A szolgáltatásbiztonság analízise

Előadásvázlat „Szolgáltatásbiztonságra tervezés” tárgyból

Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

Tartalomjegyzék:

1 Bevezetés

Az analízis célja: A komponens szintű hibák (hibaállapotok, események) és a rendszerszintű hibajelenség kapcsolatának kiderítése:

- Mihez vezet rendszerszinten egy vagy több komponens hibája?
- Mik lehetnek egy rendszerszintű hibajelenség okai?
- Milyen valószínűségi és időadatok jellemzik a rendszerszintű hibajelenséget, ha adottak a komponensek hibavalószínűségei, illetve meghibásodási tényezői?

Az analízis által vizsgált illetve használt jellemzők:

- Valószínűségi időfüggvények: megbízhatóság, rendelkezésre állás, biztonságosság
- Aszimptotikus valószínűség: Készenlét
- Várható értékek: MTFF, MTTF (MUT), MTTR (MDT), MTBF
- Paraméterek: meghibásodási tényező, hibaterjedési valószínűség

Az analízis típusai ok-okozati szempontból:

- *Előrelépő* (induktív) keresés: Kiindulási hiba hatásainak továbbkövetése a rendszerben; időbeli és oksági továbblépés.
- *Visszalépő* (deduktív) keresés: A rendszerszintű hiba okainak visszavezetése komponens hibákra.

Az analízis típusai a rendszer struktúrája szempontjából:

- *Alulról felfelé* (bottom-up) történő keresés: Alrendszerek (komponensek) felől a teljes rendszer felé.
- *Felülről lefelé* (top-down) történő keresés: Magasabb szintű absztrakciók finomítása (rendszer felől az alrendszerek felé).

Módszerek:

- Kvalitatív technikák: Szisztematikus módszert adnak az elemzéshez, azonosítják a beavatkozás helyét (pl. egyszeres hibaok, kritikus hibák).
- Kvantitatív technikák: A komponensek paraméterei alapján kiszámítják a rendszerszintű hiba jellemzőit (valószínűségi és időadatok) a komponens szintű jellemzők alapján.

2 Kvalitatív analízis technikák

2.1 Ellenőrző lista (Checklist)

- Tapasztalatok (ökölszabályok) összegyűjtése és alkalmazása/ellenőrzése újabb rendszerek esetén:
    - ismert hibaforrások (tapasztalatok) ki ne maradjanak
    - létező szabványokhoz, megoldásokhoz való illeszkedés biztosított legyen
- Rendszeres frissítés alapkövetelmény
- Probléma: Nagy méretűvé válhat (alkalmazása nehézkes); téves biztonságérzetet adhat.

2.2 Hibafa analízis (Fault Tree Analysis, FTA)

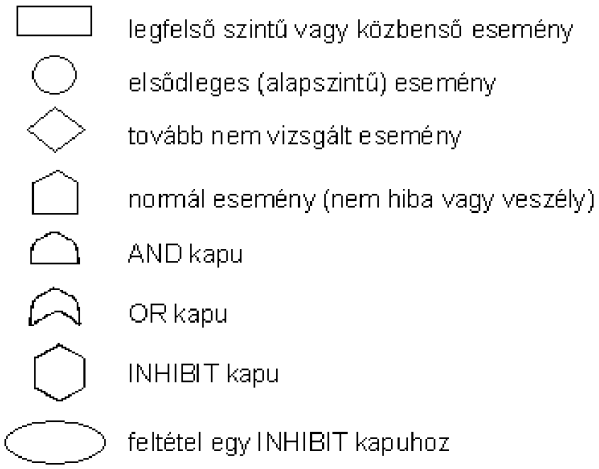
Rendszerszintű hiba okainak vizsgálata (az elektronikai és repülőgépiparból elterjedt technika)

2.2.1 Az elemzés célja

- Legfelső szintű események (hibák) meghatározása (rendszerterv, szolgáltatások alapján)
- Felülről lefelé haladó (egyben deduktív) elemzés:
    - *Legfelső szintű esemény*: azonosított rendszerszintű hibajelenség
    - *Közbenső események*: amelyek a legfelső szintű esemény kialakulásához vezetnek (szükséges vagy elégséges)
    - *Pszeudo-események*: események kombinációi (Boole relációk: AND, OR, NOT)
    - *Elsődleges (vagy alapszintű) események*: nincs további felbontás alacsonyabb szintű eseményekre

2.2.2 Hibafa konstrukció

- Specifikáció (legfelső szintű esemény) – architektúra (közbenső események, pszeudo-esemény) – cserélhető komponensek (alapszintű események) bejárása
- Grafikus reprezentáció: Szabványos szimbólumkészlet (igazságtábla helyett áttekinthetőbb): AND, OR, INHIBIT kapuk
- Hardver esetén: automatizált technikák léteznek a hibafa konstrukcióra (jól megfogható struktúra és hibamódok)
- Szoftver esetén: vezérlési struktúrák vizsgálatán alapul



A hibafa szokásos elemei

2.2.3 Kvalitatív analízis

- Hibafa redukciója (közbenső események feloldása): Diszjunktív normál forma kialakítása; a legfelső szintű esemény kifejezése az alapszintű eseményekkel.
- Vágat: AND kapcsolatban lévő események halmaza a legfelső OR alatt a diszjunktív normál forma szerinti alakban
    - Redukálható vágat: Eseményeinek részhalmaza is vágatot képez

- Minimális vágathalmaz: tovább nem redukálható vágatok
- Azonosítható: rendszerszintű hibához vezető kombinációk
    - Egyszeres hibaok (SPOF): Önmagában is a rendszerszintű hibához vezet (egy vágatot képez)
    - Több vágathalmazban szereplő esemény: kiküszöbölése hatásosan redukálja a hibafát
- Alkalmazási korlátok:
    - Részletes tervezés után használható
    - Csak *statikus* kombinációkat ad meg: nincs sorrendbeli függőség, dinamikus viselkedés leírása, eseményszekvenciák kezelése

Mintapéldák:

- NVP hibafa: SPOF azonosítható (szavazó), közös módusú hiba felveendő.
- RB hibafa: A hibafa nem tükrözi a működési módot, mivel csak egy „snapshot”.
- Hardver struktúra: Minimális vágathalmazok meghatározása.

2.2.4 Kvantitatív analízis

Alapszintű eseményekhez rendelt valószínűségek alapján a rendszerszintű esemény valószínűsége számítható.

A hibafa kapuinak szerepe:

- AND kapu: valószínűségek szorzata:
  $P(A \wedge B) = P(A) \cdot P(B \mid A) = P(B) \cdot P(A \mid B)$,
  itt $P(A \wedge B) = P(A) \cdot P(B)$ ha függetlenek
- OR kapu: valószínűségek összege:
  $P(A \vee B) = P(A) + P(B) - P(A \wedge B) \leq P(A) + P(B)$ felülbecslés szokásos

Probléma: Közös módusú hibák, paraméterértékek forrása (elsődleges események valószínűségei).

2.3 Eseményfa analízis (Event Tree Analysis, ETA)

- Előrelépő analízis elsődleges események következményeinek kiderítésére
    - *kezdeti esemény*: egy komponens hibája/veszélyes helyzete
    - *következő események*: más rendszerkomponensek származtatott vagy rákövetkező hibái
    - sorrendezés: időbeli/oksági viszony, balról jobbra haladva
    - esemény (komponens hiba) bekövetkezése vagy nem bekövetkezése mint elágazás jelenik meg a diagramon.
- Analízis:
    - Egy ”útvonal” valószínűsége: események valószínűségeinek szorzata
    - Használatos: többszintű védelmi rendszerek analízise esetén
- Előnyök:
    - Események sorrendje vizsgálható
    - Hibajelenség (baleset) forgatókönyvek származtathatók

- Korlátok: komplexitás, többszörös események kezelése, sorrendezés kritikus
- Mintapélda: Tartály túlhevülés, RB séma megadása.

2.4 Ok-következmény analízis (Cause-Consequence Analysis, CCA)

- Eseményszekvencia és oksági függések egyidejű ábrázolása
    - *Következmény fa*: egy elsődleges esemény következményei (esemény szekvencia)
    - *Csatolt hibafák*: az egyes eseményekhez tartozó döntés (bekövetkezik vagy nem) indokainak felderítésére
    - Szimbólumok a grafikus reprezentációban: ld. hibafa és eseményfa
- Előnyök: a hibafa és eseményfa előnyeit ötvözi
- Korlátok: minden kezdeti eseményhez külön diagram szükséges

2.5 Veszély és működőképesség analízis (Hazards and Operability Analysis, HAZOP)

- Kémiai folyamatok esetén elterjedt: balesetet az anyagáramlás tervezettől való eltérése okozza; ezeket a lehetséges eltéréseket néhány jellemző kulcsszó alapján át lehet tekinteni:
    - NO, NONE: nincs működés (pl. nincs anyagáramlás)
    - MORE: több eredmény (pl. több anyag áramlik)
    - LESS: kevesebb eredmény (pl. kevesebb anyag)
    - AS WELL AS: más, nem tervezett aktivitás
    - PART OF: csak részben valósul meg
    - REVERSE: az előírtak ellenkezője történik
    - OTHER THAN: egészen más történik, mint az előírt
- Alkalmazás informatikai rendszerekben: az információ áramlásának vizsgálata
    - lehetséges eltérések azonosítása (szisztematikus keresés a folyamatdiagramok alapján)
    - az eltérésekhez tartozó veszélyek azonosítása
    - okok azonosítása, kiküszöbölés
- Előny: a tervezési dokumentáció alapján elvégezhető
- Hátrány: a folyamat szakértőit igényli, munkaigényes

2.6 Hibamód és -hatás analízis (Failure Modes and Effects Analysis, FMEA)

- Alkalmazás:
    - Komponensek, hibamódok felsorolása, valószínűségekkel illetve gyakoriságokkal
    - Más komponensekre illetve a rendszerre gyakorolt hatások azonosítása (előrelépő analízis)

- FMEA táblázat:

Komponens	Hibamód	Hiba valószínűség vagy gyakoriság	Hatás
...	...	...	...

- Előnyök: egyszeres hibapontok (SPOF) felismerhetők
- Hátrányok: hibamódok ismertek kell legyenek, többszörös hibák nem vizsgálhatók
- Kiegészítés: Hibamód, -hatás és kritikusság vizsgálat (Failure Modes, Effects and Criticality Analysis, FMECA)
    - FMEA kiegészítése a hiba kritikusság jellemzésével (rangsorolás kategóriákba)
    - Megelőző/korrigáló akciók szintén felsorolhatók

2.7 Állapot alapú veszély analízis (State Machine Hazard Analysis)

- Állapotgép alapú modell: állapotok, átmenetek, feltételek, trigger események és akciók
- Biztonsági analízis: meghatározni, a rendszer beléphet-e veszélyes állapotba (a veszélyes állapotokat tartalmazó állapotpartícióba)
    - elméleti megoldás: kezdeti állapotból az állapottér felderítése
    - gyakorlati megközelítés: visszafelé keresés a veszélyes állapotból (hogyan kerülhető el)
- Előnyök: automatizálható, tervezési modell használható
- Hátrányok: veszélyes állapot formális specifikációja nehéz, állapottér-robbanás (konkurens rendszerek esetén)

2.8 Emberi hibák analízise (Human Error Analysis)

- Kvalitatív módszerek: az emberi tevékenységek/műveletek esetén megvizsgálni:
    - lehetséges funkcionális hibák (pl. kihagyás, összecserélés) illetve teljesítményhibák (pl. lassú, túl gyors)
    - a hibák (veszélyek) hatása, ezek kritikussága
    - okok elemzése: pl. fizikai és mentális elvárások
    - a hibák elkerülésének módszerei
- Táblázatos forma:

Művelet	Veszély	Hatások	Okok	Elkerülés
...	...	...	...	...

- Kvantitatív technikák:
    - emberi hibákhoz valószínűséget rendelni; befolyásol: pszichológiai hatások (stressz), ember-gép interfész, betanítás, utasítások, függőség más műveletektől
    - mérés: dokumentált környezetben (nehézkes); személyenként eltérő lehet a veszélyhelyzetben való viselkedés

3 Kvantitatív megbízhatósági modellezés és analízis technikák

A formális megbízhatósági modellek készítésének és megoldásának (az állapot alapú, kvantitatív módszereknek) célja: A komponensek meghibásodási jellemzői alapján a rendszerszintű jellemzőket kiszámítani; ezek alapján

- architektúra változatokat összehasonlítani,
- érzékenységvizsgálatot végezni (melyik komponens tulajdonságaira érzékenyek a rendszerszintű jellemzők),
- megbízhatóság szempontjából szűk keresztmetszetet azonosítani (mit érdemes kicserélni).

Jellemzők:

- A komponens szintű paraméterek nehezen mérhetők, sokszor csak becslések vannak.
- Egyszerűsítő hipotézisek kellenek (magas absztrakciós szint, a meghibásodás illetve javítás egyszerűsített modellje).
- Modell validációjára lehet szükség (log elemzés, hibainjektálás).

Ismétlés a szolgáltatásbiztonság jellemzőiről:

- Rendszer állapottér partíciók: H (hibamentes, *U*-val is jelölhető), F (hibás, *D*-vel is jelölhető)
- Rendszerállapot: $s(t)$
- Megbízhatóság: $r(t) = P\{\forall t' < t : s(t') \in H\}$ (folyamatos szolgáltatás)
- Rendelkezésre állás: $a(t) = P\{s(t) \in H\}$ (javítva szolgáltatásra kész)
- Meghibásodási tényező: $\lambda(t) = \lim_{\triangle t \to 0} \frac{1}{\triangle t} P\{s(t + \triangle t) \in F \mid s(t) \in H\}$ (a t időpillanatban következik be a meghibásodás)

  A definíció alapján: $\lambda(t) = -\frac{1}{r(t)} \frac{dr(t)}{dt}$
- Kádgörbe: Középső szakaszán $\lambda(t)$ konstans, így $r(t) = e^{-\lambda t}$ (egyébként $r(t) = e^{-\int_0^t \lambda(t)dt}$)
- Megbízhatóság: Hibamentes állapotban maradás valószínűsége egy komponens esetén, ha λ a meghibásodási tényező: $e^{-\lambda t}$

3.1 Boole modellek

Használat: Egyszerű architektúrák esetén a megbízhatóság számítására (nincsenek függőségek a komponensek között). Alapmodell: Komponensek független hibás vagy hibamentes állapota.

Soros architektúra: Bármely komponens hibája esetén rendszerhiba (OR hibafa).

- k komponens esetén: $r_R(t) = \prod_{i=1}^{k} r_i(t)$

Párhuzamos architektúra: A komponensek egymást helyettesíthetik (tartalékok, AND hibafa).

- $1 - r_R(t) = \prod_{i=1}^{k} (1 - r_i(t))$
- k egyforma komponens esetén: $r_R(t) = 1 - (1 - r(t))^k$
- Ugyanitt MTFF= $\frac{1}{\lambda} \sum_{i=1}^{k} \frac{1}{i}$ (levezetését ld. később, Markov-láncok segítségével); egy-egy újabb komponens egyre kevesebb „pluszt” jelent.

Kanonikus rendszer: Soros és párhuzamos komponensekből áll:

- Megbízhatósági blokk diagram: Komponensek a „blokkok”, a soros vagy párhuzamos kapcsolat jelenik meg a diagramon a „kapcsolásuk” formájában. A rendszer hibamentes, ha van út a kiindulási és a végpont között.
  - Ez nem az áramköri kapcsolás, hanem a megbízhatósági modell; ld. két párhuzamosan kapcsolt dióda rövidzár hibára soros, szakadás hibára párhuzamos megbízhatósági modellel jellemezhető.
- N egyforma egység; M komponens hiba esetén rendszerhiba:

  $$P\{k \textit{ hiba van}\} = \binom{N}{k}(1-r)^k r^{N-k}$$; összegzés kell M-re.
- Ideális többségi szavazás (TMR):

  $$r_R = \binom{3}{0}(1-r)^0 r^3 + \binom{3}{1}(1-r)^1 r^2 = 3r^2 - 2r^3$$

  Itt *MTFF*=$\frac{5}{6}\frac{1}{\lambda}$, rosszabb, mint egy komponens esetén! De az $r(t)$ görbe a kezdeti szakaszán magasabb, így missziókritikus rendszerekben jól alkalmazható.

Készenlét összevetése:

- Soros rendszer: $K_R = \prod_{i=1}^{k} K_i$, párhuzamos rendszer: $1 - K_R = \prod_{i=1}^{k}(1-K_i)$
- Példák: (1) párhuzamos rendszer, (2) stand-by rendszer nem-ideális kapcsolóval, (3) ideális többségi szavazás, (4) többségi szavazás nem-ideális szavazóval; szavazó határ-megbízhatósága.

3.2 Markov modellek

Használat: Modellezhetők a javított rendszerek, degradált állapotok illetve függőségben lévő komponensek.

Ismétlés a Markov láncokról:

- Diszkrét állapotok és átmenetek ezek között
- Emlékezetnélküliség

  $$P\{s(t_k) = s_j \mid s(t_{k-1}) = s_i \wedge s(t_{k-2}) = s_k \wedge \ldots \wedge s(t_0) = s_0\} = P\{s(t_k) = s_j \mid s(t_{k-1}) = s_i\}$$
- Állapotátmenet valószínűsége: $Q_{ij}(t_k, t_{k-1}) = P\{s(t_k) = s_j \mid s(t_{k-1}) = s_i\}$
- Homogén Markov-folyamat: $Q_{ij}(t + \Delta t, t) = Q_{ij}(\Delta t)$
  - Állapotátmeneti intenzitás: $R_{ij} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} Q_{ij}(\Delta t)$

    azaz $R_{ij} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P\{s(t + \Delta t) = s_j \mid s(t) = s_i\}$
  - Állapot elhagyásának összesített rátája s_i állapotban: $E_i = \sum_{j=1, i \neq j}^{k} R_{ij}$
- CTMC: Folytonos idejű Markov-lánc $(S, \underline{\underline{R}})$, itt R_{ij} értékek $\underline{\underline{R}}$-be rendezve
  - Állapotvalószínűségek: $\pi(s_i, s_j, t) = P\{s_i\textit{-ből indulva } t \textit{ idő elteltével } s_j\textit{-ben lesz a rendszer}\}$

CTMC tranziens megoldása: Állapotvalószínűség időfüggvények kiszámítása s_0 kezdőállapotra:

- Minden s_j állapotra $\pi(s_0, s_j, t)$, azaz a $\underline{\pi}(s_0, t)$ sorvektor; hasonlóan E_j értékek $\underline{E}$-be rendezve

- Megoldás alapja: $\underline{\underline{A}} = \underline{\underline{R}} - \mathbf{diag}(\underline{E})$ "infinitezimális generátormátrix"

$$\frac{d\underline{\pi}(s_0,t)}{dt} = \underline{\pi}(s_0,t)\underline{\underline{A}}$$

- A differenciál-egyenletrendszer megoldás módja: Laplace-transzformáció vagy idősorok:

$$\underline{\pi}(s_0,t) = \underline{\pi}(s_0,0)e^{-\underline{\underline{A}}t}, \text{ ahol } e^{\underline{\underline{A}}t} = \sum_{i=0}^{\infty} \frac{\underline{\underline{A}}t^i}{i!}$$

- Egyszerű eredmények:
  - $P\{s_i$*-ből* s_j*-be megy* t *időn belül*$\} = 1 - e^{-R_{ij}t}$
  - Tartózkodási idő: $P\{s_i$*-ben marad* t *ideig*$\} = e^{-E_i t}$
    állapotban tartózkodás ideje negatív exponenciális eloszlású valószínűségi változó

CTMC állandósult állapotbeli megoldása: Állapotvalószínűségek kiszámítása

- $\lim_{\Delta t \to \infty} \underline{\pi}(s_0,t) = \underline{\pi}(s_0)$ létezik, ha a CTMC véges állapotú és irreducibilis; itt kb: minden állapotból van javítás
- $\underline{\pi}(s_0,t) = \underline{\pi}(s_0)$ nincs időfüggés
- Megoldás alapja:

$$0 = \underline{\pi}(s_0)\underline{\underline{A}}, \text{ ahol } \sum_{j=1}^{n} \pi(s_0,s_j) = 1$$

Analógia a szolgáltatásbiztonság és a CTMC alapfogalmai között:

Szolgáltatásbiztonság	CTMC
F, H állapotok	s_i, s_j állapotok
$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P\{s(t+\Delta t) \in F \mid s(t) \in H\}$	$R_{ij}(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P\{s(t+\Delta t) = s_j \mid s(t) = s_i\}$
$\lambda(t) = \lambda$ konstans	$R_{ij}(t) = R_{ij}$, homogén a Markov-folyamat
$r(t) = P\{s(t') \in H, \forall t' < t\} = e^{-\lambda t}$ hibamentes állapotban maradás	$P\{s_i$-ben marad t ideig$\} = e^{-R_{ij}t}$ s_i állapotban tartózkodás ideje (itt $E_i = R_{ij}$)

CTMC modell alapelemei:

- Kezdőállapot (H része), degradált állapotok, hibajelenség (F része)
- Komponens meghibásodás: Ráta a λ meghibásodási tényező
  - Komponens hibamentes állapotának „tartási ideje": $e^{-\lambda t}$
- Több komponens állapotának összevont modellezése: $\lambda_a + \lambda_b + \lambda_c$ ráta;
  exp. eloszlású valószínűségi változók minimuma: exp. eloszlású, a paraméterek összegződnek.
- Állapotok összevonása: Feltétel: Átmenetek azonos kimenő rátákkal azonos állapot(ok)ba. Kimenő ráta ugyanaz marad a közös cél állapotba, bemenő ráták azonos állapotból összegezve lesznek.
- Javítás: μ javítási tényező (exponenciális eloszlás paramétere; matematikai kezelhetőség érdekében feltételezve)

- Teljes javítás (hibamentes állapotba), részleges javítás (degradált állapotba), megelőző javítás (degradált állapotból)

Eredmények egy F és H állapot-partíciókra osztott, s_0 kezdőállapottal rendelkező CTMC alapján:

- $r(t) = \sum_{s_i \in H} \pi(s_0, s_i, t)$ a javítási élek elhagyásával kapott CTMC alapján számolva
- $a(t) = \sum_{s_i \in H} \pi(s_0, s_i, t)$ a javítási élek bennhagyásával számolva
- $K = \sum_{s_i \in H} \pi(s_0, s_i)$ állandósult állapotban (létezik, ha minden hibaállapotból van javítás)

A CTMC állandósult állapotbeli megoldásának nehézségei: *Kiegyenlítetlenség* a numerikus értékekben

- A μ javítási tényező nagy (gyors javítás)
- A λ meghibásodási tényező kicsi (ritka meghibásodás)
- $\lambda + \mu$ jellegű összegek számítógépes ábrázolása nehéz ($1 + 10^{-20}$ jellegű értékekkel műveletek; pl. kivonás után nem elhanyagolható lesz a 10^{-20})

Mintapéldák felrajzolása:

- Nem javított rendszerek: (1) meleg tartalék, (2) csökkentett terhelésű tartalék, (3) hideg tartalék
- Stand-by rendszer: (1) ideális kapcsolóval, (2) nem ideális kapcsolóval, hibára kikapcsol illetve (3) hibára bekapcsolva maradó komponensekkel
- Javított rendszerek: (1) TMR preventív és teljes javítással, (2) egyszerelős rendszer (egyszerre csak egy elem javítható; egyszerűsödő állandósult állapotbeli megoldás)
- k számú meleg tartalék komponensből álló rendszer MTFF számítása Markov-láncokkal: $MTFF = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i}$, mivel minden állapotban $\frac{1}{(n-i)\lambda}$ a várható tartózkodási idő.

3.3 Sztochasztikus Petri-háló modellek

CTMC-nél magasabb szintű formalizmus (konkurens folyamatok jól modellezhetők), jó eszköztámogatással.

3.3.1 Formalizmusok

Petri-háló: PN=(P,T,I,O), ahol P helyek, T átmenetek, I input élek, O output élek.

Sztochasztikus Petri-háló: $SPN = (P, T, I, O, m_0, \Lambda)$, ahol

- m_0 a kezdeti jelölés
- $\Lambda : T \to \Re$ átmenetekhez rendelt időzítés, a negatív exponenciális valószínűségi eloszlásfüggvény paramétere.
  Az időzített átmenet engedélyezetté válásakor sorsol egy τ időzítést ebből a λ paraméterű eloszlásból, aminek letelte után tüzel:
  $P\{\tau < t\} = 1 - e^{-\lambda t}$
- SPN elérhetőségi gráfja alkot CTMC-t (jelölések valószínűsége számítható).

Általánosított sztochasztikus Petri-háló: $GSPN = (P, T, I, O, m_0, H, \Pi, L, G)$, ahol

- $H \subseteq P \times T$ tiltó élek
- $\Pi : T \to Z$ prioritások

- 0 az időzített átmenetek prioritása
- >0 az azonnali átmenetek priortása; ez alapján feloldhatók a konfliktusok közöttük.

- $L : T \rightarrow \Re$ paraméter:
  - időzített átmenetek esetén: az időzítés negatív exponenciális valószínűség eloszlásfv. paramétere,
  - azonnali átmenetek esetén: súlyok az azonos prioritású, konfliktusban lévő engedélyezett átmenetek közötti véletlenszerű választáshoz.
- $G : T \rightarrow$*Boole-fv.*; őrfeltétel, az adott átmenet engedélyezetté válásához igaznak kell lennie; a Boole-fv. a jelöléseken értelmezett (pl. *m(P)>2*, ahol *m(P)* a *P* hely jelölését jelenti.

Sztochasztikus "reward" hálózatok: Kiterjesztés a reward (haszon, jutalom) függvényekkel.

- *rate reward*: állapotokon (jelöléseken) értelmezett, időintervallumra összegezhető integrálva a rátát (a ráta időpillanatra vonatkozik).
  Példa: `if (m(Premium)==1) or m(Minimum)==1) then ra=1 else ra=0`.
- *impulse reward*: egy-egy eseményhez rendelhető (pl. tüzelés), időintervallumra összegezhető az impulzusok összeadásával.
  Példa: `if (fire(Repair)) then ri=12` (egy-egy javítás költsége 12 egység).

Időzített Petri-háló (Timed Petri Net): tetszőleges valószínűségi eloszlásfv. az időzítés sorsolásához.

- Itt már nem CTMC az alapszintű modell, analitikus megoldás helyett szimulációra lehet szükség.
- Pontos időzítési szemantika kell (ha a tüzelés megtörténte előtt megszűnik az átmenet engedélyezettsége, majd újra engedélyezett lesz, mi lesz az időzítés értéke: új időzítést sorsol az eredeti eloszlásfüggvény alapján, vagy az eredeti időzítésből maradt időt várja ki tüzelésig).

3.3.2 Részhálók a megbízhatósági modellezéshez

A GSPN formalizmust használva modellezzük a meghibásodási, hibaterjedési és javítási folyamatot. Az alapértelmezett részhálók adaptíven finomíthatók (az interfész helyek megtartásával).

Komponens meghibásodási folyamat: Hibamentes $\rightarrow$ hibaállapot $\rightarrow$ hibajelenség.

- Jellemzi a meghibásodási tényező és a hiba késleltetése (lappangási idő).
- Különböző lesz állapottal rendelkező vagy nem rendelkező komponensek esetén.
- Különböző, ha van/nincs kezdeti hiba, vagy ha a meghibásodás nem időfüggő.
- Interfész helyek: *H* (hibamentes), *E* (hibaállapot), *F* (hibajelenség).

Hibaterjedési folyamat:

- Jellemzi a hibaterjedési valószínűség illetve a komponens hiba és rendszerhiba közötti logikai kapcsolat (hibafa).
- Nem-redundáns rendszerek: Hibaterjedési részháló az interfész helyek között.

Komponens $\rightarrow$ (rész)rendszer állapotok közötti kapcsolatok:

- Nemredundáns komponens - részrendszer - rendszer hierarchia: H és F interfész helyek a (rész)rendszerekhez rendelve, OR hibafa alapján terjed a hiba felfelé.

- Redundáns komponensek és részrendszer: A hibaterjedést egy általános hibafa írja el, ezt valósítjuk meg a kapukhoz tartozó GSPN alhálók segítségével.
  A javítás „terjedésének” modellezésére a meghibásodáshoz felvett hibafa duálisát kel használni.

Javítási folyamat:

- Tranziens hiba implicit javítása: (csaknem) azonnali, de hibaterjedésre lehetőséget ad.
- Állandósult hiba explicit javítása: időzített átmenettel modellezhető.
- Javítási stratégiák modellezhetők: pl. nempreemtív prioritásos javítás közös javítóval, preemptív javítás stb.
- Feltételes javítás: Tiltó éllel modellezhető.