Risk Modeling and Analysis of Interdependencies of Critical Infrastructures Using Colored Timed Petri Nets[⋆]

Chrysovalandis Agathangelou, Chryssis Georgiou, Ileana Papailiou,
Anna Philippou, Loucas Pouis, Georgios Tertytchny, and Despina Vakana

Department of Computer Science, University of Cyprus
Email: fcslab@cs.ucy.ac.cy

Abstract. Petri Nets (PNs) and their variations are a graphical, mathematical language that can be used for the specification, analysis and verification of discrete event systems, including Critical Infrastructures (CIs). Colored PNs are an extension of classical PNs that are suitable for modeling and analyzing complex interconnected CIs. Timed PNs are another extension of PNs that support timing constraints and events. In this work we present a novel Risk Assessment methodology based on Timed Colored PNs for modeling and analyzing CIs with interdependencies, time-critical events and cascading effects.

Keywords: Risk assessment, Critical Infrastructures, Interdependencies, Cascading effects, Timed Colored Petri Nets

1 Introduction

Motivation: A *Critical Infrastructure* (CI) is an asset, system or part which is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, and the disruption or destruction of which would have a significant impact as a result of the failure to maintain those functions [3]. Therefore, CIs are vital and must meet their security constrains and specifications (e.g., time constrains) to the highest degree possible. In such infrastructures, *interdependencies* between the various key components (physical or cyber) within or across infrastructures, and *cascading effects* (i.e., a malfunction of one component affects another, and so on) are crucial (see, e.g., [16, 11, 12, 15, 19, 18]). To this respect, a lot of research has been conducted in developing Risk Assessment (RA) methodologies for CIs dealing with interdependencies, time-critical events and cascading effects (e.g., [10–12, 6, 14]).

The Petri Nets (PNs) formalism [13, 4] is a powerful mathematical modeling and analysis method applicable to a wide range of domains. The Colored Petri Nets (CPNs) formalism [9] is an extension of PN's that is particularly suited to modeling and analyzing complex interconnected systems [1]. In particular, colors can be used to model different interactions between components or transect

[⋆] This work is supported by the European Commission under grant agreement HOME/2013/CIPS/AG/4000005093 (MEDUSA: http://medusa.cs.unipi.gr/).

interdependencies between them. Furthermore, CPNs support a large number of formal analysis methods including verification and model-checking and are associated with a wide range of tools [9, 7]. With respect to RA, CPNs are suitable for both system modeling (model the system's behavior) and accident modeling (model the accident sequence of a system) [1, 20]. The Timed Colored Petri Nets (TCPNs) formalism [8] enhances the basic CPN framework with timing considerations. From the above, one may conclude that the TCPN framework is a natural choice for developing RA methodologies for CIs with interdependencies, time-critical events and cascading effects. To the best of our knowledge, no such TCPN-based RA methodology has been proposed so far.

Contribution: This work presents a novel application of TCPNs to RA of CIs with interdependencies, time-critical events and cascading effects. Our methodology exploits the power of CPNs in modeling interdependencies and cascading effects as well as the various risks involved in complex CIs, together with the timing considerations offered by Timed CPNs (e.g., for calculating the delay of the process on one component due to the effect of a risk to another component). Although our motivation for this research is to study the interdependencies and cascading effects on sea ports' supply chain, our proposed methodology is suitable for general CIs, and it is presented as such.

Related Work: Extensive research has been conducted for RA of CIs, in general, but also for considering interdependencies and cascading effects (e.g., [16, 11, 12, 18, 5, 19, 15, 6, 14]). Also, the PN formalism and its variations (Colored, Timed, Stochastic, etc) has been used for RA. For example, Vernez et al. [20] discuss the uses and the application perspectives of basic PNs in the field of risk analysis and accident modeling. Aloini et al. [1] argue how CPNs can be used to model risk factors in Enterprise Resource Planning (ERP) projects, focusing on the problem of interdependence in risk assessment. Bernardi et al. [2] propose a method for assessing the risk of timing failures by evaluating the design of a specific software. They use Timed Petri Net (TPN) modeling and analysis techniques, and show the effectiveness of their method, based on a case study of a real-time embedded system. Rossi et al. [17] describe a method for identifying and assessing operational risks in a supply chain. Their approach uses the analogy between logistics networks and dynamical systems and it applies attributed Petri nets and associated tools. However, to the best of our knowledge, our work is the first to combine CPNs and TPNs for developing a RA methodology for CIs while dealing with interdependencies, timing constrains and cascading effects.

Paper organization: In Sect. 2 we give a brief overview of TCPNs and in Sect. 3 we present our RA methodology. Sect. 4 presents a case study on a simplified real-world scenario on ports' supply chain. Sect. 5 concludes the paper.

2 Timed Coloured Petri Nets

Petri Nets (PNs) are a graphical, mathematical language that can be used for the specification, analysis and verification of discrete event systems [13]. They are associated with a formal execution semantics based on which a rich mathematical theory for process analysis has been developed. Since their inception by Carl Adam Petri in 1962, they have been applied to a wide range of applications and they have been extended into a variety of formalisms such as Timed Petri Nets, Stochastic Petri Nets, and Colored Petri Nets [4].

In its basic form, a Petri net is a directed graph consisting of places (denoted as circles), transitions (denoted as bars or rectangles), and directed arcs that connect places to transitions and vice versa, as illustrated in Figure 1. Places in a Petri net may contain a discrete number of marks called *tokens* which determine the execution/firing of transitions: a condition on a transition determines that the transition is taken if a sufficient number of tokens exists on the place preceding the transition in which case it will consume the required number of tokens while creating tokens on the places following the transition.

Colored Petri Nets (CPNs) are an extension of classical PNs in which tokens may be differentiated with the use of colors [20]. Each token is assigned a color from a color set so it is possible to distinguish one from another while modeling complex behaviors. Moreover, CPNs combine the capabilities of PNs with those of the Standard ML programming language. The resulting flexibility and computational power make them a suitable formalism for modeling and analysing risk in critical infrastructures as is the goal of our work. More precisely, since the notion of time is essential in the problem under study, we employ a timed extension of CPNs, namely, Timed Colored Petri Nets (TCPNs) which allows to add timing information to CPN models [8]. A TCPN is defined as follows:

Definition 1. [8] A *timed non-hierarchical Colored Petri Net* is a nine-tuple $TCPN = (P, T, A, \Sigma, V, C, G, E, I)$ where:

1. P is a finite set of *places.*
2. T is a finite set of *transitions* such that $P \cap T = \emptyset$.
3. $A \subseteq P \times T \cup T \times P$ is a set of directed *arcs.*
4. Σ is a finite set of non-empty *colour sets.* Each colour set is either untimed or timed.
5. V is a finite set of *typed variables/tokens* such that $\mathit{Type}[v] \in \Sigma$ for all variables $v \in V$.
6. $C : P \to \Sigma$ is a *set function* that assigns a colour set to each place. A place p is timed if $C(p)$ is timed, otherwise p is untimed.
7. $G : T \to \mathit{EXPR}_V$ is a *guard function* that assigns a guard to each transition t such that $\mathit{Type}[G(t)] = \mathit{Bool}$ and EXPR_V is the set of Standard ML expressions making references to variables in V.
8. $E : A \to \mathit{EXPR}_V$ is an *arc expression function* that assigns an arc expression to each arc α such that
   - $\mathit{Type}[E(\alpha)] = C(p)_{MS}$ if p is untimed;
   - $\mathit{Type}[E(\alpha)] = C(p)_{TMS}$ if p is timed.

   Here, p is the place connected to the arc α.
9. $I : P \to \mathit{EXPR}_\emptyset$ is an *initialisation function* that assigns an initialisation expression to each place p such that
   - $\mathit{Type}[I(p)] = C(p)_{MS}$ if p is untimed;

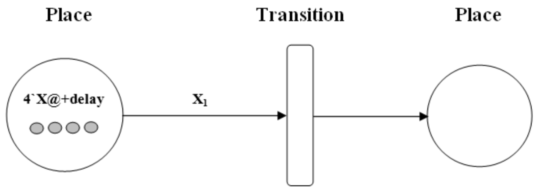

Fig. 1. Simple Example of a Timed Colored Petri Net

- $Type[I(p)] = C(p)_{TMS}$ if p is timed.

A simple example of a TCPN can be seen in Figure 1. Specifically, there are two places of color-set X and one transition. The initial marking of the first place has four tokens (typed variables) of color-set X which is timed. These tokens will be ready to be removed by the occurring transition at some time delay. Hence, a token starts from the first place and goes to the second through the transition.

3 Methodology

In this section we describe our methodology for risk assessment of critical infrastructures that takes into consideration interdependencies and cascading effects. The proposed methodology is based on TCPNs and it allows to construct models of CIs at various levels of abstraction due to its modularity. It also allows us to model risks and their impact on different components of an infrastructure, interdependencies that may exist between risks and cascading effects of events between different components of an infrastructure. As such, the results obtained can be used to evaluate adopted procedures and modify them as necessary or explore alternative security measures in order to limit the effects of various risks.

The main phases of our methodology are the following: 1) Establish the context. 2) Risk identification. 3) Cascading effects identification. 4) Model construction. 5) Model evaluation. We now proceed to describe each phase.

1. Establish the context: The goal of this step is to establish the boundaries and characteristics of the process under study. This involves the specification of the infrastructures involved and the entities and activities operating therein. Consider a set Inf of infrastructures. For each infrastructure, we identify the *entities* participating in the process. These may involve people, organizations, documents, resources, etc. Once these are identified, the level of abstraction of the model is determined. For instance, if we consider a port supply chain and an associated entity *cargo*, the cargo could be considered as a single unit or as a set of individual containers. Following this consideration, a set $\mathcal{E}_i$ is identified containing all entities appearing in infrastructure $i \in Inf$.

The second component of interest is that of *control points* or *states*. Control points are different states of infrastructures in the process under study. They are characterized by the activities in which they may engage which may take place either within or between infrastructures and are carried out by entities and upon entities involved in the system. Thus states are connected with each other by activities and sequences of activities correspond to procedures and may involve physical or electronic flows, capital flows, information flows, etc. Specifically, for each infrastructure $i \in \mathit{Inf}$, we identify the following:

- the set S_i of states;
- a function *entities* that associates each state $s \in S_i$ with the set of entities from $\mathcal{E}_i$ available to the state;
- a set $\mathcal{A}_i^j$ of pairs of the form (s, s') where s is a state of infrastructure i and s' is a state of some infrastructure j (where possibly i and j coincide) such that state s may execute a transition that provides input to state s';
- for each $(s, s') \in \bigcup_j \mathcal{A}_i^j$ an expression $e_{s,s'}$ that describes the coupling between states s and s' and, in particular, how s manipulates its entities to produce a new set of entities that will be consequently available to state s'.

As an example, consider two control points in some infrastructure i involved in off-loading cargo from a ship, the first, s, being the state where accompanying documentation must be verified and the second, s', being the state where allocation of resources for off-loading is decided. Then we have that $s, s' \in S_i$ and $(s, s') \in \mathcal{A}_i^i$. Furthermore, *entities*(s) should include the relevant accompanying documents and *entities*(s') should include the relevant resources. Finally, an expression $e_{s,s'}$ should be determined stating the conditions under which the presence of the accompanying documents will move the procedure towards the allocation of which resources.

2. Risk identification: This step concerns the identification of the risks that may affect the CIs under study and characterization of their potential impact. A risk can be any form of threat or vulnerability from external sources such as terrorist attack, or from internal sources such as malfunction of machinery. For each of the identified risks r_i the following parameters must be defined:

- Probability of occurrence, p_i: this is an estimate of the frequency of occurrence of risk r_i.
- Risk effects: the relationships between each risk and the states of the modeled procedure. In particular, we must specify the states of the CIs that are affected by the occurrence of a risk. We denote by S_i^R the set of states affected by risk r_i and, given a state $s \in S_i^R$, we denote by *effect*(s, r) the expression specifying how the entities of state s are affected by risk r.
- Impact duration, d_i: the time during which the risk will be in effect.

Following risk identification and characterization of risk effects, the next step is to consider and determine the relationships between risks. Specifically, this phase concerns the identification of risk interdependencies, that is, cases where the activation of a risk may trigger the activation of another risk. For example, a human error risk activation on the crane may also activate the risk of machinery malfunction because of damage to the machinery as caused by the first risk. The effects, probability of occurrence, and duration of the impact of an interdependency between risks must also be carefully defined as above.

3. Identify cascading effects: Following the modeling of the CIs under study and the identification of risks that may affect them, the next step concerns the identification of cascading effects between the various CIs. Note that we have a cascading effect when a disruption/event in one infrastructure has an effect on a component in another infrastructure. In other words, this phase is concerned with the identification of interdependencies between different CIs and characterization of their effects. The output of this phase is (1) a set of pairs (s, s') where s is a state of some infrastructure i and s' is a state of some distinct infrastructure j where a cascading effect exists between the two states and (2) an expression characterizing the effect between the two states for each such pair: given a characterization of the risk-based event in the first state, the expression should be defined so as to characterize the triggering of a risk-based event in the second state. This effect can be modeled depending on the situation as a conditional probability or a fixed event.

For example, consider a port's supply chain, and suppose that there is an electricity disruption in the electrical infrastructure. This may interrupt computer-based activities, such as processing the required forms at the port's customs service, yielding a delay of loading the cargo to a truck and hence a delay of the cargo delivery to its final destination. To model this cascading effect we consider the pair (s, s') where s is the state of failure in the electrical infrastructure and s' is the state of document-processing in the port infrastructure, and the associated expression capturing the effect of this coupling can be modeled either as a conditional probability or as a fixed delay: a conditional probability would capture the probability of additional delay inflicted in the case of the electricity disruption, and, in the case of a fixed delay, the specific amount will be specified and applied to the procedures in place. Similar effects can be considered for the delays in loading the truck or having the cargo reach its final destination.

4. Model construction as Petri net: In this step the components that were previously compiled are modeled as a Timed Colored Petri net. We describe how the components of a $TCPN = (P, T, A, \Sigma, V, C, G, E, I)$ are defined following the above analysis.

Beginning with the set of places P of the Petri net, we associate a place with each state $s \in S_i$ for each infrastructure $i \in Inf$, and each risk $r \in R$. Thus, the set P of places is defined as $P = \bigcup_{i \in Inf} S_i \cup R$. Secondly, we must define the set of transitions T and the set of arcs A. These sets correspond to the procedure flows/activities enabled from the various states. In particular, we include a transition and two associated arcs to connect the places corresponding to any two states s, s', with $(s, s') \in \mathcal{A}_i^j$ for some i and j. Here we note that while each CI under study will correspond to a Petri net component describing internal procedures and activities, different infrastructures will also be connected by transitions whenever a state within one infrastructure requires input from another infrastructure.

While these interconnected components constitute the backbone of the Petri net model, another component to be superimposed upon this structure corresponds to the risks identified in Step 3: we include a transition and two associated arcs between the place of each risk r and the states it affects as defined by $\mathcal{E}_r^R$. Finally, we add to the structure a transition and arcs between the associated places between two states related by a cascading effect. In this manner the components T and A of the TCPN are determined.

Moving on to the fourth component pertaining to the color sets to be employed, one must define all the color-types of the tokens/variables that will be used, thus constructing component Σ. Depending on the nature of an entity, its color can have different fields corresponding to the characteristics of interest of the entity as identified in Step 1. For example, a container in a port supply chain can be associated with a color with different fields corresponding to its arrival time, the damage it sustained during transport, its degree of fragility or the presence of accompanying documents.

Next, we must define the set of variables V. Specifically, we employ one variable for each entity defined in the sets $\mathcal{E}_i$ as identified in the first step of the methodology. A variable is associated with an appropriate color set. Similarly, each place is associated with a color set, as required in component C of a TCPN, by merging together the color sets of the entities associated with the state modeled by the place.

Moving on to components G, E of a TCPN, these two functions are used to associate transitions and arcs with expressions as specified by expressions $e_{s,s'}$ and $\mathit{effect}(s, r)$ which determine the conditions and the effects of transitions between places taking place as well as the precondition and the postcondition of the transitions. Here we note that these expressions may associate probabilities of firing transitions (e.g. risk occurrence probabilities will be associated on transitions manifesting a risk) and timed variables will be added in order to measure time passage on transition availability (e.g. duration of the impact of a risk, or duration of a certain activity).

Finally, component I is used to initialize the system by associating each place of the composed Petri net with initial conditions (initial number of tokens in each place and values for all the fields of each token) depending on the scenarios one wants to check. Alternatively, one may define input transitions on places that continuously produce tokens. These transitions can trigger the initial state of the risk chain and they can be programmed to produce tokens according to a function that may depend on time and/or according to a probability distribution. Figure 2 illustrates the skeleton of a Petri net constructed in this step of the methodology. Place *Start* and transition T_1 illustrate the presence of an initial

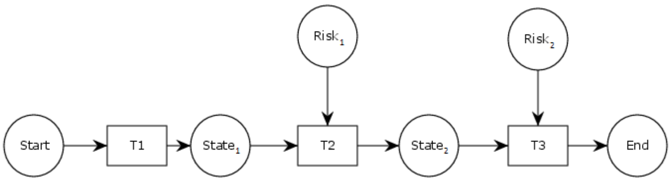

Fig. 2. A representation of the relationships between activities of a CI and risks.

transition that produces an initial set of tokens as input to the first state of the net, namely $Start_1$, and then further tokens according to its programming.

Once the TCPN is constructed, it can be implemented within one of the various available tools for analysis of timed colored Petri net models.

5. Model evaluation: After the construction of the TCPN model and its implementation within an associated tool, scenarios for analysis must be prepared. These scenarios could include evaluation of the effects of a critical initiating event, evaluation of the effect of interdependencies between risks, or of cascading effects between infrastructures. It is also possible to experiment with different security measures and evaluate the gains and tradeoffs between different approaches.

The above can be carried out by taking into account the different modes of analysis offered by TCPN tools which include simulation and model checking. Using simulation, the user can perform a suite of experiments and reach conclusions based on the resulting executions of the system. In turn, model checking is a powerful technology that performs an analysis of the complete state-space given rise to by the system as opposed to single runs. Using this approach the user must express properties of interest in a suitable language, a *temporal logic*, where a property could be of the form of reachability (is a certain state reachable from the initial state of the system?), safety (a certain state is never reachable from the initial state of the system), liveness (all executions will reach a desirable state), probabilistic properties (the probability that more than 5% of containers are lost is less than 2%), temporal properties (no more than 3% of the containers arrive with a delay greater than 2 times units).

After the simulation or model checking or both, the results must be analyzed. In particular, the acceptable margins for various metrics related to the efficiency of the procedure must be defined. For example, the port operator would accept only 0.001% of containers to be lost and only a maximum average delay of one day. During this phase, the "weak" parts of the procedure can be identified, therefore it is possible to select or develop security hardening measures, or modify the procedure to be more efficient. Afterwards, analysis can be performed to the improved model to see if the new metrics are now between the accepted margins. This step can be repeated until the results are satisfying. Additionally, evaluation may explore which risks may cause cascading effects on which parts of the procedure and modify the procedure in order to alleviate these effects.

Discussion: We conclude that Petri nets offer a promising approach towards the risk modeling and analysis of interdependencies of critical infrastructures. The graphical part of the language enables a very quick and intuitive representation of the states and activities as well as the interdependencies existing within infrastructures, and to associate risks with parts of a procedure. Simultaneously, the programming language accompanying the graphical language enables the enunciation of complex behaviors pertaining to activities or the effects of a risk. The presence of probabilistic and temporal constructs is particularly useful for specifying relevant aspects of both risks and activities. Moreover, the approach is both modular and scalable: it is easy to add extra flows and connections, or to introduce additional partially independent systems related to the procedure or work at a finer level of abstraction, by implementing nets within nets.

Furthermore, the methodology is supported by any TCPN tool, hence one can select among a number of tools, some of which are freely available. Finally, the methodology is general and not application-specific, and it can be used to model and evaluate any type of critical infrastructure/risks/dependencies. As such, it is very flexible and one can "flex" it to their needs.

4 Case Study

In this section we demonstrate the applicability of our proposed methodology on a simple port supply chain procedure. This procedure is extracted from a real-world scenario, simplified to serve as a demonstrating example while showing the broad spectrum and capabilities of the proposed methodology: it can easily be extended to cover more complicated procedures. We first present the scenario, then we show how we can model it as a TCPN and apply our methodology to obtain, by simulation, some simple statistics.

Scenario: We consider the following scenario. Suppose that a truck-ship docks in the harbor. Afterwards a crane moves the cargo from the truck-ship to a truck and then the truck moves the cargo to a warehouse. For this scenario we consider the following five risks (the probability values are indicative and used for the purpose of our example):

Risk Number	Risk Name	Risk Effect	Activation Probability
R1	Work Stoppage	Delay	5%
R2	Untrained Staff	Damage or Lost	2%
R3	Mishandled Operation - Human Error	Damage or Lost	5%
R4	Heavy Weather Conditions	Damage or Lost	3%
R5	Terrorist Attack	Damage or Lost	0.5%

Table 1. Risks of present scenario

Furthermore, the activation probability of each risk is the same in all phases and the cargo has a possibility of 85% to be destroyed and of 15% to be lost. Also, the occurrence of risk R2 doubles the activation probability of R3.

Modeling: We first identify the four states of the procedure: *truck-ship*, *crane*, *truck* and *warehouse*. Then we establish the risks, namely their relations with the procedure's states as well as their interconnections. In this scenario, for the interest of space, we have not considered cascading effects; however these can be added, for example as conditional probabilities, as explained in Section 3.

At this point we have collected all the necessary information, thus we are ready to construct our TCPN. We depict a graphical view of the TCPN derived using the CPN Tool of [9], which we also use to run our simulations, in Fig. 3. As

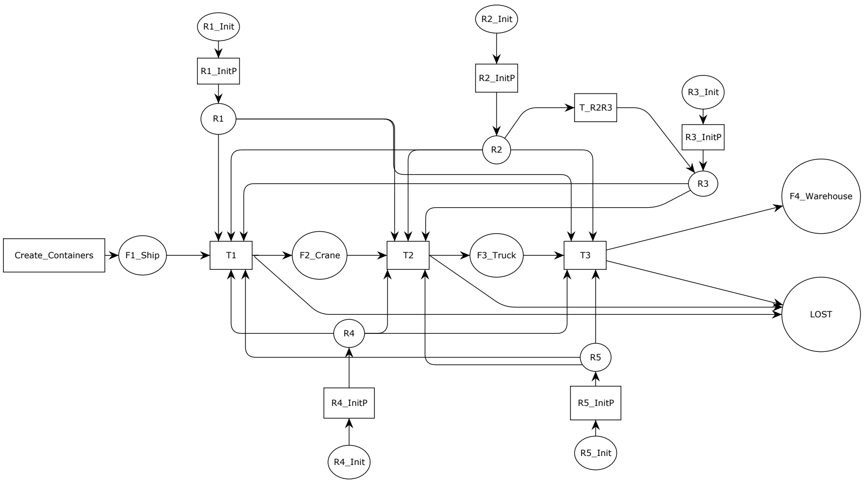

Fig. 3. The Model of the case study in CPN Tools

illustrated in the figure there are fifteen places in total. Specifically, four places represent the states of the modeled procedure and five places represent the risks. In addition, for implementation purposes (as required by the CPN tool) we have created five places for initializing risks and a place in which the lost cargo ends up (terminal case beyond the regular one – warehouse). Hence, component P of the constructed TCPN is the set consisting of these 15 places.

In addition, our TCPN model has ten transitions, where three of which model procedure flow, five deal with the risk initialization flow, one specifies the interdependency between the risks $R2$ and $R3$ and the last one generates the containers. Hence T consists of these 10 transitions.

The depicted arcs represent the flow between places and transitions, e.g., the arc from place *F1_Ship* to transition $T1$ as well as the interdependencies among risks, e.g., the arcs between place $R2$ to transition *T_R2R3* and then to place $R3$. Thus $A = \{\text{all depicted arcs}\}$.

We now define the color-sets used. Except from the standard color-sets, like int and bool (integers and boolean types), it was necessary to define two scenario-specific color-sets: one for the representation of the cargo, *containers*, and one for the representation of the risks, *risks*. The *containers* color-set consists of four fields: *ContainerNumber*, *Status* : $\{Destroyed, Lost, Arrived, Delay\}$, *DelayTime* and *LastReachablePhase*. In turn, the *risks* color-set consists of two fields: *RiskNumber* and *RiskActivationStatus*. A color-set must be assigned to each of these fields; for instance, *ContainerNumber* and *RiskNumber* are of integer color-set. In addition, the *containers* color-set is *timed* because we want to associate containers with delays, while the *risks* color-set is untimed because in our model risks are timeless. These color-sets compose set Σ.

We use different variables in order to represent the different entities, such as cargo and risks. In our case, we assume that each cargo consists of several containers which we associate with individual variables of the *containers* color-set. Similarly, for each risk we declare individual variables of the *risks* color-set. These typed-variables compose set V. Hereafter, we will refer to these variables as *tokens*. Then we assign a color-set to each place based on the entities available to each state. For example, the *containers* color-set is assigned to place *F1_Ship* and the *risks* color-set is assigned to place $R1$. Thus, we compose function C.

Next, we assign a guard function for each transition. In general, transitions take as input tokens representing containers and/or risks. Flow transitions check which risks are activated and then calculate and apply the corresponding effects. Risk transitions activate the relevant risks. These guard functions compose function G. Furthermore, at each arc we assign an arc expression function to control the flow of tokens. These functions compose function E. Finally, we assign an initialization expression to the initial risk places such as $R1$*_Init* and transition *Create_Containers*. In particular, we define the initial tokens of these places and transition. Thus, we compose function I.

Simulation Results: We have run simulations to obtain some simple statistics with respect to the impact of interdependencies on the cargo successfully reaching the warehouse. For this purpose, we have run simulations with an interdependency (between risks $R2$ and $R3$) and without the interdependency (we removed the transition between $R2$ and $R3$); we run each test 6 times obtaining the averages shown in Table 2.

Container Status	Average With Interdependencies	Average Without Interdependencies
Arrived	76,17%	80,17%
Damaged	10,83%	6,17%
Lost	1,67%	0,33%
Delay	11,33%	13,33%

Table 2. Average Results

The derived results are rather expected: the average number of damaged and lost containers are reduced in the absence of the interdependency. Although the simulated scenario is simple, we believe that it sufficiently demonstrates the usability and the potential of our proposed TCPN-based methodology.

5 Conclusions

We have proposed a novel risk assessment methodology based on Timed Colored Petri Nets (TCPNs) for modeling and analyzing Critical Infrastructures with interdependencies, time-critical events and cascading effects. Ongoing work is focusing on applying our methodology on complex real-life scenarios on sea ports and their supply chain. Our analysis, besides simulations will also take advantage of the verification and model-checking capabilities of TCPNs.

References

1. D. Aloini, R. Dulmin, and V. Mininno. Modelling and assessing ERP project risks: A Petri net approach. *European J. of Operational Research*, 220(2):484–495, 2012.

2. S. Bernardi, J. Campos, and J. Merseguer. Timing-failure risk assessment of UML design using time Petri net bound techniques. *IEEE Transactions on Industrial Informatics,* 7(1):90–104, 2011.
3. E. Council. Council directive 2008/114/EC on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection. *Official Journal L345*, 3:0075-0082, 2008.
4. M. Diaz. *Petri nets: fundamental models, verification and applications.* John Wiley & Sons, 2013.
5. G. Giannopoulos, R. Filippini, and M. Schimmer. Risk assessment methodologies for critical infrastructure protection, part I: A state of the art, JRC Technical Notes, 2012.
6. Y. Haimes, J. Santos, K. Crowther, M. Henry, C. Lian, and Z. Yan. Risk analysis in interdependent infrastructures. *Critical infrastructure protection*, pages 297–310, 2007.
7. M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. Snoopy a unifying Petri net tool. In *Application and Theory of Petri Nets*, LNCS volume 7347, pages 398–407, Springer, 2012.
8. K. Jensen and L. M. Kristensen. *Coloured Petri nets: modelling and validation of concurrent systems.* Springer, 2009.
9. K. Jensen, L. M. Kristensen, and L. Wells. Coloured Petri nets and CPN tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
10. M. Klaver, H. Luiijf, A. Nieuwenhuijs, F. Cavenne, A. Ulisse, and G. Bridegeman. European risk assessment methodology for critical infrastructures. In INFRA 2008, pages 1–5, 2008.
11. P. Kotzanikolaou, M. Theoharidou, and D. Gritzalis. Assessing *n*-order dependencies between critical infrastructures. *International Journal of Critical Infrastructures*, 9(1):93–110, 2013.
12. P. Kotzanikolaou, M. Theoharidou, and D. Gritzalis. Interdependencies between critical infrastructures: Analyzing the risk of cascading effects. In *Critical Information Infrastructure Security*, pages 104–115. Springer, 2013.
13. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
14. U. D. of Homeland Security. Nipp supplemental tool: Executing a critical infrastructure risk management approach, 2013.
15. R. Renda. Protecting critical infrastructure in the EU. *CEPS Task Force*, 2010.
16. S. Rinaldi, J. Peerenboom, and T. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems*, 21(6):11–25, 2001.
17. T. Rossi and M. Pero. A formal method for analysing and assessing operational risk in supply chains. *International J. of Operational Research*, 13(1):90–109, 2012.
18. M. Theoharidou, P. Kotzanikolaou, and D. Gritzalis. Risk assessment methodology for interdependent critical infrastructures. *International Journal of Risk Assessment and Management*, 15(2):128–148, 2011.
19. M. Van Eeten, A. Nieuwenhuijs, E. Luiijf, M. Klaver, and E. Cruz. The state and the threat of cascading failure across critical infrastructures: the implications of empirical evidence from media incident reports. *Public Administration*, 89(2):381–400, 2011.
20. D. Vernez, D. Buchs, and G. Pierrehumbert. Perspectives in the use of coloured Petri nets for risk analysis and accident modelling. *Safety Science*, 41(5):445–463, 2003.