Pregled metoda procjene i mjerenja oslonjivosti informacijskog sustava

Nikola Šamec
Državni informacijski i komunikacijski sustav zaštite i spašavanja,
Državna uprava za zaštitu i spašavanje
Nehajska 5, 10000 Zagreb
nikola.samec@duzs.hr

Sažetak- **Oslonjivost informacijskog sustava odražava stupanj povjerenja korisnika u sustav. Današnja tehnologija omogućava korištenje naprednih informacijskih sustava umrežavanjem komunikacijskih mreža na različitim geografskim lokacijama. Takvi informacijski sustavi nazivaju se oslonjivi jer je nemoguće da se događaj u jednom informacijskom sustavu ne reflektira u drugome sustavu. Svrha ovog istraživanja je dati pregled dosadašnjih istraživanja u području oslonjivosti računalnih sustava sa naglaskom na metode procjene i mjerenja koje se koriste u analizi oslonjivih sustava. Dosadašnji radovi uglavnom su obuhvatili područje modeliranja i procjene te u manjoj mjeri metriku obilježja dostupnosti, pouzdanosti, sigurnosti i zaštite informacijskih sustava. Metode dostizanja oslonjivosti informacijskih sustava važan su segment u izgradnji oslonjivih sustava, a njihova važnost postaje veće što su sustavi veći i složeniji. Danas se često, zbog ekspanzivnog razvoja informacijske tehnologije, događa da su upravo sustavi u tom segmentu industrije u manjoj mjeri oslonjivi od željenih kriterija. Ovakve pojave dovode do prijetnji u radu operativnog sustava, te zahtijevaju rješavanje problema bez mogućnosti isključivanja sustava što je ponekad vrlo zahtjevan posao. Iz ovih činjenica proizlazi da je razvoj novih i napredniji metoda simulacije rada informacijskih sustava važan jer bi se na taj način predvidio veći broj neželjenih situacija u samoj fazi kreiranja oslonjivosti sustava tj. smanjio bi se broj prijetnji, a time i broj intervencija u vrijeme rada operativnog sustava.**

Ključne riječi- oslonjivost, procjena, mjerenje, metode

I. UVOD

Zaštita i preživljavanje složenih informacijskih sustava koji pružaju usluge infrastrukturi naprednog društva postaje važan prioritet nacionalna i svjetska značaja. [1]

Sustavi u realnom vremenu koriste se u kritičnom području primjene kao što su: svemirski letovi, nuklearna tehnologija, zrakoplovstvo i dr. Zatajenje ovakvih sustava može dovesti do velikih oštećenja, gubitka života i negativnog utjecaja na okoliš. Zajednički nazivnik takve vrste zatajenja u poslovnim sustavima je gubitak vrijednosti na tržištu kapitala. U svom istraživanju (Bharadwaj et al., 2009) navodi da zatajenje informacijskih sustava uzrokuje izvanprosječan pad od 2% cijene dionica u periodu od 2 dana nakon zatajenja sustava.[36]

Na temelju funkcije i opsega zatajenja sustava u realnom vremenu, razlikujemo tri vrste sustava, a to su sljedeći:

Sigurnosni-kritični sustavi koji su potrebni kako bi se osigurala sigurnost EUC (eng. equipment under control- oprema pod kontrolom) za ljude i okoliš. Primjeri takvih sustava su: gašenje sustava nuklearnog reaktora, digitalna kontrola leta zrakoplova i sl.[28]

Misijski-kritični sustavi su oni čije zatajenje rezultira neuspjehom misije. Na primjer, CCU (eng. control & coding unit – jedinica za kontrolu i kodiranje) jednog sustava zrakoplova ili sustav za navigaciju svemirske letjelice itd.[28]

Poslovni-kritički sustavi: za karakteristiku imaju zatajenje koje je rezultat nedostupnosti EUC-a (eng. equipment under control- oprema pod kontrolom) ili zatajenje npr. poslužiteljske ili komunikacijske opreme u poslovnom okruženju (sustav kontrole reaktora nuklearne elektrane, nedostupnost poslužitelja u bankarskom sektoru, kvar na usmjerivaču).[28]

Cilj ovog rada je prikaz metoda procjene i mjerenja oslonjivosti informacijskih sustava. Dati će se detaljniji osvrt na nekoliko bitnih pojmova važnih za razumijevanje ove tematike, a to su:

Koncept oslonjivost i terminološka struktura koju je u svom radu „Dependability: Basic Concepts and Terminology“ objavio Jean-Claud Laprie,

Metode za procjenu oslonjivosti sa naglaskom na Markovljeve lance.

Metode mjerenja oslonjivost pouzdanosti, dostupnosti, lakoće održavanja i sigurnosti,

Alati za podršku metodama za procjenu i mjerenju oslonjivosti informacijskih sustava.

II. OSLONJIVOST INFORMACIJSKOG SUSTAVA

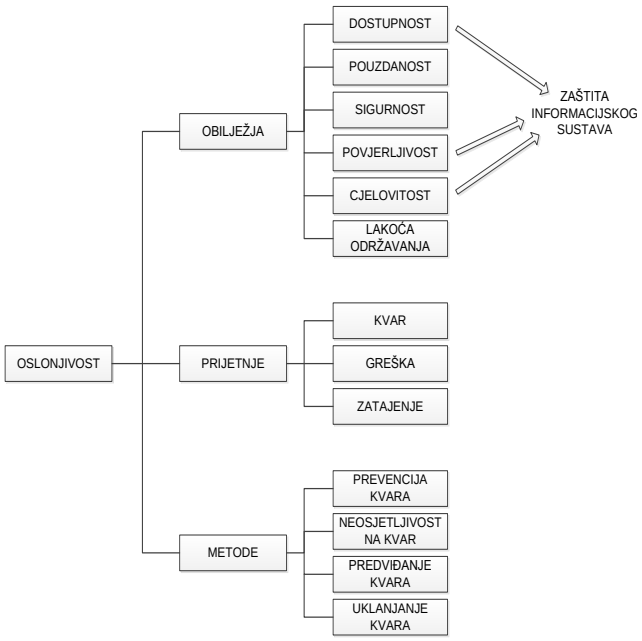
Osnova definicija oslonjivosti (eng. Dependability) informacijskih sustava je da je to sposobnost obavljanja usluge za koju jamčimo da će biti izvršena. Postoji i alternativna definicija prema kojoj je oslonjivost definirana kao sposobnost sustava da izbjegne češću i težu nedostupnost usluga od onoga što je prihvatljivo. [2]

Oslonjivost proučava dva ili više sustava koji djeluju međusobno u zavisnoj vezi, odnosno kako neki događaj u jednom sustavu utječe na rad drugih sustava te na cjelokupni sustav. Vremenski razvoj koncepta oslonjivosti kroz četiri desetljeća, autor Jean-Claud Laprie („Dependability: Basic Concepts and Terminology“), prikazao je terminološkom strukturom (Slika 1.). Iz čega su proizašla tri osnovna elementa promatranja:

- Obilježja oslonjivosti
- Prijetnje oslonjivosti
- Metode za postizanje oslonjivosti

A. *OBILJEŽJE OSLONJIVOSTI*

- *Dostupnost* (eng. Availavility) je pripravnost za isporuku ispravne usluge [2],[3],[32]
- *Pouzdanost* (eng. Reliability) je sposobnost sustava da kontinuirano isporučuje ispravnu uslugu [2],[3],[32]
- *Sigurnost* (eng. Safety) je sposobnost sustava da radi bez katastrofalnih posljedica na korisnika i okoliš [2], [4],[5],[32]
- *Povjerljivost* (eng. Confidentiality) je nepostojanje neovlaštenoga otkrivanja informacija [2],[3],[6],[7],[32]
- *Cjelovitost* (eng. Integrity), sprečavanje nepravilnih izmjena sustava [2],[3],[6],[7]
- *Lakoću održavanja* (eng. Maintainability) je sposobnost lakoće popravka i održavanja sustava [2],[3],[32]
- *Zaštita IS* (eng. Security) je sposobnost sustava da se štiti od slučajnih ili namjernih napada, a nastaje objedinjavanjem obilježja dostupnosti, povjerljivosti i cjelovitosti. [8],[32]



Slika 1. Prikaz elemenata oslonjivosti

B. *PRIJETNJE*

- *Kvar* (eng. Fault) je nedopušteno odstupanje od najmanje jedne karakteristične veličine sustava u prihvatljivim i uobičajenim standardnim uvjetima. [9]
- *Greška* (eng. Error) je stanje sustava koje može izazvati sistemsko zatajenje. Greška može biti latentna ili otkrivana [2],[3]
- *Zatajenje* (eng. Failure) je trajni prekid sposobnosti sustava za obavljanje tražene funkcije pod određenim uvjetima. [9]

C. *METODE ZA POSTIZANJE OSLONJIVOSTI*

- *Prevencija kvara* (eng. Fault prevention) je sprečavanje pojavljivanje kvara, a postiže se kontrolom kvalitete tijekom dizajniranja i proizvodnje hardvera i softvera. [2],[3]
- *Neosjetljivost na kvar* (eng. Fault tolerance) odnosi se na informacijske sustave koji isporučuju ispravnu uslugu uslijed postojanja aktivnog kvara i imaju dobro razvijenu neosjetljivost na kvar. Ova tehnika se obično koristi za otkrivanje grešaka i naknadni oporavak sustava. [2],[3],[32]
- *Predviđanje kvara* (eng. Fault forecasting) se provodi procjenjujući ponašanje sustava s obzirom na pojavu kvara. Procjenjuje se sadašnje stanje, budući kvarovi i posljedice kvara [2],[3]
- *Uklanjanje kvara* (eng. Fault removal) primjenjuje se u fazi razvoja i radnog vijeka informacijskih sustava. U fazi razvoja izvodi se: provjera, dijagnoza i popravak, dok se u fazi rada vrši korektivno i preventivno održavanje. [2],[3]

Kombinacijama ovih četiri metoda postiže se razvoj oslonjivog informacijskog sustava.

III. METODA PROCJENE OSLONJIVOSTI INFORMACIJSKOG SUSTAVA

Ovom metodom vrši se procjena sadašnjeg stanja vjerojatnosti pojave kvara, te procjena vjerojatnosti posljedica kvara. Primjenom ovih metoda moguće je procijeniti stupanj povjerenja na temelju sposobnosti sustava da zadovolji određene ciljeve. Procjena oslonjivosti primjenjuje se kod:

- Procijene usporedbe mogućih rješenja
- Predviđanja razine otpornosti u radu
- Procjene pouzdanosti, resursa i troškova na temelju kvantificiranih predviđanja [8]

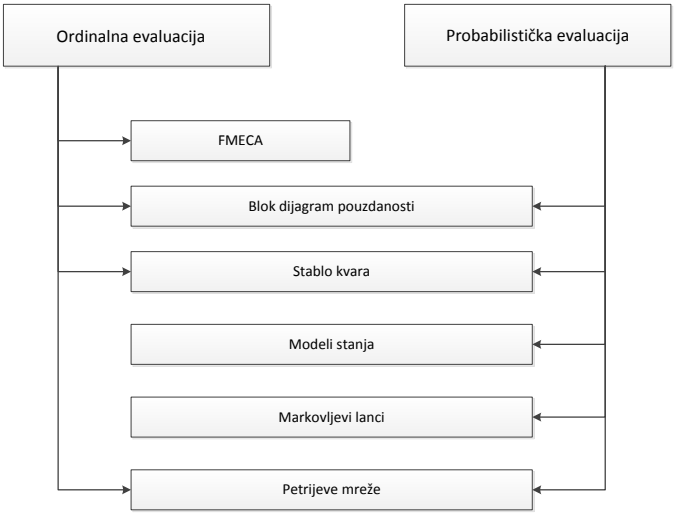
Procjena sustava može se promatrati kroz dva pristupa:

- *Kvalitativan* ili *ordinalni* pristup, pri čemu se vrši identifikacija, klasifikacija te rangiranje načina zatajenja ili kombinacije događaja koji uzrokuju zatajenje sustava [3]
- *Kvantitativan* ili *probabilistički* pristup uključuje ocjenjivanje vjerojatnosti u kojoj su se mjeri neki od atributa zadovoljili. Važna komponenta u ovom pristupu je mjerenje atributa. [3]

Na slici 2. prikazane su metode za procjenu koje se koriste za vrednovanje rješenja. Postoje dvije vrste metoda za procjenu:

Ordinalne (kvalitativna) metode imaju za cilj identificirati, klasificirati i rangirati načine zatajenja ili kombinacije događaja koji vode do zatajenja sustava.[2]

Probabilističke (kvantitativna) metode imaju cilj izradu procjene vjerojatnosti koliko su neki od atributa oslonjivosti zadovoljili zadane kriterije.[2]



Slika 2. Metode za procjenu
(izvor. ReSIST courseware-M. Kaâniche, K. Kanoun, J-C. Laprie — Dependability and Security Evaluation) [27]

A. ORDINALNE METODE ZA PROCJENU

1) FMECA metoda

FMECA (eng.kratica Failure Modes, Effects, and Criticality Analysis) metoda je zamišljena da se koristi za modeliranje hardvera, ali se naknadno počela primjenjivati i na softveru (SEEA: Software Error Effect Analysis). Izvorno je razvijena 40–ih godina za američku vojsku pod standardom MIL-P-1629.[25]. Ranih 60-ih standard je prihvatila NASA u svrhu razvoja svemirskih programa kao što su: Viking, Voyager, Magellan i Galilieo.[26]

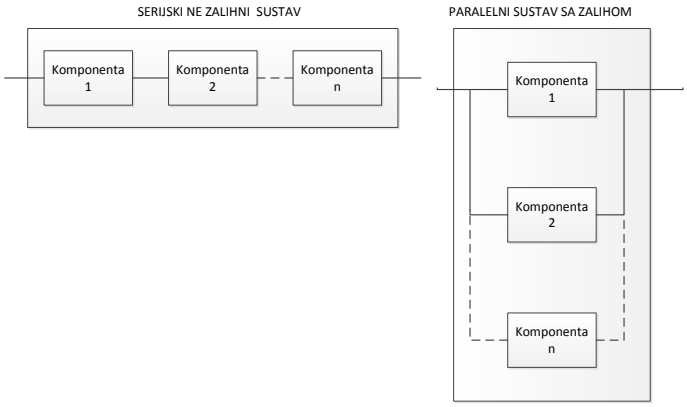
Ubrzo se metoda proširila na civilno zrakoplovstvo i auto industriju.

Metoda se koristi za prepoznavanje svake komponente ili funkcije, otkrivanje načina kako se zatajenje pojavljuje i koje posljedice uzrokuje na sustav. Moguće je napraviti procjenu za svaki mogući prepoznat slučaj zatajenja te izvršiti prioritizaciju na osnovu težine posljedice koju prouzrokuje zatajenje. FMECA proces prepoznaje kritične načine oštećenja te time pospješuje formalno priznanje rizika za projekt i daje poticaj za promjenu dizajna sustava.[8]

2) Blok dijagram pouzdanosti

Blok dijagram pouzdanosti (eng. Reliability Block Diagram) provodi analizu pouzdanosti i dostupnosti velikih i složenih sustava pomoću blok dijagrama za prikaz mrežne veze. Metoda je razvijena od strane američke vojske pod standardom MIL STD-756B.

Grafička metoda sastoji se od dvije vrste elemenata, blokova koji predstavljaju komponente sustava i „dummy“ čvoriša koji služe za povezivanje među komponentama. Grafička topologija opisuje kako pouzdanost pojedine komponente utječe na pouzdanost sustava. Oslonjivost modela blokova i čvorišta ovisni su o stanju svojih komponenata. Sustav se smatra operativan ako su krajnji „dummy“ čvorovi povezani, bilo kojem trenutku vremena,. Ako to nije slučaj, sustav se smatra neispravnim. Serijski sustavi smatraju se kao ne zalihni sustavi dok su paralelni zalihni sustavi (Slika 3.).[15]



Slika 3. Blok dijagram pouzdanosti prikaz serijskog i paralelnog sustava

3) Stablo kvara (eng. Fault Tree)

Ovu metodu razvio je H.A.Watson iz Bell laboratorija u suradnji sa američkim ratnim zrakoplovstvom, a kasnije se ova metoda proširila i na područje civilnog zrakoplovstvo. [29] Model predstavlja grafički prikaz kombinacije događaja koje uzrokuju pojavu željenog događaja. Stablom se mogu modelirati zatajenja hardvera i softvera, ljudske pogreške, greške prilikom održavanja sustava i utjecaj okoline na sustav. Model prepoznaje odnose između neželjenih događaja u sustavu i zatajenja podsustava koji pridonose zatajenju cijelog sustava. Model za procjenu pouzdanosti se razvija prema „top-down“ načelu, te je metodu moguće primijeniti u svakoj fazi dizajna sustava. Metoda daje kvalitativnu i kvantitativnu procjenu pouzdanosti sustava. Stablo kvara predstavlja aciklički graf s unutarnjim čvorovima (koji imaju logiku vrata npr. AND ili OR), vanjskim čvorovima (lišće ili osnovni događaji, a koji predstavljaju komponente sustava) i rubovima koji predstavljaju protok informacija zatajenja u smislu Boolean entiteta (TRUE ili FALSE, 0 ili 1). Veze koje su spojene na rubove utvrđuju operativnu ovisnost sustava na komponente. U bilo kojem trenutku vremena, logika vrijednosti korijenskog čvora određuje hoće li sustav biti operativan. [15]

4) Stablo napada

Stablo napada (eng. Attacks tree) je stablo odlučivanja, a temelji se na modeliranju uz pomoć grafičkih i matematičkih metoda. Ova metoda je u izvornom obliku razvijena za obavještajnu djelatnost i prvi puta se ideja o stablima logike prijetnji pojavljuje u literaturi kasnih 80 godina. Kasnije, B. Schneier u svom radu objavljuje koncept stabla napada. [10]. Metoda Stablo napada je usko vezana uz koncept ideje metode stabla kvara jer se opisuju skupovi događaja koji mogu dovesti do sistemskog zatajenja. Ovim stablom se modeliraju svi eventualni napadi na sustav, koji pružaju formalni, metodički način za opis sigurnosti sustava koji je temeljen na različitim vrstama napada. U stablu napada, napadi na sustav

prikazani su u strukturi stabala, pri čemu su ciljevi napada čvorovi korijena, a načini napada su čvorovi listova. Sigurnost velikog sustava se može modelirati sa skupom stabala napada, gdje korijen svakog stabla predstavlja napad koji može znatno oštetiti sustav. U strukturnom prikazu, stablo se prikazuje sa dvije vrste čvorova (AND ili OR čvorovi).
Stablo napada pruža sustavni pregled za opis sigurnosnih propusta, čime je moguće procijeniti rizike i donijeti sigurnosne odluke. [15] [31]

5) Graf pouzdanosti

Procjena pouzdanosti sustava je važan element za rješavanje problema u dizajniranju, izgradnji i održavanju sustava.
Graf pouzdanosti (eng. Reliability Graphs), G=(U,V) je poseban tip acikličkog dijagrama gdje U predstavlja set čvorova, a V set rubova u grafu. Svaka komponenta sastoji se od direktnih rubova spojenih sa dva čvora. Zatajenje komponente je prikazano kao brisanje ruba sa grafa. Neki specijalni rubovi označavaju se kao beskonačni i predstavljaju komponentu koja se ne može pokvariti. Dva čvora u grafu predstavljaju izvor i krajeve čvora. Izvorni čvor nema dolaznih rubova a krajnji čvor nema odlaznih rubova. Sustav se smatra ispravnim sve dok postoji i jedan direktan put od izvora prema kraju.[30]

B. PROBABILISTIČKA METODA PROCJENE

Veliki nedostatak prethodno opisanih ordinalnih metoda procjene je pretpostavka stohastička neovisnost između komponenti u sustavu što za posljedicu ima nemogućnost primjene u složenijim sustavima. U tu svrhu koriste se probabilističke metode procjene. Ove metode su temeljenje na prostoru stanja (eng. state space based) te se u ovu skupinu ubrajaju: vremenski neprekinuti Markovljev lanac (eng. CTMC), Markovljevi modeli nagrađivanja, Petrijeve mreže i dr.
Markovljev lanac predstavlja diskretni slučajni proces koji se najčešće označava kao dijagram prijelaza stanja. Markovljev „zakon“ kaže da sljedeći korak ovisi samo o trenutnoj situaciji, a što se može prikazati sljedećom jednadžbom 1.:

$$P\left(X_{n+1|}X_1, X_2, \dots, X_n\right) = P(X_{n+1}|X_n) \qquad (1)$$

Nemoguće je predvidjeti buduće stanje, ali je korisno za bilježenje statističkih svojstava. Markovljevi lanci poznaju sljedeća stanja prostora: početno, prijelazno i konačno.

Vremenski neprekinuti Markovljev lanac (eng. CTMC) je matematički model koji omogućuje promjenu stanja u svakom trenutku vremena. Generator matrice Q izražava stopu prijelaza umjesto vjerojatnosti. Opis stanja vremenskog neprekinutog Markovljevog lanca može se eksplicitno koristiti za praćenje stanja komponenti i podsustava jednog sustava. Svako stanje predstavlja posebno stanje greške, a prijelazi stopu kvara komponente. Stanja prikazuju broj kvarova komponenti u nekom vremenu. CTMC je vremenski homogen proces odnosno događaji kao zatajenja i popravci su neovisni jedan o drugome. [11]
Markovljev lanac s diskretnim vremenom (eng. DTMC) je matematički model koji vrši promjenu stanja sustava nakon točno određenog vremena. Prijelaz na sljedeće stanje ovisi o vjerojatnosti prijelaza. Svaki redak matrice vjerojatnosti prijelaza predstavlja izlaz iz tog stanja, a svaki stupac prijelaza je ulaz u stanje dok je suma reda jednaka jedan. Ovaj model se koristi za predviđanje vjerojatnosti pojave zatajenja hardvera u budućnosti. [11]
Petrijeva mreža je grafički obrazac za formalni opis logičke interakcije između dijelova ili tijeka aktivnosti u složenom sustavu. Izvorne Petrijeve mreže nemaju vremensku dimenziju, za proučavanje oslonjivosti te je potrebno uvesti veličinu trajanja događaja povezanu sa prijelaznim stanjima. Petrijeve mreže mogu se proširiti udruživanjem s veličinom vremena trajanja prijelaza što rezultira prikazom vremenske dimenzije. Poseban slučaj vremenskih Petrijeve mreža su stohastičke Petrijeve mreže (eng. Stochastic Petri Net, SPN) gdje se vremena okidanja smatraju slučajnim varijablama s eksponencijalnom distribucijom. SPN mogu se automatski pretvoriti u osnovni Markovljev model i na taj način riješiti. SPN u grafičkom smislu sastoji se od dvije vrste čvorova; mjesta i prijelaza. Mjesta obično predstavljaju uvjete u sustavu dok prijelazi predstavljaju događaje koji uzrokuju promjene uvjeta u sustavu. Oznake (eng. Token) su točke pridružene mjestu, a prikazuje stanje mjesta. Spojnice spajaju mjesta preko prijelaza, ako ulaze u mjesta onda su ulazne a ako izlaze onda su izlazne spojnice.[13]

1) Markovljevi lanci u procjeni oslonjivosti informacijskih sustava

Teorija Markovljevog procesa koristi se za izračunavanje obilježja dostupnosti primjenom modela stohastičke logike kontinuirane u vremenu i modela za provjeru algoritma. Markovljevim lancima procjenjuje se pouzdanost složenih hardvera sa sustavima otpornima na kvar (zalihnost). Ova probabilistička metoda primjenjuje se i za modeliranje softverske pouzdanosti. Ovakav pristup je pogodan za procjenu pouzdanosti već u fazi dizajniranja samog sustava, čak i prije nego što modeli „crne kutije“ i realnih komponenti ili softvera sustava postanu dostupni. Markovljev model može poslužiti kao temelj za Markovljev nagradni model (MRM) koji se koristi za mjerenje učinaka na sustavu degradacije performansi. Mjerenje se izvodi u kontekstu izvodivosti (eng. Performability), odnosno kombinacijom performansi i pouzdanosti te omogućuje nagrađivanje sustava za vrijeme provedeno u stanjima koja predstavljaju spremnost sustava. [12]

2) Stohastičke Petrijeve mreže u procjeni oslonjivosti informacijskih sustava

Stohastičke Petrijeve mreže (SPM) predstavljaju alat za opis i analizu sustava. Od početka primjene, ova metode koristi se za rješavanje problema u području procjene pouzdanosti, dostupnosti, performansi te analizi softverskih i hardverskih sustava. [12]
SPM se sastoji od ulaznih i izlaznih vrata, mjesta i aktivnosti. Aktivnosti (tranzicija u Petri mrežama) mogu biti vremenske i trenutne. Vremenske aktivnosti odnose se na aktivnosti koje utječu na vrijeme potrebno da sustav izvede zadatak, dok

trenutne aktivnosti predstavljaju aktivnosti sustava koje nisu vremenski ovisne.
SPM se ostvaraju pridruživanjem funkcije distribucije vremena aktivnosti (eng. activity time distribution function) sa svakom vremenskom aktivnošću i distribucijom vjerojatnosti za svaki skup slučaja. Sa vremenskom aktivnošću povezana je i funkcija reaktiviranja. Pomoću ove funkcije moguće je za svako obilježavanje unutar mreže pridružiti skup reaktivacijskih oznaka. Općenito, ako se aktivnost pokreće u određenoj oznaci, tj. skupu reaktiviranih oznaka, onda se aktivnost reaktivira uvijek kad se postigne jedna od oznaka u skup reaktivacijskih oznaka.
SPM mogu biti riješene analizom ili simulacijom, ovisno o karakteristikama sustava. Rješavanjem SPM-a dobije se procjena performansa tj. oslonjivosti sustava koje uključuju prezentacije značajki kao što su: paralelni rad, pravovremenost i neosjetljivosti na kvar. Analitičkim metodama rješavaju se kada su sve distribucije vremenskih aktivnosti eksponencijalne, a aktivnosti se aktiviraju dovoljno često da osiguraju da njihova stopa ovisi samo o trenutnom stanju. Kada je to slučaj, postoje stohastički procesi koji se mogu koristiti za dobivanje analitičkih rješenja. Ako to nije slučaj tada se simulacija može koristiti za procjenu stanja sustava.[38]

IV. METODE MJERENJA OSLONJIVOSTI INFORMACIJSKIH SUSTAVA

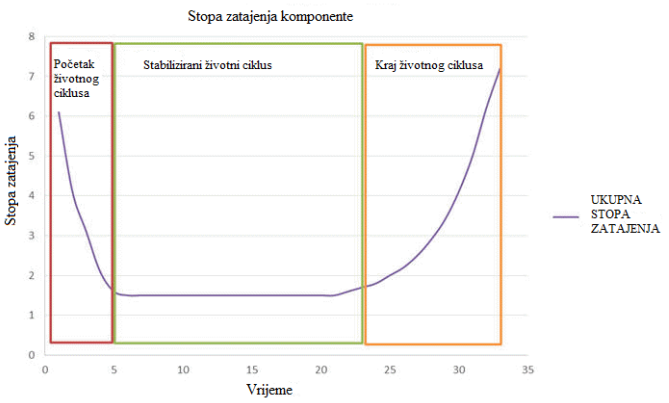
Za utvrđivanje oslonjivosti informacijskih sustava potrebna je primjena metrike.
Kvantitativna procjena oslonjivosti sustava može se podijeliti u dva koraka. Prvi korak je konstrukcija modela gdje se ponašanje razmatranog sustava temelji na osnovnim stohastičkih procesima koji odgovaraju ponašanju sustava komponenti i njihovih interakcija. U drugom koraku radi se matematička obrada modela gdje je izlazni proizvod analitike brojčana vrijednost mjere pouzdanosti sustava.
Obrada modela ocjenjuje se kroz kvantitativne mjere kako je objašnjeno u prethodnom poglavlju primjenom kombinatornih (blok dijagram pouzdanosti, stablo kvara, itd.) i ordinalnih metoda (Markovljevi lanci, stohastičke Petrijeve mreže itd).
Evolucija oslonjivosti u odnosu na životni ciklus sustava karakterizirana je atributima kao što su stabilnost, rast i smanjenje. Ovi pojmovi prikazuju učestalost zatajenja, odnosno broj zatajenja u jedinici vremena koje primjećuje korisnik. Učestalost zatajenja u prvom dijelu se smanjuje (rast pouzdanosti), zatim slijedi period stabilizacije (stabilna pouzdanost) da bi se nakon nekog vremena učestalost zatajenja povećala (smanjenje pouzdanosti). [2, 3]
Izmjenjivanjem ispravnog/neispravnog stanja pružanja usluga mjeri se pouzdanost, dostupnost i lakoća održavanja. [2, 3]

A. POUZDANOST

Mjerenje pouzdanosti je vjerojatnost da sustav tijekom određenog vremenskog intervala pruža ispravnu uslugu. Analiza pouzdanosti ovisi o stohastičkom modelu uzrokovanja, trajanja i učestalosti zatajenja hardvera i softvera.

1) Hardversko zatajenje

Ovaj tip zatajenje je obično karakteriziran kao krivulja kade (Slika 4.). Mogućnost kvara hardvera je visoka u početnom dijelu životnog ciklusa sustava da bi se nakon nekog vremena stabilizirala. Kako životni ciklus sustava dolazi kraju, povećava se mogućnost kvara.



Slika 4. Stopa zatajenja komponenta
Izvor: http://www.eventhelix.com

2) Softversko zatajenje

Za proučavanje softverskih zatajenja bitno je bilježiti povijest zatajenja. Softverska zatajenja ovisna su o složenosti softvera, veličini koda, iskustvu programera, postotku koda ponovo upotrjebljenog iz prethodnih stabilnih projekta i testiranju softvera prije upotrebe.

3) Parametri pouzdanosti

- MTBF (eng. Mean Time Between Failure, srednje vrijeme između zatajenja) je srednje vrijeme zatajenja hardverske komponente koju je procijenio proizvođač. MTBF za softvere dobije se množenjem zatajenja s brojem izvršenih procesa u sekundi
- MTTR (eng. Mean Time to Repair, srednje vrijeme do popravka) je srednje vrijeme potrebno za popravak hardverskog modula. MTTR za softverski modul računa se kao vrijeme potrebno da se softver ponovno pokrene nakon kvara. Vrijednost MTTR mora težiti prema 0.

B. DOSTUPNOST

Mjera za isporuku ispravne usluge s obzirom na ispravnu/neispravnu uslugu: [2, 3]
Dostupnost hardverskog ili softverskog modula može se prikazati jednadžbom 2.:

$$A = \frac{MTBF}{MTBF+MTTR} \quad (2)$$

Dostupnost sustava može se mjeriti modeliranjem sustava povezivanjem dijelova sustava u seriju i paralelu.

1) Dostupnost u seriji

Ako zatajenje komponente uzrokuje zatajenje cijelog sustava smatra se da su komponente spojene serijski (Slika 5.)

Slika 5. Komponente spojene u seriju

Iz slike 5 može se zaključiti kako je sustav dostupan samo ako su komponente X i Y dostupne što se matematički može prikazati sljedećom jednadžbom 3.:

$$A = A_x A_y \tag{3}$$

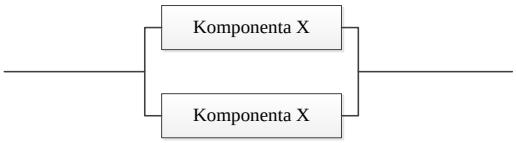
Iz jednadžbe proizlazi da je dostupnost sustava manja od dostupnosti pojedinih komponenti sustava.

2) Dostupnost u paraleli

Ako zatajenje svih komponenti uzrokuje zatajenje cijelog sustava smatra se da su komponente spojene paralelno (Slika 6). Sustav se smatra ispravnim sve dok je i zadnja komponenta dostupna. Iz toga slijedi da je u kombinaciji dostupnost 1 (obje komponente su nedostupne). Paralelna dostupnost može se izraziti jednadžbom 4.:

$$A = 1 - (1 - A_x)^2 \tag{4}$$

Dostupnost dvije komponente u paraleli je uvijek puno veća od dostupnosti pojedinih komponenti sustava.



Slika 6. Komponente u paraleli

C. *LAKOĆA ODRŽAVANJA*

Mjera vremena za obnovu isporuke usluge od posljednje neispravnosti ili mjera kontinuiranog pružanja neispravne usluge: [2, 3]

Model za određivanje kvalitete softverskog proizvoda (standard ISO 9126) uzima u obzir nekoliko glavnih značajki (kvaliteta, prilagodljivost, promjenjivost, stabilnost, provjerljivost) među kojima je i indeks lakoće održavanja (MI). Mjerenje lakoće održavanja provodi se kroz procjenu vanjskih i unutarnjih karakteristika te karakteristika kvalitete. Za određivanje karakteristika lakoće održavanja definirano je 16 vanjskih mjerila i 9 unutarnjih mjerila kvalitete. [14]

1) Vanjska mjerenja

Predložena vanjska mjerenja računaju se mjerenjem učinkovitosti aktivnosti održavanja od strane tehničkog osoblja. [14]

ISO TR 9126-Part 2: External Measures je tehničko izvješće koje opisuje popis mjera (metrika) za procjenu različitih karakteristike definiranih u modelima.[34]

2) Unutarnja mjerenja

Predložene unutarnje metode mjerenja temelje se na usporedbi značajki modela koji trenutno radi i zamišljenog modela. Unutarnja mjerenja mogu se provoditi i mjerenjem aktivnosti održavanja [14]

ISO TR 9126- Part 3: Internal Measures definira unutarnju metriku za kvantitativno mjerenje kvalitete u smislu karakteristika i podznačajki definiranih u ISO/IEC 9126-1.[35]

D. *ZAŠTITA*

Zaštita informacijskog sustava predstavlja sposobnost sustava da se štiti od slučajnih ili namjernih napada u nekom vremenu.

Indeks računalne sigurnosti (eng. Security Computation Indeks - SCI),“ SCI“ opisan je u radu Soh-a 1993 godine, a izračunava pomoću Markovljevih lanaca. Cilj ove metode je kvantificirati ukupni sigurnosni aspekt. Indeks sigurnosne ranjivosti (eng. Security Vulnerabity Index), „SVI“ predložen je od strane Alvess- Foss-a 1995 godine. Indeks se izvodi uz procjenu tri čimbenika. Prisutnost (ili odsustvo) čimbenika utječe na ukupnu ranjivost sustava, a indeks može poprimiti vrijednosti od 0 do 1.

Pristup metodom kvantitativnog vrednovanja pod nazivom „graf privilegija“ predložen je od strane Daciera, 1994 godine. Ovaj koncept temelji se na stohastičkim Petrijevim mrežama i modelima matrice.[16]

E. *SIGURNOST*

Mjerenje sigurnosti usko je povezano sa mjerenjem pouzdanosti. Sigurnost sustava, kao funkcija vremena $S(t)$, je vjerojatnost da neće doći do katastrofalnog zatajenja u periodu $[t_0, t]$. [24]

U radu Tang i ostali [41] predstavljena je metoda mjerenja sigurnosti sustava instrumenta i kontrole. Metoda uzima u obzir podatke o zatajenjima sustava te vrši procjenu pouzdanosti, dostupnosti, identificira problematična područja. Ova metoda mjerenja sigurnosti koristi se u nuklearnoj tehnologiji i kontroli zračne plovidbe.

F. *MJERENJE UKUPNE OSLONJIVOSTI*

Mjerenje oslonjivosti (kao što su pouzdanost,dostupnosti, sigurnost) su važni kriteriji za projektiranje informacijskih sustava, kao i za procjenu. U naredna tri primjera dane su metode koje predlažu različiti autori kako mjeriti ukupno oslonjivost sustava

Alternativno mjerenje oslonjivosti koje uključuje sigurnost predloženo je od strane McDermida, 1994 godine [17]. On je proučavao oslonjivost korištenjem pojmova gubitka i rizika. Te je njihove vrijednosti u jedinici vremena predložio kao mjerilo oslonjivosti. Procjena rizika radi se u ranoj fazi dizajniranja sustava, dok se gubitak mjeri u operativnoj fazi. Prednost ovakvog mjerenja je što se vrlo lako može prikazati u ekonomskim terminima.

Jonsson i ostali[39] u svom radu predlaže vektorizirano mjerenje temeljeno na Markovljevom procesu. Mjerenje obuhvaća obilježja pouzdanosti, sigurnosti i zaštite. Metoda mjerenja pogodna je za mjerenje oslonjivosti na autonomnim sustavima kojima upravlja računalo, kao što su svemirske letjelice i razni kontrolni sustavi.

Zanimljiva ideja za mjerenje oslonjivosti sustava predložena je u radu autora Rüdiger i ostalih).[40] Ideja mjerenja oslonjivosti se svodi na proučavanje ponašanja sustava kada je

sustav zaposlen odnosno kada nije zaposlen u funkciji vremena $D(t)$.

V. PROGRAMSKA PODRŠKA METODAMA PROCJENE I MJERENJA OSLONJIVOSTI INFORMACIJSKIH SUSTAVA

U nastavku su navedeni programski paketi za modeliranje oslonjivosti, uključujući metode modela izgradnje i modela rješenja. Neki od novijih programskih paketa za analizu oslonjivosti su: SURF-2, GREAT-SPN, Ultra SAN, Möbius, SHARPE, DRAWNET++, SPNP, DEEM, TimeNET, DSPNexpress, ADVISER, ARIES, CARE III, METFAC, SAVE, SURE, ASSIST, HARP i dr. Pretpostavke i svojstva programskih paketa analiziraju se prema različitim kriterijima, podržanoj strukturi, rješenju metode, kreatoru metode i drugim elementima. U tablici 1. navedeni su najpoznatiji alati za modeliranje i mjerenje oslonjivosti s obzirom na primijenjene tehnike i kreatora alata.

Naziv alata (metode)	**Korišteni modeli**	**Kreator**
SURF-2	GSPN, Markovljev lanac	LAAS, Francuska
Great-SPN	GSPN i stohastičke dobro oblikovane mreže (SWN)	Torino, Italy
UltraSAN	Stohastičke aktivne mreže (SAN)	UIUC, USA
Möbius	SAN, Markovljevi lanci…	UIUC, USA
SHARPE	stablo kvara, modele čekanja u redovima, Markovljevi lanci, SAN…	DUKE,SAD
SPNP	SPN, Stohastičke nagradne Petirjeve mreže, ne Markovljevi modeli	DUKE,SAD
DRAWNET++	Stabla kvara, SWN	U.del Piemonte orientale, U.Torino, U. Napoli, Italy
DEEM	SPN, Markovljev regenerativan proces	UNIFI-PISA, Italy
Time NET	ne Markovljevi SPN	Hamburg, Germany
DSPNexpress	Determinističke i stohastičke Petrijeve mreže	Dortmund, Germany

Tablica 1. Pregled alata s obzirom na korišteni model i stvaratelja [27]

A. *SURF-2*

Surf 2 je alat za procjenu oslonjivosti hardverskih i softverskih sustava, a temelji se na strogoj izgradnji, validaciji i numeričkom rješavanju Markovljevih modela. Model je izgrađen u laboratoriju LAAS, Francuska 1996. godine. Sustav ponašanja modeliran je sa Markovljevim lancima i stohastičkim Petrijevim mrežama. Glavna ideja modela je da se jednostavnim načinom uspoređuje pouzdanost različitih arhitektura sustava. U model je moguće dodati „nagradnu" strukturu kako bi se kombinirale mjere pouzdanosti, performansa i cijena. [18]

B. *Great-SPN*

GreatSPN je alat koji podržava dizajn te kvalitativnu i kvantitativnu analizu generalizirane stohastičke Petrijeve mreže i stohastičkih dobro-oblikovanih mreža (SWN). GreatSPN pojavio se kasnih 80-godina prošlog stoljeća. Od njegovog objavljivanja do danas razvijeno je više različitih verzija ovog alata, kao što su npr.: GreatSPN1.7 i GreatSPN2.0.

Kako bi se razumjela struktura ovog alata i način njegovog korištenja za primjer se može uzeti jedna od prethodno spomenutih verzija. GreatSPN2.0 sastoji se od zasebnih programa koji surađuju u izgradnji i analizi PN modela dijeljenjem datoteke. Korištenjem mogućnosti komunikacijske mreže, moguće je izvesti razne analize na različitim strojevima u distribuiranom računalnom okruženju. Modularna struktura GreatSPN2.0 omogućuje dodavanje novih modula za analizu, kao i novih istraživačkih rezultata. Svi moduli su pisani u C programskom jeziku koji jamči prenosivost i učinkovitost na različitim Unix strojevima. Sva rješenja modula koriste mrežnu i lokalnu pohranu podataka. U alatu su ugrađene sljedeće osnovne analize: algoritmi za brzo izračunavanje performansi granica na temelju linearne tehnike programiranja (rade na strukturnoj razini), algoritmi za analizu stohastičkih dobro oblikovanih mreža koje pružaju korisniku mogućnost izrade modela složenih sustava i učinkovitiju analizu stanja prostora. [19],[33]

C. *UltraSAN*

UltraSAN je programski alat za procjenu sustava baziran na modelu zastupljenih stohastičkih aktivnih mreža (SAN). SAN ima obilježja stohastičke Petrijeve mreže i modela čekanja u redovima, a koristi sljedeće parametre: distribuciju, vremensku aktivnost, pravila stanja tranzicije i varijablu nagrađivanja. Korištenjem raznih analitičkih i simulacijskih modula, moguće je ovim alatom odrediti: učinkovitost, oslonjivost i izvodivost. UltaSAN omogućava i grafički prikaz rezultata dobivenih iz izvješća. Za određivanje valjanog modela, potrebno je specificirati skup podmreža pomoću SAN editora i svrstati prema hijerarhijskoj ljestvici. Specificiranjem modela moguće je koristiti parametre kao npr. globalne varijable koje mogu biti vrijednost ili raspon vrijednosti, a skup tih vrijednosti zove se studija.

UltraSAN nudi šest analitičkih tehnika za rješavanje prijelaznih i stacionarnih stanja. Tri tehnike koje omogućuju rješavanje stacionarnih stanja: *direct steady-state, iterative steady state*, i deterministički *iterative steady state*. Za rješavanje prijelaznih stanja koriste se *transient instant-of time*, *PDF interval-of-time* i *expected interval-of-time* tehnike.[23] Za rješavanje stacionarnih stanja moguće je računati srednju vrijednost, varijancu i gustoću vjerojatnosti, a za rješavanje uniformnih prijelaznih stanja moguće je izračunati srednju vrijednost, distribuciju u vremenu, varijancu i gustoću tijekom vremenskog intervala. Isto tako alat je moguće koristiti i za simulaciju prijelaznih i stacionarnih stanja modela sa općenitom distribucijom aktivnosti. [20]

D. *Möbius*

Möbius (eng. kratica za Model-Based Environment for Validation of System Reliability, Availability, Security and Performance) je programski alat za modeliranje ponašanja kompleksnih sustava. Alat je osmišljen na University of Illinois, USA, a izvorno je zamišljen za proučavanje pouzdanosti, dostupnosti i učinkovitosti računalnih i mrežnih sustava. Njegov prilagodljiv pristup omogućuje inženjerima i znanstvenicima da predstavljaju svoje sustave u modelu jezika koji odgovara njihovom problematičnom području, a zatim točno i učinkovito rješavanje sustava koristeći tehnike rješavanja koje najbolje odgovaraju veličini i složenosti problema. Alat podržava stohastičke Petrijeve mreže, Markovljeve lance i stohastičke procese algebre. Modeli se prezentiraju numerički i grafički, a izrađeni su sa pravom razinom detalja te imaju mogućost prilagođavanja ponašanju sustavu interesa. Ovim alatom mogu se izgraditi detaljni matematički izrazi koji mjere točnu informaciju o sustavu (npr., pouzdanost, dostupnost, performanse i sigurnost). Mjerenje može biti provedeno u određenim vremenskim točkama (tijekom razdoblja ili kad sustav dosegne stacionarno stanje).Funkcionalnost sustava može se definirati i kao model ulaznih parametara. U takvom se slučaju ponašanje sustava može automatski proučavati preko širokog opsega vrijednosti procesnih parametara, a s konačnim ciljem određivanja područja sigurnog rada sustava, te otkrivanja mogućih ograničenja sustava. Dodatno je, moguće proučavati i različita ponašanja sustava koja je inače nemoguće eksperimentalno odrediti upotrebom prototipa. [21]

E. *SHARPE*

SHARP (eng. kratica za Symbolic Hierarchical Automated Reliability and Performance Evaluator) je alat koji pruža specifikaciju i metode rješenja za većinu najčešće korištenih vrsta modela za procjenu performanse, pouzdanosti i izvodivosti. Modeli koji su prisutni u SHARP-u su; stablo kvara, model čekanja u redovima i modeli stanja prostora (Markovljevi lanci, semi Markovljevi lanaci sa nagradom i stohastičke Petrijeve mreže). SHARP-om se mogu mjeriti intervali, stacionarna i prijelazna stanja. Za svaki model SHARP ima više analitičkih algoritama. Ovaj alat omogućuje mjerenje modela koji se može upotrijebiti kao parametar drugog modela, stoga se smatra da je alat hijerarhijski orijentiran. Korisničko sučelje podržava command-line i grafičko sučelje (GUI) koje je izrađeno u Java na Duke University, USA. [22]

VI. ZAKLJUČAK

Termin oslonjivost sustava je relativno nov, a osniva se na stvarnim potrebama poslovnih sustava za pouzdanim, dostupnim i sigurnim cjelovitim sustavom koji se lako održava i štiti povjerljive podatke, te je zaštićen od vanjskih i unutarnjih prijetnji i siguran je za svoju okolinu. Pregledom literature, osim za potrebe poslovnih sustava (npr. auto industrija, zrakoplovstvo), oslonjivi informacijski sustavi razvijaju se za potrebe misijskih i kritičnih sustava svemirskih programa NASA, ESA te vojne industrije, te su upravo ove organizacije prve standardizirale metode za procjenu, mjerenje i modeliranje (koje nije obrađeno u ovom radu) oslonjivih sustava. Trend razvoja oslonjivih sustava kreće se u smjeru novih istraživanja sigurnosti, zaštiti IS-a i pouzdanosti informacijskih sustava u domeni kritičnih sustava. Proaktivnim pristupom prema novim metodama, metrici, alatima za osiguranje sigurnosti, forenzici, upravljanju slabostima sustava i individualnim pristupu traže se rješenja za buduće okoline informacijskih sustava. U radu je obrađena izvorna terminologija koju je Laprie definirao 80-ih godina prošlog stoljeća. Procjena sustava oslonjivosti je važna jer se vrši procjena trenutnog stanja vjerojatnosti pojave kvara, mogućnost pojave i procjena vjerojatnosti posljedice kvarova. U radu su opisani i najčešći primjenjivani modeli ordinalne i probabilističke prirode. Mjerenjem oslonjivih sustava dobivamo vrijednost koliko je oslonjivost nekog sustava bolja od drugih, tako da su u radu prikazani i modeli i načini mjerenja pouzdanosti, dostupnosti, zaštite, sigurnosti i lakoće održavanja.

Daljnji smjer istraživanja oslonjivosti informacijskih sustava biti će posvećen pronalaženju nove metode procjene i mjerenja oslonjivosti informacijskih sustava u poslovnim organizacijama. Ideja je da se procjena i mjera iskažu kroz trošak zatajenja u nekom vremenu t.

Činjenica da zatajenje informacijskog sustava (temeljenog na informacijskoj i telekomunikacijskoj tehnologiji) može u konačnici rezultirati značajnim troškovima, dovela je do potrebe da se takva vrsta događaja dobro unaprijed odredi i procijeni. [37]. Izrađena metoda procjene i mjerenja oslonjivosti informacijskog sustava u poslovnim organizacijama će proći praktičnu evaluaciju na konkretnim poslovnim organizacijama, te će biti ocjenjena od strane eksperata iz područja.

VII. LITERATURA

[1] Wilson, STRATUS Computer System", in Resilient Computing Systems, 1985, pages 208-231

[2] Avižienis A, Laprie J-C, Randall B (2000) Fundamental concepts of dependability. In: Proc. of 3rd Information Survivability Workshop, pp 7-11, October 2000

[3] Avižienis A, Laprie J-C, Randell B, Landwehr C (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Transaction Dependable and Secure Computing, January-March 2004 (vol. 1 no1.), pp. 11-33

[4] IEC 61508: Functional safety of electric/electronic/programmable electronic safety-related systems, Parts 0-7; Oct. 1998-May (2000)

[5] Atoosa Thunem P-J (2005). Security Research from a Multi-disciplinary and Multi-sectoral Perspective. Lecture Notes in Computer Science (LNCS 3688). Springer Berlin / Heidelberg

[6] Ross J.Anderson (2001) Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley Computer Publishing, USA

[7] MIL-STD-1553B: Aircraft internal time division command/response multiplex data bus, 30 April 1975

[8] Mohamed Kaâniche, Karama Kanoun, Jean-Claude Laprie, Dependability and Security evaluation of computer-based systems 2009

[9] Rolf Isermann, Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Springer 2006

[10] Schneirer, Dr Dobbs Journal, Atack Tress, December 1999

[11] Dr. Peter Tröger, Dependable Systems, State-Based Dependability Modeling, Dependable System Course 2013, pp 8-29

[12] Irene Eusgeld, Bernhard Fechner, Felix Salfner, Max Walter, Philipp Limbourg and Lijun Zhang, Hardware Reliability, Springer 2008

[13] Jens Happe, Analytical Performance Metrics in Dependability Metrics, Springer 2008, pp 214 - 218

[14] Ilja Heitlager, Tobias Kuipers , Joost Visser, A Practical Model for Measuring Maintainability, 2007 in QUATIC '07 Proceedings of the 6th International Conference on Quality of Information and Communications Technology, pp30

[15] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model-based evaluation: From dependability to security. IEEE Trans. on Dependability and Security, 1(1):48–65, 2004.

[16] Erland Jonsson, An integrated framework for security and dependability, in NSPW '98 Proceedings of the 1998 workshop on New security paradigms,Pages 22-29

[17] J. McDermid, "On Dependability, Its Measurement and Its Management", in High Integrity Systems, Vol. 1, No. 1, 1994, Oxford University Press, pp. 17-26.

[18] http://homepages.laas.fr/surf4tst/what-uk.html#modele.

[19] http://www.di.unito.it/~greatspn/index.html

[20] W. Douglas Obal II, M. Akber Qureshi, Daniel D. Deavours, William H. Sanders, Overview of UltraSAN, in Computer Performance and Dependability Symposium, 1996., Proceedings of IEEE International, 4-6 Sep 1996

[21] https://www.mobius.illinois.edu/

[22] http://sharpe.pratt.duke.edu/

[23] W.H. Sanders, W.D.Obal II, M.A.Qureshi and F.k. Widjanarko, UltraSAN Ver3: Architecture, Features, and Implementation, 1995

[24] Jean-Claude Laprie, Dependable Computing: Concepts, Limits, Challengs, Invited paper to FTCS-25, the 25th IEEE International Symposium on Fault-Tolerant Computing,Pasadena, California, USA, June 27-30, 1995, Special Issue, pp. 42-54.

[25] U.S. Department of Defense, Procedures for Performing a Failure Mode, Effects and Criticality Analysis, MIL–P–1629, 1949

[26] NASA, Failure Modes, Effects and Criticality Analysis (FMECA), Practice no. PD-AP-1307

[27] ReSIST courseware-M. Kaâniche, K. Kanoun, J-C. Laprie — Dependability and Security Evaluation, 2009

[28] Ajit Kumar Verma • Srividya Ajit, Manoj Kumar, Dependability of Networked Computer-based Systems, Springer, 2011

[29] Clifton A. Ericson II; Fault Tree Analysis-A History from the Proceedings of The 17th International System Safety Conference, 1999

[30] Jogesh K. Muppala, Ricardo M. Frics and Kishor S.Trivedi, Techniques for system dependability evaluation, Springer, 2000 pp 9-10

[31] Terrance R Ingoldsby Amenaza Technologies Limited, Attack Tree-based Threat Risk Analysis,2010

[32] Al Avizienis, Jean-Claud Laprie, Brian Randell, Dependability and its Threats: A Taxonomy, 18th IFIP World Computer Congress, Toulouse 2004

[33] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli and G. Franceschinis,The GreatSPN Tool: Recent Enhancements, ACM Performance Evaluation Review Special Issue on Tools for Performance Evalaluation, Volume 36, Issue 4, September 2009, Pages 4-9.

[34] "ISO/IEC TR 9126-2: Software engineering - product quality - part 2: External metrics," Geneva, Switzerland, 2003

[35] "ISO/IEC TR 9126-3: Software engineering - product quality - part 3: Internal metrics," Geneva, Switzerland, 2003.

[36] Anandhi Bharadwaj, Mark Keil, Magnus Mähring, Effects of information technology failures on the market value of firms, Journal of Strategic Information Systems, 18:66-79, 2009. (Best Paper Award in JSIS, 2009).

[37] Jakupović Alen, Utjecaj oslonjivosti informacijskog sustava na poslovne organizacije, 2013

[38] W. H. Sanders and J. F. Meyer, „A Unified Approach for Specifying Measures of Performance, Dependability, and Performability," in Dependable Computing for Critical Applications, Vol 4: of Dependable Computing and Fault-Tolerant Systems (ed., A. Avizienis and J. Laprie), Springer-Verlag, 1991

[39] E. Jonsson, S. Asmussen, A Practical Dependability Measure For Embedded Computer Systems, in Proc. IFAC 12th World Congress, Vol. 2, (Sydney, Australia), pp.647-52, 1993.

[40] Jan Rüdiger, AchimWagner and Essam Badreddin, Dependability of Autonomous Mobile Systems, ICINCO-RA 2, page 137-142. INSTICC Press, (2008)

[41] Dong Tang, Myron Hecht, Herbert Hecht, Robert Brill, Measurement-Based Dependability Evaluation for Safety-Grade Digital Systems, Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies, University Park, PA, May 6-9, 1996, pp. 535-542.