

# Assessment of Service Composition Plan using Colored Petri Nets

Dr.A.Bhuvaneshwari<sup>1</sup>, S.Uma<sup>2</sup>, S.Sakthitharan<sup>3</sup>, G.Srinivasan<sup>4</sup>

<sup>1</sup>Professor, CSE Department, Adhiparasakthi Engineering College, Melmaruvathur, India  
*bhuvan@adhiparasakthi.in*

<sup>2</sup>Assistant Professor, CSE Department, Adhiparasakthi Engineering College, Melmaruvathur, India  
*hiumas.sm@gmail.com*

<sup>3</sup>Assistant Professor, CSE Department, Adhiparasakthi Engineering College, Melmaruvathur, India  
*tharansakthi@adhiparasakthi.in*

<sup>4</sup>Assistant Professor, CSE Department, Adhiparasakthi Engineering College, Melmaruvathur, India  
*nandha987@gmail.com*

**Abstract:** Semantic web service composition constructs the OWL-S composite process automatically based on AI planning techniques. The need for verification of the service composition plan is essential as it is error prone. Colored Petri Nets had been proposed for verification due to its graphical, contains formal semantics and allows various analyses. The verification is done by constructing a CPN for an online book purchasing scenario using CPN tools and analyzing it through simulation. "CPN Tools" is the tool used to simulate the entire process flow and provide the statistics about liveness, fairness and boundedness properties. An online book purchase has been considered as a case study.

**Keywords:** Semantic Web Service Composition, Verification, Colored Petri Nets, Reachability, Semantic Reliability

## 1. Introduction

With the success of web services in internet and intranet, web service composition has turned out to be a widely accepted approach to construct the business processes. To attain the goal of a business process a combination of related web services is required. The automation of web service composition is a challenge mostly addressed by the planning domain of Artificial Intelligence [1] and further supported by the development of semantic web services [2]. AI planning automatically generates a composition plan to provide the order in which the related web services should be executed. Semantic web services augment semantics to the existing web service description and support automation of web service discovery and composition.

The generated composition plans are error prone as they possess characteristics like concurrency, correlation, and may contain deadlock. Hence the verification of the correctness of their behavior is essential and carried out with various techniques [3], [4]. Among those techniques

petri nets [5], [6] have gained popularity due to their graphical and formal representation.

A Petri net is a bipartite directed graph with two types of node namely places and transitions, a set of arcs that connect places with transitions or transitions with places and tokens located in places that represent the current state of the net. The behavior of web service composition is an ordered set of operations which can be mapped into petri nets. The mapping is done by representing web service operations with transitions and the state of service operations with places.

The composition plan is expected to be semantically reliable i.e. the type of the tokens within a place and also the tokens transferred between two transitions should be functionally compatible. Such verification is not possible using petri nets as it lacks in specifying the type of the token. Hence the Colored Petri Nets [7], a variation of Petri nets that combines the qualities of petri nets and a high level programming language can be employed to solve the problem. The qualities of petri nets provide the primitives for the description of the synchronization of concurrent processes, while that of programming languages provide the primitives for the definition of types for tokens.

Colored Petri Net (CP-nets or CPN) is a graphic oriented language with well-defined mathematical foundation for design, specification, simulation and verification of systems. CP-nets have two forms of representation namely (i) Algebraic Colored Petri nets and (ii) Graphical Colored Petri nets. Algebraic CP-nets represent net structure in an algebraic form and the dynamic behavior as algebraic equations. The structure of the net is represented in a matrix. The state of a CP-net is represented using a state equation. The solution of this state equation helps in analyzing the properties of the net. Hence the CP-net in algebraic form may be verified using reachability analysis based on the state equation.

Graphical CP-nets represent the net structure as a graph and have four major elements: places, transitions, arcs and tokens. The places describe the states of the system, the transitions describe the actions and arc expressions describe how the state of the CP-net changes when the transitions occur. Each place contains a set of markers called tokens. Each token is associated with a color to indicate its identity. As in petri nets the tokens are moved from one place to another on firing a

transition. The major difference is in the color of the involved tokens which specify the functional dependency between the places. Often the color associated with the tokens represents a type of the data-value. The analysis of a CP-net in graphical form may be done by simulation with the aid of available tools. CPN Tools[8] is preferred for its flexible user interface, improved interaction techniques and graphical feedback that informs about the status and simulations.

This paper deals with the usage of Colored Petri Nets to verify the semantic web service composition plan represented by OWL-S. The verification is carried out by employing reachability analysis of an algebraic CP-net and simulation using CPN Tools. As QoS of a generated plan had been investigated in a different dimension in the previous works [9], [10] only the semantic reliability of the plan is considered in this paper.

The next section provides the literature related to the verification of the composition plan, Section 3 provides a detailed description of Colored Petri Nets and CPN Tools, Section 4 explains the relationship between OWL-S and CPN, Section 5 presents the motivating scenario that reveals the need for Colored Petri Nets, Section 6 provides the construction of CPN for the case study, Section 7 provides the mathematical analysis of CP-nets and Section 8 provides the simulation and state-space analysis of CP-nets using CPN Tools and Section 9 provides the conclusion of the paper.

## 2. Related work

This section discusses the various verification techniques and tools that could be applied for verifying web service composition. SPIN is a tool used for verification of software systems. The input to this tool is a program written using Promela, a meta-language for processes and their properties are specified using linear temporal logic. It checks whether the program satisfies the stated properties. In order to make use of SPIN to verify the OWL-S processes they should be converted into Promela.

Process algebra is a small concurrent language that abstracts from many details and focuses on particular features. Process algebras are modeled by means of labeled transition systems. The process algebra name Finite State Process represents a finite labeled transition system. A tool, LTSA (Labeled Transition System Analyzer) for Finite State Process traverses the state-space defined by a model and reports about the first safety or progress error that is encountered but not more than one of each type. FSP used in LTSA can be efficiently used only for design time validation.

Petri Nets were used to model and verify the business processes. Many researchers had worked on variations of Petri Nets namely workflow nets, business process nets and colored petri nets. Among them Colored Petri Nets additionally verifies the type of the data which helps in checking the semantic reliability of the composition plan. As most of the available techniques focuses on design time validation, Colored Petri Net that focuses on data with their type is preferred in this paper for verification of composition plan. The application areas of CPN include the design and verification of communication protocols, distributed systems, embedded systems, automated production systems, VLSI chips and automated business processes.

In the previous works [11, 12] of the authors the verification of the generated composition plan was carried out using LTSA. As it was experienced that LTSA can identify only the first progress error, an alternate verification tool was required. The survey lead to the identification of CP-nets for the verification of composition plan as specified in [13] which performs the mathematical verification of the CP-nets using verification algorithms for reachability, boundedness, semantic consistency which includes QoS consistency.

## 3. Colored Petri Net

Colored Petri Nets provide a framework for the construction and verification of business processes constructed as composition plan. CPN models are executable and hence can be simulated to verify the behavior of composition plan. Simulation provides a visual feedback of every step and realization of application domains. The state space method of CP-nets validates and verifies the functional correctness of the composition plan. A state space is constructed which verifies the behavioral properties of the composition plan such as absence of deadlocks, reachability of a given state and guaranteed delivery of a given service. The basic constructs of a CP-net are as follows:

**3.1 Places:** Places are the containers of data transferred between the transitions in a CP-net.

**3.2 Markings:** Markings represent the state of a CP-net which consists of a number of tokens positioned on the individual places. Each token possess a value (color) based on the type of the place on which it inhabit. The token values and the data types are referred as token colors and color sets respectively. The tokens present on a particular place are called as the marking of that place. The marking of a place is a multi-set of token values which means that a place may have several tokens with the same token value. As an example, possible marking of the place book database is the following:

$1^{\wedge}("T1", "A1", 100, false) ++ \text{ }^{\wedge}("T1", "A4", 200, true)$

This marking contains 1 token with value ("T1", "A1", 100, false) and 2 tokens with the value ("T1", "A4", 200, true). Conventionally, multi-sets are written as a sum (++) using the symbol prime ( $\wedge$ ) to denote the number of appearances of an element.

### 3.3 Types:

Each place has an associated type (color set) determining the type of data that place may contain. The type is similar to a data type in a programming language. They may be complex as a record that contains one field as a text string, another as a real and third as an integer. As an example, a possible type definition is the following:

```
colset Author = STRING;
colset Price = INT;
colset Avail = BOOL;
colset APA = product Author*Price*Avail;
```

The definition APA is a complex type that consists of a string, integer and boolean data type.

### 3.4 Transitions:

The actions of a CP-net are represented by means of transitions, drawn as rectangles with their names written inside.

The transition “findauthor” models the action that finds the author of the book for a given title.

### 3.5 Arcs and arc expressions:

The transitions and places are connected by arcs. When a transition occurs the tokens are removed from the places connected to incoming arcs and added to the places connected to outgoing arcs. The number of tokens added and removed by a transition is determined by the arc expressions written next to the arcs. A double arc represents the bidirectional flow of data with same arc expression.

The **formal definition of the CPN** can be given as: A CPN is a tuple  $CPN = (\Sigma, P, T, A, N, C, G, E, I)$  where:

- (i)  $\Sigma$  is a finite set of non-empty types also called color sets.
- (ii)  $P$  is a finite set of places.
- (iii)  $T$  is a finite set of transitions.
- (iv)  $A$  is a finite set of **arcs** such that:  
 $P \cap T = P \cap A = T \cap A = \emptyset$ .
- (v)  $N$  is a node function.

It is defined from  $A$  into  $P \times T \cup T \times P$ .

- (vi)  $C$  is a color function.

It is defined from  $P$  into  $\Sigma$ .

- (vii)  $G$  is a guard function. It is defined from  $T$  into expressions such that:

$$\forall t \in T: [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma]$$

- (viii)  $E$  is an arc expression function. It is defined from  $A$  into expressions such that:

$$\forall a \in A: [Type(E(a)) = C(P)_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma] \text{ where } P \text{ is the place of } N(a).$$

- (ix)  $I$  is an initialization function. It is defined from  $P$  into closed expressions such that:

$$\forall p \in P: [Type(I(p)) = C(p)_{MS}].$$

The net structure of CPN can be denoted by a matrix with  $n$  rows and  $m$  columns where  $n$  and  $m$  represent the number of transitions and number of places respectively.

$A = [a_{ij}]$  an  $n \times m$  matrix of integers where

$$a_{ij} = a_{ij}^+ - a_{ij}^-$$

$$a_{ij}^+ = \begin{cases} 1, & \text{if } (t_i, p_j) \in F \\ 0, & \text{otherwise} \end{cases}$$

$$a_{ij}^- = \begin{cases} 1, & \text{if } (p_j, t_i) \in F \\ 0, & \text{otherwise} \end{cases}$$

$$A^+ = [a_{ij}^+]_{n \times m} \text{ and } A^- = [a_{ij}^-]_{n \times m}$$

$A$  is the incidence matrix which defines the whole structure of CPN,  $A^+$  is the output matrix that represents the arcs from transition  $i$  to place  $j$  and  $A^-$  is the input matrix that represents the arcs from place  $j$  to transition  $i$ . When a transition triggers and there is a state change in the system, the final state can be obtained by evaluating the state equation of the system. The state equation of the CPN is given by  $M_i = M_{i-1} + A^T * \sigma$ , where  $M_i$  is an intermediate state and  $M_{i-1}$  is its previous state,  $\sigma$  is the firing vector and  $A^T$  is the transpose of the incidence matrix. The properties of the CP-net are analyzed based on the incidence matrix and the solution obtained for the state equation.

CPN Tools analyzes CP-nets in two ways such as (a) simulation and (b) state space analysis. **Simulation feedback** is updated during the syntax check and simulations. Green circles indicate the number of tokens that are present at the time of simulation on each place, and current markings appear in green text boxes next to the places. Green halos around transitions are used to indicate enabled transitions. Pages

containing enabled transitions are underlined with green in the index, and their page tabs are also underlined with green. The full or partial **state spaces** are generated and analyzed for CP-nets by the CPN Tools. The state space requires a syntactical constraint which specifies that all places and transitions should have unique names. The state space report contain information about: statistics about the generation of the state space, boundedness properties, home properties and liveness properties.

The statistical report contains information about the size in terms of the number of nodes and arcs in the state space. The boundedness properties contain information about the maximal and minimal number of tokens that may be located on the individual places in reachable markings. **Home properties** tell about the markings or sets of markings to which it is always possible to return. **Liveness properties** tell that a set of binding elements remains active.

## 4. OWL-S/CPN relationship

A Semantic Web Service is a semantically described web service using service ontology, which facilitates machine readability of capabilities and integrates service with application domain. The semantic descriptions of web services are essential to permit automation in discovery, composition and execution among various users and application domains. The major efforts in specifying the semantic description of web services are Web Ontology Language for Services(OWL-S) which provides the machine readable form of properties and capabilities of a web service, Web Service Modeling Ontology (WSMO) which includes definitions for goals, mediators and web services and Semantic Annotations for WSDL (SA-WSDL) which defines the semantic annotation by referring semantic models. As OWL-S is the most popularly used language for semantic web services, this paper considers and provides its relationship with CP-nets.

OWL-S ontology shown in Figure 1 provides three essential types of knowledge about a web service: (i) Service Profile providing the information required to discover a service, (ii) Service Model providing information about how to use the web service and (iii) Service Grounding specifying the details about how to access the web service. The upper ontology of OWL-S is shown in the following Figure. The class Service refers a web service and acts as a domain for the properties *presents*, *describedBy*, and *supports*. The respective ranges of those properties are the classes ServiceProfile, ServiceModel, and ServiceGrounding. Each instance of Service will *presents* a ServiceProfile description, be *describedBy* a ServiceModel description, and *supports* a ServiceGrounding description. The details of profiles, models, and groundings may vary from one instance of Service to another.

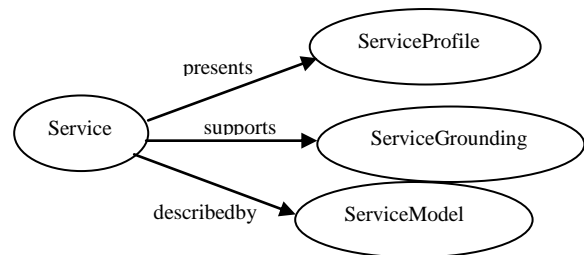


Figure 1: OWL-S ontology

The basic constructs of the CP-nets is provided in detail in Section 3. The relationship between OWL-S and CPN can be tabulated as in Table 1. The service parameters of OWL-S includes the input and output data of a service which are represented as places in a CPN.

The service operations are the logic implemented to execute the service represented as transitions in a CPN. The types of input and output data are represented by the color set of the CPN. The service precondition is the condition that must be satisfied to invoke the service and the service effect is the state of the environment after the service execution. They are represented with the input and output places of a CPN respectively. The direction of message transmission and the values that are transmitted are represented with the arcs and arc expressions of a CPN.

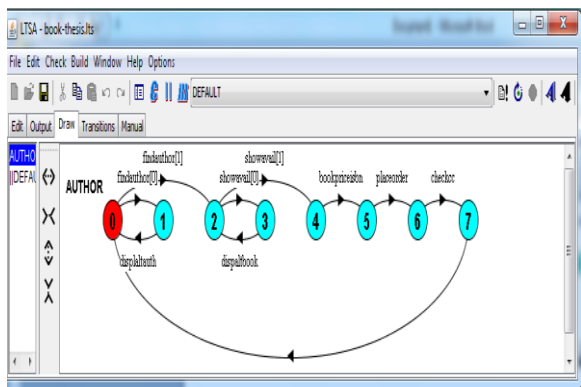
**Table 1:** Relationship between OWL-S and CPN

S. No.	OWL-S	CPN
1	Service Parameters	Places
2	Service Operations	Transitions
3	Type of Service Parameters	Color Set
4	Service Precondition	Input Places
5	Service Effects	Output Places
6	Direction of message transmission	Arcs and arc expression

### 5. Motivating Scenario

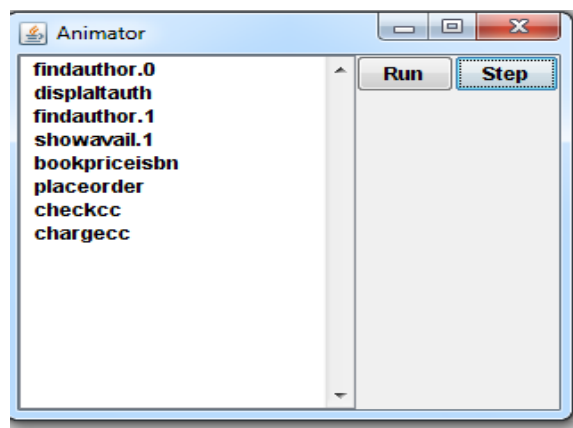
A scenario in which the user needs to purchase a book through online shopping requires the composition of web services. When a user provides title of the book as input, the services to find the details of the author, price and availability of the book, find an alternate book if the requested book is not available, check the validity of the credit card and process the credit card should be arranged. The arrangement should check the availability of the requested book and if available purchase the same. Otherwise find an alternate book and purchase it. This can be constructed as a web service composition plan by using AI planning techniques.

The LTSA tool provides a Finite State Process (FSP) translation mechanism to convert generated workflows into a complete FSP specification. It mechanically assists the user in building and translating the OWL-S implementation to FSP. The FSP for the plan generated for the process specified in the case study is shown in Figure 2.



**Figure 2:** Plan generated by the Planner

The FSM generated by the LTSA tool can be verified and validated by the Animator plug-in which helps in visualizing the execution order of the generated plan. Figure 3 shows the execution pattern of the plan generated if the book is available.



**Figure 3:** Execution of the Plan

However LTSA can verify only the first safety or progress error that is encountered but not more than one of each type. When both safety and progress violations are there at the same time due to different root causes the decision should be made on which error has to be addressed. Giving priority to the progress issue rather than the safety issue had the effect of hiding of the safety issue that would come out later.

Hence an alternate approach was required for the verification process and Colored Petri Nets were employed. The verification of reachability and safety properties of the service composition plan was done by a mathematical approach. The boundedness, liveness, fairness and home properties were verified using CPN Tools.

### 5. Construction of CPN using CPN Tools

A sample OWL-S process is shown in Figure 4 for the findauthor service which takes the title of the book as input and returns the author of the book.

```
<process:AtomicProcess rdf:ID="FINDAUTHOR
  _PROCESS">
  <service:describes rdf:resource="# FINDAUTHOR
    _SERVICE"/>
  <process:hasInput rdf:resource="# _BOOKTITLE"/>
  <process:hasOutput rdf:resource="# _AUTHOR "/>
  <process:hasOutput rdf:resource="# _ EDITION"/>
  <process:Input
    rdf:ID="_BOOKTITLE"><process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyU
    RI">http://127.0.0.1/ontology/books.owl#Title</process:par
    ameterType>
  </process:Input>
  <process:Output
    rdf:ID="_AUTHOR"><process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyU
    RI">http://127.0.0.1/ontology/books.owl#Author</process:
    parameterType>
  </process:Output>
  <process:hasEffect>
  <expr:SWRL-Condition rdf:ID=" AuthorFound">
  <swrl:classPredicate
    rdf:resource="http://127.0.0.1/ontology/ontosem.owl#be_av
    ailable"/>
  <swrl:argument1
    rdf:resource="# _AUTHOR"/></expr:SWRL-Condition>
```

## 7. Simulation and State-Space Analysis

The verification of the service composition plan using the CPN tool is explained below.

Step 1: The author and price are found from the database. If the book is available then the credit card processing is proceeded as shown in Figure 9

Figure 4: OWL-S Snippet - “findauthor” service

The OWL-S processes involved in the case study are converted into CPN as shown in Figure 5. The color set of a place is typically written below the place and is declared using the Standard ML [9] programming languages. For example, place Title has the color set Title which is a string. The declarations of the color sets, functions and variables used in CPN are listed in Figure 6.

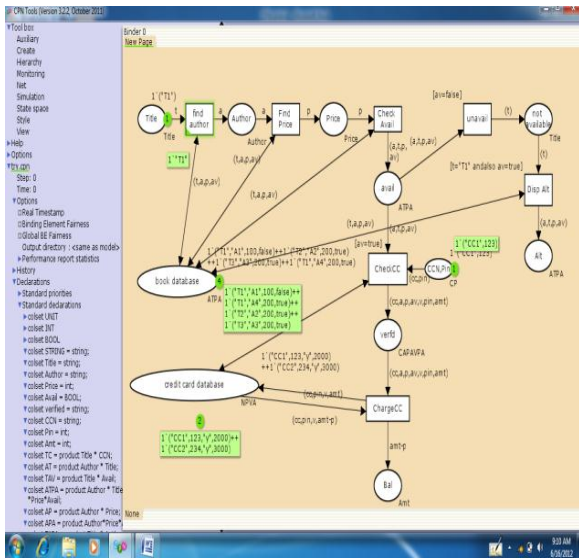


Figure 5: CPN for Book Purchase OWL-S process

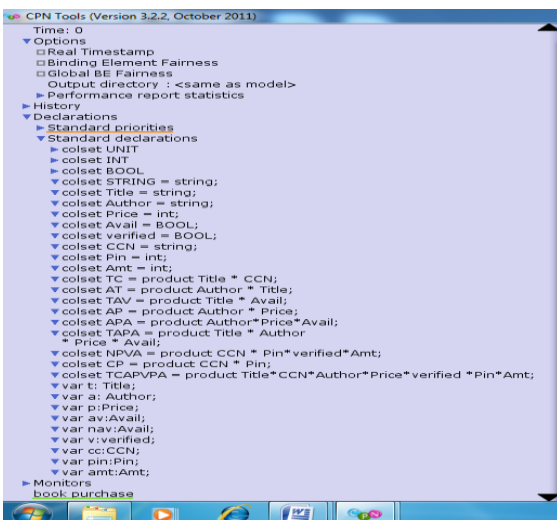


Figure 6: Color set - online book purchase scenario

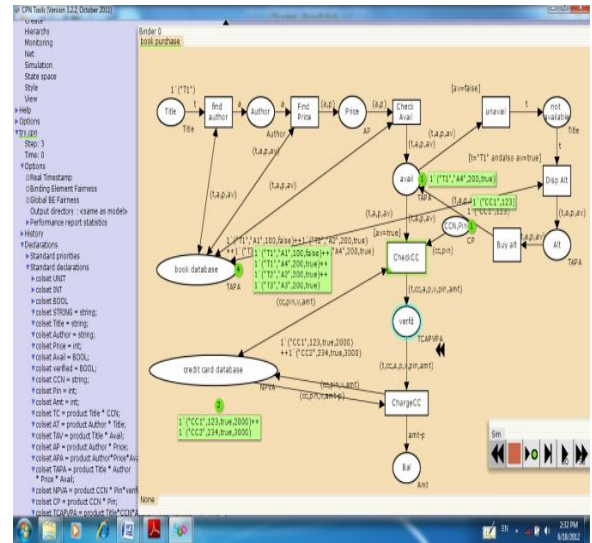
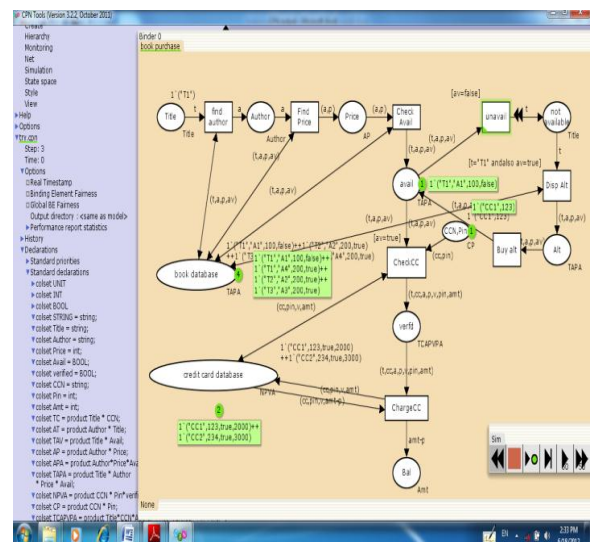


Figure 9: Book available and credit card checked

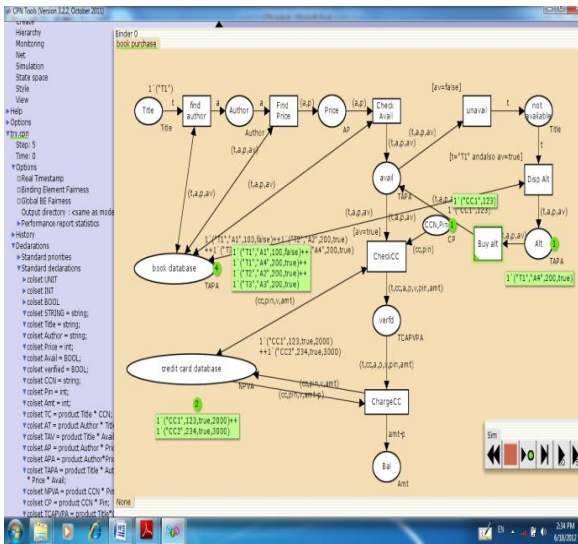
Step 2: Else if the book is not available the alternate book is selected which are shown in the following Figures Figure 10 and Figure 11

Step 3: Finally the book is purchased and the credit card is used for purchase and the amount is debited from the credit card which is reflected in Figure 12.

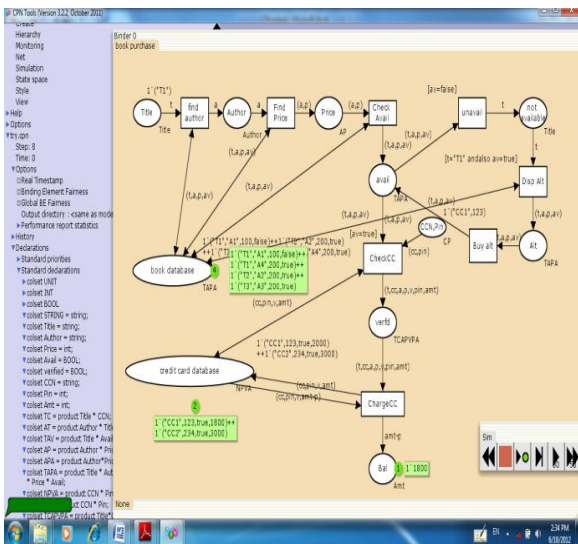
Simulation works like testing a program and can be used to prove or verify properties of the trivial systems. Hence it can be alternated by state space analysis. A state space constructs a directed graph with a node for each reachable marking and an arc for each occurring binding element. The state space analysis of the colored petri net constructed for the case study consists of three parts. The first part provides the statistics of the full state space which contains 11 nodes and 11 arcs in the CPN calculated in 0 seconds as shown in Table 2.



**Figure 10: Unavailability of Book**



**Figure 11: Alternate book Selection**



**Figure 12: Credit Card Processing**

The second part shows the integer bounds of the places in the CPN as given in Table 3. The upper and lower integer bounds are the maximal and minimal number of tokens that can be located on the individual places in the reachable markings. The place Title has either one or not token which indicates that the title of the book may be available or not which also reflects in the avail and notavail places. This indicates that the system behaves as expected.

**Table 2: Statistics of Full State Space**

Statistics	
State Space	SCC graph
Nodes: 1	Nodes: 11
Arcs : 11	Arcs : 11
Secs : 0	Secs : 0
Status : Full	

**Table 3: Boundedness – Upper and Lower Bounds**

Boundedness		
Place	Upper Bound	Lower Bound

Title	1	0
avail	1	0
book_database	4	4
credit_card_database	2	2
not_available	1	0
verfd	1	0

The third part shows the Home and Liveness Properties. A home marking is a marking which is reachable from all reachable markings, i.e., a marking which can always be reached independently of what has previously happened. Here there is a single home marking  $M_{11}$ . A dead marking is a marking with no enabled transitions. Here the case study has a single dead marking, and that is similar to the home marking. Dead transitions are similar to dead code in a programming language and it means that each transition is enabled in at least one reachable marking. A live transition is a transition which can always, no matter what happens, become enabled again. From the liveness properties shown in Table 4 it can be observed that there are no dead and live transitions. When there are dead markings as in our case study there cannot be any live transitions.

**Table 4: Home and Liveness**

*Home and Liveness Properties*

Home Markings	[11]
Dead Markings	[11]
Dead Transition Instances	None
Live Transition Instances	None

**9. Conclusion**

The verification of the correctness of generated plans is carried out by using colored petri nets. The verification is done by converting an OWL-S process into CPN using CPN Tools and analyzing simulation and state space analysis. The simulation and state space analysis done using CPN Tools verified the boundedness, liveness and provided a statistical report about the constructed CPN. Thus the verification of the service composition plan had been successfully carried out using colored petri nets.

**10. References**

1. Joachim Peer, "Towards Automatic Web Service Composition using AI Planning Techniques", International Journal of AI Planning Techniques, pp.1-18, 2003
2. Katia Sycara, P. Massimo, Paolucci. Anupriya, Ankolekar. Naveen, Srinivasan. "Automated discovery, interaction and composition of Semantic Web services", International Journal of Web Semantics Vol.1, No.1, pp. 27-46, 2003.
3. G.J. Holzmann. "The Spin Model Checker: Primer and Reference Manual", Addison-Wesley, Boston, MA, USA, 2004.

4. J. Magee and J. Kramer, "Concurrency: State Models and Java Programs", Wiley, New York, NY, USA, second edition, 2006
5. Juan C. Vidal, Manuel Lama, Alberto Bugarín, "Toward the use of Petri nets for the formalization of OWL-S Choreographies", *International Journal of Knowledge information System*, pp. 1-37, 2011.
6. Wang J., *Timed Petri Nets: Theory and Application*, Kluwer Academic Publishers, 1998.
7. R. Farahbod, U. Glasser, and M. Vajihollahi, "Specification and validation of the business process execution language for web services", *Proceedings of the 11th International Workshop on Abstract State Machines*,
8. <http://www.daimi.au.dk/CPNtools/>
9. Bhuvaneshwari, A. Dr.Karpagam, G.R. "An Optimized Service Selection for Service Composition", *Proceedings of the International Conference: ICAC 2011 On Advances in Computing*, PSG College of Technology, Coimbatore, India, Dec 9-11, 2011, pp. 153-162
10. Bhuvaneshwari, A. Dr. Karpagam, G.R. "QoS Considerations for a Semantic Web Service Composition", *European Journal of Scientific Research*, Vol.65, No.3, 2011, pp. 403-415
11. Bhuvaneshwari, A. and Karpagam, G.R. "Applying fluent calculus for automated and dynamic semantic web service composition", *Int. Conf. on Intelligent Semantic Web-Services and Applications*, June, Amman, Jordan. 2010, pp.124-131.
12. Karpagam, G.R. and Bhuvaneshwari, A. "AI planning-based semantic web service composition", *Int. J. Innovative Computing and Applications*, Vol. 3, No. 3, pp.126-135, 2011.
13. Yue Ni, Yushun Fan, "Model Transformation and formal verification for Semantic Web Services Composition", *Journal of Advances in Engineering Software*. 41: 879 – 885, 2010.



**Mr. Sakthitharan. S.** has gained his Bachelors and Masters degree in Computer Science and working as Assistant Professor at Adhiparasakthi Engineering College with experience of 1.5 years. He has presented papers in International conferences. He has been awarded as 'Young Investigator Award' by IRNet.



**Mr. G. Srinivasan** is Assistant Professor with 1.5 years experience in Department of Computer Science and Engineering in Adhiparasakthi Engineering College. He has presented papers in international conferences in the area of Data Mining and Warehousing. His current research areas includes networking and cloud computing.

## Author Profile



**Dr. A. Bhuvaneshwari** is Professor with 15 years of experience in Department of Computer science and Engineering in Adhiparasakthi Engineering College. She is an ISTE member and interested in the field of web services, Model Driven Architecture and cloud service discovery. She has published papers in IEEE and ACM digital

libraries and National/International Journals and reviewer of International Journals. She has authored a book on Computer Programming.



**Ms. S. Uma** is Assitant Professor with 7.5 years of experience in Department of Computer Science and Engineering in Adhiparasakthi Engineering College. She has published papers in national journals in the areas including networks and information security.