Discrete event system formal approaches contribution onto global reliability Markov chain generation

Changyi Xu, Eric Niel, Nicolae Brinzei

▶ **To cite this version:**

Changyi Xu, Eric Niel, Nicolae Brinzei. Discrete event system formal approaches contribution onto global reliability Markov chain generation. 21e Congrès de maîtrise des risques et de sûrete de fonctionnement, $\lambda\mu$21, Oct 2018, Reims, France. hal-02064997

HAL Id: hal-02064997
https://hal.science/hal-02064997

Submitted on 12 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrete Event System Formal Approaches Contribution onto Global Reliability Markov Chain Generation

Contribution des approches formelles des systèmes à événements discrets pour la génération des modèles markoviens de fiabilité

Changyi XU, Eric NIEL*
INSA de Lyon, Laboratoire Ampère UMR 5005
21 avenue Jean Capelle, 69621, Villeurbanne
{changyi.xu ;eric.niel@insa-lyon.fr

Nicolae BRINZEI
Université de Lorraine ,CRAN, UMR 7039
{nicolae.brinzei@univ-lorraine.fr}

Résumé :
Cet article présente une méthodologie visant à générer un modèle de fiabilité d'une structure de service dans le contexte du Model Based Safety Assessment (MBSA) reposant sur les approches formelles des systèmes à événements discrets (SED). La génération s'opère en 3 étapes : modélisation globale du dysfonctionnel, intégration des contraintes de de réparation locales, identification de la panne totale (TBS), réduction du modèle (GRA) pour exploitation quantitative. Les apports majeurs relèvent des liens formels des modèles d'états des composants à leurs spécifications et à la structure fonctionnelle du système. La panne totale s'exprime par les techniques conventionnelles de la SdF telles que; les diagramme de fiabilité (DdF) ou arbre de fautes (AdF).

Summary
This research aims to synthesize a global reliability model, by applying Automata Theory to Model Based Safety Assessment (MBSA). Failure logic expression issued from conventional Reliability Block Diagram or Fault tree Analysis will identify specific Total breakdown States (TBS) and express requirement restricting the Global Faulty Automata (GFA) onto Global Reliability Automata (GRA) establishment thanks to formal composition. The main contributions are relevant to automata translation, requirement expression, TBS identification and TBS treatment.

Background

Model Based Safety Assessment (MBSA) is an inherent task in most safety-critical engineering projects, it is a guarantee for the safe-design. Appreciating the effectiveness of a system through its faulty behavior model, by considering both the operating properties and system architecture, is a common basis for MBSA [1]. Conventionally the establishment of MBSA requires professional skills for engineers and contains four major steps: system modeling, simulation for validation, fault logic expression integration, performances assessment [2].
Linked to Model Based System Engineering (MBSE) for which developments are objects of interconnected models MBSA considers various criteria conform to operating capacity and causality [3]. Fault propagation relies to functional and operational structure are then totally taken into consideration. The location arrangement of the local components establishes logic connections (basic structure as parallel and series operating structure). Each faulty behavior of one local component directly influence the global behavior of the whole system through that connection. So the global system behavior results of diversity influenced both by the local component performance and system architecture.
Moreover, the whole system structure can be appreciated into series, parallel or more complex operating relationship. Indeed, related to granularity each local component operates in collaboration following an admissible principle. Also related to those logic connections, when several local components break leading to the total failure of the whole system. This principle defines the fault logic expression of this faulty system. So analyzing the system architecture to fulfil the system logic expression and then to establish a model strictly satisfying all the logic expression belongs to MBSA approach[4]. The proposal aims to produce a formal as well as automatic oversimplified approach coherent to MBSA with application to Markov chain reliability model generation.

Research Introduction

Thanks to Reliability Block Diagram (RBD)[5] and Fault tree Analysis (FtA)[6], the capacity and complexity of the faulty system can be analyzed and the Boolean Value fault logic expression formula can be elaborated, in which several Boolean Value semantics are connected by conjunction and disjunction logic symbols.
Markov chain model is used for representing the behavior of local components (operating state, failure state, repair transform, failure transform, etc). Each local Markov chain can be translated into event driven local automata for the purpose of composition and satisfying deterministic requirement for dependability analysis [8]. A set of Boolean Value state properties is established consistent with the local component behaviors and a formal associate function is established for the automata states and these properties[9].

GFA is achieved as the automata composition combining faulty behavior. Via the automata composition, the local Boolean Value state properties are generated to be the composed state properties Boolean value in GFA, which are the representatives of the combining faulty behavior [8].
A judgment calculating operation is used to check each Boolean Value property of each state in GFA whether to be a sufficient condition of the fault logic expression formula (the judgment operation can be performed by tools, for example Reduced Ordered Binary Decision Diagram)[10], if the result is true, it implies that this state belong to TBS. The TBS identification contribution is achieved by this judgment operation applied to all the states in GFA.
System operation requirements expression are issued by the synchronization of GFA and event trajectory specifications (ETS). The function of ETS is to specify the GFA following the operation requirements, and ETS state evolution trajectory is formally planned by several specification automata work function establishment based on the requirements [8].
TBS treatment lies in the blocking operation applied to the identified TBS, automatically prosecuted by the tool

Supremica [12]. After synchronizing GFA and ETS, GRA respecting the system operation requirement is established by the TBS identification and TBS treatment. The global reliability Markov chain model is by a further translation issued from GRA and mean time indicators could be then prosecuted [1].

Figure 1 is the work flow of this research and the main contributions will follow 5 major steps:

1. Transform Markov chains to event driven automata. The Markov chains model of the local faulty components includes physical performance as states (operating state, failure state, etc) and behaviors as transitions (failure event transition, repair event transition, etc). This Markov chain can be translated into local dependability event driven automata. This translation satisfies the deterministic dependability analysis requirement. The state Boolean Value properties is established consistent with physical performance.
2. Cartesian product [11] of automata to produce GFA. In order to establish GFA, the Cartesian product is used for the composition operator of all the local dependability models and state Boolean Value properties is generated via composition.
3. TBS identification. By analyzing the system architecture using RBD and FtA, the system failure logic Boolean Value expression formula is established. Judgement operation of state Boolean Value properties comparing to System Failure Logic Expression Formula is applied for TBS identification [9][10]
4. Operation Requirement expression. Establishing formal work function of ETS according to the operation requirement, the ETS is able to specify event evolution trajectory in GFA to be requirement respecting [8].
5. GRA issuing. After synchronizing ETS and GFA, the TBS treatment is applied to the model and GRA is established with TBS blocking operation.

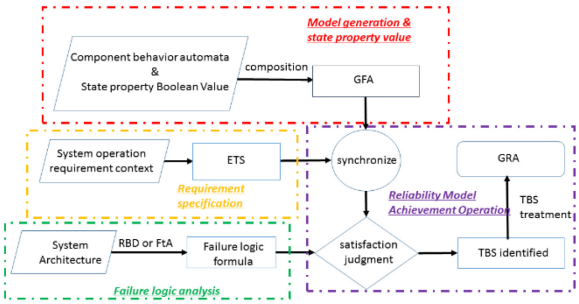

Figure 1. Work Flow

Markov chains and Event Driven Automata Transformation

Shown in Figure 2 left, a Continuous Time Markov Chain representing a fail-repair component with a failure rate λ_{f1}, a repair rate μ_{r1}, initial state S0 and failure state S1. According to Arden's lemma, Continuous Time Markov Chain is able to be transformed as a sequence of unique events. In equation {1}, A and B are the universe of event, Li is one of the sub languages representing a system, A and B are the universe of system unique events. If there is no empty sequence in A, the solution of {1} is unique Li=BA*. If an empty sequence belongs to A, the equation solution is Li=(B+C)A*, there C⊆E* and E is the event set.

$$Li = LiA + B \qquad \{1\}$$

Normally, the Markov chain transition is labelled by an evolution rate and as the Markov chain can also be expressed as a sequence of unique events, so the transition labelled by happening rate and unique event is shown in figure 2 middle: λ_{f1} is presented by failure behavior event f1 with a failure rate λ; μ_{r1} is presented by repair behavior event r1 with a repair rate μ [7]. Moreover, regardless the happening rate, event is established as component behavior, the focus on the event function results the event driven automaton like figure 2 right side.

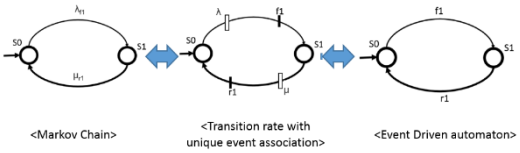

Figure 2. Markov chain and Event Driven automata transform

According to equation {1}, shown in figure 2 <Transition rate with unique event association>, Markov chain model is defined by CTMC=(Qmc; Σmc; δmc; q0), Qmc is the state set{S0,S1}, Σmc is the alphabet which is labelled by two parameters rate{ λ , μ } and the event name{ f1, r1}, qo is the initial state{S0}; δmc is the transition function.

The proof of formalizing CTMC is based on equation {1} solution :

$$\begin{cases} L1 = L2f1 \\ L2 = L1r1 \end{cases}$$

So, L(1) = L1f1r1 and the solution is:

$$\begin{cases} L1 = (f1r1)* \\ L2 = (f1r1)*f1 \end{cases}$$

The alphabet Σmc expalined by the work function is expressed:

$$\begin{cases} \delta mc\,(S0, f1) = S1, \text{by the rate } \lambda \\ \delta mc\,(S1, r1) = S0, \text{by the rate } \mu \end{cases}$$

Event driven automaton transformed from the Continuous Time Markov Chain is defined: G=(Q; E; δ; q0; qm):
Q is the state set: Q= Qmc={ S0, S1} ;
E is the event set: E= { f1,r1 };
Initial state q0= S0;
Marked state qm= S1;
And the work function compared to δmc is expressed:

$$\begin{cases} \delta\,(S0, f1) = S1 \\ \delta\,(S1, r1) = S0 \end{cases}$$

By this transform method, the faulty component model previously delivered by continuous time Markov chains is able to be delivered by associated event driven automata. The purpose of this method is to present transition rate to component behavior marked by visual system event with the ability prepared for model composition, by the reason composition operation of event driven automata offers a formal solution for model generation, moreover after the global faulty automaton model has been established the associated Markov chain is able to be achieved for global system time indictor assessment.

GFA Establishment

1 Local faulty component model establishment

As it is introduced, principle for modeling a component is: performs to be automata states, behaviors to be automata transition events. Figure 3 shows two local faulty components by the help of the transform between continuous time Markov chains and Event driven automata modeling for component G1 and component G2. To consummate the component perform property, Boolean value state property is adopted to improve the state significance description. Boolean value state property is in fact Boolean value symbols of state characteristic, for example when there is a need to denote one state on the failure characteristic, a Boolean value symbol < F > can be created to associate with this state, and if there is a need to

explain one other state with the opposite characteristic of failure (not failure or working well), the negation < ¬ F1 > is used. Furthermore, the combination of different Boolean value state properties with disjunction logic operation (symbol: ⋁) and conjunction logic operation (symbol: ⋀) has the function to express more complex component performs.

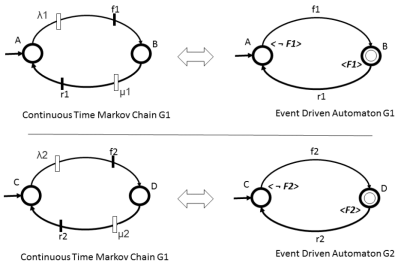

Figure 3. Local faulty model

To be uniformed with automaton definition, the Boolean value state property applied to the automata theory is given:

$$G = (Q; E; \delta; q_0; qm; AP; L) \quad \{2\}$$

Q, E, δ, q0, qm reminded to be the same as defined before: state set, transition set, work function, initial state and marked state. New conceptions are as follow:

AP is the set of Boolean value state property;
L:Q→AP is the associate function corresponding with state and Boolean value state property;

Assume component G1 and component G1 have the behaviors of failure and repair, and performs are working and break. The Local faulty component model: "Boolean value state property associated event driven automaton" is as follow:

Component G1:

$$G1 = (Q1; E1; \delta1; q_01; qm1; AP1; L1)$$

where,
Q1= {A, B}, A is working state, B is break state;
E1= {f1, r1}, f1 is failure event, r1 is repair event;
δ1 is work function;
q_01= {A};
qm1= {B};
AP1= {F1}, F1 denotes the Boolean value state property of "component G1 system failure";
L1 is the associate function and L1 (A)= <¬ F1>, L1 (B)= <F1>;

Component G2:

$$G2 = (Q2; E2; \delta2; q_02; qm2; AP2; L2)$$

where,
Q2= {C, D}, C is working state, D is break state;
E1= {f2, r2}, f2 is failure event, r2 is repair event;
δ1 is work function;
q_02= {C};
qm2= {D};
AP2= {F2}, F2 denotes the Boolean value state property of "component G2 system failure";
L2 is the associate function and L2 (C)= <¬ F2>, L2 (D)= <F2>;

2 Model generation by Cartesian product

Model generation is issued by the local automata composition operation theory to establish a global automaton. Composition operation is in fact the Cartesian product, assume two automata for our components: G1 = (Q1; E1; δ1; q_01; qm1) and G2 = (Q2; E2; δ2; q_02; qm2), we use symbol '//' to denote the automata composition and symbol '×' to denote the Cartesian product, the composition operation of two automata is defined[8][11]:

G3= G1//G2

$$= Ac(Q1\times Q2; E1\cup E2; \delta3; q_01\times q_02; qm1\times qm2) \quad \{3\}$$

Ac stands for taking the accessible part.

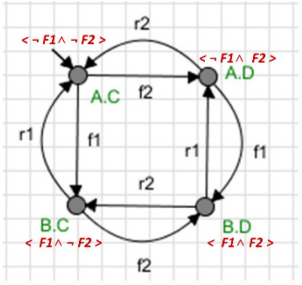

Figure 4. Automata Composition

And the work function δ3 issued by Cartesian product is:

$$\delta3\ ((x,y),e) = \begin{cases} (x', y) & \text{if } e \in E1 \text{ and } \delta1(x,e) = x' \\ (x, y') & \text{if } e \in E2 \text{ and } \delta2(y,e) = y' \end{cases} \quad \{4\}$$

Shown in figure 4, Cartesian product applied to our faulty component G1 and G2, the result of G3= G1//G2 is as follow:

Component G3:

G3= G1//G2

Where,
Q3= Q1×Q2={(A,C) (A,D) (B,C) (B,D)};
E3= E1∪E2={f1,f2,r1,r2};
δ3 is explained as equation{4};
q_03= $q_01\times q_02$={(A,C)};
qm3= qm1×qm2={(B,D)};

3 State property conjunction via model generation

The globally Boolean value state property improvement of equation {3} is by the prerequisites that automata model G1 and G2 are already locally associated with Boolean value state property. Through the Cartesian product of these two automata, the Boolean value state property association in global automaton G3 is linked by logic connection of the associate function of G1 and G2.
Given the Boolean value state property associated automata model G1 = (Q1; E1; δ1; q_01; qm1; AP1; L1) and G2 = (Q2; E2; δ2; q_02; qm2; AP2; L2)

G3= G1//G2
= Ac(Q1×Q2;E1∪E2;δ3; $q_01\times q_02$;qm1×qm2; AP1∪AP2; L1×L2) {5}

The global Boolean value state property set:

$$AP3=AP1\times AP2$$

Means the union of two local Boolean value state property set, as AP1= {F1} and AP2={F2}, then:

$$AP1\cup AP2=\{ F1, F2 \}$$

The associate function in the global automata G1//G2 is in fact the Cartesian product result from the two associate function elements, but what needs to be emphasized is the global Boolean value state property is represented by the connection of two local Boolean value state property, connected by conjunction 'and' logic (symbol: ⋀) .
As L1:Q1→AP1 and L2:Q2→AP2, then L3 is:

$$L3=L1\times L2: Q1\times Q2\rightarrow AP1\times AP1 \quad \{6\}$$

L3 represented by the logic conjunction operation is:

$$L3(x,y)=L1(x)\wedge L2(y) \quad \{7\}$$

where,

$$x \in Q1;\ y \in Q2;\ (x,y) \in Q1 \times Q2 = Q3$$

In figure 4, locally L1(A)= <¬ F1>, L1(B)= <F1>, L2(C)=< ¬ F2>, L2(D)= <F2>, via the model generation of G1//G2, the global associate function result of G3 is:

$$L3(A,C)=L1(A) \wedge L2(C)= <\neg F1 \wedge \neg F2>$$
$$L3(A,D)=L1(A) \wedge L2(D)= <\neg F1 \wedge F2>$$
$$L3(B,C)=L1(B) \wedge L2(C)= <F1 \wedge \neg F2>$$
$$L3(B,D)=L1(B) \wedge L2(D)= <F1 \wedge F2>$$

The GFA is achieved by the model generation of local faulty automata. Consummating the expression of system performs, Boolean value state property is associated with each local state and via the model generation of local automata, the global states in GFA are also associated with Boolean value state property, this state property not only has a full expression of the GFA system perform but also is the key factor for further TBS identification and TBS treatment.

TBS identification

1 System failure logic expression formula obtaining

Thanks to basic system dependability analysis technologies, Reliability Block Diagram (RBD)[5] and Fault tree Analysis (FtA)[6], system architecture is able to be analyzed. Based on functional dependency, the simplest structure are parallel or series in the application of RBD and FtA. System Failure Logic Expression Formula provides the system Boolean Value failure property in logic connection form will help for the TBS identification aim.

There exists direct translation RBD safe structure between FtA by the relation that parallel structure in RBD is equal to 'and' gate in FtA and series structure in RBD is equal to 'or' gate in FtA. By connecting the local system Boolean Value failure property of the system with combination of "and" gate and "or" gate, from the FtA, we can get the System Failure Logic Expression Formula of this global system with the help of computer calculation software, such as GRIF.

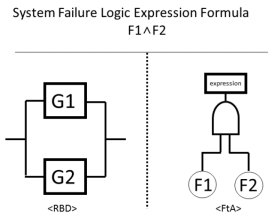

Figure 5. System Failure Logic Expression Formula

System Failure Logic Expression Formula is expressed as the form that a set of Boolean Value failure property connected with the symbol '∨'(logic connection "or") and with the symbol '∧' (logic connection "and"), showing how the local components breakdown contribute to the globally total breakdown of this architecture.

Suppose system architecture of our two components G1 and G2 are working in the hot redundancy form: only if G1 and G2 are both breakdown, the whole system breaks down. In figure 5, as the parallel safe structure in RBD, in FtA there is an 'and' gate connected F1 and F2, here as introduced before F1 and F2 are denoted as Boolean Value failure property of two local components. Based on the calculation principle of FtA: 'and' gate result in a conjunction '∧' of the Boolean Value properties; 'or' gate results in a disjunction '∨' of the Boolean Value properties, the System Failure Logic Expression Formula is expressed:

$$\Psi = F1 \wedge F2$$

Ψ is prepared for the TBS identification reference which is compared to the state Boolean Value property in this approach. And also for the case of operation requirement expression designing specification automata ETS, as the event trajectory ETS will be possible formalized by a sequence of failure events exciting Ψ happening.

2 TBS identification judgment method

System Failure Logic Expression Formula is established through RBD and FtA or a given formula by designer, identification judgment method detail is issued by the compare of the Boolean value state property of each state in GFA with System Failure Logic Expression Formula, the states whose significance satisfys the total breakdown condition will result that its Boolean value state property is a sufficient condition of System Failure Logic Expression Formula and this state is operated to be extracted from the global state set into a TBS state set.

Judgment method mathematical formula exposition is as following:

Given a automaton G = (Q; E; δ; q_0; qm; AP; L)

The Boolean value state property of one state satisfying the System Failure Logic Expression Ψ contributes to the TBS state set Q_{TBS}.

$$Q_{TBS}=\{x|\ x \in Q,\ L(x) \vDash \Psi\}$$

{8}

Here, the satisfaction symbol '⊨' is applied to denote L(x) is a sufficient condition of Ψ and Q_{TBS} is a subset of Q: $Q_{TBS} \subseteq Q$.

The Boolean value state property satisfaction judgment is applied in the verification technique of almost automata model checking tools. In our approach, a brief judgment calculation tool named BDDs (Reduced Ordered Binary Decision Diagrams) is applied for the TBS identification of GFA.

In BDDs, the state property is coded as a combination of Boolean value property elements, the identification formula is coded as disjunction or conjunction of system Boolean value property elements like formula Ψ, where state property value and identification formula are both consistent as what they are defined. The sufficient condition judgment satisfaction operation is implemented by the BDDs code order operation 'imp' which means 'imply'. Submitting BDDs code, if the result only show an assignment 'T', this means the state property value is a sufficient condition of the identification formula, which indicates that this state is identified to be TBS and this state should be extracted into Q_{TBS}. If BDDs result shows not 'T', this state is not a satisfying state (The result will shows a diagram telling how Boolean Value develops will be 'T' or how it will be 'F', 'F' meaning false, this result sort is not our concerning knowledge) [10].

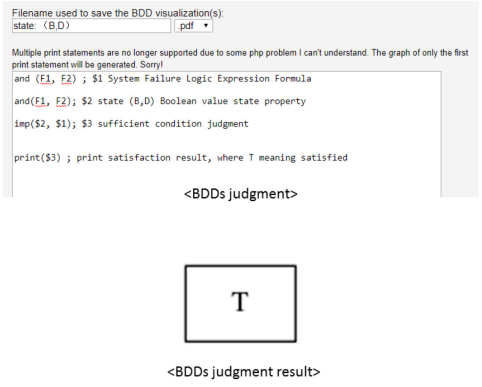

Figure 6. state (B,D) from G3 BDDs method

State (B,D) from GFA G3 is treated by BDDs method, as it is known for the Boolean value state property: L3(B,D)=L1(B)∧L2(D)= F1∧F2 is shown in figure 6 top part BDDs coding $2, System Failure Logic Expression Formula Ψ=F1∧F2 and it is expressed as $1. $3 is the judgment operation for whether L3(B,D) is a sufficient condition of Ψ. Based on equation {8},the judgment result is 'T' and this indicates that (B,D)∈ Q_{TBS} to be a total break state TBS.
Same treatment applied to (A,C), (A,D), (B,C), the BDDs results indicate they are not TBS. So TBS identification judgment method applied to GFA is:

$$Q3_{TBS}=\{ (B,D) \}$$

Operation Requirement Expression

Normally the system operation is required to respect several principles, thus an operation requirement context is established to instruct the system operators. And there are various sorts of operation requirements, for example: when there happens a series of breakdown accidents onto the local components, the repair action should be prosecuted, then a problem of repair action before and after order is available, for one certain case "One important component has the priority to be first repaired" and other rule that "FIFO: the component firstly breaks down and then following another component breaks down secondly, requirement context rules that respect the time order first breaks first repaired". Operation Requirement Expression means GFA behaviors should respect the operation requirement context, which is issued by several formal specification automata (ETS) synchronizing with GFA, specifying GFA transition evolutions.
For the problem that requirement context various as different cases various, it is not possible to establish all the ETS in the world, but a formal work function of ETS respecting a sequence events happing can be established. Two cases are used to explore our approach.

"One important component has the priority to be first repaired". Assume that our faulty component G1 and G2, component G1 should be first repaired whenever G1 breaks down, so this requirement context should be established in ETS by a sequence of failure event (denoted by SFE) happening:

SFE1=" f1 "

This means that if f1 happens, the operation requirement is excited to be effective.
Given the definition ETS1= (Q_{SP1}; E_{SP1}; δ_{SP1}; q_{0SP1}; q_{mSP1}),
Q_{SP1} is the state set {S0, S1};
E_{SP1} is the event set E_{SP1}=E1∪E2={f1,f2,r1,r2};
q_{0SP1} is initial state S0;
q_{mSP1} is marked state S1, the transition sequence route from initial state to marked state is SFE1=" f1 ";

Then based on SFE1 the work function δ_{SP1} is defined:
Evolution function:

$$\delta_{SP1}(So,f1)=S1$$
$$\delta_{SP1}(S1,r1)=S0$$

Self-loop function:

$$\delta_{SP1}(S0, E_{SP1}\backslash f1)=S0$$
$$\delta_{SP1}(S1, E_{SP1}\backslash r2)=S1$$

here, '\' denotes for complement set operation.

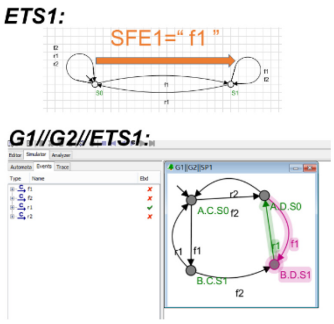

Figure 7. specification ETS1

Shown in Figure 7, the specification ETS1 is designed like: from the initial S0 state after SFE1 S1 is reachable and requirement context is effective at S1, where S1 has the self-loop {f1,f2} with meaning only r1 is allowed (r2 is forbidden here), the self-loop of S0 is {r2, r1, f2} is of the meaning all the events except SFE1="f1" happening makes no requirement context excited at S0. By SP1 whenever f1 happens, r1 is only allowed, which is equal to "component G1 should be first repaired whenever G1 breaks down". Moreover the simulation result proof our method of the result "whenever f1 and f2 have both happened, r1 is priority to be excited than r2"

FIFO: Assume that our faulty component G1 and G2, G1 first breaks down and then following G2 breaks down, G1 is of priority to repair. So this requirement context should be established in ETS by a sequence of failure event (denoted by SFE) happening:

SFE2=" f1 f2 "

This means that f1 happens and then f2 happens, the operation requirement is excited to be effective.
Given the definition ETS2= (Q_{SP2}; E_{SP2}; δ_{SP2}; q_{0SP2}; q_{mSP2})
Q_{SP2}={S0,S1,S2};
E_{SP1}=E1∪E2={f1,f2,r1,r2};
q_{0SP2}=S0;
q_{mSP2}=S2;
Then based on SFE2 the work function δ_{SP2} is defined:
Evolution function:

$$\delta_{SP2}(So,f1)=S1$$
$$\delta_{SP2}(S1,r1)=S0$$
$$\delta_{SP2}(S1,f2)=S2$$
$$\delta_{SP2}(S2,r1)=S0$$

Self-loop function:

$$\delta_{SP2}(S0, E_{SP2}\backslash f1)=S0$$
$$\delta_{SP2}(S1, E_{SP1}\backslash r1\backslash f2)=S1$$
$$\delta_{SP2}(S2, E_{SP1}\backslash r1\backslash r2)=S2$$

here, '\' denotes for complement set operation.

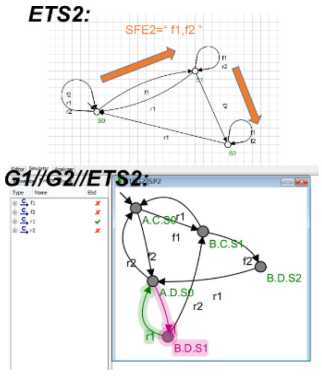

Figure 8. specification ETS2

Shown in Figure 8, the specification ETS2 is designed like: from the initial S0 state after SFE2 S2 is reachable and requirement context is effective. By the self-loop designing of S2 the only repair event allowed is r1, which is equal to "G1 first breaks down and then following G2 breaks down, G1 is of priority to repair" and the simulation result shows the correctness in figure 8 lower part.

TBS treatment for GRA establishment

The establishment principle of a reliability model is to convert break down state as an absorbing state thus the TBS treatment lies in the GRA establishment by converting TBS output transitions deleted, operated on GFA. Thanks to a new application of automata tool Supremica, after TBS is identified, the TBS treatment is implemented automatically and formally.

According to equation {8}, with the hot redundancy working form system architecture，the TBS set is

$$Q3_{TBS}=\{ (B,D) \}$$

Assume that the FIFO operation requirement: "G1 first breaks down and then following G2 breaks down, G1 is of priority to repair" is respected, so ETS2 shown in figure 8 is used for the final GRA establishment.

Based on the work flow in figure 1, we synchronize GFA with ETS2: G3//ETS2=G1//G2//ETS2 shown in figure 8. Up to this step, our GFA model is already applied by Operation Requirement Expression.

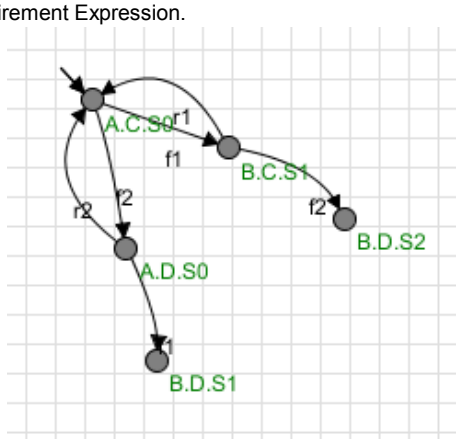

Figure 9. GRA establishment

Continuing operating on GFA, normally the TBS identification is issued by the judgment of state Boolean value property with System Failure Logic Expression Formula: Ψ=F1∧F2 of the GFA shown in figure 8 lower part like what it is introduced in the TBS identification part. But considering ETS2 is only established by event trajectory specifying without any new Boolean value property added into it (in ETS2, S0 and S1 have no association with state Boolean value property), there is no local state Boolean value property generation via the composition of G3 and ETS2. So to be brief, the automaton state Boolean value property reminded to be the same before or after specification automata composition, state (B,D,S1) and state (B,D,S2) share the same state Boolean value property:

$$L(B,D,S1)= L(B,D,S2)= L3(B,D)=<F1 \wedge F2>.$$

So according to $Q3_{TBS}$= {(B,D)}, in the GFA automaton G1//G2//ETS2

$$Q_{TBS}= \{ (B,D,S1), (B,D,S2) \}$$

Absorbing operation TBS treatment applied to Q_{TBS}, the result is represented in figure 9. The further work is to transform this result back to Markov chain model, with the help of event and rate association relationship in 'Markov chains and Event Driven Automata Transformation' part, regardless of the event and focus on the transition rate to establish Global Markov chain, thus the model will help to the system assessment time indicator calculation with a application in the MBSA field.

Conclusion

The related contributions belongs to MBSA by combining requirement and faulty structures. The major steps are local behavior model, requirement expression, TBS identification and TBS absorbing. They all satisfy the incompatibility properties for dependability analysis. By the help of system assessment tools, such as GRIF, the mean time indicators assessment is possible. Furthermore, all models can be simulated and are able to validate the design steps.

References

[1]NIEL, Eric et CRAYE, Etienne. Maitrise des risques et surete de fonctionnement des systemes de production. Productique: information,commande, communication. Lavoisier, 2002.

[2]JOSHI, Anjali, HEIMDAHL, Mats PE, MILLER, Steven P., et al.Model-based safety analysis. 2006.

[3]WYMORE, A. Wayne. Model-based systems engineering. CRC press, 1993.

[4]FORD JR, Lester R. Network flow theory. RAND CORP SANTA MONICA CA, 1956.

[5]ČEPIN, Marko. Reliability block diagram. In: Assessment of Power System Reliability. Springer, London, 2011. p. 119-123.

[6]ERICSON, Clifton A. Fault tree analysis. In: System Safety Conference, Orlando, Florida. 1999. p. 1-9.

[7]IONESCU, Dorina-Romina. Evaluation quantitative de séquences d'événements en sûreté de fonctionnement à l'aide de la théorie des langages probabilistes. 2016. Thèse de doctorat. Université de Lorraine (Nancy).

[8]CASSANDRAS, Christos G. et LAFORTUNE, Stephane. Introduction to discrete event systems. Springer Science & Business Media, 2009.

[9] BAIER, Christel et KATOEN, Joost-Pieter. Principles of model checking. MIT press, 2008.

[10]WEAVER, Sean, FRANCO, John, et SCHLIPF, John. Extending existential quantification in conjunctions of BDDs. Journal on Satisfiability, Boolean Modeling and Computation, 2006, vol. 1, p. 89-110.

[11]EILENBERG, Samuel. Automata, languages, and machines. Academic press, 1974.

[12]FABIAN, Martin, FLORDAL, Hugo, et al. Supremica--An integrated environment for verification, synthesis and simulation of discrete event systems. 2016