A Case Study of Container Terminal Management using Ontology Knowledge based mapping on UML and CPN.

Authors:

Hamza Zafar

Kashif Mujeeb

Muhammad Farooq

Table of Contents

1. Abstract:

Trade of cargo is attached with the human life. As the life evolved; the trade pattern also evolved and cargo became transported in containers. And now container terminal management becomes a serious issue to manage; because it directly involves the money. In this research we have developed a simulation model which is made by ontology. The Ontology is mapped on UML and CPN Tool. The Ontology is made on Protégé whereas UML and CPN are made on Eclipse and CPN Tool respectively.

2. Introduction:

From the beginning of time human trade cargo but the means of trade at different times were different. Humans trade their cargo in bulk form or in bagged form over animals, boats. As the time evolved new technology arrived the trade patterns also evolved and human started trading on vehicle, instead of bulk they tried to manage it more and more. As the population increased there become need the transport cargo on sea. And now a days the trade main route is through sea because trade through sea is most cheapest and we can trade more cargo at once through the sea. The trade is being done in containers as whole cargo packing operation is done at exporter side and there is no worry of managing cargo at ship terminal side. But as the trade increased today the issue to manage the container terminal is becoming hot topic for research.

For the management of container terminal we constructed a simulation model in cpn tool and its consistency is maintained by developing a ontology, class diagram and activity diagram.

In rest of paper the system view of container trading is shown in section 3, the ontology development in section 4. The class diagram and activity diagram which are made in eclipse are discussed in section 5. The CPN mapping is shown in section 6. Finally, the consistency mapping is discussed in section 7. We also described the future work in section 8.

3. System Overview

The Trading system is mirror system in which the same procedure is being carried out in two different sub-systems i.e. Land and Sea. The Container comes to container terminal from exporter and from ship it goes to outer container terminal and from there it goes to importer. The figure No.1 describes the complete picture of system.

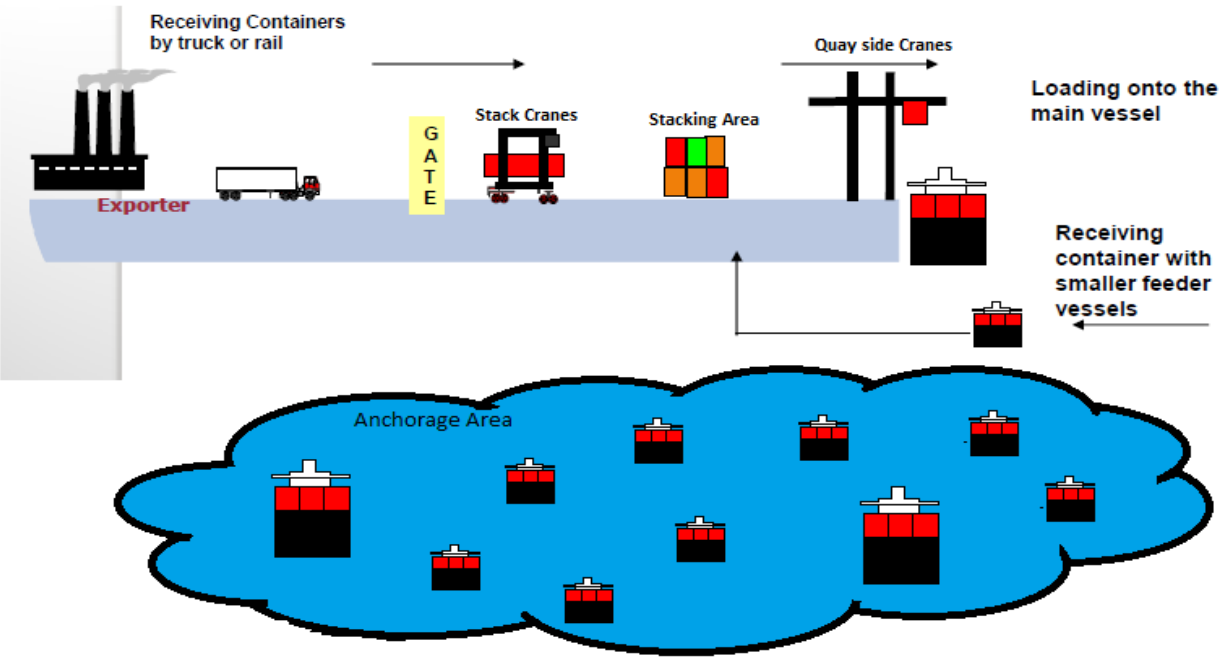

Figure No.1 System Diagram

As shown in the figure No.1 that from exporter the cargo is loaded on the vehicle (Trucks) and it reaches to the container terminal gate where it is placed in stacking area. The stacking area is the storage area for the container in container terminal which is used when container comes either side of transportations. From this area the export container is loaded on the ship with the help of Quay side cranes. And the import cargo is loaded on Lorries.

There are two types of ships shown in the figure Mother Container and Feeder. The difference in both ships is simple, that Mother has more capacity to carry container and it sails through long distance; we may say from one continent to other. In fact when a ship arrives at any port it does not berthed (connected) to the terminal directly, but it does when there is empty container terminal available. In case there is no terminal available then ship has to wait at anchorage area. When the terminal available the ships is berthed with the help of tug; a small ship.

4. Ontology:

The ontology of system diagram (figure No.1) is developed using active concept and non-active concept. The Active Concepts are those which perform action in the system and the Non-Active Concepts are whose which are passive and does not perform action. The consideration of active and non-active concepts can vary in different scenario. Another difference that active concepts always lay in the domain

of object properties whereas non-active concepts lay in the range of object properties.

We conceptually divided our whole system into three sub-system and build an ontology shown in figure No.2

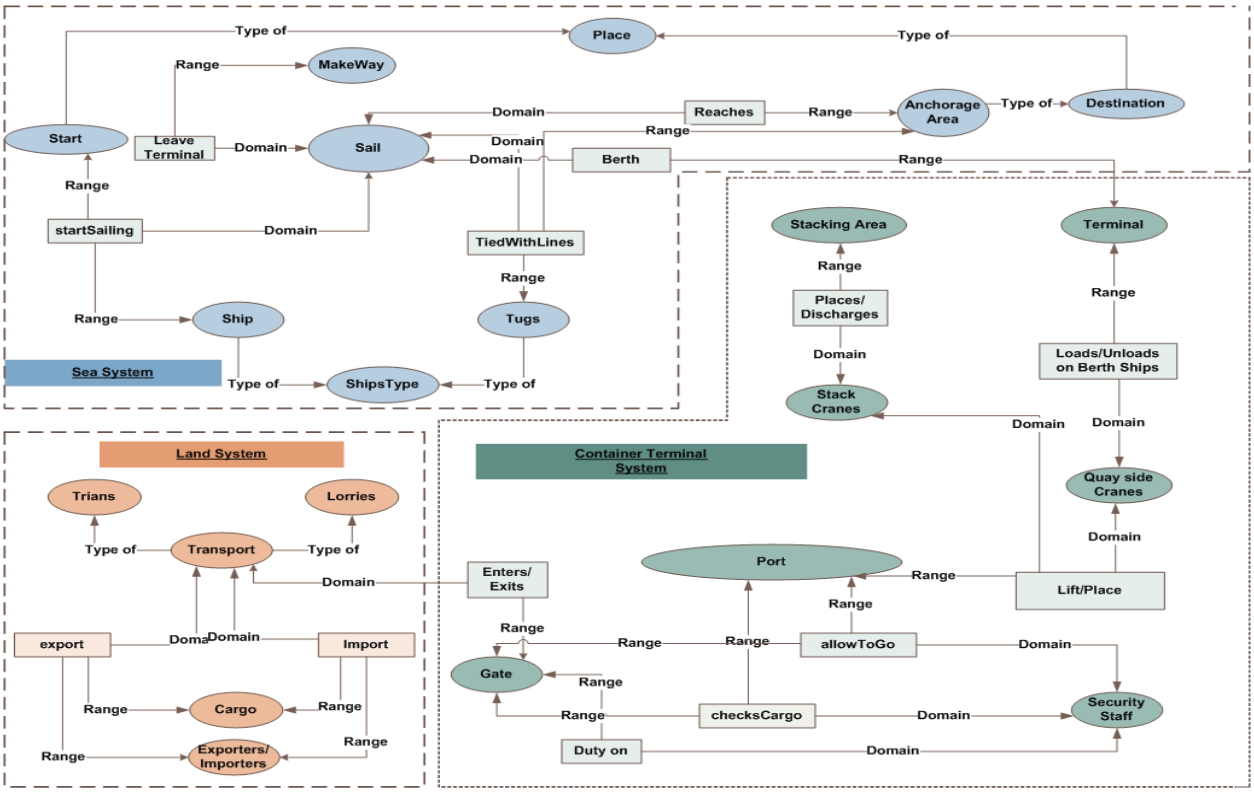

Figure No.2 Ontology

The figure No.2 describes the ontology of system. The three sub-system are separated by dotted lines. In Sea System the whole sea sailing and berthing is shown by Ship, Tug, destination and Sail concepts whereas the object properties shows the role of concepts. Similarly the Land system shows the import and export procedure from the transport. In container Terminal System the Securtiy checking and cargo stacking, loading and discharging of berthed is shown with the help of concepts and object properties. This ontology is then mapped in protégé so that it may be machine understandable. In figure No.3 Protégé Ontolgy diagram made in Jamablaya is shown.

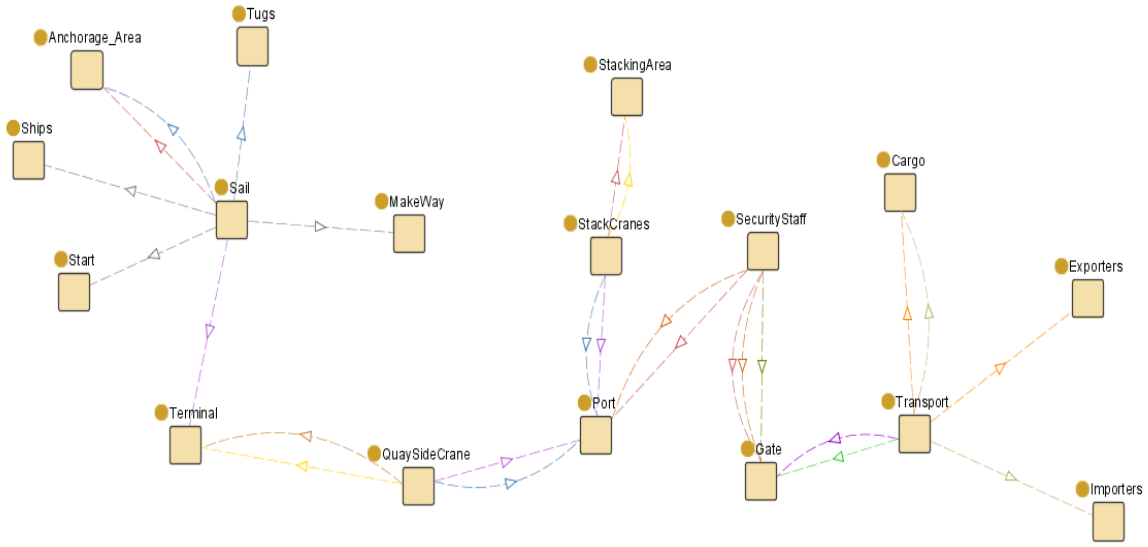

Figure No.3 Ontology by Jamblaya

5. Unified Modelling Language:

Since Ontology gives the description of concepts. From ontology one can't understand the sequence of action of system, so there is need to develop an activity diagram. Since activity diagram cannot be made without the class diagram, so we used eclipse geneymade and constructed a class diagram and then the activity diagram.

5.1. Class Diagram:

The all active concepts in the ontology are mapped on the concrete class and object properties are mapped operations in the concrete class. The non-active concepts are made on the association classes. The data type properties defined in the protégé are made as attributes of the restive class. The class diagram is shown in figure No.4.

5.2. Activity Diagram:

The Activity Diagram is shown in figure No.5. In activity diagram the active concepts are mapped in swim lane. Whereas the non-active concepts are mapped at data store for holding the data. The object properties were created as action node. From start node we traverse to node by control flow or data flow. The data flow have arguments (data type in the ontology).

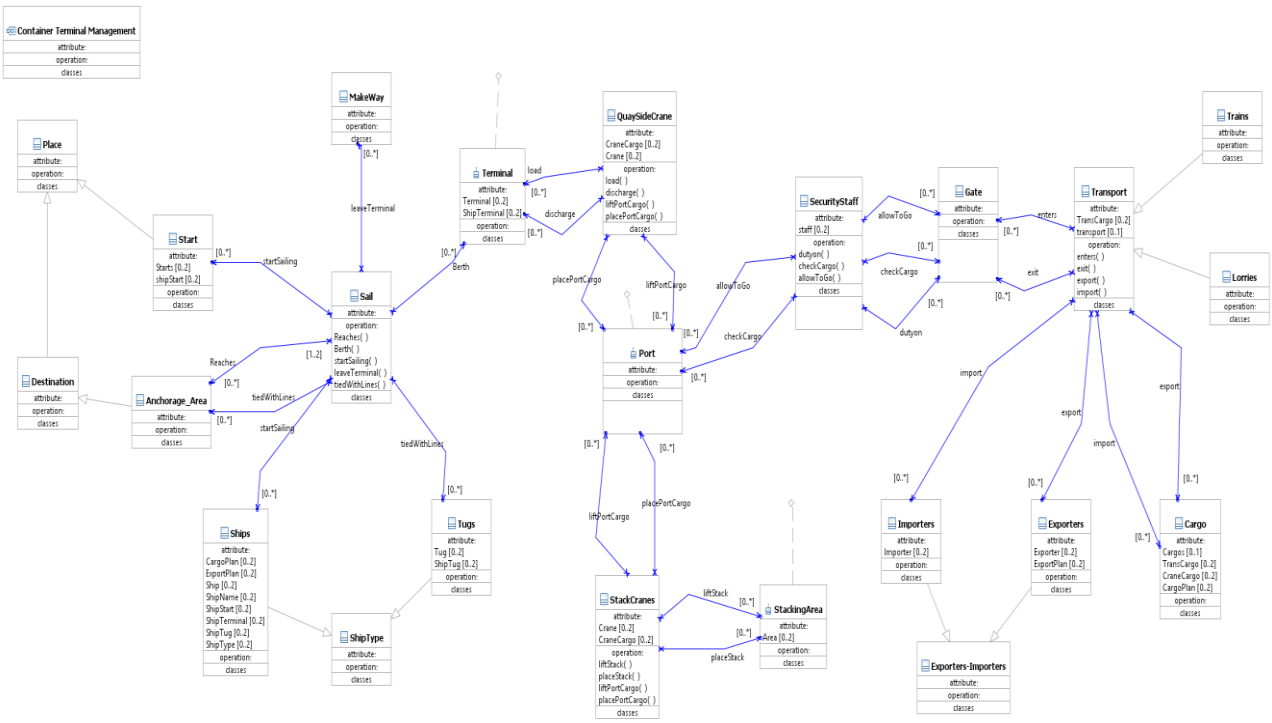

Figure No.4 Class Diagram generated from the Protégé.

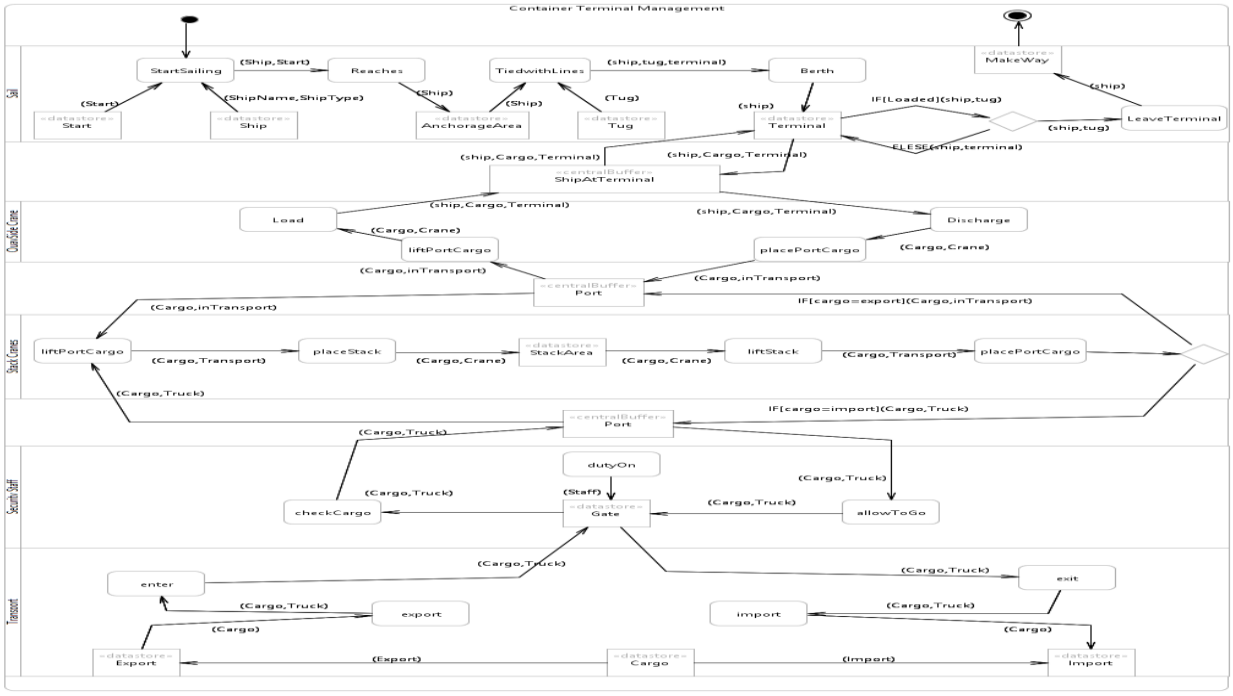

Figure No.5 Activity Diagram

The activity starts at startSailing action which has two datastores connected start and ship. Then it action takes to other action to reaches with arguments (ships, start). And with the berth action node it reaches at the terminal where we have

defined condition mark to for sailing activity. When the ship is loaded and discharged then it must leave the terminal. The other swim lane, data stores, buffer describes the overall flow of procedure.

6. CPN Tool Simulation:

We build our simulation using ontology by considering the examples of Karachi International Container Terminal. The cpn shows simulation of our domain of focus. We considered the limited resource present at the KICT and then simulate the tokens. The super page of the cpn is shown at figure No.6. figure No.7 to figure No.11

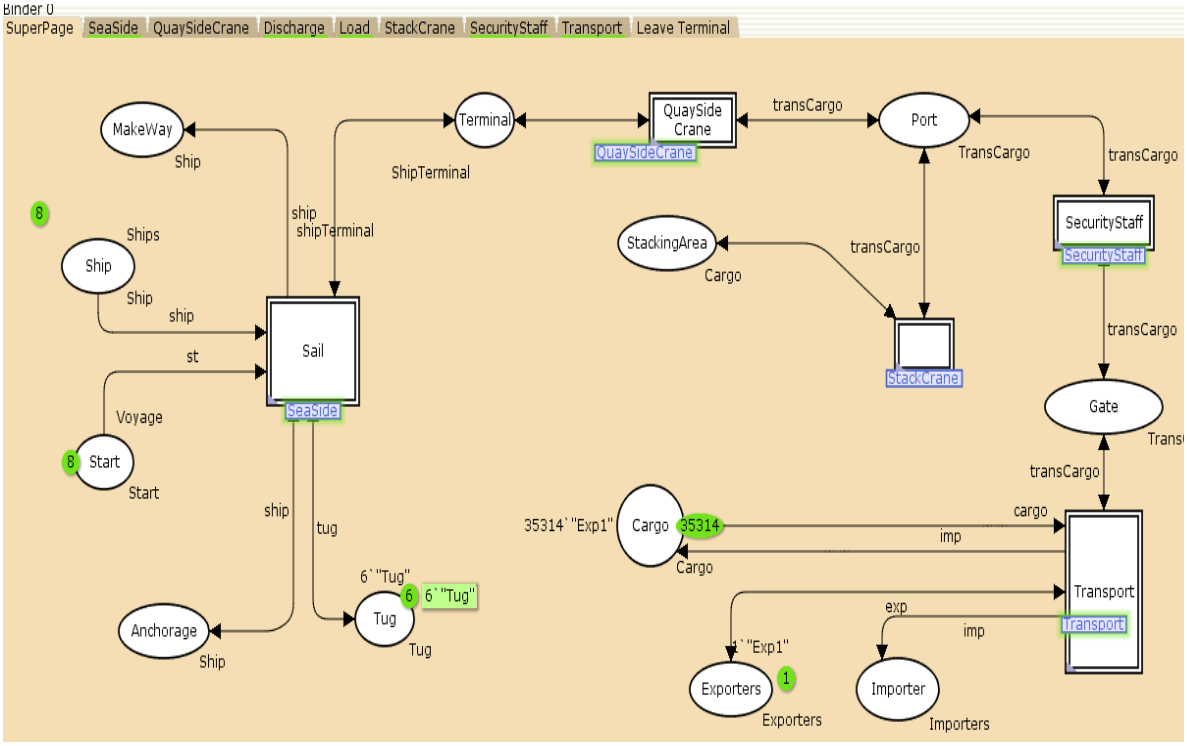

Figure No.6 CPN Super Page

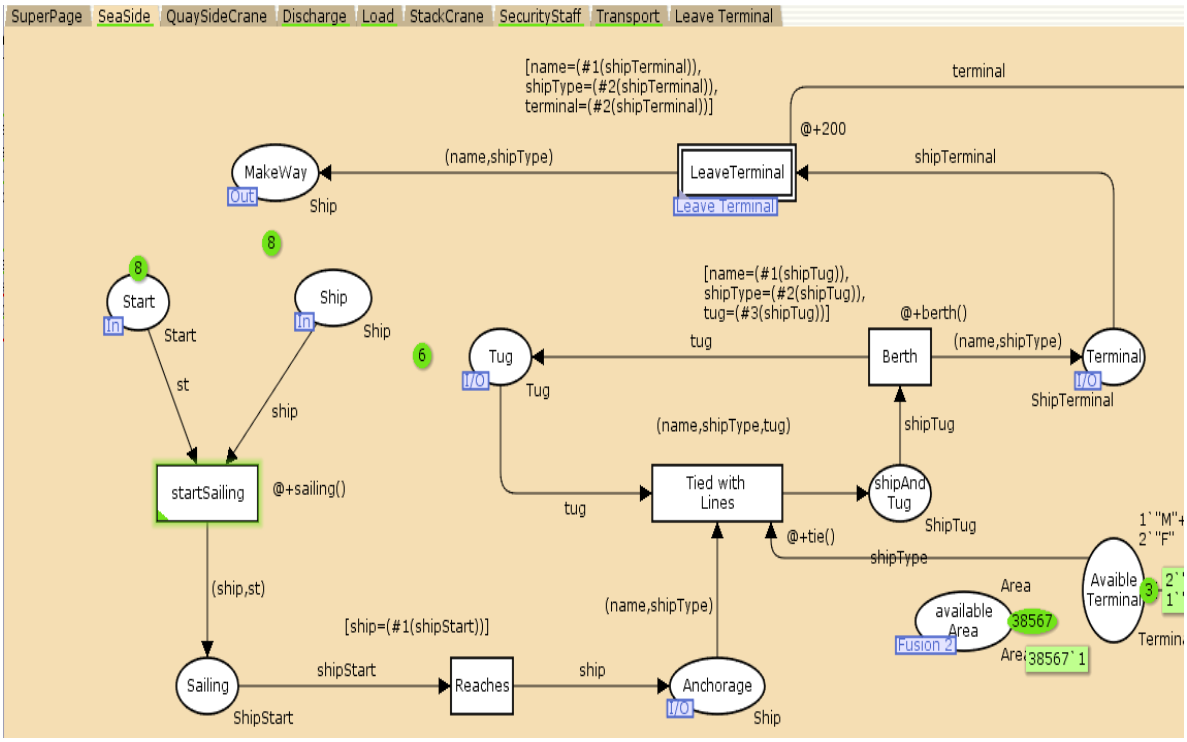

Figure No.7 Sea Side Page

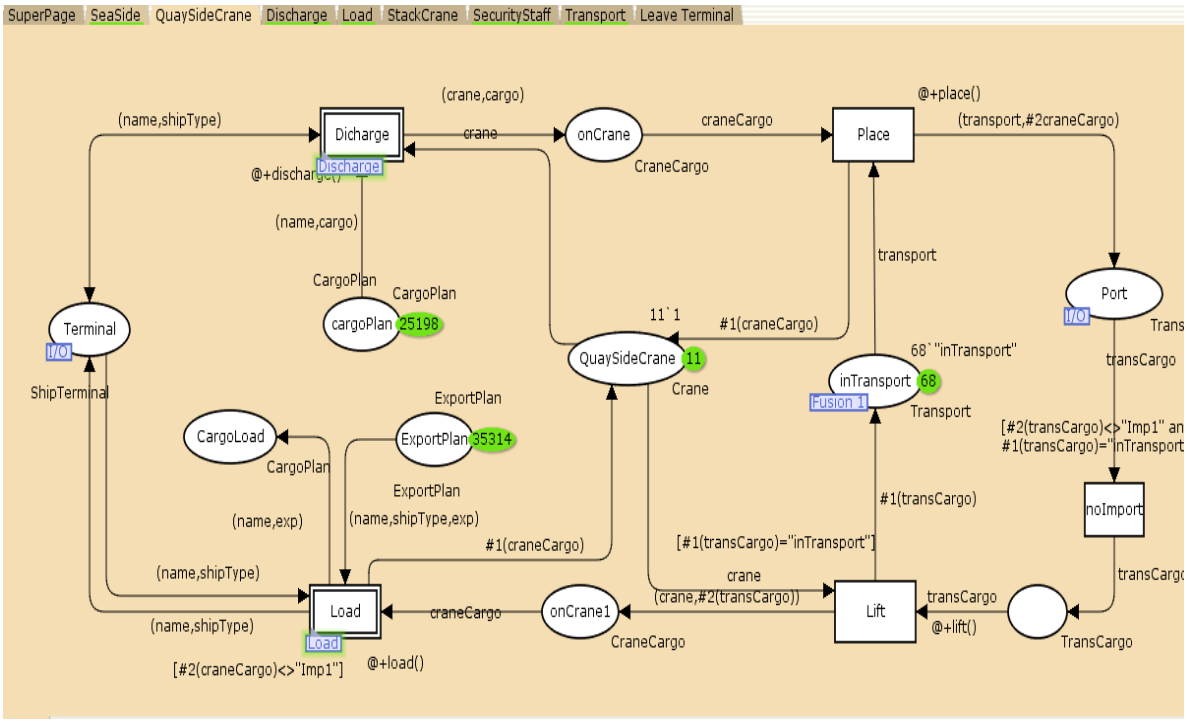

Figure No.8 Quay Side Crane

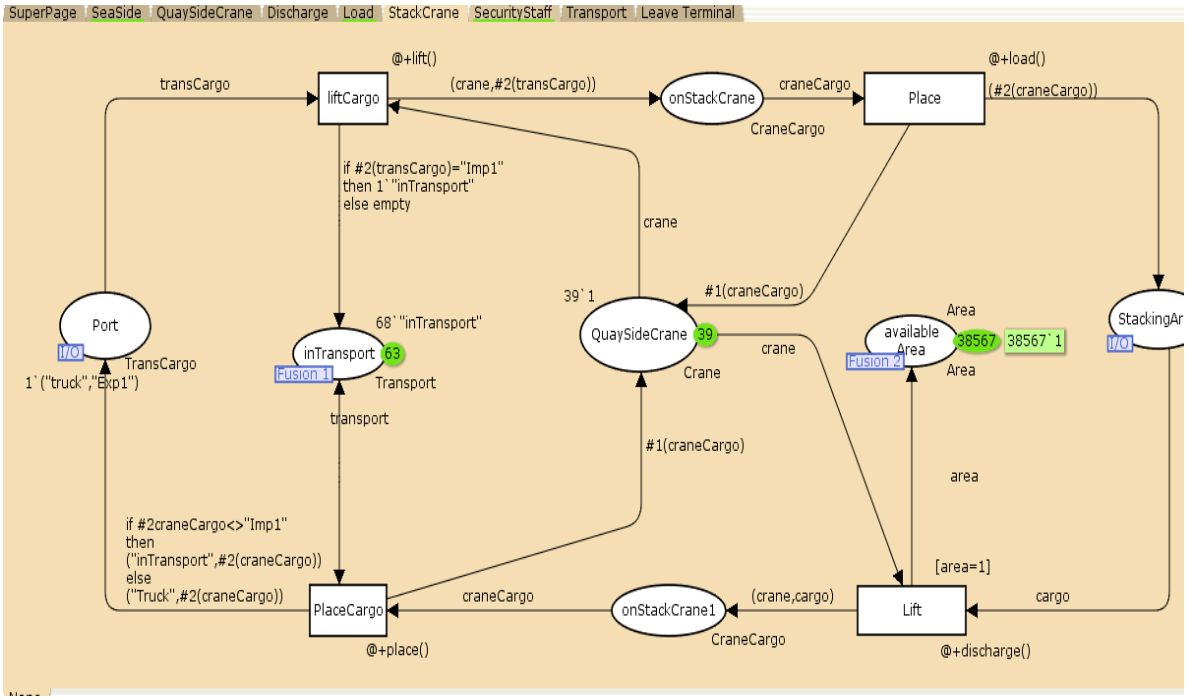

Figure No.9 Stack Crane

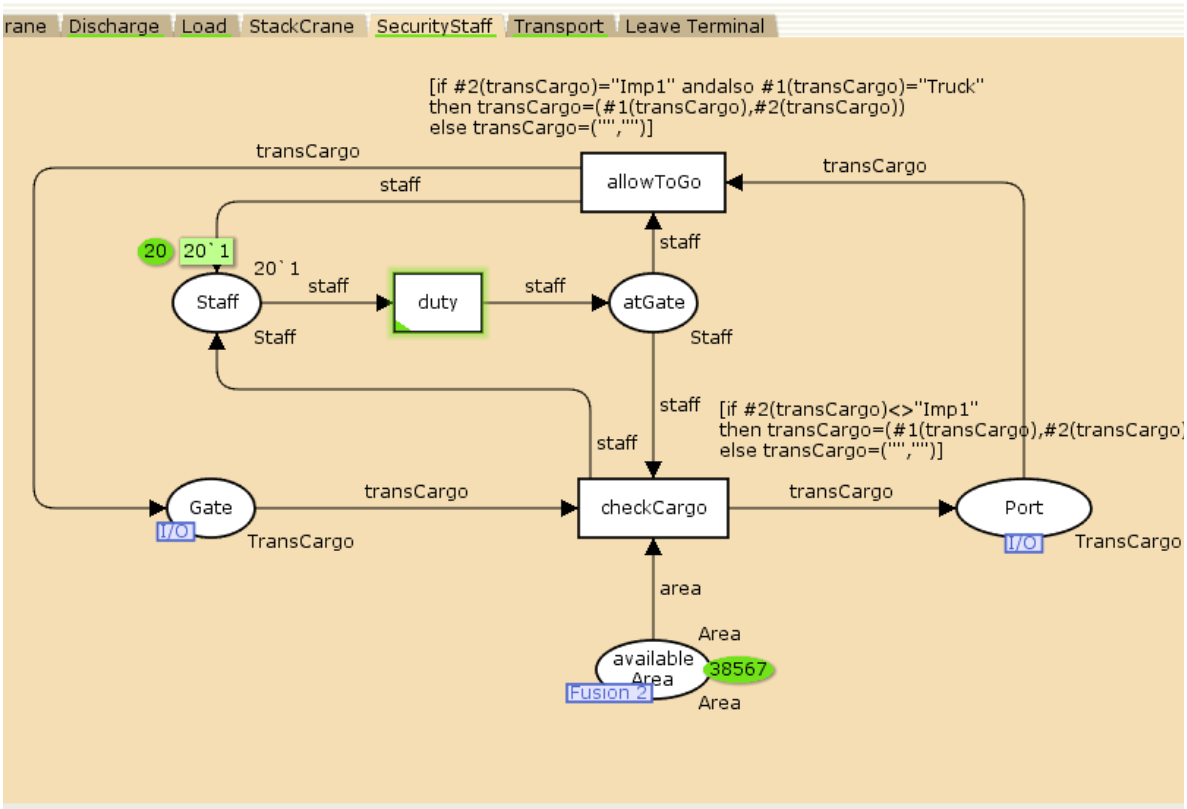

Figure No 10 Security Staff

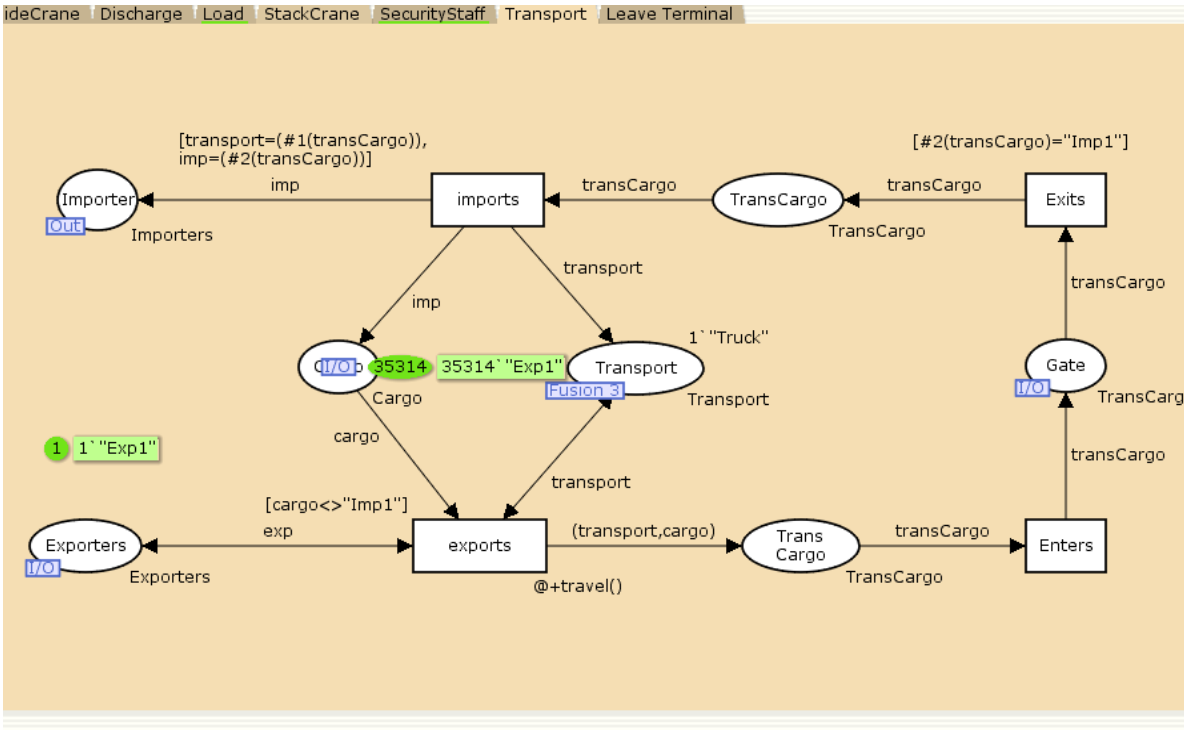

Figure No 11. Transport

describes the subpages. The super page describes overall system overview. And the Sail, Quay -Side, Stack Crane, Security Staff and Transport transition matches the Active concepts where as the all object properties are made in respective transition.

In Quayside crane the crane lift cargo from the terminal ships, and place it at the inTransport. And also lift cargo from the inTransport and loaded it to the Termianl ship. The inTransport travel at port and takes cargo from the stacking area to terminal.

If we consider the stacking Crane conspect in this concept the crane lift cargo and place on the stack.

The cargo imports and exports from the gate, where security staff is present to check the cargo. The Transport take cargo from the exporter and give cargo to the importers.

7. Consistency Mapping

The all concepts are made consistently on Ontology to CPN Tool through mapping. In Figure 12, Figure 13 and Table 1 we present the dataType and swrl Mapping.

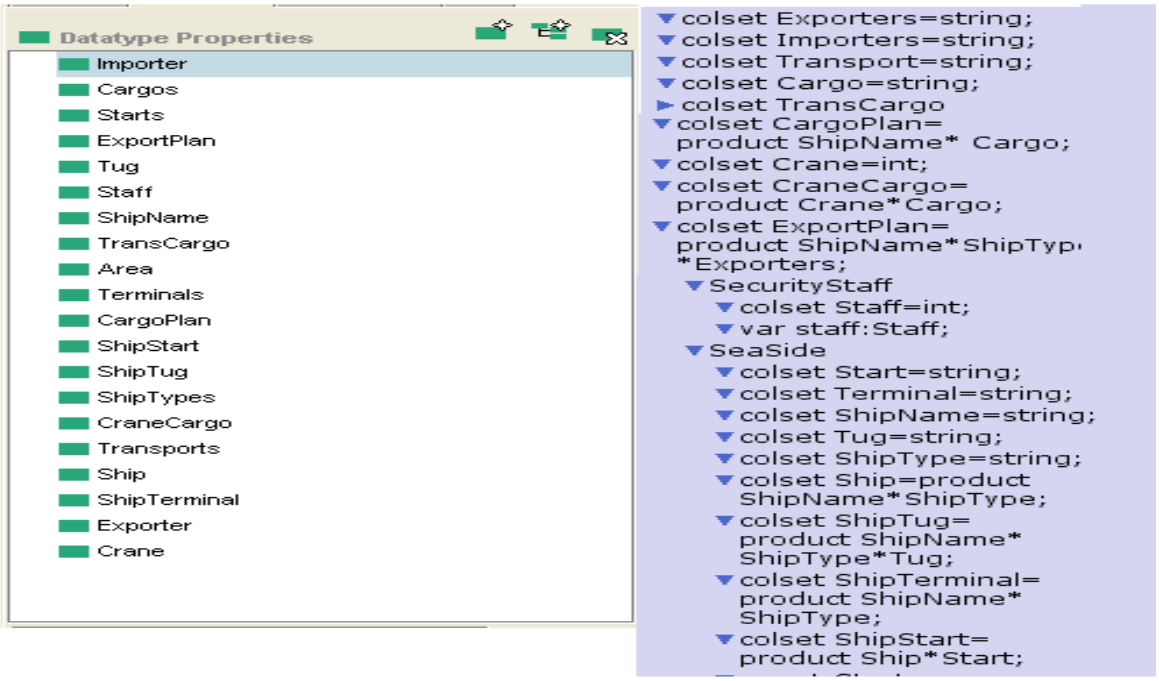

Figure No.12 Data Type Mapping

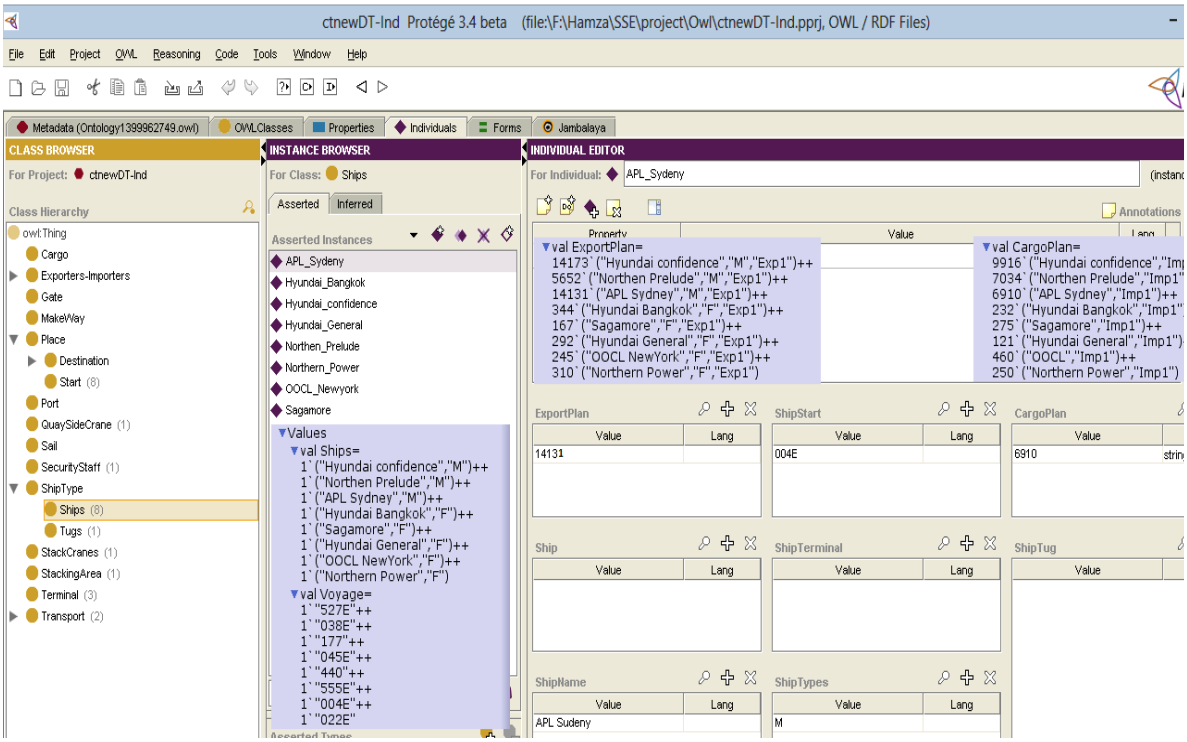

Figure No13. Instances Mapping

Cargo(?c) ∧ **Transport(?t) ∧** **allowToGo(?t, ?c) ∧** **swrlb:equal(?c, "Imp1") ∧** **swrlb:equal(?t, "Truck")** **→ abox:setValue(?t, "Truck") ∧** **abox:setValue(?c, "Imp1")**	[if #2(transCargo)="Imp1" andalso #1(transCargo)="Truck" then transCargo=(#1(transCargo),#2(transCargo)) else transCargo=("","")]
Cargo(?c) ∧ **Transport(?t) ∧** **checksCargo (?t, ?c) ∧** **swrlb:equal(?c, "Exp1")** **→ abox:setValue(?t, "Truck") ∧** **abox:setValue(?c, "Exp1"**	[if #2(transCargo)<>"Imp1" then transCargo=(#1(transCargo),#2(transCargo)) else transCargo=("","")]

Table No.1 SWRL Mapping to CPN Guard Condition

The Table No.1 describes the SWRL mapping to guard If condition. In 1st SWRL we used allowToGo property which is mapped on the allowToGo transition on CPN. And the CheksCargo Object property is mapped at the checkCargo Transition at cpn tool.

8. Futue Work:

We have tried to mapped the whole trading system in this model. Since this domain considered the time; so we will include the time concept in the model.
Also the more SWRL rule must be included to have better simulation view of the domain and resource management.

Table 1.Create Ontology

<table>
<tr><th colspan="3">Step 3.1</th><th>Step 3.2</th><th>Step 3.3</th><th colspan="4">Step 3.4</th></tr>
<tr><th>Step 3.1.1</th><th>Step 3.1.2</th><th>Step 3.1.3</th><td rowspan="2"><u>Active Concept:</u>

Sale
Quay Side Crane
Stack Crane
Security Staff
Transport

<u>Non-active Concept:</u>

Start
Anchorage Area
Ship
Tug
Terminal
Stacking Area
Gate
Importer
Exporter</td><td rowspan="2"><u>Concept:</u>
<u>Property:</u>
<u>Value type:</u>

This step is repeated for all the properties in the OP Ontology.</td><td colspan="4"><u>Object property with Domain Range:</u></td></tr>
<tr><td><u>Concepts:</u>

Place
- Start
- Destination
-- Anchorage Area
Ship Types
-Ship
-Tugs
Sail
Terminal
Quay Side Crane
Stacking Area
Stack Crane
Port
Security Staff
Gate
Transport
-Lorries
-Trains
Exp-Imp
-Exporter
-Importer</td><td><u>Datatype Properties/slots:</u>

Cargo
Importer
Starts
Export Plan
Tug
Staff
Ship Name
TransCargo
Area
Terminal
Cargo Plan
Ship Start
Ship Tug
Ship Types
Crane Cargo
Transport
Ship
Ship Terminal
Exporter
Crane</td><td><u>Object Property:</u>

Enters
Import
Reaches
Exports
Allow to go
Lift stack
Exit
Load
Place Stack
Discharge
Place port cargo
Leave terminal
Tied With Lines
Lift port cargo
Start Sailing
Berth
Duty on
Checks Cargo</td><td colspan="3"><table>
<tr><th>Property</th><th>Domain</th><th>Range</th></tr>
<tr><td>Start Sailing</td><td>Sail</td><td>-Start
-Ship</td></tr>
<tr><td>Leave_Terminal</td><td>Sail</td><td>-Make Way
-Terminal</td></tr>
<tr><td>Reaches</td><td>Sail</td><td>Anchorage Area</td></tr>
<tr><td>Tied With Lines</td><td>Sail</td><td>Tug</td></tr>
<tr><td>Berth</td><td>Sail</td><td>Terminal</td></tr>
<tr><td>Export</td><td>Transport</td><td>-Cargo
-Exporter</td></tr>
<tr><td>Import</td><td>Transport</td><td>-Cargo
-Importer</td></tr>
<tr><td>Enters</td><td>Transport</td><td>Gate</td></tr>
<tr><td>Exits</td><td>Transport</td><td>Gate</td></tr>
<tr><td>Duty on</td><td>Security Staff</td><td>-Gate</td></tr>
<tr><td>Checks Cargo</td><td>Security Staff</td><td>-Gate
-Port</td></tr>
<tr><td>Allow to Go</td><td>Security Staff</td><td>-Gate
-Port</td></tr>
<tr><td>Place</td><td>Stack Crane</td><td>Stacking Area</td></tr>
<tr><td>Discharge</td><td>Stack Crane</td><td>Stacking Area</td></tr>
<tr><td>Lift</td><td>-Stack Crane
-Quay Crane</td><td>Port</td></tr>
<tr><td>Place</td><td>-Stack Crane
-Quay Crane</td><td>Port</td></tr>
<tr><td>Load</td><td>Quay Crane</td><td>Terminal</td></tr>
<tr><td>Unload</td><td>Quay Crane</td><td>Terminal</td></tr>
</table></td></tr>
</table>

Table 2.Generate Class Diagram from OWL Ontology

Step 4.1	Step 4.2	Step 4.3	Step 4.4

Classes:

Places
-Start
-Destination
--Anchorage Area
Make Way
Ships
-Ships Type
-Tug
Sail
Terminal
Quay Side Crane
Port
Stack Cranes
Stacking Area
Security Staff
Gate
Cargo
Transport
-Lorries
-Trains
Exp-Imp
-Exporter
-Importer

Concrete Classes:
Sail
Quay Side Crane
Stack Crane
Security Staff
Transport

Association Classes:
Start
Anchorage Area
Ships
Tugs
Make Way
Terminal
Port
Stacking Area
Gate
Importer

Class:
Attribute:

Start
-start -ship Start

Ships
-cargo Plan -Export Plan -ship -ship Name -Ship Start -Ship Terminal -Ship Tug -Ship Type

Tug
-Tug -Tug Ship

Terminal
-Terminal -Ship Terminal

Quay Crane
-Crane -Crane Cargo

Stack Crane
-Crane -Crane Cargo

Stack Area
-Area

Security Staff
-Staff

Class (Operation/function) & Associations:

Sail	
Operation	**Association**
-Reach	-Anchorage
-Berth	-Terminal
-Start Sailing	-Start
-leave Terminal	-Make Way
-Tied with Lines	-Anchorage -Tug

Quay Side Crane	
Operation	**Association**
-Load	-Terminal
-Discharge	-Terminal
-lift Port Cargo	-Port
-Place Port Cargo	-Port

Stack Crane	
Operation	**Association**
-Lift Stack	-Stacking Area
-Place Stack	-Stacking Area
-lift Port Cargo	-Port
-Place Port Cargo	-Port

Security Staff	
Operation	**Association**
-duty on	-Gate
-Checks Cargo	-Gate -Port
-allow to go	-Gate -Port

Transport	
Operation	**Association**

Association:

Navigable end:

Non Navigable end:

Cardinality:

This step is repeated for all the cardinalities in the OP Ontology

<table>
<tr>
<td>Exporter
Cargo</td>
<td>
<table>
<tr><td>Transport</td></tr>
<tr><td>-transport
-trans Cargo</td></tr>
</table>
<table>
<tr><td>Cargo</td></tr>
<tr><td>-cargo
-transCargo
-crane Cargo
-cargo Plan</td></tr>
</table>
</td>
<td>
<table>
<tr><td>-Enter</td><td>-Gate</td></tr>
<tr><td>-Exit</td><td>-Gate</td></tr>
<tr><td>-Export</td><td>-Exporter
-Cargo</td></tr>
<tr><td>-Import</td><td>-Importer
-Cargo</td></tr>
</table>
</td>
<td></td>
</tr>
</table>

Table 3.Create CPN model using Onto-UML to CPN mapping scheme

colset:	**product colset:**	**variables:**	**Substitution Transition:**	**Places:**	**Starting Place:**	**Input Arcs:** **Transition:** **Places:** **Output Arcs:** **Places:** **Transition:** *This step is repeated for all Object Flow Arrows in theOPActivity diagram.*	**Arc Expressions:**	**Subpage**	**Transitions:**	**Input Arcs:** **Transition:** **Outputport:** *This step is repeated for all Object Property of Particular active concepts.*	**Output Arcs:** **Inputport:** **Transition:** *This step is repeated for all Object Property of Particular active concepts.*	**Manual addition of Intermediate places**
Exporter Importer Transport Cargo Staff Start Terminal Ship Name Tug Ship Type Crane Area	TransCargo Ship Ship Tug Ship Termianl Ship Start CargoPlan CraneCargo ExportPlan StackArea	exp imp transCargo cargo transport staff St shipStart ship terminal tug shipTug shipTerminal name shipType cargoPlan crane craneCargo exportPlan area stackArea	Sail Quay Crane StackCrane Security Staff Transport	Ship Start Make way Anchorage Tug Terminal Port Stacking Area Gate Cargo Exporter Importer	Start		.	**Sea Side** **Input:** Start Ship **Output** MakeWay **I/O:** Anchorage Tug Terminal **Quay Crane** **I/O:** Terminal Port **Quay Crane** **I/O:** Stack_Area Port **Security Staff** **I/O:** Gate Port **Transport** **I/O:** Gare Exporter	**Person Subpage** **Sea Side** -Reach -Berth -Start Sailing -leave Terminal -Tied with Lines **Quay Crane** -Load Discharge -lift Port Cargo -Place Port Cargo -Load **Transport** -Enter -Exit -Export -Import -Enter **Security Staff** -duty on -Checks Cargo -allow to go			
							St Ship shipStart tug shipType terminal craneCargo (name,exp) transCargo exp imp					