

Curriculum Vitae di Fausto Spoto

Informazioni generali

- Data e luogo di nascita: 13 gennaio 1973 a Palermo
- Residenza: Via XX Settembre, 52, Verona
- Telefono: +39 3204352527
- e-mail: `fausto.spoto@univr.it`
- Nazionalità italiana
- Lingue conosciute: Inglese e Francese scritti e parlati
- Servizio civile espletato presso l'Azienda Sanitaria Locale di Pisa.

Studi e collaborazioni scientifiche

- 7/1991 Diploma di maturità scientifica conseguito con voto finale di 60 sessantesimi
- 7/1995 Laurea con lode in *Scienze dell'Informazione* conseguita presso il Dipartimento di Informatica dell'Università di Pisa con una tesi dal titolo *Semantica Concreta e Astratta di Prolog con Cut*. Relatore: prof Giorgio Levi. Controrelatore: prof Egon Boerger.
- 10/1998-3/1999 Visita la School of Computer Studies della University of Leeds, Inghilterra, come parte del dottorato di ricerca, sotto la supervisione locale della dott.a Patricia M. Hill.
- 2/2000 Dottorato in informatica conseguito presso il Dipartimento di Informatica dell'Università di Pisa. Titolo della tesi: *Analysis of Logic Programs through Linear Refinement*. Supervisore: prof Giorgio Levi. Referees: prof Michael Codish e prof Maurice Bruynooghe.
- 1-12/2000 Postdoc presso il centro di ricerca IRISA dell'INRIA/CNRS di Rennes, Francia, sotto la supervisione del prof Thomas Jensen.
- 1/2001-10/2002 Postdoc presso il Dipartimento Scientifico e Tecnologico (ora di Informatica) di Verona, sotto la supervisione del prof Roberto Giacobazzi.
- 9-12/2001 Ricercatore invitato presso la School of Computing della University of Leeds, dove collabora con la dott.a Patricia M. Hill nell'ambito del progetto EPSRC GR/R53401.
- 6/2002 Visita il Nijmeegs Instituut voor Informatica en Informatiekunde della Katholieke Universiteit Nijmegen, Olanda, dove collabora con il dottor Erik Poll.
- 7-8/2002 Visita la IT University e il DIKU di Copenhagen, Danimarca, dove collabora con il prof Fritz Henglein.
- 10-11/2002 Visita la School of Computing di Leeds dove collabora con la dott.a Patricia M. Hill.
- 11/2002-10/2005 Ricercatore universitario presso il Dipartimento di Informatica di Verona.
- 7-9/2003 Visita l'Université de Rennes I, Francia, dove collabora col prof Thomas Jensen.
- 11/2003 Visita l'Université de La Réunion, Francia, dove collabora col prof Frédéric Mesnard.
- 10/2004 Visita l'Université de La Réunion, Francia, dove collabora col prof Frédéric Mesnard.
- 11/2005-oggi Professore associato presso il Dipartimento di Informatica di Verona.
- 6/2006 Visita l'Universidad Politécnica de Madrid, dove tiene un seminario sull'analisi statica del Java bytecode.

- 10/2006 Visita l'Université de La Réunion, Francia, dove collabora coi prof Frédéric Mesnard ed Étienne Payet.
- 10/2007 Visita l'Université de La Réunion, Francia, dove collabora coi prof Frédéric Mesnard ed Étienne Payet.
- 12/2007-1/2008 Visita La Chiang Mai University (Tailandia) con una borsa ASEM-DUO, dove insegna un corso di interpretazione astratta.
- 10/2008 Visita l'Université de La Réunion, Francia, dove collabora coi prof Frédéric Mesnard ed Étienne Payet.
- 7/2009-9/2009 Visita l'University of Washington (Stati Uniti) dove collabora col professor Michael Ernst.
- 11/2010 Contribuisce a fondare la società Julia Srl (www.juliasoft.com), di cui diventa presidente.
- 7/2011-9/2011 Visita l'università di Aachen (Germania) tramite una borsa di studio della von Humboldt Stiftung.

Borse di studio e collaborazioni a progetti di ricerca

- 7/1999 Borsa europea Training and Mobility of Researchers (TMR) per partecipare alla conferenza *Practice and Principles of Declarative Programming '99* a Parigi.
- 1999 Collabora al progetto inglese EPSRC gr/m05645 *Software Support for Constraint Logic Programming*, finanziato per il periodo ottobre 1998-settembre 2001.
- 2000 Collabora, durante il postdoc a Rennes, al progetto di ricerca europeo *SECSAFE* sulla sicurezza di sistemi informatici basata su analisi statica, finanziato per il periodo 2000-2003.
- 2001-2002 Collabora a Verona al progetto MURST *Abstract Interpretation, Type-Systems and Control-Flow Analysis*, finanziato per il periodo 2000-2002.
- 2001 È coautore, insieme a Patricia M. Hill, del progetto di ricerca EPSRC gr53401 *Escape Analysis for Object-Oriented Languages*, finanziato per il periodo ottobre 2001-settembre 2002.
- 2002 Collabora, durante la visita a Nijmegen, al progetto di ricerca europeo *VERIFICARD* sulla verifica automatica di carte a microprocessore programmate in JavaCard, finanziato per gli anni 2001-2003.
- 2002-2004 Collabora a Verona al progetto MURST *CoVer: Constraint-Based Verification of Reactive Systems*, finanziato per il periodo 2002-2004.
- 2004 È coautore, insieme a Patricia M. Hill, del progetto di ricerca *Static Analysis for Numerical Stability* finanziato dalla Royal Society of Sciences per il periodo settembre 2004-agosto 2006.
- 2002-2005 Collabora a Verona al progetto FIRB *SPY-Mod: Interpretazione Astratta e Model Checking per la Verifica di Sistemi Embedded*, finanziato per il periodo 2002-2005.
- 2005-2007 Collabora a Verona al progetto MURST *AIDA* finanziato per il triennio 2005-2007.
- 2008 Ottiene una borsa ASEM-DUO per visitare la Chiang Mai University (Tailandia), dove insegna un corso sull'interpretazione astratta.
- 2008 Collabora a Verona al nuovo progetto PRIN *AIDA* finanziato per il triennio 2008-2010.
- 2009 Ottiene una borsa di mobilità *CooperInt* per recarsi all'Università di Washington, Seattle, Stati Uniti, dove collabora col prof. Michael Ernst.
- 2011 Ottiene una borsa della prestigiosa von Humboldt Stiftung per recarsi in studio all'università di Aachen, in Germania.

Sintesi dell'attività scientifica

La sua attività di ricerca è stata inizialmente centrata sull'analisi di programmi informatici scritti in linguaggi logici [1,5,12,21], al fine di derivare proprietà di condivisione di strutture dati fra variabili [3,14,18], *freeness* [3,13,14] e tipo [2,11,15]. In tale contesto, lo sviluppo delle analisi statiche è stato basato sull'interpretazione astratta e la teoria del raffinamento lineare dei domini, che permette di aggiungere direzionalità e composizionalità alle analisi statiche.

L'attività scientifica si è poi in parte spostata, centrandosi sull'analisi statica di programmi Java e Java bytecode [8,16,29,34,35,36], al fine di verificare proprietà dei programmi prima della loro esecuzione, come le classi degli oggetti legati alle variabili [4,17], la fuga delle strutture dati dall'ambito sintattico in cui sono allocate [6,19,20], l'insieme dei campi degli oggetti modificati durante l'esecuzione di una parte di codice [22,35]. La diffusione del linguaggio Java per la programmazione di applicazioni di rete o locali e la diffusione del bytecode Java come strumento di scambio dei programmi Java compilati danno a questa ricerca un'importanza industriale crescente, anche in conseguenza della sempre maggiore coscienza, da parte delle aziende software, dell'importanza della verifica del codice prima della commercializzazione dei programmi. L'uso di Java e Java bytecode in contesti *insicuri* come i telefoni cellulari apre altre prospettive alla verifica dei programmi prima della loro commercializzazione. Dal punto di vista teorico, tutti questi risultati sono stati basati sulla teoria dell'*interpretazione astratta*, sviluppata negli anni '70 dai coniugi Cousot. Il vantaggio di questa teoria è di poter costruire delle analisi statiche che sono dimostrabilmente corrette; inoltre essa permette di comparare analisi diverse rispetto alla loro precisione e di ottenere analisi ottimali. Fausto Spoto ha contribuito ad applicare l'interpretazione astratta a linguaggi di programmazione che usano strutture dati dinamiche in memoria (quindi anche ai linguaggi a oggetti) e ai linguaggi di basso livello (come il Java bytecode).

In questo contesto, Fausto Spoto ha contribuito a sviluppare uno schema teorico formale all'interno del quale molte analisi statiche possono essere sviluppate in modo denotazionale, in modo da poter essere naturalmente composizionali, sensibili al flusso e al contesto e capaci di modellare proprietà funzionali del codice e non solo proprietà degli stati della computazione. Questo è stato ottenuto con lo sviluppo di una trasformazione *magic-sets* di programmi che fornisce informazioni sui punti interni di un programma anche usando un'analisi denotazionale [29]. Questo ha permesso lo sviluppo teorico e l'implementazione di analisi denotazionali per il flusso di informazioni all'interno del codice [7,23,26], per la condivisione fra strutture dati [24], per la nullness di variabili e campi [31], per la lunghezza dei cammini di puntatori dalle variabili dei programmi e quindi la terminazione [8,27,32,33] e per la non ciclicità delle strutture dati legate alle variabili [28]. Queste analisi sono state migliorate usando informazioni sui parametri che ogni metodo può modificare durante la sua esecuzione [30]. Esse sono state inoltre implementate all'interno dell'analizzatore Julia [24], scritto da Fausto Spoto, che permette già adesso di analizzare programmi di dimensione elevata (fino a 10000 metodi). Un'interfaccia grafica permette l'uso dell'analizzatore via internet.

I risultati della ricerca di Fausto Spoto devono essere in primo luogo giudicati dal punto di vista della pulizia formale e della qualità delle dimostrazioni di correttezza ottenute per tutte le analisi statiche citate. Inoltre va valutato il loro impatto tecnologico: non solo tali analisi sono state implementate all'interno dell'analizzatore Julia, ma la qualità e l'efficienza delle implementazioni è tale che società come la Aonix (www.aonix.com) e la Google (www.google.com) hanno, di loro iniziativa, contattato Fausto Spoto per usare le analisi o svilupparne di nuove. A tal fine, risulta sempre più chiaro che le analisi statiche di Julia e le nuove analisi in corso di sviluppo potranno avere un valore economico nel prossimo futuro. È nata quindi l'idea di costruire una piccola realtà imprenditoriale intorno al programma Julia, sotto forma di spin-off universitaria di Verona. Una proposta in tal senso ha vinto la prima selezione del concorso *StartCup Veneto* per idee imprenditoriali innovative. Ne è conseguita la scrittura di un *business plan* della società, sulla base del quale è stata poi fondata Julia Srl nel novembre 2010.

Esperienza didattica

Inizia la sua attività didattica nel 1997 con esercitazioni per il corso di *Programmazione* di Pisa, in collaborazione col prof Dino Pedreschi, e poi a Leeds per il corso di *Functional Programming*, in collaborazione col dottor Mark Tarver. Una volta a Verona, tiene inizialmente il *Laboratorio di Architetture*, per poi stabilizzare dal 2002 il suo insegnamento sui corsi di *Programmazione, Linguaggi e Tecniche Speciali di Programmazione* (un corso su programmazione logica e Prolog) e *Compilatori*. Per quest'ultimo corso è stato necessario identificare un programma compatibile col numero relativamente piccolo di ore di lezione disponibili ma al contempo abbastanza significativo da includere elementi di teoria ed esperienze dirette di sviluppo di un piccolo compilatore. Il corso è stato inoltre centrato sulla compilazione di un linguaggio ad oggetti, differenziandosi così da corsi più tradizionali sulla compilazione. A tal fine è stato realizzato un piccolo compilatore didattico Kitten per un linguaggio ad oggetti. Durante alcune visite alla Réunion, insegna corsi di programmazione imperativa e logica. Si è anche occupato fino al 2008 dei corsi di *Informatica di Base* per non informatici presso l'Università di Verona. Questi corsi, sebbene di livello elementare, hanno richiesto un impegno notevole di chiarezza e capacità didattica al fine di renderne i contenuti accessibili a studenti non informatici.

Prodotti software

Il suo lavoro di ricerca ha portato allo sviluppo dei seguenti prodotti software (ulteriori informazioni si trovano nella descrizione dell'attività scientifica):

- **Julia**: un analizzatore generico e localizzato per l'intero Java bytecode. Include analisi di classe per l'estrazione delle applicazioni, analisi di non interferenza, di fuga, di condivisione di strutture dati, di non ciclicità delle strutture dati, di nullness e di terminazione. È in assoluto il più evoluto sistema di analisi per il Java bytecode e quindi anche per Java. È al momento utilizzabile da linea di comando e tramite un'interfaccia web.
- **Kitten**: un compilatore didattico usato dagli studenti del corso di *Compilatori* a Verona. Il suo scopo è di permettere agli studenti di visionare i risultati intermedi della compilazione, dalla lista dei token alla sintassi astratta, dall'informazione di tipo al codice intermedio, e di intervenire direttamente sul compilatore per aggiungere nuove funzionalità al linguaggio compilato, sfruttando la struttura ad oggetti del compilatore stesso.

Pubblicazioni

Riviste internazionali

1. Fausto Spoto. *Operational and Goal-Independent Denotational Semantics for Prolog with Cut*. In **Journal of Logic Programming**, 42(1):1-46, January 2000.
2. Patricia M. Hill and Fausto Spoto. *Generalising Def and Pos to Type Analysis*. In **Journal of Logic and Computation**, 12(3), June 2002, pages 497-542.
3. Giorgio Levi and Fausto Spoto. *Pair-Independence and Freeness Analysis through Linear Refinement*. In **Information and Computation**, 182(1), pages 14-52, 2003.
4. Fausto Spoto and Thomas Jensen. *Class Analyses as Abstract Interpretations of Trace Semantics*. In **ACM TOPLAS**, volume 25, September 2003, pages 578-630.
5. Patricia M. Hill and Fausto Spoto. *Logic Programs as Compact Denotations*. In **Elsevier Computer Languages, Systems and Structures**, volume 29, issue 3, pages 45-73, October 2003.
6. Patricia M. Hill and Fausto Spoto. *Deriving Escape Analysis by Abstract Interpretation*. In **Higher Order and Symbolic Computation**, volume 19, pages 415-463, 2006.

7. Fausto Spoto. *Optimality and Condensing of Information Flow through Linear Refinement*. In **Theoretical Computer Science**, 2007, volume 388, pages 53-82. Elsevier.
8. Fausto Spoto, Fred Mesnard and Etienne Payet. *A Termination Analyser for Java Bytecode Based on Path-Length*. In **ACM TOPLAS**, volume 32, number 3, 2010.
9. Fausto Spoto and Etienne Payet. *Magic-sets for Localised Analysis of Java Bytecode*. In **Higher-Order and Symbolic Computation**, volume 23, number 1, pages 29–86, 2010.
10. Fausto Spoto. *Precise null-Pointer Analysis*. In **Software and Systems Modeling**, volume 10, number 2, pages 219–252, 2011.

Conferenze internazionali

11. Giorgio Levi and Fausto Spoto. *An Experiment in Domain Refinement: Type Domains and Type Representations for Logic Programs*. In Catuscia Palamidessi, Hugh Glaser, and Karl Meinke, editors, proc. of *PLILP/ALP'98*, Principles of Declarative Programming, volume 1490 of **Lecture Notes in Computer Science**, pages 152-169, Pisa, Italy, September 1998.
12. Fausto Spoto and Giorgio Levi. *Abstract Interpretation of Prolog Programs*. In A. M. Haeberer, editor, proc. of *AMAST'98*, the 7th International Conference on Algebraic Methodology and Software Technology, volume 1548 of **Lecture Notes in Computer Science**, pages 455-470, Amazonia, Manaus, Brazil, January 1999.
13. Patricia M. Hill and Fausto Spoto. *Freeness Analysis through Linear Refinement*. In Static Analysis Symposium, *SAS'99*, volume 1694 of **Lecture Notes in Computer Science**, pages 85-100, Venice, Italy, September 1999.
14. Giorgio Levi and Fausto Spoto. *Non Pair-Sharing and Freeness Analysis through Linear Refinement*. In proc. of *PEPM'00*, the ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, pages 52-61, Boston, USA, January 2000.
15. Patricia M. Hill and Fausto Spoto. *Analysis of Downward Closed Properties of Logic Programs*. In *AMAST'00*, Algebraic Methodology and Software Technology, volume 1816 of **Lecture Notes in Computer Science**, pages 181-196, Iowa City, USA, May 2000.
16. Fausto Spoto. *Watchpoint Semantics: A Tool for Compositional and Focussed Static Analyses*. In P. Cousot, editor, *SAS'01*, the Static Analysis Symposium, volume 2126 of **Lecture Notes in Computer Science**, pages 127-145. Paris, France, July 2001.
17. Thomas Jensen and Fausto Spoto. *Class Analysis of Object-Oriented Programs through Abstract Interpretation*. In F. Honsell and M. Miculan, editors, *FOSSACS'01*, Foundations of Software Science and Computation Structure, volume 2030 of **Lecture Notes in Computer Science**, pages 261-275. Genova, Italy, April 2001.
18. Gianluca Amato and Fausto Spoto. *Abstract Compilation for Sharing Analysis*. In H. Kuchen and K. Ueda, editors, proc. of *FLOPS'01*, the Fuji International Symposium on Functional and Logic Programming, volume 2024 of **Lecture Notes in Computer Science**, pages 311-325, Tokyo, Japan, March 2001.
19. Patricia M. Hill and Fausto Spoto. *A Foundation of Escape Analysis*. In H. Kirchner and C. Ringeissen editors, proc. of *AMAST'02*, Algebraic Methodology and Software Technology, volume 2422 of **Lecture Notes in Computer Science**, pages 380-395, St. Gilles les Bains, La Réunion island, France, September 2002.

20. Patricia M. Hill and Fausto Spoto. *A Refinement of the Escape Property*. In A. Cortesi, editor, proc. of *VMCAI'02* workshop on Verification, Model Checking and Abstract Interpretation, volume 2294 of **Lecture Notes in Computer Science**, pages 154-166, Venice, Italy, January 2002.
21. Patricia M. Hill and Fausto Spoto. *Logic Programs as Compact Denotations*. In G. Gupta editor, proc. of *PADL'03*, Practical Aspects of Declarative Languages, volume 2562 of **Lecture Notes in Computer Science**, pages 339-356, New Orleans, Louisiana, USA.
22. Fausto Spoto and Erik Poll. *Static Analysis for JML's assignable Clauses*. In G. Ghelli editor, proc. of *FOOL-10*, the 10th International Workshop on Foundations of Object-Oriented Languages, New Orleans, Louisiana, USA, January 2003.
23. Samir Genaim and Fausto Spoto, *Information Flow Analysis for Java Bytecode*. In R. Cousot editor, proc. of *VMCAI'05*, the Sixth International Conference on Verification, Model Checking and Abstract Interpretation, volume 3385 of **Lecture Notes in Computer Science**, pages 346-362. Paris, France, January 2005.
24. Fausto Spoto. *Julia: A Generic Static Analyser for the Java Bytecode*. In proc. of *FTfJP'05*, the 7th Workshop on Formal Techniques for Java-like Programs, Glasgow, Scotland, July 2005.
25. Stefano Secci and Fausto Spoto, *Pair-Sharing Analysis of Object-Oriented Programs*. In proc. of *SAS'05*, the 12th International Static Analysis Symposium, volume 3672 of **Lecture Notes in Computer Science**, pages 320-335. London, United Kingdom, September 2005.
26. Fausto Spoto, *Information Flow is Linear Refinement of Constancy*. In proc. of *ICTAC'05*, the International Colloquium on Theoretical Aspects of Computing, volume 3722 of **Lecture Notes in Computer Science**, pages 351-365. Hanoi, Vietnam, October 2005.
27. Fausto Spoto, Patricia M. Hill and Étienne Payet. *Path-Length Analysis for Object-Oriented Programs*. Presented at *EAAI'06*, the 1st International Workshop on Emerging Applications of Abstract Interpretation, March 2006. Vienna, Austria
28. Stefano Rossignoli and Fausto Spoto, *Detecting Non-Cyclicity by Abstract Compilation into Boolean Functions*. In proc. of *VMCAI'05*, the 7th International Conference on Verification, Model Checking and Abstract Interpretation, volume 3855 of **Lecture Notes in Computer Science**, pages 95-110. Charleston, South Carolina, USA, January 2006.
29. Étienne Payet and Fausto Spoto. *Magic-Sets Transformation for the Analysis of Java Bytecode*. In proc. of *SAS'07*, Static Analysis Symposium, August 2007, Kogens Lyngby, Denmark, volume 4634 of **Lecture Notes in Computer Science**, pages 452-467
30. Samir Genaim and Fausto Spoto. *Constancy Analysis*. In Marieke Huisman, editor, *FTfJP'08*, 10th Workshop on Formal Techniques for Java-like Programs, Paphos, Cyprus, July 2008.
31. Fausto Spoto. *Nullness Analysis in Boolean Form*. In proc. of *SEFM'08*, the 6th IEEE International Conference on Software Engineering and Formal Methods, Cape Town, South Africa, November 2008, pages 21-30, **IEEE Computer Society Press**.
32. Étienne Payet and Fausto Spoto. *Experiments with Non-Termination Analysis for Java Bytecode*. In Samir Genaim and Elvira Albert editors, *Bytecode'09*, the 4th International Workshop on Bytecode Semantics, Verification, Analysis and Transformation, York, UK, March 2009. Volume 253(3) of **Electronic Notes in Theoretical Computer Science**, pages 83-96.

33. Fausto Spoto, Lunjin Lu and Fred Mesnard. *Using CLP Simplifications to Improve Java Bytecode Termination Analysis*. In Samir Genaim and Elvira Albert editors, *Bytecode'09*, the 4th International Workshop on Bytecode Semantics, Verification, Analysis and Transformation, York, UK, March 2009. Volume 253(3) of **Electronic Notes in Theoretical Computer Science**, pages 129-144.
34. Fausto Spoto. *The Nullness Analyser of Julia*. In Edmund M. Clarke and Andrei Voronkov editors, *LPAR'10*, the 16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, Dakar, Senegal, April 2010. Volume 6355 of **Lecture Notes in Computer Science**, pages 405-424.
35. Fausto Spoto and Michael D. Ernst. *Inference of Field Initialization*. In *ICSE'11*, the 33rd ACM International Conference on Software Engineering, pages 231-240. Honolulu, Hawaii, 2011.
36. Etienne Payet and Fausto Spoto. *Static Analysis of Android Programs*. In *CADE'11*, the 23rd International Conference on Automated Deduction. Volume 6803 of **Lecture Notes in Computer Science**, pages 439-445. Wroclaw, Poland.

Curatele

37. Fausto Spoto (ed.). *Proceedings of the First Workshop on Bytecode Semantics, Verification, Analysis and Transformation* (Bytecode'05). Volume 141, Issue 1 of **Electronic Notes in Theoretical Computer Science**, pages 1-274 (5 December 2005). Edinburgh, UK, April 2005.
38. Marieke Huisman and Fausto Spoto (eds). *Proceedings of the Second Workshop on Bytecode Semantics, Verification, Analysis and Transformation* (Bytecode'07). Volume 190, Issue 1 of **Electronic Notes in Theoretical Computer Science**, pages 1-160 (31 July 2007). Braga, Portugal, March 2007.
39. Fausto Spoto (ed). *Proceedings of the Software Verification track* of the 23rd Annual ACM Symposium on Applied Computing (SV'08). Pages 326-394. **ACM Press**. Fortaleza, Brazil, March 2008.

Organizzazione di eventi scientifici

- Membro dell'organizing committee di *SAS'98* (Static Analysis Symposium) e *PLILP/ALP'98* (Programming Languages, Implementations, Logics and Programs/Algebraic and Logic Programming), Pisa, Italy, 1998
- Organizing committee chair di *AMiLP'03* (Algebraic Methods in Language Processing), Verona, Italy, 2003
- Program committee member di *WLPE'03* (Workshop on Logic Programming Environments), Mumbai, India, 2003
- Workshops coordinator di *ICLP'04* (International Conference on Logic Programming), Saint-Malo, France, 2004
- Workshops coordinator e local organisation chair di *LOPSTR'04+PEPM'04+PPDP'04+SAS'04* (International Symposium on Logic-based Program Synthesis and Transformation, ACM SIGPLAN 2004 Symposium on Partial Evaluation and Semantics Based Program Manipulation, ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming e Static Analysis Symposium), Verona, 2004
- Programme committee member di *CILC'04* (Convegno Italiano di Logica Computazionale), Parma, 2004

- Workshop chair di *Bytecode'05*, First Workshop on Bytecode Semantics, Verification, Analysis and Transformation, Edinburgh, Scotland, UK, 2005.
- Programme committee member di *CILC'05* (Convegno Italiano di Logica Computazionale), Roma, 2005
- Programme committee member della software verification track di SAC'06 (ACM Symposium on Applied Computing), Dijon, Francia, 2006
- Workshop co-chair di *Bytecode'07*, Second Workshop on Bytecode Semantics, Verification, Analysis and Transformation, Braga, Portogallo, 2007
- Programme co-chair della software verification track di SAC'07 (ACM Symposium on Applied Computing), Seoul, Corea del Sud, 2007
- Programme chair della software verification track di SAC'08 (ACM Symposium on Applied Computing), Fortaleza, Brazil, 2008.
- Programme Committee member di *Bytecode'08*, *Third Workshop on Bytecode Semantics, Verification, Analysis and Transformation*, Budapest, Hungary, 2008.
- Programme committee member di *Bytecode'09*, *Fourth Workshop on Bytecode Semantics, Verification, Analysis and Transformation*, York, UK, 2009.
- Programme committee member di *PEPM'10*, *ACM SIGPLAN 2010 Workshop on Partial Evaluation and Program Manipulation*.
- Programme committee member di *Bytecode'10*, *Fifth Workshop on Bytecode Semantics, Verification, Analysis and Transformation*, Cyprus, 2010.
- Programme committee member di *Bytecode'11*, *Sixth Workshop on Bytecode Semantics, Verification, Analysis and Transformation*, Saarbrücken, Germany, 2011.

Attività di referaggio di articoli scientifici

Oltre che per i sei workshop di cui è stato programme committee member o chair (enumerati sopra), Fausto Spoto è stato contattato per attività di referaggio dalle seguenti riviste internazionali: *Information and Computation*, *Journal of Systems and Software*, *Theoretical Computer Science*, *Mathematical Structures in Computer Science*, *New Generation Computing*, *ACM Transactions on Computational Logic*, *ACM Transactions on Programming Languages and Systems*, *Theory and Practice of Logic Programming*, *Journal of Computer Science and Technology*; e per le seguenti conferenze internazionali: *Algebraic Methodology And Software Technology*, *European Symposium on Programming*, *International Colloquium on Automata, Languages and Programming*, *International Conference on Logic Programming*, *International Symposium on Logic-based Program Synthesis and Transformation*, *Mathematical Foundations of Programming Semantics*, *Principles and Practice of Declarative Programming*, *Workshop on Quantitative Aspects of Programming Languages*, *Static Analysis Symposium* e infine *Verification, Model Checking and Abstract Interpretation*. Tutti i referaggi sono sempre stati effettuati nei tempi richiesti, senza sollecitazioni da parte della rivista o conferenza.

Verona, 9 novembre 2011