Curriculum Vitae et Studiorum

Marco Bernardo

Dati Personali

Nome e cognome: Marco Bernardo.

Luogo e data di nascita: [redacted]

Codice fiscale: [redacted]

Recapiti: Università degli Studi di Urbino Carlo Bo
Dipartimento di Scienze Pure e Applicate
Sezione di Scienze e Tecnologie dell'Informazione
Piazza della Repubblica 13, 61029 Urbino (PU)

(telefono: +39-0722-304416, e-mail: marco.bernardo@uniurb.it,
fax: +39-0722-304423, url: http://www.sti.uniurb.it/bernardo/).

Titoli di Studio

Feb 1999 Dottorato di Ricerca in Informatica, X ciclo, conseguito presso il consorzio costituito dalle Università di Bologna (sede amministrativa), Padova e Venezia. Titolo della tesi: *Theory and Application of Extended Markovian Process Algebra.* Relatore: Prof. R. Gorrieri. Controrelatore: Prof. E. Brinksma. Commissione: Prof. M. Bonuccelli, Prof. R. De Nicola, Prof.ssa P. Inverardi.

Mar 1994 Laurea in Scienze dell'Informazione conseguita presso l'Università di Bologna. Titolo della tesi: *Verso l'Integrazione di Modelli di Concorrenza Stocastici.* Relatore: Prof. R. Gorrieri. Correlatore: Prof. L. Donatiello. Controrelatore: Prof. C. Maioli. Voto di laurea: 110/110 e lode.

Lug 1989 Diploma di Ragioniere Programmatore conseguito presso l'Istituto Tecnico Commerciale Statale "Pier Crescenzi" di Bologna. Voto di maturità: 60/60.

Attività Professionale

Mag 2018 - Professore Ordinario, settore scientifico-disciplinare INF/01 – Informatica, settore concorsuale 01/B1 – Informatica, presso il Dipartimento di Scienze Pure e Applicate dell'Università di Urbino.

Nov 2001 - Apr 2018 Professore Associato, confermato dal novembre 2004, settore scientifico-disciplinare INF/01 – Informatica, settore concorsuale 01/B1 – Informatica, presso il Dipartimento di Scienze Pure e Applicate (precedentemente Dipartimento di Scienze di Base e Fondamenti, Dipartimento di Matematica Fisica e Informatica, Istituto di Scienze e Tecnologie dell'Informazione nonché Facoltà di Scienze e Tecnologie, Facoltà di Scienze Matematiche Fisiche e Naturali) dell'Università di Urbino.

Nov 1999 - Ott 2001 Ricercatore Universitario, settore scientifico-disciplinare K05B – Informatica, presso il Dipartimento di Informatica, nell'ambito della Facoltà di Scienze Matematiche Fisiche e Naturali, dell'Università di Torino.

Mag 1999 - Ott 1999 Assegnista di Ricerca presso il Dipartimento di Scienze dell'Informazione dell'Università di Bologna.

1 Attività di Ricerca

1.1 Argomenti Trattati e Risultati Ottenuti

L'attività di ricerca di Marco Bernardo si svolge nell'ambito dei metodi formali dell'informatica ed è finalizzata allo sviluppo di metodologie per la progettazione di sistemi di elaborazione, di comunicazione e software sulla base dei seguenti principi:

- formalità – utilizzo di rigorosi fondamenti matematici per definire linguaggi di modellazione, così da evitare ambiguità e consentire l'analisi delle descrizioni dei sistemi e la predizione delle loro proprietà;
- espressività – capacità di rappresentare agevolmente concetti inerenti a computazione e comunicazione come pure aspetti funzionali ed extra funzionali;
- usabilità – predisposizione di astrazioni relative a paradigmi, modelli e stili che ricorrono frequentemente nella pratica dello sviluppo dei sistemi, nascondendo i dettagli tecnici;
- efficienza – automazione del processo di sviluppo dei sistemi per sintetizzare i sistemi stessi a partire dalle loro descrizioni formali preservando le proprietà verificate su queste ultime.

L'obiettivo finale di questa attività di ricerca è incrementare la qualità dei sistemi in termini di affidabilità e prestazioni e, al tempo stesso, ridurre i costi di sviluppo e manutenzione dei sistemi derivanti dalla scoperta tardiva di errori di progettazione.

Tale attività di ricerca è inquadrabile nelle seguenti aree dell'informatica:

- semantica dei linguaggi di programmazione concorrente;
- algebre di processi, equivalenze comportamentali, logiche modali;
- sistemi di transizione etichettati e reti di Petri;
- modellazione e verifica di sistemi concorrenti e distribuiti;
- valutazione delle prestazioni di sistemi e reti di calcolatori;
- fondamenti di architettura e ingegneria del software;
- supporto automatico per lo sviluppo di software basato su modelli;

e ha dato luogo a:

- l'algebra di processi stocastica **EMPA$_{gr}$** – Extended Markovian Process Algebra, assieme alla sua semantica interleaving basata su catene di Markov a tempo continuo con reward e alla sua semantica concorrente basata su reti di Petri stocastiche generalizzate, ed estensioni stocastiche di equivalenze bisimulazione e testing, illustrate nella tesi di dottorato [127] e nelle pubblicazioni [44, 45, 40, 76, 31];
- il linguaggio di descrizione architetturale **ÆMILIA** basato su EMPA$_{gr}$, assieme a diverse metodologie orientate alle componenti per rilevare incongruenze architetturali di natura funzionale ed extra funzionale sulla base di equivalenze comportamentali e reti di code, nonché per generare codice corretto per costruzione, come illustrato nel libro [1] e nella pubblicazione [36];
- lo strumento di modellazione e verifica **TwoTowers**, che supporta la descrizione di sistemi in ÆMILIA e l'analisi di descrizioni di sistemi attraverso model checking, equivalence checking, information flow security analysis e performance evaluation (reward Markov chain solution e discrete event simulation), come illustrato nella tesi di dottorato [127] e nelle pubblicazioni [88, 84];
- il metamodello **ULTRAS** – Uniform Labeled Transition System, che offre una visione unificante di numerosi modelli comportamentali ed equivalenze comportamentali apparsi nella letteratura dell'informatica teorica e ha originato nuove equivalenze comportamentali con proprietà notevoli in presenza di nondeterminismo e aspetti quantitativi, come illustrato nelle pubblicazioni [34, 59, 28, 2, 33, 30, 32].

Nelle prossime sezioni verranno descritti i risultati teorici ottenuti seguendo l'evoluzione cronologica della suddetta attività di ricerca, nonché l'applicazione pratica dei risultati in strumenti software e casi di studio.

1.1.1 Integrazione di Approcci Formali alla Modellazione Prestazionale

Sebbene numerose tecniche di descrizione formale e relativi strumenti software siano stati sviluppati per supportare la verifica di proprietà funzionali di sistemi complessi, fino agli inizi degli anni 1980 minore attenzione è stata dedicata in letteratura alle caratteristiche temporali e quindi agli aspetti prestazionali.

Per superare gli inconvenienti che ne derivano, in [127, 50] viene proposto un approccio formale finalizzato ad integrare la modellazione e l'analisi di proprietà sia funzionali che prestazionali. L'obiettivo dell'approccio è di permettere la valutazione delle prestazioni di un sistema sin dalle fasi iniziali del suo sviluppo, onde evitare che la tardiva scoperta di inefficienze faccia crescere i costi di progettazione e, al tempo stesso, garantire che proprietà funzionali e proprietà prestazionali siano valutate su modelli del sistema coerenti tra loro. Ulteriore obiettivo dell'approccio è di mettere a disposizione del progettista un opportuno supporto linguistico che consenta la descrizione composizionale di sistemi complessi e l'applicazione di tecniche di analisi efficienti, riusando auspicabilmente strumenti software già esistenti.

L'approccio integrato, ispirato al lavoro di Olderog, si basa su algebre di processi stocastiche e reti di Petri stocastiche, così da sfruttarne i punti di forza complementari. L'approccio si divide in due fasi. Nella prima fase il progettista descrive il sistema tramite una famiglia di equazioni di un'algebra di processi stocastica. In questa fase il progettista ha quindi a disposizione i meccanismi di composizione, astrazione ed equivalenza tipici delle algebre di processi. Dalla rappresentazione algebrica del sistema viene automaticamente prodotto il relativo modello semantico integrato sotto forma di sistema di transizione etichettato sia coi nomi che con le durate delle azioni. Questo modello integrato può poi essere proiettato su un modello semantico funzionale (sistema di transizione etichettato solo coi nomi) e un modello semantico prestazionale (catena di Markov) che sono coerenti tra loro ed analizzabili separatamente tramite strumenti software già esistenti che implementano tecniche di analisi ben note (equivalence checking, model checking, analisi di sicurezza, soluzione di catene di Markov con reward, simulazione ad eventi discreti).

Nella seconda fase, dalla descrizione algebrica viene automaticamente ricavata una rete di Petri stocastica equivalente. Quest'ultima può essere sfruttata dal punto di vista dell'efficienza conducendo attraverso strumenti software già esistenti un'analisi strutturale delle proprietà qualitative e quantitative (calcolo di invarianti e di limiti prestazionali), che quindi non necessita della costruzione dello spazio degli stati.

1.1.2 Algebre di Processi Stocastiche

Espressività Nell'ambito dell'approccio integrato descritto al punto precedente, in [127, 50, 51, 100] viene sviluppata una nuova algebra di processi markoviana. Essa viene chiamata EMPA_{gr}, acronimo di *Extended Markovian Process Algebra with generative-reactive synchronizations*, ed è ispirata alle algebre di processi markoviane TIPP di Herzog ed altri e PEPA di Hillston. Poichè EMPA_{gr} consente di associare durate esponenzialmente distribuite alle azioni, in virtù della proprietà di assenza di memoria delle distribuzioni esponenziali la semantica può essere definita nel consueto stile interleaving e il modello stocastico sottostante risulta essere una catena di Markov a tempo continuo.

EMPA_{gr} si propone di raggiungere il potere espressivo delle reti di Petri stocastiche generalizzate (GSPN) proposte da Ajmone Marsan, Balbo e Conte. A tale scopo, oltre alle azioni esponenzialmente temporizzate, EMPA_{gr} viene dotata di azioni immediate aventi durata nulla e priorità e pesi associati ad esse, che sono utili per rappresentare eventi di durata trascurabile come pure scelte basate su priorità e probabilità. Inoltre, per avere una forma naturale di sincronizzazione tra azioni, come PEPA viene dotata di azioni passive aventi durata non specificata e pesi associati ad esse.

Secondo la terminologia di van Glabbeek, Smolka e Steffen, in EMPA_{gr} la scelta tra azioni esponenzialmente temporizzate – basata sulla race policy – e la scelta tra azioni immediate – basata sulla preselection policy – è di tipo generativo, mentre la scelta tra azioni passive è di tipo reattivo, cioè i pesi vengono considerati solo per scegliere tra azioni passive aventi lo stesso nome. L'unica forma di sincronizzazione ammessa è quella di tipo generativo-reattivo, in cui possono partecipare più azioni aventi lo stesso nome a patto che

al più una di esse sia esponenzialmente temporizzata o immediata. Questo semplice e intuitivo vincolo sulla sincronizzazione risulta non impedire la simulazione di meccanismi di comunicazione più complessi.

Oltre alla semantica interleaving, la quale produce modelli semantici integrati nella forma di sistemi di transizione etichettati, grazie al suo potere espressivo per $\mathrm{EMPA}_{\mathrm{gr}}$ viene sviluppata anche una semantica che associa GSPN ai termini algebrici. Questa semantica è ispirata all'approccio di Degano, De Nicola e Montanari basato sull'idea di codificare le posizioni (rispetto agli operatori statici) dei sottotermini sequenziali all'interno dei posti della rete. Le transizioni della rete vengono poi generate tramite regole induttive simili a quelle per il modello semantico integrato interleaving. La semantica basata su GSPN è dimostrata essere corretta rispetto alla semantica integrata interleaving dei termini di $\mathrm{EMPA}_{\mathrm{gr}}$. Rilevanti sono la gestione delle normalizzazioni delle durate delle transizioni delle GSPN prodotte e la reinterpretabilità a livello algebrico dei risultati dell'analisi condotta sulle GSPN generate.

In [29] viene fatto uno studio nell'ambito delle algebre di processi markoviane per stabilire il potere espressivo relativo dei linguaggi nei quali la durata delle azioni è espressa all'interno delle azioni stesse (come TIPP, PEPA ed $\mathrm{EMPA}_{\mathrm{gr}}$) e i linguaggi nei quali le azioni sono istantanee e il passaggio del tempo è espresso separatamente (come IML), nei tre casi in cui l'esecuzione è urgente per tutte le azioni, ritardabile per tutte le azioni o urgente solo per le azioni interne. Attraverso la definizione tra le due classi di linguaggi di una funzione di codifica diretta e una funzione di codifica inversa che preservano la bisimilarità markoviana forte sotto determinate condizioni, viene dimostrato che le due diverse tipologie sono in larga misura riconciliabili, anche se i linguaggi della seconda classe sono più espressivi perché consentono di descrivere anche scelte non-deterministiche. Lo studio si occupa inoltre delle algebre di processi in cui il tempo è rappresentato tramite durate deterministiche anziché stocastiche, per le quali Corradini aveva già definito una funzione di codifica diretta che preserva la bisimilarità temporizzata forte; viene dimostrato che sotto certi vincoli è possibile definire anche una funzione di codifica inversa, che però preserva soltanto la bisimilarità temporizzata debole.

Equivalenze Comportamentali Nell'ambito delle algebre di processi stocastiche, le equivalenze comportamentali hanno l'obiettivo di individuare termini che denotano sistemi aventi lo stesso comportamento sia dal punto di vista funzionale che dal punto di vista prestazionale.

In [127, 51, 100] viene sviluppata per $\mathrm{EMPA}_{\mathrm{gr}}$ un'equivalenza comportamentale che viene chiamata bisimilarità markoviana estesa, in quanto estende la bisimilarità markoviana di Hillston mediante la gestione di azioni immediate e passive. Due termini sono posti in relazione da questa equivalenza qualora essi siano in grado di imitare passo dopo passo l'uno il comportamento funzionale e prestazionale dell'altro. La bisimilarità markoviana estesa viene dimostrata essere una congruenza rispetto a tutti gli operatori di $\mathrm{EMPA}_{\mathrm{gr}}$, permettendo quindi la manipolazione composizionale dei termini senza alterarne il significato. Più precisamente, essa risulta essere la più grande congruenza contenuta nell'intersezione della bisimilarità puramente funzionale di Milner e della lumpability ordinaria per catene di Markov. La bisimilarità markoviana estesa risulta inoltre avere un'assiomatizzazione corretta e completa, la quale evidenzia le leggi equazionali che caratterizzano l'equivalenza stessa.

In [101, 40, 77, 37, 73, 70, 75] viene introdotta per una generica algebra di processi markoviana con azioni esponenzialmente temporizzate e passive un'equivalenza comportamentale che viene chiamata equivalenza testing markoviana, la quale è ispirata all'equivalenza testing probabilistica di Cleaveland, Smolka ed altri. Due termini sono posti in relazione da questa equivalenza qualora essi siano in grado di superare un test arbitrario – formalizzato anch'esso attraverso un termine – entro il medesimo tempo medio e con la medesima probabilità. L'equivalenza testing markoviana, che è meno restrittiva della bisimilarità markoviana, viene dimostrata essere utile dal punto di vista prestazionale in quanto induce un'aggregazione esatta a livello di catena di Markov che viene chiamata T-lumpability, la quale non era precedentemente nota in letteratura ed è meno restrittiva della lumpability ordinaria. L'equivalenza testing markoviana viene provata avere una caratterizzazione alternativa basata su tracce estese, che quindi astrae dal concetto di test, e poter essere indifferentemente definita considerando le distribuzioni di probabilità che quantificano i tempi di esecuzione dei test o i rispettivi valori attesi. L'equivalenza testing markoviana risulta inoltre essere una congruenza rispetto ai tipici operatori delle algebre di processi, possedere un'assiomatizzazione corretta e completa e avere una caratterizzazione logica modale.

In [40, 37] viene studiata per la stessa algebra di processi l'equivalenza a tracce markoviana definita da Wolf, Baier ed altri. Due termini sono posti in relazione da questa equivalenza qualora essi siano in grado di eseguire una traccia arbitraria entro il medesimo tempo medio e con la medesima probabilità. L'equivalenza a tracce markoviana, la quale è meno restrittiva dell'equivalenza testing markoviana, viene dimostrata godere di proprietà analoghe a quelle dell'equivalenza testing markoviana, soprattutto per quanto riguarda l'esattezza dell'aggregazione indotta a livello di catena di Markov, che coincide con la T-lumpability. L'unica differenza di rilievo è che purtroppo l'equivalenza a tracce markoviana non risulta essere una congruenza rispetto all'operatore di composizione parallela.

In [76] viene sviluppata per $\mathrm{EMPA}_{\mathrm{gr}}$ una variante debole della bisimilarità markoviana estesa, che astrae dalle azioni interne immediate in quanto queste non producono effetti visibili e vengono eseguite in un tempo trascurabile. Sotto certe condizioni relative all'uso dei livelli di priorità associati alle azioni, tale equivalenza viene dimostrata essere una congruenza rispetto a tutti gli operatori di $\mathrm{EMPA}_{\mathrm{gr}}$ e avere un'assiomatizzazione corretta e completa.

In [31] viene introdotta per una generica algebra di processi markoviana una variante debole della bisimilarità markoviana di Hillston, la quale astrae dalle azioni interne esponenzialmente temporizzate sulla base del principio che, allo stato stazionario, nessuna differenza è osservabile tra una sequenza di azioni interne esponenzialmente temporizzate e una singola azione interna esponenzialmente temporizzata avente la stessa durata media e la stessa probabilità di esecuzione della sequenza. Viene evidenziata l'esistenza di un rapporto antitetico tra composizionalità ed esattezza. La definizione più naturale di questa equivalenza risulta indurre una pseudo-aggregazione esatta allo stato stazionario a livello di catena di Markov che viene chiamata W-lumpability, la quale non era precedentemente nota in letteratura ed è meno restrittiva della lumpability ordinaria, ma è una congruenza solo nell'ambito dei termini sequenziali, sui quali ammette un'assiomatizzazione corretta e completa. Per ottenere una congruenza rispetto alla composizione parallela, la definizione viene generalizzata in modo da tenere conto dell'intreccio di computazioni interne di componenti sequenziali distinte, ma la conseguente pseudo-aggregazione che viene chiamata GW-lumpability risulta essere esatta allo stato stazionario solo per una classe di termini concorrenti. In [55] l'approccio viene applicato alla bisimilarità debole per automi di Markov e confrontato con la bisimilarità debole per sistemi di transizione etichettati con tempo deterministico.

Misure Prestazionali In [127, 44, 94] viene preso in esame il problema di specificare le misure prestazionali di interesse per sistemi modellati con $\mathrm{EMPA}_{\mathrm{gr}}$. L'idea è di adattare al contesto algebrico la tecnica dei reward tradizionalmente impiegata nell'ambito delle catene di Markov. Un reward è un numero reale che esprime la velocità alla quale un guadagno o una perdita viene accumulato mentre si soggiorna in un certo stato oppure il guadagno o la perdita ottenuto istantaneamente all'atto dell'esecuzione di una certa transizione. Il valore di una misura prestazionale espressa mediante reward è dato dalla somma pesata delle probabilità dei vari stati e delle frequenze delle varie transizioni, dove i pesi sono i reward associati agli stati e alle transizioni.

Poiché le catene di Markov sono sistemi di transizione etichettati mentre $\mathrm{EMPA}_{\mathrm{gr}}$ è un linguaggio basato su azioni in cui il concetto di stato rimane implicito, in $\mathrm{EMPA}_{\mathrm{gr}}$ i reward vengono specificati associandoli alle azioni e indicando se sono cumulativi o istantanei. Viene dimostrato che la bisimilarità markoviana estesa sviluppata per $\mathrm{EMPA}_{\mathrm{gr}}$ e la relativa assiomatizzazione possono essere facilmente adattate in modo da comprendere anche il trattamento composizionale dei reward cumulativi ed istantanei senza alterare il valore delle misure prestazionali che essi descrivono.

Distribuzioni Generali L'uso combinato di azioni esponenzialmente temporizzate e azioni immediate pesate permette di approssimare molte distribuzioni di uso frequente tramite distribuzioni phase-type. Inoltre l'introduzione dei dati nella sintassi consente in qualche modo di gestire durate arbitrariamente distribuite. Tuttavia, è opportuno permettere l'espressione diretta di azioni aventi durate arbitrariamente distribuite e l'analisi dei relativi sistemi tramite tecniche numeriche.

Questo tema viene inizialmente affrontato in [112] – e poi sviluppato da Bravetti nella sua tesi di dottorato – tramite la definizione di una nuova algebra di processi stocastica chiamata GSMPA, acronimo di *Generalized Semi-Markovian Process Algebra*. Poiché la proprietà di assenza di memoria non può più essere sfruttata,

la semantica integrata di GSMPA viene definita nello stile ST di Van Glabbeek e Vaandrager anziché nello stile interleaving, così da identificare in maniera naturale gli stati in cui ciascuna azione inizia o termina.

I modelli stocastici sottostanti risultano essere dei processi semi-markoviani generalizzati (GSMP) invece di catene di Markov. Tali processi stocastici possono essere analizzati non solo per via simulativa, ma anche per via numerica qualora certe condizioni di insensibilità siano soddisfatte, le quali permettono la sostituzione di distribuzioni generali con distribuzioni esponenziali aventi gli stessi valori attesi. GSMPA viene dotata di un'equivalenza comportamentale nello stile bisimulazione che tiene conto del nuovo modello stocastico, la quale risulta essere una congruenza rispetto a tutti gli operatori.

1.1.3 Rappresentazioni Compatte di Specifiche in Algebre di Processi

Modelli Simbolici In [127, 42] viene affrontato il problema di modellare sistemi dove i dati ricoprono un ruolo non trascurabile, attraverso algebre di processi che ammettono variabili tipate e azioni di input e di output per consentire lo scambio di valori tra termini durante le sincronizzazioni. L'obiettivo è quello di generare modelli semantici finiti anche quando i dati appartengono a domini infiniti. Ciò viene ottenuto mediante opportune regole semantiche simboliche che producono sistemi di transizione ulteriormente etichettati con guardie booleane e assegnamenti. Questi modelli rappresentano versioni compatte di modelli simbolici ispirati a quelli di Lin, ai quali si applicano gli algoritmi di verifica di bisimilarità proposti da Li e Chen. Le regole simboliche sono dimostrate essere corrette rispetto alle regole concrete e all'ordine di valutazione degli assegnamenti, come pure produrre modelli simbolici che soddisfano un criterio di ottimalità dal punto di vista della loro compattezza.

Reti di Petri In [127, 52, 98, 45] viene affrontato il problema di generare reti di Petri compatte a partire da termini algebrici. In prima istanza l'idea è di generare un posto nella rete per ogni sottotermine sequenziale a prescindere dalla sua posizione (rispetto agli operatori statici). Con tale approccio tutte le transizioni della rete possono essere generate con una singola regola. Le reti di Petri risultanti sono più compatte e talvolta finite anziché infinite, però contengono archi inibitori e contestuali necessari alla costruzione che non svolgono alcun ruolo una volta ottenute le reti stesse. Questo inconveniente viene successivamente superato riconoscendo che è sufficiente decorare opportunamente le azioni all'interno dei posti della rete ai fini di evitare l'introduzione di archi inibitori e contestuali. Tutte le semantiche a rete proposte risultano corrette rispetto alla semantica interleaving.

1.1.4 Linguaggi di Descrizione Architetturale Basati su Algebre di Processi

Notazioni Orientate alle Componenti e Tipi Architetturali Nella fase di progettazione architetturale di un sistema software l'attenzione è rivolta alla scelta delle componenti che dovranno costituire il sistema e al modo in cui queste componenti dovranno interagire, piuttosto che alle questioni legate all'individuazione delle strutture dati e degli algoritmi più appropriati. Pertanto in questa fase risulta particolarmente adeguato e promettente l'uso di metodi formali composizionali come le algebre di processi.

In [46, 41, 36] vengono discusse alcune linee guida metodologiche che portano allo sviluppo di PADL, un linguaggio di descrizione architetturale basato su algebre di processi per la modellazione composizionale, grafica, gerarchica e controllata di sistemi complessi a topologia statica. Tale linguaggio si compone di una notazione testuale ispirata ai linguaggi Wright di Allen e Garlan e Darwin/FSP di Kramer e Magee, come pure di una notazione grafica ispirata ai grafi di flusso di Milner.

PADL permette di descrivere famiglie di sistemi – chiamate tipi architetturali – accumunati dal comportamento osservabile delle loro componenti e dai vincoli imposti alle loro topologie. Ogni tipo architetturale, il quale è un'approssimazione di uno stile architetturale, viene specificato in PADL attraverso i suoi parametri, i tipi dei suoi elementi architetturali (componenti e connettori), la sua topologia e le sue eventuali modificazioni comportamentali.

All'interno di un tipo architetturale, ciascun tipo di elemento architetturale viene definito attraverso i suoi parametri, il suo comportamento e le sue interazioni. Il comportamento viene descritto tramite una famiglia di termini sequenziali dell'algebra di processi, mentre le interazioni col resto del sistema vengono

descritte tramite una famiglia di azioni presenti nel comportamento. A ciascuna interazione sono associati dei qualificatori che stabiliscono se si tratta di un'interazione di input o di output come pure il grado di sincronicità (sincrona, semi-sincrona, asincrona) e il grado di molteplicità (uno-a-uno, uno-a-molti inclusiva come broadcast, uno-a-molti selettiva come server-clients) delle comunicazioni in cui può essere coinvolta.

La topologia dichiara le istanze di ogni tipo di elemento architetturale presenti nel sistema coi rispettivi parametri effettivi, le interazioni di tali istanze dette architetturali che fungono da interfaccia per l'intero sistema (utili per la modellazione gerarchica) e le connessioni tra le interazioni di tali istanze in conformità ai qualificatori specificati. La topologia stessa può essere parametrizzata, perché sono a disposizione meccanismi di indicizzazione per la dichiarazione iterativa di più istanze di uno stesso tipo di elemento architetturale e per la dichiarazione concisa delle loro connessioni.

Infine, la sezione di modificazioni comportamentali, la quale è opzionale, consente di esprimere il fatto che certe azioni debbono essere rese inosservabili, impedite o cambiate di nome. Questo può essere utile per condurre determinati tipi di analisi, ad esempio nel campo della sicurezza [38, 35].

Le varie istanze di un tipo architetturale (cioè i sistemi che compongono la famiglia) sono ottenute invocando il tipo architetturale e passandogli elementi di tipi architetturali effettivi, una topologia effettiva e modificazioni comportamentali effettive che siano conformi alle corrispondenti entità formali definite nella specifica PADL del tipo architetturale. Dal punto di vista comportamentale, un tipo di elemento architetturale effettivo è conforme ad un tipo di elemento architetturale formale se le famiglie di termini algebrici che specificano i rispettivi comportamenti sono debolmente bisimili nel senso di Milner. Dal punto di vista topologico, una topologia effettiva è conforme ad una topologia formale se ne rappresenta una variazione esogena (che ha luogo presso le interazioni architetturali replicando alcune parti della topologia formale), una variazione endogena (che ha luogo all'interno della topologia formale attraverso il meccanismo di parametrizzazione della topologia formale stessa) o una variazione and/or (che ha luogo all'interno della topologia formale variando il numero di elementi architetturali dei tipi opportuni connessi a interazioni uno-a-molti).

Analisi e Diagnostica Orientate alle Componenti Dal punto di vista delle algebre di processi, creare un linguaggio di descrizione architetturale come PADL che è basato su di esse può essere visto come un tentativo di forzare il progettista a modellare famiglie di sistemi in un modo più controllato, che in particolare dia risalto ai principali concetti architetturali. Questo zucchero sintattico da un lato incrementa l'usabilità delle algebre di processi in quanto ne nasconde gli aspetti tecnici, ma dall'altro lato non è sufficiente a creare un linguaggio di descrizione architetturale che possa essere utile in pratica. È infatti necessario riuscire ad analizzare le proprietà dell'architettura dei sistemi descritti se si vuole essere in grado di effettuare delle scelte opportune nelle fasi iniziali della progettazione.

La semantica di una descrizione PADL viene data per traduzione in un'algebra di processi, componendo in parallelo – secondo le connessioni dichiarate – i termini che costituiscono il comportamento delle singole istanze dei tipi di elementi architetturali ed applicando le eventuali variazioni comportamentali che sono state definite. Di conseguenza tutte le tecniche di analisi sviluppate per le algebre di processi sono applicabili ai sistemi modellati con PADL.

Tuttavia, ciò che serve al livello architetturale è una tecnica per verificare che, quando si assemblano componenti che sono corrette se considerate isolatamente, il sistema risultante sia anch'esso corretto. Poiché i sistemi che vengono costruiti sono formati da numerose componenti, è importante che la tecnica sia efficiente e fornisca informazioni diagnostiche utili nel caso di rilevamento di incongruenze architetturali, al fine di individuare in maniera puntuale le componenti che ne sono responsabili.

Ispirandosi a precedenti lavori di Allen e Garlan come pure di Inverardi ed altri, in [46, 41, 36] PADL viene dotato di una tecnica basata su verifica di equivalenza per l'analisi e la diagnosi orientate alle componenti di incongruenze architetturali. Tale tecnica opera su tipi architetturali aventi una topologia statica arbitraria – sia aciclica che ciclica – e scala rispetto alle loro istanze comportamentali ed estensioni topologiche. In caso di rilevamento di incongruenza, le formule di logica modale che spiegano la violazione dell'equivalenza vengono usate come informazioni diagnostiche per localizzare la sorgente dell'incongruenza.

Aspetti Prestazionali: Reti di Code e Misure Prestazionali Un'evoluzione di PADL è ÆMILIA [7, 6], un linguaggio di descrizione architetturale che supporta l'analisi prestazionale in quanto basato su $EMPA_{gr}$. ÆMILIA è dotato degli stessi meccanismi di PADL. Oltre a possedere gli stessi modelli semantici della relativa traduzione in $EMPA_{gr}$, una descrizione in ÆMILIA può avere ad essa associato anche un modello a rete di code. Trattandosi di un modello prestazionale orientato alle componenti, esso consente un più efficiente calcolo di misure prestazionali standard e una più facile interpretazione del valore di tali misure al livello della descrizione architetturale.

Al fine di essere efficace, un linguaggio orientato alle componenti come ÆMILIA deve essere dotato di una notazione orientata alle componenti per esprimere le misure prestazionali di interesse. A tale scopo in [39] viene definito MSL, acronimo di *Measure Specification Language*. Esso è basato su una combinazione di reward e logica e mette a disposizione del progettista un apposito meccanismo per definire una misura prestazionale indicando semplicemente le interazioni o le parti del comportamento delle componenti che contribuiscono a determinarne il valore. MSL viene interpretato sia su catene di Markov che su algebre di processi markoviane, estendendo precedenti risultati di congruenza e assiomatizzazione ottenuti per la bisimilarità markoviana con reward. Inoltre risulta essere più espressivo rispetto alle tradizionali strutture di reward per notazioni di modellazione con stato implicito. In [66] il linguaggio MSL viene integrato con la logica temporale stocastica CSRL per combinare l'usabilità del primo con l'espressività della seconda.

Generazione Automatica di Codice Corretto per Costruzione La descrizione del progetto architetturale di un sistema software è un documento di riferimento che guida l'intero sviluppo del sistema. Al fine di produrre software di qualità in tempi rapidi, in [36] viene proposto un approccio per la generazione automatica di codice Java multithreaded a partire da specifiche PADL. L'approccio si basa sullo sviluppo di un package Java ispirato ai principali concetti architetturali, il quale implementa un modello di sincronizzazione generale. L'utilizzo di quest'ultimo durante la traduzione automatica di specifiche PADL in programmi Java garantisce la corretta sincronizzazione di thread ed eventuali monitor, come pure la preservazione di certe proprietà precedentemente verificate al livello architetturale. L'implementazione del traduttore PADL2Java viene poi sviluppata da Bontà nella sua tesi di dottorato.

1.1.5 Strumenti Software

TwoTowers Al fine di consentire un'effettiva fruibilità dei linguaggi di descrizione architetturale basati su algebre di processi che sono stati messi a punto, come pure di applicare in maniera automatica alle relative specifiche le varie tecniche di analisi precedentemente indicate, viene implementato TwoTowers [127, 109, 104, 45, 88, 84, 71, 64, 159]. Esso è uno strumento software open source che consente la verifica funzionale (equivalence checking e model checking), l'analisi di sicurezza (information flow analysis) e la valutazione prestazionale (soluzione di catene di Markov con reward e simulazione ad eventi discreti) di sistemi di elaborazione, comunicazione e software descritti tramite ÆMILIA estesa con dati.

L'interfaccia grafica di TwoTowers mette a disposizione un compilatore, un equivalence checker, un model checker, un analizzatore della sicurezza e un valutatore delle prestazioni. Il compilatore è incaricato di effettuare il riconoscimento sintattico di specifiche ÆMILIA e di segnalare eventuali errori lessicali, sintattici e semantici statici; in assenza di errori, sulla base della semantica per traduzione di ÆMILIA in $EMPA_{gr}$ viene generato su richiesta il modello semantico operazionale integrato, funzionale o prestazionale oppure soltanto il numero di stati e di transizioni di tale sistema di transizione etichettato.

L'equivalence checker verifica attraverso l'applicazione dell'algoritmo di Kanellakis e Smolka se due specifiche ÆMILIA corrette e a stati finiti sono equivalenti – rispetto a bisimilarità forte di Milner, bisimilarità debole di Milner, bisimilarità markoviana forte o bisimilarità markoviana debole – producendo in caso negativo una formula distinguente espressa in una versione verbosa della logica modale di Hennessy e Milner o sue varianti probabilistiche ottenuta mediante un adattamento dell'algoritmo di Cleaveland.

Il model checker verifica attraverso procedure basate su binary decision diagram dello strumento software NuSMV sviluppato da Cimatti ed altri se un insieme di proprietà funzionali espresse attraverso formule di una versione verbosa della logica temporale LTL sono soddisfatte da una specifica ÆMILIA corretta e a stati finiti, producendo in caso negativo dei controesempi.

L'analizzatore della sicurezza controlla attraverso l'equivalence checker se una specifica ÆMILIA corretta e a stati finiti soddisfa proprietà di non interferenza o non deducibilità su composizione, le quali stabiliscono l'assenza di flussi illegali di informazioni da componenti del sistema ad alta sicurezza verso componenti del sistema a bassa sicurezza; in caso negativo, viene prodotta come diagnostica una formula espressa in una versione verbosa della logica modale di Hennessy e Milner.

Il valutatore delle prestazioni esamina le caratteristiche quantitative di una specifica ÆMILIA corretta e prestazionalmente chiusa. In primo luogo, esso può calcolare la distribuzione della probabilità di stato stazionario – tramite metodo di eliminazione di Gauss o variante adattativa del sovrarilassamento stocastico simmetrico – oppure transitorio – tramite metodo di uniformizzazione – della sottostante catena di Markov. In secondo luogo, esso può calcolare – tramite gli stessi metodi – un insieme di misure prestazionali allo stato stazionario o transitorio definite attraverso reward associati agli stati e alle transizioni della sottostante catena di Markov. In terzo luogo, esso può stimare con un determinato livello di confidenza attraverso simulazione ad eventi discreti la media, la varianza o la distribuzione di un insieme di misure prestazionali specificate attraverso un'estensione dei reward di stato e di transizione; questa tecnica può essere applicata anche nel caso di stati infiniti o distribuzioni generali, le quali vengono rappresentate attraverso lo stesso meccanismo simbolico usato per la gestione dei dati.

Numerosi sono i casi di studio condotti con TwoTowers: alternating bit protocol [127]; ATM switches [127, 107]; audio processing systems [87]; client-server architectures [88]; compiler architectures [6]; compressing proxy systems [46]; cruise control systems [46]; CSMA/CD protocol [127]; dining philosophers [127]; dynamic power management [5]; guest analogy [46]; leader election [91]; multilevel security routing systems [38, 3]; mutual exclusion [127]; pipe-filter architectures [46]; queueing networks [127]; token ring protocol [127]; VoIP systems [49, 47, 43]; web services [96].

1.1.6 Metamodelli Comportamentali

ULTRAS Lo scopo dei metamodelli comportamentali è da un lato di offrire una visione uniforme di modelli diversi che sono apparsi in letteratura (teorie unificanti) e dall'altro di fornire metodologie, risultati e strumenti generali che possono essere istanziati ad un'ampia gamma di modelli comportamentali specifici (servizi di riuso). In particolare, l'attività di sviluppo di una nuova teoria, un nuovo modello o un nuovo calcolo che tenga conto di certi aspetti comportamentali beneficerebbe in modo naturale dell'esistenza di un opportuno metamodello. Ciò dovrebbe ridurre lo sforzo necessario per definire sintassi, semantica ed equivalenze, investigare composizionalità e caratterizzazione equazionali o logiche e sviluppare i relativi algoritmi di verifica.

Nell'ambito dei modelli a stati e transizioni, in [34, 59] viene sviluppato il metamodello ULTRAS, acronimo di *Uniform Labeled Transition System*. Ricordato che ogni transizione di un sistema di transizione etichettato prevede uno stato di partenza e uno stato di destinazione, l'idea alla base del metamodello ULTRAS è quella di trasformare la destinazione in una funzione di distribuzione della raggiungibilità, che ad ogni stato associa il grado di raggiungibilità con quella transizione tratto da un insieme preordinato dotato di minimo (che rappresenta l'irraggiungibilità). Scegliendo opportunamente tali funzioni, è possibile codificare all'interno del metamodello numerosi modelli comportamentali specifici di largo uso come i sistemi di transizione etichettati (cioè i processi puramente nondeterministici), le catene di Markov a tempo discreto etichettate con azioni (cioè i processi puramente probabilistici), le catene di Markov a tempo continuo etichettate con azioni (cioè i processi puramente markoviani), i processi di decisione markoviani con eventuale nondeterminismo interno, i processi di decisione markoviani a tempo continuo con eventuale nondeterminismo interno, gli automi temporizzati (dove il tempo è deterministico), gli automi temporizzati probabilistici e gli automi di Markov.

Sul metamodello ULTRAS vengono altresì definite delle metaequivalenze comportamentali seguendo l'approccio bisimulazione di Milner, l'approccio testing di De Nicola ed Hennessy e l'approccio a tracce della teoria dei linguaggi. Viene mostrato che istanziandole opportunamente si riottengono le equivalenze comportamentali definite in letteratura per le classi di modelli comportamentali specifici sopra menzionati. Inoltre, nel caso particolare dei processi probabilistici con nondeterminismo interno, si ottengono delle nuove equivalenze comportamentali che hanno proprietà interessanti perché sono meno discriminanti delle corrispondenti

equivalenze presenti in letteratura. Diversamente dalla bisimilarità di Segala e Lynch, la nuova bisimilarità per quei processi è caratterizzata dalla logica PML di Larsen e Skou opportunamente reinterpretata, esattamente come la bisimilarità per processi probabilistici senza nondeterminismo interno [30]. Diversamente dalle equivalenze testing di Yi, Larsen e Jonsson e di Segala, la nuova equivalenza testing per quei processi è pienamente compatibile con le equivalenze testing per processi puramente nondeterministici e per processi probabilistici senza nondeterminismo interno [33]. Diversamente dalla equivalenza a tracce di Segala, la nuova equivalenza a tracce per quei processi è una congruenza rispetto alla composizione parallela ed è meno discriminante dell'equivalenza testing a prescindere dalla classe di scheduler utilizzati per risolvere il nondeterminismo [33]. Esistono dunque vari modi di definire le equivalenze comportamentali per i processi probabilistici con nondeterminismo interno, a seconda che si richieda una corrispondenza piena o parziale tra le risoluzioni di nondeterminismo di due processi oppure che si considerino soltanto le probabilità minime e massime di esecuzione delle computazioni all'interno di quelle risoluzioni. Le equivalenze risultanti danno luogo ad uno spettro molto più complesso di quello studiato da Van Glabbeek per i processi puramente nondeterministici, che cambia anche in funzione della classe di scheduler utilizzati [32].

In [28, 2], il metamodello ULTRAS viene rivisitato importando la nozione di risoluzione di nondeterminismo e introducendo quella di semianello coerente con la raggiugibilità, in maniera tale da mettere a punto due diverse definizioni per ciascuna delle metaequivalenze comportamentali (pre-metaequivalenza e post-metaquivalenza), che differiscono per l'ordine di alcuni quantificatori universali. Questo consente di catturare, oltre alle equivalenze che già venivano catturate e a quelle nuove, anche quelle che non venivano in precedenza catturate, nonché di stabilire delle relazioni più precise con altri metamodelli comportamentali quali il metamodello WLTS di Klin e, soprattutto, il metamodello FUTS di De Nicola, Latella, Loreti e Massink. Inoltre, viene studiata la composizionalità rispetto a tipici operatori di algebre di processi per le varie metaequivalenze risultanti. I metarisultati di congruenza per la composizione parallela confermano l'esistenza di una differenza fondazionale nella composizionalità delle semantiche basate su bisimulazione e su tracce, rispettivamente, che era emersa nel contesto specifico dei processi probabilistici e nondeterministici.