Neural, Parallel, and Scientific Computations, **26**, No. 1 (2018), 65-74 ISSN: 1056-2176

MODELING AND VERIFICATION ANALYSIS OF A FLEXIBLE MANUFACTURING SYSTEM: A MODAL LOGIC APPROACH

ZVI RETCHKIMAN KONIGSBERG

Centro de Investigacion en Computacion
Instituto Politecnico Nacional
Mexico D.F, MEXICO

ABSTRACT: A flexible manufacturing system is an efficient production line with versatile machines, an automatic transport system and a sophisticated decision making system. This paper proposes a formal modeling and verification analysis methodology, which consists in representing the flexible manufacturing system by means of a modal logic formula. Then, using the concept of logic implication, and transforming this logical implication relation into a set of clauses, a modal resolution qualitative method for verification (satisfiability) as well as performance issues, for some queries is applied.

AMS Subject Classification: 08A99, 93D35, 93D99, 39A11
Key Words: flexible manufacturing system, modal logic, model, verification, unsatisfiability, modal resolution method

Received: December 11, 2017 ; **Accepted:** March 3, 2018 ;
Published: April 26, 2018 **doi:** 10.12732/npsc.v26i1.4
Dynamic Publishers, Inc., Acad. Publishers, Ltd. https://acadsol.eu/npsc

1. INTRODUCTION

A flexible manufacturing system is an efficient production line with versatile machines, an automatic transport system and a sophisticated decision making system. Flexible manufacturing systems can be formed by subsystems that work concurrently. In this paper a multirobotic system, an example of a flexible manufacturing system, is presented. The multirobotic system consists of two robot arms which perform pick and place operations accessing a common workspace at times to obtain or transfer parts. It is assumed that the common workspace has a buffer with a limited space for products. the process represents the operation of the two robots serving two different machining tools, with one robot arm transferring products from one machining tool to the buffer, and the other robot arm transferring semiproducts from the buffer to the other machining tool. This paper proposes a well defined syntax modeling and verification analysis methodology which consists in representing the flexible manufacturing system as a modal logic formula. Other non-classical methodologies as Petri nets and propositional logic have been employed too ([1] and [2]). The modal logic approach introduces two new operators that enable abstract relations like necessarily true and possibly true to be expressed directly, called alethic modalities, what is not possible using first order logic. For example, the statement: 7 is a prime number, is necessarily true always and everywhere, in contrast, the statement the head of state of this country is a king is possibly true, because its truth changes from place to place and from time to time. Other modalities that have been formalized in modal logic include temporal modalities, or modalities of time, deontic modalities, epistemic modalities, and doxastic modalities.

The main idea consists in modeling the flexible manufacturing system by means of a modal logic formula. Then, using the concept of logic implication, and transforming this logical implication relation into a set of clauses, a modal resolution qualitative method for verification (satisfiability) as well as performance issues, for some queries is applied. The paper is organized as follows. In section 2, a modal logic background summary is given. In section 3, the modal resolution principle for unsatisfiability is recalled. In section 4, the flexible manufacturing system is addressed. Finally, the paper ends with some conclusions.

2. MODAL LOGIC BACKGROUND

This section presents a summary of modal logic theory. The reader interested in more details is encouraged to see [3, 4].

Definition 1. A modal language $\mathcal{L}$ is an infinite collection of distinct symbols, no one of which is properly contained in another, separated into the following categories: parentheses, connectives, possibility modality, necessity modality, proposition variables $\Phi_0 = \{p_1, p_2, \cdots\}$ (called atoms), contradiction(falsity), true(tautology).

Definition 2. Well-formed formulas, or formulas for short, in modal logic are defined recursively as follows:(i). An atom is a formula, $\perp$ (false is a formula), T (true is a formula) (ii). If F and G are formulas then, $\sim (F)$, $(F \vee G)$, $(F \wedge G)$, $(F \leftrightarrow G)$, $\Box F$, $\diamond F$, are formulas. $\diamond A \equiv \sim \Box \sim A$. Formulas are generated only by a finite number of applications of (i) and (ii), therefore the set of welled formed formulas is enumerable infinite.

Remark 3. *It is important to underline the unique readability of the formulas which is secured by the assumption that the operators are one to one.*

Definition 4. A Kripke frame (frame) $\mathcal{F}$ is a pair $(W, \mathcal{R})$ in which W is a set of worlds (time, states, etc), and $\mathcal{R} \subseteq W \times W$ is a binary relation over W.

Definition 5. A Kripke model (model) $\mathcal{M}$ over frame $\mathcal{F}$ is a triple $(\mathcal{F}, \pi) = (W, \mathcal{R}, \pi)$ where $\pi : \Phi_0 \to 2^W$ the set of worlds where each element of Φ_0 is true is an assignment or interpretation.

Definition 6. Given any model $\mathcal{M}$, a world $w \varepsilon W$, the notion of true at w is defined as follows:

- $\mathcal{M}, w \models p_n \Leftrightarrow w \varepsilon \pi(p_n), n = 1, 2, \cdots;$
- $\mathcal{M}, w \models \sim F \Leftrightarrow w \not\models F;$
- $\mathcal{M}, w \models F \wedge G \Leftrightarrow w \models F$ and $w \models G;$
- $\mathcal{M}, w \models F \vee G \Leftrightarrow w \models F$ or $w \models G;$
- $\mathcal{M}, w \models F \to G \Leftrightarrow$ if $w \models F$ then $w \models G;$
- $\mathcal{M}, w \models F \equiv G \Leftrightarrow w \models F$ iff $w \models G;$
- $\mathcal{M}, w \models \diamond F \Leftrightarrow$ there exists $u \varepsilon W$ such that $(w, u) \varepsilon \mathcal{R}, \mathcal{M}, u \models F;$
- $\mathcal{M}, w \models \Box F \Leftrightarrow$ for all $u \varepsilon W$ such that $(w, u) \varepsilon \mathcal{R}, \mathcal{M}, u \models F.$

Definition 7. A formula F is consistent (satisfiable, true at w) in a model $\mathcal{M}$ in a world $w \varepsilon W$ iff $\mathcal{M}, w \models F$, then we say that $\mathcal{M}$ is a model for F. If this happens for all worlds $w \varepsilon W$ then we say it is true.

Definition 8. A formula F is inconsistent (unsatisfiable) in a model $\mathcal{M}$ iff $\mathcal{M}, w \not\models F$ for every world $w \varepsilon W$, then we say that $\mathcal{M}$ is a countermodel for F.

Definition 9. A formula F is valid in a class of models $\mathcal{CM}$ if and only if it is true for all models in the class. This will be denoted by $\models_{\mathcal{CM}} F$.

Definition 10. A formula F is valid iff it is valid for every class of models $\mathcal{CM}$. This will be denoted by $\models F$.

Definition 11. A formula G is a logical implication of formulas $F_1, F_2, \ldots, F_n$ if and only if for every model $\mathcal{M}$, that makes $F_1, F_2, \ldots, F_n$ true, G is also true in $\mathcal{M}$.

The following characterization of logical implication plays a very important role as will be shown in the rest of the paper.

Theorem 12. *Given formulas $F_1, F_2, \ldots, F_n, G$, G is a logical implication of F_1, F_2, $\ldots$, F_n if and only if the formula $((F_1 \wedge F_2 \wedge \ldots, \wedge F_n) \to G)$ is valid in a class of models if and only if the formula $(F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge \sim (G))$ is unsatisfiable.*

Proof. Setting the class of models equal to all the models that make $F_1 \wedge F_2 \wedge \ldots, \wedge F_n$ true. The first iff follows directly by the definition of validity in a class of models, and logical implication. For the second one, since $F_1 \wedge F_2 \wedge \ldots, \wedge F_n \to G$ is valid in a class of models, every model that makes $F_1 \wedge F_2 \wedge \ldots, \wedge F_n$ true does not satisfy $\sim (G)$, therefore $(F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge \sim (G))$ can not be satisfied. Reversing this last argument we obtain the last implication. □

Next, given a class of models $\mathcal{CM}$, we define the syntactic mechanisms capable of generating the formulas valid on $\mathcal{CM}$.

Axioms:

(1) All instances of propositional logic tautologies.

(2) $\Box(F \to G) \to \Box F \to \Box G$.

Rules of inference: (1) Modus ponens

$$\frac{F, F \to G}{G}$$

(2) Necessitation

$$\frac{F}{\Box G}$$

We write $\vdash F$ if F can be deduced from the axioms and the inference rules.

Theorem 13. *(Completeness [3]) A formula F is valid iff it is provable i.e., $\models F \Leftrightarrow \vdash F$*

Definition 14. A formula F in modal logic is said to be in disjunctive normal form normal (DNF) if and only if is a disjunction (perhaps with zero disjunct) of the form $F == L_1 \vee L_2 \vee \cdots L_n \vee \Box D_1 \vee \Box D_2 \vee \cdots \Box D_m \vee \Diamond H_1 \vee \Diamond H_2 \vee \cdots \Diamond H_j$, where each L_i is an atom or its negation, each D_i is a DNF, and each H_i is a CNF (next defined). A formula G is said to be in conjunctive normal form (CNF) if it is a conjunction of F_i DNF i.e., $G = F_1 \wedge F_2 \wedge \cdots \wedge F_n$ which will be denoted by the set $G = \{F_1, F_2, \ldots, F_n\}$

Definition 15. A formula in DNF is called a clause. A clause with only one element is called a unit clause. A clause with zero disjunct is empty and it will be denoted by the $\perp$ symbol. Since the empty clause has no literal that can be satisfied by a model, the empty clause is always false.

Definition 16. The modal degree of a formula F denoted by $d(F)$ is recursively defined as follows:

- if F is a literal then its degree is zero;
- $d(F \triangle G) = max(d(F), d(G))$, where $\triangle$ is $\wedge$ or $\vee$;
- $d(\sim F)) = d(F)$;
- $d(\nabla F) = d(F) + 1$, where ∇ stands for $\Box$ or $\diamond$.

Given a formula F, the following inductive procedure transforms F into a CNF in such a way that the original formula is equal to its CNF form therefore satisfying validity. (1) Using axioms 1 and 2, the definition $\sim \Box F \equiv \diamond \sim F$ and the inference rules, eliminate all propositional other than $\wedge, \vee, \sim$ and move negations inside so that they are immediately before propositional variables, (2) If $d(F) = 0$ then apply the propositional procedure [5] , (3) If $F = \Box F_1$ with F_1 in CNF, apply the theorem $\Box(F \wedge G) \equiv \Box F \wedge \Box G$ to distribute the $\Box$ operator (this is proved with the aid of axiom 2). (4) If $F = \diamond F_1$ with F_1 in CNF, then do not do anything. (5) Otherwise, we have a combination of different formulas which can be handled using the preceding rules.

Therefore, we have proved the following result.

Theorem 17. *Let S be a set of clauses that represents a formula F in its CNF. Then F is unsatisfiable if and only if S is unsatisfiable.*

3. THE MODAL LOGIC RESOLUTION PRINCIPLE

We shall next present the resolution principle inspired by the propositional logic resolution principle introduced by Robinson (see [5], the references quoted therein, and [6]). It can be applied directly to any set S of clauses to test the unsatisfiability of S. Resolution is a decidable, sound and complete proof system i.e., a formula in clausal form is unsatisfiable if and only if there exists an algorithm reporting that it is unsatisfiable. Therefore it provides a consistent methodology free of contradictions. It is composed of rules for computing resolvents, simplification rules and rules of inference. The first ones compute resolvents, simplified by the simplification rules, and then inferred by the rules of inference.

Definition 18. [6] Let $\Sigma(A, B) \to C$, and $\Gamma(A) \to C$ be two relations on clauses defined by the following formal system:

Axioms:
(1). $\Sigma(p, \sim p) \rightarrow \perp$.
(2). $\Sigma(\perp, A) \rightarrow \perp$.
Σ rules:

$$\vee - \text{rule} : \frac{\Sigma(A,B) \rightarrow C}{\Sigma(A \vee D_1, B \vee D_2) \rightarrow C \vee D_1 \vee D_2}$$

$$\Box \diamond - \text{rule} : \frac{\Sigma(A,B) \rightarrow C}{\Sigma(\Box A, \diamond(B,E)) \rightarrow \diamond(B,C,E)}$$

$$\Box\Box - \text{rule} : \frac{\Sigma(A,B) \rightarrow C}{\Sigma(\Box A, \Box B) \rightarrow \Box C}$$

Γ rules:

$$\diamond - \text{rule} \quad 1 : \frac{\Sigma(A,B) \rightarrow C}{\Gamma(\diamond(A,B,F)) \rightarrow \diamond(A,B,C,F)}$$

$$\diamond - \text{rule} \quad 2 : \frac{\Gamma(A) \rightarrow B}{\Gamma(\diamond(A,F)) \rightarrow \diamond(B,A,F)}$$

$$\vee - \text{rule} : \frac{\Gamma(A) \rightarrow B}{\Gamma(A \vee C) \rightarrow B \vee C}$$

$$\Box - \text{rule} : \frac{\Gamma(A) \rightarrow B}{\Gamma(\Box A) \rightarrow \Box B}$$

where A, B, C, D, D_1, D_2, denote general clauses, E, F denote sets (conjunctions) of clauses, and $(A < E)$ denotes the result of appending the clauses A to the set E.

Simplification rules:
The relation 'A can be simplified in B' denoted $A \simeq B$ is the least congruence relation containing: (S1) $\diamond \perp \simeq \perp$, (S2) $\perp \vee D \simeq D$, (S3) $\perp, E \simeq \perp$, (S4) $A \vee A \vee D \simeq A \vee D$. The simplified formula obtained is called the normal form of the original formula and is the one to be considered when computing resolvents.

Inference rules:
(R1).

$$\frac{C}{D} \quad if \quad \Gamma(C) \rightarrow D$$

(R2).

$$\frac{C_1 \quad C_2}{D} \quad \text{if} \quad \Sigma(C_1, C_2) \rightarrow D,$$

where C, C_1, C_2, D are general clauses.
A deduction of a clause D from a set S of clauses can be seen as a tree whose root is D, whose leaves are clauses of S, and every internal node C has sons A and B

(respectively A) iff the rule R2 (respectively Rl) can be applied with premises A and B (respectively A) and conclusion C. The size of a deduction is the number of nodes of this tree. We say that D is a-consequence of S iff there is a deduction of D from S denoted by $S \vdash D$. These definitions and notations are extended to sets of consequences: if S' is a set of clauses, $S \vdash S^{'}$ iff $S \vdash D$ for every $D \varepsilon S^{'}$. A deduction of $\perp$ from S is a refutation of S.

Theorem 19. *[6] The resolution proof system is decidable.*

The main two results of this subsection: the completeness theorem for the resolution proof system, and that proofs in the resolution proof system are actually proofs in our modal logic axiomatic system are next presented.

Theorem 20. *[6] A set S of clauses is unsatisfiable if and only if there is a deduction of the empty clause $\perp$ from S.*

Theorem 21. *If there exists a deduction D from S in the resolution proof system then there is a deduction D from S in our modal logic axiomatic system.*

Proof. Let us proceed by induction on the size of the deduction. Base case: the deduction is an axiom i.e., it is either $\Sigma(p, \sim p) \rightarrow \perp$ or $\Sigma(\perp, A) \rightarrow \perp$ which in our modal logic system correspond to $p \wedge \sim p \rightarrow \perp$ and $\perp \wedge A \rightarrow \perp$ which are propositional logic tautologies. Next, let us assume that the conclusion holds for every deduction of size less than or equal to $k-1$. Then, we have a deduction of S_{k-1} from S in our modal logic system and a one length deduction in the resolution proof system of S_k from S_{k-1} which turns out to be also a deduction in our modal logic system (due to the induction hypothesis). Therefore concatenating both we get a deduction of S_k from S. □

Remark 22. *Indeed it is straightforward to show that one length deductions in the resolution proof system are proofs in our modal logic system, since they are the result of applying the Σ and Γ rules, see [6].*

4. FLEXIBLE MANUFACTURING SYSTEM

A flexible manufacturing system is an efficient production line with versatile machines, an automatic transport system and a sophisticated decision making system. Flexible manufacturing systems can be formed by subsystems that work concurrently. The multirobotic system consists of two robot arms which perform pick and place operations accessing a common workspace at times to obtain or transfer parts. In order to avoid collision, there is a priority protocol control unit that guarantees access to the common workspace to only one of the two robot arms. In the case that the priority protocol control unit breakdowns the whole system is reset to its starting point. It is assumed that the common workspace has a buffer with a limited space for products. the process represents the operation of the two robots serving two different machining tools, with one robot arm transferring products from one machining tool to the buffer, and the other robot arm transferring semiproducts from the buffer to the other machining tool.

The multirobotic system behavior as described as follows: (1) Propositional variables ($i = 1, 2$): R_iSP Robot i are at the starting point, R_iWAWP Robot i waits for access to the common workspace, R_iPWP Robot i performs in the working place, PP priority protocol, R_iFPPI Robot i follows protocol priority instruction, PB_i buffers; (2) Rules of Inference: (a) and (b) ($i = 1, 2$) if R_iSP then R_iWAWP, (c) if R_1WAWP and PB_1 and not $\diamond R_2WAWP$ then R_1PWP and PB_2, (d) if R_2WAWP and PB_2 and not $\diamond R_1WAWP$ then R_2PWP and PB_1, (e) and (f) ($i = 1, 2$) if R_iPWP then R_iSP, (g) if R_1WAWP and PB_1 and $\diamond R_2WAWP$ and PB_2 and PP then R_1FPPI and R_2FPPI, (h) if R_2WAWP and PB_2 and $\diamond R_1WAWP$ and PB_1 and PP then R_1FPPI and R_2FPPI, (i) if R_1WAWP and PB_1 and $\diamond R_2WAWP$ and PB_2 and not PP then R_1SP and R_2SP, (J) if R_2WAWP and PB_2 and $\diamond R_1WAWP$ and PB_1 and not PP then R_1SP and R_2SP, (k) if R_1WAWP and PB_1 and $\diamond R_2WAWP$ and not PB_2 then R_1PWP and PB_2 and R_2WAWP, (l) if R_2WAWP and PB_2 and $\diamond R_1WAWP$ and not PB_1 then R_2PWP and PB_1 and R_1WAWP, (m) if R_1FPPI and R_2FPPI then R_1PWP and R_2SP, (n) and (o) if R_iWAWP then $\diamond R_1WAWP$.

Remark 23. *The main idea consists of: the flexible manufacturing system is expressed by a modal logic formula, some query is expressed as an additional formula. The query is assumed to be a logical implication of the flexible manufacturing formula (see theorem 12). Then, transforming this logical implication relation into a set of clauses by using the techniques given in Section 3, its validity can be checked. It is important to point out that other type of behaviors can be incorporated in to the model by the modeler, making it as close to reality as needed.*

The formula that models the multirobotic system turns out to be:

$$[(R_1SP) \to (R_1WAWP)] \wedge [(R_2SP) \to (R_2WAWP)] \tag{1}$$

$\wedge[(R_1WAWP) \wedge (PB_1) \wedge \sim \diamond R_2WAWP \to (R_1PWP) \wedge (PB_2)] \wedge [(R_2WAWP) \wedge (PB_2) \wedge \sim \diamond R_1WAWP \to (R_2PWP) \wedge (PB_1)] \wedge [(R_1PWP) \to R_1SP] \wedge [(R_2PWP) \to R_2SP] \wedge [(R_1WAWP) \wedge (PB_1) \wedge \diamond R_2WAWP \wedge (PB_2) \wedge PP \to (R_1FPI) \wedge (R_2FPI)] \wedge [(R_2WAWP) \wedge (PB_2) \wedge \diamond R_1WAWP \wedge (PB_1) \wedge PP \to (R_1FPI) \wedge (R_2FPI)] \wedge$

$[(R_1WAWP) \wedge (PB_1) \wedge \diamond R_2WAWP \wedge (PB_2) \wedge \sim PP \rightarrow (R_1SP) \wedge (R_2SP)] \wedge$ $[(R_2WAWP) \wedge (PB_2) \wedge \diamond R_1WAWP \wedge (PB_1) \wedge \sim PP \rightarrow (R_1SP) \wedge (R_2SP)] \wedge$ $[(R_1WAWP) \wedge (PB_1) \wedge \diamond R_2WAWP \wedge (\sim PB_2) \rightarrow (R_1PWP) \wedge PB_2 \wedge (R_2WAWP)] \wedge$ $[(R_2WAWP) \wedge (PB_2) \wedge \diamond R_1WAWP \wedge (\sim PB_1) \rightarrow (R_2PWP) \wedge PB_1 \wedge (R_1WAWP)] \wedge$ $[(R_1FPI) \wedge (R_2FPI) \rightarrow (R_1PWP) \wedge (R_2SP] \wedge [(R_1WAWP \rightarrow \diamond R_1WAWP] \wedge$ $[(R_2WAWP \rightarrow \diamond R_2WAWP]$

We are interested in verifying, the following statement:

Claim:

$$R_1WAWP \wedge (PB_1) \wedge \diamond R_2WAWP \wedge (PB_2) \wedge PP \rightarrow R_2SP \wedge R_1WAWP.$$

The set of clauses is given by:

$$
\begin{aligned}
S = \{ & (\sim (R_1SP) \vee (R_1WAWP)), (\sim (R_2SP) \vee (R_2WAWP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \diamond R_2WAWP \vee (R_1PWP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \diamond R_2WAWP \vee PB_2), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \diamond R_1WAWP \vee (R_2PWP))), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \diamond R_1WAWP \vee PB_1), \\
& (\sim (R_1PWP) \vee (R_1SP)), \\
& (\sim (R_2PWP) \vee (R_2SP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee \sim PP \vee (R_1FPPI)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee \sim PP \vee (R_2FPPI)), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee \sim (PB_1) \vee \sim PP \vee (R_2FPPI)), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee \sim (PB_1) \vee \sim PP \vee (R_1FPPI)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee PP \vee (R_1SP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee PP \vee (R_2SP)), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee \sim (PB_1) \vee PP \vee (R_2SP)), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee \sim (PB_1) \vee PP \vee (R_1SP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee (PB_2) \vee (R_1PWP)), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee (PB_2) \vee PB_2), \\
& (\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee (PB_2) \vee R_2WAWP), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee (PB_1) \vee (R_2PWP)), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee (PB_1) \vee PB_1), \\
& (\sim (R_2WAWP) \vee \sim (PB_2) \vee \Box \sim R_1WAWP \vee (PB_1) \vee R_1WAWP), \\
& (\sim (R_1FPPI)) \vee \sim (R_2FPPI) \vee (R_1PWP)), \\
& (\sim (R_1FPPI)) \vee \sim (R_2FPPI) \vee (R_2SP)),
\end{aligned}
$$

$(\sim R_1WAWP \vee \diamond R_1WAWP)$,

$(\sim R_2WAWP \vee \diamond R_2WAWP)$,

$(R_1WAWP), (PB_1), (\diamond R_2WAWP), (PB_2), (PP), (\sim R_1WAWP \vee \sim (R_2SP))$.

Then a resolution refutation proof is as follows:

(a) $(\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee$
$\sim PP \vee (R_1FPPI))(R_1WAWP)(PB_1)(\diamond R_2WAWP)(PB_2)(PP) \rightarrow R_1FPPI$;

(b) $(\sim (R_1WAWP) \vee \sim (PB_1) \vee \Box \sim R_2WAWP \vee \sim (PB_2) \vee$
$\sim PP \vee (R_2FPPI))(R_1WAWP)(PB_1)(\diamond R_2WAWP)(PB_2)(PP) \rightarrow R_2FPPI$;

(c) $(\sim (R_1FPPI)) \vee \sim (R_2FPPI) \vee (R_2SP))(R_1FPPI)(R_2FPPI) \rightarrow R_2SP$;

(d) $(\sim R_1WAWP \vee \sim (R_2SP))(R_2SP) \rightarrow (\sim R_1WAWP)$;

(e) $(\sim R_1WAWP)(R_1WAWP) \rightarrow \perp$.

Therefore we have proved that the claim is true, this result is consistent with reality.

5. CONCLUSIONS

The main contribution of the paper consists in the study of the flexible manufacturing system by means of a formal reasoning deductive methodology based on modal logic theory. The modal logic approach introduces new operators that enable abstract relations like necessarily true and possibly true to be expressed directly. The results obtained are consistent with how the flexible manufacturing system performs.

REFERENCES

[1] Z. Retchkiman, *Stabilization/Regulation of a Multirobotic system modeled with Petri nets using vector Lyapunov and Comparison Principles*, Third International Conference on Dynamic Systems and Applications, Atlanta 2001.

[2] Z. Retchkiman, *A Mathematical Logic Validation Procedure for Petri Net Models*, International Journal of Pure and Applied Mathematics, Vol 115, No.3, 2017

[3] B.F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, 1980.

[4] P. Blackburn, J. van Benthem, and F. Wolter, *Handbook of Modal Logic*, Elsevier, 2007.

[5] M. Davis et al, *Computability, Complexity and Languages*, Academic Press, 1983.

[6] P. Enjalbert, L. Del Cerro, *Modal Resolution in Clausal Form*, Theoretical Computer Science, North Holland, 1989.