Design and Verification of Loosely Coupled Inter-Organizational Workflows with Multi-Level Security

Boleslaw Mikolajczak
University of Massachusetts, Computer and Information Science Department, Dartmouth, MA 02747, USA
Polish-Japanese Institute of Information Technology, Warsaw, Poland
bmikolajczak@umassd.edu

Nirmal Gami
University of Massachusetts, Computer and Information Science Department, Dartmouth, MA 02747, USA
g_ngami@umassd.edu

Abstract—**Inter-Organizational Workflows (IOWF) become important as they provide solution for data sharing, heterogeneity in resources and work coordination at global level. However, a secured computing infrastructure like Multilevel Security (MLS) is needed to support today's vast businesses.**

In this paper Message Sequence Charts (MSC) are used to specify the positive and negative interactions between cooperating organizations. Petri nets are used to model the workflows in each organization. IOWF is obtained by combining Message Sequence Charts (MSC) and workflows of local organizations.

We present an algorithm to incorporate MLS features into IOWFs using Bell-LaPadula security model. In this model security labels of subject and object are verified before the subject can access the object. The algorithm reduces the workflows of participating organizations using the reduction rules while preserving the communication patterns between organizations. We also present an algorithm to identify implicit places in the IOWF with MLS features. Our method and algorithms are illustrated by a running example.

Index Terms—**multi-level security features, loosely coupled inter-organizational workflows, correctness of inter-organizational workflows, Bell-LaPadula security model, Petri nets, implicit places.**

I. Introduction

The Internet, which is the primary medium for conducting e-commerce, is by design an open non-secure medium. Inter-Organizational Workflows allow data sharing and work coordination at the global level as the globalization of business becomes a common practice. However, the prolific use of Inter-Organizational Workflows for critical and strategic applications makes security an essential and integral part. Another major problem with Inter-Organizational Workflows is that they often use heterogeneous and distributed hardware and software systems to execute a given workflow. This gives rise to decentralized security policies and mechanisms that need to be managed.

Inter-organizational workflows merged with multilevel security provide the necessary security. However sophisticated techniques are required to review, analyze, and test this approach for correct behavior.

Workflows are case-based, i.e., every piece of work is executed for a specific case. Cases are handled by executing tasks in a specific order. Workflow process definition specifies which tasks need to be executed and in what order. Each task has pre and post conditions: the preconditions should hold before the task is executed, and the post conditions should hold after execution of the task. Most work items are executed by a resource. A resource is either a machine or a person. Resources are allowed to deal with specific work items.

Workflow has three dimensions: the case dimension, the process dimension and the resource dimension. The case dimension signifies the fact that all cases are handled individually. Cases do not directly influence each other. Clearly they influence each other indirectly via the sharing of resources and data. In the process dimension, the workflow process, i.e., the tasks and the routing along these tasks, is specified. In the resource dimension, the resources are grouped into roles and organizational units. A workflow management system (WfMS) is a software system that supports the modeling, execution, and administration of business processes. Before a workflow can be executed it has to be described in a way that WfMS is able to understand. This description is called workflow specification.

In this paper we address the following practical problem of contemporary electronic commerce: how to design correct and secure electronic commerce enterprise that involves many organizations cooperating infrequently through synchronous or asynchronous message passing. Both correctness and security are critical features of such system.

The above problem is difficult because complex communication patterns between organizations may lead to inter-organizational workflow systems that are not correct in terms of soundness and consistency. Adding requirements of MLS to such systems constitutes additional level of complexity.

IOWFs were previously analyzed and classified by several authors [15, 16, 17]. These papers provide a methodology of IOWFs modeling and analysis using Petri nets. Issues of information systems security and, in particular, of MLS were also presented in [12, 13, 14].

In this paper we model workflows and IOWFs using Petri nets. We also apply methodologies of Petri nets and software engineering of distributed computing systems. Security features of MLS are expressed and integrated in terms of Petri nets on top of IOWFs. Such merging of IOWFs and MLS features and related analysis' techniques constitute main elements of novelty.

The paper presents a Petri net-based novel modeling technique that merges MLS features with IOWFs. It also presents an algorithm that reduces number of places in Petri net representing IOWF with MLS using the implicit places' concept.

II. MULTILEVEL SECURITY

It is a concept involving mandatory access control (MAC), i.e. the system enforces security policy regardless of the actions of system users or administrators. MLS systems [7] strive to enforce the security restrictions with incredibly high reliability so as to not leak any data at all.

Access control decisions are based on clearance level for users/subjects and classification level for information/objects in the system. The term multilevel is used because both people and information are classified into different levels of trust and sensitivity. These are referred to as security labels or security levels.

In addition to hierarchical clearance levels (e.g. Secret, Top Secret, Unclassified) information is marked by a classification level depending upon its sensitivity level. This marking (classification level) indicates another restriction placed on the distribution of a particular classified data item. A security label may include classification level identifiers in addition to a hierarchical clearance level.

A system with classification levels generally acquires a large number of distinct security levels: one for every legal combination of a hierarchical clearance level with zero or more classification levels. Fig. 1 shows a system that contains Top Secret (T), Secret (S) and Unclassified (U) hierarchical clearance levels. Information has been classified with compartments A and B. The arrows in the lattice show security levels that can be read by which other security levels [1, 2, 3].

Two security levels can be compared based upon their clearance levels and classification levels. Given two security levels, first their clearance levels are compared. If the clearance levels are different then hierarchical ordering of clearance levels is used to determine which security level has higher precedence over the other. This is followed by comparison of their classification levels to determine the reading and writing rights. For example in Fig. 1, if we have to compare two security labels T{A,B} and S{A} then we conclude that T{A,B} has higher precedence than S{A} based on hierarchical ordering of classification level. Then we compare classification levels of two given security labels. We conclude that T{A,B} can read data labeled S{A} since it contains the A compartment. If we have to compare security labels T{} and S{A} then we first conclude that T{} has higher precedence than S{A} based on hierarchical ordering of classification level. Then we compare classification levels of two given security labels. We conclude that T{} cannot read data labeled S{A} since it does not contain the A compartment.

If clearance levels are same then the classification levels determine the higher precedence as well as the reading and writing rights.

There is also a supremum security level comprising of highest clearance level and all the classification levels. In Fig. 1, if a user has Top Secret clearance with access to both compartments A and B, i.e. a supremum security label of T{A, B}, then he/she has the permission to read any data on the system. There is also an infimum security level formed by lowest clearance level and zero classification levels. In Fig. 1 U{} is the infimum security level. The interrelationships between these levels generally based upon reading and writing rules form a directed acyclic graph, with nodes representing the security levels. We call this graph to be a security lattice.

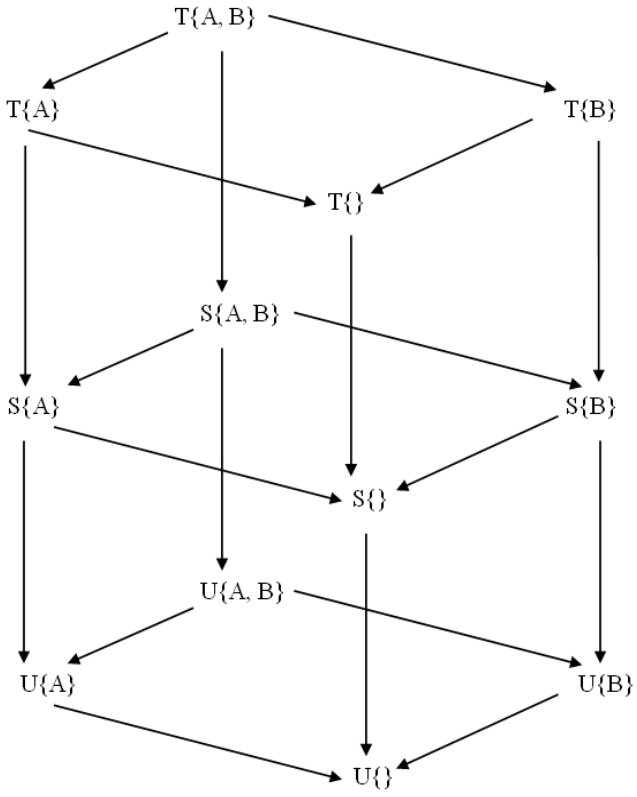

Figure 1: Security lattice for Unclassified (U), Secret (S), Top Secret (T) users with compartments A, B.

Today's globally spread businesses use heterogeneous and distributed hardware and software systems to execute a given workflow. This gives rise to decentralized security policies and mechanisms that need to be managed. The prolific use of inter-organizational workflows for critical and strategic applications gives rise to a major concern regarding security and a need for a more reliable security mechanism.

The Bell-LaPadula Model [7, 8, 9] also called the multi-level model, was proposed by Bell and LaPadula in 1970s. It is a formal state transition model of computer security policy that describes a set of access control rules. In this model, the entities in a computer system are divided into abstract sets of subjects and objects. A "subject" is somebody (user) who wants access to an "object" (information, data file, system). The concept of a secure state is defined, and it is proven that each state transition preserves security by moving from secure state to secure state, thereby inductively proving that the system is secure.

A system state is defined to be "secure" if the only permitted access modes of subjects to objects are in accordance with a specific security policy. In order to determine whether or not a specific access mode is allowed, the clearance of a subject is compared to the classification of the object, and a determination is made as to whether the subject is authorized for the specific access mode. The Bell-LaPadula model supports mandatory access control by determining the access rights from the security levels associated with subjects and objects.

The concept of a secure state is defined by two properties: the simple security (ss) property and the *-property.

(1) *ss-property* allows all low-level information to be available at a higher level. It restricts high-level information to be available at a lower level. A subject is allowed to read an object only if the former's security label is identical or higher than latter's security label (no read up).

(2) **-property* ensures there is no write down. A subject with a higher security label should not write an object of lower security label.

These two restrictions ensure there is no flow of information from higher security label to lower security label subjects/objects. However these two properties are not sufficient to ensure that security is not compromised since it could be possible that leakage of information can occur through indirect means via covert channels.

In some situations, special devices are required to make lower security label information available at higher security label and vice-versa. We can use downgraders and starlight link [10, 11] for this purpose. Downgraders are devices that allow lower security label information to flow to higher security label but not vice versa. Starlight link is a secured mechanism that allows higher security label to write to lower security label.

Within an organization there are various subjects with hierarchical security levels ranging from high to low level. Also most organizations have various classification levels for information, depending upon its sensitivity. Bell-LaPadula security model requires identification of subjects and objects in the system and assigning security labels to them. This can be easily done because of the way organizations are composed. Thus we use Bell-LaPadula model to incorporate MLS in IOWF.

III. MODELING INTER-ORGANIZATIONAL WORKFLOWS

In order to explain the application of algorithms presented in coming sections of the paper, we consider an inter-organizational workflow between two organizations namely, Car Company and Tire Company. We assume that the local workflows are executed correctly. First a brief explanation of various communication scenarios that can occur between participating organizations of this inter-organizational workflow is presented. Message sequence charts are used for this purpose. A set of scenarios is a behavioral specification that can be positive or negative.

***Positive Scenarios**:*

These are desirable behaviors. System should be able to execute every positive scenario at least once, leaving the system in a specified safe state. We have following positive scenarios for our example:

***Successful ordering (Fig. 2)**:*

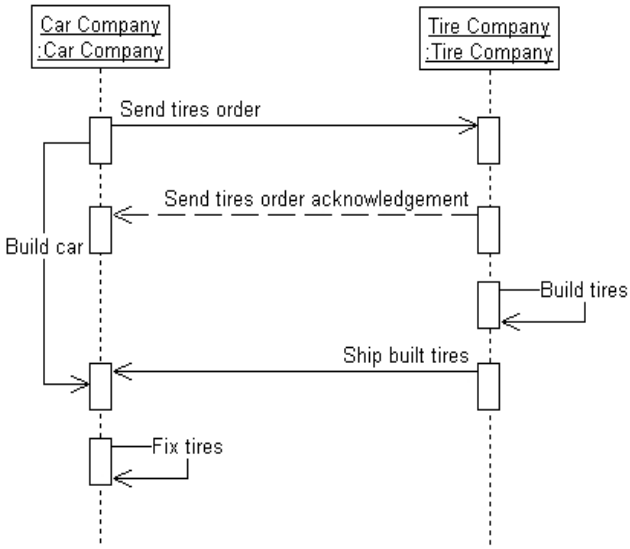

Figure 2: MSC for successful ordering

Car Company initiates the workflow. It sends 'tires order' to the Tire Company and starts building the car. Upon receiving the order, Tire Company sends acknowledgement. If Car Company finishes building the car then it waits for the tires. Once the tires are built, tires are shipped to the Car Company. Once the tires are received, tires are fixed onto the car.

***Resources unavailable (Fig. 3)**:*

Car Company sends the 'tires order'. Tire Company sends 'Resources unavailable' notice. Car Company can

then send the order to another company or send the order again at a later time.

Timeout (Fig. 4):

Car Company sends the 'tires order'. If the Car Company does not get an acknowledgement and 'timeout' occurs, then it resends the tires order to the Tire Company. Car Company sends the 'tires order'. Tire Company can then send 'suggest modification' message depending upon its available resources. Car Company can choose to send either 'accept or decline modification' message.

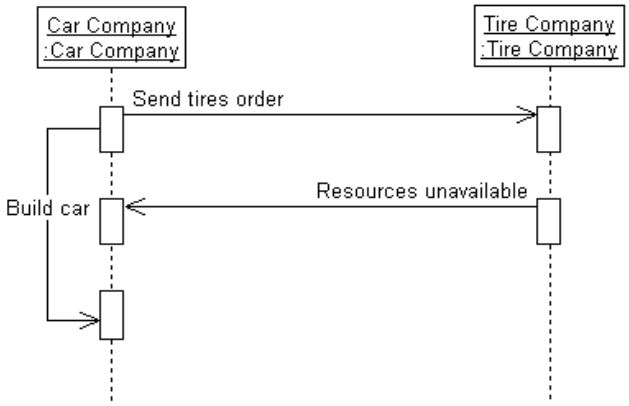

Figure 3: MSC for resources unavailable

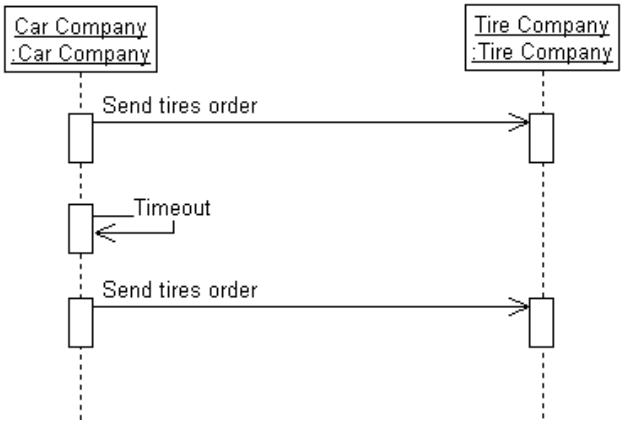

Figure 4: MSC for timeout

Tire Company will then send 'acknowledgement' and then start building the tires. The Car Company builds the car and waits for the tires. Tire Company ships the tires to the Car Company. Upon receiving, the tires are fixed onto the car.

Negative Scenarios

These indicate undesirable behaviors that the designer is aware of. System should not exhibit any of these scenarios. Following are the negative scenarios.

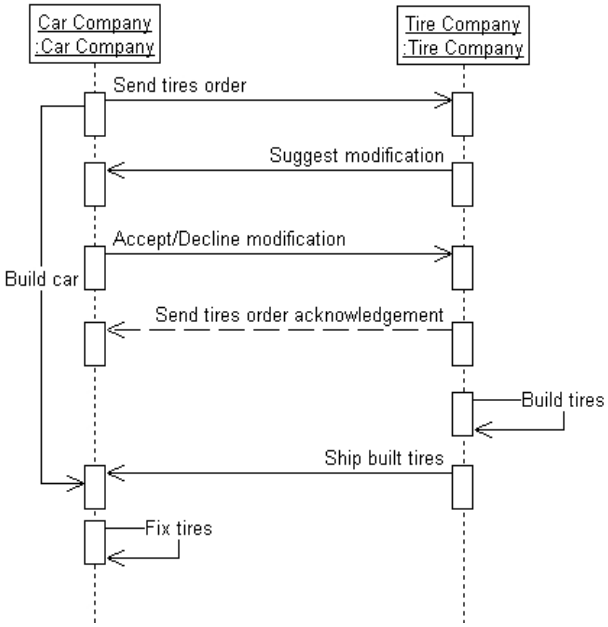

Figure 5: MSC for suggest modification with positive/negative response

No acknowledgement (Fig. 6):

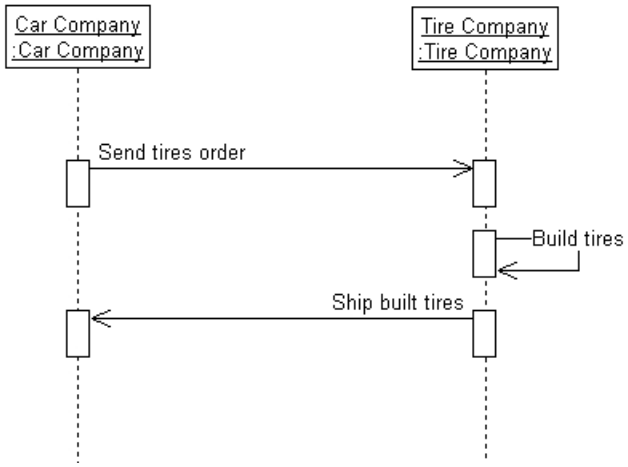

Figure 6: MSC for no acknowledgement

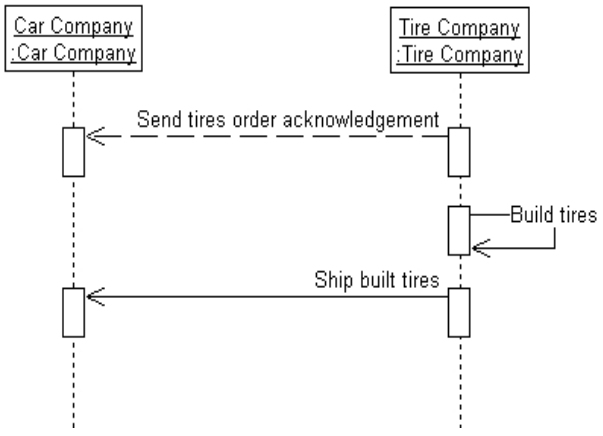

Figure 7: MSC for acknowledgement without order

Car Company sends the 'tires order'. Tire Company builds the tires and ships the same, without sending

'acknowledgement'. On the other hand, because the Car Company did not receive an acknowledgement, 'timeout' scenario could be executed.

Acknowledgement without order (Fig. 7):

Tire Company sends 'acknowledgement' and ships built tires to Car Company that did not order them.

Cancel order (Fig. 8 and Fig. 9):

Car Company sends 'tires order' to Tire Company. Then at a later stage, Car Company cancels the order by sending 'cancel order' message to Tire Company.

Car Company sends 'tires order' to Tire Company. At a later stage, Car Company sends 'make update' message to Tire Company in order to make changes to their initial tire order. The order update information will be treated as a new order.

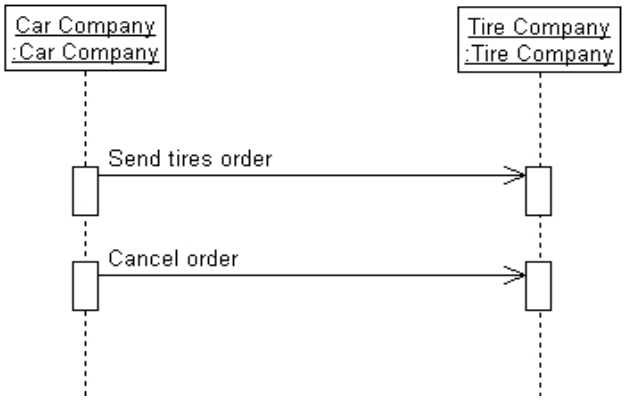

Figure 8: MSC for canceled order

Make update:

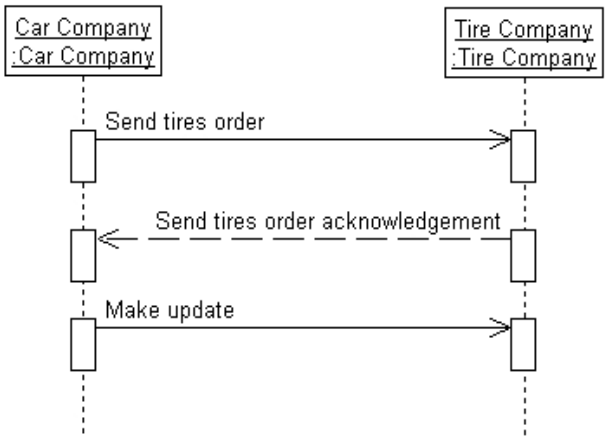

Figure 9: MSC for making update

Fig. 10 shows the Car Company workflow. First, a detailed order with technical specifications is prepared by the owner. This order contains details such as performance requirements, weather conditions, environment in which car will run and more. From this tire information is sent to the Tire manager while other parts of the car are assembled by the Car builder. Tire manager prepares a 'tires order' containing tire specification. This 'tires order' is sent to the Tire Company. Tire manager waits for the 'acknowledgement'. If acknowledgement is received he waits for the tires else if there is no response for specified time then 'timeout' occurs and the order is resent. Tire manager can also receive 'resources unavailable' notice from the Tire Company. In this case he resends the tire order at a later time. Tire manager can also receive order modification suggestions from the Tire Company in the form of 'suggest modification' message. Then Tire manager can send the 'accept or decline modification' message. When tires are built, Tire receiver receives them. He checks the received tire specification and informs the Tire Assembler. Tire Assembler then puts the tires on the car and informs the owner that the car is built.

Fig. 11 shows the Tire Company workflow. This workflow is executed when resources are available and a 'tires order' is received by the owner. The owner can send an order received 'acknowledgement' or send a 'resources unavailable' notice or 'suggest modifications' message. After acknowledgement is sent order processing begins. The owner sends it to Threads maker. Thread maker starts making the thread selection. Choices of rain threads and normal threads are available. At the same time, Thread maker sends the rims requirement to Rim maker. Two choices of available rims are Rim A and Rim B. Tire maker then combines the threads and the rims to build the tires. Built tires are shipped to the Car Company by Tire shipper.

Next we use the positive and negative scenario message sequence charts to combine the workflows of participating organizations and to model the inter-organizational workflow [13, 14, 15] such that all positive scenarios can occur, while negative scenarios cannot occur. This IOWF is represented in Fig. 12. Tire order, 'resources unavailable', 'suggested modifications', 'modification response', 'tire order acknowledgement' and 'built tires' form the information exchanged between organizations.

IV. MERGING MULTILEVEL SECURITY INTO INTERORGANIZATIONAL WORKFLOWS

Now we present an algorithm to merge multilevel security into an inter-organizational workflow.

Algorithm 1 (Merging MLS into IOWF)
Input: IOWF and MLS lattice
Output: IOWF with MLS features

Step 1. Identify a set of subjects $A = \{A_1, A_2 ... A_p\}$ where $p \geq 1$ for any one of the participating workflow.
Step 2. Determine a set of hierarchical clearance levels $\{X_1, X_2 ... X_m\}$ for subjects, where $1 \leq m \leq p$ and X_j has higher precedence than X_i for $j > i$.
Step 3. Identify a set of objects $B = \{B_1, B_2 ... B_q\}$ where $q \geq 0$ in the same participating workflow.
Step 4. Determine a set of classification levels $\{Y_1, Y_2 ... Y_n\}$ for objects depending upon its sensitivity, where $0 \leq n \leq q$.
Step 5. Combine clearance levels and classification levels to obtain security lattice with security labels $S_k = X_i\{Y_1, Y_2 ... Y_j\}$ where $i \leq m$, $j \leq n$, $k \leq m2^n$, as nodes.

Step 6. Assign security labels to subjects and objects taking into account Bell-LaPadula security model and the working of the participating workflow, to form a security lattice of applicable security labels. If A is a set of all subjects and S is the set of all possible security labels, then there exists a many-to-one function f_1: A → S, i.e. each element in set A has a corresponding element in set S. If B is a set of all objects and S is the set of all possible security labels, then there exists a many-to-one function f_2: B → S, i.e. each element in set B has a corresponding element in set S.

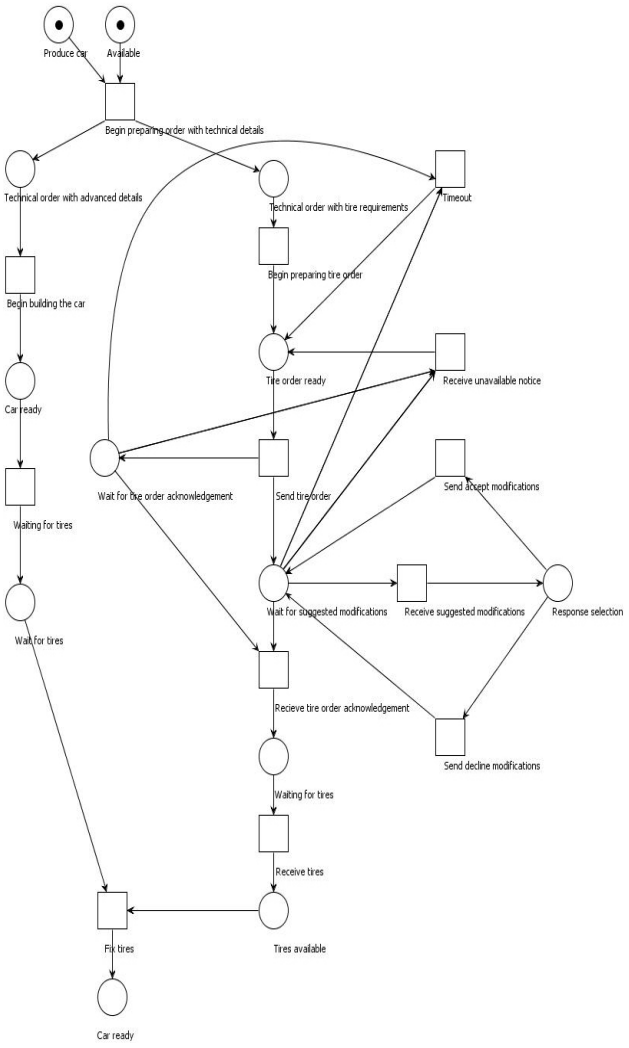

Figure 10: Car Company workflow

Step 7. Repeat steps 1 to 6 for all participating organizations.

Step 8. Combine security lattices of participating organization taking into account which security label can read which other security label, to obtain security lattice for the complete IOWF. If S_1 and S_2 are two security labels such that S_1 can read S_2 then introduce an arrow from S_1 to S_2 in the security lattice indicating reading right.9. Compare security label of subject with security label of object it is trying to access. Grant access only if the subject is cleared to access that object, otherwise deny access.

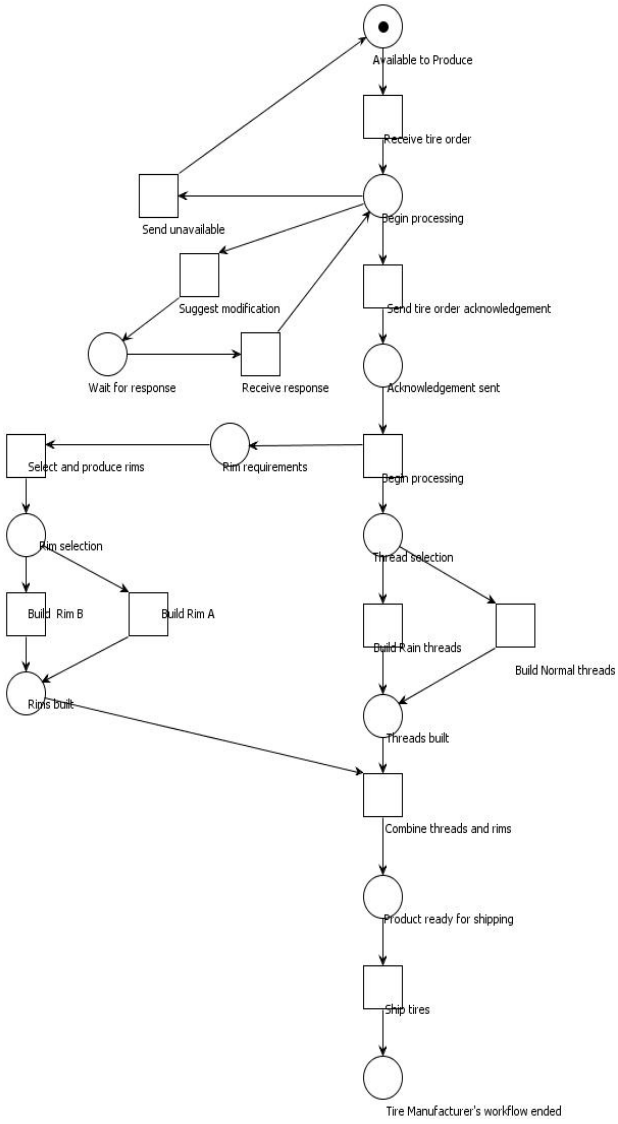

Figure 11: Tire Company workflow

.

The application of Algorithm 1 is shown with the help of the Car Company and the Tire Company example.

Consider the Car Company. We first identify the subjects in the system. We have identified five subjects in the Car Company namely the owner, tire manager, car builder, tire receiver and tire assembler. We assign clearance level to each person. Then we identify the sensitive information that we need to protect. In Car Company, sensitive information comprises of car order, tire order containing tire specification, unavailable notice, suggested modifications, modification response, timeout specification, advanced car details, received tire information, and car ready notice. In all these sensitive information there are two types that need to be kept separate, namely tire information ‘t’ and car details ‘c’.

We combine clearance level and classification level to obtain security labels and assign them to subjects and objects. This assignment takes into account the company working and the Bell-LaPadula security model. For example, Tire manger would be concerned with tires and

not car details and because he has Secret clearance, his security label is S{t}. This would give him access to tire order, unavailable notice, suggested modifications, modification response, and timeout specification and received tire information. Security labels obtained by following the above steps are indicated in Tab. 1 and Fig. 13.

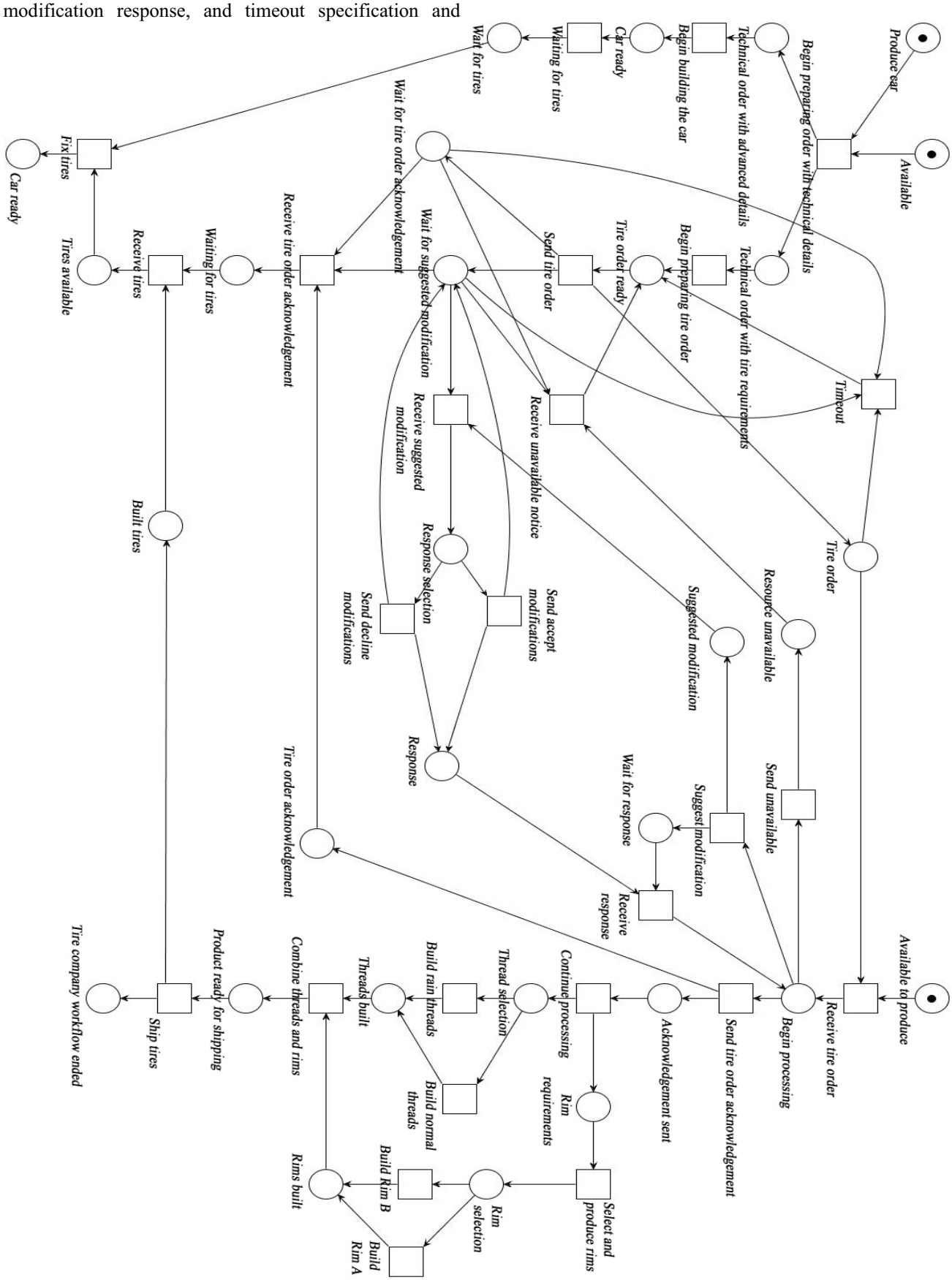

Figure 12: IOWF comprising Car and Tire Company

Table 1: Car Company security labels

Car Company					
Subjects	**Clearance level**	**Security label**		**Objects**	**Security label**
Owner	Top Secret	T{c, t}		Car order	T{c, t}
Tire Manager	Secret	S{t}		Tire order, Tire order acknowledgement, Unavailable notice, Suggested modifications, Modification response, Timeout specification	S{t}
Car Builder	Secret	S{c}		Car details	S{c}
Tire Assembler	Secret	S{c, t}		Received tire information,	U{t}
Tire Receiver	Unclassified	U{t}		Ready notice	U{c, t}

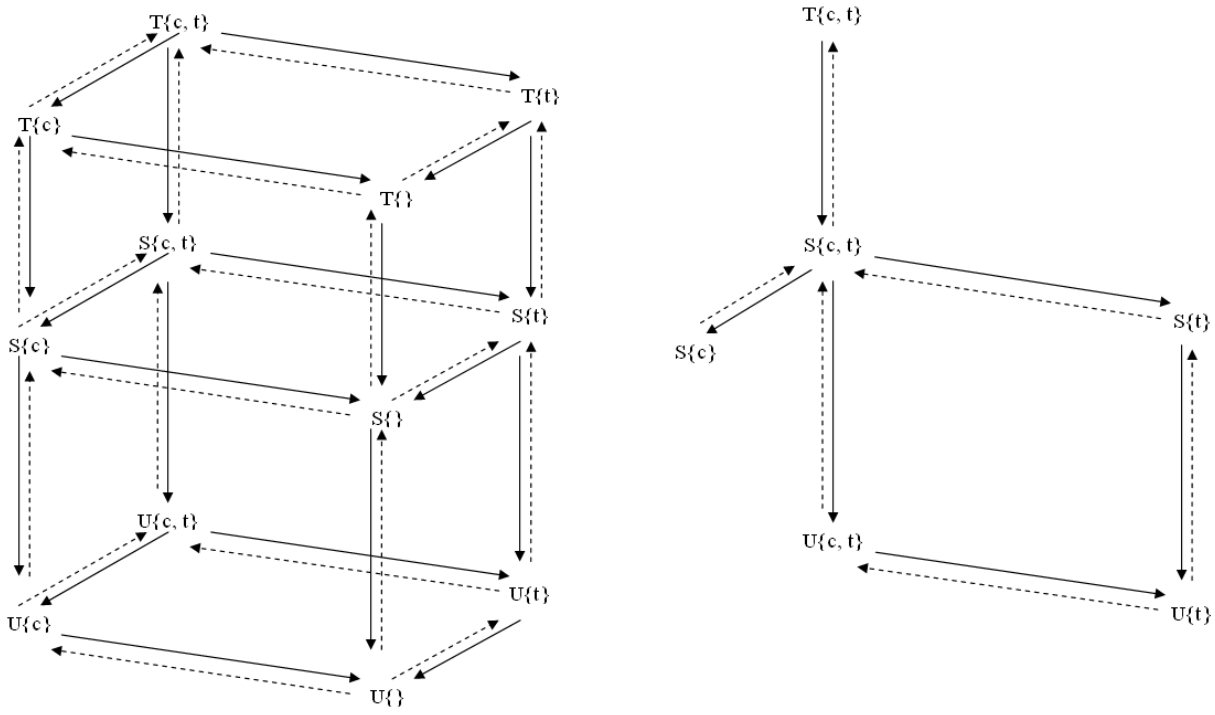

Figure 13: Car Company security lattice,
Left figure: With all permissible security labels,
Right figure: With applicable security labels

Table 2: Tire Company security labels

Tire Company					
Subjects	**Clearance level**	**Security label**		**Objects**	**Security label**
Owner	Top Secret	T{t, th, r}		Tire Order, Tire order acknowledgement, Unavailable notice, Suggested modifications, Modification response	S{t}
Tire maker	Secret	S{t, th, r}		Threads information	U{th}
Threads maker	Unclassified	U{t, th, r}			
Rim maker	Unclassified	U{r}		Rim information	U{r}
Tire shipper	Unclassified	U{t}		Tire information	U{t}

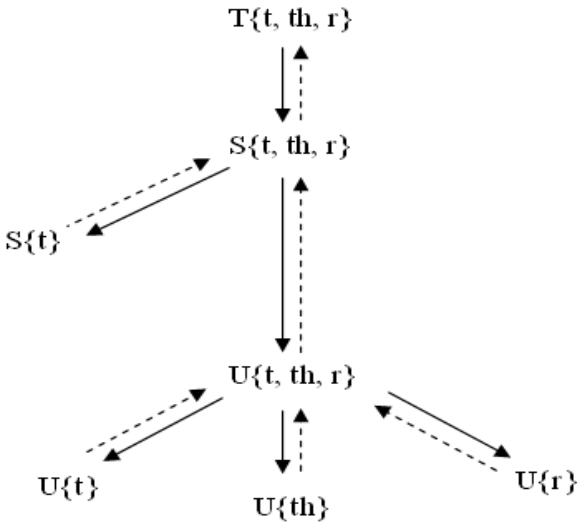

Figure 14: Tire Company security lattice with applicable security labels

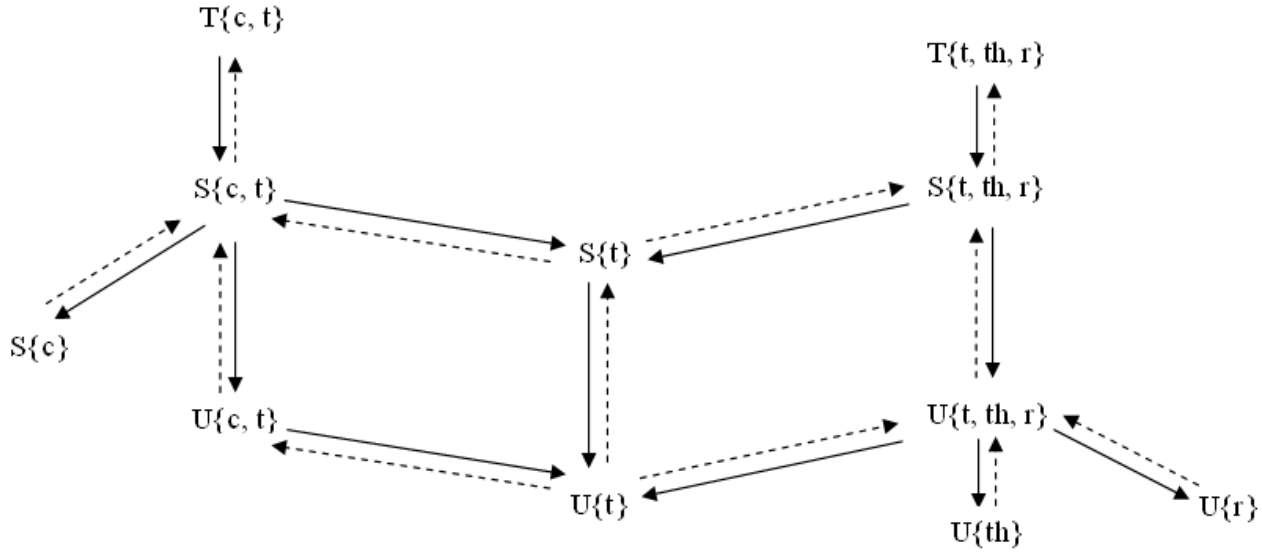

Figure 15: IOWF security lattice with applicable security labels

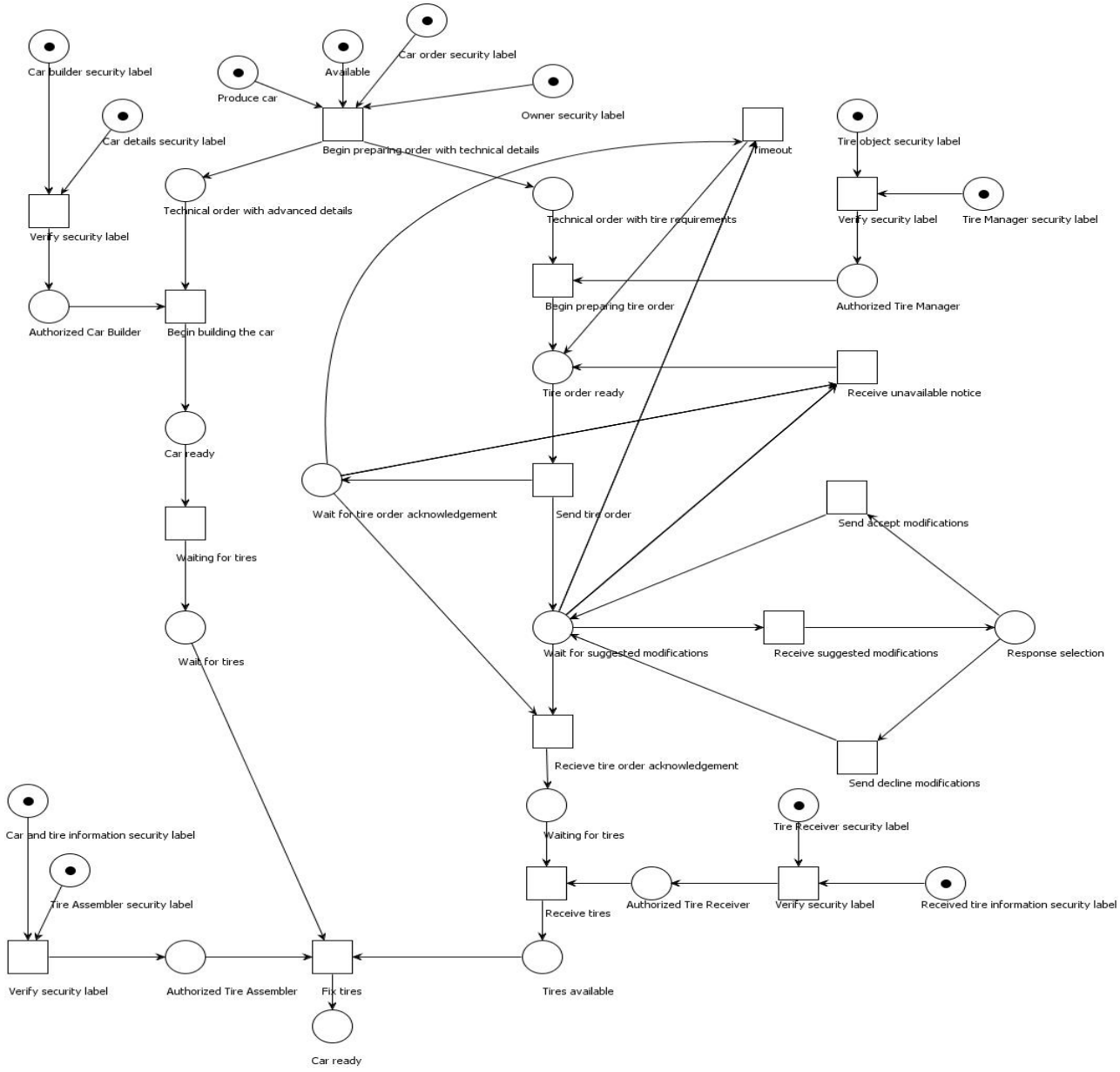

Figure 16: Car Company workflow with MLS features.

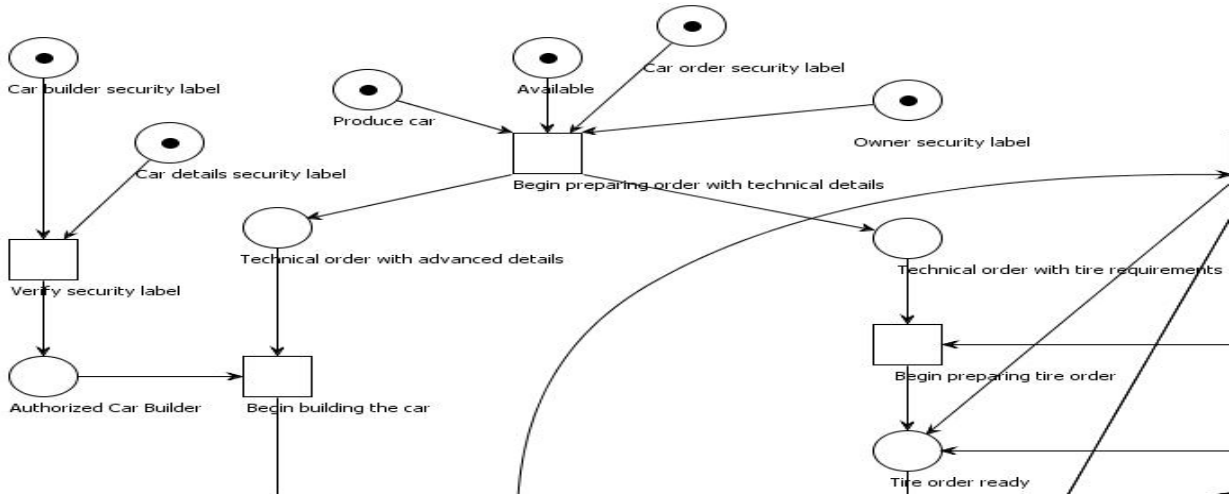

Figure 17: Car Builder (subject) accessing Technical order (object).

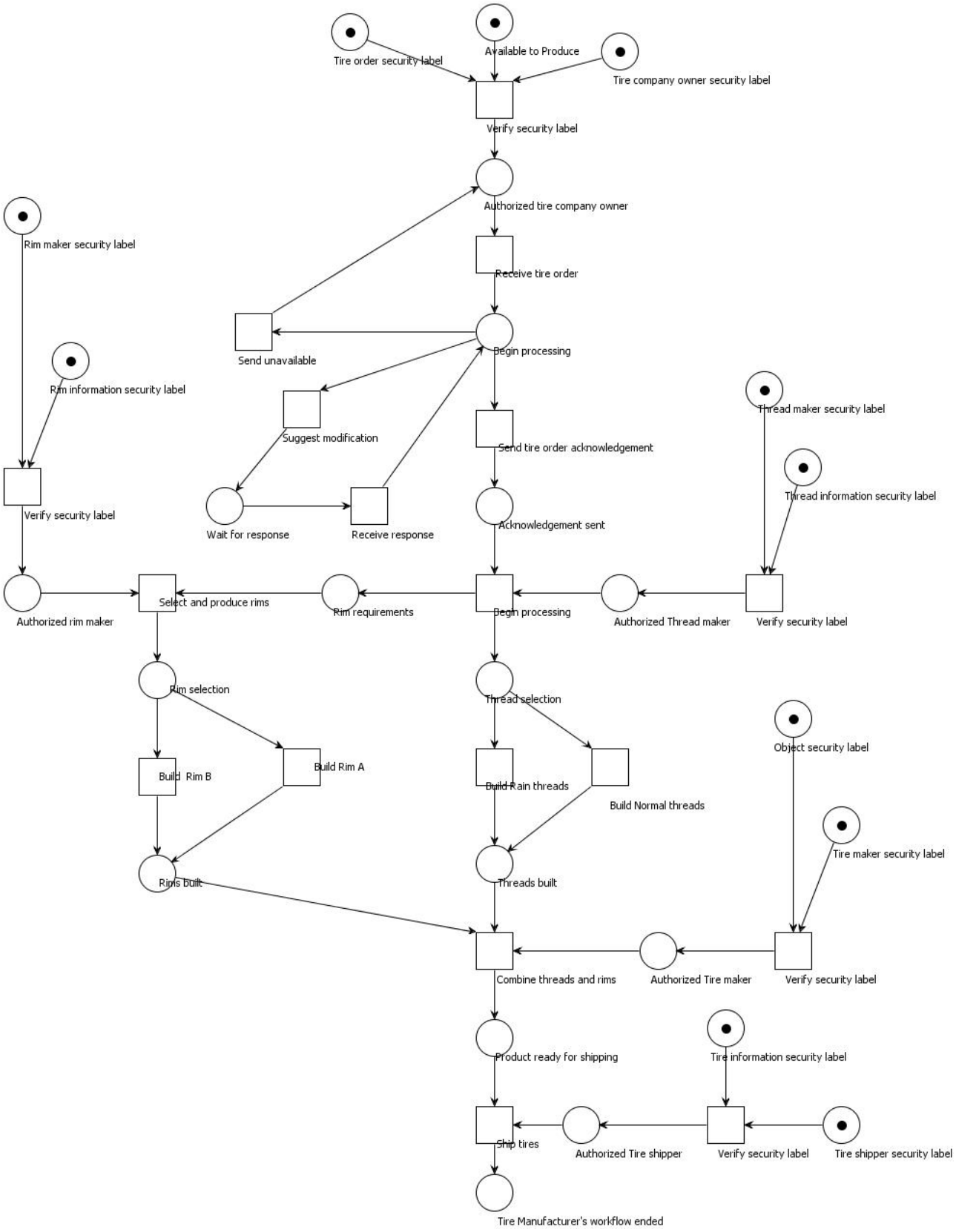

Figure 18: Tire Company workflow with MLS features.

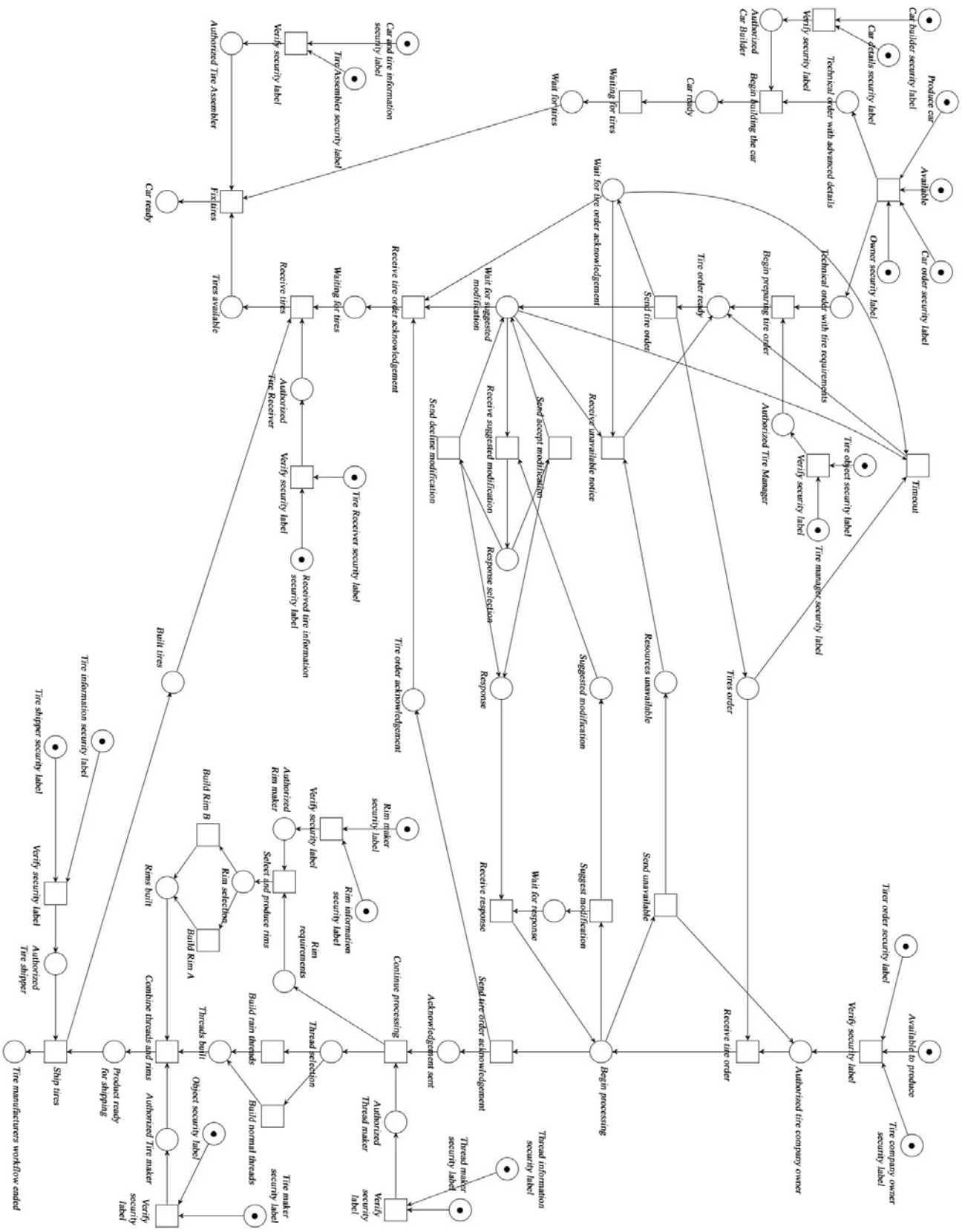

Figure 19: Car and Tire Company IOWF with MLS features.

Fig. 13 represents lattice of security labels. Figure on left shows all permissible security labels obtained by combining clearance levels and classification levels. Figure on right contains only those security labels that are applicable to our example. Also the straight arrows and dotted arrows indicate reading and writing rules i.e. S{t} can read U{t} but not S{c, t}. Similarly S{t} can be written by U{t} and not vice versa.

Our next step is to apply Algorithm 1 to the Tire Company. We identify subjects and objects in the company and assign security labels. Note that there are different classifications, namely threads 'th' and rims 'r' in Tire Company. The subjects, objects and their security labels are indicated in the following Table 2.

Now we incorporate these MLS features added in individual organizations into complete IOWF. For this we first form the security lattice for the complete IOWF as shown in Fig. 15. The Tire receiver in the Car Company and the Tire shipper in the Tire Company will have access to the same built tire information (having security label U{t}) and hence they should have same security labels, i.e. U{t}. Tire manager in the Car Company having security label S{t} sends the tire order (having security label S{t}) to the owner of Tire Company having security label T{t,th,r}. This allows the owner of Tire Company to read the tires order. The owner of Tire Company should not be able to read Car details having security label S{c} and hence the owner's security label does not have the 'c' component. Similarly, taking into account what role the subjects play in the whole inter-organizational workflow, the lattice of Fig. 15 is obtained by combining security lattices shown in Fig. 13 and 14.

We now have a representation of IOWF in PNML as shown in Fig. 12 and a security lattice indicating reading and writing rules among applicable security labels shown in Fig. 15. To incorporate MLS features in IOWF, we need to combine both of these. As the final step of Algorithm 1, we need to verify the security label of subject with the security label of object it is trying to access. Access is granted only if the subject is cleared to access that object, otherwise access is denied. In Petri nets, a subject accesses an object during transition's firing. So we verify subject and object security labels before allowing the transition to be enabled. We implement this concept incrementally. We first apply it to individual organizations. This is shown in Fig. 16 and 18.

Consider Car Builder trying to access Technical order with advanced details. Fig. 17 obtained from Fig. 16, indicates how this object access takes pace. Before Car Builder can access Technical order with advanced details, security label of Car Builder, S{c} and security label of car details, S{c} are compared. If Car Builder has clearance to access car details then there will be a token in the place 'Authorized Car Builder'. Because of this transition 'Begin building the car' can fire when there is a token in 'Technical order with advanced details'. In other words, when advanced car detail is available and we have an authorized Car Builder then he can start building the car. Similar verifications are done at other object accesses.

Now we include these additional places and transitions that correspond to subject and object security label verification as shown in Fig. 16 and 18 into the inter-organizational workflow shown in Fig. 12. Thus we incorporate the MLS features in the IOWF. The final result is shown in Fig. 19.

V. CORRECTNESS OF INTERORGANIZATIONAL WORKFLOWS

In IOWFs each business partner has a private workflow process that is connected to the workflow processes of some of the other partners. It involves communication between the workflows of all participating organizations. Error in design of IOWF are thus difficult to detect and can result in some serious consequences. Therefore, there is need to detect the correctness of the IOWF. There are two concepts to verify the correctness of IOWF namely soundness and consistency. A workflow is sound if and only if, for any case, the process terminates properly, i.e., termination is guaranteed, there are no dangling tasks and there is no deadlock in the workflow. Consistency deals with verifying whether the implementation of IOWF meets the original specification.

In order to check the consistency [20] of IOWF, instead of checking all possible firing sequence the concept of implicit places is used to avoid state explosion. A place in a net system is a constraint on the firing of its output transitions. If the removal of a place does not change the behavior of the original net system, that place represents a redundancy in the system and can be removed. A place whose removal preserves the behavior of the system is called an implicit place, also called a redundant place [6, 13]. An implicit or redundant place always contains sufficient tokens to allow for the firing of transitions connected to it.

Behavior of a net system implies sequences of fireable transitions and marking of places in the net system. The behavior of the net system can be represented by the reachability graph.

Implicit places allow for the efficient verification of consistency. The generalized concept of implicit place set can be described as follows:

Let (PN, M) be a marked Petri net with PN = (P, T, F) and $P_1 \subseteq P$. P_1 is an implicit place set if and only if for every reachable state M' and any transition $t \in T$: if each place in $(\bullet t \setminus P_1)$ contains a token in state M', then each place in $(\bullet t \cap P_1)$ contains a token in M'. Place p P is an implicit place if and only if {p} is an implicit place set.

In Fig. 20 p_5 is implicit as it does not influence the behavior of the workflow. A token is placed in p_5 when transition t_1 fires. Then transition t_2 fires followed by t_3. Even if p_5 was removed, it would have not affected the flow of transition firings, as can be seen from the reachability graphs. The set $\{p_5\}$ is implicit place set

for the workflow in Fig. 20. Removal of implicit place is significant especially in larger workflows.

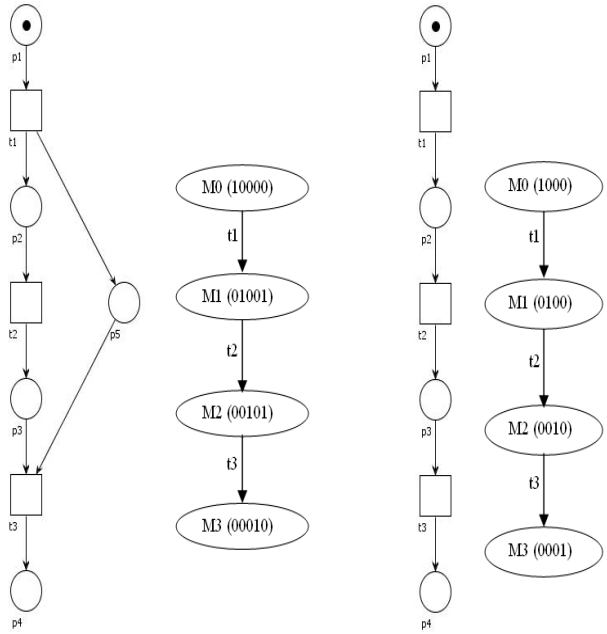

Figure 20: Left two figures: Workflow with implicit places and its reachability graph,
Right two figures: Workflow with removed implicit places and its reachability graph

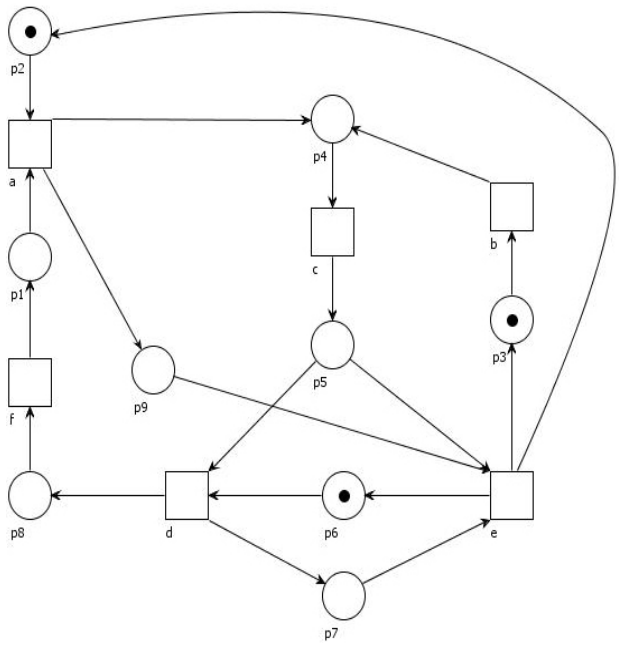

Figure 21: Example workflow

If p is not the only input place of its output transition, then p may be implicit. If a transition has only one input place then that input place cannot be implicit, because in order for the transition to fire, the input place must be present and eventually be marked. In other words, we need to analyze only those input places for which the connected transitions have more than one input place.

Hence we first need to identify transitions with more than one input place and form a set T_P of such transitions. Next we form a set of input places to any transition in T_P and denote it as P_P. The concept of implicit place can be defined as follows...

Let (PN, M) be a marked Petri net with PN= (P, T, F) with $P_P \subseteq P$ and $T_P \subseteq T$ such that T_P is a set of transitions with more the one input place and P_P is set of places corresponding to $\bullet T_P$. If there is a path from $\bullet p_i$ excluding p_i to any one of the other places corresponding to identical rows in Pre $[P_P, T_P]$ then p_i is implicit. Below we present an algorithm to identify implicit places in a workflow.

Algorithm 2: Identification of Implicit Places

Input: Petri Net representation of a Workflow
Output: Equivalent Petri Net representation of a Workflow without implicit places

Step 1. For a given workflow identify a set $T_p = \{t_1, t_2 ..., t_n\}$ where $n \geq 0$ of transitions with more the one input place.
Step 2. Identify a set $P_p = \{p_1, p_2 ... p_m\}$ where $m \geq n$ of input places corresponding to transitions in the above set T_p.
Step 3. If there is a path from $\bullet p_i$ excluding p_i to any one of the other places corresponding to identical rows in Pre$[P_p, T_p]$ then p_i is implicit.
Step 4. Repeat steps 1 to 3 for all places corresponding to identical rows in Pre$[P_p, T_p]$.

We will show the application of Algorithm 2 with the help of an example shown in Fig. 21. In this workflow, $T_P = \{a, d, e\}$ and $P_P = \{p_1, p_2, p_5, p_6, p_7, p_9\}$. Now we compute Pre$[P_P, T_P]$.

Pre $[P_P, T_P]$ =

	a	d	e
p_1	1	0	0
p_2	1	0	0
p_5	0	1	1
p_6	0	1	0
p_7	0	0	1
p_9	0	0	1

Rows corresponding to p_1 and p_2 are identical. Hence p_1 or p_2 can be implicit. Rows corresponding to p_7 and p_9 are identical. Hence p_7 or p_9 can be implicit. $\bullet p_1 = \{f\}$ No path exists from f to p_2, excluding p_1. Hence p_1 is not implicit. $\bullet p_2 = \{e\}$ Path from e to p_1 exists. Hence p_2 is implicit. $\bullet p_7 = \{d\}$ Path from d to p_9 exists. Hence p_7 is implicit. $\bullet p_9 = \{a\}$ Path from a to p_7 exists. Hence p_9 is implicit.

Thus implicit places are p_2 and p_7 or p_2 and p_9. We can remove these implicit places without affecting the behavior of the net. The workflow resulting from removal of places p_2 and p_9 is shown in Fig. 22.

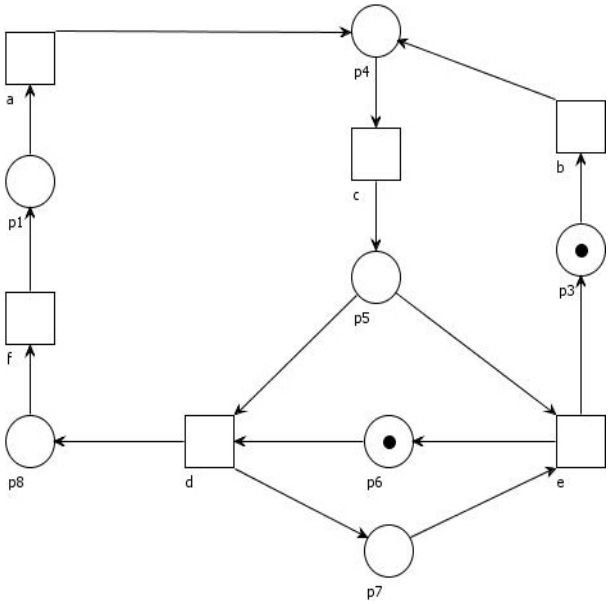

Figure 22: Example Workflow with implicit places removed

Algorithm 2 is helpful when deciding soundness [5] and consistency [4] of IOWFs with MLS because it reduces the size of Petri net models representing workflows.

VI. Conclusion

This paper has centered on developing new technique to verify correctness of loosely coupled IOWF with MLS features. The concepts of soundness and consistency were pursued for this purpose.

The first part of the paper examined various possible conceptual architectures for supporting inter-organizational workflows, multilevel security and various security models to implement multilevel security features. We modeled Car and Tire Company IOWF with loosely coupled architecture using Petri nets via connecting communication elements. An algorithm was presented to add multilevel security features to this IOWF using Bell-LaPadula security model. Thus using the approach and algorithms presented in this paper companies involved in e-commerce can review, analyze and test IOWF for correct behavior.

We reduced the local workflow structure, assuming they were executed correctly. The next step in this work would be to consider the effect of local workflow structure on the method to analyze for soundness and consistency. Algorithms presented in this paper can be extended with proper modifications for application to other IOWF architectures [17]. Specifications for real systems have a tendency to become large and complex. Future work will also aim at using Colored Petri nets for representation of IOWF.

References

[1] Atluri V. and Huang W.K., "An Authorization Model for Workflows", *Proceedings of the Fifth European Symposium on Research in Computer Security*, Rome, Italy, Lecture Notes in Computer Science, No.1146, Springer-Verlag, pp. 44-64, 1996.

[2] Atluri V. and Huang W.K., "An Extended Petri Net Model for Supporting Workflows in a Multilevel Secure Environment", *Proc. of the IFIP Working Conference on Database Security*, pp. 199–216, 1996.

[3] Gami, N., Mikolajczak, B. "Integration of Multilevel Security Features into Loosely Coupled Inter-Organizational Workflow", *Proc. Fourth International Conference on Information Technology: New Generations, ITNG '07*, April 2-4 2007, Las Vegas, Nevada, USA

[4] Gami, N., Mikolajczak, B., "*Consistency of Loosely Coupled Inter-Organizational Workflow with Multi-level Security Features"*, *Proc. of the 5th International Workshop on Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS 2007), 9th International Conference on Enterprise Information Systems*, Funchal, Madeira - Portugal, June 12-16, 2007, INSTICC Press.

[5] Gami, N., Mikolajczak, B., "Soundness of Loosely Coupled Inter-Organizational Workflows with Multi-level Security Features", *International Conference on Information and Knowledge Engineering IKE'07, within The 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing WORLDCOMP'07*, Las Vegas, Nevada, USA, June 25-28, 2007.

[6] Girault C., and Valk, R., "Petri nets for systems engineering: a guide to modeling, verification, and applications", Berlin, New York, Springer, pp. 278-281, 2003.

[7] Clark D.D. and Wilson D.R., "A Comparison of Commercial and Military Computer Security Policies", *In Proceedings of IEEE Symposium on security and Privacy*, pp. 184-194, 1987

[8] Graham G.S. and Denning P.J., "Protection - Principles and Practice", *In AFIPS Conference Proceedings,* Volume 40, Spring Joint Computer Conference, Montvale, New Jersey, 1972.

[9] Knorr, K., "Multilevel Security and Information Flow in Petri Net Workflows", *Proceedings of the 11th Conference on Advanced Information Systems Engineering*, Heidelberg, Germany, 2001.

[10] Bishop M., "Conspiracy and Information Flow in the Take-Grant Protection Model", *Journal of Computer Security,* Vol. 4, pp. 331-359, 1996.

[11] McLean J., "A comment on the basic security theorem of Bell and LaPadula", *Information Processing Letters*, Vol. 20, pp. 67-70, Elsevier North-Holland, New York, 1985.

[12] Myong, H.K, Froscher, N.J., Eppinger J.B. and Moskowitz S.I., "An Architecture for Multilevel Secure Interoperability", *Proceedings from Center for High Assurance Computer Systems,* Information Technology Division, Washington, DC, 1997.

[13] Myong H.K., Froscher N.J., Eppinger J.B. and Moskowitz S.I., "A Strategy for an MLS Workflow Management System", *IFIP Workshop on Database Security*, 1999.

[14] Anderson J. R., "Security Engineering: A guide to building dependable distributed systems", pp. 137-160, John Wiley & Sons, New York, 2001.
[15] W.M.P. van der Aalst, "Inter-organizational Workflows: An Approach based on Message Sequence Charts and Petri Nets", *Systems Analysis - Modeling - Simulation*, pp. 335-367, 1999.
[16] W.M.P. van der Aalst, "Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques", *Business Process Management, Models, Techniques, and Empirical Studies*, pp.16 -183, 2000.
[17] W.M.P. van der Aalst, "Loosely Coupled Inter-organizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries", *Information and Management,* pp. 67-75, 2000.
[18] http://is.tm.tue.nl/research/woflan/
[19] https://woped.ba-karlsruhe.de/woped/documentation/
[20] Li X., Hu J., Bu L., Zhao J. and Zheng G., "Consistency Checking of Concurrent Models for Scenario-Based Specifications", *Proceedings of 12th International SDL Forum*, Grimstad, Norway, pp. 298-312, 2005.

Boleslaw Mikolajczak was born in Poznan, Poland in 1946. He received Master of Engineering degree in control engineering from Technical University of Poznan in 1970 and Master of Science degree in Mathematics from Adam Mickiewicz University in 1972. He received Ph.D. degree in computer science from Technical University of Poznan in 1974 and Doctor Habilitis degree in computer science from Techncal University of Poznan. He specializes in foundations of computer science, in modeling, analysis, and design of parallel and distributed algorithms for computing systems using formal models such as automata and Petri nets and related model checking techniques.

Between 1974 and 1979 he was an Assistant Professor of computer science at the Technical University of Poznan, Poland. Between 1979 and 1986 he served as an Associate Professor at the Technical University of Poznan. He chaired a research group in automata, Petri nets and other models of computation between 1980-1986. Starting in 1986 he continued his academic career in the United States. Since 1995 he is a Professor of Computer and Information Science and Chairman of the Computer and Information Science at the University of Massachusetts Dartmouth.

He is an author of research monograph entitled "Periodic Transformations of Finite Automata" published by the Polish Academy of Sciences in 1979. He is also editor and co-author of a research monograph entitled "Algebraic and Structural Automata Theory" published by both Polish Academy of Sciences in 1986 (in Polish) and North-Holland in 1991, respectively. He is an author of over 100 journal and conference publications on modeling and analysis of parallel and distributed systems.

Dr. Mikolajczak is a member of IEEE Computer Society, ACM, and Polish Cybernetics Society. His biographies were published in several editions of Marquis Who is Who in the World, Who is Who in America, Who is Who in American Education and Who is Who in Engineering Education, and Who is Who in Polish America. Since 2005 he serves as an ABET professional evaluator of computer science programs.

Nirmal Gami was born in India. Between 2005 and 2007 he was a graduate student in computer science program at the University of Massachusetts Dartmouth. In 2007 he graduated with Master of Science degree in computer science.