

Curriculum dell'attività scientifica e didattica di Viviana BONO

Viviana Bono è nata a Casale Monferrato (AL) il 29 febbraio 1968
ed è residente a Torino in via Maria Vittoria 23; Tel. 011 8395015;
e-mail: bono@di.unito.it;
url: <http://www.di.unito.it/~bono>

Studi

- Laurea in Scienze dell'Informazione, conseguita il 23 febbraio 1993 presso l'Università degli Studi di Torino, con la votazione di 110/110 e lode.
Tesi di laurea di ricerca nell'area dei fondamenti dei linguaggi funzionali: "Logica Lineare e Tipi Intersezione", relatore prof.ssa Battistina Venneri, controrelatore prof.ssa Simonetta Ronchi della Rocca.
- Nel maggio 1994 vince una borsa annuale per laureati dell'Università di Torino per attività di perfezionamento all'estero. Il sito scelto per tale soggiorno è il "Laboratoire des Mathématiques de Luminy" (già "Laboratoire des Mathématiques Discrètes"), C.N.R.S., a Marsiglia (Francia) e il suo direttore di ricerca è il professor Jean-Yves Girard.
- Nell'a.a. 1994-1995, durante la permanenza a Marsiglia, consegue il diploma di D.E.A. in Matematiche Discrete e Fondamenti dell'Informatica ("Diplôme d'Etudes Approfondies de Mathématiques Discrètes et Fondements de l'Informatique", Université d' Aix-Marseille), sotto la direzione di Laurent Regnier, ricercatore al suddetto *laboratoire*.
Nel *memoire* del D.E.A. "Etude du problème de la relecture en géométrie de l'interaction" viene presentato l'abbozzo di un algoritmo per la rilettera del risultato di un programma funzionale dall'espressione corrispondente in geometria dell'interazione.
- Nel febbraio 1995 inizia il Dottorato di Ricerca in Informatica (X ciclo) con sede presso il Dipartimento di Informatica dell'Università degli Studi di Torino. Il suo supervisore è la prof.ssa Mariangiola Dezani-Ciancaglini. La durata legale del dottorato è di 4 anni.
- L'11 febbraio 1999 presenta la sua tesi di dottorato "Type Systems for the Object-Oriented Paradigm" alla commissione nazionale formata dai professori A. Bertoni (Dipartimento di Scienze dell'Informazione dell'Università di Milano), F. Esposito (Dipartimento di Informatica dell'Università di Bari), G. Ghelli (Dipartimento di Informatica dell'Università di Pisa) e ottiene il titolo di Dottore di Ricerca in Informatica. La tesi è stata favorevolmente recensita da Sophia Drossopoulou (Imperial College, Londra) e da Giuseppe Castagna (E.N.S., Parigi).

Attività accademica e professionale

- Dal 1 gennaio al 31 dicembre 1999 è *lecturer* alla School of Computer Science dell'Università di Birmingham (School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.).
- Dall'11 gennaio 2000 al 31 dicembre 2004 è ricercatore universitario (settore INF/01) al Dipartimento di Informatica dell'Università di Torino (Corso Svizzera 185, 10149 Torino).
- Ha ottenuto l'idoneità alla seconda fascia della docenza (settore INF/01) nel 2003 (Università degli Studi di Udine, decreto rettorale N. 1028 del 23 ottobre 2003).
- Dal 1 gennaio 2005 è professore associato (settore INF/01) al Dipartimento di Informatica dell'Università degli Studi di Torino (Corso Svizzera 185, 10149 Torino). È stata nominata professore associato confermato per il settore scientifico disciplinare INF/01 Informatica, presso la Facoltà di Scienze M.F.N. dell'Università degli Studi di Torino, con D.D.n. 3278 del 28.05.2008, a decorrere dal 1.1.2008.

Attività scientifica

L'attività di ricerca di V. Bono riguarda principalmente lo studio di proprietà e il design di linguaggi object-oriented, ma V. Bono si è anche occupata di logiche substrutturali legate alla nozione di sottotipo. In sintesi:

1. Design di linguaggi e sistemi di tipi per il paradigma object-oriented.
2. Studio della semantica dei tipi intersezione e del subtyping tramite metodologie logiche.

1. Design di linguaggi e sistemi di tipi per il paradigma object-oriented.

In questo filone di ricerca sono utilizzati come strumenti dei metodi formali, quali la semantica operativa e i sistemi di tipi, in modo da lavorare su una base teorica stabile e ben conosciuta. I vantaggi di tale scelta sono la possibilità di analizzare in modo incrementale i requisiti di un linguaggio object-oriented e di poter provare alcune buone proprietà, tra cui la più importante è la proprietà di *soundness*, ovvero la garanzia che se un oggetto è staticamente ben tipato, allora nessuna invocazione di metodo su tale oggetto incorrerà in un errore di "message-not-understood" a tempo di esecuzione.

L'applicazione di metodi formali ai linguaggi di programmazione ad oggetti non ha soltanto il vantaggio di aiutare a comprendere i loro aspetti fondazionali, nati dall'esperienza di programmazione, più che da teorie, ma permette anche di contribuire a chiarire una certa confusione creatasi nell'area: ogni linguaggio reale supporta diverse e disparate caratteristiche,

modellando alcuni aspetti in modo efficiente e ignorandone altri, soprattutto trascurando l'ortogonalità spesso esistente fra tali caratteristiche e creando legami che dipendono solo dall'implementazione usata (ad esempio, fra ereditarietà e relazione di sottotipo). Un esempio riuscito di sinergia tra teoria e pratica è stato lo studio dei generici per Java e la loro introduzione in Java 1.5. La ricerca di V. Bono si colloca in questo contesto di integrazione tra risultati formali e pratica di programmazione.

I risultati dei lavori [R1,R2,CO1,CO2,CO3,CO4,CO5,CO6,CO7,CO8] sono confluiti e sono stati completati nella tesi di dottorato.

Nei lavori [R1,R4,CO1,CO3,CO5,CO6,CO8,CO9] sono stati progettati e analizzati formalmente alcuni calcoli (intesi come *core* di linguaggi di programmazione) che integrano primitive object-oriented in linguaggi funzionali con tipi, trattando in modo sistematico caratteristiche quali il subtyping, l'ereditarietà modellata tramite costrutti primitivi oppure costrutti diretti, la semantica degli oggetti e delle operazioni su di essi. Il lavoro [CO13] è un *survey* sugli oggetti estensibili, che sono alla base della maggior parte dei risultati dei lavori sopra citati. In [CO16], viene presentata una metodologia per la prova della decidibilità dell'inferenza di tipo di un frammento importante del calcolo degli oggetti estensibili.

Il lavoro [CO7] è importante perché getta le basi di risultati posteriori, in particolare, propone di utilizzare l'ereditarietà basata sul costrutto di *mixin* in alternativa all'ereditarietà classica basata sul costrutto di classe. Un *mixin* è una classe parametrica rispetto alla sua superclasse, che va quindi “completata” per poter essere istanziata. La sua incompletezza intrinseca si presta bene ad essere utilizzata in contesti di codice mobile e distribuito, come esemplificato nei lavori [R5,CO11,CO12,CO14,CO15,CO17,CO18]. In tali lavori, si affrontano problematiche tipiche dei sistemi distribuiti e mobili, quali quelli dovuti alla coordinazione del codice e ai conflitti che nascono dalla composizione di componenti eterogenee. Anche in tale contesto mobile e distribuito, il principio-guida è quello di garantire, tramite proprietà statiche quali quelle indotte da opportuni sistemi di tipi, un alto livello di soundness (ovvero di assenza di errori a tempo di esecuzione).

Il concetto di incompletezza, insieme al meccanismo della composizione, ha portato allo studio degli “oggetti incompleti”, oggetti che possono avere componenti non ancora disponibili a *client* esterni, che mettono insieme alcune buone proprietà dei linguaggi basati sulle classi (quali la disciplina di programmazione) e dei linguaggi basati sugli oggetti (quali la flessibilità). L'idea di base degli oggetti incompleti è già presente in [R2,CO2,CO4], ma è in [R6,CO19-CO24] che si possono trovare alcune proposte mature di linguaggi *core* basati sugli oggetti incompleti, insieme allo studio delle problematiche che presentano, quali quelle legate all'integrazione di opportune nozioni di sottotipo con la composizione di oggetti. In [CO30,CO31], due versioni degli oggetti incompleti (una basata sulla *consultation* e una sulla *delegation*) vengono integrate in un contesto di tipi nominali Java-like. Questi risultati sono piuttosto interessanti, perché gli studi precedenti erano tutti in contesti di tipi strutturali.

Un altro aspetto metodologico di grande importanza nella ricerca di V. Bono è l'*input* che

arriva dall'ingegneria del software, in particolare sul problema del riuso del software, che il meccanismo dell'ereditarietà dei linguaggi object-oriented *mainstream* non risolve. I lavori di V. Bono degli ultimi anni risentono delle influenze di concetti come le architetture software rappresentate dai *design pattern* e le metodologie di sviluppo *agile*. In particolare, nei lavori sugli oggetti incompleti, si può vedere come alcuni pattern molto usati, quali il *decorator*, possono essere internalizzati in un linguaggio di programmazione, con l'ovvio vantaggio di avere un controllo a tempo di compilazione di queste architetture software.

In [CO25,CO26,CO27], il problema del riuso del software viene affrontato, rispettivamente, nei primi due lavori, proponendo una versione altamente modulare dei costruttori degli oggetti, in alternativa a quelli monolitici della maggior parte dei linguaggi ad oggetti *mainstream*, e nel terzo, presentando una proposta per evitare conflitti accidentali di nomi (di metodi e campi), quando il software già utilizzato in applicazioni *third-party* viene modificato o aggiornato. Di JavaMIP (Java esteso con i costruttori modulari) esiste un'implementazione, i cui riferimenti si trovano in [CO26].

I lavori [C28,CO29,CO32] si basano sul costrutto di *trait*, che è una collezione di metodi che si può utilizzare per comporre classi, in alternativa o insieme all'ereditarietà, in modo da delegare a quest'ultima la costruzione concettuale delle gerarchie e lasciare il compito del riuso ai *trait*. I primi due lavori propongono una versione Java-like dei *trait*, in cui tutta la potenza dei *trait* e dei loro operatori viene integrata in un contesto di tipi nominali. Il terzo lavoro propone una tecnica di re-ingegnerizzazione da codice Java classico a codice Java con *trait*. Questa tecnica è stata parzialmente implementata con un *tool*, i cui riferimenti si trovano in [CO32].

2. Studio della semantica dei tipi intersezione e del subtyping tramite metodologie logiche.

Un interessante approccio allo studio dei sistemi di tipi è l'*isomorfismo Curry-Howard*, in cui i programmi sono visti come dimostrazioni in un sistema logico e i tipi sono il corrispondente dei teoremi in tale sistema. Per modellare i tipi funzionali, è sufficiente il frammento implicativo della logica intuizionistica del prim'ordine. Invece, le caratteristiche *proof-teoretiche* dei tipi intersezione, ovvero un più stretto legame con le premesse della loro deduzione, rendono difficile un'interpretazione logica in un frammento puramente proposizionale. Nella sua tesi di laurea, V. Bono approccia il problema studiando un isomorfismo tra tipi intersezione e un frammento della *logica lineare*, similmente a quanto fatto da B. Venneri con la logica rilevante minimale. Alcuni risultati riguardanti la logica rilevante minimale in senso più fondazionale si possono trovare in [AC1].

In seguito, nella stessa linea di ricerca e insieme al prof. Jerzy Tiuryn di Varsavia, V. Bono ha esteso il sistema di sottotipo polimorfo con l'operatore di tipo "prodotto cartesiano" [R3]. Il nuovo sistema è stato estensivamente studiato dal punto di vista logico ed è risultato interessante poichè la congettura che le relazioni di sottotipo che coinvolgono il prodotto non fossero derivabili nel sistema primitivo si è rivelata vera.

Infine, la ricerca sul significato logico dei tipi intersezione è evoluta in [R7], lavoro in cui si presenta un calcolo tipato con i tipi intersezione, in cui viene introdotto un costruttore “parallelo” sui termini, che corrisponde al costruttore di tipo dell’intersezione.

Riferimenti bibliografici

Vedere allegato “Elenco delle pubblicazioni trasmesse da Viviana BONO”.

Citazioni in libri

- Gli articoli: “An Imperative First-Order Calculus with Object Extension” (V. Bono, K. Fisher), “A Core Calculus of Classes and Objects” (V. Bono, A.J. Patel, V. Shmatikov, J.C. Mitchell), “A Core Calculus of Classes and Mixins” (V. Bono, A.J. Patel, V. Shmatikov) sono citati nel libro di Benjamin C. Pierce “Types and Programming Languages”, The MIT Press, 2002.
- L’articolo: “Matching for the lambda calculus of objects” (V. Bono, M. Bugliesi) è menzionato a pagina 364 del libro di Kim B. Bruce “Foundations of Object-Oriented Programming Languages: Types and Semantics”, The MIT Press, 2002 (gli autori sono citati nell’indice analitico).

Altre attività accademiche

- *Task Leader* per il *task* “Lambda Calcoli con Oggetti” del progetto COFIN’99 “Concurrency Theory, Higher-Order Languages, and Type Structures (TOSCA)”, coordinatore nazionale: prof. Ugo Montanari, 1999-2001.
- Ha partecipato a progetti nazionali ed internazionali e contribuito alla stesura di proposte di progetti, sia nazionali che internazionali, successivamente finanziati, fra cui:
 - international reserch project DART (<http://www.macs.hw.ac.uk/DART/>);
 - international reserch project TYPES (<http://www.cs.chalmers.se/Cs/Research/Logic/Types/>);
 - progetto nazionale PRIN EOS (<http://bart.disi.unige.it/EOS/>).
 - progetto nazionale PRIN EOSDUE (attivo)
(http://bart.disi.unige.it/EOS2/www_static_pages/index.html);
- *Invited lecturer* alla “Mini-school on theoretical computer science” ad Aussois, Francia, 29 Gennaio–1 Febbraio 2003 (<http://www.lama.univ-savoie.fr/david/aussois>).
- Co-organizzatore, con Michele Bugliesi dell’Università di Venezia, del workshop WOOD 2003 (Workshop on Object-Oriented Developments 2003) affiliato a ETAPS 2003, che si è tenuto in Varsavia, Polonia, nell’aprile 2003 (<http://www.dsi.unive.it/wood2003>).

- Co-organizzatore, con Michele Bugliesi dell'Università di Venezia e Sophia Drossopoulou dell'Imperial College, Londra, U.K., del workshop WOOD 2004 (Workshop on Object-Oriented Developments) affiliato a CONCUR 2004, che si è tenuto in Londra, U.K., nell'agosto 2004 (<http://www.dsi.unive.it/wood2004>).
- Membro dei seguenti Comitati di Programma:
FOOL 9 (2002, <http://www.cs.hmc.edu/~stone/FOOL/FOOL9.html>), SAC 2004 - OOPS track (<http://oops.disi.unige.it/OOPS04/>), SAC 2005 - OOPS track (<http://oops.disi.unige.it/OOPS05/>), MFCS 2006 (<http://www.mfcs.sk/mfcs2006/>), MFCS 2008 (<http://mfcs08.mat.umk.pl/>), FOOL'09 (<http://www.cs.cmu.edu/~aldrich/FOOL09/>).
- Membro dello Steering Committee del workshop internazionale “Foundations of Object Oriented Languages” (FOOL) (<http://www.cs.hmc.edu/~stone/FOOL/>).
- Recensore di libri del suo settore di ricerca e di articoli sottomessi alle riviste e conferenze del suo settore di ricerca.

Soggiorni di studio e ricerca all'estero

- Marsiglia (Francia), a.a. 1994-1995: si veda la sezione **Studi** di questo stesso documento.
- Università di Stanford (USA), gennaio–luglio e ottobre–dicembre 1997: collaborazione come *visiting researcher* con il gruppo di “Programming Languages” guidato dal professor John C. Mitchell.
- Varsavia (Polonia), visite annuali di durata settimanale o bisettimanale tra il novembre 1996 e il febbraio 2007: collaborazioni con i prof.ri J. Tiuryn e P. Urzyczyn e Jaroslaw Kuśmierek (studente di dottorato) dell'Università di Varsavia.
- Birmingham (U.K.): *lectureship* per l'anno 1999.

Studenti di dottorato

- E' stata co-advisor di Silvia Likavec (insieme a Pierre Lescanne, École Normale Supérieure de Lyon, Francia), che ha ottenuto il Dottorato in Informatica in co-tutela nel febbraio 2005.
- E' stata revisore per la tesi di dottorato di Sonia Fagorzi (Università di Genova, Italia) nel 2005.
- E' co-advisor di Jaroslaw Kuśmierek (insieme a Pawel Urzyczyn, Università di Varsavia, Polonia), che sta svolgendo un programma di PhD in Computer Science all'Università di Varsavia, Polonia.

Attività didattica - Corsi

- a.a. 07/08:
 - Programmazione I e Laboratorio - B (Teoria, Turno di Laboratorio T1) - Corso di Laurea in Informatica, primo anno, 8 CFU.
 - Sperimentazioni di Ingegneria del Software - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 4 CFU.
 - Informatica II (introduzione alle basi di dati e alle reti di calcolatori). - Corso di Laurea in Scienze Strategiche (Civili, II anno), 6 CFU.

- a.a. 06/07:
 - Programmazione I e Laboratorio - B (Teoria, Turno di Laboratorio T1) - Corso di Laurea in Informatica, primo anno, 8 CFU.
 - Sperimentazioni di Ingegneria del Software - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 4 CFU.
 - Informatica II (introduzione alle basi di dati e alle reti di calcolatori). - Corso di Laurea in Scienze Strategiche (Civili, II anno), 6 CFU.

- a.a. 05/06:
 - Programmazione I e Laboratorio - B (Teoria, Turni di Laboratorio T1 e T3) - Corso di Laurea in Informatica, primo anno, 8 CFU.
 - Sperimentazioni di Ingegneria del Software - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 4 CFU.
 - Informatica II (introduzione alle basi di dati e alle reti di calcolatori) - Corso di Laurea in Scienze Strategiche (Civili, II anno), 6 CFU.

- a.a. 04/05:
 - Programmazione I e Laboratorio - B (Turno di Laboratorio T4) - Corso di Laurea in Informatica, primo anno, 3 CFU.
 - Basi di Dati e Laboratorio - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 6 CFU.
 - Sperimentazioni di Ingegneria del Software - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 2 CFU.
 - Informatica II (introduzione alle basi di dati e alle reti di calcolatori) - Corso di Laurea in Scienze Strategiche (Civili, II anno), 6 CFU.

- a.a. 03/04:

- Basi di Dati e Laboratorio - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti", 6 CFU.
- Sperimentazioni di Ingegneria del Software - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti" 2 CFU.
- Informatica II (introduzione alle basi di dati e alle reti di calcolatori) - Corso di Laurea in Scienze Strategiche (Civili, II anno), 6 CFU.
- a.a. 02/03:
 - Programmazione II e Laboratorio - B - Corso di Laurea in Informatica.
 - Basi di Dati e Laboratorio - Corso di Laurea in Informatica, Indirizzo "Sistemi e Reti".
 - Sistemi di Elaborazione dell'Informazione (introduzione alla programmazione tramite l'uso di Java) - Diploma Universitario di Operatore di Beni Culturali (Facolt di Lettere e Filosofia, sezione distaccata di Biella).
- a.a. 01/02:
 - Programmazione II e Laboratorio - B - Corso di Laurea in Informatica.
 - Informatica Applicata (introduzione alle basi di dati, algebra relazionale, SQL, modello entity-relationship) - Diploma Universitario di Operatore di Beni Culturali (Facolt di Lettere e Filosofia, sezione distaccata di Biella).
- a.a. 00/01:
 - Laboratorio di Informatica: programmazione - Diploma in Informatica.
- a.a. 99/00:
 - Laboratorio di Informatica: programmazione - Diploma in Informatica (assistenza a Ugo de'Liguoro).
 - Data Structures - Conversion Master degree, School of Computer Science, The University of Birmingham.
 - Programming Methodology - Conversion Master degree, School of Computer Science, The University of Birmingham.

Attività didattica - Tesi triennali

(ordinate secondo il cognome del tesista)

- X-Klaim: mobilità e mixin, Antonio ALTAMURA, 2003 - 2004, aprile.
- Progettazione di un supporto alla didattica, Francesca CALABRÒ, 2005 - 2006, aprile.
- Progettazione di un software con funzioni di calendario e rubrica, Carlo CARLINO, 2005 - 2006, aprile.
- Costrutti linguistici per il riuso del software, Stefano DAL FORNO, Relatore: Ferruccio DAMIANI. Co-relatore: Viviana BONO, 2006 - 2007, giugno.
- I Design Pattern in Java, Roberto DE GIUSEPPE, 2004 - 2005, aprile.
- Analisi e sviluppo di sistemi per la segmentazione editoriale, Marco DI IULIO, Co-Relatore: Alberto MESSINA, 2006 - 2007, giugno.
- X-Klaim: Un linguaggio per computazioni orientate agli oggetti e mobili in ambiente distribuito, Simone GIORDANO, 2003 - 2004, febbraio.
- Un'implementazione del modello di memoria di Java, Valdete GJONI, 2005 - 2006, aprile.
- Sviluppo di uno strumento per l'insegnamento della programmazione in Java, Ilir HALILAJ, 2007 - 2008, giugno.
- Uno strumento per la visualizzazione dell'esecuzione di programmi Java, Michele LICATA, 2006 - 2007, dicembre.
- Un'applicazione Python per la gestione di database MySQL, Marco MARCHIARO, 2005 - 2006, dicembre.
- JavaMIP2Java: un preprocessore per un'estensione di Java, Giovanni MONTEFERRANTE, 2005 - 2006, settembre.
- Un possibile approccio alla soluzione di alcuni problemi legati all'ereditarietà singola nei linguaggi object-oriented, Marco NADDEO, Co-Relatore: Lorenzo BETTINI, 2006 - 2007, aprile.
- Haskell: il linguaggio, Luca PEINETTI, 2004 - 2005, aprile.
- Il linguaggio di programmazione Scala, Enrico PIGOZZI, 2003 - 2004, febbraio.
- Sviluppo e implementazione di uno strumento di supporto per la didattica, Erika PONTONIO, 2006 - 2007, giugno.
- Object-oriented middleware, Marco RASA, 2004 - 2005, aprile.

- Sviluppo Agile di un C.U.P per la gestione delle prestazioni sanitarie, Domenico SOLAZZO, 2005 - 2006, aprile.
- Programmazione Aspect-Oriented con AspectJ, Andrea VAGLIENGO, 2003 - 2004, aprile.
- Gestione del sistema di Help Desk nelle organizzazioni medio/grandi, Federico VALETTI, 2004 - 2005, aprile.
- Il linguaggio Haskell, Alessia VISCONTI, 2004 - 2005, aprile.
- Compilatore Z: XML tra source e target, Giuseppe ZUFFANTI, 2003 - 2004, dicembre.

Attività dipartimentali

È membro della Commissione Web del Dipartimento di Informatica. In particolare, si è occupata della progettazione di una base di dati locale al Dipartimento di Informatica per raccogliere le informazioni sui prodotti di ricerca (in collaborazione con i colleghi dei Servizi ICT), ha gestito dal 2002 la raccolta dei dati per il Nucleo di Valutazione dell'Università di Torino e ha curato la migrazione dei dati dalla base di dati locale al Catalogo d'Ateneo (in collaborazione con i colleghi dei Servizi ICT).

Data

Firma