **2014**, *27*, 1070–1077. [CrossRef]
29. Gachui, N.A. Wizard Navigation Functionality to Automated User Interfaces Using Finite State Machines. Master's Thesis, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Keneya, 2016.
30. Wonham, W.M. Supervisory control of discrete-event systems. In *Encyclopedia of Systems and Control*; Springer: London, UK, 2015; pp. 1396–1404.
31. Milner, R. *Communication and Concurrency*; Prentice Hall: New York, NY, USA, 1989; Volume 84.
32. De Moura, L.; Owre, S.; Shankar, N. The SAL language manual. In *Computer Science Laboratory*; Technical Report CSL-01-01; SRI International: Menlo Park, CA, USA, 2003.
33. Shankar, N. Symbolic Analysis of Transition Systems. In Proceedings of the International Workshop on Abstract State Machines, Theory and Applications, London, UK, 19–24 March 2000; pp. 287–302.
34. Lin, Y.-D.; Liao, F.-Z.; Huang, S.-K.; Lai, Y.-C. Browser fuzzing by scheduled mutation and generation of document object models. In Proceedings of the 2015 International Carnahan Conference on Security Technology (ICCST), Taipei, Taiwan, 21–24 September 2015; pp. 1–6.
35. Bolton, M.L.; Jiménez, N.; van Paassen, M.M.; Trujillo, M. Automatically generating specification properties from task models for the formal verification of human–automation interaction. *IEEE Trans. Hum. Mach. Syst.* **2014**, *44*, 561–575. [CrossRef]
36. Bolton, M.L.; Bass, E.J. Using model checking to explore checklist-guided pilot behavior. *Int. J. Aviat. Psychol.* **2012**, *22*, 343–366. [CrossRef]