۴. پیش‌بینی خطاهای به‌هم‌وابسته[1]

مفاهیم کلّی

پیش‌بینی خطا جنبه‌ای است که برای تأمین قابلیت اعتماد[2] سیستم به آن رجوع‌می‌شود. پیش‌بینی خطا، یکی از اجزاء و مؤلّفه‌های اجتناب خطا[3] را تشکیل‌می‌دهد. برای پیش‌بینی خطا، رفتار سیستم را حین بروز خطا ارزیابی‌کرده و برپایه‌ی مجموعه‌ای از روش‌ها و تکنیک‌هایی که به برآورد *بروز*[4]، *خلق*[5] و *پی‌آمدهای*[6] خطا می‌پردازد، این عمل را انجام می‌دهیم.

پیش‌بینی خطاهایی که با منابع بروز خطاهای هم‌شیوه مرتبط‌اند، برمبنای تقسیم‌بندی اعمال‌شده در قسمت۱-۲ دسته‌بندی می‌گردد. (وجود منبع آغازگر[7] و رابطه[8]ی بین دو زیرسیستم یا مؤلّفه)

جنبه‌های "**خلق و بروز خطا**" در تکنیک‌های پیش‌بینی درجهت تعیین منابع آغازگر یا هم‌بسته‌ی CMFها مؤثرند. در حالی‌که "**پی‌آمدها**" برای تعیین دقیق طبیعت شکست به‌کارگرفته‌می‌شوند.

ارزیابی رفتار سیستم در رابطه با خطاهای غیرمترقّبه[9]، (آنچه در پیش‌بینی خطا مورد نیاز است)، می‌تواند کیفی یا کمّی مورد بررسی قرارگیرد[LAPR]:

- **ارزیابی کیفی**[10]

ارزیابی کیفی برای تعیین و دسته‌بندی شکست‌ها یا روش‌هایی که برای اجتناب از آن‌ها به‌کارمی‌رود، مورداستفاده‌قرارمی‌گیرد. این امر باید در رابطه با نیازمندی‌های معین‌گردیده در EN 954 [EN954] درنظرگرفته‌شود، که نیازمند آن است که تحلیل‌گران به‌بازخوانی رفتار سیستم

---

1 Correlated Foalt Forecasting
2 dependability
3 Fault avoidance
4 presence
5 creation
6 consequences
7 Initiating source
8 correlation
9 Accidental faults
10 Qualitative evaluation

در بروز یک خطای ازپیش‌معین‌شده[1] بپردازند. این امر این پرسش را برمی‌انگیزد که تعیین خطاهایی که برای ارزیابی مورد نظر قرارمی‌گیرند، در پرسش از سیستم انجام‌می‌گیرد یا روی یک مدل؟ این مسئله برای مؤلفه‌های به‌کارگرفته‌شده‌ی پیچیده‌تر تبدیل به مسئله‌ای حاد و حساس می‌گردد.

درحقیقت، اگرچه خطاها در مؤلفه‌های گسسته یا کم‌تر مجتمع الکترونیک نیز رخ‌می‌دهند (در اتصال کوتاه، مدارهای باز، گیرکردن صفر یا یک) که در درجه‌ی رضایت‌بخش[2] شناخته‌شده‌اند، این مورد برای مؤلفه‌های نوع ریزپردازنده‌ی پیچیده و حتّا برای نرم‌افزار کم‌تر از این، ابداً وجود ندارد. در این موارد، اگر ما خواهان تحلیل سطح‌های مؤلّفه‌ها باشیم، فرضیاتی برمبنای شیوه‌های شکست در این مدارها شکل‌داده‌می‌شوند. این چنین فرضیاتی به‌زودی محدودیت‌های خود را آشکارمی‌کنند، و راه حل عموماً برای هم‌آهنگی باید در سطحی بالاتر رود و شکست‌ها را در سطح تابعی مورد بررسی قراردهد.
پاراگراف ۴-۳ چند روش مناسب، برای ارزیابیِ کیفیِ قابلیتِ اعتمادِ سیستم‌هایِ الکترونیکی که مؤلفه‌های پیچیده‌ای دارند، معرّفی‌می‌کند.

- **ارزیابی کمّی**

ارزیابی «معین»، در سطح یک قطعه‌ی مربوط به الکترومکانیک، مؤلفه‌های مجتمع کوچک یا گسسته که شیوه‌های شکست شناخته‌شده‌ای دارند، برای رسیدن به یک درجه‌ی رضایت‌بخش، افزایش safety را فراهم‌می‌کند. اما از طرف دیگر، توانایی‌های آن در مؤلفه‌های مجتمع بسیاربزرگ محدودتر نیز می‌شود، آن‌چنان که تعیین شیوه‌های شکستِ این مؤلفه‌ها با قطعیت و کامل، غیرممکن می‌شود.
دیگر راهبردها، باید با این مشکل رودررو شوند، همانند ارزیابی کمّی‌ای که در استاندارد IEC62061 آمده‌است، که خطاهای تصادفی سخت‌افزار را به‌شمارمی‌آورد، در یک توسعه‌ی مطمئن، خطاهای هم شیوه را نیز جواب‌گوست. چندین نتیجه می‌تواند به‌وسیله‌ی این نتایج تشریح‌شود: قابلیت اعتماد، دردسترس‌بودن، سلامت. با توجه به زمینه‌ای که این مقاله بر آن متمرکز است، که سلامتِ ماشین‌آلات می‌باشد، تنها محاسبات لازمبرای تعیین احتمال شکست‌های خطرناک انجام‌می‌شوند.

---
[1] predetermined
[2] Satisfactory degree

۴-۲ تحلیل‌های مقدم بر ارزیابی

قبل از هرگونه تحلیل در ارزیابی، خواه کیفی و خواه کمّی، برای هر نوع خطای مورد سؤال، یک سری آزمایشات درنظرگرفته شده، مقدّم‌اند. در اینجا مفهوم ریسک و هازارد و مراحل تحلیل از استاندارد IEC 61508 گرفته‌شده‌اند:

دانش سیستم

این مرحله که هنوز به‌واسطه‌ی Mosleh [MOSL] یا «مفهوم» (در استاندارد CEI 61508) شناخته‌می‌شود، تا کنون به پیش‌بینی خطا اختصاص‌نیافته‌است. این امر شامل به‌دست‌آوردن شناخت و فهم کلی درباره‌ی سیستمی که باید مورد تحلیل قرارگیرد، توابعی که باید برآورده‌شوند و محیط فیزیکی‌ای که باید نرم‌افزار در آن نموکند، می‌شود.

شروط محدودکننده‌ی متفاوت علی‌الخصوص بررسی‌می‌شوند: فیزیک سیستم و محدودیت‌های توابع، هم‌بستگی‌ها و واسط‌های عملیاتی با دیگر سیستم‌ها.

در مورد خطاهای وابسته، عناصر طراحی، عملیات، نگهداری و تست رویه‌ها، به‌طور مشابه فزونی‌می‌یابند تا چندین شکست مؤلّفه در این فاز مورد جست‌وجو قرارگیرند.

تحلیل هازارد و ریسک

ابتدائاً سطوح هازارد و رویدادهای به‌وجودآورنده‌ی آن‌ها باید تعیین گردند. این تحلیل‌ها باید در مورد همه‌ی حالت‌های قابل پیش‌بینی که شامل شکست‌ها و به‌کارگیری‌های ناصحیح هستند، انجام‌گیرد. ریسک با رویدادهای خطرناک معین‌شده مطابقت‌داده‌می‌شود و سپس تعیین‌می‌گردد. این مرحله، که مقدم بر هر گونه تحلیلی انجام‌می‌گیرد،به‌شدت به کاربردی که سیستم برای آن توسعه‌داده‌می‌شود، بستگی‌دارد.

روش‌هایی که به انجام این تحلیل‌ها کمک‌می‌کند در [1050] و [EN954] آمده‌است.

آماده‌سازی برای مدل‌سازی

به دلیل پیچیدگی مدل‌های امروزین، مدل‌سازی، مرحله‌ای لازم برای ارزیابی است. آماده‌سازی در این قسمت، شامل جمع‌آوری اطلاعات درباره‌ی:

- فرآیندها و تکنولوژی‌های پیاده‌سازی‌شده.
- رویه ها و تکرارهای تست‌ها.
- مدهای شکست و نرخ آن‌ها.
- نرخ خطایی که با تشخیص پوشش‌داده‌می‌شود.

- رویه‌های نگه‌داری و تعمیر.

در نمونه‌های معین، تحلیل‌گر می‌تواند برای تکمیل دانش خود روی موارد زیر تمرکز کند:

۱) نوع و کارکرد (استفاده‌کننده‌های) مؤلفه‌ها.

۲) شیوه‌ی استفاده‌ی مؤلفه‌ها

۳) شروط داخلی (محیطی) و خارجی مؤلفه‌ها.

۴) محدودیت‌های استفاده از واسط‌های سیستم و مؤلفه‌ها.

۵) مکان مؤلفه‌ها.

۶) حالت اولیه‌ی مؤلفه‌ها و خصوصیات عملیاتی‌شان.

اطلاعات به‌عنوان پایه‌ی هم‌بستگی، تهیه‌می‌شود و هم‌چنین برای تشخیص مفهوم گروه‌های مؤلفه‌های علّت‌معمولی[MOSH]. این اطلاعات اولیه با تمرکز بر ، برای مثال به‌وسیله‌ی تهیه‌ی لیست‌های مقابله، صفات معمول چندین مؤلفه و مکانیسم‌های شکست شبیه هم که به یک شکست هم‌شیوه منجرمی‌شوند، به‌دست‌می‌آیند. (درواقع، نقاط ضعف در رابطه‌با علت‌های مشترک یا پتانسیل‌های هم‌بستگی سیستم.) نشانه‌های زیر می‌تواند به تحلیل‌گر کمک‌کند:

- مؤلفه‌های همانند، فعال و از نظر کارکردی غیرگونه‌گون که برای حشو به‌کارمی‌روند، باید به عنوان گروه‌های علت‌مشترک فرض‌شوند.
- مؤلفه‌های گونه‌گون (توسعه‌یافته) که بخش‌های حشو و مکرّری دارند، را نمی‌توان مستقل محسوب‌کرد.

نکته:

آمادگی برای خطا داشتن گروهی از مؤلفه‌های علت‌مشترک تنها به درجه‌ی شباهت آن‌ها بستگی‌ندارد، بلکه به وجود و تأثیر عواملی که در برابر شیوه‌های مشترک آن‌ها را محافظت‌می‌کنند، نیز دارد.

به دلیل پیچیدگی مدل و زمان مورد نیاز برای تحلیل، انتخابی برای عدم بروز وقایع علت‌مشترک کم‌تأثیر می‌تواند صورت‌گیرد. این انتخاب می‌تواند برپایه‌ی ارزیابی کمّی (برای مثال درخت خطا) باشد.

۴-۳ ارزیابی کیفی

عموماً دو روش در این قسمت برای پیش‌بینی خطاهای هم‌شیوه به‌کارگرفته‌می‌شود[VILL].

۴-۳-۱ درخت‌های تحلیل خطا (FTA)

مروری بر روش

این روشِ استنتاجی[1]، کار را از یک شکستِ خطرناکِ سیستم می‌آغازد، که برای مثال به‌وسیله‌ی تحلیل‌های ریسک معین گردیده‌است، و ترکیبی از رویدادها که می‌تواند به این شکست منجرشود، را جست‌وجو می‌کند.

درخت خطا، تأثیرات شکست، بر یک مؤلفه یا بر یک سری از مؤلفه‌ها، در یک رویداد خطرناک را مدل‌می‌کند. رویدادهای اوّلیه به‌وسیله‌ی دروازه‌های and و or به‌یک‌دیگر متصل‌می‌گردند، تا مسیرهای گوناگونی را که منجر به شکست خطرناک می‌شوند، را نشان‌دهند. این کار خطاهای تصادفی، سیستماتیک و هم‌شیوه را آشکارمی‌سازد. لزوماً، همه‌ی شاخه‌های منطقی یک FTA باید تا رسیدن به رویدادهای پایه‌ای توسعه‌داده‌شوند. در عمل، درخت توسعه‌داده‌می‌شود تا قابلیت تحلیل تاثیر ورودی، پردازش و خروجیِ شکست را داشته‌باشد. برای اجتناب از هر خطای تفسیر و اطمینان از این‌که رویدادهای پایه به‌درستی پردازش‌شده‌اند، آنچه در پی‌می‌آید، باید بررسی‌شود[2]:

- رویدادهای خطرناک به‌حساب‌آیند[3].
- روال‌های شکست مؤلفه‌ها
- ترکیب‌های رویدادهای پایه
- مسیرهای منجربه‌شکست
- شکست‌های هم‌روال و شکست‌های سیستماتیک
- تقابل‌های ممکن

تحلیل‌های درخت خطا، درحال‌حاضر، در دامنه‌ی وسیعی[4]، در مورد جنبه‌های سخت‌افزاری، مورد استفاده‌قرارمی‌گیرند. هم‌چنین می‌توانند برای ارزیابی خطاهای نرم افزار به‌کارگرفته‌شوند، خصوصاً درجای‌که برای توابع یا پیمانه[5]های بحرانی جست‌وجو می‌شوند.

- **تعیین آغازگرها و هم‌بستگی‌ها[6]**

FTA مدلسازی ارتباط بین مؤلفه‌ها را فراهم‌می‌کند.[VILL]

آغازگرها برای بخش‌های آن، با دنبال‌کردن و توسعه‌ی شاخه‌ها تا رسیدن به ازهم‌پاشیدگی‌های مؤلفه‌ای[1]، اختلالات[2] خارجی یا خطاهای انسانی‌ای که دیده‌می‌شوند[3]، مشخص‌می‌گردند.

---

1 deductive
2 verify
3 take into account
4 widespread
5 module
6 correlations

- **تعیین پی‌آمدها**

ساختار استنتاجی روش درخت خطا، یک راه علت‌ومعلولی، برای جست‌وجوی پی‌آمدهای خطا ندارد. خواه ریشه‌ی آن علت مشترکی داشته‌باشد یا نه.

۴-۳-۲. مدهای شکست و تحلیل اثر

- **نظری کلی بر این روش**

این روش یک راهبرد استقرائی است که از توابع یا مؤلفه‌های سیستمی که باید مورد تحلیل قرارگیرد، کار را، (به‌منظور تعیین شکست‌های خطرناکی که می‌توانند آن‌ها را تحت‌تأثیرقراردهند)، آغازمی‌کند. شکست‌هایی را که به یک روال شکست، (که سخت‌افزار یا نرم‌افزار را تحت‌تأثیر قرارمی‌دهد)، منجرمی‌شوند، را برجسته‌می‌کند. وقتی شکست‌های توابع را مورد رسیدگی قرارمی‌دهیم، راهبردی که در ادامه می‌آید، به‌کارگرفته‌می‌شود:

- تعیین شیوه‌های شکست[4] برای هر تابعی که در تحلیل مد نظر قرارگرفته‌است.

این امر پس از آزمودن[5] مشخصه‌ی[6] این توابع، هدایت‌می‌شود؛ که غالباً بیش از سه کلاس از یک‌دیگر قابل تفکیک نیست: توابعی که نیازهای آن‌ها کاملاً برآورده[7]نشده‌است. توابعی که صحیح نیستند[8]. و توابعی که در آن‌ها نتیجه با آنچه مورد انتظار است متفاوت است.

[SERV] ۵ شیوه‌ی شکست را بدون بیان جزییات آن‌ها از یک دیگر تفکیک‌می‌کند و آن‌ها را درنظرمی‌گیرد.

- تعین تأثیرات محلی شکست‌ها. عموماً این مورد به بررسی جزءجزء حالت مقادیر گرفته‌شده به‌وسیله‌ی خروجی تابعِ روال‌های اولیه‌ی شکست[9]، محدودمی‌شود.
- تعیین تأثیرات سراسری شکست‌ها. تأثیرات محلی شکست که در سطح سیستم منتشرمی‌گردند در این مرحله مورد ارزیابی قرارمی‌گیرند.
- دسته‌بندی تأثیرات سراسری برطبق میزان بحرانی‌بودن[10] آن‌ها این دسته‌بندی با استفاده از نتایج تحلیل‌های ریسک و هازارد که پیش‌تر انجام‌گرفته، شکل‌می‌گیرد.

---

1 Component degradation
2 perturbation
3 reach
4 failure modes
5 examine
6 specification
7 fulfill
8 incorrect
9 Elementary failure modes
10 criticity

- تعیین شکست‌های خطرناک یک آستانه‌ی بحرانیت[1]، تعیین‌می‌گردد. شکست‌هایی که ورای این آستانه قرارمی‌گیرند، خطرناک تلقی‌می‌گردند.

FMEA در سیستم‌های سخت‌افزاری کاربرد وسیعی دارد. این تکنیک می‌تواند برای تحلیل خطاهای نرم‌افزار نیز به‌کاررود.[THIR]

- **تعیین آغازگرها و هم‌بسته‌ها**

FMEA روشی مستقیم برای تعیین آغازگرها و هم‌بستگی‌ها نیست. به‌منظور این‌که به شکل مؤثری برای برجستگی شکست‌های هم‌روال از آن بهره‌گیری[2]شود، باید آن‌را در پیوند[3] با دیگر روش‌ها، که توانایی تعیین آغازگرها و هم‌بستگی‌های بین مؤلفه‌ها و زیر-سیستم‌ها را دارند، به‌کارگرفت.

هورتول[HOUR]، فی‌المثل، پیشنهاد ترکیب MADE را با FAULT TREE برای تعیین توابع بحرانی نرم‌افزار را می‌دهد. دانش این توابع ما را قادرمی‌سازد، که بر زمینه‌ی بررسی برای تشخیص خطاهای هم‌شیوه‌ی نرم‌افزار، تمرکز بیش‌تری صورت‌دهیم.

- **تعیین پی‌آمدها**

فراهم‌آوردن خطاهای هم‌بسته، به درستی نشان‌می‌دهد که، FMEA روشی است که می‌تواند برای جست‌وجوی پی‌آمدهای این خطاها به‌کاررود.

۴-۴. ارزیابی کمّی

نشان‌دادن شیوه‌های معمول[4] در ارزیابی کمّی بسته به دانش تحلیل‌گر از سیستم بسیار متغیر است:

- اگر هم‌بستگی به وضوح مشخص‌شده‌باشد (محیط، خطاهای انسانی)، این موارد می‌توانند، فی‌المثل در درخت خطای سیستم مدلسازی شوند.
- چنان‌چه هم‌بستگی در سیستم‌های مشابه درشمارآمده‌باشد، قیاسی برای انجام کار کفایت‌می‌کند.
- در مواردی که علل ریشه ای به سختی قابل‌تشخیص‌اند، به وسیله‌ی مدل‌های پارامتریکِ مربوط به علل‌مشترک[5]، پردازش صورت‌می‌گیرد. پاراگراف‌های ۴-۴-۱ و ۴-۴-۲ به مورد سوم توجه‌دارند.

۴-۴-۱. مدل‌سازی خطاهای هم‌شیوه در سخت‌افزار

---

[1] Criticity threshold
[2] contribute
[3] associate
[4] Common modes
[5] Common cause

مدلی که بیش از همه کاربرد دارد، خصوصاً در مورد حشوهای تکی[1]، فاکتور β است.[ISA][61508][NUREG] یک مدل تک‌پارامتری برای تعیین علل مشترک مرتبط با سخت‌افزار. برای جاهایی که حشو درجات بالائی دارد، مدل‌های چندپارامتری استفاده‌می‌شوند.(multiple Greek letter، فاکتور α و نرخ خطای دوجمله‌ای). در قسمت باقیمانده‌ی این بخش، هر دو روش مدل‌سازی فاکتور β و multiple Greek letter را بررسی‌می‌کنیم.

فاکتور بتا

این مدل بسته به آن‌که کاربردش در کجا باشد، امکان دارد، یک یا چندین گروه از مؤلفه‌های علل مشترک را تفکیک‌کند[2]. برای یک گروه از سه مؤلفه A و B و C، نرخ خطای کلی[3] λی مؤلفه‌های این گروه، صریحاً برابر است با:

$$\lambda = \lambda_i + 2.\lambda^2 + \lambda_d$$

با:

-λi نرخ شکست تصادفی مستقل

- λ2 نرخ شکست‌هایی که با تأثیر دوبه‌دوی مؤلفه‌ها به‌وجودمی‌آید. (مثلاً AB و AC برای مؤلفه‌ی A). با این فرض که تأثیر مؤلفه‌ای دوگانه مشخص است.

- λd : نرخ شکست با علل ریشه‌ای مشترک سه‌گانه (۳ مؤلفه‌ای). (مثلاً در اینجا ABC)

فاکتور β فرض‌می‌کند که همه‌ی مؤلفه‌های یک گروه از مؤلفه‌های علل‌مشترک، failشده‌اند. این امر منجر به این می‌شود که که تنها بر شکست‌هایی که بر سه مؤلفه تذثیرگذارند، تمرکزشود. (λd) و آن‌هایی که تنها بر دو مؤلفه تأثیرگذارند، نادیده‌گرفته‌شوند و فاکتور λ2 نادیده‌گرفته‌شود. کسر β از نرخ شکست مؤلفه‌ها با رویدادهای علت‌مشترکِ دیگر مؤلفه‌های گروه هم‌آهنگ می‌گردد.و آنچه در پی‌می‌آید، حاصل‌می‌شود:

$$\lambda i = (1-\beta).\lambda$$
$$\lambda d = \beta.\lambda$$

از β در محاسبه‌ی نرخ شکست منطبق با خطاهای علت‌مشترک، استفاده‌می‌گردد.

---

[1] Single redundencies
[2] isolate
[3] Total failure rate

استاندارد IEC61508، روشی برای محاسبه‌ی نرخ شکست در رابطه با خطاهای علت‌مشترکی که در محاسبه‌اش از گراف‌های مارکوف استفاده‌می‌شود، را توصیف‌می‌کند.

این نرخ در تشخیص قابلیت‌های پیش‌نهادشده به وسیله ی ریزپردازنده به شمارمی‌آید، که منجر به شکستن فاکتور قبلی به دو قسمت جداگانه می‌شود:

β: اضافه‌شده به شکست‌های خطرناک ناشناخته.

β_D: شکست‌های خظرناک شناخته‌شده.

$$\lambda_{cmf} = \beta.\ \lambda_{DU} + \beta_D.\ \lambda_{DU}$$

λ_{DU} : نرخ شکست‌های خطرناک ناشناخته از یک کانل تنها.

λ_{DD} : نرخ شکست‌های خطرناک شناخته‌شده از یک کانال تنها.

فاکتورهای β و β_D با استفاده از تحقیقات میدانی تعیین‌می‌شوند:

- اندازه‌ی دفاعی که در برابر شکست‌های هم‌شیوه روی‌می‌دهد، به‌وسیله‌ی تشخیص این‌که کدام‌یک از قسمت‌ها با استفاده از تست‌های تشخیص خطا، improve شده‌اند.(X) از آن‌هایی که به‌وسیله‌ی تست‌های یکسان بهبودداده‌نشده‌اند. (Y) xi و yi انتساب به هر اندازه ست. این اندازه‌گیری در یک جدول مسیر است. مقادیر X و Y این‌گونه تعیین می‌شوند:

$$X = \Sigma x_i\ ,\ Y = \Sigma y_i$$

- آزمایش‌ها که برای تشخیص خطاها در یک کانال طراحی‌شده اند، در هر دو نوع تست‌های با قابلیت تشخیص و تست‌هایی که برای تکرار زاویه ی دید اجرامی‌شوند، با این مورد تعیین می‌شوند.

$$B = f(\ x+y)\ \ ,\ \ \beta_D = f(\ (z+1).x + y\ )$$

Z به‌وسیله ی جدول تعیین‌می‌شود. بسته به میزان پوشانندگی ابزارهای تشخیص خطا مقدار z برای کنترلمنطقی صفر است، که می‌دهد $\beta_D = \beta$. برای سنسورها و عامل‌ها تا جایی‌که زمان بین دو تست به بیش از یک هفته برسد، صفر است.

- نکته. کاربرد روش پیش‌نهادی برای تعیین و می‌تواند مورد بحث‌قرارگیرد.

می‌توان نتایج زیر را به‌دست‌آورد:

- برای ساختار حشو یکنواخت، اجرای تست تشخیص خطا، قانع‌کننده است. مثلاً یک تست اجراشده حداقل یک بار در دقیقه با پوشانندگی ۹۹٪ برآورنده ی فرمول زیر است:

$$\lambda^1_{CMF} = 0.02 * \lambda_{DU} + 0.01 * \lambda_{DD}$$

- برای یک ساختار حشو گونه‌گون، اجرای یک تست تشخیص خطای در حد استاندارد (برای مثال، تست کم‌تر از یک بار در یک دقیقه و با پوشانندگی ۶۰٪) اگر اجراشود کافی است. نتایج زیر به دست‌می‌آید:

$$\lambda^2_{CMF} = 0.02 * \lambda_{DU} + 0.02 * \lambda_{DD}$$

–λ_{DU} و λ_{DD} این‌گونه تعیین‌می‌شوند. با توجه به خطاهای هم‌شیوه‌ی تصادفی سخت‌افزار، تکرار گونه‌گون کم تر از تکرار یکنواخت اثرگذار است.

Multiple Greek Letter برای یک‌سری مؤلفه‌ی تعیین‌شده

این روش توسعه‌ای از فاکتور β است. که شامل تأثیرات محتمل یک مؤلفه بر روی دیگر مؤلفه‌های گروه علت‌مشترک یکسان است. در این حالت فاکتورهای مضرب $\lambda 2$ بزرگ‌تر از صفر نیستند. برای یک ساختار حشو سه‌تایی، پارامترهای مدل عبارتند از:

λ: احتمال شکست در رابطه با رویدادهای مستقل و علت‌مشترک.

β: احتمال شرطی این‌که علت مشترک شکست یک مؤلفه در ریشه‌ی مؤلفه‌ی دوم باشد.

γ: احتمال شرطی این‌که علت مشترک شکست یک مؤلفه در ریشه‌ی شکست دو مؤلفه‌ی دیگر باشد.

بنابراین داریم:

$$\lambda_i = (1-\beta) * \lambda$$
$$\lambda_2 = (1/2) * \beta * (1-\gamma) * \lambda$$
$$\lambda_D = \beta * \gamma * \lambda$$

۴-۴-۲. مدل‌سازی خطاهای هم‌شیوه در نرم‌افزار

مقاله‌ای که اخیرا توسط لیتل‌وود [LITT] منتشرشده، نشان‌می‌دهد که در مقایسه با خطاهای تصادفی سخت‌افزار، خطاهای نرم‌افزار به این آسانی نمی‌توانند موضوع مدل‌سازی قرارگیرند:

«...»[LITT]

این نگرش با فرضیه‌های نگران‌کننده‌ی مستقل انجام‌شده برای مدل‌سازی قابلیت اعتماد یک ساختار چندنسخه‌ای، پذیرفته‌می‌شود[LYU]. در قیاس با خطاهای سخت‌افزاری، تکنیک‌های مدل‌سازی

خطاهای هم‌شیوه در نرم‌افزار، به قدر لازم رشد و توسعه نداشته‌اند، تا بتوانند به عنوان روش ارزیابی احتمال شکست در سیستم خطرخیز، معرفی‌شوند.

۴-۴-۳. ارزیابی کمّی با استفاده از گراف‌های مارکوف

ارزیابی کمی با استفاده از گراف‌های مارکوف ابتدا به مدل‌سازی عملیات سیستم در یک فرم گرافی می‌پردازد. شکست‌های هم‌شیوه با اضافه‌کردن انتقال‌ها به این گراف درشمارمی‌آیند. احتمال بروز شکست‌های خطرناک با استفاده از محاسبات ماتریسی از نرخ‌های شکست تعیین‌می‌گردند. این کار با استفاده از فرمول‌های داده‌شده در بخش ۴-۴-۱ انجام‌می‌گیرد.[ISA]{61508][DEACBr][BOUS].

۴-۴-۴. ارزیابی کمّی با استفاده از درخت‌های تحلیل خطا

درخت‌های خطا به‌آسانی قابل کمّی‌سازی‌اند، اگر رویدادهای اولیه مستقل‌باشند. در این موارد، امکان تبدیل درخت به عبارت بولی، با استفاده از قوانین زیر وجوددارد:

Basic event → Boolean variable
AND gate → Boolean product
OR gate → Boolean sum

شرط استقلال رویدادهای اولیه، رفتار خاصی را در مورد شکست‌های علت‌مشترک طلب‌می‌کند[ISA]. مثالی که در ادامه‌می‌آید نحوه‌ی طرح این شکست‌ها را نشان‌می‌دهد:

مثال:

رویداد E داده‌شده‌است، که از ترکیب در پیمانه‌ی ورودی وابسته‌ی (A1,A2)، و دو پیمانه‌ی خروجی وابسته‌ی دیگر (B1,B2) که در عین حال (A1,A2) از (B1,B2) مستقل است، به‌وجودآمده‌است. عبارت

شکل ۷: درخت خطا بدون CMF

بولی E عبارت‌است‌از:

$$\mathbf{E = A1 + A2 + B1 + B2}$$

که به احتمالات زیر منجرمی‌شود:

$$P(E) = P(A1+A2+B1+B2)$$
$$P(E) = P(A1) + P(A2) + P(B1) + P(B2)$$
$$- P(A1.A2) - P(A1.B1) - P(A1.B2) - P(A2.B1) - P(A2.B2) - P(B1.B2)$$
$$+ P(A1.A2.B1) + P(A1.A2.B2) + P(A1.B1.B2) + P(A2.B1.B2)$$
$$- P(A1.A2.B1.B2)$$

از آنجا که گفته‌شده پیمانه‌های ورودی و خروجی مستقل از یک‌دیگرند، عبارت احتمالی بالا می‌تواند به عبارت زیر ساده‌شود:

$$P(E) = P(A1) + P(A2) + P(B1) + P(B2) - P(A1.A2) - P(B1.B2)$$

در این‌جا $P(A1.A2)$ نشان‌گر قسمت مشترک A1 و A2 است و می‌تواند به رویداد «هم‌شیوه» تعبیرشود A_{CMF} (همان‌گونه که برای مورد B1,B2 می‌تواند.)

درعمل، این امکان وجوددارد که معادله‌ی قبل را به صورت فرم درختی و با طرح A_{CMF} که A1 و A2 را تبدیل به دو رویداد مستقل‌می‌کند (این مورد برای پیمانه‌های خروجی نیز صحیح است.) در آوریم. بنابراین درخت به صورت زیر درمی‌آید:

شکل۸: درخت خطا به‌همراه CMF

اگر شکست‌های هم‌شیوه بین (A1,A2) و (B1,B2) را با فاکتورهای β_A,β_B مشخص‌سازیم، نرخ خطا منطبق با رویدادهای ابتدایی درخت چنین خواهدبود:

$$(1-\beta_A).\lambda_A \ \& \ (1-\beta_B).\lambda_B \text{ for } A_1 \text{ and } A_2 \ \& \text{ for } B_1 \text{ and } B_2$$
$$\beta_A.\lambda_A \ \& \ \beta_B.\lambda_B \text{ for } A_{CMF} \ \& \ B_{CMF}$$

۴-۵. مرور کلّی[1]

آن‌گونه که تشریح‌شد، پیش‌گوئی خطاهای به‌هم‌وابسته، تخمین بروز و خلق خطاها و پی‌آمدهای آن‌هاست.

جدول ۲ مروری بر نتایج اصلی مرتبط با پیش‌بینی خطاها در توانایی‌های هر روش در جنبه‌های «خلق‌وبروز» و «پی‌آمدها» ارائه‌می‌دهد. امکان ارزیابی‌های کمّی و کیفی در شیوه‌های معمول خطا، به وسیله‌ی درخت خطا، آن را روشی خوب برای پیش‌بینی خطاهای هم‌شیوه نشان‌می‌دهد. این توانایی باید به‌وسیله‌ی کاربردهای مختلف در موراد واقعی[2] اثبات‌گردد.

---

[1] overview
[2] concrete

۵. لیست‌های مقابله[1]

۶. نتیجه‌گیری

مراجع

[1] Check lists