Pisa Dependable Computing Center

Ente Per le Nuove Tecnologie, l'Energia e l'Ambiente

Integrazione di Tecniche di Fault-Forecasting

Contributo PDCC nell'ambito del Terzo Obiettivo Intermedio del Contratto di Ricerca del 30/10/97

15 Dicembre, 1998

Andrea Bondavalli, Silvano Chiaradonna, Ivan Mura

Technical Note
PDCC-TN-0003 A

Integrazione di Tecniche di Fault-Forecasting

Andrea Bondavalli[*], Silvano Chiaradonna[*], Ivan Mura[**]

** PDCC/CNUCE-CNR; CNUCE Istituto del CNR, Via S. Maria 36, 56126 Pisa, Italy
S.Chiaradonna@guest.cnuce.cnr.it, a.bondavalli@cnuce.cnr.it,

* PDCC/DII; Dipartimento di Ingegneria Informatica, Univerista' di Pisa, via Diotisalvi 2
56127 PISA, Italy; mura@iet.unipi.it

1 Tecniche di Fault-Forecasting

Le tecniche di fault-forecasting forniscono mezzi per aumentare il livello di confidenza che può essere riposto nella capacità del sistema di fornire un servizio conforme alle sue specifiche [28]. A tal fine, il fault-forecasting punta a stimare la presenza, la creazione e le possibili conseguenze dei guasti o degli errori sulla dependability, e permette, come conseguenza, di eliminare gli stessi o di prevedere mezzi per tollerarli.

I metodi per il fault-forecasting possono essere divisi in due gruppi: i) metodi deterministici o qualitativi, il cui scopo é capire come i malfunzionamenti dei componenti possono risultare nel malfunzionamento del sistema e quali possono essere le possibili conseguenze, e ii) metodi probabilistici o quantitativi, il cui obiettivo é la stima degli attributi della dependability del sistema o delle sue componenti. É però importante osservare che i metodi e le tecniche di analisi possono essere utilizzati per entrambi le forme di valutazione. Ad esempio, i fault-tree sono tipicamente adottati come mezzo per la valutazione probabilistica della Reliability o della Safety del sistema, sebbene costituiscano un tipico metodo qualitativo per il fault-forecasting.

Nel valutare gli attributi della dependability di un sistema realisticamente complesso, nessuna delle metodologie di fault-forecasting é di per se sufficiente a garantire una stima affidabile ed adeguatamente accurata. Questa incompletezza deve essere superata utilizzando in modo combinato le varie tecniche, per sfruttare al meglio le specifiche capacità di ognuna di esse.

Nel seguito presenteremo brevemente le idee base per alcuni metodi qualitativi, per passare poi a descrivere in dettaglio quelli quantitativi basati sulla valutazione probabilistica degli attributi della dependability. Nel corso dell'esposizione metteremo l'accento sulla necessaria complementarità tra i vari approcci al fault-forecasting, in particolare per quelli quantitativi, evidenziando i punti più delicati dell'analisi, ovvero quelli per i quali una accurata valutazione della dependability del sistema deve fare ricorso non unicamente ad una sola tecnica, ma ad una combinazione di

diverse tecniche. Infine, i legami e la complementarità tra le varie forme di fault-forecasting saranno esplicitamente identificate.

2 Metodi qualitativi

I metodi qualitativi per il fault-forecasting possono essere suddivisi in due grandi famiglie. La prima comprende metodi che, procedendo dalle cause verso gli effetti puntano a determinare le conseguenze al livello del sistema dei malfunzionamenti dei componenti. Rientrano in questa famiglia le FMEA (Failure Mode Effect Analysis). La seconda contiene i metodi che procedendo lungo il cammino inverso dagli effetti verso le cause puntano a identificare le cause dei fallimenti del sistema al livello dei componenti. I fault-tree rappresentano un tipico esempio di tali metodi.

2.1 FMEA

La FMEA é una tecnica induttiva il cui principio fondamentale é l'analisi, per ogni componente, degli errori che vi si possono produrre, per identificare in maniera sistematica l'insieme dei modi di fallimento del componente, come pure le conseguenze dei fallimenti a livello del sistema [42].
La FMEA fa generalmente riferimento alla struttura funzionale del sistema, e permette di metterne a fuoco i punti deboli per quanto riguarda la dependability. Questa tecnica può essere utilizzata in linea di principio in un qualsiasi istante del ciclo di vita del sistema, ma é preferibile applicarla il più precocemente possibile, per minimizzare i costi di eventuali modifiche.
Il principio generale d'applicazione della tecnica consiste, dopo avere elencato i diversi modi di fallimento sulla base della descrizione funzionale o strutturale del sistema, nel dettagliare in una tabella per ogni modo di fallimento di ciascun componente le seguenti informazioni:

- le sue possibili cause;
- il suo effetto, che può riguardare solamente il componente stesso o può propagare ad altri componenti fino a giungere al livello del sistema;
- i metodi che possono essere utilizzati per il rilevamento (detection) del fallimento;
- le azioni correttive da mettere in atto, in particolare quando si tratti di un modo di fallimento tra quelli identificati come particolarmente pericolosi per il sistema.

Quando il metodo delle FMEA viene applicato iterativamente su un sotto-componente, é possibile mettere in luce le omissioni di alcuni modi di fallimento per il componente di cui esso fa parte. In effetti, i modi di fallimento del componente del livello superiore appaiono come le combinazioni dei modi di fallimento dei suoi sotto-componenti. In particolare, i modi di fallimento dei sotto-componenti aventi effetti globali devono necessariamente corrispondere a dei modi di fallimento del componente del livello superiore.
Le tabelle risultanti dall'applicazione della FMEA sono utili per la progettazione, poiché permettono di guidare alcune scelte e di identificare (e modificare) il più presto possibile alcuni punti

deboli. Esse forniscono inoltre un supporto prezioso per la validazione del sistema anche con altri metodi di fault-forecasting, poiché mettono in luce i punti critici che dovranno essere accuratamente testati, o modellati, e premettono di associare una causa, ovvero il modo di fallimento di un componente elementare, a un effetto complesso quanto un fallimento osservabile del sistema.
Ad ogni modo, occorre mettere in chiaro che l'approccio basato su FMEA soffre di tre notevoli limiti:

- i modi di fallimento sono studiati l'uno dopo l'altro (per evitare l'esplosione combinatoria dell'analisi), e questo non permette di prendere in considerazione i malfunzionamenti multipli;
- non si può avere la certezza che l'analisi sia completa, nel senso che potrebbe non avere tenuto in considerazione tutti i modi di fallimento dei componenti del sistema;
- il volume di informazione da trattare può risultare di notevoli dimensioni, soprattutto nel caso di sistemi complessi (che sono d'atro canto proprio quelli di interesse).

2.2 Fault-tree

I fault-tree sono una tecnica di analisi deduttiva che permette di ricercare le combinazioni di eventi che possono condurre ad un evento indesiderato, tipicamente il fallimento catastrofico del sistema [7].
I fault-tree sono comunemente utilizzati come complemento alla FMEA, in quanto permettono di prendere in considerazione quelle combinazioni di malfunzionamenti che non sono considerate con la FMEA. I fault-tree possono inoltre essere utilizzati più facilmente durante le prime fasi del processo di sviluppo di un sistema, in quanto si prestano a rappresentare ad un livello di astrazione elevato i vari scenari di occorrenza di un evento che può compromettere la dependability del sistema.
Un fault-tree é costituito da livelli successivi di eventi connessi da delle porte che rappresentano gli operatori logici AND e OR. Ogni evento all'uscita di una data porta é ottenuto dalla combinazione degli eventi situati in ingresso alla porta, combinati secondo la funzione logica espressa dalla porta stessa. L'evento indesiderato oggetto dell'analisi costituisce la radice dell'albero. Il principio di costruzione di un fault-tree consiste nel decomporre ogni evento incontrato, partendo dalla radice dell'albero, fino ad arrivare a degli eventi giudicati elementare e pertanto non ulteriormente decomposti. Un evento può essere giudicato elementare sia perché é indipendente da altri eventi, sia perché la sua probabilità di occorrenza può essere stimata, sia più semplicemente perché non si desidera (o non si può') decomporlo ulteriormente.
L'analisi di un fault-tree é basata sul calcolo dei tagli minimali (min-cuts). Un taglio é definito come un insieme di eventi elementari che in accordo alla logica espressa dall'albero permettono il realizzarsi dell'evento indesiderato. Un taglio é detto minimale se non é contenuto in nessun altro taglio del fault-tree. Lo studio dei tagli minimali permette di identificare le combinazioni di

eventi elementari che possono condurre all'evento indesiderabile. In particolare, é possibile scoprire se il sistema possiede dei "single-point of failure", ovvero dei min-cuts costituiti da un singolo elemento.
I fault-tree possono essere utilizzati per valutazioni quantitative degli attributi della dependability in modo molto semplice, associando delle probabilità di occorrenza agli eventi basilari, per ottenere alla fine la probabilità di occorrenza dell'evento indesiderato. I valori di tali probabilità possono essere ottenuti tramite altre tecniche di fault forecasting, come il testing o la modellizzazione.

3 Metodi quantitativi

I metodi quantitativi sono stati ampiamente studiati nella letteratura, e numerose tecniche ben collaudate sono state sviluppate e utilizzate con successo. In funzione del livello d'astrazione considerato per il sistema da analizzare, possono essere utilizzate tecniche analitiche (o assiomatiche), e sperimentali.
I metodi analitici comprendono principalmente metodi basati sui fault-tree (che abbiamo già visto in precedenza) e metodi basati su rappresentazioni dello spazio degli stati del sistema da analizzare, come le catene di Markov, le reti di Petri o le reti di code.
Tra i metodi sperimentali consideriamo il fault injection ed il testing, che offrono due approcci complementari alla modellizzazione assiomatica, particolarmente utili per la stima degli attributi degli attributi della dependability dei componenti hardware e del software, rispettivamente.

3.1 Modellizzazione

La modellizzazione si basa sulla descrizione analitica (tramite equazioni), o grafica (tramite diagrammi), del comportamento del sistema. Le misure sugli attributi della dependability sono ottenute in funzione dei parametri del modello, che tipicamente includono distribuzioni di probabilità per rappresentare l'aleatorietà dei fenomeni connessi ai malfunzionamenti. Tra i principali tipi di modelli utilizzati, possiamo citare i fault-tree, e i metodi basati su rappresentazioni dello stato del sistema, come i processi Markoviani e le reti di Petri stocastiche. L'utilizzazione di queste metodologie di modellizzazione é stata da lungo riconosciuta come un fattore determinante per le scelte di progetto dei sistemi, si vedano per esempio gli studi relativi ai sistemi SIFT, FTMP es ESS in [40]. Un grande numero di tools sono stati sviluppati negli ultimi venti anni al fine di assistere i progettisti dei sistemi nel compito di modellare e stimare la dependability dei sistemi, una rassegna si puo' trovare in [21].
Gli studi sulla valutazione degli attributi della dependability tramite modelli sono stati largamente dominati dall'ipotesi che i componenti del sistema si trovino ad operare in uno stato di affidabilità stazionaria, corrispondente a tassi di fallimento costanti. Una tale ipotesi é giustificata per i componenti hardware, in quanto in caso di guasto fisico i componenti difettosi sono in genere rimpiazzati con componenti identici dal punto di vista dell'affidabilità. Questa ipotesi

non é in genere ritenuta adeguata per i malfunzionamenti dovuti a errori nel progetto, come ad esempio quelli che si trovano nei componenti software, in quanto la rimozione (correzione nel caso specifico) di un errore porta ad una crescita dell'affidabilità del componente. Altre tecniche di fault forecasting sono utilizzate per stimare la crescita dell'affidabilità del software, tipicamente basate sul ciclo testing-fault-removal-testing dei programmi.
Il fault-forecasting basato su modellizzazione del sistema può essere decomposto in tre fasi strettamente connesse tra loro che riguardano:

- la scelta delle misure da valutare, che fa generalmente parte dei requisiti del sistema;
- la costruzione di uno o più modelli, che corrispondono alla descrizione del comportamento del sistema studiato a partire dai processi stocastici elementari e in funzione delle misure selezionate;
- il trattamento del modello o dei modelli, che corrisponde al calcolo delle misure degli attributi di dependability.

La scelta della misura da valutare dipende considerevolmente dal dominio d'applicazione del sistema considerato. Ad esempio, l'Availability é una misura di interesse per sistemi di telecomunicazione, la Reliability per una sonda spaziale, la Safety per il sistema di controllo a bordo di un sistema di trasporto, etc, etc. Al fine di identificare le misure di dependability di interesse, il comportamento di un sistema informatico può essere schematicamente descritto in rapporto a due tipi di servizio fornito: servizio proprio, ovvero conforme alle specifiche di funzionamento, e servizio improprio, difforme dalle specifiche di funzionamento. Le transizioni tra questi due tipi di servizio sono governate dai processi di fallimento (da servizio proprio a servizio improprio), e di recovery (da servizio improprio a servizio proprio) del servizio.
Le misure principali di dependability puntano a caratterizzare i tempi di durata del servizio proprio. A seconda che si consideri il solo processi di fallimento o entrambi i processi d fallimento e di recovery, le misure possono essere classificate in due categorie:

- le misure che caratterizzano il tempo di permanenza continuato nello stato in cui il sistema fornisce servizio proprio (prima dell'assorbimento nello stato di servizio improprio): ad esempio, rientrano in questa classe la Reliability ed il Mean Time To Failure.
- le misure che permettono di caratterizzare la fornitura di servizio in rapporto all'alternanza servizio proprio-servizio improprio: ad esempio rientrano in questa seconda classe diverse forme dell'Availability (istantanea, asintotica o su di un intervallo).

I sistemi informatici attuali sono per la maggior parte capaci di fornire diverse modalità di servizio. A seconda del punto di vista considerato per valutare gli attributi della dependability, due grandi categorie di casi estremi possono essere identificate, in accordo alle quali un sistema presenta:

- diversi modi di fornire un servizio proprio, ed un solo modo di fornire servizio improprio;
- diversi modi di fornire un servizio improprio, ed un solo modo di fornire servizio proprio.

Esempi caratteristici di sistemi capaci di fornire più modi di servizio proprio, ognuno caratterizzato da un dato livello di performance, sono i sistemi multiprocessore capaci di assicurare un

graceful degrading. Tali sistemi sono stati oggetto di numerosi studi destinata alla definizione e la valutazione di misure che generalizzano le misure classiche della dependability associandovi tipiche misure di performance. Queste misure combinate sono note sotto il nome generale di misure di performability [8, 32].

Grazie alla loro potenzialità modellistiche, in particolare rispetto alle dipendenze stocastiche tra le componenti di un sistema, le catene di Markov omogenee sono divenute il tipo di modello preponderante per l'analisi degli attributi della dependability, soppiantando altri metodi che erano in auge in passato, quali i fault-tree. Le catene di Markov omogenee inducono delle ipotesi particolari per quanto riguarda le distribuzioni dei processi stocastici considerati (distribuzioni esponenziali). Come conseguenza, i parametri dei modelli, come i tassi di fallimento, sono considerati costanti nel tempo. Benché restrittiva, questa ipotesi é largamente riconosciuta come valida per i fallimenti imputabili a guasti dei componenti hardware. Diversi studi sono stati condotti per estendere la classe di distribuzioni che possono essere considerate. Oltre la simulazione, diverse soluzioni sono state proposte per modellare eventi non esponenziali: l'utilizzo dei processi semi-Markoviani , delle catene di Markov non-omogenee, o ancora il metodo degli stadi fittizi. Una rassegna di queste tecniche é presentata in [26].

Il principale problema nella costruzione di una catena di Markov che rappresenti il comportamento del sistema complesso, é costituito dalla crescita esponenziale del numero degli stati. Diverse forme di descrizione testuale o grafica sono state sviluppate per rappresentare in modo conciso catene di Markov di grandi dimensioni: algebre di Kronecker [3], reti di Petri stocastiche [1], regole di produzione come nel tool METFAC [12], diagrammi PERT [38] e tanti altri ancora.

Le reti di Petri e le loro numerose estensioni appaiono ormai come l'approccio di portata generale per la descrizione di sistemi complessi, e sono integrate in numerosi tool di valutazione automatica della dependability, quali UltraSAN [39], SPNP [17], Surf-2 [9]. In effetti, esse permettono di dedurre direttamente la catena di Markov associata a partire dalla generazione del grafo di raggiungibilità, di esprimere facilmente le dipendenze stocastiche tra i componenti, e di verificare i modelli rispetto alle loro proprietà strutturali, come l'assenza di deadlock.

Un aspetto particolarmente delicato per la modellizzazione é rappresentato dalla distinzione tra i processi relativi a:

- i malfunzionamenti dei componenti, vale a dire l'occorrenza dei guasti nel sistema (e eventualmente al recovery del servizio)
- il trattamento dei guasti/errori eseguito dai meccanismi di tolleranza ai guasti del sistema.

L'esperienza, e numerosi studi teorici, hanno dimostrato che l'efficacia, più precisamente la copertura [6, 11], dei meccanismi di tolleranza ai guasti, ha una influenza preponderante sulla dependability ed in particolare sulle misure della dependability abitualmente tenute in considerazione per stimare il livello di sicurezza effettivamente raggiunto dal sistema. Come conseguenza, uno sforzo notevole é stato intrapreso nel corso di questi ultimi anni per modellare la copertura imperfetta fornita dai meccanismi di tolleranza ai guasti. Nonostante questo sforzo, la

modellizzazione della copertura delle tecniche di fault tolerance é limitata, sia per la complessità dei meccanismi che intervengono nella tolleranza ai guasti, sia per l'imperfezione delle tecniche di fault diagnosis e fault treatment. Inoltre, il problema della quantificazione di questi parametri non può essere risolto, come per i parametri relativi ai processi di guasto o di riparazione, a partire dai dati statistici disponibili sui componenti del sistema o di sistemi analoghi, ma deve essere trattato caso per caso, in relazione al particolare sistema in esame.
Per la risoluzione dei modelli sono stati sviluppati numerosi algoritmi ormai ben collaudati. Due diversi tipi di problemi possono complicare questo compito:

- le grandi dimensioni dei modelli da trattare;
- il fenomeno della stiffness che caratterizza questi modelli.

L'esplosione del numero degli stati interviene quando si devono manipolare rappresentazioni molto compatte di processi stocastici di notevoli dimensioni, come nel casi di modelli a rete di Petri, in cui anche un modello con pochi posti e transizioni é in grado di rappresentare un numero infinito di stati di una catena Markoviana. Il fenomeno della stiffness é invece legato alla differenza di ordine di grandezza esistente tra i parametri che descrivono i processi di occorrenza dei guasti da un lato, e i meccanismi di tolleranza ai guasti dall'altra; questa differenza può seriamente penalizzare la fase di soluzione dei modelli, sia per quanto riguarda la durata che l'accuratezza dei risultati forniti, in particolare per valutazioni di transienti di lunga durata.
Le analisi di sensitività delle misure valutate rispetto ai parametri dei modelli rappresentano un parte rilevante della valutazione. Oltre all'impatto già citato della copertura ai guasti, gli studi di sensitività possono essere condotti al variare dei tassi di fallimento [22], ed anche al variare delle distribuzioni dei processi stocastici, come in [25].

3.2 Fault Injection

A dispetto dei continui progressi registrati dagli sforzi per la modellizzazione dei sistemi, esistono dei limiti riguardo alle possibilità offerte dall'approccio analitico. Quindi, uno sforzo sperimentale specifico rimane necessario per testare il comportamento, in presenza dei guasti, dei meccanismi di fault tolerance dei sistemi studiati. La fault injection costituisce un approccio privilegiato per misurare i parametri caratteristici dei meccanismi di trattamento dei guasti e degli errori, e, a questo titolo, le esperienze di fault injection contribuiscono alla stima dell'efficacia della copertura della tolleranza ai guasti e sono dunque di fatto complementari alle tecniche analitiche viste in precedenza.
Quale che sia il livello di astrazione del sistema, i lavori sulla fault injection puntano in generale alla fault-forecasting, o studiando la propagazione degli errori [14, 16, 41], o la latenza degli errori [15, 20], o ancora la copertura dei meccanismi di fault tolerance [37, 43].
Diverse tecniche e metodi sono stati sviluppati ed integrati in tools specifici per la fault injection [4, 13, 23]. Due tipi di criteri devono essere considerati per caratterizzare gli studi di fault injection destinati a validare un sistema tollerante ai guasti: il livello di astrazione del sistema e la

forma di applicazione dell'injection. Per quanto riguarda il livello di astrazione, si distingue il caso in cui il sistema sotto esame é:

- un sistema fisicamente disponibile (eventualmente un prototipo);
- un modello di simulazione che descrive la struttura e/o il comportamento del sistema.

Per quanto riguarda la forma di applicazione dell'injection, distinguiamo tra:

- injection fisica, quando i guasti sono introdotti direttamente sui componenti fisici tramite alterazioni meccaniche o elettromagnetiche;
- injection logica, quando vengono alterati i valori di variabili Booleane o i dati contenuti nelle memorie.

La maggior parte degli studi accorpa questi due criteri in base alla forma di applicazione, e si suole cosí più semplicemente distinguere l'injection fisica e la simulazione dei guasti.
La maggior parte degli studi sull'injection fisica utilizza l'injection al livello dei pin dei circuiti integrati. Questo tipo di injection corrisponde all'applicazione di svariate combinazioni d'errore destinate a simulare le conseguenze dei guasti che possono effettivamente occorrere durante la vita operazionale dei componenti. Un tipo di injection più realistica é possibile per una classe di guasti di interesse in alcuni tipi specifici di sistemi, come quelli per applicazioni spaziali: l'irraggiamento ed il bombardamento dei componenti con ioni pesanti possono simulare in modo accurato l'ambiente ostile in cui tali sistemi si trovano ad operare. A parte i problemi di rappresentatività dell'injection sui pin dei circuiti integrati, si vengono sempre più ponendo dei problemi di accessibilità dei componenti stessi, a causa sia dei livelli di integrazione sempre più spinta che dell'accelerazione dei cicli di clock. La ricerca di soluzioni a questi problemi ha portato ad adottare sempre più di tecniche di injection ibride, che inseriscono i guasti al livello logico del componente fisico: si parla perciò in questi casi di emulazione dei guasti.
La simulazione dei guasti é ampiamente utilizzata come tecnica per l'elaborazione dei vettori di test in ambito industriale, per esempio per il controllo di qualità della produzione dei circuiti integrati. Attualmente esiste un numero piuttosto ristretto di esempi di valutazione dei meccanismi per la tolleranza ai guasti al livello dei modelli di simulazione.
Come abbiamo già citato in precedenza, le tecniche di fault injection si prestano in modo eccellente a stimare la copertura delle tecniche di fault tolerance, formalmente definita come:

- *la probabilità condizionale che, dato che un guasto é presente nel sistema, il sistema (o la tecnica di fault tolerance a questo preposta) riesca a tollerarlo*.

É importante sottolineare che la nozione di coverage si applica anche all'errore, in particolare per caratterizzare l'efficacia delle differenti tappe del trattamento dell'errore. Un'altra precisazione essenziale riguarda l'aspetto dinamico del processo di trattamento dei guasti/errori (latenza), che porta ad una definizione di coverage sotto forma di una funzione di ripartizione della variabile aleatoria che rappresenta l'istante al quale occorre il trattamento atteso . Tenuto conto che non tutti i guasti possono essere coperti, questa funzione é difettiva, nel senso che il suo valore asintotico é inferiore all'unità [18]. Il valore asintotico del coverage corrisponde di fatto al fattore di copertura comunemente utilizzato nei modelli analitici per la valutazione della

dependability. Nonostante numerosi studi (sia basati sulla modellizzazione che sulla fault injection) si siano occupati di entrambi gli aspetti (fattore di copertura e latenza), pochi sono quelli che hanno cercato di integrarli trattandoli in modo unificato, si vedano per esmpio [5, 18, 31].
La copertura imperfetta della tolleranza ai guasti é imputabile a due grandi tipi di imperfezioni:

- errata definizione/applicazione dei meccanismi di fault tolerance in rapporto alle ipotesi di guasto considerate, che si traduce in una inadeguata copertura dei meccanismi di trattamento degli errori e dei guasti;
- assunzione di ipotesi di guasto diverse da quelle che possono effettivamente essere tenute in conto in fase operativa, che risulta in una mancanza di copertura delle ipotesi di guasto.

La fault injection é particolarmente adatta a rivelare le imperfezioni del primo tra i due tipi sopra elencati. Altre forme di analisi sono invece necessarie per stimare la copertura delle ipotesi di guasto. Nel seguito faremo perciò focalizzeremo la nostra attenzione sulla valutazione della prima componente della copertura.
Tenuto conto dell'impossibilità in generale di percorrere in modo esaustivo lo spazio delle coppie guasto/funzionalità che possono riguardare il sistema, i risultati della fault injection non possono che essere parziali. In effetti i risultati sono tipicamente espressi nel modo seguente:

- *la percentuale x% dei guasti inseriti quando il sistema esegue la funzionalità y risulta in un errore detectato*.

É quindi necessario ricorrere alle tecniche statistiche per valutare il coverage a partire dai risultati d'un esperimento di fault injection. Due obiettivi da tenere in particolare considerazione per questa valutazione sono:

1) minimizzare l'errore della stima per la parte testata del sistema;
2) estendere il risultato della stima alla parte non testata.

Questi due obiettivi sono generali e si applicano ad un gran numero di esperienze di fault injection. Per quanto riguarda il punto 1), anche se si restringe la portata della stima alla parte testata del sistema, il rischio di errore può provenire:

- dalla non rappresentatività della funzionalità del sistema (e quindi degli errori provocati);
- delle esperienze non significative riguardanti i guasti inseriti e non tradottisi in errori;
- dal fatto che non tutti i guasti hanno la stessa probabilità di occorrenza.

Il rischio legato al diverso peso delle funzionalità del sistema può essere parzialmente risolto con una opportuna politica di inserimento dei guasti. Il secondo punto é invece molto più delicato, perché necessita di identificare quali guasti non danno luogo ad errori per poterli eliminare dalle statistiche. Non é sempre facile capire perché un guasto non si traduca in errore: in effetti questo può avvenire perché il guasto rimane dormiente, o perché esso viene attivato localmente ma non si propaga, o perché esso é stato effettivamente tollerato grazie ai meccanismi di fault tolerance. Alcune tecniche specifiche sono state sviluppate per la comprensione di questi possibili eventi, ed incorporate in vari tool per la fault injection, come ad esempio in RIFLE [30]. Il terzo punto mette in luce il fatto che é necessario tenere in considerazione la probabilità di occorrenza di ciascun guasto nel calcolo del coverage: in effetti, se non si é potuto tenere in conto

di queste probabilità in fase di pianificazione dell'esperimento al momento della scelta di guasti da inserire, il semplice conteggio della errori detectati in rapporto a quelli introdotti può portare a delle stime errate del coverage.
Un prezioso aiuto per la stima delle probabilità di occorrenza dei diversi scenari di guasto può essere fornito da altre tecniche di fault-forecasting, tipicamente quelle di modellizzazione analitica e di testing.
Infine, per quanto riguarda l'obiettivo 2) relativo all'utilizzo dello stimatore del coverage per la parte testata al fine di inferire il coverage del sistema completo, è importante osservare che gli intervalli di confidenza che è possibile stimare per il coverage del sistema completo sono in generale troppo ampi per presentare un interesse pratico. Tuttavia, in alcuni casi particolari, l'estensione della portata del test può essere considerevolmente migliorata a posteriori tenendo in considerazione informazioni supplementari sulla struttura del sistema. Ad esempio negli studi di fault injection fisica sui pin dei circuiti integrati, le informazioni strutturali relative al cablaggio dei circuiti possono essere vantaggiosamente utilizzate a questo scopo [37].

3.3 Testing

Le tecniche di testing rappresentano un approccio sperimentale alla fault forecasting particolarmente adeguato per l'analisi del software [19]. Esistono pochi modi per potere dare garanzie sull'affidabilità prodotti software, e si basano tutte sull'uso di metodi formali durante il ciclo di sviluppo dei prodotti. L'approccio formale non é pero praticabile per software di grandi dimensioni a causa della sua inerente complessità.
Un modo alternativo per misurare l'affidabilità del software é osservare in modo diretto il suo comportamento durante il normale esercizio, ed inferire da queste osservazioni le misure sull'affidabilità [10, 29, 33, 35]. Per un sistema critico dal punto di vista della dependability, occorre simulare l'uso del software prima di metterlo realmente in esercizio, in modo da ottenere a priori una valutazione della sua bontà. Si parla in questo caso di testing operazionale o statistico, più brevemente testing. É importante osservare che per sistema ultra-dependable, il testing può diventare una operazione alquanto complessa. I livelli di affidabilità richiesta possono essere talmente elevati da che mostrare il loro raggiungimento é pressoché impossibile. Il problema deriva dalla scarsezza di informazioni in base al quale il giudizio deve essere emesso, in rapporto ai requisiti estremamente stringenti.
Gli input da utilizzare per il test (profilo del test) dovranno essere scelti in modo da fornire un realistico carico di lavoro per il software sotto esame, ovvero le probabilità di scelta per il testing devono riflettere le probabilità di occorrenza degli input durante la reale vita operativa [34]. La difficoltà di questa scelta dipende dalla particolare applicazione: ad esempio può essere relativamente semplice riprodurre le condizioni operative di un software del sistema di controllo di un aereo, mentre é arduo identificare a priori i possibili comportamenti di un utente di un sistema transazionale.

Ovviamente, il testing può anche essere utilizzato per migliorare l'affidabilità del software, come strumento di fault removal. In questo caso, la scelta del profilo di test mirerà a riprodurre le condizioni più critiche per il sistema, per massimizzare la probabilità di attivazione dei guasti.
La tipica strategia seguita per il testing considera le seguenti entità [27]:

- programma: un generico componente (tipicamente software) del sistema da analizzare;
- oracolo: un agente umano ma tipicamente automatico (un altro programma), a cui é delegato il compito di emettere il giudizio approvato/rifiutato sulla base dell'input del programma e dell'output del test.

Questa strategia é in genere automatizzata per permettere l'esecuzione di un numero molto elevato di test. Appare subito evidente come l'oracolo giochi un ruolo basilare nel testing. Poiché esso é l'unico responsabile della classificazione dei risultati del test, ed in definitiva della stima dell'affidabilità, occorre garantire che l'oracolo sia esso stesso affidabile nel suo giudizio. Formalmente definiamo il coverage dell'oracolo come segue:

- il coverage di un oracolo é la probabilità che l'oracolo riconosca come tale il risultato errato di un test.

Anche il problema dei falsi allarmi, ovvero dei risultati corretti erroneamente rifiutati deve essere tenuto in considerazione. Ogni falso allarme richiede necessariamente l'intervento umano per riconoscere che l'output del test era effettivamente corretto, e può cosí portare a un notevole spreco di tempo se la procedura era stata automatizzata. Un oracolo può essere reso più efficiente se lo si rende capace di osservare anche lo stato interno de programma. In questo modo, eventuali risultati intermedi erronei della computazione possono essere riconosciuti come tali, anche se il risultato finale non lo fosse. Alcuni metodi sono stati definiti per tener in considerazioine le imperfezioni dell'oracolo nella statistica finale sull'affidabilità del programma, si veda ad esempio [2].
La stima degli attributi della dependability a partire dai risultati ottenuti con il testing é un problema di inferenza statistica, esattamente come già visto per le tecniche di fault injection. Le tecniche classiche di inferenza si basano sulla derivazione di intervalli di confidenza per la misura di interesse [35]. Per la natura estremamente sfuggente degli eventi di interesse, un gran numero di esperimenti di fault injection é in genere necessario per restringere in modo adeguato l'ampiezza di tali intervalli. Un diversa metodologia per la valutazione é quella basata sull'approccio Bayesiano [29, 33]. Le tecniche Bayesiane producono stime sulle probabilità degli eventi di interesse, basandosi su di un procedimento di correzione iterativo di una stima iniziale, senza generare intervalli di confidenza.

4 Legami tra le varie tecniche di fault-forecasting

Nel corso della rassegna presentata nei precedenti paragrafi abbiamo messo in luce diversi punti di contatto tra le varie tecniche di fault-forecasting, punti nei quali le informazioni richieste per la stima accurata degli attributi della dependability con una data metodologia possono essere

ottenute applicando una diversa tecnica di analisi. In particolare, esistono forti legami tra le tecniche quantitative, il cui uso combinato é sicuramente necessario per la validazione di sistemi complessi costituiti da molteplici elementi hardware e software, e che utilizzano tecniche di tolleranza ai guasti.

Due componenti essenziali sono da considerare nella costruzione di un modello per la valutazione della dependability. Esse sono rispettivamente:

- i processi relativi all'occorrenza dei guasti nel sistema (e eventualmente al loro trattamento);
- le azioni legate al trattamento dei guasti/errori da parte dei meccanismi di tolleranza ai guasti.

A queste due può essere aggiunta la modellizzazione dei processi di normale attività del sistema (erogazione del servizio proprio), in particolare nei casi in cui é necessario stimare misure combinate di performance e dependability.

Sul piano concettuale, si può considerare la valutazione analitica come una forma di astrazione estrema della fault injection: i guasti sono introdotti nel modello assiomatico di modellizzazione tramite i tassi di fallimento, mentre nella fault injection propriamente detta i guasti sono degli stimoli applicati o a dei modelli di simulazione o a dei modelli fisici. Quindi, mentre i modelli assiomatici permettono effettivamente di valutare le misure di dependability essendo essi capaci di tenere in conto sia i processi di guasto che quelli relativi alla fault tolerance, le esperienze di fault injection permettono unicamente di valutare misure condizionali poiché esse caratterizzano solamente i processi di propagazione degli errori ed il loro trattamento da parte dei meccanismi di tolleranza ai guasti.

D'altronde,come già osservato, nei modelli assiomatici è essenziale sia tenere in considerazione che valutare accuratamente i parametri della fault tolerance, e questa valutazione si basa sui risultati degli esperimenti di fault injection. La valutazione degli attributi della dependability è dunque basata su di un approccio iterativo che deve strettamente legare valutazione analitica e sperimentale.

Analogamente, le tecniche basate su modelli analitici possono essere considerate un'astrazione del testing rispetto alla modellizzazione dei processi di normale attività del sistema. Nel modello assiomatico, il carico di lavoro é introdotto nel sistema tramite opportune variabili aleatorie, ed il fallimento del test é rappresentato da un evento che modella la fine dell'a fornitura di servizio proprio. Anche in questo caso, i parametri relativi all'affidabilità dei componenti (in particolare componenti software) sono da ottenere con altri metodi, meno astratti, come modelli più dettagliati o tecniche di testing.

I diagrammi nelle seguente figure descrivono le principali tappe e relazioni che caratterizzano i vari approcci analitici e sperimentali alla valutazione quantitativa, e quelli qualitativi.

Consideriamo ad esempio le relazioni tra modellizzazione analitica e fault injection, graficamente raffigurate in Figura 1. Ognuno dei due approcci può essere utilizzato indipendente come esposto nelle precedenti sezioni, in un ciclo di modellizzazione, valutazione, modifica del si-

stema, modellizzazione e cosí via. Due interazioni principali tra i metodi assiomatici e sperimentali possono essere identificate sul diagramma. Queste interazioni sono rappresentate dalle frecce a tratto discontinuo.

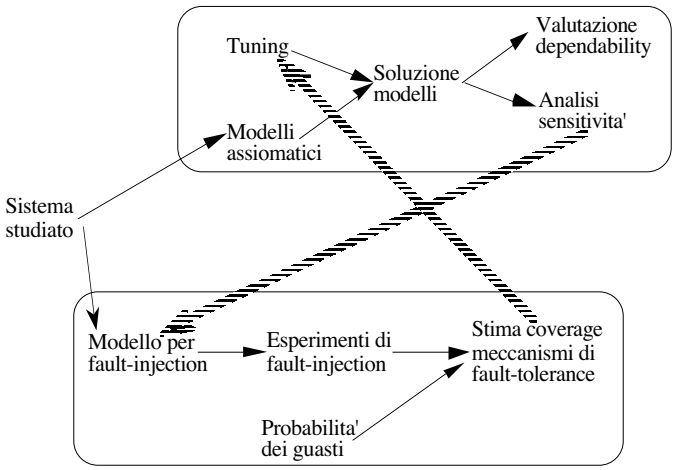

Figura 1: Relazioni tra modellizzazione analitica e fault injection

L'approccio sperimentale di fault injection permette di ottenere delle misure sperimentali caratteristiche dei meccanismi di tolleranza ai guasti. La transizione tra "analisi di sensitività"e "modello per fault injection" illustra l'impatto che la valutazione dei modelli analitici hanno sulla specifica dei test di fault injection, in particolare per ciò che riguarda i tipi di guasti da iniettare e le osservazioni da effettuare. Se un parametro si rivela particolarmente critico per la valutazione della dependability, approfondite analisi dovranno essere condotte per una sua accurata stima.

A complemento di questa interazione, ne esiste un'altra rappresentata dalla freccia da "stima coverage" a tuning, che rappresenta l'impatto dell'approccio sperimentale sull'approccio analitico per quanto riguarda il tuning dei parametri della copertura dei modelli assiomatici iniziali quindi il raffinamento successivo di questi modelli.

Ancora, altro tipo di interazione é possibile con l'utilizzo della modellizzazione assiomatica dei componenti a basso livello per la stima delle probabilità di occorrenza dei guasti, in modo da ottenere informazioni accurate per l'analisi statistica dei risultati della fault injection.

Le relazioni tra modellizzazione assiomatica e testing sono simili a quelle viste per il fault injection. In particolare, come illustrato in Figura 2, i risultati del testing costituiscono una sorgente di informazione sui valori dei parametri usati per istanziare i modelli assiomatici, e le analisi di sensitività che su questi ultimi possono essere condotte aiutano a capire quali parti del sistema devono essere sottoposte ad un test più accurato. La definizione dei profili di test può quindi trarre vantaggio dai risultati della modellizzazione assiomatica.

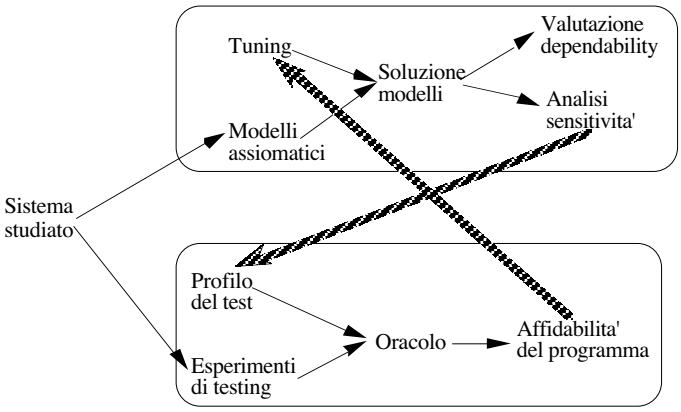

Figura 2: Relazioni tra modellizzazione analitica e testing

Anche le tecniche qualitative possono fornire informazioni utili per altre tecniche di fault-forecasting. Ad esempio, come mostrato in Figura 3, la FMEA può essere usata per capire quali parti del sistema devono essere studiate più approfonditamente, poiché i loro malfunzionamenti hanno gravi conseguenze sulla capacità del sistema di fornire il servizio proprio. Queste parti critiche del sistema possono quindi essere oggetto di studi realizzati con le tecniche quantitative, per valutare numericamente il rischio relativo.

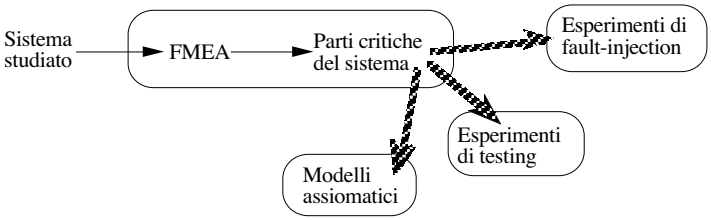

Figura 3: Relazioni tra FMEA e le tecniche quantitative

Un altro tipo di uso combinato di diverse tecniche quantitative permette di ottenere le probabilità di occorrenza degli eventi elementari che costituiscono le foglie dei fault-tree, come mostrato in Figura 4.

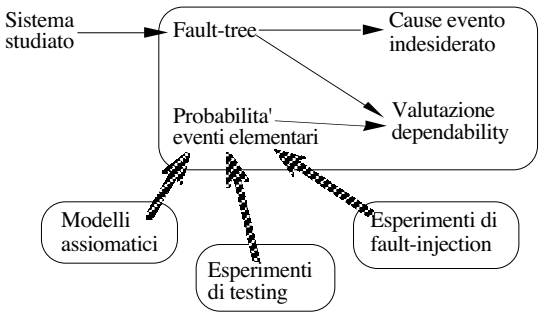

Figura 4: Uso combinato di tecniche quantitative

In questo modo, la stima delle misure di dependability può essere effettuata in modo gerarchico, utilizzando le tecniche più raffinate e costose per analizzare in dettaglio i componenti elementari dell'albero, e combinando i risultati cosí ottenuti con il fault-tree per ottenere le misure finali.

Sebbene questa necessaria complementarità tra i vari tipi di tecniche per il fault forecasting sia ormai assodata e largamente riconosciuta per lo meno sul piano concettuale, esistono pochissimi lavori in letteratura che l'hanno effettivamente applicata per la valutazione di sistemi reali (un esempio è dato da [40]). Un tale approccio alla valutazione combinata tramite modellizzazione analitica e fault injection è stato il punto centrale della validazione dell'architettura distribuita tollerante ai guasti del progetto ESPRIT Delta-4 [24]. Più recentemente, nell'ambito del progetto ESPRIT 20716 Generic Upgradable Architecture for Real-Time Dependable Systems (GUARDS), é stato proposto un approccio di validazione e design che integra le tecniche di modellizzazione analitica e di fault injection, come é mostrato nella Figura 5, tratta da [36].

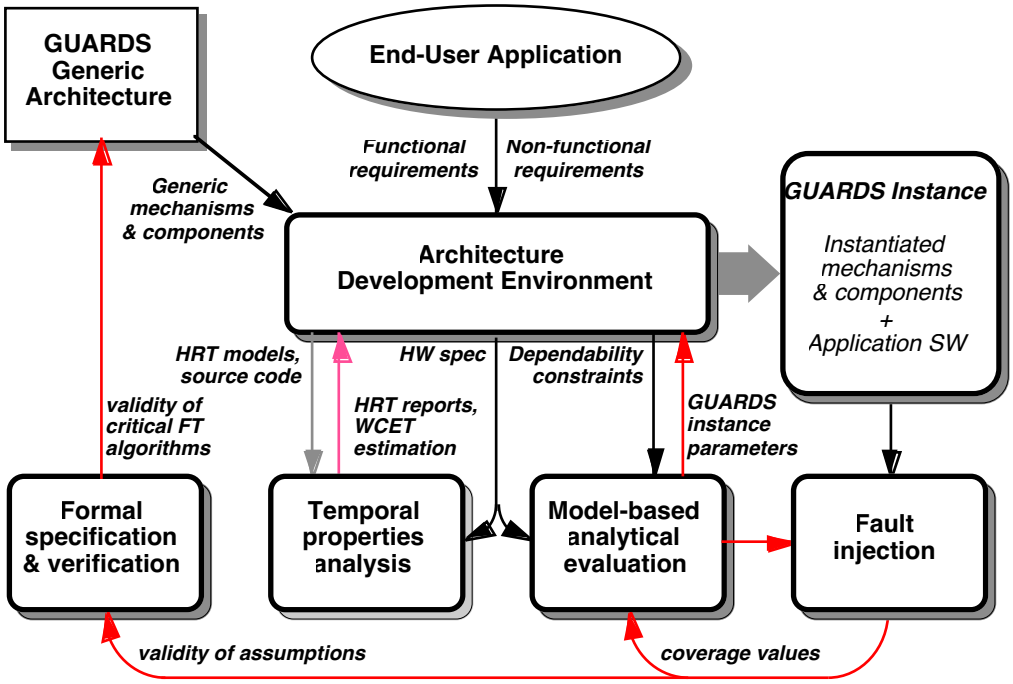

Figura 5: Interazioni tra strategie di validazione e di design architetturale nel progetto ESPRIT GUARDS

L'architettura GUARDS punta sulla riconfigurabilità e sulla modularità per fornire soluzioni adattabili alle specifiche applicazioni. Oltre alle tecniche di valutazione basate su modelli analitici e su fault injection, anche i metodi formali sono utilizzati come mezzo per la validazione dell'architettura. La fault injection é allora utilizzata per a) verificare la validità delle assunzioni necessarie per l'applicazione dei metodi formali e b) per stimare i fattori di copertura inclusi nei modelli analitici per la valutazione della dependability.

Bibliografia

[1] Ajmone Marsan M., Balbo G. and Conte G., "*A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems*" ACM TOCS, 1984. Vol. 2 (2): pp. 93-122.

[2] Amman P. E., Brilliant S. S. and Knight J., "*The effect of imperfect error detection on reliability assessment via life testing"* IEEE Transactions on Software Engineering, 1994. Vol. 20 (2): pp. 142-148.

[3] Amoia V., Micheli G. D. and Santomauro M., "*Computer-oriented formulation of transition rate matrices via Kronecker algebra"* IEEE Transactions on Reliability, 1981. Vol. 30 (2): pp. 123-132.

[4] Arlat J., *"Fault injection for the experimehntal validation of fault-tolerant systems"*, in Workshop on Fault-Tolerant Systems, 1992, Kyoto, Japan.

[5] Arlat J., Costes A., Crouzet Y., Laprie J. C. and Powell D., "*Fault injection and dependability evaluation of fault-tolerant systems"* IEEE Transactions on Computers, 1993. Vol. 42 (8): pp. 913-923.

[6] Arnold T. F., "*The concept of coverage and its effects on the reliability model of repairable systems"* IEEE Transactions on Reliability, 1973. Vol. 22 (3): pp. 251-254.

[7] Barlow R. E., Fussel J. B. and Singpurwalla N. D., *Reliability and Fault-Tree Analysis*. 1975, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

[8] Beaudry M. D., "*Performance-Related Reliability Measures for Computing Systems"* IEEE Transactions on Computers, 1978. Vol. 27 (6): pp. 540-547.

[9] Beounes C., Aguera M., Arlat J., Bourdeau C., Doucet J. E., Kanoun K., Laprie J. C., Metge S., Moreira de Souza J., Powell D. and Spiesser P., *"SURF-2: a program for dependability evaluation of complex hardware and software systems"*, in 23-rd International Symposium on Fault-Tolerant Computing (FTCS-23), 1993, Toulouse, France: IEEE Computer Society Press.

[10] Bertolino A. and Strigini L., "*On the Use of Testability Measures for Dependability Assessment"* IEEE Transactions on Software Engineering, 1996. Vol. 22 (2).

[11] Bouricius W. G., Carter W. C. and Schneider P. R., *"Reliability modeling techniques for self-repairing computer systems"*, in 24-th National Conference, 1969, ACM Press.

[12] Carrasco J. A. and Figueras J., *"METFAC: design and implementation of a software tool for modeling and evaluation of complex fault-tolerant computing systems"*, in 16-th International Symposium on Fault-Tolerant Computing (FTCS-8), 1986, Vienna, Austria: IEEE Computer Society Press.

[13] Carreira J., Madeira H. and Silva J. G., *"Xception: Software Fault Injection and Monitoring in Processor Functional Unit"*, in Fifth International Working Conference on Dependable Computing for Critical Applications, 1995, Urbana-Champaign, U.S.A.

[14] Chillarege R. and Bowen N. S., *"Understanding large system failures - a fault injection experiment"*, in 19-th International Symposium on Fault-Tolerant Computing (FTCS-22), 1989, Chicago, Il, USA: IEEE Computer Society Press.

[15] Chillarege R. and Iyer R. K., "*Measurement-based analysis of error latency"* IEEE Transactions on Computers, 1987. Vol. 36 529-537.

[16] Choi G. S. and Iyer R. K., "*FOCUS: an experimental environment for fault sensitivity analysis"* IEEE Transactions on Computers, 1992. Vol. 41 (12): pp. 1515-1526.

[17] Ciardo G., Muppala J. and Trivedi K. S., *"SPNP: stochastic Petri net package"*, in International Conference on Petri Nets and Performance Models, 1989, Kyoto, Japan.

[18] Dugan J. B. and Trivedi K. S., "*Coverage modeling for dependability analysis of fault-tolerant systems"* IEEE Transactions on Computers, 1989. Vol. 38 (6): pp. 775-787.

[19] Frankl P., Hamlet D., Littlewood B. and Strigini L., *"Choosing a testing method to deliver reliability"*, in 19-th International Conference on Software Engineering (ICSE97), 1977.

[20] Geist R., Smotherman M. and Talley R., *"Modeling recovery time distributions in ultrareliable fault-tolerant systems"*, in 20-th International Symposium on Fault-Tolerant

Computing (FTCS-20), 1990, Newcastle-Upon-Tyne, UK: IEEE Computer Society Press.

[21] Geist R. M. and Trivedi K. S., "*Reliability estimation of fault-tolerant systems: tools and techniques"* IEEE Computer, 1990. Vol. 23 (7): pp. 52-61.

[22] Iyer R. K., "*Reliability evaluation of fault-tolerant systems: effect of variability in failure rates"* IEEE Transactions on Computers, 1984. Vol. 33 (2): pp. 197-200.

[23] Kanawati G. A., Kanawati N. A. and Abraham J. A., "*FERRARI: a flexible software-based fault and error injection system"* IEEE Transactions on Computers, 1995. Vol. 44 (2): pp. 248-260.

[24] Kanoun K., Arlat J., Burril L., Crouzet Y., Graf S., Martins E., MacInnes A., Powell D., Richier J. L. and Voiron J., "Validation", in Dealta-4: a Generic Architecture for Dependable Distributed Computing, Powell D., Editor. 1991, Springer-Verlag: Berlin, Germany. pp. 371-406.

[25] Kuhen P. J., "*Approximate analysis of general queuing networks by decomposition"* IEEE Transactions on Communications, 1979. Vol. 27 (1): pp. 113-126.

[26] Laprie J. C., *"Trustable evalution of computer systems dependability"*, in Int. Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems, 1984, Pisa, Italy: North-Holland.

[27] Laprie J. C., ed. *Dependability: Basic Concepts and Associated Terminology*. Dependable Computing and Fault-Tolerant Systems, ed. Laprie J.C. 1991.

[28] Laprie J. C., "Dependability-Its Attribues, Impairments and Means", in Predictably Dependable Computing Systems, Randell B., Laprie J.C., Kopetz H.andLittlewood B., Editor. 1995, Springer-Verlag: pp. 3-24.

[29] Littlewood B. and Strigini L., "*Validation of ultra-high dependability for software-based systems"* Communications ACM, 1993. Vol. 36 (11): pp. 69-80.

[30] Madeira H., Furtado P. and Joao Silva J. G., *RIFLE: a general purpose faul injector system*. 1991, University of Coimbra, Coimbra Portugal.

[31] McGough J., Smotherman M. and Trivedi K. S., "*The conservativeness of reliability estimates based on instantaneous coverage"* IEEE Transactions on Computers, 1985. Vol. 34 (7): pp. 602-609.

[32] Meyer J. F., "*On Evaluating the Performability of Degradable Computing Systems"* IEEE Transactions on Computers, 1980. Vol. C-29 (8): pp. 720-731.

[33] Miller K. W., Morrel J. W., Noonan R. E., Park S. K., Nicol D. M., Murril B. W. and Voas J. M., "*Estimating the probability of failure when testing reveals no failures"* IEEE Transactions on Software Engineering, 1992. Vol. 18 (1): pp. 33-44.

[34] Musa J. D., "*Operational profiles in software reliability engineering"* IEEE Software, 1993. Vol. (Mar.): pp. 14-32.

[35] Parnas D. L., van Scouwen A. J. and Kwan S. P., "*Evaluation of safety-critical software"* Communications ACM, 1990. Vol. 33 (6): pp. 636-648.

[36] Powell D., Arlat J., Beus-Dukic L., Bondavalli A., Coppola P., Fantechi A., Jenn E., Rabejac C. and Wellings A., "*GUARDS: a Generic Upgradable Architecture for Real-Time Dependable Systems"* Submitted to: Transactions on Computers, 1998.

[37] Powell D., Martins E., Arlat J. and Crouzet Y., *"Estimators for fault tolerance coverage evaluation"*, in 23-th International Symposium on Fault-Tolerant Computing (FTCS-23), 1993, Toulouse, France: IEEE Computer Society Press.

[38] Sahner R. A. and Trivedi K. S., "*Reliability modeling using SHARPE"* IEEE Transactions on Reliability, 1987. Vol. 36 (2): pp. 186-193.

[39] Sanders W. H., Obal II W. D., Qureshi M. A. and Widjanarko F. K., "*The UltraSAN modeling environment"* Performance Evaluation, 1995. Vol. 21 (Special Issue "Performance Evaluation Tools").

[40] Siewiorek D. P. and Swarz R. S., *Reliable Computer Systems*. 1992, Bedford, MA, USA: Digital Press.

[41] Steininger A. and Schweinzer H., *"A model for the analysis of the fault-injection process"*, in 25-th International Symposium in Fault-Tolerant Computing (FTCS-25), 1995, Pasadena CA, USA: IEEE Computer Society Press.

[42] Villemeur A., *Reliability, Availability, Maintanability and Safety Assessment*. Vol. Vol1, methods and techniques. 1991, Wiley.

[43] Walter C. J., "*Evaluation and design of an ultrareliable distributed architecture for fault-tolerance"* IEEE Transactions on Reliability, 1990. Vol. 39 (4): pp. 492-499.