Curriculum vitae et studiorum

Sandro Morasca

July 27, 2009

1 Notizie generali

Nato a Como il 23 luglio 1960, ha conseguito il Diploma di Maturità Classica presso il Liceo Classico Statale "A. Volta" di Como nell'anno scolastico 1978-1979 con la votazione di 60/60.

Si è laureato in Ingegneria Elettronica a indirizzo Sistemi-Organizzativo presso il Politecnico di Milano il 19 luglio 1985 con la votazione di 100/100 e lode.

Ha conseguito l'abilitazione all'esercizio della professione di ingegnere nella prima sessione dell'anno 1986 con la votazione di 100/100.

Ha conseguito il titolo di dottore di ricerca in Ingegneria Elettronica dell'Informazione e dei Sistemi (3 ciclo) il 23 settembre 1991.

Ha trascorso due anni presso l'Università del Maryland a College Park, in qualità di Faculty Research Assistant dal 1991 al 1993.

Ha prestato servizio presso il Politecnico di Milano-Facoltà di Ingegneria di Como in qualità di ricercatore dal 1993 al 1998.

Ha prestato servizio presso il Politecnico di Milano-Facoltà di Ingegneria di Como in qualità di professore associato dal 1998 al 2000.

È stato Visiting Professor presso l'Università del Maryland a College Park dall'ottobre 1999 al febbraio 2000.

È attualmente professore ordinario presso l'Università degli Studi dell'Insubria - Facoltà di Scienze Matematiche, Fisiche e Chimiche di Como.

2 Attività didattiche

2.1 Collaborazione a corsi

- Corso "Macchine per l'elaborazione dell'informazione" tenuto dal prof. Renato Stefanelli presso il Politecnico di Milano (anno accademico 1986-1987).
- Corso "Programmazione dei calcolatori elettronici" tenuto dal prof. Pierluigi Della Vigna presso il Politecnico di Milano (anni accademici 1987-1988, 1988-1989 e 1989-1990).
- Corso "Introduzione agli algoritmi ed alla programmazione" tenuto dal prof. Pierluigi Della Vigna presso la Scuola a Fini Speciali del Politecnico di Milano Diretta in Informatica (Gestionale) in Como (anni accademici 1987-1988, 1988-1989 e 1989-1990).
- Corso "Ingegneria del Software" tenuto dal prof. Carlo Ghezzi presso il Politecnico di Milano (anno accademico 1989-1990).
- Corso "Elementi di Informatica" tenuto dal prof. Nello Scarabottolo presso il Politecnico di Milano-Sede di Como (anno accademico 1990-1991).
- Corso "Fondamenti di Informatica" tenuto dal prof. Della Vigna presso il Politecnico di Milano-Sede di Como (anni accademici 1993-1994, 1994-1995 e 1995-1996).

- Corso "Fondamenti di Informatica 2" tenuto dal prof. Pozzi presso il Politecnico di Milano-Sede di Como (anno accademico 1997-1998).

2.2 Titolarità di corsi

- Corso "Fondamenti di Informatica" presso la Facoltà di Ingegneria di Como del Politecnico di Milano (dall'anno accademico 1996-1997 all'anno accademico 1999-2000).
- Corso "Informatica A (per gestionali)" (10 crediti) presso la Facoltà di Ingegneria di Como del Politecnico di Milano (anno accademico 2000-2001).
- Corso "Cultura Tecnologica del Progetto" (5 crediti) presso la Facoltà di Disegno Industriale di Milano Bovisa-Como del Politecnico di Milano (anno accademico 2000-2001).
- Corso "Laboratorio di Programmazione I" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Universit degli Studi dell'Insubria (anno accademico 2000-2001).
- Corso "Programmazione e Laboratorio di Programmazione" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Universit degli Studi dell'Insubria (anno accademico 2001-2002).
- Corso "Progettazione di Software" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Università degli Studi dell'Insubria (anno accademico 2001-2002).
- Corso "Progettazione del Software I con Laboratorio" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Università degli Studi dell'Insubria (dall' anno accademico 2001-2002 ad oggi).
- Corso "Progettazione del Software II con Laboratorio" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Università degli Studi dell'Insubria (dall' anno accademico 2001-2002 ad oggi).
- Corso "Ingegneria del Software I con Laboratorio" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Università degli Studi dell'Insubria (dall' anno accademico 2001-2002 ad oggi).
- Corso "Ingegneria del Software II con Laboratorio" presso la Facoltà di Scienze Matematiche, Fisiche e Naturali di Como dell'Università degli Studi dell'Insubria (dall' anno accademico 2001-2002 ad oggi).

2.3 Altri corsi

Ha preso parte a Corsi di Istruzione Permanente presso il Dipartimento di Elettronica e Informazione del Politecnico di Milano.

Ha tenuto corsi di Dottorato presso il Dipartimento di Elettronica e Informazione del Politecnico di Milano.

A partire dal 1988 ha tenuto corsi presso importanti società nazionali, quali ENIDATA (Milano), Alenia (Roma), SIG (Palermo), SSGRR (L'Aquila), Telettra (Milano), Engineering (Roma), Elsag Bailey (Genova), Systech (Milano), ASL di Como e presso istituti di istruzione e ricerca nazionali e internazionali, quali l'Università di Catania, l'Università di Buenos Aires (Argentina), l'Università di San Luis Potosì (Messico), la Scuola Universitaria Professionale della Svizzera Italiana (SUPSI) di Manno (Svizzera) e l'Universidad de Castilla-La Mancha di Ciudad Real (Spagna).

2.4 Redazione di libri didattici

È coautore del libro "Ingegneria del software - Progettazione, sviluppo e verifica" edito da Mondadori [24].

È coautore del libro "Fondamenti di Informatica - Dal problema al programma" edito da ETAS libri [25].

3 Attività di ricerca

3.1 Studi empirici nell'ingegneria del software

Una panoramica e un'analisi del settore della misura nell'Ingegneria del Software sono contenuti in [26].

3.1.1 Studi di caso basati su misure e modelli di qualità esistenti

Tale attività ha avuto inizio durante la stesura della tesi di laurea [87], in cui è stata effettuata una valutazione critica delle teorie esistenti e poi una verifica sperimentale della loro utilità tramite la loro applicazione al linguaggio Pascal, con la proposta di varie strategie di conteggio dei vari elementi del linguaggio. I risultati di tale prima ricerca sono stati pubblicati in [29].

Nell'ambito del progetto PROTAGORA è stata studiata la produttività in un ambiente applicativo reale nella Pubblica Amministrazione [52] tramite l'uso del paradigma GQM. La produttività è stata studiata dal punto di vista funzionale (in termini di punti funzione) ed interno (dimensione del prodotto) e sono stati messi in luce alcuni fattori che influenzano l'uno o l'altro punto di vista in modo da poter aiutare i manager a prendere decisioni rispetto allo sviluppo di applicazioni.

Inoltre, è stata studiata la correlazione tra misure del software e la probabilità di avere errori in moduli software. Gli studi riportati in [53, 20] mostrano la possibilità di predire la predisposizione all'errore sulla base di misure statiche del software, con la costruzione di modelli di regressione logistica e l'uso di cross-validation su un noto benchmark per il testing.

Nell'ambito del progetto europeo CEMP sono stati studiati gli aspetti relativi all'introduzione di misure in un ambiente industriale. In particolare, sono stati studiati due progetti, il primo in fase di sviluppo [13] e il secondo [11, 86] in fase di manutenzione. Tale attività ha permesso di identificare e quantificare i fattori che influenzano maggiormente l'ambiente applicativo studiato. Inoltre, si sono tratte indicazioni per la messa a punto di un approccio di introduzione delle misure, che potranno essere utilizzate in future applicazioni.

È stata studiata la percezione che sviluppatori hanno relativamente alle mutue influenze delle varie qualità del software [74]. Si crede comunemente che esista un "trade-off" tra le varie qualità, ovvero che all'aumentare del livello di una qualità sia verosimile che il livello di altre qualità tenda ad abbassarsi. È stato condotto uno studio sperimentale relativo alle qualità dello standard ISO9126 tramite un'indagine basata su questionari che ha coinvolto professionisti del mondo del software. Sorprendentemente, è risultato che l'opinione comune è che nella maggioranza dei casi le qualità dello standard ISO9126 non sono in conflitto tra di loro, ovvero all'innalzarzi del livello di una si innalza anche il livello delle altre.

Bibliografia [11, 13, 20, 29, 52, 53, 74, 87, 86]

3.1.2 Definizione di misure innovative per la codifica e le fasi alte dello sviluppo

L'attività ha riguardato la definizione di nuovi modelli di misure del software nell'ambito della programmazione tradizionale [82] e concorrente. In particolare, è stata definita una misura per linguaggi concorrenti basata su reti di Petri, estensione del numero ciclomatico, che permette di effettuare una valutazione della complessità del flusso di controllo. Tale misura è in grado di catturare tanto le caratteristiche sequenziali di un programma quanto quelle concorrenti [36]. Si è poi definito un insieme di misure atte a catturare diverse caratteristiche delle reti di Petri in aggiunta alla complessità [44, 51].

In questa area di ricerca si sono effettuate la definizione e la convalida sperimentale di nuove misure per la valutazione del progetto di alto livello del software. Tali misure riguardano tutte le caratteristiche del progetto di alto livello, quali la coesione dei moduli e il disaccoppiamento tra moduli. Dopo una definizione iniziale [41], tali misure sono state sottoposte a un ulteriore raffinamento e a un'analisi sperimentale che ne ha confermato la validità e l'utilità su progetti industriali [14].

Nella prospettiva di estendere la valutazione quantitativa alle prime fasi di sviluppo del software l'attenzione si è poi spostata sull'analisi quantitativa delle specifiche del software [47]. In

particolare è stato condotto un esperimento riguardante specifiche scritte nel linguaggio logico temporale orientato a oggetti TRIO+ per un progetto industriale in cui si sono esaminati i fattori che possono influenzare lo sforzo necessario per scrivere le specifiche e il numero di versioni richieste. Tale studio ha permesso di costruire modelli predittivi di notevole accuratezza.

Infine, è stato proposto un modello, basato su tecniche affidabilistiche, per la valutazione quantitativa dell'efficacia dei processi di scoperta di difetti nei documenti prodotti durante lo sviluppo del software e la stima del numero di tali difetti [8].

Più recentemente, si è investigato se la strutturazione gerarchica dei diagrammi Statecharts favorisca la comprensibilità degli stessi. Uno studio in condotto in ambito accademito e industriale sembra non confermare l'ipotesi secondo cui la strutturazione gerarchica fornisca vantaggi di comprensibilità [73, 22].

Bibliografia [8, 14, 36, 41, 44, 47, 51, 73, ?, 82]

3.1.3 Aspetti teorici

È stata proposta una caratterizzazione teorica delle misure delle qualità interne dei prodotti software, quali dimensione, complessità, coesione e disaccoppiamento [6]. Ciascuna di tali qualità è stata caratterizzata tramite un insieme di proprietà assiomatiche, che le misure devono soddisfare perch possano effettivamente essere considerate adatte alla misura di tale qualità. Ciò ha permesso di rendere pi chiare le differenze e le similarità tra le varie qualità interne dei prodotti software, che molto spesso sono invece riunite nelle due sole categorie di dimensione e complessità. La caratterizzazione proposta può essere applicata a svariati tipi di prodotti del processo software, ovvero alle specifiche, al progetto, al codice, e ai casi di test. L'approccio proposto è stato esteso, dalle misure di tipo ratio per cui era stato definito all'inizio, a misure definite su altri livelli di misura, ossia a intervalli, ordinali e nominali [46]. Tale caratterizzazione risulta uno dei passi fondamentali dell'approccio per la definizione di metriche del software proposto in [19]. La caratterizzazione assiomatica delle misure è stata raffinata [76] per mettere in luce 1) quali assiomi siano davvero fondamentali e quali proprietà siano invece derivate, 2) la sostanziale identità tra l'assiomatizzazione della dimensione del software e la cosiddetta "Teoria della misura" che è alla base delle teorie della derivazione e dell'integrazione in matematica e 3) le relazioni tra le le proprietà delle varie qualità interne del software.

È stata effettuata un'accurata analisi critica sia del modo in cui la teoria della misurazione (Measurement Theory) è stata impiegata nell'ingegneria del software sia del modo in cui, secondo alcune proposte, dovrebbe essere impiegata. In [7, 9, 10, 12, 85] è stato messo in luce come l'uso della teoria della misurazione nell'ingegneria del software empirica sia talvolta stato scorretto e come la teoria della misurazione costituisca senz'altro un prezioso strumento il cui uso deve però essere effettuato senza dogmatismi e temperato dal buon senso.

Inoltre, sono state affrontate le tematiche relative alla convalida di misure per caratteristiche interne del software [84]. L'idea fondamentale è che una misura può dirsi valida se quantifica effettivamente la caratteristica che intende catturare e risulta utile nel contesto di un preciso obiettivo di misura. Ciò comporta l'esigenza di una convalida teorica (ovvero effettuata prima della riprova sperimentale) che mostri che una misura è adeguata per la misura di una determinata caratteristica che viene ritenuta rilevante per un determinato obiettivo, ovvero influente sul fenomeno oggetto dell'obiettivo. D'altro lato, deve essere effettuata una convalida sperimentale, che provi che la misura (e perciò la caratteristica misurata) è correlata al fenomeno che si sta studiando.

Si sono anche estese le tematiche relative alla teoria della misurazione in sé, in modo tale da comprendere in maniera adeguata i casi in cui non è possibile giungere a un ordinamento totale tra le entità da misurare rispetto alla caratteristica che dev'essere quantificata [56]. Si è mostrato che è possibile costruire un sistema concettuale parallelo al sistema utilizzato dalla teoria della misurazione e un sistema di scale di misura (nominali, ordinali, a intervallo, proporzionali) cos come esistono nella teoria della misurazione. Nuovi tipi di scale di misura sono stati introdotti in [64], dove si è inoltre mostrato come in alcuni casi abbia senso calcolare il valor medio di una distribuzione ordinale o addirittura di una distribuzione in cui non si abbia un ordinamento totale tra le entità.

L'analisi è poi stata estesa alle cosiddette caratteristiche esterne del software (ad esempio manutenibilità, usabilità), per le quali, sulla scorta di quanto viene fatto per l'affidabilità, è stata proposta una valutazione in termini di stima e non già di misura diretta. In particolare, per tali caratteristiche viene effettuata una stima in senso probabilistico che permette di ricomprendere tutte queste caratteristiche in un unico schema [80].

Il processo di definizione di misure è stato oggetto di studi, volti a determinare modalità operative per l'identificazione dei fattori rilevanti in dipendenza di contesti applicativi e obiettivi diversi, rispetto ai quali le misure devono essere effettuate [40]. Ciò ha portato alla proposta, sulla base dell'esperienza raccolta, di un approccio per la definizione di misure del software che tengano pienamente conto delle caratteristiche del processo software in cui devono essere applicate e delle ragioni per cui l'attività di misura viene effettuata [19].

Bibliografia [6, 7, 9, 10, 12, 19, 40, 46, 56, 64, 76, 80, 84, 85]

3.1.4 Misure per il web

È stato effettuato uno studio per valutare l'accuratezza delle stime di sforzo necessario per produrre il progetto di un'applicazione web [55]. La fase di progetto di un'applicazione web può richiedere una parte rilevante dello sforzo totale di sviluppo ed è articolata in diverse sottofasi. Lo studio empirico condotto su studenti ha mostrato come vi sia una tendenza a sottovalutare lo sforzo richiesto per le varie fasi. Tuttavia, la previsione dello sforzo può effettuata dai soggetti dello studio essere utilizzata per far parte di un modello di predizione dello sforzo. Inoltre, lo studio empirico ha mostrato come vi sia una correlazione tra caratteristiche del progetto (quali la dimensione e la complessità) e lo sforzo [59, 28]. Lo studio è stato replicato sia con studenti di Ingegneria sia con studenti di Comunicazione, per valutare l'impatto della formazione degli studenti sullo sforzo necessario per portare a compimento le varie attività connesse al progetto del software per il web [21]. Inoltre, ciò ha permesso di valutare l'esistenza di aspetti comuni e invarianti rispetto alla formazione dei soggetti.

Bibliografia [55, 59, 28, 21]

3.1.5 Valutazione del software Open Source

È stato proposto un modello, detto OpenBQR, per la valutazione del software Open Source [71]. Il modello prende spunto dai modelli esistenti in letteratura e si propone di fornire uno strumento che risulti più facile da usare e completo. Una valutazione preliminare ha fornito elementi incoraggianti per poterne proseguire la valutazione in un ampio insieme di progetti. Inoltre, un'indagine preliminare ha permesso di avere una prima valutazione relativamente a quali importanti fattori che influenzano la credibilità (trustworthiness) del software sia possibile effettivamente misurare sui prodotti Open Source. Ciò può dare agli sviluppatori utili indicazioni su quali aspetti e documenti migliorare per far sì che un prodotto possa essere valutato da chi lo vuole utilizzare [75, 78]. Una proposta per la certificazione del processo di test Open Source è stata introdotta in [77]. Una proposta iniziale per la valutazione e la certificazione dei portali Open Source si trova in [79].

Bibliografia [71, 72, 75, 77, 78, 79]

3.1.6 Tecniche innovative di analisi dei dati

Tecniche innovative di analisi dei dati. Nello studio all'interno del progetto CEMP si è utilizzato l'insieme di dati sperimentali raccolto a riguardo del progetto nella fase di manutenzione per confrontare utilità e complementarità di due tecniche per l'analisi dei dati (regressione logistica e rough sets) [18, 45]. Si è mostrato che l'uso combinato di tali due tecniche permette di migliorare il potere predittivo dei modelli ottenuti e ridurre gli errori di Tipo I e di Tipo II. È stato sviluppato un nuovo metodo per l'analisi dei dati sperimentali che estende al caso di variabili indipendenti continue le tecniche degli alberi di decisione [50, 54]. Le tecniche disponibili in letteratura non sono infatti in grado di utilizzare direttamente le variabili continue, ce devono invece essere preventivamente

rese discrete. Ciò comporta una perdita di informazione, che implica una perdita di potere predittivo delle variabili indipendenti continue. Inoltre, il processo di discretizzazione delle variabili continue è in larga misura soggettivo e arbitrario. Il metodo di analisi proposto permette invece di superare tali problemi presenti nelle tecniche della letteratura in modo naturale e perfettamente congruente con gli alberi di decisione. Il metodo è stato applicato alla predizione della presenza di difetti nei moduli di un sistema software e ha dimostrato di essere pi efficace delle proposte di riferimento della letteratura. È stato poi definito un approccio assiomatico per la definizione di indici di concentrazione di distribuzioni di dati che può essere utile nel caso di dati di tipo nominale. Tale approccio permette di definire diversi indici di concentrazione che possono essere scelti in base a specifiche esigenze e sorprendentemente porta a utilizzare la trasformata Z come metodo per la costruzione di tali indici. Inoltre, si è mostrato come non sia vera in assoluto l'affermazione secondo la quale non abbia mai senso effettuare l'operazione di media per dati di tipo ordinale. Si è dimostrato un teorema sotto forma di condizione necessaria e sufficiente che fornisce le condizioni pi ampie perch l'operazione di media possa avere senso anche per dati di tipo ordinale [64].

Bibliografia [18, 45, 50, 54, 64]

3.1.7 Didattica dell'ingegneria del software e metodi empirici

Sulla base di esperimenti effettuati utilizzando gli studenti dei corsi di software come soggetti, si è cercato di determinare innanzitutto quali siano le figure professionali e i ruoli coinvolti nell'effettuazione di esperimenti durante i corsi software con studenti. La letteratura si sofferma in particolare soltanto sulle questioni relative alla validità esterna degli esperimenti condotti con studenti, per cui viene preso in considerazione solo il punto di vista dello sperimentatore. È invece fondamentale considerare altri tre punti di vista, ovvero lo studente, l'insegnante e l'industria. Ciascuno di questi quattro ruoli deve trarre beneficio dall'effettuazione di questi studi, che pertanto devono essere inseriti all'interno dei corsi relativi al software in maniera opportuna in modo da bilanciare le esigenze [27, 60, 23]. Uno studio relativo a un corso per dottorandi si trova in [27, 60]. Alcuni aspetti della relazione tra l'industria e il mondo della didattica sono esplorati in [67, 68]. Inoltre, si è iniziato ad affrontare il tema dell'insegnamento dei metodi empirici agli studenti [69].

Bibliografia [27, 60, 67, 68, 69, 23]

3.2 Specifica di sistemi concorrenti ed in tempo reale

Tale area costituisce il tema principale dell'attività svolta nell'ambito del dottorato di ricerca [88]. È stato definito in modo pienamente formale un modello di rete di Petri di alto livello, detto delle reti ambiente/relazione (reti ER), che integra le caratteristiche delle precedenti proposte di estensione temporale alle reti di Petri e delle reti di Petri di alto livello. Tale formalismo è in grado di dare una rappresentazione naturale delle caratteristiche funzionali e temporali di un sistema e delle loro interrelazioni. Si è dimostrato che tale modello risulta una generalizzazione dei modelli di reti di Petri temporizzate e in pi permette di rappresentare situazioni di interesse che tali modelli non sono in grado di specificare. Inoltre le modalità di rappresentazione del comportamento funzionale e temporale di un sistema all'interno delle reti ER sono del tutto uniformi: ciò ha permesso di mostrare come sia possibile dare una rappresentazione di tali aspetti interamente nell'ambito delle reti di Petri di alto livello, senza dover introdurre nuove primitive temporali ad hoc [2, 30, 31, 81]. L'estensione di tale tipo di introduzione delle caratteristiche temporali all'intera classe delle reti di Petri di alto livello è stata effettuata mostrando l'equivalenza delle reti ER rispetto a tali modelli [3].

È stata poi data un'estensione del modello delle reti ER in senso stocastico: ciò permette di rappresentare quei sistemi concorrenti e in tempo reale il cui comportamento è regolato da leggi probabilistiche per mezzo di una rete di Petri di alto livello [83]. Inoltre, è stata data un'estensione delle reti ER in cui è possibile modellare sistemi con pi riferimenti temporali [48].

Parte delle ricerche è stata dedicata alla definizione, alla progettazione e alla realizzazione di un ambiente per la specifica e la verifica di sistemi concorrenti ed in tempo reale, basato sul modello delle reti ER definito [30, 32], in collaborazione con il CEFRIEL.

Inoltre, data la generalità del modello delle reti ER, si è vagliata la possibilità dell'estensione della sua applicazione ad altre tipologie di sistemi. Uno studio preliminare ha portato al loro impiego per la specifica dell'hardware [35, 37].

È stata poi studiata la possibilità di utilizzare l'estensione stocastica delle reti ER descritta in [83] per modellare processi produttivi del software, con particolare riferimento alla definizione e allo studio delle loro prestazioni quando vengano apportate modifiche ai processi stessi [39].

Infine, si è definita un'estensione temporizzata [57, 61, 62] di UML, in modo tale da poter trattare adeguatamente anche gli aspetti temporali in UML. La formalizzazione del tempo viene effettuata sia nei modelli operazionali di UML come gli Statechart sia nei modelli descrittivi come OCL.

Bibliografia [2, 3, 30, 31, 32, 34, 35, 37, 39, 48, 57, 61, 62, 81, 83, 88]

3.3 Verifica del software

3.3.1 Modelli operazionali

Il modello delle reti ER è stato preso come base per l'esecuzione simbolica di sistemi concorrenti, a livello sia di specifica sia di programma. È stato definito innanzitutto un algoritmo generale di esecuzione simbolica per una classe di reti ER in grado di modellare appieno sistemi concorrenti e sistemi a tempo discreto. È stato poi definito un altro algoritmo, per una sottoclasse di tali reti, da cui possono essere rappresentati buona parte dei sistemi di interesse, che permette di ridurre significativamente la quantità di informazioni che devono essere ritenute [1, 33]. Sempre dal punto di vista dell'analisi, è stato definito un insieme di criteri per la selezione di casi di test, che si basano sulla copertura topologica della rete [34]. È stato poi studiato il problema dell'analisi delle proprietà temporali delle reti TB (caso particolare delle reti ER, in cui si considera solo il comportamento temporale del sistema e si prescinde da quello funzionale). Ciò ha portato alla definizione di metodi per valutare la raggiungibilità di un determinato stato entro un prespecificato intervallo temporale [4].

Bibliografia [1, 4, 33, 34]

3.3.2 Modelli di tipo logico

È stato studiato il problema della generazione di casi di test funzionali per sistemi in tempo reale a partire da specifiche di tipo logico. In particolare, per il linguaggio di logica temporale TRIO, è stato individuato un algoritmo per la derivazione (semi)automatica di tali casi di test [38]. Al fine di ridurre la complessità intrinseca a tale procedimento, sono stati definiti vari criteri, che costituiscono l'analogo dei criteri di copertura nella derivazione di casi di test strutturali [5]. Ciò ha portato al progetto e alla realizzazione prototipale di uno strumento per la generazione di tali casi di test.

In [42]sono stati poi proposti diversi criteri e tecniche per la generazione di casi di test "in grande" per sistemi in tempo reale specificati tramite TRIO+, estensione a oggetti del linguaggio TRIO. Inoltre, in [16] è stata proposta una tecnica per la generazione di casi di test funzionali per sistemi in tempo reale con struttura modulare e gerarchica che può essere applicata a una vasta classe di linguaggi di specifica. Per tale tecnica è anche stato realizzato uno strumento automatizzato per agevolarne l'uso [17, 49].

Bibliografia [5, 16, 17, 42, 49]

3.3.3 Analisi mutazionale

La tecnica dell'analisi mutazionale, usualmente applicata a programmi sequenziali, è stata estesa alla programmazione concorrente. Si è preso il linguaggio Ada come riferimento e si sono introdotti svariati operatori mutanti specifici per i costrutti concorrenti del linguaggio. L'applicazione di tale tecnica permette di valutare quanto bene un insieme di dati di test riesca ad esercitare le caratteristiche di concorrenza di un programma Ada [43].

Bibliografia [43]

3.3.4 Regola di terminazione del testing

Uno dei problemi fondamentali della verifica del software risiede nell'individuazione di regole per la terminazione del testing. La letteratura contiene diverse proposte che sono solitamente basate su criteri apparentemente oggettivi. Tuttavia, soprattutto nel caso di sistemi con requisiti di alta affidabilità, tali proposte risultano largamente inadeguate. È stato studiato un approccio basato su tecniche statistiche bayesiane con l'idea che il testing in realtà termini quando i verificatori hanno raggiunto un sufficiente grado di confidenza del buon funzionamento del software. È stata effettuata una sistemazione teorica dell'uso di tali tecniche statistiche a questo scopo ed è stato proposto un modello per descrivere come varia l'evoluzione del grado di confidenza dei verificatori nel buon funzionamento del software durante il processo di test. È stato poi effettuato un esperimento che ha portato a confermare in buona parte le ipotesi iniziali [54].

Bibliografia [54]

3.3.5 Studi analitici

Nella verifica del software è importante essere in grado di confrontare tecniche di testing diverse in base alla loro efficacia nel causare un elevato numero di malfunzionamenti oppure nel causare almeno un malfunzionamento. È stato effettuato uno studio teorico che mostra quali sono le condizioni meno restrittive per poter allocare in maniera razionale i dati di test ai vari sottodomini che compongono il dominio di un programma. In particolare, è stato dimostrato un teorema sotto forma di condizione necessaria e sufficiente che permette di confrontare il numero atteso di malfunzionamenti nel causati da tecniche di testing diverse di qualunque tipo sulla base della conoscenza dell'ordinamento dei sottodomini in base al loro tasso di malfunzionamento. Pertanto, non è necessario conoscere il valore esatto dei tassi di malfunzionamento dei singoli sottodomini. Inoltre, la teoria è stata estesa anche al caso di conoscenza di un ordinamento parziale tra i tassi di malfunzionamento dei sottodomini e alla probabilità di riscontrare almeno un malfunzionamento [63].

È stato proposto un approccio per valutare la frequenza media di malfunzionamento di un programma sulla base del test statistico delle ipotesi [70]. L'approccio può essere utilizzato per stabilire regole di terminazione e valutare la qualità di un programma sulla base della sua frequenza media di malfunzionamento durante le fasi finali del test. La proposta mostra come stabilire e soddisfare dei limiti conservativi per il numero minimo di esecuzioni di test necessarie per raggiungere una frequenza media di malfunzionamento desiderata con un livello specificato di significatività statistica, sulla base degli obiettivi di qualità del test e del profilo di esecuzione del test. L'approccio inoltre rende meno forti alcune assunzioni della letteratura, per cui può essere utilizzato in un insieme più ampio di casi reali.

Bibliografia [63, 70]

3.3.6 Verifica per sistemi Web

È stato studiato un approccio per la generazione di dati di test per un generatore di applicazioni Web. In particolare, i dati di test vengono definiti come modelli di applicazioni Web codificati in WebML. I vari modelli esercitano produzioni diverse della grammatica di WebML, a riguardo della quale sono state definite diverse misure di copertura. Un'indagine empirica preliminare in un ambiente industriale ha permesso di studiare quanto bene i modelli esistenti utilizzati come dati di test coprano le produzioni grammaticali, in modo da evidenziare possibili miglioramenti derivanti dall'ampliamento dei dati di test esistenti [65].

Bibliografia [65]

4 Varie

4.1 Borse di studio

È risultato vincitore delle seguenti borse di studio.

- Una borsa di studio sul tema di ricerca "Testing funzionale di sistemi software" bandita dalla società Selenia S. p. A. per gli anni 1987, 1988 e 1989.
- Una borsa di studio della durata di diciotto mesi bandita dal consorzio europeo ERCIM per ricerche da condursi presso università e istituzioni scientifiche europee.
- Due borse di studio della durata di dodici mesi bandite dal CNR per ricerche da condursi presso università e istituzioni scientifiche italiane.
- Due borse di studio della durata di dodici mesi bandite dal CNR per ricerche da condursi presso università e istituzioni scientifiche estere.

4.2 Premi

È risultato vincitore dei seguenti premi.

- Premio "Francesco Somaini" nel 1979 come miglior studente del liceo classico "A. Volta" di Como iscrittosi a una facoltà scientifica in quell'anno.
- Premio "Electrical Engineering Student Award" 1988 bandito dalla IEEE North Italy section.
- Premio del concorso "Seymour Cray" 1991.
- Premio finale per una borsa di studio CNR per ricerche da condursi presso università e istituzioni scientifiche estere.

4.3 Progetti

Le attività di ricerca sopra riportate sono state svolte parzialmente nell'ambito di progetti nazionali ed internazionali.

- Progetto MPI 40% "Reti di Petri: modelli, applicazioni, strumenti"
- Progetto MPI 60% "Strumenti per la specifica e la verifica di sistemi software in tempo reale"
- Progetto CINI "PROTAGORA"
- Progetto cofinanziato MURST "MOSAICO"
- Progetto cofinanziato MIUR "QUACK"
- Progetto Esprit "IPTES"
- Progetto ESSI "CEMP"
- Progetto ESSI "PROMOTE"
- Progetto IST "ESERNET" (Thematic Network)
- Progetto IST "QualiPSo" (Integrated Project)

In particolare, all'interno del progetto QualiPSo, é l'Activity Leader dell'attivitá relativa alla valutazione della qualitá dei prodotti e componenti Open Source.

4.4 Tutorial

Ha presentato il tutorial "Formal Methods in Software Measurement and Software Measurement in Formal Methods" alle conferenze internazionali METRICS 2003 e Formal Methods Europe 2003. Ha presentato (con Luciano Baresi) il tutorial "An Introduction to Web Quality" alla conferenza internazionale ICWE (International Conference on Web Engineering) 2004.

Ha presentato il tutorial "Fundamental Aspects of Empirical Software Engineering" alla conferenza internazionale ESELAW 2007.

4.5 Keynote e seminari

È stato invitato a tenere un discorso keynote alle seguenti conferenze

- OSS 2007
- ESELAW 2007
- SAST 2008

È stato invitato a tenere seminari in diverse istituzioni universitarie, tra cui l'Università di Stoccarda (Germania), l'Universidad de Castilla-La Mancha di Ciudad Real (Spagna), la Norwegian University of Science and Technology a Trondheim (Norvegia), la Libera Università di Bolzano, l'Universidad Politécnica de Valencia (Spagna), la North Carolina State University a Raleigh (USA). È stato inoltre invitato a tenere una lezione presso la 6th International Summer School on Software Engineering svolta a Salerno dal 16 al 18 settembre 2009.

4.6 Attività di revisore in conferenze

Ha fatto parte dei comitati di programma delle conferenze internazionali "Euromicro", "Euromicro Workshop on Real-Time Systems", "Workshop on Industrial-strength Formal Techniques" (WIFT), "International Conference on Software Engineering and Software Engineering" (SEKE), "International Conference on Software Maintenance" (ICSM), "International Workshop on Empirical Studies of Software Maintenance" (WESS), "METRICS", "European Conference on Software Quality" (ECSQ), "Brasilian Symposium on Software Engineering" (SBES), "International Conference on Web Engineering" (ICWE), "International Symposium on Software Reliability Engineering" (ISSRE), "International Symposium on Empirical Software Engineering" (ISESE), "Fundamental Approaches in Software Engineering" (FASE), "Latin American Web Congress" (LA-Web), "International Conference on Software Engineering Advances" (ICSEA), "International Conference on Open Source Systems" (OSS), "International Symposium on Empirical Software Engineering and Measurement" (ESEM).

4.7 Coordinamento scientifico di conferenze

- Program Chair di un Workshop sulla misura di software orientato a oggetti all'interno di ECOOP'98.
- Program co-Chair del "First International Workshop on Web Quality" nell'ambito della conferenza internazionale ICWE (International Conference on Web Engineering) 2004.
- Program co-Chair di un workshop su "Measurement and Metrics" nell'ambito della conferenza internazionale WWW 2005.
- Program co-Chair del "1st International Workshop on Trust in Open Source Software" (TOSS).
- Program co-Chair di ICSEA 2007.
- Chair del "Second International Doctoral Symposium on Empirical Software Engineering" nell'ambito della International Empirical Software Engineering Week 2007.
- È stato componente dello Steering Committee della conferenza internazionale METRICS.
- È stato componente dello Steering Committee della conferenza internazionale ESEM.

4.8 Organizzazione di conferenze

Ha collaborato, in qualità di Finance Chair, all'organizzazione del "Sixth International Workshop on Software Specification and Design", sotto gli auspici dell'IEEE e dell'ACM.

È stato componente del comitato organizzatore del Congresso AICA 2001.

Ha organizzato in qualità di General Chair la conferenza internazionale METRICS 2005 a Como.

4.9 Attività di revisore in riviste scientifiche

È revisore di articoli per conferenze internazionali, come ad esempio ICSE ed ESEC, e riviste internazionali, quali IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Software Engineering e ACM Transactions on Software Engineering Methodology.

È stato guest editor di una Special Issue on Knowledge Discovery from Software Engineering Data del International Journal of Software Engineering and Knowledge Engineering [15].

È componente dell'Editorial Board della rivista internazionale Empirical Software Engineering: An International Journal, edita da Springer-Verlag.

È componente del Comitato dei Revisori della rivista internazionale International Journal of Open Source Software and Processes (IJOSSP), edita da IGI Infernational.

4.10 Partecipazione a commissioni ministeriali

È stato componente della commissione sul Software Open Source del Ministero per le Riforme e le Innovazioni nella Pubblica Amministrazione.

References

5 Elenco dei lavori

5.1 Pubblicazioni su rivista

[1] Carlo Ghezzi, Dino Mandrioli, Sandro Morasca, Mauro Pezzè, "Symbolic Execution of Concurrent Programs Using Petri Nets", Computer Languages, Vol. 14, pagg. 263-281, dicembre 1989, Pergamon Press, ISSN 0096-0551.

[2] Carlo Ghezzi, Dino Mandrioli, Sandro Morasca, Mauro Pezzè, "A Unified High-level Petri Net Model for Time-critical Systems", IEEE Transactions on Software Engineering, Vol. SE-17, pagg. 160-172, febbraio 1991, ISSN: 0098-5589.

[3] Sandro Morasca, Mauro Pezzè, Marco Trubian, "Timed High-level Nets", The Journal of Real-time Systems, Vol. 3, pagg. 165-189, luglio 1991, ISSN: 0922-6443.

[4] Carlo Ghezzi, Sandro Morasca, Mauro Pezzè, "Validating Timing Requirements for TB Net Specifications", The Journal of Systems and Software, Vol. 27, n. 7, pagg. 97-117, novembre 1994, ISSN: 0164-1212.

[5] Dino Mandrioli, Sandro Morasca, Angelo Morzenti, "Generating Test Cases for Real-Time Systems from Logic Specifications", ACM Transactions on Computer Systems, Vol 13. n. 4, pagg. 365 - 398, novembre 1995, ISSN: 0734-2071.

[6] Lionel Briand, Sandro Morasca, Victor Basili, "Property-based Software Engineering Measurement", IEEE Transactions on Software Engineering, pagg. 68-86, gennaio 1996, ISSN: 0098-5589.

[7] Lionel Briand, Khaled El Emam, Sandro Morasca, "On the Application of Measurement Theory in Software Engineering", Empirical Software Engineering: An International Journal, Vol. 1, n. 1, pagg. 61 - 88, 1996, ISSN: 0098-5589.

[8] Sandro Morasca, "Assessment of Fault Detection Processes: An Approach Based on Reliability Techniques", IEEE Transactions on Reliability, Vol. TR-45, pagg. 632 - 637, dicembre 1996, ISSN 0018-9529.

[9] Sandro Morasca, Lionel Briand, Victor Basili, Elaine Weyuker, Marv Zelkowitz, "Comments on "Towards a Framework for Software Measurement Validation"", IEEE Transactions on Software Engineering, pagg. 187 - 188, marzo 1997 (correspondence paper), ISSN: 0098-5589.

[10] Lionel Briand, Sandro Morasca, Victor Basili, "Response to: Comments on "Property-based Software Enginering Measurement: Refining the Additivity Property"", IEEE Transactions on Software Engineering, pagg. 196 - 197, marzo 1997 (correspondence paper), ISSN: 0098-5589.

[11] Sandro Morasca, "Applying QIP/GQM in a Maintenance Project", Empirical Software Engineering: An International Journal, Vol. 2, No. 2, 1997, pagg. 161 - 164 (position paper), ISSN: 1382-3256. (Una versione precedente dell'articolo è stata pubblicata al congresso WESS'96, Monterey (USA), 8 novembre 1996.)

[12] Lionel Briand, Khaled El Emam, Sandro Morasca, "Reply to "Comments to the Paper: Briand, El Emam, Morasca: On the Application of Measurement Theory in Software Engineering"", Empirical Software Engineering: An International Journal, 1997 (correspondence paper), ISSN: 1382-3256.

[13] Alfonso Fuggetta, Luigi Lavazza, Sandro Morasca, Stefano Cinti, Giandomenico Oldano, Elena Orazi, "Applying G/Q/M in an Industrial Software Factory", ACM Transactions on Software Engineering Methodology, Vol. 7, n. 4, pagg. 411 - 448, ottobre 1998, ISSN: 1049-331X.

[14] Lionel Briand, Sandro Morasca, Victor Basili, "Defining and Validating Measures for Object-based High-Level Design", IEEE Transactions on Software Engineering, Vol. SE-25, No.5, pagg. 722 - 741, settembre/ottobre 1999, ISSN: 0098-5589.

[15] Sandro Morasca, Günther Ruhe, "Guest editors' introduction of the Special issue on Knowledge Discovery from Software Engineering Data", International Journal of Software Engineering and Knowledge Engineering, Vol. 9, No. 5, pp. 496 - 498, ottobre 1999, ISSN: 0218-1940.

[16] Pierluigi San Pietro, Angelo Morzenti, Sandro Morasca, "Generation of Execution Sequences for Modular Time-dependent Systems", IEEE Transactions on Software Engineering, Vol. SE-26, No. 2, pagg. 128 - 149, febbraio 2000, ISSN: 0098-5589.

[17] Sandro Morasca, Angelo Morzenti, Pierluigi San Pietro, "A Case Study on Applying a Tool for Automated System Analysis Based on Modular Specifications Written in TRIO+", Automated Software Engineering, Vol. 7, No. 2, pagg. 125-155, maggio 2000, ISSN: 0928-8910.

[18] Sandro Morasca, Günther Ruhe, "A Hybrid Approach to Analyze Software Engineering Measurement Data and Its Application to Predict Module Fault-proneness in Maintenance", The Journal of Systems and Software, Vol. 53, No. 3, pagg 225 - 237, settembre 2000, ISSN: 0164-1212.

[19] Lionel Briand, Sandro Morasca, Victor Basili, "An Operational Process for Goal-Driven Definition of Measures", IEEE Transactions on Software Engineering, Vol. 28, No. 12, pagg. 1106 - 1125, dicembre 2002, ISSN: 0098-5589.

[20] Giovanni Denaro, Mauro Pezzè, Sandro Morasca, "Towards Industrially Relevant Models of Software Fault Proneness", International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 4, pagg. 395-417, agosto 2003, ISSN: 0218-1940.

[21] Luciano Baresi, Sandro Morasca, "Three empirical studies on estimating the design effort of Web applications", ACM Transactions on Software Engineering Methodology, Vol. 16, n. 4, pagg. 411 - 448, settembre 2007, ISSN: 1049-331X.

[22] José Antonio Cruz Lemus, Marcela Genero, Esperanza Manso, Sandro Morasca, Mario Piattini, "Assessing the Understandability of UML Statechart Diagrams with Composite States - A Family of Empirical Studies", accettato per pubblicazione su Empirical Software Engineering: An International Journal (Springer-Verlag).

[23] Jeff Carter, Letizia Jaccheri, Sandro Morasca, Forrest Shull, "A Checklist for Integrating Student Empirical Studies with Research and Teaching Goals", accettato per pubblicazione su Empirical Software Engineering: An International Journal (Springer-Verlag).

5.2 Libri

[24] Carlo Ghezzi, Alfonso Fuggetta, Sandro Morasca, Angelo Morzenti, Mauro Pezzè, "Ingegneria del software - Progettazione, sviluppo e verifica", Mondadori Informatica, Milano, 1991, ISBN: 8871316215.

[25] Augusto Celentano, Pierluigi Della Vigna, Piero Fraternali, Sandro Morasca, "Fondamenti di Informatica - Dal problema al programma", ETAS libri, 1999, ISBN: 8845308367.

5.3 Capitoli di libri

[26] Sandro Morasca, "Software Measurement", in "Handbook of Software Engineering and Knowledge Engineering - Volume 1: Fundamentals" (libro sottoposto ad attività di recensione), pagg. 239 - 276, Knowledge Systems Institute, Skokie, IL, USA, 2001, ISBN: 981-02-4973-X.

[27] Jeff Carter, Letizia Jaccheri, Sandro Morasca, Forrest Shull, "Using Empirical Studies during Software Courses", in "ESERNET Method and Experience Book" (libro sottoposto ad attività di recensione), LNCS 2765, pagg. 81 - 103, Springer-Verlag, 2003, ISBN: 3-540-40672-7.

[28] Luciano Baresi, Sebastiano Colazzo, Luca Mainetti, and Sandro Morasca, "W2000: A Modeling Notation for Complex Web Applications", in Web Engineering (ed. Emilia Mendes e Nile Mosley)(libro sottoposto ad attività di recensione), pagg. 335 - 364, Springer-Verlag, 2006, ISBN: 3-540-28196-7.

5.4 Pubblicazioni su atti di congressi

[29] Roberto Lecciso, Stefano Mainetti, Sandro Morasca, "Software Metrics: a Critical Evaluation and an Application to Pascal", EUROMICRO '86, Venezia, 16-18 settembre 1986, pagg. 605-615, North-Holland.

[30] Sandro Morasca, Mauro Pezzè, "Reti di Petri estese e sviluppo di sistemi in tempo reale", Congresso annuale AICA, Cagliari, 28 - 30 settembre 1988, pagg. 267-281.

[31] Carlo Ghezzi, Dino Mandrioli, Sandro Morasca, Mauro Pezzè, "A General Way to Put Time in Petri Nets", Proceedings of the IEEE 5th International Workshop on Software Specification and Design, Pittsburgh (USA), 19 - 20 maggio 1989, pagg. 60-67, IEEE CS Press, ISSN: 0163-5948.

[32] Sandro Morasca, Mauro Pezzè, "The Rationale of an Environment for Real-time Software", EUROMICRO Workshop on Real Time, Como 14 - 16 giugno 1989, pagg. 37-42, IEEE CS Press.

[33] Sandro Morasca, Mauro Pezzè, "Validation of Concurrent Ada Programs Using Symbolic Execution", 2nd European Software Engineering Conference, Coventry (GB), settembre 1989, pagg. 469-486, LNCS 387, Springer-Verlag, ISBN: 3-540-51635-2.

[34] Sandro Morasca, Mauro Pezzè, "Using High-level Petri Nets for Testing Concurrent and Real-time Systems", Workshop on Real-time Systems - Theory and Applications, York, 28 - 29 settembre 1989, pagg. 119-131, North-Holland.

[35] Giacomo Buonanno, Sandro Morasca, Mauro Pezzè, Kim Portman, Donatella Sciuto, "Using High-level Petri Nets for Timing Specification of Hardware", Tau '90, 15 - 17 agosto 1990, Vancouver (Canada).

[36] Flavio De Paoli, Sandro Morasca, "Extending Software Complexity Metrics to Concurrent Programs", COMPSAC '90, Chicago (USA), 29 ottobre - 2 novembre 1990, pagg. 414-419, IEEE CS Press.

[37] Giacomo Buonanno, Sandro Morasca, Mauro Pezzè, Kim Portman, Donatella Sciuto, "A New Timed Petri Net Model for Hardware Representation", CHDL 91, Marsiglia (Francia), 22 - 24 aprile 1991, pagg. 261-280, North-Holland.

[38] Dino Mandrioli, Sandro Morasca, Angelo Morzenti, "Functional Test Case Generation for Real-time Systems", 3rd IFIP Working conference on Dependable Computing for Critical Applications, Palermo 14-16 settembre 1992, pagg. 13 - 26.

[39] Gianluigi Caldiera, Sandro Morasca, "Performance Evaluation of Software Process Evolution", International Workshop on Evolution of Software Processes, McGill University, Montreal (Canada) 26-28 gennaio 1993.

[40] Lionel Briand, Sandro Morasca, "Useful Metrics Must Depend on a Process and a Goal", International Workshop on Metrics in Software Maintenance, Montreal (Canada), 26 settembre 1993.

[41] Lionel Briand, Sandro Morasca, Victor Basili, "Measuring and Assessing Maintainability at the End of High Level Design", International Conference on Software Maintenance, Montreal (Canada), 27-30 settembre 1993, pagg. 88 - 97, IEEE CS Press, ISBN: 0-8186-4600-4.

[42] Sandro Morasca, Angelo Morzenti, Pierluigi San Pietro, "Generating Functional Test Cases in-the-large for Time-critical Systems from Logic-based Specifications", ISSTA '96, San Diego (USA), 8-10 gennaio 1996, pagg. 39 - 52, ACM Press, ISBN: 0897917871.

[43] Sandro Morasca, Mauro Pezzè, Sergio Silva-Barradas, "Mutation Analysis for Concurrent Ada Programs", Quality Week'96, San Francisco (USA), 21-24 maggio 1996.

[44] S. Morasca, "Defining Measures for Petri Net-based Specifications of Concurrent Software", AOWSM'97, Coeur d'Alene (USA), 12 - 13 maggio 1997.

[45] Sandro Morasca, Günther Ruhe, "Knowledge Discovery from Software Engineering Measurement Data: A Comparative Study of Two Analysis Techniques", SEKE'97, Madrid (Spagna), 18 - 20 giugno 1997, pagg. 450 - 458, ISBN: 0-9641699-5-9.

[46] Sandro Morasca, Lionel Briand, "Towards a Theoretical Framework for Measuring Software Attributes", METRICS'97, Albuquerque (USA), 5 - 7 novembre 1997, pagg. 119 - 126.

[47] Lionel Briand, Sandro Morasca, "Software Measurement and Formal Methods: a Case Study Centered on TRIO+ Specifications", IEEE ICFEM'97 (IEEE International Conference on Formal Engineering Methods), Hiroshima (Giappone), 12 - 14 novembre 1997, pagg. 315 - 325, IEEE CS Press, ISSN: 0-8186-8002-4.

[48] Sandro Morasca, "Using MTER Nets for Modeling Multiple Time References in Distributed Systems", DCCS'98, Como, 9 - 11 settembre 1998, pagg. 113 - 118, ISBN: 0-08-043242-5.

[49] Sandro Morasca, Angelo Morzenti, Pierluigi San Pietro, "A Tool for Automated System Analysis Based on Modular Specifications", 13th IEEE International Conference on Automated Software Engineering - ASE'98, Honolulu (USA), 13 - 16 ottobre 1998, pagg. 2 - 11, IEEE CS Press, ISBN: 0-8186-8750-9.

[50] Sandro Morasca, "Towards a Way to Deal with Continuous Attributes in Decision Trees", SEKE'99 (poster paper), Kaiserslautern (Germania), 16 - 19 giugno 1999, pagg. 262 - 266, ISBN: 1-891706-01-2.

[51] Sandro Morasca, "Measuring Attributes of Concurrent Software Specifications in Petri Nets", METRICS'99, Boca Raton, Florida, USA, , 4 - 6 novembre 1999, pagg. 100 - 110, IEEE CS Press, ISBN: 0-7695-0403-5.

[52] Sandro Morasca, Giuliano Russo, "An Empirical Study of Software Productivity", COMPSAC '01, 8 - 12 novembre 2001, Chicago (USA), pagg. 317-322, IEEE CS Press, ISBN: 0-7695-1372-7.

[53] Giovanni Denaro, Sandro Morasca, Mauro Pezzè, "Deriving Models of Software Fault-proneness", 14th International Conference on Software Engineering and Knowledge Engineering (SEKE 2002), 15 - 19 luglio 2002, Ischia, pagg. 361 - 368, ACM Press, ISBN: 1-58113-556-4.

[54] Sandro Morasca, "A Proposal for Using Continuous Attributes in Classification Trees", 14th International Conference on Software Engineering and Knowledge Engineering (SEKE 2002), 15 - 19 luglio 2002, Ischia, pagg. 417-424, ACM Press, ISBN: 1-58113-556-4.

[55] Luciano Baresi, Sandro Morasca, Paolo Paolini, "An Empirical Study on the Design Effort Of Web Applications", 3rd International Conference on Web Information Systems Engineering (WISE 2002), 11 - 14 dicembre 2002, Singapore, pagg. 345-354, IEEE CS Press, ISBN: 0-7695-1766-8.

[56] Sandro Morasca, "Foundations of a Weak Measurement-Theoretic Approach to Software Measurement", Fundamental Approaches in Software Engineering (FASE) 2003, 7 - 11 aprile 2003, Varsavia (Polonia), LNCS 2621, pagg. 200 - 215, Springer Verlag, ISBN: 3-540-00899-3.

[57] Luigi Lavazza, Sandro Morasca, Angelo Morzenti, "A Dual Language Approach Extension to UML for the Development of Time-Critical Component-Based Systems", atti di TACoS (Test and Analysis of Component Based Systems) 2003, 13 aprile 2003, Varsavia (Polonia), ISBN: 3-540-00899-3.

[58] Sandro Morasca, "A Bayesian Approach to Software Testing Evaluation", 15th International Conference on Software Engineering and Knowledge Engineering (SEKE 2003), 1 - 3 luglio 2003, Redwood City (USA), pagg. 706 - 713, ISBN: 1-891706-12-8.

[59] Luciano Baresi, Sandro Morasca, Paolo Paolini, "Estimating the Design Effort of Web Applications: a Case Study Centered on TRIO+ Specifications", METRICS'03, 3 - 5 settembre 2003, Sydney (Australia), pagg. 62 - 72, IEEE CS Press, ISBN: 0-7695-1987-3.

[60] Jeff Carter, Letizia Jaccheri, Sandro Morasca, Forrest Shull, "Issues in Using Students in Empirical Studies in Software Engineering Education", METRICS'03, 3 - 5 settembre 2003, Sydney (Australia), pagg. 239 - 249, IEEE CS Press, ISBN: 0-7695-1987-3.

[61] Luigi Lavazza, Sandro Morasca, Angelo Morzenti, "A Dual Language Approach to the Development of Time-Critical Systems", Workshop on Critical System Development, 21 ottobre 2003, San Francisco (USA) (poster paper), pubblicato come Rapporto Interno della Technische Universität Muenchen TUM-I0323, settembre 2003, pagg. 113 - 119.

[62] Luigi Lavazza, Sandro Morasca, Angelo Morzenti, " A Dual Language Approach to the Development of Time-Critical Systems with UML", atti di TACoS (Test and Analysis of Component Based Systems) 2004, 27 - 28 marzo 2004, Barcellona (Spagna), ISSN: 1571-0661.

[63] Sandro Morasca, Stefano Serra Capizzano, "On the Analytical Comparison of Testing Techniques", ISSTA 2004, Boston (USA), 11-14 luglio 2004, pagg. 154 - 164, ISBN 1-58113-820-2.

[64] Sandro Morasca, "On the Definition and Use of Aggregate Indices for Nominal, Ordinal, and Other Scales", METRICS'04, 14-16 settembre 2004, Chicago (USA), pagg. 46 - 57, IEEE CS Press, ISBN: 0-7695-2129-0.

[65] Luciano Baresi, Piero Fraternali, Massimo Tisi, Sandro Morasca, "Towards Model-Driven Testing of a Web Application Generator", ICWE 2005, 25 - 29 luglio 2005, Sydney (Australia), LNCS 3579, pagg. 75 - 86, ISBN: 3-540-27996-2.

[66] Letizia Jaccheri, Sandro Morasca, "Understanding the Role of Software Metrics in an Empirical Software Engineering Course for PhD Students", Workshop "Methods for Learning Metrics" di METRICS 2005, 19 settembre 2005, Como.

[67] Letizia Jaccheri, Sandro Morasca, "On the Importance of Dialogue with Industry about Software Engineering Education", Third International Summit on Software Engineering Education (SSEE III), 20 maggio 2006, Shanghai (Cina).

[68] Sandro Morasca, "Courses for Software Professionals as Two-way Communication Channels between Academia and Industry", Third International Summit on Software Engineering Education (SSEE III), 20 maggio 2006, Shanghai (Cina).

[69] Austen Rainer, Marcus Ciolkowski, Dietmar Pfahl, Barbara Kitchenham, Sandro Morasca, Matthias M. Müller, Guilherme Horta Travassos, Sira Vegas, "Teaching Empirical Methods to Undergraduate Students Working Group Results", Empirical Software Engineering Issues, 26-30 giugno 2006 Dagstuhl Castle (Germania), pagg. 158-162, IEEE Computer Society Press, ISBN/ISSN: 978-3-540-71300-5.

[70] Sandro Morasca, "On the Assessment of the Mean Failure Frequency of Software in Late Testing", ICSEA'06, 29 ottobre - 3 novembre 2006, Tahiti (Polinesia Francese), pagg. 15-20, IEEE Computer Society Press, ISBN/ISSN: 0-7695-2703-5.

[71] Davide Taibi, Luigi Lavazza, Sandro Morasca, "OpenBQR: a framework for the assessment of OSS", Third International Conference on Open Source Systems (OSS 2007), in IFIP International Federation for Information Processing, Volume 234, Open Source Development, Adoption and Innovation, eds. J. Feller, B. Fitzgerald, W. Scacchi, A. Sillitti, 11-14 giugno 2007, Limerick (Irlanda), pagg. 173-186, ISBN: 13: 978-0-387-72485-0.

[72] Sandro Morasca, Alberto Sillitti, "1st International Workshop on Trust in Open Source Software (TOSS)", Third International Conference on Open Source Systems (OSS 2007), in IFIP International Federation for Information Processing, Volume 234, Open Source Development, Adoption and Innovation, eds. J. Feller, B. Fitzgerald, W. Scacchi, A. Sillitti, 11-14 giugno 2007, Limerick (Irlanda), pagg. 371-373, ISBN: 13: 978-0-387-72485-0.

[73] José A. Cruz-Lemus, Marcela Genero, Sandro Morasca, Mario Piattini, "Assessing the Understandability of UML Statechart Diagrams with Composite States in an Industrial Environment", Quality of Information Systems (QoIS 2007), 5-9 novembre 2007, Auckland (Nuova Zelanda), LNCS 4802, pagg. 213 - 222, ISSN: 0302-9743.

[74] Sandro Morasca, "Subjective Assessment of the Mutual Influence of ISO9126 Software Qualities: an Empirical Study", 20th International Conference on Software Engineering and Knowledge Engineering (SEKE 2008), 1 - 3 luglio 2008, Redwood City (USA), pagg. 297 - 302, ISBN: 1-891706-22-5.

[75] Davide Taibi, Vieri del Bianco, Davide Dalle Carbonare, Luigi Lavazza, Sandro Morasca, "Towards The Evaluation of OSS Trustworthiness: Lessons Learned From The Observation of Relevant OSS Projects", in Proceedings di OSS 2008, 7 - 10 settembre 2008, Milano, pagg. 389 - 395, ISBN: 978-0-387-09683-4.

[76] Sandro Morasca, "Refining the Axiomatic Definition of Internal Software Attributes", Second International Symposium on Empirical Software Engineering and Measurement (ESEM 2008), 8 - 10 ottobre 2008, Kaiserslautern (Germania), pagg. 188 - 197, ISBN: 978-1-59593-971-5.

[77] Sandro Morasca, Davide Taibi, Davide Tosi, "Towards Certifying the Testing Process of Open-Source Software: new challenges or old methodologies?", FLOSS 2009, 16 maggio 2009, Vancouver (Canada).

[78] Davide Taibi, Vieri del Bianco, Luigi Lavazza, Sandro Morasca, "Quality of Open Source Software: the QualiPSo Trustworthiness Model", in Proceedings di OSS 2009, 4-5 giugno 2009, Skövde, Svezia, pagg. 199 - 212, ISBN: 978-3-642-02031-5.

[79] Luigi Lavazza, Sandro Morasca, Davide Taibi, Davide Tosi, "Verso un Processo di Certificazione dei Portali Web di Progetti Software Open Source", III Conferenza Italiana sul Software Libero (CONFSL '09). Bologna, Giugno 2009.

[80] Sandro Morasca, "A Probability-based Approach for Measuring External Attributes of Software Artifacts ", accettato per pubblicazione ad ESEM 2009, 15-16 ottobre 2009, Lake Buena Vista (USA).

5.5 Rapporti interni

[81] Sandro Morasca, Mauro Pezzè, "A Net-based Model for Time-dependent Systems", Politecnico di Milano - Dipartimento di Elettronica, Rapporto interno n 89-005, 1989.

[82] Sandro Morasca, "Using Information to Measure Software Complexity", Politecnico di Milano - Dipartimento di Elettronica, Rapporto interno n 89-022, 1989.

[83] Sandro Morasca, "The Proposal of a High-level Stochastic Petri Net", Politecnico di Milano - Dipartimento di Elettronica, Rapporto Interno n 90-005, 1990.

[84] Lionel Briand, Khaled El Emam, Sandro Morasca, "Theoretical and Empirical Validation of Software Product Measures", Politecnico di Milano - Dipartimento di Elettronica, Rapporto interno n 96-044, 1996.

[85] Lionel Briand, Khaled El Emam, Sandro Morasca, "Measurement Theory, Scales, and Statistics in Software Engineering", Politecnico di Milano - Dipartimento di Elettronica, Rapporto interno n 96-045, 1996.

[86] Sandro Morasca, Federico Macchi, Marco Grigoletti, Cristiano Gusmeroli, "Goal-Driven Measurement in a Maintenance Project", Politecnico di Milano - Dipartimento di Elettronica, Rapporto interno n 96-047, 1996.

5.6 Tesi di laurea e di dottorato

[87] Roberto Lecciso, Stefano Mainetti, Sandro Morasca, "Metriche del software: un'esposizione critico-sperimentale e un'applicazione al Pascal", tesi di laurea in Ingegneria Elettronica, Politecnico di Milano, discussa il 19 luglio 1985.

[88] Sandro Morasca, "Un modello e un ambiente per la specifica e la verifica di sistemi concorrenti e in tempo reale", tesi di dottorato in Ingegneria Elettronica dell'Informazione e dei Sistemi, Terzo Ciclo, Politecnico di Milano, Dipartimento di Elettronica, discussa il 23 settembre 1991.

6 Indirizzo

Sandro Morasca
Università degli Studi dell'Insubria
Dipartimento di Informatica e Comunicazione
Via Carloni 78
22100 Como
tel.: 031/2386228
FAX: 031/2386119
e-mail: sandro.morasca@uninsubria.it
Como, 27 luglio 2009