

**PROGETTO DI UNITÀ DI RICERCA - MODELLO B**  
**Anno 2008 - prot. 20089M932N\_003**

## 1 - Area Scientifico-disciplinare

01: Scienze matematiche e informatiche 100%

---

## 2 - Coordinatore Scientifico

ROSSI FRANCESCA

*Professore Ordinario*

*Università degli Studi di PADOVA*

*Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI*

*Dipartimento di MATEMATICA PURA E APPLICATA*

---

## 3 - Responsabile dell'Unità di Ricerca

GABRIELLI MAURIZIO

*Professore Ordinario*

20/11/1960

GBBMRZ60S20D612L

*Università degli Studi di BOLOGNA*

*Dipartimento di SCIENZE DELL'INFORMAZIONE*

051 2094500  
(Prefisso e telefono)

051 2094510  
(Numero fax)

[gabbri@cs.unibo.it](mailto:gabbri@cs.unibo.it)

---

## 4 - Curriculum scientifico

### Testo italiano

Maurizio Gabbrielli è professore ordinario di Informatica presso il Dipartimento di Scienze dell'Informazione dell'Alma Mater Studiorum - Università di Bologna. Laureato in Scienze dell'Informazione ha ottenuto il dottorato di ricerca in Informatica presso l'università di Pisa, ha insegnato nelle università di Pisa e Udine ed è stato ricercatore per due anni presso il CWI (Centro per la Matematica e l'Informatica) di Amsterdam. I suoi interessi di ricerca includono i metodi formali di analisi e verifica, la programmazione con vincoli e la teoria della concorrenza. È autore di più di 70 pubblicazioni su libri, riviste e conferenze internazionali, è stato membro del comitato di programma, Conference e Program Chair di varie conferenze internazionali, fra cui POPL, ICLP e PPDP. Ha partecipato a vari progetti di ricerca europei ed italiani ed è stato coordinatore scientifico del progetto PRIN COVER. È membro dell'advisory board della rivista "Theory and Practice of Logic Programming", del board della European Association for Programming Languages and Systems ed è stato presidente dell'associazione italiana di programmazione logica, membro del consiglio direttivo della conferenza ACM PPDP e membro del comitato esecutivo dell'Associazione Internazionale di Programmazione Logica (ALP).

### Testo inglese

Maurizio Gabbrielli is professor of computer science at the Department of Computer Science of the University of Bologna. He received his Phd. in Computer Science in 1992 from the University of Pisa. In 1993-95 he was employed at CWI (Amsterdam), from 1995 to 1998 he was assistant professors at the University of Pisa and from 1998 to 2001 he was associate professor at the University of Udine. His research interests include formal methods for program verification and analysis, constraint programming and concurrency theory. He is author of more than 70 publications on international journals, conference proceedings and books, served as a PC member, conference and program chair of several international workshops and conferences including POPL, ICLP, PPDP. He participated in several Italian and European projects and was the national co-ordinator of the PRIN project COVER. He is currently member of the advisory board of the journal "Theory and Practice of Logic Programming" and of the EAPLS board and has been member of the ALP (Association for Logic Programming) executive committee, president of the Italian association for logic programming and chair of the Steering Committee of the ACM conference PPDP.

## 5 - Pubblicazioni scientifiche più significative del Responsabile dell'Unità di Ricerca

1. GABBRIELLI M., M.C. MEO, P. TACCHELLA (2008). A compositional Semantics for CHR with propagation rules. In: T. Schrijvers, T. Fruehwirth. *Recent Advances in Constraint Handling Rules, LNAI 5388*, p. 119 - 160, BERLIN: Springer-Verlag
2. S. BISTARELLI, GABBRIELLI M., M.C. MEO, F. SANTINI (2008). Timed Soft Concurrent Constraint Programs. In: *Proc. of Coordination Models and Languages, 10th International Conference, COORDINATION 2008*. Oslo, Norway, June 4-6, 2008, BERLIN: Springer-Verlag, vol. LNCS 5052, p. 50 - 66
3. P. TACCHELLA, GABBRIELLI M., M.C. MEO (2007). Unfolding in CHR. In: *Proceedings of the PPDP International conference*. Wroclaw (Poland), July 2007, NEW YORK: ACM Press, p. 179 - 186
4. GABBRIELLI M., S. MARTINI (2006). Linguaggi di programmazione. MILANO: McGraw-Hill Italia, p. 458, ISBN: 88-386-62614-
5. G. DELZANNO, GABBRIELLI M. (2005). Compositional Verification of Asynchronous Processes via Constraint Solving. In: *Proceedings of ICALP 2005*. Lisboa, 2005, HEIDELBERG: Springer-Verlag, vol. LNCS 3580, p. 1239 - 1250
6. G. DELZANNO, GABBRIELLI M., M.C. MEO (2005). A compositional semantics for CHR. In: *Proceedings of PPDP*. Lisboa, 2005, NEW YORK: ACM Press, p. ?
7. GABBRIELLI M., G. GUPTA (EDS.) (2005). *Proceedings of the 21st International Conference on Logic Programming*. HEIDELBERG: Springer-Verlag, p. 454, ISBN: 3-540-29208-X
8. DE BOER F., GABBRIELLI M., MEO M.C. (2004). Proving Correctness of Timed Concurrent Constraint Programs. *ACM TRANSACTIONS ON COMPUTATIONAL LOGIC*, vol. 5(4); p. xx - yy, ISSN: 1529-3785
9. G. DELZANNO, GABBRIELLI M., M.C. MEO. (2003). Compositional verification of infinite state systems. In: *ICLP 03*
10. ETALLE S., GABBRIELLI M., MEO C. (2001). Trasformation of CCP programs. *ACM TRANSACTIONS ON PROGRAMMING LANGUAGES AND SYSTEMS*, ISSN: 0164-0925
11. F.S. DE BOER, GABBRIELLI M., M. C.MEO. (2001). A Temporal Logic for reasoning about Timed Concurrent Constraint Programs. In: *TIME 01*
12. DE BOER F., GABBRIELLI M., GABBRIELLI M. (2000). A Timed Concurrent Constraint Language. *INFORMATION AND COMPUTATION*, vol. 161, ISSN: 0890-5401
13. ETALLE S., GABBRIELLI M. (1998). Partial Evaluation of Concurrent Constraint Languages. *ACM COMPUTING SURVEYS*, vol. 30(3es), ISSN: 0360-0300
14. F.S. DE BOER, GABBRIELLI M., E. MARCHIORI, C. PALAMIDESSI (1997). Proving Concurrent Constraint Programs Correct. *ACM TRANSACTIONS ON PROGRAMMING LANGUAGES AND SYSTEMS*, vol. 19(5); p. 685-725, ISSN: 0164-0925
15. FALASCHI M., GABBRIELLI M., MARRIOTT K., PALAMIDESSI K. (1997). Constraint Logic Programming with Dynamic Scheduling: A Semantics Based on Closure Operators. *INFORMATION AND COMPUTATION*, vol. 137(1); p. 41-67, ISSN: 0890-5401
16. M. FALASCHI, GABBRIELLI M., K. MARRIOTT, C. PALAMIDESSI (1997). Confluence in Concurrent Constraint Programming. *THEORETICAL COMPUTER SCIENCE*, vol. 183, ISSN: 0304-3975
17. S. ETALLE, GABBRIELLI M. (1996). Transformations of CLP Modules. *THEORETICAL COMPUTER SCIENCE*, vol. 166; p. 101--146, ISSN: 0304-3975
18. BOSSI A., GABBRIELLI M., LEVI G., MARTELLI M. (1994). The s-semantics approach: Theory and applications. *JOURNAL OF LOGIC PROGRAMMING*, vol. 19-20; p. 149--197, ISSN: 0743-1066
19. F.S. DE BOER, GABBRIELLI M., E. MARCHIORI, C. PALAMIDESSI (1994). Proving Concurrent Constraint Programs Correct. In: *In Proc. Twentyfirst Annual ACM SIGACT/SIGPLAN Symposium on Principles of Programming Languages (POPL 94)* ACM Press
20. FALASCHI M., GABBRIELLI M., MARRIOTT K., PALAMIDESSI C (1993). Compositional analysis of concurrent constraint languages. In: *Proc. Eighth Annual IEEE Symposium on Logic in Computer Science (LICS 93)*, p. 210-221

## 6 - Elenco dei partecipanti all'Unità di Ricerca

### 6.1 - Componenti

Componenti della sede dell'Unità di Ricerca

n°	Cognome	Nome	Università/Ente	Qualifica	Disponibilità temporale indicativa prevista	
					1° anno	2° anno
1.	GABBRIELLI	Maurizio	Università degli Studi di BOLOGNA	Professore Ordinario	6	6
2.	KIZILTAN	Zeynep	Università degli Studi di BOLOGNA	Ricercatore non confermato	3	3
<b>TOTALE</b>					<b>9</b>	<b>9</b>

Componenti di altre Università / Enti vigilati

n°	Cognome	Nome	Università/Ente	Qualifica	Disponibilità temporale indicativa prevista	
					1°	2°

				anno	anno	
1.	PETTOROSS	Alberto	Università degli Studi di ROMA "Tor Vergata"	Professore Ordinario	6	6
2.	PROIETTI	Maurizio	Consiglio Nazionale delle Ricerche	Primo ricercatore	4	4
<b>TOTALE</b>					<b>10</b>	<b>10</b>

Titolari di assegni di ricerca

n°	Cognome	Nome	Università/Ente	Disponibilità temporale indicativa prevista	
				1° anno	2° anno
1.	DI GIUSTO	Cinzia	Università degli Studi di BOLOGNA	4	4
2.	SENNI	Valerio	Università degli Studi di ROMA "Tor Vergata"	5	5
<b>TOTALE</b>				<b>9</b>	<b>9</b>

Titolari di borse

n°	Cognome	Nome	Università/Ente	Qualifica	Disponibilità temporale indicativa prevista	
					1° anno	2° anno
1.	CAMPORESI	Ferdinanda	Università degli Studi di BOLOGNA	Dottorando	4	4
2.	MAURO	Jacopo	Università degli Studi di BOLOGNA	Dottorando	4	4
3.	VITALE	Antonio	Università degli Studi di BOLOGNA	Dottorando	3	3
<b>TOTALE</b>					<b>11</b>	<b>11</b>

### 6.1 bis Vice-responsabile

*KIZILTAN Zeynep*

### 6.2 - Altro personale

*Nessuno*

### 6.3 - Personale a contratto da destinare a questo specifico Progetto

n°	Tipologia di contratto	Costo previsto	Disponibilità temporale indicativa prevista		Note
			1° anno	2° anno	
1.	Assegnista	20.000	11		
<b>TOTALE</b>		<b>20.000</b>	<b>11</b>	<b>0</b>	

### 6.4 - Dottorati a carico del PRIN da destinare a questo specifico Progetto

*Nessuno*

## 7 - Titolo specifico del Progetto svolto dall'Unità di Ricerca

### Testo italiano

*Tecniche basate su vincoli per la modellazione, la risoluzione e la verifica di problemi complessi.*

### Testo inglese

*Constraints based techniques for modeling, solving and verifying complex problems.*

## 8 - Abstract del Progetto svolto dall'Unità di Ricerca

### Testo italiano

*In questo progetto nell'unità di Bologna intendiamo studiare la modellazione, la soluzione e la verifica di problemi complessi nell'ambito del Service Oriented Computing (SOC), della bioinformatica e dello scheduling usando la programmazione con vincoli (CP).*

*Nello specifico, per la modellazione siamo interessati a rappresentare contratti e Service Level Agreements (SLA) in termini di linguaggi basati su vincoli. Le nozioni di contratto e SLA sono fondamentali in SOC dato che permettono di descrivere interazioni reciproche tra parti comunicanti ed esprimere proprietà relative alla qualità dei servizi come costi, prestazioni, disponibilità, ecc.*

*I linguaggi esistenti per la descrizione di web-services (es. WSDL, WS-CDL e WS-BPEL) non sono adatti per la descrizione di contratti e SLA e attualmente non c'è consenso su uno dei vari formalismi proposti in questo senso. Intendiamo dunque sviluppare un linguaggio per la descrizione di contratti che sia sufficientemente espressivo e che permetta la decidibilità di proprietà interessanti. Questo potrebbe produrre vari vantaggi. Innanzitutto permetterebbe di vedere la compatibilità tra contratti in termini di soluzione di problemi di vincoli, rendendo così possibile l'utilizzo di efficienti risolutori di vincoli per verificare, ad esempio, che un servizio offerto corrisponda ai requisiti di una determinata richiesta. La natura dinamica delle interazioni può essere modellata includendo in questo linguaggio sia meccanismi di constraint solving che di sincronizzazione, seguendo il modelli del Concurrent Constraint Programming (CCP) e del Constraint Handling Rules (CHR). Utilizzando "soft constraints" sarebbe poi possibile ottenere la flessibilità necessaria a modellare caratteristiche di SLA. Infine la verifica di sistemi che implementano servizi potrebbe essere semplificata usando le proprietà di decidibilità del linguaggio e opportuni strumenti di trasformazione e verifica per linguaggi con vincoli, come illustrato in seguito.*

*Riguardo alla soluzione di vincoli siamo interessati a problemi provenienti dai settori della bioinformatica e dello scheduling dove, senza una scelta attenta della rappresentazione del problema e delle tecniche di soluzione di vincoli usate, ci si trova spesso di fronte a problemi intrattabili. Due approcci diffusi in questo campo sono la progettazione di vincoli globali "dedicati" e algoritmi di ricerca ibridi. Siamo interessati all'applicazione di queste tecniche per risolvere problemi complessi quali la determinazione di strutture proteiche, l'inferenza di aplotipi, l'allineamento RNA, "nurse scheduling" e "car sequencing". Questi problemi presentano diversi vincoli complessi e ricorrenti che si possono esprimere come globali. Ad esempio, nel problema di determinare la struttura di una proteina viene richiesto che un punto nel reticolo non possa venire usato da due diversi amminoacidi e questo può essere facilmente modellato utilizzando il vincolo globale "all-different". Il nostro obiettivo primario è dunque identificare nuovi vincoli globali, studiarne complessità computazionale e progettare propagatori efficaci ed efficienti. Questi problemi sui vincoli potrebbero inoltre avere caratteristiche di ottimizzazione sfruttabili da algoritmi dedicati di ricerca locale. Sempre in relazione al problema di determinazione delle strutture proteiche, un costo energetico potrebbe venire associato a ogni coppia di amminoacidi non consecutivi quando entrano in contatto. In questo caso, la soluzione dovrebbe minimizzare il costo totale di energia. Inoltre, siamo interessati allo studio e alla sperimentazione di integrazioni tra algoritmi di ricerca locale e algoritmi di ricerca su alberi utilizzati da CP.*

*Per la verifica di sistemi infine intendiamo sviluppare metodologie e strumenti basati su trasformazione di programmi per la verifica e la sintesi automatiche di sistemi software. In particolare, considereremo metodologie di trasformazione che utilizzano regole e strategie per lo sviluppo di programmi la cui correttezza possa venire certificata tramite verifica formale. Per la scrittura di specifiche useremo linguaggi basati su vincoli (es. CLP, CCP e CHR) che hanno un alto potere espressivo e permettono una descrizione del sistema puramente dichiarativa. A partire da queste specifiche possiamo applicare regole di trasformazione che preservano la correttezza e, tramite strategie adeguate, derivare programmi corretti ed efficienti. Più in dettaglio, studieremo proprietà di correttezza di regole di trasformazione come "unfolding", "folding" e "replacement", considerando anche domini di vincoli specifici. Studieremo poi strategie di trasformazione adatte a verifica e sintesi di sistemi concorrenti e distribuiti, applicando la descrizione di contratti fornita in termini di linguaggi con vincoli. Infine, svilupperemo un tool che implementi le tecniche proposte ed elaboreremo casi di studio nel contesto della modellazione di contratti.*

### Testo inglese

*In this project in the unit of Bologna we aim at modeling, solving and verifying complex problems arising in the fields of Service Oriented Computing (SOC), bioinformatics, and scheduling, by using Constraint Programming (CP).*

*More specifically, as for modeling we are mainly interested in representing contracts and Service Level Agreements (SLA) in terms of constraint based languages. The notions of contract and SLA are very important in SOC since they allow to describe the mutual interactions between communicating parties and to express properties related to the quality of service such as cost, performance, availability etc. The existing languages for describing Web services (e.g. WSDL, WS-CDL and WS-BPEL) are not adequate for describing contracts and SLA and, so far, there exists no agreement on a specific proposal in this sense.*

*We intend to develop a constraint based language for contract modeling which is sufficiently expressive and, yet, allows the decidability of interesting properties. This could provide several benefits. Firstly, by considering contracts as constraints one can quite naturally see contract compatibility as an instance of constraint satisfaction, thus allowing to use existing (efficient) solvers for checking, for example, whether an offered service fulfills the requirements of a client. The dynamic nature of the interactions could be modeled by using a suitable integration of concurrency and constraint solving, along the lines of the Concurrent Constraint Programming (CCP) and Constraint Handling Rules (CHR) paradigms. Secondly, by using soft constraints (which can also be smoothly included in concurrent languages) one could obtain the flexibility needed to model SLA features which are important in practical applications. Thirdly, the task of verification of software systems implementing services could be simplified by using the decidability properties of the language together with the transformation and verification tools for constraint languages that we discuss below.*

*As for solving we are interested in problems arising from bioinformatics and scheduling. In these cases, without a careful choice of the problem representation and of the underlying constraint solving techniques, the practical problems that one has to face are often intractable. Two common approaches to address this efficiency issue are the design of dedicated global constraints and hybrid search algorithms.*

*We are interested in applying these techniques for solving challenging problems such as protein structure determination, haplotype inference, RNA alignment, nurse scheduling and car sequencing. Such problems exhibit many recurring and complex constraints which can be turned into global constraints. For example, in the protein structure determination problem, we require that a point in the lattice cannot be used to accommodate two distinct amino acids and this can be modeled by using the (global) all-different constraint. Our primary objective is to identify new global constraints, study their computational complexity and design effective and efficient propagators.*

*Our constraint problems might as well have an optimization characteristic which can be best exploited by dedicated local search algorithms. Taking again the previous example, an energy cost might be associated to each of non-consecutive amino acids when they are in contact. In this case, the solution should as well minimize the total energy cost. We are also interested in studying and experimenting with the suitable integrations of local search algorithms and the tree search algorithms employed by CP.*

*Concerning verification we intend to develop methodologies and tools based on program transformation for the automated verification and synthesis of software systems. In particular, we will consider the transformation methodology that uses rules and strategies for developing programs whose correctness with respect to their specifications is certified by a formal derivation. For writing specifications we will use constraint based languages (e.g. Constraint Logic Programming, CCP and CHR) which have a very high expressive power and permit a purely declarative description of systems. Starting from these specifications we can apply transformation rules that preserve correctness and, guided by suitable strategies, allow us to derive correct and efficient programs.*

More in detail, we will first study the correctness properties of such transformation rules as unfolding, folding, and replacement, by taking into account also specific constraint domains. Then we will study suitable transformation strategies for the verification and synthesis of concurrent and distributed systems, as described above, and we will apply them to the description of contracts given in terms of constraint languages. Finally we will develop a tool that implements the proposed techniques and we will work out case studies in the context of contract modeling.

## 9 - Settori di ricerca ERC (European Research Council)

*PE Mathematics, physical sciences, information and communication, engineering, universe and earth sciences*

*PE5 Information and communication: informatics and information systems, computer science, scientific computing, communication technology, intelligent systems*

*PE5\_8 Intelligent systems*

*PE5\_7 Theoretical computer science*

## 10 - Parole chiave

**Testo italiano**

VERIFICA BASATA SU VINCOLI  
VINCOLI GLOBALI

**Testo inglese**

CONSTRAINT BASED VERIFICATION  
GLOBAL CONSTRAINTS

## 11 - Stato dell'arte

**Testo italiano**

*Nel seguito presenteremo prima alcune nozioni relative alla tecnologia Service Oriented Computing (SOC) rilevanti ai fini di questo progetto. Quindi analizzeremo lo stato dell'arte relativo a specifiche tecniche basate su vincoli e trasformazioni che intendiamo utilizzare.*

*Attualmente l'approccio component-based è una metodologia importante per la progettazione e lo sviluppo di un'ampia gamma di sistemi software. Basandosi su concetti preesistenti quali tipi di dato astratti e oggetti, la nozione di componente può essere così definita come [SGM02]: "Un componente software è un'unità di composizione con una specifica interfaccia contrattuale e con esplicite dipendenze dal contesto. Un'interfaccia è un insieme di operazioni, ciascuna con il proprio nome, che possono venire invocate da un client. Le dipendenze dal contesto sono specifiche di ciò che l'ambiente dovrebbe fornire al fine di far funzionare le componenti".*

*La specifica della mutua interazione tra componenti può essere definita in termini di "contratti", ossia di vincoli comportamentali reciproci tra componenti comunicanti per il raggiungimento di qualche obiettivo.*

*La nozione di contratto è particolarmente rilevante nel contesto delle Service Oriented Architectures (SOA) e del SOC, due aree principalmente usate nel contesto del Web anche se, in accordo alla definizione W3C, più generalmente esse si riferiscono a modelli architetturali e computazionali per "insiemi di componenti che possono essere invocati e la cui interfaccia può essere pubblicata e scoperta".*

*Esistono già varie proposte di linguaggi per la specifica e la composizione, di servizi (web) a diversi livelli di astrazione, dai linguaggi di descrizione come WSDL (che vede i servizi in termini di una collezione di porte) a linguaggi di orchestrazione (XLANG, WSFL e WS-BPEL) e di coreografia (WS-CDL e BPEL4Chor) tramite i quali è possibile definire la composizione di servizi o in termini di un meta-servizio centralizzate (l'orchestratore) o considerando le interazioni reciproche tra di essi (la coreografia) senza centralizzazioni. Esistono anche alcune proposte specifiche [FHR+04, CCL+06, BZ07, CGP08] per descrivere sia i contratti che le loro proprietà fondamentali, tuttavia manca ancora una teoria generale di riferimento. Inoltre, la maggior parte delle proposte non tiene in considerazione i cosiddetti Service Level Agreements (SLA), ossia aspetti relativi alla qualità servizi quali costo, le prestazioni, la disponibilità, ecc.*

*Intendiamo studiare sia i contratti che altre problematiche relative alla bio-informatica e allo scheduling dal punto di vista della programmazione con vincoli (CP). Questa è una tecnologia di successo basata su precise basi teoriche e comprendente aspetti molto diversi, che vanno da quelli linguistici fino ad aspetti di combinatoria e di ottimizzazione. Dal punto di vista linguistico particolarmente interessante per questo progetto è la programmazione concorrente con vincoli (ccp) [SRP86], un paradigma che combina un modello computazionale basato sui vincoli con meccanismi di sincronizzazione e comunicazione tipici dei linguaggi concorrenti. In particolare sono rilevanti per noi alcune recenti estensioni del ccp che includono funzionalità di passaggio di nomi [BM07] e soft constraints [BF08] e che possono venire utilizzate per esprimere proprietà di SLA. Considereremo queste estensioni come punti di partenza anche se sembrano troppo espressive e complicate per i nostri obiettivi. Considereremo inoltre anche Constraint Handling Rules (CHR) [Fr98], un altro linguaggio concorrente basato su vincoli, come punto iniziale alternativo per il nostro linguaggio dedicato ai contratti.*

*Da un punto di vista più pratico particolarmente interessanti nel contesto della CP, e per questo progetto, sono i problemi di soddisfazione di vincoli (CSP) e quelli di ottimizzazione che scaturiscono da esigenze reali in moltissimi ambiti applicativi (ad esempio, un tipico esempio di CSP è il problema di gestire la turnazione degli impiegati di un'azienda). Anche se in generale risolvere un CSP è un problema intrattabile [Ma77] la CP si è dimostrata vincente nel risolvere in pratica molti CSP [Wa96] ed infatti esistono numerosi risolutori efficienti.*

*Per risolvere un CSP utilizzando le tecniche di programmazione con vincoli il primo problema che bisogna affrontare è quello di modellare il problema definendo una serie di variabili decisionali, il loro dominio e i vincoli fra di esse che devono essere soddisfatti. L'efficienza nel risolvere i problemi dipende fortemente da queste scelte di modellazione e formulare un modello efficace richiede una notevole esperienza.*

*Inoltre i risolutori di vincoli cercano le soluzioni esplorando lo spazio di ricerca ed utilizzando metodi deduttivi per ridurre la dimensione e lo sforzo computazionale della ricerca (mediante opportune propagazioni). Dunque buoni algoritmi di ricerca e deduzione sono vitali per risolvere i problemi basati su vincoli [RB+06].*

*I vincoli globali sono un altro aspetto centrale nella modellazione e nella risoluzione di problemi basati su vincoli. Questi vincoli catturano dei pattern utilizzati comunemente in vari problemi e aiutano a specificare vincoli complessi o utilizzati frequentemente in maniera molto semplice. Ad ogni vincolo globale viene associato un algoritmo di propagazione che viene utilizzato per ridurre lo spazio di ricerca in maniera efficiente sfruttando alcune proprietà tipiche del vincolo [HP+06, Re94, Ma77, FH+06, BH+08].*

*Importanti per i risolutori di vincoli sono anche gli algoritmi di ricerca su alberi basata su backtracking e la ricerca locale. Queste due tecniche sono in qualche modo complementari e difatti la loro integrazione è stata proposta per superarne i rispettivi limiti e combinarne i vantaggi al fine di migliorare le prestazioni [Mi03].*

*Infine rilevante per questo progetto è anche la trasformazione dei programmi, una metodologia di sviluppo di programmi che utilizza regole di trasformazione, che preservano la correttezza, e strategie di trasformazione, che permettono di applicare le regole in modo automatico in modo tale da derivare programmi più efficienti. Inizialmente proposta nell'ambito della programmazione funzionale [BD77] la trasformazione dei programmi è stata poi estesa ad altri paradigmi quali la programmazione logica [TS84], la programmazione logica con vincoli [EG96] e la programmazione concorrente con vincoli [EGM01]. Inoltre negli ultimi due decenni sono state sviluppate varie tecniche per la specializzazione, la sintesi e la verifica di proprietà dei programmi basate su trasformazioni (si veda [PP02] per*

una rassegna nel caso dei programmi logici).

Metodi basati sulla programmazione logica con vincoli e sulla trasformazione dei programmi sono stati anche proposti per la verifica di proprietà temporali di sistemi reattivi con un numero potenzialmente infinito di stati [DP99,FP01,FO97]. Tali metodi estendono il model checking tradizionale, che è limitato alla verifica di sistemi a stati finiti [CG99]. I metodi basati sulla programmazione logica con vincoli offrono alcuni vantaggi rispetto ad altri metodi per verificare sistemi a stati infiniti (si veda ad esempio [SU99,ZP04], dove il model checking viene combinato con tecniche deduttive e tecniche di astrazione), in quanto i vincoli permettono di rappresentare agevolmente in modo simbolico insiemi infiniti di stati.

Una metodologia correlata alla verifica è la sintesi automatica di sistemi reattivi da specifiche espresse in logica temporale. Nel caso di sistemi a stati finiti sono stati proposti metodi di sintesi basati su tecniche deduttive, su automi a stati finiti (che riconoscono stringhe o alberi infiniti), e sulla teoria dei giochi [CE82,MW84,KV01,PP06]. Assai limitata è la letteratura che tratta del problema della sintesi di sistemi a stati infiniti (si veda [AH01] per un approccio che utilizza tecniche simboliche).

#### **Testo inglese**

In the following we will first provide some more information on the Service Oriented Computing (SOC) issues which are relevant for this project. Then we will provide an up to date description of the state of the art for the specific constraint and transformation techniques that we intend to use.

Nowadays the component-based approach is a very important methodology which is used for the design and development of a variety of software systems. Building on previous concepts such as abstract data types and objects the notion of component can be defined as follows [SGM02]: "A software component is a unit of composition with contractually specified interfaces and explicit context dependencies. An interface is a set of named operations that can be invoked by clients. Context dependencies are specifications of what the deployment environment needs to provide, such that the components can function."

The specification of the mutual interaction between components can be described in terms of "contracts", namely specifications of mutual behavioral constraints among communicating components to achieve some goals. This is particularly relevant in the context of Service Oriented Architectures (SOA) and SOC. These two notions are commonly used in the context of the Web even though, more generally, according to a W3C definition they refers to architectures and computation models for "set of components which can be invoked, and whose interface descriptions can be published and discovered".

There exist already several proposals for languages which allow to specify (Web) services and their composition, at different levels of abstractions, ranging from description languages such as WSDL (which allows to describe services essentially as collections of ports), to orchestration languages (XLANG, WSFL and WS-BPEL) and choreography languages (WS-CDL and BPEL4Chor) which allow to define composition of services either in terms of a centralized meta-service (the orchestrator) or by considering the reciprocal interactions (the choreography) among the different services (without centralization).

There exist also some specific proposals [FHR+04,CCL+06,BZ07,CGP08] for describing contracts and their relevant properties. However a general, established theory of contracts is still missing. Furthermore most languages do not take into account the, so called, Service Level Agreements (SLA), that is aspects of contracts such as cost, performance, availability etc. which are related to the quality of service.

We plan to investigate this issue of contracts, as well as other problems arising in bio-informatics and scheduling, from the Constraint programming (CP) perspective. CP is a successful technology which is based on strong theoretical foundations and involve many different aspects, ranging from programming languages issues to combinatorial and optimization techniques.

From the linguistic point of view, particularly significant for this project is concurrent constraint programming (ccp) [SRP86], a paradigm which combines a computational model based on constraint programming with synchronization and communication mechanisms typical of concurrent languages. In particular, interesting for us are some recent extensions of ccp which include name-passing features [BM07] or soft constraints [BF08] and which allow to express SLA properties. We will consider these extensions as starting points, even though they seem too expressive (and too complicate) for our purposes. We will also consider Constraint Handling Rules (CHR) [Fr98], another concurrent constraint based language which could provide an alternative starting point.

From a more practical point of view particularly important in the field of CP, and for this project, are Constraint Satisfaction Problems (CSP) and optimization problems, which are ubiquitous in many real-life application areas. An example of a CSP is personnel rostering in a company: The challenge is to allocate personnel for the required tasks subject to company practice, personal expertise and preferences, and labour union regulations.

Even though solving CSP is computationally intractable in general [Ma77], CP has proven quite successful to solve CSP in practice [Wa96] and many very efficient solvers exist. To solve a CSP by using CP the problem is first modeled by defining a set of decision variables, their domains and the constraints on the variables that the solutions must satisfy. The efficiency of a constraint program highly depends on these modeling decisions. Hence, formulating an effective model requires considerable skills in modeling.

Moreover, CP solves a CSP by searching the space of possible assignments, possibly using inference methods to reduce the domains and the search effort (through suitable propagations). Thus, powerful search and inference algorithms are vital for effective constraint solving (see [RB+06] for a survey).

Particularly important for modeling and solving in CP are global constraints. They capture common patterns and help to specify complex and recurring constraints in a simple way. Each global constraint comes with a propagation algorithm. They bring about powerful inference by exploiting the structure of the constraint, thus help prune the search space efficiently and effectively [HP+06, Re94, Ma77, FH+06, BH+08].

Important for CP solvers are also backtracking tree search and local search (LS) which have somehow complementary features. The first one can be very effective at (dis)proving the feasibility of tightly constrained problems with complex constraints, but lacks good optimization characteristics. On the contrary, LS can be superior at optimization problems that are loosely constrained, but has weaknesses such as not being able to prove problem unfeasibility and having difficulties with complex constraints. Therefore, their integration has been studied to overcome their limitations and exploit the advantages of both, thus increasing search performance [Mi03].

Relevant to this proposal is also program transformation, a methodology for program development which uses transformation rules that preserve program correctness and transformation strategies that guide the application of the rules in an automatic way, so to derive efficient programs. This methodology has been first proposed in the area of functional programming [BD77] and then applied to other paradigms such as logic programming [TS84], constraint logic programming [EG96] and concurrent constraint programming [EGM01]. Moreover in the last two decades many transformation-based techniques have been proposed, such as program specialization, program synthesis, and program verification (see, for instance, [PP02] for a review in the case of logic programs).

Methods based on constraint logic programming and program transformation have also been proposed for the verification of temporal properties of reactive systems with a potentially infinite number of states [DP99,FP01,FO97]. These methods extend traditional model checking, which is restricted to the verification of finite state systems [CG99]. The verification methods based on constraint logic programming have some advantages with respect to other verification methods for infinite state systems (see [SU99,ZP04] where model checking is combined with deduction and abstraction techniques). Indeed, constraints allow us to easily represent in a symbolic way infinite sets of states.

A methodology related to verification is the automatic synthesis of reactive systems from temporal logic specifications. Several synthesis methods have been proposed in the case of finite state systems. These methods are based on deductive techniques, finite state automata (recognizing infinite strings or infinite trees), and game theory [CE82,MW84,KV01,PP06]. Very few works deal with the synthesis of reactive systems in the case of an infinite number of states (see [AH01] for an approach based on symbolic techniques).

---

## **12 - Riferimenti bibliografici**

[AH01] L. de Alfaro, T. Henzinger, R. Majumdar: *Symbolic Algorithms for Infinite-state Games*, Proc. CONCUR '01, LNCS 2154, Springer, pp. 536-550, 2001.

[Ap03] K.R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.

[BD77] R.M. Burstall, J. Darlington: *A Transformation System for Developing Recursive Programs*, J. ACM, 24 (1), pp. 44-67, 1977.

[BF08] S. Bistarelli, F. Santini. *A Nonmonotonic Soft Concurrent Constraint Language for SLA Negotiation*. Views on Designing Complex Architectures (VODCA'08), ENTCS, 2008.

[BH+08] C. Bessiere, E. Hebrard, B. Hnich, Z. Kiziltan, T. Walsh. *SLIDE: A Useful Special Case of the CARDPATH Constraint*. In Proceedings of the 18th European Conference on Artificial Intelligence, pp. 475--479. IOS Press, 2008.

- [BGMS08] Stefano Bistarelli, Maurizio Gabbriellini, Maria Chiara Meo, and Francesco Santini. Timed soft concurrent constraint programs. In *Proc. 10th Int. COORDINATION*, volume 5052 of LNCS, pages 50-66. Springer, 2008.
- [BZ07] M.Bravetti, G. Zavattaro. Towards a Unifying Theory for Choreography Conformance and Contract Compliance. In *Proc. SC 07. LNCS*, 2007.
- [BM07] Maria Grazia Buscemi, Ugo Montanari. CC-Pi: A Constraint-Based Language for Specifying Service Level Agreements. In *proc. ESOP 2007*, 18-32, LNCS.
- [BGMP97] F.S. de Boer, M. Gabbriellini, E. Marchiori and C. Palamidessi. Proving Concurrent Constraint Programs Correct. *TOPLAS*,19(5):685-725, ACM Press,1997.
- [dBGM04] Frank S. de Boer, Maurizio Gabbriellini, and Maria Chiara Meo. Proving correctness of timed concurrent constraint programs. *TOCL*, 5(4):706-731, ACM Press 2004.
- [CCL+06] Samuele Carpineti, Giuseppe Castagna, Cosimo Laneve, and Luca Padovani. A Formal Account of Contracts for Web Services. In *WS-FM'06*, volume 4184 of LNCS, pages 148-162, 2006.
- [CGP08] Giuseppe Castagna, Nils Gesbert, Luca Padovani. A theory of contracts for web services. IN *Proc. POPL 2008*: 261-272. ACM Press, 2008.
- [CE82] E.M. Clarke, E.A. Emerson: Design and synthesis of synchronization skeletons using branching time temporal logic, *Logics of Programs*, LNCS 131, Springer, pp. 52-71, 1982.
- [CG99] E.M. Clarke, O. Grumberg, D.A. Peled: *Model Checking*, The MIT Press, 1999.
- [DG05] Giorgio Delzanno and Maurizio Gabbriellini. Compositional verification of asynchronous processes via constraint solving. In *Proc. 32nd Int. ICALP*, volume 3580 of LNCS, pages 1239-1250. Springer, 2005.
- [DP99] G. Delzanno, A. Podelski: *Model Checking in CLP*, *Proc. TACAS'99*, LNCS 1579, Springer, pp. 223-239, 1999.
- [DGM09] C. Di Giusto, M. Gabbriellini and M.C. Meo. Expressiveness of Multiple Heads in CHR. In *SOFSEM '09*, volume 5404 of LNCS, pages 205--216. Springer, 2009
- [DDG+09] G. Delzanno, C. Di Giusto, M. Gabbriellini, C. Laneve and G. Zavattaro. The K-Lattice: Decidability Boundaries for Qualitative Analysis in Biological Languages. Submitted, 2009.
- [EG96] S. Etalle, M. Gabbriellini: *Transformations of CLP Modules*, *Theo.Comp.Sci.*, 166, pp. 101-146, 1996.
- [EGM01] A. Etalle, M. Gabbriellini and M.C. Meo. Transformation of CCP programs. *Transactions on Programming Languages and Systems (TOPLAS)*. 2001.
- [HP+06] W. J. van Hove, G. Pesant, L.-M. Rousseau, A. Sabharwal. Revisiting the Sequence Constraint. In *Proceedings of the 12th CP*, pp. 620--634. LNCS 4204, Springer, 2006.
- [FF+02] P. Flener, A.M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, T. Walsh. Matrix modelling: Exploiting common patterns in constraint programming. In *Proceedings of the International CP-02 Workshop on Reformulating Constraint Satisfaction Problems*, pp. 27--41, 2002.
- [FH+09] A. M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, T. Walsh. Filtering algorithms for the multiset ordering constraint. *Artificial Intelligence*, 173(2): 299--328, 2009.
- [FH06] A. M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, T. Walsh. Propagation algorithms for lexicographic ordering constraints. *Artificial Intelligence*, 170(10): 803-834, 2006.
- [FP01] F. Fioravanti, A. Pettorossi, M. Proietti: *Verifying CTL Properties of Infinite State Systems by Specializing Constraint Logic Programs*, *Proc. VCL '01*, Florence, Italy, pp. 85-96, 2001.
- [FO97] L. Fribourg, H. Olsen: A decompositional approach for computing least fixed-points of Datalog programs with Z-counters, *Constraints*, 2, pp. 305-335, 1997.
- [Fr98] Thom W. Fruehwirth. Theory and practice of constraint handling rules. *J. Log. Program.*, 37(1-3):95-138, 1998.
- [FHR+04] Cedric Fournet, C. A. R. Hoare, Sriram K. Rajamani, and Jakob Rehof. Stuck-Free Conformance. In *CAV'04*, volume 3114 of LNCS, pages 242-254, 2004.
- [KV01] O. Kupferman, M.Y. Vardi, *Synthesizing Distributed Systems*, *Proc. LICS '01*, IEEE Press, pp. 389-398, 2001
- [Ma77] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99--118, 1977.
- [MW84] Z. Manna, P. Wolper: *Synthesis of Communicating Processes from Temporal Logic Specifications*, *ACM TOPLAS*, 6 (1), pp. 68-93, 1984.
- [Mi03] M. Milano. *Constraint and Integer Programming: Toward a Unified Methodology*. Kluwer Academic Publishers, 2003.
- [PD+05] A. Dal Palù, A. Dovier, E. Pontelli. A New Constraint Solver for 3D Lattices and Its Application to the Protein Folding Problem. In *Proc.s of the 12th Int. Conf.on Logic for Programming, Artificial Intelligence, and Reasoning*, pp. 48--63. LNCS 3835, Springer, 2005.
- [PP02] A. Pettorossi, M. Proietti: Program Derivation = Rules + Strategies, In: *Computational Logic: Logic Programming and Beyond*, LNAI 2407, Springer, pp. 273-309, 2002.
- [PP06] N. Piterman, A. Pnueli, Y. Sa'ar: *Synthesis of Reactive(1) Designs*, *Proc. VMCAI 2006*, LNCS 3855, pp. 364-380, 2006.
- [RB+06] F. Rossi, P. van Beek, T. Walsh (Eds.). *Handbook of Constraint Programming*. Elsevier, 2006.
- [Re94] J.C. Régin, A filtering algorithm for constraints of difference in CSPs. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 362--367. AAAI Press, 1994.
- [Sa08] F. Santini. Managing Quality of Service with Soft Constraints. *ICLP 2008*, volume 5366/2008, LNCS, pages 815-817, 2008.
- [SU99] H.B. Sipma, T.E. Uribe, Z. Manna: Deductive model checking, *Formal Methods in System Design*, 15, pp. 49-74, 1999.
- [SRP96] V. Saraswat, M. Rinard, and P. Panangaden. Semantic foundations of concurrent constraint programming. IN *proc. POPL '91*. ACM Press, 1991.
- [SGM02] C. Szyperski, D. Gruntz, and S. Murer. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, 2002.

[TS84] H. Tamaki, T. Sato: *Unfold/Fold Transformation of Logic Programs*, Proc. ICLP '84, Uppsala, Sweden, pp. 127-138, 1984.

[Wa96] M.G. Wallace. *Practical applications of constraint programming*. Constraints, 1(1-2): 139--168, 1996.

[ZL07] Z. Kiziltan, A. Lodi, M. Milano, F. Parisini. *CP-Based Local Branching*. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming*, pp. 847--855. LNCS 4741, Springer, 2007.

[ZP04] L.D. Zuck, A. Pnueli: *Model checking and abstraction to the aid of parameterized systems: A survey*, Computer Languages, Systems & Structures, 30, 3-4, pp. 139-169, 2004.

## 13 - Descrizione del programma e dei compiti dell'Unità di Ricerca

### Testo italiano

Quando consideriamo applicazioni software complesse come quelle utilizzate nel campo del Service Oriented Computing (SOC) e della bioinformatica, l'utilizzo della programmazione con vincoli (CP) può fornire tre vantaggi.

Per prima cosa la CP permette una formulazione dichiarativa del problema ad alto livello semplificando così il compito della modellazione e della specifica del problema. In seconda istanza la tecnica della CP utilizza risolutori efficienti che sfruttano diverse tecniche per affrontare problemi teoricamente intrattabili. Infine i linguaggi basati su vincoli, vista la loro natura dichiarativa, permettono di definire dei tools per la trasformazione, la verifica e la sintesi di sistemi software molto semplici se paragonati a quelli usati per paradigmi.

Lo scopo di questo progetto è quello di sviluppare dei modelli e dei tool basati sulla programmazione con vincoli, per modellare, risolvere e ragionare sulla correttezza di problemi complessi nei campi del Service Oriented Computing, della bioinformatica e dello scheduling. Nei paragrafi seguenti tratteremo più in dettaglio questi tre aspetti che per comodità sono organizzati in tre work-packages (WP).

(WP1 Modellizzazione) Lavorando in stretta cooperazione con le unità di Perugia e Padova in questo WP siamo particolarmente interessati allo sviluppo di un linguaggio basato su vincoli per rappresentare contratti e aspetti di Service Level Agreements (SLA) nel campo del SOC (maggiori dettagli su questi ambiti possono essere trovate nella sezione "stato dell'arte").

Recentemente nel contesto del SOC c'è stato molto interesse nelle nozioni di contratto e di SLA. Infatti da un punto di vista pratico è molto importante poter garantire che un insieme di servizi cooperanti, possibilmente specificati mediante WSDL, WS-CDL o WS-BPEL, si comporti correttamente. "Correttamente" qui significa molte cose diverse: la specifica dei servizi deve essere consistente, il loro comportamento deve soddisfare le specifiche, i servizi offerti devono poter soddisfare le richieste dei clienti. Inoltre spesso devono poter essere verificati aspetti quantitativi quali costi, performance ecc.

Uno dei problemi principali da affrontare qui è quello dell'espressività del linguaggio che si usa: se si descrivono i servizi usando linguaggi "poveri" (come ad esempio WSDL) ci si dimenticano completamente gli aspetti comportamentali e quindi non si possono verificare proprietà rilevanti dal punto di vista pratico quali, ad esempio, "Questo servizio soddisfa quella richiesta"? D'altro canto se si usano linguaggi molto espressivi (spesso Turing completi) quali WS-CDL, WS-BPEL e quelli proposti in [BM07,BF08] le proprietà interessanti diventano indecidibili. Quindi sarebbe importante definire un linguaggio adatto per modellare contratti che avesse sufficiente potere espressivo ed anche interessanti proprietà computazionali. Tuttavia una teoria generale dei contratti è ancora mancante e molte domande interessanti circa la natura dei contratti, come essi possono essere composti e come le loro proprietà possono essere verificate, attendono ancora delle risposte soddisfacenti. Noi crediamo che alcune di queste risposte possano essere fornite dalla CP per i seguenti motivi.

Per prima cosa i contratti, o più generalmente le interazioni tra vari componenti, possono essere rappresentate in modo naturale sotto forma di vincoli. Tale rappresentazione ci permette, per esempio, di rappresentare la compatibilità rispetto ad un contratto come un vincolo che deve essere soddisfatto e quindi di usare i più moderni ed efficienti risolutori per testare se un servizio offerto soddisfa i requisiti imposti da un contratto.

La natura dinamica delle interazioni può essere modellata utilizzando un calcolo appropriato che includa calcolo basato su vincoli e di meccanismi di sincronizzazione, seguendo l'approccio di Concurrent Constraint Programming (CCP) e Constraint Handling Rules (CHR). Questo è stato fatto anche in [BM07], tuttavia, a differenza di [BM07], noi ci proponiamo di ottenere un modello molto più semplice e senza meccanismi di passaggio di nomi analoghi a quello del pi-calcolo.

Un secondo punto riguarda l'utilizzo dei "soft constraints" e cioè vincoli a cui è associato un peso e che possono essere utilizzati per esprimere preferenze ma anche probabilità, incertezza e altri aspetti quantitativi (si veda il modello B delle unità di Perugia e Padova per maggiori informazioni sui soft constraints). L'utilizzo dei soft constraint potrebbe garantire la flessibilità richiesta per esprimere proprietà legate alla qualità dei servizi come i costi, le performance, la disponibilità ecc., aspetti che sono di importanza vitale in molte applicazioni pratiche nel SOC [Sa08]. Un esempio tipico è la modellazione dei trade-off: ad esempio si può essere disposti ad utilizzare un servizio meno affidabile nel caso in cui sia meno costoso. I soft constraints possono essere facilmente inclusi nei linguaggi concorrenti, si veda per esempio [BF08,BGMS08] e usati per modellare SLA, come mostrato in [BF08], tuttavia noi intendiamo sviluppare un linguaggio più semplice di quello proposto in [BF08].

Infine, utilizzando linguaggi concorrenti basati sui vincoli si possono studiare le tecniche di verifica dei contratti in maniera analoga a [dBGM04, DG05] ma anche tools basati sulla trasformazione per la verifica e la sintesi (si veda il WP 3 sotto). Il punto importante qui è che linguaggi quali CCP e CHR ammettono delle tecniche di verifica molto più semplici di quelle necessarie per altri linguaggi concorrenti (si veda per esempio [BGMP97]). In particolare, intendiamo identificare un linguaggio basato su vincoli per la modellazione dei contratti e dei SLA che sia abbastanza espressivo da permettere la modellazione di esempi reali e allo stesso tempo renda decidibili alcune proprietà interessanti. Infatti, contrariamente a quanto potrebbe sembrare a prima vista, esistono linguaggi Turing equivalenti dove proprietà quali la raggiungibilità di uno stato sono decidibili. In [DDG+09] questo risultato è mostrato per linguaggi ispirati alla biologia. Nel nostro caso, l'identificazione di un linguaggio con proprietà simili potrebbe portare alla creazione di tool per la verifica nell'ambito del SOC molto importanti anche per applicazioni pratiche.

(WP2 Soluzione di vincoli) Per quanto riguarda l'aspetto di soluzione di vincoli il nostro obiettivo, in stretta collaborazione con le unità di Udine e Roma, è lo sfruttamento della potenza della CP per affrontare problemi di soddisfacibilità di vincoli forniti dall'industria e dalla ricerca biologica. In particolare siamo interessati alla risoluzione di problemi quali: determinazione della struttura delle proteine; deduzione dell'aplotipo; allineamento dell'RNA; scheduling dei turni delle infermiere e sequenzializzazioni di operazioni produttive. Lo risoluzione di questi importanti problemi è considerata nelle rispettive aree di pertinenza una vera e propria sfida computazionale.

Vorremmo migliorare l'efficienza della risoluzione dei problemi con vincoli proponendo modelli e tecniche di ricerca più performanti. Nelle righe seguenti presenteremo brevemente i nostri obiettivi illustrando cinque temi che possono contribuire allo sviluppo di questa linea di ricerca.

(i) Un problema di soddisfacibilità di vincoli (CSP) può essere modellato in vari modi. Per esempio, nel problema della determinazione della struttura delle proteine si deve decidere la disposizione degli aminoacidi in un reticolo tridimensionale. Le variabili possono rappresentare gli aminoacidi e i domini i nodi del reticolo. Alternativamente possiamo però utilizzare un modello duale nel quale le variabili sono i nodi del reticolo mentre i valori delle variabili rappresentano gli aminoacidi. Spesso modelli alternativi hanno diversi punti di forza e quindi vari modelli possono essere integrati in modo da ottenere un nuovo modello più efficace. Vorremmo condurre una analisi sistematica dei vari modelli che si prestano alla risoluzione dei CSP utilizzando i principali algoritmi di ricerca. In questa analisi ci aspettiamo di trovare pattern di modellazione comuni, come fatto per esempio in [FF+02].

(ii) Nei problemi di scheduling ci si trova spesso di fronte a sequenze di variabili decisionali e a vincoli su sotto sequenze che devono essere applicate all'intera sequenza. Per esempio, dovendo decidere le macchine da produrre in una linea di montaggio, possiamo avere un vincolo del tipo: lungo la catena produttiva solo una macchina su tre può avere installato il tettuccio apribile. Tale tipo di vincoli può essere modellato utilizzando i cosiddetti vincoli globali [HP+06,BH+08]. In



generale, molti pattern ricorrenti dei problemi di sequenziamento di task devono ancora essere scoperti.

I vincoli globali possono aiutare anche a specificare vincoli complessi in modo conciso. Per esempio nel problema della determinazione della struttura di una proteina ad un certo punto, il vincolo (globale) all-different permette di specificare che due aminoacidi non possono essere posti sullo stesso nodo del reticolo [Re94]. Il nostro obiettivo primario è quindi quello di identificare i pattern più spesso ricorrenti e i constraint più complessi che possono essere espressi mediante vincoli globali.

(iii) Dopo avere identificato i vincoli globali più promettenti vorremmo studiarne la complessità computazionale e combinatoriale. Per esempio, un vincolo globale utile nel determinare la struttura delle proteine è il vincolo noto come saw che combina gli effetti di due altri vincoli (all different e contiguous). Intendiamo provare che la propagazione di questo vincolo è un problema NP completo riducendolo al problema del cammino Hamiltoniano che è uno tra i più noti problemi NP completi.

(iv) I vincoli globali migliorano la ricerca della soluzioni se possono essere effettivamente propagati in modo efficiente. Vorremmo quindi definire dei meccanismi di propagazione sfruttando le strutture e le proprietà di questi vincoli globali e cercare dei software in grado di gestire queste tecniche in modo efficiente. Per esempio i problemi di sequenziamento di task esibiscono caratteristiche che possono essere sfruttate utilizzando la programmazione dinamica. Ci aspettiamo che un gran numero di vincoli globali per problemi bioinformatici possano avere meccanismi di propagazione NP-completi. Per questi vincoli vorremmo realizzare degli algoritmi polinomiali approssimati.

(v) I nostri problemi possono includere aspetti di ottimizzazione e per questo si possono sfruttare appositi algoritmi di ricerca locale. Ad esempio, nella determinazione della struttura della proteina, possiamo immaginare che ci sia un costo energetico associato a ogni coppia non consecutiva di aminoacidi quando sono in contatto. In questo caso la soluzione cercata potrebbe essere quella di minimizzare il costo totale dell'energia. Un nostro obiettivo è dunque quello di studiare e sperimentare integrazioni di algoritmi di ricerca locale e di ricerca su alberi per meglio affrontare istanze difficili e di grandi dimensioni.

Come detto prima in questo WP collaboreremo estensivamente con le unità di Udine e Roma. In particolare integreremo i nostri risultati riguardo i problemi bioinformatici nel loro risolutore di vincoli COLA [3]. Realizzeremo prototipi dei nostri global constraint e di approcci ibridi di ricerca conducendo uno studio sperimentale per valutare il loro contributo nella soluzione di CSP.

(WP3 Trasformazioni e verifica) In questo WP ci concentreremo su vari aspetti della trasformazione di linguaggi basati su vincoli, riguardanti la metodologia di base, la sua applicazione nel campo della verifica e della sintesi di sistemi concorrenti e distribuiti, con particolare enfasi sugli artefatti software specificati usando il linguaggio per la modellizzazione di contratti sviluppato nel WP1. Inoltre svilupperemo dei tool che implementino le tecniche proposte. L'attività di ricerca sarà articolata nei seguenti punti.

#### (i) - Regole e strategie di trasformazione

Si proporranno estensioni delle regole di trasformazione per programmi logici con vincoli (CLP) e per CCP quali l'unfolding, il folding e il rimpiazzamento, tenendo anche in considerazione le specifiche classi di vincoli utilizzati per descrivere sistemi concorrenti e distribuiti a stati infiniti (vedi Punto (ii)). Tra queste classi citiamo le disequazioni lineari, sia su interi che reali, e le disequazioni tra stringhe, alberi e altre strutture di dati.

Si elaboreranno strategie di applicazione delle regole di trasformazione con l'obiettivo di migliorare l'efficienza dei programmi. In particolare si studieranno algoritmi per automatizzare l'unfolding e il folding in presenza di vincoli. Tali algoritmi verranno utilizzati all'interno di strategie più complesse, tra le quali avranno particolare rilevanza: (a) la specializzazione di un programma rispetto ad un determinato contesto d'uso, espresso da vincoli sui dati di input, (b) l'eliminazione delle variabili esistenziali, che consente di evitare il calcolo di valori non necessari, e (c) la riduzione del nondeterminismo utilizzando vincoli sullo spazio di ricerca inferiti attraverso un'analisi statica del programma.

#### (ii) - Specifica, verifica e sintesi di sistemi a stati infiniti mediante trasformazione di programmi logici con vincoli

Si studieranno vari metodi per: (a) specificare sistemi a stati infiniti mediante linguaggi con vincoli (principalmente CLP e CCP) e (b) esprimere le proprietà temporali dei sistemi mediante opportuni predicati definiti in questi programmi. Per esempio, sistemi che utilizzano contatori interi e reti di Petri possono essere facilmente codificati da programmi logici con vincoli lineari a soluzioni intere. Si studieranno inoltre codifiche di sistemi parametrizzati che, per rappresentare reti di processi, utilizzano vincoli su domini complessi come, ad esempio, insiemi di stringhe, alberi e grafi.

Si utilizzerà l'approccio trasformazionale sviluppato al Punto (i) per effettuare la verifica di proprietà di sistemi concorrenti a stati infiniti. Il metodo di verifica trasformazionale prende in considerazione un programma che descrive le proprietà temporali del sistema e applica a tale programma strategie di trasformazione con l'obiettivo di ottenere un programma finale dove le proprietà d'interesse siano verificabili attraverso semplici test di soddisfacibilità di vincoli. Si studieranno strategie per verificare proprietà esprimibili per mezzo della logica temporale CTL.

Inoltre intendiamo esplorare l'uso di tecniche di trasformazione per realizzare la sintesi di sistemi concorrenti a stati infiniti. Le proprietà temporali del sistema da sintetizzare saranno specificate da un programma con vincoli e le transizioni che definiscono tale sistema verranno derivate per mezzo di opportune regole e strategie di trasformazione. Considereremo in particolare l'applicazione di queste tecniche al linguaggio per la modellizzazione di contratti sviluppato nel WP 1.

#### (iii) - Implementazione del sistema di trasformazione

Le tecniche di trasformazione, verifica e sintesi verranno implementate come moduli del sistema MAP per la trasformazione dei programmi logici con vincoli. Con tale sistema verranno interfacciati i vari solutori di vincoli necessari alla realizzazione delle tecniche descritte ai Punti (i) e (ii). Il sistema verrà utilizzato come strumento per la sperimentazione delle tecniche proposte. Per sperimentare la verifica e la sintesi si considereranno vari protocolli che sono modellabili da sistemi a stati infiniti e protocolli parametrizzati nel numero dei processi partecipanti al protocollo stesso con varie topologie di rete di connessione. Infine, si svilupperanno casi di studio nel contesto della modellizzazione di contratti.

#### Testo inglese

When considering complex software applications, like those arising in the fields of Service Oriented Computing and bioinformatics, the use of constraint programming (CP) can provide three advantages.

Firstly CP allows a very high level, declarative formulation of the problem thus simplifying the task of modeling, specifying and prototyping. Secondly, CP today can use very efficient solvers which combine many different techniques, thus allowing us to tackle problems which are theoretically intractable. Thirdly, given their declarative nature, constraint based languages allow us to build tools for the transformation, the verification and the synthesis of software systems which are considerably simpler than those needed for other paradigms.

In this project we then aim at developing models and tools based on CP for modeling, solving and reasoning on complex problems arising from the fields of Service Oriented Computing (SOC), bioinformatics, and scheduling. In the following we will provide some more details on these three aspects which, for the sake of convenience, are organized in three distinct work-packages (WP).

(WP1 Modeling) Working in close cooperation with the sites of Perugia and Padova in this WP we are mainly interested in developing a constraint based language for representing contracts and Service Level Agreements (SLA) in the field of SOC (see the part on background for a description of these notions).

Recently in the context of SOC there has been quite some interest in the notion of contract and of SLA. In fact, from a practical point of view, it is clearly very important to guarantee that a given set of cooperating services, possibly specified by using WSDL, WS-CDL or WS-BPEL, behaves correctly. Here "correctly" means a variety of things: the composition of services has to be consistent, their behavior have to satisfy a given specification, the offered service has to satisfy the request of clients. Moreover often one has to verify some quantitative requirements such as cost, performance etc. One of the main problems that one has to face here is the expressivity of the language: If one describes services by using "poor" languages (such as WSDL, for example) one forgets completely their behavioral aspects and therefore cannot verify practically relevant properties such as, for example, "Does this service fulfill that request?". On the other hand, if one uses very expressive (often Turing complete) languages such as WS-CDL, WS-BPEL and those proposed in [BM07,BF08], interesting behavioral properties become easily undecidable.

Even though there exists specific models and languages for contracts [FHR+04,CCL+06,BZ07,CGP08] a general, established theory of contracts is still missing and several interesting questions about their precise nature, how they can be composed and how they can be verified still wait for some answers.

We believe that some of these answer can be provided by constraint programming for the following reasons.

First, contracts and, more generally, interactions between components can be naturally represented in terms of constraints. Such a representation allows us, for example, to see contract compatibility as an instance of constraint satisfaction and therefore to use existing and efficient constraint solvers for checking whether an offered service fulfills the requirements of a requested contract. The dynamic nature of the interactions arising in the context of some negotiations between different components could be modeled by using a suitable calculus which includes both constraint solving and synchronization mechanisms, following the approach of Concurrent Constraint Programming (CCP) and of Constraint Handling Rules (CHR). This has been done also in [BM07], however, differently from [BM07], we do not plan to include name mechanisms analogous to those of pi-calculus and we aim to a much simpler calculus.

A second important point concerns the use of "soft constraints", that is constraints which have associated a weight and therefore allows us to express preferences, as well as probability, fuzziness and many other quantitative aspects (see the B model of Perugia and Padova in this project for more information on soft constraints). The use of soft constraints would provide the flexibility needed to express properties related to the quality of service, such as cost, performance, availability etc., which are important in practical applications of SOC [Sa08]. A typical example is the modeling of trade-offs: sometimes one is willing to use a less-trustworthy service if there is an economic incentive. Soft constraints can be smoothly included in concurrent languages [BF08, BGMS08] and used for modeling SLA, as shown in [BF08], however we aim at a language simpler than that one used in [BF08].

Third, in the context of constraint based languages one could study verification techniques for contracts along the lines of [dBGM04, DG05], as well as transformation based tools for the verification and the synthesis, as we discuss in WP3 below. The point here is that languages such CCP and CHR admit a proof theory and therefore verifications techniques much simpler than those needed for other concurrent languages, as demonstrated in [BGMP97]. In particular, we aim at defining a suitable language for contract modeling which is sufficiently expressive and, yet, has nice computational properties. In fact, in spite of what a naive thought could suggest, there exist indeed quite expressive languages (even Turing complete) where properties like reachability of a state are decidable (this is a typically undecidable property for standard Turing machines). This has already been shown for bio-inspired languages [DDG+09]. In our case, identifying a language with similar properties could lead to the definition of verification tools for SOC quite important for practical applications.

(WP2 Solving) Here, in cooperation with the units in Udine and Roma, we aim at exploiting the power of CP to tackle some real-life CSPs originating from biological sciences and industry. In particular, we are interested in solving problems like protein structure determination, haplotype inference, RNA alignment, nurse scheduling, and car sequencing. Each of these problems are shown to be important and highly challenging in their respective areas.

We plan to enhance the efficiency of constraint solving by devising powerful constraint models and search techniques. In the following, we briefly explain our aims by illustrating five specific tasks which could contribute to this line of research.

(i) A CSP can be modeled in different ways. For instance, in the protein structure determination problem, we need to decide the placement of amino acids in a 3-d lattice. The variables could represent the amino acids and the domains represent the lattice points. Alternatively, we can use a dual model in which the variables are the lattice points and the values are the amino acids. Often, alternate models have complementary strengths and therefore such models can be integrated so as to obtain a new model that overcomes the disadvantages of one model with the advantages of the other one, and vice versa. We there want to conduct a systematic analysis of various models that can best suit to solve our CSPs using the underlying search algorithms. During this analysis, we expect to find out common modeling patterns, as for instance done in matrix modeling [FF+02].

(ii) In scheduling problems, we often have a sequence of decision variables and a sliding sequence constraint which applies down the sequence. For instance, in the car sequencing problem, we need to decide the sequence of cars on the production line. We might have a constraint on how often a particular option is met along each sequence (e.g. only 1 out of 3 cars can have the sun-roof option). Another example is the nurse scheduling problem in which we decide the sequence of shifts worked by each nurse. We might have a constraint on how many consecutive night shifts any nurse can work. Such requirements can be modeled using the existing sequence and slide constraints [HP+06, BH+08]. Many such recurring patterns in our sequencing problems are yet to be discovered.

Global constraints can also help specify complex constraints in a concise way. For instance, in the protein structure determination problem, we require that a point in the lattice cannot be used to accommodate two distinct amino acids. Such a requirement can easily be modeled using the famous all-different constraint [Re94]. Our primary objective is therefore to identify new recurring patterns and complex constraints which can be turned into useful global constraints.

(iii) After having identified useful constraints, we aim to use techniques for the analysis of computational complexity to evaluate their combinatorial complexity. For instance, a useful global constraint which arises in the protein structure determination problem is the saw constraint. This constraint combines the all-different constraint together with the contiguous constraint which requires that a list of variables represent adjacent lattice points. We want to prove that propagating this constraint is NP-complete using a reduction from the hamiltonian cycle problem which is known to be NP-complete.

(iv) A global constraint enhances constraint solving only if it can be propagated effectively and efficiently. We therefore want to design propagators by characterizing the common structural features of these global constraints and then looking for algorithmic tools that could handle these features efficiently. For example, sequencing problems exhibit a very nice feature which can be best exploited by dynamic programming. We expect that propagating many useful global constraints for the bioinformatics problems will turn out to be NP-complete. For such constraints, we plan to design polynomial approximation algorithms.

(v) Our constraint problems might as well have an optimization characteristic which can be best exploited by dedicated local search algorithms. Taking again the protein structure determination problem as an example, an energy cost might be associated to each pair of non-consecutive amino acids when they are in contact. In this case, the solution should as well minimize the total energy cost. Our objective is then to study and experiment with the suitable integrations of local search algorithms and the tree search algorithms employed by CP so as to face large-scale difficult instances of our CSPs

As previously mentioned we will extensively collaborate with the Udine unit. In particular, we will export our findings regarding the bioinformatics problems into their ad-hoc constraint solver COLA (Constraint Solving on Lattices) [3]. We will develop prototype implementations of our global constraints and hybrid search algorithms as well as conduct an extensive experimental study to assess their contribution by using the 'best' models of our CSPs identified in the first place.

(WP3 Transformation and verification) In this WP we will concentrate on several aspects of the transformational methodology for constraint based languages. We will investigate on the foundations of the methodology and on its application for the verification and the synthesis of concurrent and distributed software, with particular emphasis on software artifacts specified by using the language for contract modeling developed in WP1. We will also develop a tool that implements the proposed techniques. The research activity will be developed according to the following points.

(i) - Transformation rules and strategies

We will propose extensions of the usual transformation rules for constraint logic programs (CLP) and for CCP such as unfolding, folding, and constraint replacement, also taking into account the specific classes of constraints needed for the description of infinite state systems (see Point (ii)). Among these classes we mention: linear inequations over integers or reals, and inequations between strings, trees and other algebraic data structures.

We will design strategies for guiding the application of the transformation rules with the objective of improving program efficiency. In particular, we will study algorithms for mechanizing the applications of the unfolding and folding rules in the presence of constraints from different domains. The unfolding and folding algorithms will be used within more complex strategies, such as: (a) the specialization of a program with respect to a given context of use, expressed by constraints on the input data, (b) the elimination of existential variables for avoiding the computation of unnecessary values, and (c) the reduction of nondeterminism by using constraints on the search space inferred through a static analysis of the program.

(ii) - Specification, verification, and synthesis of infinite state systems via transformation of constraint languages

We will study various methods for: (a) specifying infinite state systems by using constraint languages (mainly CLP and CCP) and (b) expressing the temporal properties of the systems by suitable predicates defined by these programs. For instance, systems that make use of integer counters and Petri nets can easily be encoded by logic programs with linear constraints over the set of integer numbers. We will also study various encodings of parameterized systems which, in order to describe the process networks, make use of constraints over more complex domains, such as sets of strings, trees, and graphs.

We will use the transformational approach developed at Point (i) for verifying infinite state concurrent systems. The transformational verification method takes into consideration a program that describes the temporal properties of the system and applies to this program predefined transformation strategies aimed at deriving a final program where the properties of interest can be verified through simple constraint satisfiability tests. We will study strategies for verifying properties which can be expressed by the temporal logic CTL.

We will also explore the applicability of transformation techniques to the problem of synthesizing infinite state concurrent systems. Similarly to the verification techniques, the properties of the concurrent systems to be synthesized, will be specified by a constraint program and the transitions defining such a system will be derived by means of suitable transformation rules and strategies. We will consider specifically the application of these techniques to the language for contract modeling developed in WP1.

(iii) - Implementation of a transformation prototype

The transformation, verification, and synthesis techniques will be implemented as modules of the MAP system, a prototype system for the transformation of constraint logic programs. Constraint solvers for various constraint domains will be linked to the MAP system for implementing the various techniques described at Point (i) and (ii). The prototype transformation system will be used as a tool for experimenting the proposed techniques. The verification and synthesis experiments will be performed on various protocols that are modeled as infinite state systems. We will also consider protocols which are parameterized in the number of processes taking part in the protocol and, for these protocols, we will take into account various network topologies. Finally, we will work out case studies in the context of contract modeling.

## 14 - Descrizione delle attrezzature già disponibili ed utilizzabili per la ricerca proposta

**Testo italiano**

Nessuna

**Testo inglese**

Nessuna

## 15 - Descrizione delle Grandi attrezzature da acquisire (GA)

**Testo italiano**

Nessuna

**Testo inglese**

Nessuna

## 16 - Mesi persona complessivi dedicati al Progetto

	Numero	Disponibilità temporale indicativa prevista		Totale mesi persona	
		1° anno	2° anno		
<i>Componenti della sede dell'Unità di Ricerca</i>	2	9	9	18	
<i>Componenti di altre Università/Enti vigilati</i>	2	10	10	20	
<i>Titolari di assegni di ricerca</i>	2	9	9	18	
<i>Titolari di borse</i>	<i>Dottorato</i>	3	11	11	22
	<i>Post-dottorato</i>	0			
	<i>Scuola di Specializzazione</i>	0			
<i>Personale a contratto</i>	<i>Assegnisti</i>	1	11	0	11
	<i>Borsisti</i>	0			
	<i>Altre tipologie</i>	0			
<i>Dottorati a carico del PRIN da destinare a questo specifico progetto</i>	0	0	0	0	
<i>Altro personale</i>	0				
<b>TOTALE</b>	<b>10</b>	<b>50</b>	<b>39</b>	<b>89</b>	

## 17 - Costo complessivo del Progetto dell'Unità articolato per voci

Voce di spesa	Spesa in Euro	Descrizione dettagliata (in italiano)	Descrizione dettagliata (in inglese)
<b>Materiale inventariabile</b>	7.000	<i>Strumenti di calcolo (computer e relativi accessori).</i>	<i>Computers and related accessories</i>
<b>Grandi Attrezzature</b>	0		
<b>Materiale di consumo e funzionamento (comprensivo di eventuale quota forfettaria)</b>	7.000	<i>Cancelleria, telefono, spese postali e quota forfettaria.</i>	<i>Stationery, phone and postal expenses, forfait.</i>
<b>Spese per calcolo ed elaborazione dati</b>			
<b>Personale a contratto</b>	20.000	<i>Assegno di ricerca, per un anno, specifico sui temi del progetto.</i>	<i>Post-doc grant, for one year, for working on project themes.</i>
<b>Dottorati a carico del PRIN da destinare a questo specifico progetto</b>	0		
<b>Servizi esterni</b>			
<b>Missioni</b>	21.000	<i>Trasferte per partecipare a convegni ed eventi scientifici.</i>	<i>Missions for participating in scientific events.</i>
<b>Pubblicazioni (*)</b>			
<b>Partecipazione / Organizzazione convegni (*)</b>	7.000	<i>Organizzazione di convegni e incontri relativi ai temi del progetto. Partecipazione a convegni.</i>	<i>Organization of meetings related to the project topics.</i>
<b>Altro (voce da utilizzare solo in caso di spese non riconducibili alle voci sopraindicate)</b>	1.000	<i>Invito di persone esterne per seminari in Bologna su temi attinenti al progetto.</i>	<i>Invitation of external experts for seminars in Bologna on project topics.</i>
<b>Subtotale</b>	63.000		
<b>Costo convenzionale</b>	7.000		
<b>Totale</b>	70.000		

Tutti gli importi devono essere espressi in Euro arrotondati alle centinaia

(\*) sono comunque rendicontabili le spese da effettuare per pubblicazioni e presentazione dei risultati finali della ricerca nei dodici mesi successivi alla conclusione del progetto, purché le relative spese siano impegnate entro la data di scadenza del progetto e purché le pubblicazioni e la presentazione dei risultati avvengano entro nove mesi dalla conclusione del progetto.

## 18 - Prospetto finanziario dell'Unità di Ricerca

Voce di spesa	Importo in Euro
<b>a.1) finanziamenti diretti, disponibili da parte di Università/Enti vigilati di appartenenza dei ricercatori dell'unità operativa</b>	14.000
<b>a.2) finanziamenti diretti acquisibili con certezza da parte di Università/Enti vigilati di appartenenza dei ricercatori dell'unità operativa</b>	
<b>a.3) finanziamenti connessi al costo convenzionale</b>	7.000
<b>b.1) finanziamenti diretti disponibili messi a disposizione da parte di soggetti esterni</b>	
<b>b.2) finanziamenti diretti acquisibili con certezza, messi a disposizione da parte di soggetti esterni</b>	
<b>c) cofinanziamento richiesto al MIUR (max 70% del costo complessivo)</b>	49.000
<b>Totale</b>	70.000

**19 - Certifico la dichiarata disponibilità e l'utilizzabilità dei finanziamenti a.1) a.2) a.3) b.1) b.2)**

SI

Firma \_\_\_\_\_

*I dati contenuti nella domanda di finanziamento sono trattati esclusivamente per lo svolgimento delle funzioni istituzionali del MIUR. Incaricato del trattamento è il CINECA- Dipartimento Servizi per il MIUR. La consultazione è altresì riservata al MIUR - D.G. della Ricerca -- Ufficio IV -- Settore PRIN, alla Commissione di Garanzia e ai referee scientifici. Il MIUR potrà anche procedere alla diffusione dei principali dati economici e scientifici relativi ai progetti finanziati. Responsabile del procedimento è il coordinatore del settore PRIN dell'ufficio IV della D.G. della Ricerca del MIUR.*

Firma \_\_\_\_\_

Data 09/02/2009 ore 00:05

## ALLEGATO

### Curricula scientifici dei componenti il gruppo di ricerca

#### Testo italiano

##### 1. **PETTOROSS I Alberto**

###### **Curriculum:**

Laureato con lode nel 1971 in Ingegneria Elettronica presso l'Università di Roma. Master in Computer Science, Syracuse, USA, nel 1978. Dottorato di ricerca in Computer Science, Edinburgo, Gran Bretagna, nel 1984. Premiato insieme al Prof. Andrzej Skowron dalla Società Matematica Polacca nel 1988.

Contrattista presso l'Università di Roma Sapienza (1974-75); ricercatore presso lo IASI-CNR di Roma (1975-1988); professore associato (1988-2005), straordinario (2005-2008), ordinario (dal 2008) presso l'Università di Roma Tor Vergata.

Pubblicazione di più di cento articoli in vari campi della teoria della programmazione: sistemi di riscrittura; computazione parallela; trasformazione, sviluppo e verifica automatica dei programmi; programmazione logica.

Pubblicazione di sette libri didattici.

Docente invitato presso varie istituzioni, tra cui: lo Stanford Research Center, USA; l'Università di Edimburgo; l'Università di Varsavia; l'Università di Utrecht; l'Argonne National Laboratory, Illinois, USA; l'IFIP, San Paolo, Brasile; l'Università di Copenhagen; l'Università di New York; l'AFCET-GT Programmation en Logique, Parigi; l'M.I.T., Boston, USA; l'École Normale Supérieure de Cachan, Francia; la PUL di Roma; la ESSLII 2000 presso l'Università di Birmingham; l'Università Waseda di Tokyo.

Membro dal 1984 dell'IFIP WG 2.1.

Membro dal 2003 dell'Editorial Board della rivista *Fundamenta Informaticae*.

Vicepresidente dal 2005 dell'Associazione Italiana di Programmazione Logica.

Responsabile dal 1991 del Centro di Ricerca MAP presso il DISP dell'Università di Roma Tor Vergata.

Collaboratore di ricerca dello IASI-CNR dal 1988.

Relatore per i governi belga (1991) e quello olandese (1993) per la valutazione di progetti di ricerca.

Chairman dei convegni:

- *MetaProgramming in Logic*, Uppsala (Svezia), LNCS 649, Springer (1992);
- *Information: Science and Technology for the New Century*, Rome (Italia), ISBN 978-88-465-0562-0 (2000);
- *LOPSTR 2001*, Paphos (Cipro), LNCS 2372, Springer (2001).

Membro permanente dello Steering Committee delle Conferenze LOPSTR.

Membro del comitato di programma di vari congressi, tra cui:

- *International Symposium on Methodologies for Intelligent Systems* Charlotte, USA, (1986-87),
- *Symposium on Partial Evaluation and Semantics based Program Manipulation* (Copenhagen) (1993),
- *CADE 12* (Nancy) (1994),
- *MetaProgramming in Logic '94* (Pisa) (1994),
- *Logic Program Synthesis and Transformation LOPSTR* 1996, 1997, 1998, 1999, 2000, 2001, 2004, 2007, 2008,
- *European Symposium Programming '98* (Lisbon), *Ukrainian Conference UkrPROG* (Kiev),
- *International Conference Rough Sets and Current Trends in Computing* (1998),
- *International Conference Computational Logic CL 2000* (London),
- *ASIA-PEPM'02 Asian Symposium PEPM*, Aizu (Japan), 2002.

Referee per varie riviste internazionali tra cui: *ACM Transactions on Programming Languages and Systems*, *Acta Informatica*, *Computing Journal*, *Journal of Theory and Practice of Logic Programming*, *Journal of Logic and Computation*, *IEEE Software Engineering*, *Theoretical Computer Science*.

Guest editor di due numeri della rivista *Fundamenta Informaticae* sulla Trasformazione di Programmi Logici (2005-06).

Referee per vari congressi tra cui:

*ICALP*, *LICS*, *IFIP World Congress*, *Logic Programming*, *International Conference on Logic Programming*, *IJCAI*, *POPL*, *GI*, *AICA*, *CADE*, *LOPSTR*, *SAS*, *PLILP*, *Principles and Practice of Constraint Programming*, *PEPM*, *ESOP*.

Partecipazione a vari progetti di ricerca, tra cui:

- Progetto *ESPRIT* sulla Trasformazione dei Programmi Logici (1984-89);
- Collaborazione tra lo IASI-CNR di Roma e l'Università di Edimburgo (1981-86);
- Progetto Finalizzato Informatica I e II (1989-1993);
- *Erasmus* (STV-89-I-329/M) con l'Università di Utrecht (1990);
- *Network of Excellence in Computational Logic* (1991);
- *Erasmus* ICP-91-B-1063 per la mobilità (1991-92);
- *COMPULOG II ESPRIT Project* no. 6810 (1992-1995);
- *Humal Capital e Mobility* CHR-X-CT 93-00414 (1993-1996);
- Progetto "Anatra: Strumenti per l'analisi e la trasformazione dei programmi" (1994-1995);
- Progetto tra l'Europa e l'ex Unione Sovietica, *INTAS 93-1702* (1994-1998);
- Progetto CNR Programmazione Logica (1995-1996);
- Progetto Galileo "Modélisation de systèmes concurrents" con l'École Normale Supérieure di Cachan (Francia) (1997-1998);
- Progetto Cofinanziato "Tecniche formali per la specifica, l'analisi, la verifica, la sintesi e la trasformazione di sistemi software" (1998-2000).

###### **Pubblicazioni:**

♦ DANVY OLIVIER, MAIRSON HARRY, HENGLEIN FRITZ, PETTOROSS I ALBERTO (a cura di) (2008). *Automatic Program Development, A Tribute to Robert Paige*. DORDRECHT: SPRINGER, ISBN: 978-1-4020-6584

- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2008). *Totally Correct Logic Program Transformations Via Well-Founded Annotations*. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 21; p. 193-235, ISSN: 1388-3690
- ◆ SENNI VALERIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2008). *A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs*. In: PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON LOGIC PROGRAMMING, M. GARCIA DE LA BANDA AND E. PONTELLI (EDS.), LECTURE NOTES IN COMPUTER SCIENCE 5366, SPRINGER. Udine, Italy, DECEMBER 9-13, 2008, p. 284-300
- ◆ JACOVITTI GIOVANNI, PETTOROSSO ALBERTO, CONSOLO ROSANNA, SENNI VALERIO (a cura di) (2007). *Information: Science and Technology for the New Century*. ROMA: Lateran University Press, ISBN: 978-88-465-0562-0
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: 23rd International Conference on Logic Programming. Porto, Portugal, p. 364-379, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: PROCEEDINGS OF THE 23RD INTERNATIONAL CONFERENCE ON LOGIC PROGRAMMING, V. DAHL AND I. NIEMELÄ (EDS.), LECTURE NOTES IN COMPUTER SCIENCE 4670, SPRINGER. PORTO, PORTUGAL, SEPTEMBER 8-13, 2007, p. 364-379
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2006). *Pettorossi Alberto, Proietti Maurizio, Senni Valerio*. In: 22nd International Conference on Logic Programming. Seattle, Washington, USA, p. 179-195, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2006). *Transformational Verification of Parameterized Protocols Using Array Formulas*. In: 15th International Symposium on Logic Based Program Synthesis and Transformation. London, UKSpringer, p. 23-43, ISBN/ISSN: 0302-9743
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2005). *Automatic Proofs of Protocols via Program Transformation*. In: Monitoring, Security, and Rescue Techniques in Multiagent Systems, p. 99-116, ISBN/ISSN: 16-15-3871
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, RENAULT SOPHIE (2005). *Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism*. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 18; p. 121-210, ISSN: 1388-3690
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *Transformation Rules for Locally Stratified Constraint Logic Programs*. In: M. BRUYNOOGHE; K-K. LAU EDS.. Program Development in Computational Logic. p. 292-340Springer
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *A Theory of Totally Correct Logic Program Transformations*. Proceedings of the ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '04), Verona (Italy). p. 159-168ACM, ISBN/ISSN: 1-58113-835-0
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *Transformations of Logic Programs with Goals as Arguments*. THEORY AND PRACTICE OF LOGIC PROGRAMMING, vol. 4 (4); p. 495-537, ISSN: 1471-0684
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2003). *Combining Logic Programs and Monadic Second Order Logics by Program Transformation*. In: LOPSTR 2002 Twelfth International Workshop on Logic-based Program Development and Transformation. Madrid, Spain, p. 160-181, ISBN/ISSN: 0302-9743
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *A Specialization Technique For Deriving Deterministic Constraint Logic Programs and Its Application to Pattern Matching*. In: Proceedings of AGP'02, 2002 JOINT CONFERENCE ON DECLARATIVE PROGRAMMING, Madrid, Spain., p. 241-257
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Specialization with Clause Splitting For Deriving Deterministic Constraint Logic Programs*. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC'02), Hammamet, Tunisia, 2002, vol. 1, p. 188-193, ISBN/ISSN: 0-7803-7438-X
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Verification of Sets of Infinite State Systems Using Program Transformation*. In: 11th International Workshop on Logic-based Program Synthesis and Transformation. Paphos, Cyprus, p. 111-128, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSO ALBERTO (a cura di) (2002). *Proceedings of the Eleventh International Workshop on Logic-based Program Synthesis and Transformation*. (LOPSTR '01). vol. LNCS 2372, ISBN: 3-540-43915-3
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Program Derivation = Rules + Strategies*. In: A. KAKAS AND F. SADRI EDS.. Computational Logic: Logic Programming and Beyond (Essays in Honour of Robert A. Kowalski - Part I). p. 273-309Springer
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *The List Introduction Strategy for the Derivation of Logic Programs*. FORMAL ASPECTS OF COMPUTING, vol. 13; p. 233-251, ISSN: 0934-5043
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2001). *Automatic Strategies for Specializing Constraint Logic Programs*. In: Proceedings of the 10th International Workshop on Logic-based Program Synthesis and TransformationSpringer, p. 125-146, ISBN/ISSN: 0302-9743
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2001). *Verifying CTL Properties of Infinite State Systems by Specializing Constraint Logic Programs*. In: Second ACM-Sigplan International Workshop on Verification and Computational Logic. Florence, Italy, p. 85-96
- ◆ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2000). *Rules and Strategies for Contextual Specialization of Constraint Logic Programs*. ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE, vol. 30-2, ISSN: 1571-0661
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2000). *Perfect Model Checking via Unfold/Fold Transformations*. In: First International Conference on Computational Logic (CL2000). London, 24-28 July, 2000, Springer, p. 613-628, ISBN/ISSN: ISBN 3-540-67797-6
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1999). *Synthesis and Transformation of Logic Programs Using Unfold/Fold Proofs*. JOURNAL OF LOGIC PROGRAMMING, vol. 41 (2&3); p. 197-230, ISSN: 0743-1066
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1999). *Transforming Inductive Definitions*. In: In Proc. 1999 Intern. Conf. on Logic Programming, De Schreye, ed. Las Cruces, New Mexico, USAThe MIT Press, p. 486-499, ISBN/ISSN: 1061-0464
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1998). *Program Specialization via Algorithmic Unfold/Fold Transformations*. ACM COMPUTING SURVEYS, vol. 30 no. 3es; p. 6-es, ISSN: 0360-0300
- ◆ PETTOROSSO ALBERTO, PROIETTI M, RENAULT S (1997). *Reducing Nondeterminism while Specializing Logic Programs*. In: 24TH ACM SYMP. PRINCIPLES OF PROGRAMM. LANGUAGES (POPL'97). La Sorbonne, Paris, FRANCE, 1997, p. 414-427
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1996). *Rules and Strategies for Transforming Functional and Logic Programs*. ACM COMPUTING SURVEYS, vol. 28, 2; p. 360-414, ISSN: 0360-0300

2. **PROIETTI Maurizio**

**Curriculum:**

Maurizio Proietti si è laureato in Matematica presso l'Università di Roma "La Sapienza" nel 1984. È ricercatore del CNR dal 1 Dicembre 1988, e primo ricercatore dal 28 Dicembre 2001. Presta attualmente servizio presso l'Istituto di Analisi dei Sistemi ed Informatica (Roma).

Ha contribuito alla ricerca nel campo delle metodologie basate sulla logica matematica per lo sviluppo automatico del software e, in particolare, nelle seguenti aree: programmazione funzionale, logica, e con vincoli; trasformazione, sintesi e specializzazione dei programmi; dimostrazione automatica di teoremi; model checking e verifica di sistemi software; semantica e teoria delle categorie. I risultati della sua attività scientifica hanno dato luogo a circa sessanta pubblicazioni su riviste, libri, e atti di convegni internazionali con referee.

Ha tenuto serie di conferenze invitate in varie sedi, tra cui: New York University (USA), European Summer School on Logic, Language, and Information (Birmingham, UK), Waseda University (Tokyo, Japan).

È stato revisore per vari convegni e riviste internazionali, tra cui: Theory and Practice of Software Development (TAPSOFT), Conference on Automated Deduction (CADE), International Conference on Logic Programming (ICLP), International Conference on Database Theory (ICDT), International Joint Conference on Artificial Intelligence (IJCAI), ACM Symposium on Principles of Programming Languages (POPL), European Symposium on Programming (ESOP), IEEE Symposium on Logic in Computer Science (LICS), Rewriting Techniques and Applications (RTA); Theoretical Computer Science, Journal of Logic Programming, ACM Transactions on Programming Languages and Systems, Computer Journal, New Generation Computing, Journal of Symbolic Computation, International Journal of Foundations of Computer Science, Theory and Practice of Logic Programming, Information Processing Letters, Fundamenta Informaticae, Journal of Automata, Languages, and Combinatorics.

È stato membro di diversi comitati scientifici di convegni internazionali e nazionali, tra cui: International Conference on Logic Programming (ICLP), European Symposium on Programming (ESOP), International Symposium on Logic-based Program Development and Transformation (LOPSTR), ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM), International Workshop on Verification and Computational Logic (VCL), Joint Conference on Declarative Programming (APPIA-GULP-PRODE, ), Convegno Italiano di Logica Computazionale (CILC).

È stato program chair dell'International Workshop on Logic-based Program Development and Transformation (LOPSTR'95).

È stato guest editor di un numero speciale della rivista Fundamenta Informaticae sulla trasformazione dei programmi.

È stato correlatore di varie tesi di laurea e dottorato.

Ha partecipato a diversi progetti di ricerca nazionali e internazionali, tra cui: ESPRIT-ALPES (1987-1988), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo (1989-1995), Esprit BRA "Computational Logic II", Human Capital & Mobility: Logic Program Synthesis and Transformation (1993-1996), INTAS "Efficient Symbolic Computing", Progetto Speciale del CNR Strumenti per l'analisi e la trasformazione dei programmi logici (1994-1996), Programma Galileo per la cooperazione italo-francese (1996-1997), Progetto cofinanziato MURST "Sistemi formali per la Specifica, l'Analisi, la Verifica, la Sintesi e la Trasformazione di Sistemi Software" (1998-1999).

È stato responsabile scientifico di unità di ricerca nei progetti: Progetto Coordinato del CNR "Programmazione Logica" (1996-1998), Progetto Coordinato del CNR "Verifica analisi e trasformazione di programmi logici" (1998-1999), Network of Excellence in Computational Logic - COMPUNET (1995-2001), European Network of Excellence in Computational Logic (2002-), Progetto CNR RSTL "Metodi trasformativi per la dimostrazione automatica di teoremi".

**Pubblicazioni:**

♦ PETTOROSSA A, PROIETTI M. (2008). *Totally Correct Logic Program Transformations Via Well-Founded Annotations*. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 21; p. 193-235, ISSN: 1388-3690, doi: 10.1007/s10990-008-9024-6

♦ SENNI V, PETTOROSSA A, PROIETTI M. (2008). *A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs*. In: 24th International Conference on Logic Programming. Udine (Italy), 2008Springer, vol. 5366, p. 284-300, ISBN/ISSN: 0302-9743

♦ PETTOROSSA A, PROIETTI M., SENNI V (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: 23rd International Conference on Logic Programming. Porto, Portugal, 2007Springer, p. 364-379, ISBN/ISSN: 0302-9743

♦ PETTOROSSA A, PROIETTI M., SENNI V (2006). *Proving Properties of Constraint Logic Programs by Eliminating Existential Variables*. In: 22nd International Conference on Logic Programming. Seattle, Washington, USA, 2006Springer, p. 179-195, ISBN/ISSN: 0302-9743

♦ PETTOROSSA A, PROIETTI M., SENNI V (2006). *Transformational Verification of Parameterized Protocols Using Array Formulas*. In: 15th International Symposium on Logic Based Program Synthesis and Transformation. London, UK, 2005Springer, p. 23-43, ISBN/ISSN: 0302-9743

♦ FIORAVANTI F, PETTOROSSA A, PROIETTI M. (2005). *Automatic Proofs of Protocols via Program Transformation*. In: Monitoring, Security, and Rescue Techniques in Multiagent Systems. Plock, Poland, 2004Springer, p. 99-116, ISBN/ISSN: 16-15-3871

♦ PETTOROSSA A, PROIETTI M., RENAULT S (2005). *Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism*. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 18; p. 121-210, ISSN: 1388-3690

♦ FIORAVANTI F, PETTOROSSA A, PROIETTI M. (2004). *Transformation Rules for Locally Stratified Constraint Logic Programs*. In: BRUYNOOGHE M.; LAU K.-K.. Program Development in Computational Logic. p. 292-340Springer, ISBN/ISSN: 3-540-22152-2

♦ PETTOROSSA A, PROIETTI M. (2004). *A Theory of Totally Correct Logic Program Transformations*. In: ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation. Verona, Italy, 2004ACM Press, p. 159-168, ISBN/ISSN: 1-58113-835-0

♦ PETTOROSSA A, PROIETTI M. (2004). *Transformations of Logic Programs with Goals as Arguments*. THEORY AND PRACTICE OF LOGIC PROGRAMMING, vol. 4; p. 495-537, ISSN: 1471-0684

3. **KIZILTAN Zeynep**

**Curriculum:**

Zeynep Kiziltan è ricercatore universitario presso la Facoltà di Scienze MMFFN dell'Università di Bologna dal Novembre 2006. Ha svolto attività di ricerca da giugno 2004 a ottobre 2006 con un assegno di ricerca presso il Dipartimento di Elettronica, Informatica e Sistemistica della stessa università. Ha



conseguito il titolo di dottore di ricerca presso l'Università di Uppsala (Svezia) a marzo 2004 e quello di "docent" (professore associato) a ottobre 2004. La sua tesi di dottorato è stata premiata come migliore tesi europea in intelligenza artificiale del 2004 dall'ECCAI, il comitato di coordinamento europeo per l'intelligenza artificiale.

I temi di ricerca di Zeynep Kiziltan sono incentrati sull'intelligenza artificiale e sulla programmazione a vincoli. Attualmente si occupa di modellazione, vincoli globali, symmetry breaking e di integrazione di tecniche di ricerca locale e programmazione a vincoli. È autore di numerosi articoli scientifici pubblicati in prestigiose riviste e conferenze internazionali (Artificial Intelligence, Constraints, AI Communications, ACM Communications, Annals of Operations Research, IJCAI, AAAI, ECAI, CP e CP-AI-OR).

Zeynep Kiziltan collabora con numerosi gruppi di ricerca ed è stata invitata a tenere seminari e relazioni sulla propria attività di ricerca in prestigiose sedi internazionali e università italiane ed estere. Fa parte del comitato di programma di numerose conferenze e workshop internazionali su intelligenza artificiale e programmazione a vincoli, quali IJCAI, AAAI, ECAI, CP, MICAI, SARA, LION, SymCon e RCRA e serve come revisore per riviste quali Theoretical Computer Science, Artificial Intelligence, Constraints, European Journal of Operational Research, International Journal of Reasoning-based Intelligent Systems e per conferenze quali SAT, ICLP e PDP. È stata inoltre co-chair dei doctoral programme di CP'6 e CP'05, organizzatrice del workshop di IJCAI'05 su modellazione e risoluzione di problemi tramite vincoli, e co-organizzatrice del workshop di CP'04 su simmetrie e problemi di soddisfacimento di vincoli.

#### **Pubblificazioni:**

- ◆ A. M. FRISCH, B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2009). Filtering algorithms for the multiset ordering constraint. *ARTIFICIAL INTELLIGENCE*, vol. 173; p. 299 - 328, ISSN: 0004-3702
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., C.-G. QUIMPER, T. WALSH (2008). The Parameterized Complexity of Global Constraints. In: *Proc. of the 23rd AAAI Conference on Artificial Intelligence*. Chicago, USA, July 13-17, MENLO PARK: AAAI Press, p. 235 - 240
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2008). Slide: A Useful Special Case of the Cardpath Constraint. In: *Proc. of the 18th European Conference on Artificial Intelligence*. Patras, Greece, 21-25 July, AMSTERDAM: IOS Press, p. 475 - 479
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., C.-G. QUIMPER, T. WALSH (2007). Reformulating Global Constraints: The Slide and Regular Constraints. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4612; p. 80 - 92, ISSN: 0302-9743
- ◆ KIZILTAN Z., A. LODI, M. MILANO, F. PARISINI (2007). CP-based Local Branching. In: *Proc. of the 14th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*. Roma, 5-6 Luglio, s.l.: M. Gavanelli, T. Mancini, p. -
- ◆ KIZILTAN Z., A. LODI, M. MILANO, F. PARISINI (2007). CP-based Local Branching. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4741; p. 847 - 855, ISSN: 0302-9743
- ◆ A. FRISCH, B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2006). Propagation Algorithms for Lexicographic Ordering Constraints. *ARTIFICIAL INTELLIGENCE*, vol. 170(10); p. 803 - 834, ISSN: 0004-3702
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). Among, Common, and Disjoint Constraints. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3978; p. 29 - 43, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). Filtering Algorithms for the NValue Constraint. *CONSTRAINTS*, vol. 11(4); p. 271 - 293, ISSN: 1383-7133
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Range Constraint: Algorithms and Implementation. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3990; p. 59 - 73, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Roots Constraint. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4204; p. 75 - 90, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Slide Meta-Constraint. In: *Proceedings of Third International Workshop on Constraint Propagation and Implementation*. Nantes, France, September 2006. Held in conjunction with the Twelfth International Conference on Principles and Practice of Constraint Programming (CP 2006). Nantes, France, September 25, 2006, s.l.: s.n., p. 19 - 30
- ◆ KIZILTAN Z., B. HNIC (2006). The Doctoral Programme of the Twelfth International Conference on Principles and Practice of Constraint Programming, CP 2006, Nantes, France, September 24-29, 2006., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ KIZILTAN Z., B. HNIC (a cura di) (2006). Proceedings of the CP 2006 Doctoral Programme. The Doctoral Programme of the Twelfth International Conference on Principles and Practice of Constraint Programming, CP 2006, Nantes, France, September 24-29, 2006. S.L.: S.N., p. 233
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Among, Common and Disjoint Constraints. In: *Proc. of the 2005 Joint Workshop of the ERCIM and CoLogNet on Constraint Solving and Constraint Logic Programming*, UPPSALA: s.n., p. 223 - 235
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Filtering Algorithms for the NValue Constraint. In: *Proc of RCRA'05, the AI\*IA Workshop on Experimental Analysis and Benchmark of Algorithms for Artificial Intelligence*. Ferrara, Italia, 10 Giugno 2005, s.l.: M. Cadoli, M. Gavanelli, T. Mancini, p. 3 - 12
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Filtering Algorithms for the Nvalue Constraint. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3524; p. 79 - 93, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). The Range and Roots Constraints: Specifying Counting and Occurrence Problems. In: *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI'2005)*, Edinburgh, UK, July 30-August 5, 2005, SAN FRANCISCO, CA: Morgan Kaufmann, p. 60 - 65, ISBN/ISSN: 0-938075-93-4
- ◆ KIZILTAN Z., C. BESSIERE, B. HNIC, T. WALSH (2005). The Fifth Workshop on Modelling and Solving Problems with Constraints. Held at the Nineteenth International Joint Conference on Artificial Intelligence ( IJCAI 2005 ), Edinurgh, Scotland, 31 July, 2005., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ KIZILTAN Z., C. BESSIERE, B. HNIC, T. WALSH. (a cura di) (2005). Proceedings of the Fifth Workshop on Modelling and Solving Problems with Constraints. Held at the Nineteenth International Joint Conference on Artificial Intelligence ( IJCAI 2005 ), Edinurgh, Scotland, 31 July, 2005. S.L.: S.N., p. 88
- ◆ M. MILANO, KIZILTAN Z. (2005). The Doctoral Programme of the Eleventh International Conference on Principles and Practice of Constraint Programming, CP 2005, Sitges(Barcelona), Spain, October 1-5, 2005., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2004). Hybrid Modelling for Robust Solving. *ANNALS OF OPERATIONS RESEARCH*, vol. 130 (1-4); p. 19 - 39, ISSN: 0254-5330
- ◆ B. HNIC, KIZILTAN Z., T. WALSH (2004). Combining Symmetry Breaking with Other Constraints: Lexicographic Ordering with Sums. In: *Proc. of the 8th Int'l Symposium on Artificial Intelligence and Mathematics (AMAI'04)*, s.l.: s.n., p. 93 - 100
- ◆ KIZILTAN Z. (2004). Symmetry Breaking Ordering Constraints. *COMMUNICATIONS OF THE ACM*, vol. 17(3); p. 167 - 169, ISSN: 0001-0782
- ◆ KIZILTAN Z. (2004). Symmetry Breaking Ordering Constraints. *UPPSALA: Uppsala: Dep. of Information Science, CS Division*, p. 249, ISBN: 91-506-1737-0
- ◆ W. HARVEY, KIZILTAN Z. (2004). SymCon'04: THE 4TH INTERNATIONAL WORKSHOP ON SYMMETRY AND CONSTRAINT SATISFACTION

PROBLEMS. A Satellite Workshop of CP'2004 (Tenth International Conference on Principles and Practice of Constraint Programming). 27 September 2004, Toronto, Canada., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]

◆ W. HARVEY, KIZILTAN Z. (a cura di) (2004). *Proceedings of SymCon'04 : THE 4TH INTERNATIONAL WORKSHOP ON SYMMETRY AND CONSTRAINT SATISFACTION PROBLEMS. A Satellite Workshop of CP'2004 (Tenth International Conference on Principles and Practice of Constraint Programming)*. 27 September 2004, Toronto, Canada. S.L.: S.N., p. 104

4. **DI GIUSTO Cinzia**

**Curriculum:**

Cinzia Di Giusto ha terminato il dottorato in Informatica il 31/12/2008 presso il dipartimento di Scienze dell'Informazione, Università di Bologna ed è attualmente assegnista presso la stessa struttura. Gli interessi di ricerca si concentrano nell'area della programmazione logica e dei linguaggi concorrenti.

**Pubblicazioni:**

◆ DI GIUSTO C., GABBRIELLI M, MEO M.C (2009). Expressiveness of multiple heads in CHR. In: *SOFSEM 2009*, vol. 5404

5. **SENNI Valerio**

**Curriculum:**

Nato il 11-07-1980 a Milano, IT. Laurea triennale in Ingegneria Informatica conseguita 10/2002 presso Università di Roma "Tor Vergata" con voto 110/110 e L. Laurea specialistica in Ingegneria Informatica conseguita 11/2004 presso Università di Roma "Tor Vergata" con voto 110/110 e L. Dottorato in Informatica ed Ingegneria dell'Automazione conseguito 05/2008 presso Dipartimento di Informatica Sistemi e Produzione, Facoltà di Ingegneria, Università di Roma "Tor Vergata" con una tesi dal titolo "Transformation Techniques for Constraint Logic Programs with Applications to Protocol Verification". Durante il dottorato ho partecipato alla Scuola Estiva di Logica 2005, organizzata dall'Associazione Italiana di Logica e sue Applicazioni (AILA), dalla Società Italiana di Logica e Filosofia della Scienza (SILFS) e dall'Università degli Studi di Milano. Durante la quale ho seguito due corsi dal titolo: "Teoria della Dimostrazione: Lambda Calcolo" e "Teoria della Dimostrazione: Metodi del Ragionamento Automatico". Assistente al corso di Informatica Teorica negli a.a. 2005/2006, 2006/2007, e 2007/2008 presso l'Università di Roma "Tor Vergata", tenuto dal Prof. A.Pettorossi.

**Pubblicazioni:**

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2008). *Program Transformation for Development, Verification, and Synthesis of Software*, p. 7-14, Technical Report of Dipartimento di Informatica, Università degli Studi di Torino, RT 110/08, June 2008, Essay Dedicated to Alberto Martelli. (To appear in *Intelligenza Artificiale*.)

◆ SENNI V., PETTOROSSO A, PROIETTI M (2008). *A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs*. In: *Lecture Notes in Computer Science*. Udine, Italy, December 9-13, 2008, HEIDELBERG: Springer, vol. 5366, p. 284-300, ISBN/ISSN: 978-3-540-89981-5, doi: 10.1007/978-3-540-89982-2

◆ SENNI V., PETTOROSSO A, PROIETTI M (2008). *Folding Transformation Rules for Constraint Logic Programs*. In: *Online Proceedings of the Italian Annual Meeting on Computational Logic (CILC'08)*. Perugia, Italy, July 10-12, 2008

◆ IACOVITTI G, PETTOROSSO A, CONSOLO R, SENNI V. (a cura di) (2007). *Information: Science and Technology for the New Century*. di: A.A.V.V. - PUL-MURSIA, ISBN: 978-88-465-0562-0

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: *Lecture Notes in Computer Science*. Porto, Portugal, September 8-13, 2007Springer, vol. 4670, p. 364-379, ISBN/ISSN: 978-3-540-74608-9

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2006). *Proving Properties of Constraint Logic Programs by Eliminating Existential Variables*. In: *Lecture Notes in Computer Science*. Seattle, Washington, USA, August 17-20, 2006Springer, vol. 4079, p. 179-195, ISBN/ISSN: 978-3-540-36635-5

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2005). *Proofs of Program Properties via Unfold/Fold Transformations of Constraint Logic Programs (Abstract)*. In: *Transformation Techniques in Software Engineering*. Dagstuhl, Germany, 17/04 - 22/04/2005, ISBN/ISSN: 1862 - 4405

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2005). *Transformational Verification of Parameterized Protocols Using Array Formulas*. In: *Lecture Notes in Computer Science*. London, UK, September 7-9, 2005Springer, vol. 3901, p. 23-43, ISBN/ISSN: 978-3-540-32654-0

◆ PETTOROSSO A, PROIETTI M, SENNI V. (2005). *Verifying parameterized protocols by transforming stratified logic programs*. In: *Online Proceedings of the Italian Annual Meeting on Computational Logic (CILC'05)*. Roma, Italy, June 21-22, 2005

6. **CAMPORESI Ferdinanda**

**Curriculum:**

2008: Laurea Specialistica in Informatica, Università di Bologna. Titolo della tesi: "Analisi Statica di Sistemi Embedded", relatore Prof. Maurizio Gabbrielli.

2007-2008: Internship presso l'"École Normale Supérieure", Parigi, nel gruppo di Semantica e Interpretazione Astratta diretto dal Prof. Patrick Cousot.

2006: Laurea Triennale in Informatica, Università di Bologna. Titolo della tesi "Workflow Management: the pattern perspective", relatore Prof. Maurizio Gabbrielli.

pubblicazioni non disponibili

7. **MAURO Jacopo**

**Curriculum:**

Istruzione e formazione

Date: 27 ottobre 2006 - 17 luglio 2008

Titolo della qualifica rilasciata: Laurea specialistica in Informatica

Titolo tesi: Espressività ed implementazione di linguaggi dichiarativi per il planning

Nome e tipo d'organizzazione erogatrice dell'istruzione e formazione: Università degli Studi di Udine

Livello nella classificazione nazionale o internazionale: 110 e lode

Date: 28 agosto 2003 - 05 ottobre 2006

Titolo della qualifica rilasciata: Laurea triennale in Informatica

Titolo tesi: Un tool per la conversione semi-automatica di programmi Curry in Haskell e viceversa

Nome e tipo d'organizzazione erogatrice dell'istruzione e formazione: Università degli Studi di Udine

Livello nella classificazione nazionale o internazionale: 110 e lode

**Pubblicazioni:**

◆ RAFFAELE CIPRIANO, AGOSTINO DOVIER, MAURO J. (2008). *Compiling and Executing Declarative Modeling Languages to Gecode*. In: LNCS 5366. Udine, Italy, December 2008. Springer, p. 744-748, ISBN/ISSN: 0302-9743

8. **VITALE Antonio**

**Curriculum:**

Antonio Vitale è attualmente iscritto al terzo anno di dottorato in Informatica presso il dipartimento di Scienze dell'Informazione, Università di Bologna.

L'ambito di ricerca principale è legato agli aspetti di espressività e verifica di proprietà di linguaggi bio ispirati con particolare attenzione alla famiglia del kappa-calculus.

**Pubblicazioni:**

◆ VITALE A., COSIMO LANEVE (2008). *Expressivity in the kappa family*. In: *Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIV)*, vol. 218, p. 97-109

◆ VITALE A., GIANCARLO MAURI, CLAUDIO ZANDRON (2008). *Simulation of a bounded symport/antiport P system with Brane calculi*. *BIOSYSTEMS*, vol. 91, Issue 3; p. 558-571, ISSN: 0303-2647

◆ VITALE A., GIANCARLO MAURI (2007). *Communication via Mobile Vesicles in Brane Calculi*. In: *Proceedings of the First Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2006)*, vol. 171/2, p. 187-196

**Testo inglese**

1. **PETTOROSSO Alberto**

**Curriculum:**

Laurea Degree cum laude in Electronic Engineering in 1971 from the University of Rome "Sapienza". - Master in Computer Science in 1978 from Syracuse University, USA. - Ph. D. degree in Computer Science in 1984 from Edinburgh University, Great Britain. - Prize for contributions in Theory of Computation, with Professor Andrzej Skowron, from the Polish Mathematical Society in 1988.

Contract-holder at the University of Rome "Sapienza" (1974-75); Research Fellow at IASI-CNR of Roma (1975-1988); Associate Professor (1988-2005) and Full Professor (from 2005) at the University of Rome "Tor Vergata".

More than one hundred papers published in various areas of Theory of Computation and Theory of Programming and, in particular: rewriting systems; parallel computation; transformation, development, and automatic verification of program; logic programming.

Published seven university books on: *Introduction to Informatics (Part I); Learning Pascal Through Examples; Automata Theory and Formal Languages; Computability, Decidability, and Complexity; Elements of Concurrent Programming; First Order Predicate Calculus (with Maurizio Proietti); Sequential and Concurrent Programming in Java.*

Invited lecturer in various international universities and research centers. Among them: the Stanford Research Center, USA; the University of Edinburgh; the University of Warsaw (Poland); the University of Utrecht (the Netherlands); l'Argonne National Laboratory, Illinois (USA); the IFIP W.G.2.1, San Paolo, Brasil; the University of Copenhagen (Denmark); the University of New York (USA); l'AFCEC-GT Programmation en Logique, Paris (France); l'M.I.T., Boston, USA; l'Ecole Normale Supérieure de Cachan, Francia; the Pontifical Lateran University of Rome (Italy); the ESSLLI 2000 at the University of Birmingham; Waseda University of Tokyo (Japan).

Member of the IFIP WG2.1 since 1984.

Member of the Editorial Board of the *Fundamenta Informaticae Journal* since 2003.

Vice-president of GULP, the Italian Association of Users and Researchers in Logic Programming, since 2005.

Director of the Research Center MAP (Advanced Programming Methodologies) at the Department of Informatics, Systems, and Production of the University of Rome "Tor Vergata" since 1991.

Research collaborator of the Institute IASI-CNR since 1988.

Referee for the Belgian government (1991) and the Duch government (1993) for research projects in logic programming.

Chairman of the following international meetings:

- *MetaProgramming in Logic*, Uppsala (Sweden), LNCS 649, Springer (1992);
- *Information: Science and Technology for the New Century*, Rome (Italy), ISBN 978-88-465-0562-0 (2000);
- *LOPSTR 2001*, Paphos (Ciprus), LNCS 2372, Springer (2001).

Permanent Member of the Steering Committee of the LOPSTR conferences.

Member of the Programme Committees of various International Conferences. Among them:

- *International Symposium on Methodologies for Intelligent Systems* Charlotte, USA, (1986-87),
- *Symposium on Partial Evaluation and Semantics based Program Manipulation (Copenhagen)* (1993),
- *CADE 12 (Nancy)* (1994),
- *MetaProgramming in Logic '94 (Pisa)* (1994),
- *Logic Program Synthesis and Transformation LOPSTR 1996, 1997, 1998, 1999, 2000, 2001, 2004, 2007, 2008*,
- *European Symposium Programming '98 (Lisbon)*, *Ukrainian Conference UkrPROG (Kiev)*,
- *International Conference Rough Sets and Current Trends in Computing* (1998),
- *International Conference Computational Logic CL 2000 (London)*,
- *ASIA-PEPM'02 Asian Symposium PEPM, Aizu (Japan)*, 2002.

Referee for various international journals. Among them: *ACM Transactions on Programming Languages and Systems, Acta Informatica, Computing Journal, Journal of Theory and Practice of Logic Programming, Journal of Logic and Computation, IEEE Software Engineering, Theoretical Computer Science.*

Guest editor of two issues of the *Fundamenta Informaticae Journal on Logic Program Transformation* (2005-06).

Referee for various international conferences. Among them:

- ICALP, LICS, IFIP World Congress, Logic Programming, International Conference on Logic Programming, IJCAI, POPL, GI, AICA, CADE, LOPSTR, SAS, PLILP, Principles and Practice of Constraint Programming, PEPM, ESOP.

Member of various research projects:

- *Project ESPRIT on Logic Program Transformation* (1984-89);
- *Cooperation between the Institute IASI-CNR of Rome and the University of Edinburgh* (1981-86);
- *"Progetto Finalizzato Informatica I and II"* (1989-1993);
- *Erasmus (STV-89-I-329/M) with the University of Utrecht* (1990);
- *Network of Excellence in Computational Logic* (1991);
- *Erasmus ICP-91-B-1063 for Mobility* (1991-92);
- *COMPULOG II ESPRIT Project no. 6810* (1992-1995);
- *Humal Capital e Mobility CHRX-CT 93-00414* (1993-1996);
- *Project "Anatra: Tools for the Analysis and the Transformation of Programs"* (1994-1995);
- *Project between Europe and ex-Soviet-Union States, INTAS 93-1702* (1994-1998);
- *CNR Project Logic Programming* (1995-1996);
- *Project Galileo "Modélisation de systèmes concurrents" with the École Normale Supérieure of Cachan (Francia)* (1997-1998);
- *Progetto Cofinanziato "Formal Techniques for the specification, analysis, verification, synthesis, and transformation of software systems"* (1998-2000).

#### **Pubblicazioni:**

- ◆ DANVY OLIVIER, MAIRSON HARRY, HENGLEIN FRITZ, PETTOROSSO ALBERTO (a cura di) (2008). *Automatic Program Development, A Tribute to Robert Paige*. DORDRECHT: SPRINGER, ISBN: 978-1-4020-6584
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2008). *Totally Correct Logic Program Transformations Via Well-Founded Annotations. HIGHER-ORDER AND SYMBOLIC COMPUTATION*, vol. 21; p. 193-235, ISSN: 1388-3690
- ◆ SENNI VALERIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2008). *A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs*. In: *PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON LOGIC PROGRAMMING*, M. GARCIA DE LA BANDA AND E. PONTELLI (EDS.), *LECTURE NOTES IN COMPUTER SCIENCE 5366*, SPRINGER. Udine, Italy, DECEMBER 9-13, 2008, p. 284-300
- ◆ JACOVITTI GIOVANNI, PETTOROSSO ALBERTO, CONSOLO ROSANNA, SENNI VALERIO (a cura di) (2007). *Information: Science and Technology for the New Century*. ROMA: Lateran University Press, ISBN: 978-88-465-0562-0
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: *23rd International Conference on Logic Programming*. Porto, Portugal, p. 364-379, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2007). *Automatic Correctness Proofs for Logic Program Transformations*. In: *PROCEEDINGS OF THE 23RD INTERNATIONAL CONFERENCE ON LOGIC PROGRAMMING*, V. DAHL AND I. NIEMELÄ (EDS.), *LECTURE NOTES IN COMPUTER SCIENCE 4670*, SPRINGER. PORTO, PORTUGAL, SEPTEMBER 8-13, 2007, p. 364-379
- ◆ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2006). *Pettorossi Alberto, Proietti Maurizio, Senni Valerio*. In: *22nd International*

Conference on Logic Programming, Seattle, Washington, USA, p. 179-195, ISBN/ISSN: 0302-9743

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, SENNI VALERIO (2006). *Transformational Verification of Parameterized Protocols Using Array Formulas*. In: *15th International Symposium on Logic Based Program Synthesis and Transformation*. London, UKSpringer, p. 23-43, ISBN/ISSN: 0302-9743

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2005). *Automatic Proofs of Protocols via Program Transformation*. In: *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, p. 99-116, ISBN/ISSN: 16-15-3871

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO, RENAULT SOPHIE (2005). *Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism*. *HIGHER-ORDER AND SYMBOLIC COMPUTATION*, vol. 18; p. 121-210, ISSN: 1388-3690

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *Transformation Rules for Locally Stratified Constraint Logic Programs*. In: *M. BRUYNOOGHE; K-K. LAU EDS.. Program Development in Computational Logic*. p. 292-340Springer

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *A Theory of Totally Correct Logic Program Transformations*. *Proceedings of the ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '04)*, Verona (Italy). p. 159-168ACM, ISBN/ISSN: 1-58113-835-0

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2004). *Transformations of Logic Programs with Goals as Arguments*. *THEORY AND PRACTICE OF LOGIC PROGRAMMING*, vol. 4 (4); p. 495-537, ISSN: 1471-0684

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2003). *Combining Logic Programs and Monadic Second Order Logics by Program Transformation*. In: *LOPSTR 2002 Twelfth International Workshop on Logic-based Program Development and Transformation*. Madrid, Spain, p. 160-181, ISBN/ISSN: 0302-9743

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *A Specialization Technique For Deriving Deterministic Constraint Logic Programs and Its Application to Pattern Matching*. In: *Proceedings of AGP'02, 2002 JOINT CONFERENCE ON DECLARATIVE PROGRAMMING*, Madrid, Spain., p. 241-257

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Specialization with Clause Splitting For Deriving Deterministic Constraint Logic Programs*. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC'02)*, Hammamet, Tunisia, 2002, vol. 1, p. 188-193, ISBN/ISSN: 0-7803-7438-X

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Verification of Sets of Infinite State Systems Using Program Transformation*. In: *11th International Workshop on Logic-based Program Synthesis and Transformation*. Paphos, Cyprus, p. 111-128, ISBN/ISSN: 0302-9743

♦ PETTOROSSO ALBERTO (a cura di) (2002). *Proceedings of the Eleventh International Workshop on Logic-based Program Synthesis and Transformation*. (LOPSTR '01). vol. LNCS 2372, ISBN: 3-540-43915-3

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *Program Derivation = Rules + Strategies*. In: *A. KAKAS AND F. SADRI EDS.. Computational Logic: Logic Programming and Beyond (Essays in Honour of Robert A. Kowalski - Part I)*. p. 273-309Springer

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2002). *The List Introduction Strategy for the Derivation of Logic Programs*. *FORMAL ASPECTS OF COMPUTING*, vol. 13; p. 233-251, ISSN: 0934-5043

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2001). *Automatic Strategies for Specializing Constraint Logic Programs*. In: *Proceedings of the 10th International Workshop on Logic-based Program Synthesis and Transformation*Springer, p. 125-146, ISBN/ISSN: 0302-9743

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2001). *Verifying CTL Properties of Infinite State Systems by Specializing Constraint Logic Programs*. In: *Second ACM-Sigplan International Workshop on Verification and Computational Logic*. Florence, Italy, p. 85-96

♦ FIORAVANTI FABIO, PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2000). *Rules and Strategies for Contextual Specialization of Constraint Logic Programs*. *ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE*, vol. 30-2, ISSN: 1571-0661

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (2000). *Perfect Model Checking via Unfold/Fold Transformations*. In: *First International Conference on Computational Logic (CL'2000)*. London, 24-28 July, 2000, Springer, p. 613-628, ISBN/ISSN: ISBN 3-540-67797-6

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1999). *Synthesis and Transformation of Logic Programs Using Unfold/Fold Proofs*. *JOURNAL OF LOGIC PROGRAMMING*, vol. 41 (2&3); p. 197-230, ISSN: 0743-1066

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1999). *Transforming Inductive Definitions*. In: *In Proc. 1999 Intern. Conf. on Logic Programming, De Schreye, ed. Las Cruces, New Mexico, USAThe MIT Press*, p. 486-499, ISBN/ISSN: 1061-0464

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1998). *Program Specialization via Algorithmic Unfold/Fold Transformations*. *ACM COMPUTING SURVEYS*, vol. 30 no. 3es; p. 6-es, ISSN: 0360-0300

♦ PETTOROSSO ALBERTO, PROIETTI M, RENAULT S (1997). *Reducing Nondeterminism while Specializing Logic Programs*. In: *24TH ACM SYMP. PRINCIPLES OF PROGRAMM. LANGUAGES (POPL'97)*. La Sorbonne, Paris, FRANCE, 1997, p. 414-427

♦ PETTOROSSO ALBERTO, PROIETTI MAURIZIO (1996). *Rules and Strategies for Transforming Functional and Logic Programs*. *ACM COMPUTING SURVEYS*, vol. 28, 2; p. 360-414, ISSN: 0360-0300

## 2. **PROIETTI Maurizio**

### **Curriculum:**

Maurizio Proietti received a Laurea in Mathematics in March 1984 from the University of Rome "La Sapienza" (Italy). Since December 1988 he has been working as a researcher at the National Research Council of Italy (CNR), and in 2001 he was appointed senior researcher. He works at the "Istituto di Analisi dei Sistemi ed Informatica" (Rome). His research interests are in various methodologies for logic-based software development, among which are: logic, functional, and constraint programming; program transformation, synthesis, and specialization; automated theorem proving; model checking and verification of software systems; semantics and category theory. The results of his research work have been published in about sixty scientific papers on peer reviewed scientific journals, books, and conference proceedings.

He has given invited lectures at various sites, including: New York University (USA), European Summer School on Logic, Language, and Information (Birmingham, UK), Waseda University (Tokyo, Japan).

He has acted as a referee for many conferences and journals, including the following: *Theory and Practice of Software Development (TAPSOFT)*, *Conference*

on Automated Deduction (CADE), International Conference on Logic Programming (ICLP), International Conference on Database Theory (ICDT), International Joint Conference on Artificial Intelligence (IJCAI), ACM Symposium on Principles of Programming Languages (POPL), European Symposium on Programming (ESOP), IEEE Symposium on Logic in Computer Science (LICS), Rewriting Techniques and Applications (RTA); Theoretical Computer Science, Journal of Logic Programming, ACM Transactions on Programming Languages and Systems, Computer Journal, New Generation Computing, Journal of Symbolic Computation, International Journal of Foundations of Computer Science, Theory and Practice of Logic Programming, Information Processing Letters, Fundamenta Informaticae, Journal of Automata, Languages, and Combinatorics.

He has been member of the Program Committees for the following international and italian conferences: International Conference on Logic Programming (ICLP), European Symposium on Programming (ESOP), International Symposium on Logic-based Program Development and Transformation (LOPSTR), ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM), International Workshop on Verification and Computational Logic (VCL), Joint Conference on Declarative Programming (APPIA-GULP-PRODE), Convegno Italiano di Logica Computazionale (CILC). He has been Program Chair of the International Workshop on Logic-based Program Development and Transformation (LOPSTR'95).

He has been guest editor of a special issue of Fundamenta Informaticae on Program Transformation.

He has tutored several master and PhD students.

He has taken part into several national and international research projects, among which are the following: ESPRIT-ALPES Project (1987-1988), CNR Project "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" (1989-1995), Esprit BRA "Computational Logic II", Human Capital & Mobility: Logic Program Synthesis and Transformation (1993-1996), INTAS "Efficient Symbolic Computing", CNR Project "Progetto Speciale - Strumenti per l'analisi e la trasformazione dei programmi logici" (1994-1996), Galileo Programme for French-Italian cooperation (1996-1997), MURST Project "Sistemi formali per la Specifica, l'Analisi, la Verifica, la Sintesi e la Trasformazione di Sistemi Software" (1998-1999).

He has been in charge of the IASI-CNR research unit for the following research projects: CNR Project "Progetto Coordinato Programmazione Logica" (1996-1998), CNR Project "Progetto Coordinato Verifica analisi e trasformazione di programmi logici" (1998-1999), Network of Excellence in Computational Logic - COMPUNET (1995-2001), European Network of Excellence in Computational Logic (2002), CNR Project "RSTL Metodi trasformativi per la dimostrazione automatica di teoremi".

#### **Pubblicazioni:**

- ◆ PETTOROSSA A, PROIETTI M. (2008). Totally Correct Logic Program Transformations Via Well-Founded Annotations. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 21; p. 193-235, ISSN: 1388-3690, doi: 10.1007/s10990-008-9024-6
- ◆ SENNI V, PETTOROSSA A, PROIETTI M. (2008). A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs. In: 24th International Conference on Logic Programming. Udine (Italy), 2008Springer, vol. 5366, p. 284-300, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSA A, PROIETTI M., SENNI V (2007). Automatic Correctness Proofs for Logic Program Transformations. In: 23rd International Conference on Logic Programming. Porto, Portugal, 2007Springer, p. 364-379, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSA A, PROIETTI M., SENNI V (2006). Proving Properties of Constraint Logic Programs by Eliminating Existential Variables. In: 22nd International Conference on Logic Programming. Seattle, Washington, USA, 2006Springer, p. 179-195, ISBN/ISSN: 0302-9743
- ◆ PETTOROSSA A, PROIETTI M., SENNI V (2006). Transformational Verification of Parameterized Protocols Using Array Formulas. In: 15th International Symposium on Logic Based Program Synthesis and Transformation. London, UK, 2005Springer, p. 23-43, ISBN/ISSN: 0302-9743
- ◆ FIORAVANTI F, PETTOROSSA A, PROIETTI M. (2005). Automatic Proofs of Protocols via Program Transformation. In: Monitoring, Security, and Rescue Techniques in Multiagent Systems. Plock, Poland, 2004Springer, p. 99-116, ISBN/ISSN: 16-15-3871
- ◆ PETTOROSSA A, PROIETTI M., RENAULT S (2005). Derivation of Efficient Logic Programs by Specialization and Reduction of Nondeterminism. HIGHER-ORDER AND SYMBOLIC COMPUTATION, vol. 18; p. 121-210, ISSN: 1388-3690
- ◆ FIORAVANTI F, PETTOROSSA A, PROIETTI M. (2004). Transformation Rules for Locally Stratified Constraint Logic Programs. In: BRUYNOOGHE M.; LAU K.-K.. Program Development in Computational Logic. p. 292-340Springer, ISBN/ISSN: 3-540-22152-2
- ◆ PETTOROSSA A, PROIETTI M. (2004). A Theory of Totally Correct Logic Program Transformations. In: ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation. Verona, Italy, 2004ACM Press, p. 159-168, ISBN/ISSN: 1-58113-835-0
- ◆ PETTOROSSA A, PROIETTI M. (2004). Transformations of Logic Programs with Goals as Arguments. THEORY AND PRACTICE OF LOGIC PROGRAMMING, vol. 4; p. 495-537, ISSN: 1471-0684

### 3. **KIZILTAN Zeynep**

#### **Curriculum:**

Zeynep Kiziltan is an assistant professor at the Faculty of Mathematical, Physical and Natural Sciences of the University of Bologna since November 2006. She was a post-doctoral researcher between June 2004 and October 2006 at the Department of Electronics, Computer Sciences and Systems of the same university. She received her Ph.D. degree in March 2004 from the University of Uppsala (Sweden) where she was also appointed as "docent" (associate professor). The Ph.D. thesis of Dr. Kiziltan has won the 2004 Artificial Intelligence Dissertation Award of the European Coordinating Committee for Artificial Intelligence.

Dr. Kiziltan's research focuses on Artificial Intelligence (AI) and Constraint Programming (CP). She is mainly interested in modelling, global constraints, symmetry breaking, and the integration of local search techniques into CP. She is the author of several scientific articles published in prestigious international journals and conferences, such as Artificial Intelligence, Constraints, AI Communications, ACM Communications, Annals of Operations Research, IJCAI, AAI, ECAI, CP, and CP-AI-OR.

Dr. Kiziltan extensively collaborates with research groups abroad. She has been invited to international gatherings and universities (both Italian and foreign) to deliver talks on her research results. Dr. Kiziltan is involved in several professional activities. She has served on the programme committees of prestigious international conferences and workshops on CP and AI, such as IJCAI, AAI, ECAI, CP, MICAI, SARA, LION, SymCon, RCRA, and is a referee for the journals Theoretical Computer Science, Artificial Intelligence, Constraints, European Journal of Operational Research, and International Journal of

Reasoning-based Intelligent Systems, and also for conferences such as SAT, ICLP and PPDP. Besides, she has been the co-chair of the CP'06 and CP'05 Doctoral Programmes, the main organiser of the IJCAI'05 Workshop on Modelling and Solving Problems with Constraints, and the co-organiser of the CP'04 Workshop on Symmetry and Constraint Satisfaction Problems.

**Pubblicazioni:**

- ◆ A. M. FRISCH, B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2009). Filtering algorithms for the multiset ordering constraint. *ARTIFICIAL INTELLIGENCE*, vol. 173; p. 299 - 328, ISSN: 0004-3702
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., C.-G. QUIMPER, T. WALSH (2008). The Parameterized Complexity of Global Constraints. In: *Proc. of the 23rd AAAI Conference on Artificial Intelligence*. Chicago, USA, July 13-17, MENLO PARK: AAAI Press, p. 235 - 240
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2008). Slide: A Useful Special Case of the Cardpath Constraint. In: *Proc. of the 18th European Conference on Artificial Intelligence*. Patras, Greece, 21-25 July, AMSTERDAM: IOS Press, p. 475 - 479
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., C.-G. QUIMPER, T. WALSH (2007). Reformulating Global Constraints: The Slide and Regular Constraints. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4612; p. 80 - 92, ISSN: 0302-9743
- ◆ KIZILTAN Z., A. LODI, M. MILANO, F. PARISINI (2007). CP-based Local Branching. In: *Proc. of the 14th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*. Roma, 5-6 Luglio, s.l: M. Gavanelli, T. Mancini, p. -
- ◆ KIZILTAN Z., A. LODI, M. MILANO, F. PARISINI (2007). CP-based Local Branching. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4741; p. 847 - 855, ISSN: 0302-9743
- ◆ A. FRISCH, B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2006). Propagation Algorithms for Lexicographic Ordering Constraints. *ARTIFICIAL INTELLIGENCE*, vol. 170(10); p. 803 - 834, ISSN: 0004-3702
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). Among, Common, and Disjoint Constraints. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3978; p. 29 - 43, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). Filtering Algorithms for the NValue Constraint. *CONSTRAINTS*, vol. 11(4); p. 271 - 293, ISSN: 1383-7133
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Range Constraint: Algorithms and Implementation. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3990; p. 59 - 73, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Roots Constraint. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4204; p. 75 - 90, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2006). The Slide Meta-Constraint. In: *Proceedings of Third International Workshop on Constraint Propagation and Implementation*. Nantes, France, September 2006. Held in conjunction with the Twelfth International Conference on Principles and Practice of Constraint Programming (CP 2006). Nantes, France, September 25, 2006, s.l: s.n, p. 19 - 30
- ◆ KIZILTAN Z., B. HNIC (2006). The Doctoral Programme of the Twelfth International Conference on Principles and Practice of Constraint Programming, CP 2006, Nantes, France, September 24-29, 2006., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ KIZILTAN Z., B. HNIC (a cura di) (2006). Proceedings of the CP 2006 Doctoral Programme. The Doctoral Programme of the Twelfth International Conference on Principles and Practice of Constraint Programming, CP 2006, Nantes, France, September 24-29, 2006. S.L.: S.N., p. 233
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Among, Common and Disjoint Constraints. In: *Proc. of the 2005 Joint Workshop of the ERCIM and CoLogNet on Constraint Solving and Constraint Logic Programming*, UPPSALA: s.n, p. 223 - 235
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Filtering Algorithms for the NValue Constraint. In: *Proc of RCRA'05, the AI\*IA Workshop on Experimental Analysis and Benchmark of Algorithms for Artificial Intelligence*. Ferrara, Italia, 10 Giugno 2005, s.l: M. Cadoli, M. Gavanelli, T. Mancini, p. 3 - 12
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). Filtering Algorithms for the Nvalue Constraint. *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3524; p. 79 - 93, ISSN: 0302-9743
- ◆ C. BESSIERE, E. HEBRARD, B. HNIC, KIZILTAN Z., T. WALSH (2005). The Range and Roots Constraints: Specifying Counting and Occurrence Problems. In: *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI'2005)*, Edinburgh, UK, July 30-August 5, 2005, SAN FRANCISCO, CA: Morgan Kaufmann, p. 60 - 65, ISBN/ISSN: 0-938075-93-4
- ◆ KIZILTAN Z., C. BESSIERE, B. HNIC, T. WALSH (2005). The Fifth Workshop on Modelling and Solving Problems with Constraints. Held at the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinurgh, Scotland, 31 July, 2005., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ KIZILTAN Z., C. BESSIERE, B. HNIC, T. WALSH. (a cura di) (2005). Proceedings of the Fifth Workshop on Modelling and Solving Problems with Constraints. Held at the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinurgh, Scotland, 31 July, 2005. S.L.: S.N., p. 88
- ◆ M. MILANO, KIZILTAN Z. (2005). The Doctoral Programme of the Eleventh International Conference on Principles and Practice of Constraint Programming, CP 2005, Sitges(Barcelona), Spain, October 1-5, 2005., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ B. HNIC, KIZILTAN Z., I. MIGUEL, T. WALSH (2004). Hybrid Modelling for Robust Solving. *ANNALS OF OPERATIONS RESEARCH*, vol. 130 (1-4); p. 19 - 39, ISSN: 0254-5330
- ◆ B. HNIC, KIZILTAN Z., T. WALSH (2004). Combining Symmetry Breaking with Other Constraints: Lexicographic Ordering with Sums. In: *Proc. of the 8th Int'l Symposium on Artificial Intelligence and Mathematics (AMAI'04)*, s.l: s.n, p. 93 - 100
- ◆ KIZILTAN Z. (2004). Symmetry Breaking Ordering Constraints. *COMMUNICATIONS OF THE ACM*, vol. 17(3); p. 167 - 169, ISSN: 0001-0782
- ◆ KIZILTAN Z. (2004). Symmetry Breaking Ordering Constraints. *UPPSALA: Uppsala: Dep. of Information Science, CS Division*, p. 249, ISBN: 91-506-1737-0
- ◆ W. HARVEY, KIZILTAN Z. (2004). *SymCon'04: THE 4TH INTERNATIONAL WORKSHOP ON SYMMETRY AND CONSTRAINT SATISFACTION PROBLEMS*. A Satellite Workshop of CP'2004 (Tenth International Conference on Principles and Practice of Constraint Programming). 27 September 2004, Toronto, Canada., [IDEAZIONE, PROGETTAZIONE, ORDINAMENTO DI MANIFESTAZIONE]
- ◆ W. HARVEY, KIZILTAN Z. (a cura di) (2004). Proceedings of SymCon'04 : THE 4TH INTERNATIONAL WORKSHOP ON SYMMETRY AND CONSTRAINT SATISFACTION PROBLEMS. A Satellite Workshop of CP'2004 (Tenth International Conference on Principles and Practice of Constraint Programming). 27 September 2004, Toronto, Canada. S.L.: S.N., p. 104

4. **DI GIUSTO Cinzia**

**Curriculum:**

Cinzia Di Giusto has finished her PhD course in Computer Science on 31/12/2008 at Dipartimento di Scienze dell'Informazione, Università di Bologna, she is now receiving an "Assegno di Ricerca" in the same department. Her research interests are in the programming logic area and in the concurrency area.

**Pubblicazioni:**

♦ DI GIUSTO C., GABBRIELLI M, MEO M.C (2009). Expressiveness of multiple heads in CHR. In: SOFSEM 2009, vol. 5404

5. **SENNI Valerio**

**Curriculum:**

Birth date: July 11th, 1980, Milan, Italy. Bachelor degree in Computer Science, October 2002, 110 cum laude. MSc degree in Computer Science, november 2004, 110 cum laude. June 2008, PhD in Computer Science and Automation Engineering, Faculty of Engineering, Department of Computer Science and Automation of the University of Rome "Tor Vergata".  
During the PhD I've attended the Summer School of Logic 2005, organized by the AILA Italian Association for Logic and its Applications, the SILFS Italian Society for Logic and Philosophy of Science and the University of Milan. The school consisted in two courses: "Theorem Proving: Lambda Calculus" and "Methods for Automated Theorem Proving".  
I've been assistant to the course "Theoretical Computer Science" 2005/2006, 2006/2007, and 2007/2008, University of Rome "Tor Vergata", held by Prof. A.Pettorossi.

**Pubblicazioni:**

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2008). Program Transformation for Development, Verification, and Synthesis of Software, p. 7-14, Technical Report of Dipartimento di Informatica, Università degli Studi di Torino, RT 110/08, June 2008, Essay Dedicated to Alberto Martelli. (To appear in *Intelligenza Artificiale*.)

♦ SENNI V., PETTOROSSA A, PROIETTI M (2008). A Folding Algorithm for Eliminating Existential Variables from Constraint Logic Programs. In: *Lecture Notes in Computer Science*. Udine, Italy, December 9-13, 2008, HEIDELBERG: Springer, vol. 5366, p. 284-300, ISBN/ISSN: 978-3-540-89981-5, doi: 10.1007/978-3-540-89982-2

♦ SENNI V., PETTOROSSA A, PROIETTI M (2008). Folding Transformation Rules for Constraint Logic Programs. In: *Online Proceedings of the Italian Annual Meeting on Computational Logic (CILC'08)*. Perugia, Italy, July 10-12, 2008

♦ IACOVITTI G, PETTOROSSA A, CONSOLO R, SENNI V. (a cura di) (2007). *Information: Science and Technology for the New Century*. di: A.A.V.V. - PUL-MURSIA, ISBN: 978-88-465-0562-0

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2007). Automatic Correctness Proofs for Logic Program Transformations. In: *Lecture Notes in Computer Science*. Porto, Portugal, September 8-13, 2007Springer, vol. 4670, p. 364-379, ISBN/ISSN: 978-3-540-74608-9

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2006). Proving Properties of Constraint Logic Programs by Eliminating Existential Variables. In: *Lecture Notes in Computer Science*. Seattle, Washington, USA, August 17-20, 2006Springer, vol. 4079, p. 179-195, ISBN/ISSN: 978-3-540-36635-5

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2005). Proofs of Program Properties via Unfold/Fold Transformations of Constraint Logic Programs (Abstract). In: *Transformation Techniques in Software Engineering*. Dagstuhl, Germany, 17/04 - 22/04/2005, ISBN/ISSN: 1862 - 4405

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2005). Transformational Verification of Parameterized Protocols Using Array Formulas. In: *Lecture Notes in Computer Science*. London, UK, September 7-9, 2005Springer, vol. 3901, p. 23-43, ISBN/ISSN: 978-3-540-32654-0

♦ PETTOROSSA A, PROIETTI M, SENNI V. (2005). Verifying parameterized protocols by transforming stratified logic programs. In: *Online Proceedings of the Italian Annual Meeting on Computational Logic (CILC'05)*. Roma, Italy, June 21-22, 2005

6. **CAMPORESI Ferdinanda**

**Curriculum:**

2008: MSc in Computer Science, University of Bologna. Laurea thesis title "Static Analysis of Embedded Systems", supervisor Prof. Maurizio Gabbrielli.

2007-2008: Internship at "École Normale Supérieure", Paris, in the Abstract Interpretation and Semantics team; supervisor Prof. Patrick Cousot.

2006: BS in Computer Science, University of Bologna. Laurea thesis title "Workflow Management: the pattern perspective", supervisor Prof. Maurizio Gabbrielli.



7. **MAURO Jacopo**

**Curriculum:**

*Education and training*

*Dates: 27 October 2006 - 17 July 2008*

*Title of qualification awarded: Master degree in Computer Science*

*Thesis title: Espressività ed implementazione di linguaggi dichiarativi per il planning (Expressivity and implementation of declarative languages for planning)*

*Name and type of organisation providing education and training: University of Udine (University)*

*Level in national or international classification: 110 cum laude/110*

*Dates 28 August 2003 - 05 October 2006*

*Title of qualification awarded: Bachelor degree in computer science*

*Thesis title: Un tool per la conversione semi-automatica di programmi Curry in Haskell e viceversa (A tool for the semi-automatic conversion of Curry programs into Haskell programs and vice versa)*

*Name and type of organisation providing education and training: University of Udine (University)*

*Level in national or international classification: 110 cum laude/110*

**Pubblicazioni:**

◆ RAFFAELE CIPRIANO, AGOSTINO DOVIER, MAURO J. (2008). *Compiling and Executing Declarative Modeling Languages to Gecode*. In: LNCS 5366. Udine, Italy, December 2008Springer, p. 744-748, ISBN/ISSN: 0302-9743

8. **VITALE Antonio**

**Curriculum:**

*Antonio Vitale is currently in the third year of his PhD course in Computer Science at Dipartimento di Scienze dell'Informazione, Università di Bologna. The main research topics focus on expressiveness and verification issues of bio-inspired languages within the kappa-calculus family.*

**Pubblicazioni:**

◆ VITALE A., COSIMO LANEVE (2008). *Expressivity in the kappa family*. In: *Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIV)*, vol. 218, p. 97-109

◆ VITALE A., GIANCARLO MAURI, CLAUDIO ZANDRON (2008). *Simulation of a bounded symport/antiport P system with Brane calculi*. *BIOSYSTEMS*, vol. 91, Issue 3; p. 558-571, ISSN: 0303-2647

◆ VITALE A., GIANCARLO MAURI (2007). *Communication via Mobile Vesicles in Brane Calculi*. In: *Proceedings of the First Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2006)*, vol. 171/2, p. 187-196