CURRICULUM VITÆ

Massimo Merro

Professore Ordinario in Informatica

1 Informazioni generali

Indirizzo

Dipartimento di Informatica, Università degli Studi di Verona,
Strada le Grazie 15, 37134 Verona (Italy)
Phone: (+39) 045 802 7992
E-mail: massimo.merro@univr.it
Web: http://profs.scienze.univr.it/∼merro/

Formazione

- *PhD in Computer Science*, con menzione "*Trés honorable avec félicitations du jury*", ottenuto presso l'*École Nationale Supérieure des Mines de Paris* in Sophia-Antipolis, Francia, 2000. Titolo della tesi: "Locality in the π-calculus and applications to distributed objects". Supervisore: Dr. Davide Sangiorgi. Finanziato da un *TMR Marie Curie Research Training Grant*.
- *Laurea con lode* in *Scienze dell'Informazione*, Università degli Studi di Pisa. Titolo della tesi: "Priorities in Statecharts". Relatore: Prof. Andrea Maggiolo-Schettini (1996).

Qualifiche e posizioni

- *Professore di I fascia*, SSD INF/01, SC 01/B1, Informatica, presso il Dipartimento di Informatica dell'Università degli Studi di Verona (2018-).
- *Coordinatore del Collegio dei Docenti* del Dottorato in Informatica di Verona (2016-2022);
- *Professore di II fascia*, SSD INF/01, SC 01/B1, Informatica, presso il Dipartimento di Informatica dell'Università degli Studi di Verona (2006-2018).
- *Ricercatore Universitario* presso il medesimo dipartimento (2002-2006).
- *Research Fellow* presso il "Laboratoire des Méthodes de Programmation, Institut d'Informatique Fondamentale, Faculté Informatique et Communications", *École Polytechnique Fédérale de Lausanne*, Svizzera (2002).
- *Research Fellow* presso la "School of Cognitive and Computing Science", *University of Sussex*, UK (2000-2002).

Interessi di ricerca

- Fondamenti semantici di sistemi ciberfisici e sistemi in ambito "Internet of Things".
- Analisi della sicurezza di sistemi ciberfisici e sistemi in ambito "Internet of Things".
- Verifica di protocolli e sistemi complessi attraverso tecniche semantiche e di analisi statica.
- Semantica dei linguaggi di programmazione per sistemi concorrenti, distribuiti e mobili.

2 Attività di ricerca scientifica

2.1 Dati bibliometrici

Elsevier Scopus

I dati seguenti sono stati estratti dal database di *Elsevier Scopus*, in data 28 Dicembre 2023.

- articoli indicizzati: 73
- citazioni: 1238
- h-index: 23

Google Scholar

I dati seguenti sono stati estratti dal database di *Google Scholar*, in data 28 Dicembre 2023.

- citazioni: 2067
- h-index: 26
- i10-index: 51

2.2 Valutazioni VQR (dal 2004 al 2019)

- All'interno della campagna VQR 2015-2019 tutti e due gli articoli riportati di seguito hanno ricevuto valutazione *eccellente*:
  - A. Cerone, M. Merro, M. Hennessy. Modelling MAC-Layer Communications in Wireless Systems. Logical Methods in Computer Science 11(1:18):1-59, 2015.
  - R. Lanotte and M. Merro. A semantic theory of the Internet of Things. Information and Computation 259(1):72-101, 2018.
- All'interno della campagna VQR 2011-2014 tutti e due gli articoli riportati di seguito hanno ricevuto valutazione *elevato*:
  - M. Merro, F. Ballardin, E. Sibilio. A timed calculus for wireless systems. Theoretical Computer Science 412(47):6585-661, 2011.
  - M. Merro and E. Sibilio. A calculus of trustworthy ad hoc networks. Formal Aspects of Computing 25(5):801-832, 2013.
- All'interno della campagna VQR 2004-2010 tutti e tre gli articoli riportati di seguito hanno ricevuto una valutazione *eccellente*:
  - M. Merro and F. Zappa Nardelli. Behavioural Theory for Mobile Ambients. Journal of the ACM 52(6):961-1023, 2005.
  - M. Merro and M. Hennessy. A Bisimulation Semantic Theory of Safe Ambients. ACM Transactions on Programming Languages and Systems 28(2):290-330, 2006.
  - M. Merro. An Observational Theory for Mobile Ad Hoc Networks (full version). Information and Computation 207(2):194-208, 2009.

2.3 Comitati di programma di convegni internazionali/nazionali

- 20° *International Conference on Availability, Reliability and Security* (ARES'24), Ghent, Belgium, 2024.
- 7° *International Workshop on Verification and mOnitoring at Runtime EXecution* (VORTEX'22), Berlin, Germany, 2022.
- 42° *IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems* (FORTE'22), Lucca, Italy, 2022.
- 7° International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2022), Valencia, Spain, 2022.
- 6° International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2021), Barcelona, Spain, 2021.
- 1° *IEEE International Conference on Internet of Things and Intelligent Applications* (ITIA'20), Zhenjiang, China, 2020.
- 20° *Italian Conference on Theoretical Computer Science* (ICTCS'19), Como, Italy, 2019.
- 14° *International Conference on Embedded Software* (EMSOFT'17), Seoul, South Korea, 2017.
- 36° *IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems* (FORTE'16), Heraklion, Crete, 2016.
- 43° *International Colloquium on Automata, Languages and Programming* (ICALP'16) – Track B: Logic, Semantics, Automata and Theory of Programming, Rome, Italy, 2016.
- 42° *International Colloquium on Automata, Languages and Programming* (ICALP'15) – Track C: Foundations of Networked Computation: Models, Algorithms and Information Management, Kyoto, Japan, 2015.
- 38° *International Colloquium on Automata, Languages and Programming* (ICALP'11) – Track C: Foundations of Networked Computation: Models, Algorithms and Information Management, Zurich, Switzerland, 2011.
- 4° *International Conference on Frontier of Computer Science and Technology* (FCST'09), Shanghai, China, 2009.
- 2° *International Workshop on Formal Methods for Wireless Systems* (FMWS'09), Bologna, Italy, 2009.
- 2° *International Meeting on Membrane Computing and Biologically Inspired Process Calculi* (MeCBIC'08), Iasi, Romania, 2008.
- 1° *International Workshop on Formal Methods for Wireless Systems* (FMWS'08), Toronto, Canada, 2008.
- 17° *International Conference on Concurrency Theory* (CONCUR'06), Bonn, Germany, 2006.
- 10° *International Workshop on Expressivity in Concurrency* (EXPRESS'03), Marseilles, France, 2003.
- 9° *International Workshop on Expressivity in Concurrency* (EXPRESS'02), Brno, Czech Republic, 2002.

2.4 Comitati editoriali di riviste internazionali

- Associate Editor di *Frontiers in ICT, Computer and Network Security*, (2014-) `http://journal.frontiersin.org/journal/ict/section/computer-and-network-security`;
- Associate Editor di *Open Computer Science*, `http://www.degruyter.com/view/j/comp`, indicizzato su WoS e a breve anche su Scopus (2015-);
- Associate Editor di *Mobile Information Systems*, indicizzato su WoS e Scopus (2014-) `https://www.hindawi.com/journals/misy/`.

2.5 Supervisione di dottorandi e postdoc

- Research Fellow, Dr. Denis Donadel. Titolo del progetto: *Advanced ICS Honeypots*, Università degli Studi di Verona, 2024-present.
- PhD student, Marco Lucchese. Titolo provvisorio della tesi: *Design,implementation and evaluation of a physics-aware honeynet for Industrial Control Systems.* Università degli Studi di Verona, 2020-2024.
- Research Fellow, Dr. Youssef Driouich. Titolo del progetto: *Process Comprehension of Industrial Physical Processes*, Università degli Studi di Verona, 2022.
- PhD student, Dr. Andrei Munteanu. Titolo della tesi: *Formal Approaches to Control Systems Security: From Static Analysis to Runtime Enforcement*, Università degli Studi di Verona, 2017-2021.
- Research Fellow, Dr. Michele Pasqua. Titolo del progetto: *Security Static Analysis for Internet of Things*, Università degli Studi di Verona, 2018-2019.
- Research Fellow, Dr. Fabio Mogavero. Titolo del progetto: *Formal Verification of Cyber-Physical System Security*, Università degli Studi di Verona, 2017.
- Visiting PhD student, Dr. Mojgan Kamali, Abo Akademi University, Turku, Finland. Titolo del progetto: *Statistical model checking of ad hoc routing protocols*. Sett-Dic 2016.
- Research Fellow, Dr. Damiano Macedonio. Titolo del progetto: *Formal Verification of Wireless Network Protocols*, Università degli Studi di Verona, 2011-2013.
- PhD student, Dr. Eleonora Sibilio. Titolo della tesi: *Formal Methods for Wireless Systems.* Università degli Studi di Verona, 2009-2011.

2.6 Pubblicazioni scientifiche

Nelle pubblicazioni seguenti, l'ordine alfabetico degli autori sottintende un contributo paritetico degli stessi. Al contrario, quando l'ordine alfabetico non è rispettato, il primo autore ha fornito un contributo prevalente. Tutte le pubblicazioni che appaiono di seguito hanno superato il vaglio di un comitato di revisori anonimi.

Riviste internazionali

[1] J. Xiang, R. Lanotte, S. Tini, S. Chong, M. Merro. Measuring Robustness in Cyber-Physical Systems under Sensor Attacks. *Nonlinear Analysis: Hybrid Systems* vol 56:101559:1-27, 2025.

[2] V. Cozza, M. Dalla Preda, R. Lanotte, M. Lucchese, M. Merro, N. Zannone. Obfuscation Strategies for Industrial Control Systems. *International Journal of Critical Infrastructure Protection* vol 47:100717:1-16, 2024.

[3] R. Lanotte, M. Merro, and A. Munteanu Industrial Control Systems Security via Runtime Enforcement. *ACM Transactions on Privacy and Security*, vol 26(1): 4:1-4:41, 2023.

[4] R. Lanotte, M. Merro and A. Munteanu. A process calculus approach to detection and mitigation of PLC malware. *Theoretical Computer Science*, vol. 890, pp. 125-146, 2021.

[5] M. Balliu, M. Merro and M. Pasqua, M. Shcherbakov. Friendly Fire: Cross-App Interactions in IoT Platforms. *ACM Transactions on Privacy and Security*, 24(3):16:1-16:40, 2021.

[6] R. Lanotte, M. Merro and S. Tini. A Probabilistic Calculus of Cyber-Physical Systems. *Information and Computation*, vol. 279, n. 104618, pp. 1-35, 2021.

[7] R. Lanotte, M. Merro and A. Munteanu, L. Viganò. A Formal Approach to Physics-based Attacks in Cyber-physical Systems. *ACM Transactions on Privacy and Security*, 23(1):3:1-3:41, 2020.

[8] R. Lanotte, M. Merro and S. Tini. Equational Reasonings in Wireless Network Gossip Protocols. *Logical Methods in Computer Science*, 14(3), 1-47, 2018.

[9] R. Lanotte and M. Merro. A Semantic Theory of the Internet of Things, *Information and Computation*, 259(1):72-101, 2018.

[10] A. Cerone, M. Hennessy and M. Merro. Modelling MAC-Layer Communications in Wireless Systems. *Logical Methods in Computer Science*, 11(1), paper 18, 1–59, 2015.

[11] D. Macedonio and M. Merro. A semantic analysis of key management protocols for wireless sensor networks. *Science of Computer Programming*, 81:53-78, 2014.

[12] M. Merro and E. Sibilio. A Calculus of Trustworthy Ad Hoc Networks. *Formal Aspects of Computing* 25(5):801-832, 2013.

[13] M. Merro, F. Ballardin and E. Sibilio. A Timed Calculus for Wireless Systems. *Theoretical Computer Science* 412(47):6585-6611, 2011.

[14] M. Merro. An Observational Theory of the CPS-calculus. *Acta Informatica* 47(2):111-132, 2010.

[15] M. Merro. An Observational Theory for Mobile Ad Hoc Networks (full paper). *Information and Computation* 207(2):194-208, 2009.

[16] R. Fuzzati, M. Merro and U. Nestmann. Distributed Consensus, Revisited. *Acta Informatica* 44(26):377-425, 2007.

[17] M. Merro and M. Hennessy. A Bisimulation-based Semantic Theory of Safe Ambients. *ACM Transactions on Programming Languages and Systems* 28(2):290-330, 2006.

[18] M. Merro and F. Zappa Nardelli. Behavioural Theory for Mobile Ambients. *Journal of the ACM* 52(6):961-1023, 2005.

[19] M. Bugliesi, S. Crafa, M. Merro and V. Sassone. Communication and Mobility Control in Boxed Ambients. *Information and Computation* 202(1):39-86, 2005.

[20] M. Hennessy, M. Merro and J. Rathke. Towards a behavioural theory of access and mobility control in distributed systems. *Theoretical Computer Science* 322(3):615-669, 2004.

[21] M. Merro and D. Sangiorgi. On asynchrony in name-passing calculi. *Mathematical Structures in Computer Science* 14(5):715-767, 2004.

[22] M. Merro, J. Kleist and U. Nestmann. Mobile Objects as Mobile Processes. *Information and Computation* 177(2):195-241, 2002.

[23] U. Nestmann, H. Hüttel, J. Kleist and M. Merro. Aliasing Models for Mobile Objects. *Information and Computation* 175(1):3-33, 2002.

Atti di convegni internazionali

[24] F. Lupia, M. Lucchese, M. Merro and N. Zannone. ICS Honeypot Interactions: A Latitudinal Study In *2023 IEEE International Conference on Big Data (IEEE BigData 2023)*, IEE, pp. 1-10, 2023.

[25] V. Cozza, M. Dalla Preda, M. Lucchese, M. Merro and N. Zannone. Towards Obfuscation of Programmable Logic Controllers. In *18th International Conference on Availability, Reliability and Security (ARES 2023)*, ACM, pp. 121:1-121:10, 2023.

[26] M. Lucchese, F. Lupia, M. Merro, F. Paci, N. Zannone and A. Furfaro. HoneyICS: A High-interaction Physics-aware Honeynet for Industrial Control Systems. In *18th International Conference on Availability, Reliability and Security (ARES 2023)*, ACM, pp. 113:1-113:10, 2023.

[27] R. Lanotte, M. Merro and N. Zannone. Impact Analysis of Coordinated Cyber-Physical Attacks via Statistical Model Checking: A Case Study. In *43rd IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2023)*. Volume 13910 of Lecture Notes in Computer Science, pp. 75-95, Springer, 2023.

[28] S. Chong, R. Lanotte, M. Merro, S. Tini and J. Xiang. Quantitative Robustness Analysis of Sensor Attacks on Cyber-Physical Systems. In *26th ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2023)*, ACM, pp. 20:1-20:12, 2023.

[29] M. Lucchese, M. Merro, F. Paci and N. Zannone. Towards A High-interaction Physics-aware Honeynet for Industrial Control Systems. In *38th ACM/SIGAPP Symposium on Applied Computing (SAC 2022)*, pp. 76-79, ACM, 2023.

[30] M. Ceccato, Y. Driouich, R. Lanotte, M. Lucchese and M. Merro. Towards Reverse Engineering of Industrial Physical Processes. In *3rd Cyber-Physical Security for Critical Infrastructures Protection (CPS4CIP 2022)*, Volume 13785 of Lecture Notes in Computer Science, pp. 273-290, Springer, 2023.

[31] R. Lanotte, M. Merro, A. Munteanu, and S. Tini. Formal Impact Metrics for Cyber-physical Attakcs. In *34th IEEE Computer Security Foundations Symposium (CSF'21)*. IEEE Computer Society, pp. 1-16, 2021.

[32] R. Lanotte, M. Merro, and A. Munteanu. A Process Calculus Approach to Correctness Enforcement of PLCs. In *21st Italian Conference on Theoretical Computer Science (ICTCS'20)*. CEUR Workshop Proceedings, pp. 81-94, 2020.

[33] R. Lanotte, M. Merro, and A. Munteanu. Runtime Enforcemente for Control System Security. In *33th IEEE Computer Security Foundations Symposium (CSF'20)*. IEEE Computer Society, pp. 246-261, 2020.

[34] A. Munteanu, M. Pasqua and M. Merro, Impact Analysis of Cyber-Physical Attacks on a Water Tank System via Statistical Model Checking. In *8th ACM/IEEE International Conference on Formal Methods in Software Engineering (FormaliSE'20)*. ACM, pp. 34-43, 2020.

[35] M. Balliu, M. Merro, and M. Pasqua. Securing Cross-App Interactions in IoT Platforms. In *32th IEEE Computer Security Foundations Symposium (CSF'19)*. IEEE Computer Society, pp. 319-334, 2019.

[36] R. Lanotte, M. Merro and F. Mogavero. On the Decidability of Linear Bounded Periodic Cyber-Physical Systems. In *22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC'19)*, ACM Press, pp. 87-98, 2019.

[37] R. Lanotte, M. Merro and S. Tini. Towards a formal notion of impact metric for cyber-physical attacks. In *14th International Conference on integrated Formal Methods (iFM 2018)*, Volume 11023 of Lecture Notes in Computer Science, pp. 296-315, Springer, 2018.

[38] A. Munteanu, R. Muradore, M. Merro and P. Fiorini. On cyber-physical attacks in bilateral teleoperation systems: An experimental analysis. In *1st IEEE International Conference on Industrial Cyber-Physical Systems (ICPS-2018)*, pp. 159-166, IEEE Industrial Electronics Society, 2018.

[39] R. Lanotte, M. Merro and A. Munteanu. A Modest Security Analysis of Cyber-Physical Systems: A Case Study. In *38th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE'18)*. Volume 10854 of Lecture Notes in Computer Science, pp. 58-78, Springer, 2018.

[40] M. Kamali, M. Merro and A. Dal Corso. AODVv2: performance vs. loop freedom. In *44th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'18)*. Volume 10706 of Lecture Notes in Computer Science, pp. 337-350, Springer, 2018.

[41] R. Lanotte, M. Merro and S. Tini. Compositional weak metrics for group key update. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS'17)*. Volume 83 of LIPIcs series, pp. 72:1-27:16, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[42] R. Lanotte, M. Merro, R. Muradore and L. Viganò. A Formal Approach to Cyber-physical Attacks. In *30th IEEE Computer Security Foundations Symposium (CSF'17)*. IEEE Computer Society, pp. 436-450, 2017.

[43] R. Lanotte, M. Merro and S. Tini. Weak simulation quasimetric in a gossip scenario. In *37th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE'17)*. Volume 10321 of Lecture Notes in Computer Science, pp. 139-155, Springer, 2017.

[44] R. Lanotte and M. Merro. A Calculus of Cyber-Physical Systems. In *11th International Conference on Language and Automata Theory and Applications (LATA'17)*. Volume 10168 of Lecture Notes in Computer Science, pp. 115-127, Springer, 2017.

[45] M. D. Ernst, D. Macedonio, M. Merro and F. Spoto. Semantics for Locking Specifications. In *8th NASA Formal Methods Symposium (NFM'16)*. Volume 9690 of Lecture Notes in Computer Science, pp. 355-372, Springer, 2016.

[46] R. Lanotte and M. Merro. A Semantic Theory of the Internet of Things (extended abstract). In *18th IFIP International Conference on Coordination Models and Language (COORDINATION'16)*. Volume 9686 of Lecture Notes in Computer Science, pp. 157-174, Springer, 2016.

[47] A. Dal Corso, D. Macedonio and M. Merro. Statistical Model Checking of Ad Hoc Routing Protocols in Lossy Grid Networks. In *7th NASA Formal Methods Symposium (NFM'15)*. Volume 9058 of Lecture Notes in Computer Science, pp. 112-126, Springer, 2015.

[48] A. Cerone, M. Hennessy and M. Merro. Modelling MAC-Layer Communications in Wireless Systems. In *15th IFIP International Conference on Coordination Models and Language (COORDINATION'13)*. Volume 7890 of Lecture Notes in Computer Science, pp. 16-30, Springer, 2013.

[49] L. Battisti, D. Macedonio and M. Merro. Statistical Model Checking of a Clock Synchronization Protocol for Sensor Networks. In *5th IPM International Conference on Fundamentals of Software Engineering (FSEN'13)*. Volume 8161 of Lecture Notes in Computer Science, pp. 168-182, Springer, 2013.

[50] D. Macedonio and M. Merro. A Semantic Analysis of Wireless Network Security Protocols. In *4th NASA Formal Methods Symposium (NFM'12)*. Volume 7226 of Lecture Notes in Computer Science, pp. 403-417, Springer, 2012.

[51] R. Lanotte and M. Merro. Semantic Analysis of Gossip Protocols for Wireless Sensor Networks. In *22nd International Conference on Concurrency Theory (CONCUR'11)*. Volume 6901 of Lecture Notes in Computer Science, pp. 156-170, Springer, 2011.

[52] F. Ballardin and M. Merro. A Calculus for the Analysis of Wireless Network Security Protocols. In *7th Workshop on Formal Aspects in Security and Trust (FAST'10)*. Volume 6561 of Lecture Notes in Computer Science, pp. 206-222, Springer, 2011.

[53] D. Benetti, M. Merro and L. Viganò. Model Checking Ad Hoc Network Routing Protocols: ARAN vs. endairA. In *8th IEEE Conference on Software Engineering and Formal Methods (SEFM'10)*, IEEE Computer Society Press, pp. 191-202, 2010.

[54] M. Merro and E. Sibilio. A Calculus of Trustworthy Ad Hoc Networks. In *6th International Workshop on Formal Aspects in Security and Trust (FAST'09)*. Volume 5983 of Lecture Notes in Computer Science, pp. 157-172, Springer, 2010.

[55] M. Merro and E. Sibilio. A Timed Calculus for Wireless Systems. In *3rd International Conference on Fundamentals on Software Engineering (FSEN'09)*. Volume 5961 of Lecture Notes in Computer Science, pp. 228-243, Springer, 2010.

[56] M. Merro. An Observationl Theory for Mobile Ad Hoc Networks. In *23rd International Conference on the Mathematical Foundations of Program Semantics (MFPS'07)*. Electronic Notes in Theoretical Computer Science 173:275-293, Elsevier, 2007.

[57] M. Merro and C. Biasi. On the observational theory of the CPS-calculus. In *22nd International Conference on the Mathematical Foundations of Program Semantics (MFPS'06)*. Electronic Notes in Theoretical Computer Science 158:307-330, Elsevier, 2006.

[58] M. Merro and F. Zappa Nardelli. Behavioural Theory for Mobile Ambients. In *3rd International Conference on Theoretical Computer Science (IFIP TCS 2004)*, pp. 549-562, Kluwer, 2004.

[59] U. Nestmann, R. Fuzzati and M. Merro. Modeling consensus in a process calculus. In *14th International Conference on Concurrency Theory (CONCUR'03)*. Volume 2761 of Lecture Notes in Computer Science, pp. 399-414, Springer, 2003.

[60] M. Merro and F. Zappa Nardelli. Bisimulation proof techniques for mobile ambients. In *30th International Colloquium on Automata, Languages, and Programming (ICALP'03)*. Volume 2719 of Lecture Notes in Computer Science, pp. 584-598, Springer, 2003.

[61] M. Hennessy, M. Merro and J. Rathke. Towards a Behavioural Theory of Access and Mobility Control in Distributed System. In *6th International Conference on the Foundations of Software Science and Computation Structures (FOSSACS'03)*. Volume 2620 of Lecture Notes in Computer Science, pp. 282-297, Springer, 2003.

[62] M. Bugliesi, S. Crafa, M. Merro and V. Sassone. Communication Interference in Mobile Boxed Ambients. In *22th International Conference on the Foundations of Software Technology and Theoretical Computer Science (FST&TCS'02)*. Volume 2556 dof Lecture Notes in Computer Science, pp. 71-84, Springer, 2002.

[63] M. Merro and V. Sassone. Typing and Subtyping Mobility in Boxed Ambients. In *13th International Conference on Concurrency Theory (CONCUR'02)*. Volume 2421 of Lecture Notes in Computer Science, pp. 304-320, Springer, 2002.

[64] M. Merro and M. Hennessy. Bisimulation Congruences in Safe Ambients. Conference record of *29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'02)*. Volume 37(1), pp. 71-80, ACM Press, 2002.

[65] M. Merro, J. Kleist and U. Nestmann. Local π-Calculus at work: Mobile Objects as Mobile Processes. In *1st IFIP International Conference on Theoretical Computer Science (IFIP TCS'00)*. Volume 1872 of Lecture Notes in Computer Science, pp. 390-408, Springer, 2000.

[66] M. Merro. Locality and Polyadicity in Asynchronous Name-passing Calculi. In *3rd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'00)*. Volume 1784 of Lecture Notes in Computer Science, pp. 238-251, Springer, 2000.

[67] H. Hüttel, J. Kleist, M. Merro e U. Nestmann. Aliasing Models for Object Migration. In *5th International EURO-PAR'99 - PARALLEL PROCESSING Conference*. Volume 1685 of Lecture Notes in Computer Science, pp. 1353-1368, Springer, 2000.

[68] M. Merro. On Equators in Asynchronous Name-passing Calculi without Matching. In *6th International Workshop on Expressiveness in Concurrency (EXPRESS'99)*. Electronic Notes in Theoretical Computer Science 27:57-70, Elsevier, 1999.

[69] M. Merro. On the Expressiveness of Chi, Update, and Fusion calculi. In *5th International Workshop on Expressiveness in Concurrency (EXPRESS'98)*. Electronic Notes in Theoretical Computer Science 16(2):133-144, Elsevier, 1998.

[70] M. Merro and D. Sangiorgi. On Asynchrony in Name-passing Calculi. In *25th International Colloquium on Automata, Languages and Programming (ICALP'98)*. Volume 1443 of Lecture Notes in Computer Science, pp. 856-867, Springer, 1998.

[71] A. Maggiolo-Schettini e M. Merro. Priorities in Statecharts. In *5th Workshop on Analysis and Verification of Multiple-Agent Languages (LOMAPS'96)*. Volume 1192 of Lecture Notes in Computer Science, pp. 404-429, Springer, 1996.

Tesi di dottorato

[72] M. Merro. Locality in the pi-calculus and applications to distributed objects. *École Nationale Supérieure des Mines de Paris*. October 2000.

2.7 Inquadramento dell'attività di ricerca

Sistemi concorrenti. I sistemi concorrenti sono costituiti da un insieme di attività indipendenti – i processi – che tipicamente interagiscono tra loro. I *calcoli di processo* (talvolta chiamati *algebre di processo*) sono tra i formalismi più utilizzati per descrivere formalmente i sistemi concorrenti. Tra questi, il π*-calcolo* di Milner, Parrow e Walker (1999) ha assunto nell'ultimo ventennio una posizione dominante. Come il λ-calcolo, il π-calcolo è costituito da un numero esiguo di operatori: input e output su canali, composizione parallela, ricorsione, e creazione di canali privati.

Una nozione cruciale in tutti i calcoli di processo è quella di *semantica comportamentale*. È impensabile, infatti, sperare di capire il comportamento di un processo se non si è in grado di determinare quando due processi hanno il medesimo comportamento. Da un punto di vista osservazionale, due processi hanno lo stesso comportamento se un osservatore esterno non è in grado di notarne la differenza interagendo con loro. Grazie all'uso di sistemi di transizioni etichettate, attraverso cui codificare l'insieme dei contesti di esecuzione possibili, la *bisimulazione* (Park 1990) rappresenta una tra le uguaglianze comportamentali di maggior successo in ambito concorrente. Comunque, dopo la definizione di Park, vi è stata una proliferazione di nozioni diverse di bisimulazione, non sempre "giustificate", o pienamente "appropriate", in ambito concorrente. Nel seguito per "appropriatezza" di una bisimulazione intenderemo una condizione ben precisa: la bisimulazione deve caratterizzare una qualche nozione naturale di uguaglianza comportamentale contestuale, come ad esempio la *barbed congruence* (Milner e Sangiorgi 1992). Si segnala che caratterizzazioni di equivalenze contestuali in termini di bisimulazioni etichettate sono rare e notoriamente difficili.

Linguaggi per sistemi distribuiti e mobili. Negli ultimi vent'anni vi è stato un interesse crescente per i sistemi distribuiti in grado di supportare *mobilità di codice*. Cioè sistemi concorrenti, dislocati su più locazioni virtuali e/o fisiche, in grado di far migrare del codice da una locazione ad un'altra. Una forma diversa di mobilità è la *mobilità di agenti*, in cui agenti in esecuzione migrano da un sito all'altro di una rete di calcolatori alla ricerca di risorse ed informazioni.

In tale contesto si inseriscono i linguaggi di programmazione ad oggetto in ambiente distribuito, in cui gli oggetti sono liberi di migrare da una locazione ad un'altra per fornire servizi più efficienti. Negli ultimi quindici anni è stata effettuata una intensa attività di ricerca rivolta allo studio della teoria dei linguaggi di programmazione ad oggetto. Vari modelli (spesso ortogonali tra loro) sono stati proposti. La comunità dei linguaggi di programmazione ad oggetto è stata così tra le prime a cimentarsi con la nozione di mobilità, proponendo linguaggi come *Emerald* (Jul et al. 1988), *Obliq* (Cardelli 1995) fino ad arrivare a *Java* (Sun Inc. 1995).

Con la nascita dei primi linguaggi per sistemi mobili, sono sorti vari calcoli di processo per studiare formalmente i sistemi distribuiti con mobilità. Tra questi, quelli di maggior successo sono stati: $D\pi$ (Hennessy e Riely 1998), un'estensione distribuita del π-calcolo con mobilità di codice, *Mobile Ambients* (Cardelli e Gordon 1998), disegnato attorno al concetto di locazione e mobilità di agenti, *Boxed Ambients* (Bugliesi, Castagna e Crafa 2001) e Safe Ambients (Levi e Sangiorgi 1999), due varianti di Mobile Ambients che migliorano la teoria comportamentale e le proprietà di buon comportamento del calcolo originale. Tutti questi calcoli vengono detti *higher-order* poiché consentono la trasmissione/migrazione di processi. Sin dalla loro apparizione, uno dei principali problemi dei calcoli higher-order con mobilità è stato quello di definire un'appropriata nozione di equivalenza comportamentale che consentisse di verificare proprietà di rilievo. In quegli anni, la bisimulazione sembrava il candidato più promettente. Comunque, la definizione di una nozione *appropriata* di bisimulazione per calcoli higher-order con mobilità si è rivelato un problema particolarmente difficile per diversi anni.

Reti mobili ad hoc e verifica formale di protocolli di rete. La diffusione di reti wireless ad hoc con dispositivi mobili ha introdotto numerose sfide tecniche nella definizione di una semantiche formali in grado di catturarne il comportamento. Questo perché le reti ad hoc introducono nuovi aspetti di dinamicità, come ad e-sempio: la mobilità dei dispositivi, l'assenza di un nodo centrale con mansioni di gestione, l'interferenza sui canali radio dovuta alla presenza di agenti esterni, oppure la possibilità di *collisioni* a livello di comunicazione.

Un'altra sfida importante nell'ambito dei sistemi wireless è quella riguardante la *sicurezza*, visto che i dispositivi wireless sono notoriamente più esposti ad attacchi di vario tipo. Infatti, un agente malevolo può attaccare sia il dispositivo (ad esempio un sensore di una rete di sensori) oppure il canale di comunicazione (di più facile accesso, rispetto ad una rete cablata) al fine di compromettere il comportamento della rete e dei suoi protocolli. Si noti che, in generale, i dispositivi wireless hanno una bassa capacità computazionale, così la sfida nella protezione di reti wirelss consiste nel garantire un buon livello di sicurezza utilizzando risorse limitate.

Per quanto riguarda la verifica dei protocolli per reti ad hoc, la maggior parte delle analisi di protocolli per reti wireless fanno uso di simulatori a eventi discreti (ad esempio, ns-2, Opnet, Glomosin). Comunque, simulatori differenti spesso supportano modelli differenti a livello MAC, dando luogo a risultati diversi. D'altro canto, tecniche di analisi formale, come ad esempio il *model checking*, consentono un'analisi esaustiva dei protocolli, e sono in grado di fornire controesempi quando determinate proprietà di correttezza non sono verificate. Ovviamente, in generale, le tecniche di model-checking soffrono del ben noto problema di "esplosione degli stati", laddove le tecniche di simulazione consentono di trattare sistemi di dimensione medio-grande al prezzo di rinunciare all'esaustività dell'analisi e all'accuratezza numerica.

Internet of Things e sistemi ciberfisici. La diffusione sempre crescente di dispositivi portatili in grado di usare diverse forme di comunicazione wireless ha rappresentato una delle ragioni chiave del successo del cosiddetto *Mobile Internet*. In questo contesto si è velocemente diffuso un nuovo paradigma noto col nome di *Internet of Things* (IoT). In un tipico scenario IoT, dispositivi eterogenei tra loro (ad e-sempio, sensori, smartphone, attuatori, etc) possono comunicare attraverso canali wireless a corto raggio (ad esempio, Zigbee, WiFi, NFC, etc), condividere dati (ad esempio, rilevazioni fatte attraverso sensori), condividere risorse (ad esempio, accesso Internet, attuatori, smartphone) e supportare applicazioni distribuite attraverso l'aggregazione di servizi messi a disposizione dall'environment.

La gamma dei domini di applicazione di IoT è in continua crescita. Si va dal controllo ambientale alle applicazioni mediche, dalla sorveglianza alla domotica, fino ad arrivare all'automotive e all'avionics. La ricerca su IoT si sta al momento focalizzando su tecnologie per il semantic web, architetture di rete di nuova generazione, cloud computing, e sempre di più sulla sicurezza dei dispositivi (è notizia di queste settimane la botnet Mirai che ha portato a termine un gigantesco DDoS usando migliaia di dispositivi IoT). Comunque, per adesso, in ambito IoT si è fatto poco uso di tecniche formali per modellare l'interazione tra le varie componenti, e per verificare la correttezza di sistemi IoT prima della fase implementativa. Un'immediata applicazione di queste tecniche sarebbe lo sviluppo di strumenti di model-checking in grado di confutare le implementazioni rispetto alle specifiche. Inoltre, una rappresentazione formale di questi sistemi sarebbe molto utile anche ai fini di un'analisi della loro sicurezza.

Un discorso analogo a quanto detto per i sistemi in ambito IoT può essere fatto per i sistemi ciberfisci, che hanno molto in comune con i sistemi IoT, sebbene nei sistemi ciberfisici il processo fisico è in genere molto più complesso e di conseguenza può essere motivo di forti attenzioni, pubbliche e private (si pensi all'attacco Stuxnet che ha distrutto le centrifughe iraniane per l'arricchimento dell'uranio, a seguito di un complesso attacco ciberfisico lanciato da remoto).

2.8 Contributo scientifico

Modelli per linguaggi concorrenti. Gli articoli [70, 66, 21, 72] introducono e studiano *Local* π, una variante del π-calcolo particolarmente adatta per lo studio formale di proprietà di buon comportamento per linguaggi di programmazione concorrenti e/o distribuiti, come Pict (Pierce and Turner 2000), Join (Fournet and Gonthier 1996), Blue (Boudol 1997) e Obliq (Cardelli 1995).

L'ingrediente principale di Local π è una semplice disciplina di tipi secondo cui i canali ricevuti da un processo possono essere utilizzati solo in operazioni di output. Negli articoli [70, 21] vengono studiate la teoria algebrica e comportamentale di Local π, prestando particolarmente attenzione alla nozione di bisimulazione, che per un sistema di transizioni etichettato opportunamente definito, viene mostrata essere una caratterizzazione della *barbed congruence*.

L'articolo [68] studia la teoria comportamentale del π-calcolo asincrono e si focalizza sull'uso di processi speciali chiamati *equators*, e inizialmente introdotti da Honda e Tokoro (1991). L'articolo [69] studia la relazione tra gli equators e la *fusione* di nomi di canali, nello stile del Fusion calculus (Parrow e Victor 1998). Nel loro lavoro, Parrow e Victor argomentarono che il Fusion calculus fosse strettamente più espressivo del π-calcolo. L'articolo [69] mostra invece che è possibile definire un encoding fully abstract di una variante asincrona del Fusion calculus in una variante asincrona del π-calcolo.

Semantiche per linguaggi ad oggetti distribuiti. Negli articoli [67, 65, 22, 23] viene affrontato e risolto un problema aperto riguardante *Obliq*, un linguaggio ad oggetti in ambiente distribuito di particolare successo. In tale linguaggio, la migrazione di oggetti è implementata in termini di *clonazione* e *aliasing*. In linguaggi ad oggetti, un oggetto A è un alias di un oggetto B quando le richieste indirizzate ad A vengono inoltrate a B. Il problema aperto consisteva nel capire se la migrazione di oggetti, così implementata, fosse trasparente nei confronti dei clienti dell'oggetto stesso, e come ciò potesse essere provato formalmente.

Seguendo la semantica originale (ma informale) di Obliq, gli articoli [67, 23] forniscono una semantica formale per un'appropriata astrazione di Obliq. Questo ha consentito di definire una nozione standard di equivalenza comportamentale del tipo "may convergence". La correttezza della migrazione è stata formalizzata attraverso una semplice equazione comportamentale che mette a confronto il comportamento di un oggetto prima e dopo la migrazione. Nel tentativo di provare questa equazione, sono venuti alla luce parecchi controesempi che hanno provato la non correttezza della migrazione rispetto alla semantica originale.

Gli articoli [65, 22] forniscono una semantica formale per Obliq attraverso una traduzione in Local π. Rispetto alla semantica originale sono state introdotte opportune correzioni. In questo nuova semantica viene allora fornita una prova formale della trasparenza della migrazione di oggetti, combinando tecniche di prova non banali facenti parte della teoria semantica di Local π.

Strumenti semantici per la verifica formale di linguaggi concorrenti. I linguaggi moderni di programmazione implementano la concorrenza attraverso il *multithreading*, che si traduce in *true parallelism* quando si ha la disponibilità di hardware multicore. Risulta quindi essenziale poter scrivere software multithreaded corretto al fine di usare appieno le potezialità dell'hardware presente e futuro. Questa è anche la scelta più naturale adottata per servizi web, cloud computing, avionics e automotive. Comunque, la sincronizzazione di programmi multithreaded ha un costo: la possibilità di incorrere in *data race* che possono dar luogo a errori subdoli e spesso non ripetibili. Per evitare data race, i programmatori che fanno uso di linguaggi concorrenti adottano delle *discipline di lock*. Annotazioni per la specifica di tali discipline, come ad esempio l'annotazione `@GuardedBy` di Java, sono ampiamente utilizzate dai programmatori di linguaggi concorrenti.

L'articolo [45] fornisce la prima formalizzazione di due possibili semantiche dell'annotazione `@GuardedBy`. Questo articolo individua in maniera formale quando l'annotazione `@GuardedBy` garantisce l'assenza di data race. I risultati apparsi in [45] sono stati usati per estendere l'analizzatore statico Julia (Spoto 2016) in grado di verificare e inferire annotazioni `@GuardedBy` su codice Java arbitrario. Questo lavoro rappresenta il nucleo del Joint Project dal titolo "Static Analysis for Multithreading", di cui il Dr. Merro è il "principal investigator".

Semantiche per linguaggi distribuiti e mobili. Gli articoli [61, 20] definiscono una teoria comportamentale per il linguaggio con *mobilità di codice* Dπ, una variante distribuita del π-calcolo. Il principale contributo di questi due articoli è la corretta definizione di una *bisimulazione tipata* per Dπ. La correttezza della definizione di bisimulazione viene dimostrata provando un risultato di "full abstraction" tra la bisimulazione tipata e la barbed congruence (tipata) di Dπ. La prova presenta diverse difficoltà tecniche. Tale risultato è poi esteso ad uno scenario più realistico, in cui la migrazione di codice è soggetta ad autorizzazione esplicita, annotando i tipi delle locazioni con informazioni sulla mobilità del codice allocato.

I lavori [64, 17] sono i primi in letteratura che definiscono una caratterizzazione coinduttiva di ordine superiore della barbed congruence in un linguaggio con *mobilità di agenti*. In questi articoli viene introdotto un sistema di transizioni etichettato e una teoria di bisimulazioni per una estensione di Safe Ambients (Levi and Sangiorgi 1999) arricchito con passwords per controllare la mobilità di agenti. Il risultato principale consiste nella caratterizzazione della barbed congruence in termini di una bisimulazione etichettata definita sull'LTS appositamente definito. In tal modo è possibile provare un insieme di leggi algebriche che caratterizzano il linguaggio. Il lavoro svolto in [64, 17] ha rappresentato il primo passo per capire come affrontare il medesimo problema nel caso più generale dei Mobile Ambients.

Negli articoli [60, 58, 18] viene affrontato e risolto un problema aperto riguardante *Mobile Ambients*. Il problema consisteva nel definire un'adeguata semantica comportamentale che consentisse di confrontare sistemi sintatticamente diversi ma semanticamente equivalenti. L'articolo propone una definizione di bisimulazione *di ordine superiore* in grado di caratterizzare completamente la barbed congruence per Mobile Ambients. Il passaggio all'ordine superiore si rivela indispensabile per modellare computazioni in grado di migrare tra locazioni diverse. La prova di "full abstraction" affronta una serie di ostacoli tecnici dovuti a due peculiarità dei Mobile Ambients: (i) l'asincronia della mobilità, che rende difficile osservare la mobilità degli agenti; (ii) la possibilità fornita dal calcolo di "aprire" ambienti, rendendoli così accessibili al codice esterno senza alcun controllo.

Boxed Ambients è una variante dei Mobile Ambient in cui l'operatore (potenzialmente pericoloso) per l'apertura di ambienti a codice esterno, viene rimpiazzato da una forma di comunicazione tra processi in grado di attraversare gli ambienti. L'idea è quella di utilizzare tecniche standard di tipaggio dei canali di comunicazione per filtrare l'informazione che attraversa la frontiera di un dominio amministrativo. Questa modifica del linguaggio comporta un prezzo in termini di interferenze nelle comunicazioni, con effetti nefasti sulla teoria algebrica, e quindi sull'utilizzabilità del calcolo. Gli articoli [63, 62, 19] adottano le nozioni di co-azione e password, introdotte in Safe Ambients con password, e propongono nuove nozioni di tipaggio per elaborare una caratterizzazione etichettata della barbed congruence tipata che consenta di provare una teoria algebrica molto più ricca di quella esistente per Mobile Ambients e Safe Ambients.

Fondamenti semantici per sistemi wireless. Negli articoli [56, 15] viene proposto il primo calcolo per reti mobili ad hoc. Alcune peculiarità delle reti ad hoc, quali la comunicazione broadcast con portata limitata, la mobilità dei nodi, e la natura half-duplex dei canali, rende lo studio di queste reti particolarmente intessante. L'articolo definisce una teoria semantica fully abstract per le reti mobili ad hoc. Tale teoria viene successivamente utilizzata per provare l'equivalenza comportamentale di un certo numero di sistemi wireless reali quali, ad esempio, i *range repeaters*, ripetitori di segnali, usati per estendere il segnale radio.

Gli articoli [55, 13] introducono per la prima volta una nozione non banale di tempo (discreto) in un calcolo per reti wireless ad hoc. In particolare, il calcolo consente la rappresentazione di comunicazioni con una durata temporale, similmente a quanto avviene nei sistemi reali. Ciò ha consentito lo studio formale di uno dei problemi fondamentali delle reti wireless: le *collisioni* durante le comunicazioni. Una collisione sulla comunicazione si verifica quando un nodo in ricezione viene esposto a più trasmissioni in

contemporanea da parte di più stazioni. In tal caso, il nodo in ricezione non è in grado di decifrare le informazioni ricevute che dovranno quindi essere ritrasmesse. Le collisioni durante le comunicazioni sono quindi la ragione principale delle basse prestazioni, in termini di banda, delle reti wireless rispetto a quelle cablate. L'articolo propone una nozione di bisimulazione che consente di ragionare in maniera composizionale anche su sistemi di grandi dimensioni, e su protocolli di rete non banali, come il CSMA/CA, per evitare collisioni in reti wireless. Il lavoro svolto in questi due articoli rappresenta la base di partenza per lo sviluppo di logiche e tool di verifica probabilistici che consentano di stabilire la presenza di collisioni in un un sistema wireless.

Gli articoli [48, 10] estendono e generalizzano i risultati ottenuti in [55, 13]. In particolare, viene proposto un calcolo di processi più semplice in cui è possibile ridefinire il concetto di azione osservabile, tenendo conto delle interferenze dovute alla presenza di collisioni. A differenza di altri calcoli di processi basati sulla comunicazione su canali, per via delle collisioni, l'azione osservabile non è più la trasmissione di un messaggio ma la sua eventuale ricezione con successo. Inoltre, in maniera piuttosto sorprendente, la nozione di osservabilità non richiede la registrazione del passaggio del tempo. In qualche modo, questa capacità di osservazione la si ottiene a costo zero. L'articolo propone quindi un sistema di transizioni etichettato estensionale che viene usato per definire una bisimulazione in grado caratterizzare completamente la barbed congruence nel linguaggio proposto. Per via della definizione non-standard di osservabilità (dovuto alla presenza delle collisioni), la prova di full abstraction presenta diverse difficoltà tecniche.[1]

Verifica formale di protocolli per reti wireless ad hoc. Come già detto in precedenza, le reti mobili ad hoc, ed i protocolli per esse, nascondono una serie di sfide tecniche.

Ad esempio, molti protocolli per reti wireless ad hoc adottano una nozione di tempo condivisa tra dispositivi. Questa nozione viene implementata attraverso i cosiddetti *protocolli di sincronizzazione dei clock*. La verifica della correttezza di tali protocolli è rimasta inesplorata per molti anni, sebbene la compromissione di questi protocolli rappresenti una vulnerabilità spesso usata per impedire la corretta sincronizzazione dei dispositivi. L'articolo [49] applica tecniche di *model-checking statistico* (SMC Uppaal) per verificare un protocollo di sincronizzazione di ultima generazione, sviluppato all'interno del progetto Europeo QUASIMODO. Il risultato principale dell'articolo è che il protocollo in questione non è in grado di sincronizzare correttamente i dispositivi di una rete wireless in presenza di perdita di pacchetti (anche relativamente esigua). La perdita di pacchetti rappresenta la normalità in sistemi wireless reali.

I *protocolli di gossip* si collocano ad un livello intermedio nella gerarchia dei protocolli di rete, e si prendono cura di inserire nuovi messaggi, inoltrare i messaggi correnti e cancellare i vecchi messaggi. Gli articoli [51, 43, 8] definiscono un semplice calcolo di processi temporizzato probabilistico munito di una teoria comportamentale probabilistica per confrontare protocolli di gossip diversi. Questa teoria comportamentale è poi usata per provare un numero di leggi algebriche molto efficaci per stimare le performance di reti gossip, con e senza perdita di messaggi, in termini della probabilità di propagare messaggi con successo.

Infine, le reti mobili ad hoc si basano su comunicazioni node-by-node in cui i nodi giocano essenzialmente due ruoli: (i) agiscono come destinatari dei messaggi; (ii) eseguono funzionalità di routing per instradare pacchetti destinati ad altri nodi. I protocolli di routing sono così fondamentali per determinare i cammini migliori per instradare il flusso dei dati tra due nodi della rete. Nell'articolo [47] vengono usate tecniche di model-checking statistico (SMC Uppaal) per confrontare il protocollo di routing più diffuso per reti ad hoc, AODV, e il suo ammodernamento DYMO (sviluppato 15 anni dopo), in termini di informazioni valide immagazzinate nelle tabelle di routing. Il risultato principale è che, in contrasto con i risultati recenti apparsi in un articolo di Höfner e McIver, il protocollo DYMO si comporta meglio di AODV su reti di dimensioni medio-grandi, e in presenza di perdita di messaggi. L'articolo [40] utilizza SMC Uppaal per confrontare i protocolli di routing DYMO e AODVv2 (l'ultima versione di AODV) in termini di performance e assenza di loop nei cammini generati.

Verifica formale di protocolli di sicurezza per sistemi wireless. In generale, al fine di garantire una comunicazione sicura tra due o più principal, è necessario stabilire una qualche relazione di sicurezza

[1] L'articolo [48] ha ottenuto il *DisCoTec 2013 best paper award*.

basata sulla condivisione di un segreto da parte dei principal stessi. Questo segreto, che spesso consiste in una chiave crittografica, deve essere creato, distribuito e mantenuto aggiornato. Così, la gestione delle chiavi crittografiche è alla base di qualsiasi protocollo di sicurezza. Nelle reti wireless (come ad esempio le reti di sensori) i protocolli di distribuzione delle chiavi crittografiche fanno un uso limitato delle risorse computazionali. I protocolli più comunemente usati in tale ambito sono μTESLA, LiSP, LEAP, PEBL, e INF. Gli articoli [52, 50, 11] propongono un semplice calcolo di processi con tempo discreto per specificare e verificare protocolli per la distribuzione di chiavi crittografiche come quelli sopra menzionati. Il calcolo è arricchito con una teoria comportamentale di simulazione. Tale teoria viene usata per estendere alle reti wireless un framework (tGNDC) ampiamente usato per la verifica formale di proprietà di sicurezza in ambito concorrente (Gorrieri e Martinelli 2008). Attraverso tGNDC è possibile, tra l'altro, verificare la freshness di una determinata informazione o la corretta terminazione di un protocollo crittografico, nei tempi dovuti. Gli articoli [52, 50, 11] usano tGNDC per un'analisi semantica di tre protocolli reali: μTESLA, LEAP+ e LiSP. L'analisi ha consentito di trovare due attacchi di tipo *replay*, non conosciuti in letteratura, a danno degli ultimi due protocolli. L'articolo [41] definisce metriche probabilistiche ad hoc per misurare le performance di protocolli per l'aggiornamento di chiavi di gruppo.

I protocolli di routing per reti mobili ad hoc sono anch'essi esposti a diverse tipologie di attacco, derivanti dalla natura stessa delle reti ad hoc. Così, negli ultimi 15 anni sono state sviluppate diverse versioni crittografiche di protocolli di routing in grado di operare in ambiente ostile. Comunque, in generale, provare la sicurezza di protocolli crittografici è notoriamente difficile; l'esperienza in tal senso ha mostrato che analisi informali o semi-formali non sono sufficienti. L'articolo [53] applica tecniche di model-checking specificamente progettate per protocolli crittografici (attraverso il tool AVISPA) con l'obiettivo di verificare la sicurezza di due protocolli di routing crittografici: ARAN e endairA. L'analisi ha individuato in ARAN due attacchi diversi di tipo *spoofing*, non noti in letteratura.

Col termine *trust management* si intende una tecnica per specificare e interpretare security policy, credenziali, e relazioni di trust che coinvolgano diversi principal. In scenari altamente dinamici, come quello delle reti mobili ad hoc, la trattazione formale di un meccanismo di trust management è apparso subito un problema difficile. In particolare, l'assenza di un nodo che abbia mansioni di gestione, insieme alla mobilità dei nodi, rappresentano due caratteristiche che non consentono di usare per reti ad hoc meccanismi di sicurezza tradizionali usati per le reti cablate. In particolare, la nozione di "trust" o fiducia tra i nodi, non può derivare dal controllo di un certificato attraverso un'autorità centrale, ma deve provenire dalla cooperazione tra i nodi. Gli articoli [54, 12] propongono un calcolo di processi per reti mobili ad hoc, orientato alla sicurezza, in grado di inglobare un modello di trust comportamentale multilivello. Il calcolo consente di provare formalmente la proprietà di non interferenza classica (Goguen e Meseguer 1992), assicurando che l'informazione fluisca sempre da livelli di sicurezza più bassi a livelli di sicurezza più alti. Infine, il calcolo consente di provare una proprietà altamente desiderabile per reti ad hoc: la correttezza della comunicazione nonostante la compromissione di un numero massimo di nodi.

Semantica formale e sicurezza per sistemi IoT. Gli articoli [46, 9] propongono un calcolo di processi per sistemi IoT e la prima teoria semantica fully abstract per tali sistemi. Si noti che la progettazione di un calcolo di processi che modelli un nuovo paradigma richiede la comprensione e la conseguente "distillazione" degli aspetti principali del paradigma in un ambiente algebrico semplice ed elegante. Nei due articoli sopra menzionati viene proposto un calcolo di processi dotato delle semantiche operazionali classiche: una semantica a riduzione e un sistema di transizioni etichettate. A dimostrazione della bontà delle due definizioni viene provato che le due *semantiche operazionali* coincidono (Harmony theorem). Viene poi fornita una nozione di *bisimulazione*, definita a partire dal sistema di transizioni etichettate. Il contributo principale dei due articoli è un risultato di full abstraction, in cui la bisimulazione etichettata caratterizza completamente la barbed congruence del linguaggio. Si noti che, a differenza dei calcoli di processo con mobilità visti precedentemente, sia il calcolo in questione che la relativa bisimulazione sono first-order. Semplificando di molto la teoria semantica.

Le piattaforme IoT consentono agli utenti di connettere vari dispositivi intelligenti e servizi online tramite app reattive in esecuzione sul cloud. Queste app, spesso sviluppate da terze parti, eseguono semplici calcoli sui dati attivati da fonti di informazione esterne e attuano i risultati dei calcoli su pozzi di informazioni esterni. Ricerche recenti mostrano che le interazioni involontarie o dannose tra le diverse app (anche benigne) di un utente possono causare gravi rischi per la sicurezza. Questi lavori sfruttano

le tecniche di analisi dei programmi per creare strumenti in grado di svelare interferenze impreviste tra app per casi d'uso specifici. Nonostante questi sforzi iniziali, manca ancora un framework semantico per comprendere le interazioni tra le app IoT.

Negli articoli [35, 5] viene proposto un framework semantico che cattura l'essenza delle *interazioni tra app nelle piattaforme IoT*. Il framework generalizza e collega i meccanismi di applicazione sintattica a nozioni di sicurezza basate sulla bisimulazione, fornendo così una linea di base per la formulazione di criteri di solidit di questi meccanismi di applicazione. Nello specifico, presentiamo un calcolo che modella la semantica comportamentale di un sistema di app eseguite contemporaneamente e lo utilizziamo per definire politiche semantiche desiderabili mirate alla sicurezza e alla protezione delle app IoT. Per dimostrare l'utilità del nostro framework, definiamo e implementiamo analisi statiche per applicare la sicurezza e la protezione tra app e le dimostriamo valide rispetto alle nostre condizioni semantiche. Sfruttiamo anche le app del mondo reale per convalidare i vantaggi pratici dei nostri strumenti in base ai meccanismi di applicazione proposti.

Semantica formale per sistemi Ciberfisici. Negli articoli [46, 9] la componente fisica dei sistemi è stata opportunamente astratta per fornire un calcolo sufficientemente maneggevole per poter ragionare su sistemi IoT. Questo non è possibile quando si passa a sistemi ciberfisici, che spesso modellano processi fisici molto complessi e che vengono tipicamente rappresentati attraverso un sistema di equazioni differenziali (o equazioni alle differenze). Gli articoli [44, 6] propongono un calcolo di processi ibrido per specificare e ragionare su sistemi ciberfisici. In tale calcolo vi è una chiara distinzione tra la parte fisica del sistema (spesso chiamata "plant") e la parte ciber, individuata dal controllore e da altre componenti di supervisione dell'intero sistema. Nell'articolo [44] viene definita la prima bisimulazione per sistemi ciberfisici che consenta di ragionare in maniera composizionale. La composizionalità della semantica comportamentale è fondamentale visto che i sistemi in questioni possono essere anche di grosse dimensioni. Nell'articolo [6] proponiamo una versione probabilistica del calcolo per la modellazione e lo studio dei CPSs. La dinamica del calcolo espressa in termini di un sistema di transizione etichettato probabilistico nello stile SOS di Plotkin. Questo viene utilizzato per definire una semantica comportamentale probabilistica. Per un confronto accurato tra CPS, forniamo due metriche probabilistiche composizionali per formalizzare la nozione di distanza comportamentale tra sistemi, anche nel caso di computazioni limitate nel tempo.

Sicurezza dei sistemi Ciberfisici. Nell'articolo [42] si estende il calcolo proposto in [44] per un trattamento formale di attacchi ciberfisici di tipo *integrity* e *DoS* sui dispositivi fisici: sensori e attuatori. L'obiettivo è quello di formalizzare un *threat model* per CPSs e studiare attacchi a dispositivi fisici. Nella definizione del threat model si è prestata particolare attenzione a modellare gli aspetti temporali dell'attacco: inizio e durata. Per esempio un attacco lanciato quando il sistema si trova in uno stato fisico vicino alla soglia di ammissibilità potrebbe avere grande successo. Ma anche la durata dell'attacco può essere determinante: la rottura di un reattore chimico può richiedere solo pochi minuti, la rottura di un motore potrebbe richiedere ore, e la distruzione di centrifughe potrebbe richiedere mesi. Il lavoro sviluppato in [42] rappresenta un passo fondamentale per lo sviluppo di strumenti semi-automatici per la *verifica statica della sicurezza di sistemi ciberfisici*.

L'analisi statica di CPSs è la principale motivazione principale dell'articolo [36] in cui vengono gettate le fondamenta per tecniche di model checking per l'analisi di sistemi ciberfisici lineari supportata da logiche ad hoc, orientate al rilevamento statico di attacchi ciberfisici.

Gli articoli [39, 34, 27] mostrano esempi di sistemi ciberfisici, via via più complessi, in cui l'analisi di sicurezza statica viene effettuata attraverso strumenti di *model-checking* o di *model-checking statistico*. In particolare, l'articolo [27] studia l'impatto di *attacchi coordinati* che raggiungono il loro obiettivo malevolo operando contemporaneamente su componenti diverse del sistema sotto attacco.

Gli articoli [37, 31] rappresentano un primo passo verso la definizione di metriche formali per la quantificazione dell'*impatto di attacchi* ciberfisici sui processi fisici che stanno al cuore dei sistemi ciberfisici. Questi lavori si basano sulla definizione di opportune metriche comportamentali probabilistiche temporizzate.

Gli articoli [28, 1] propongono un framework formale per l'analisi quantitativa degli attacchi di sensori di sistemi industiali, utilizzando il formalismo della logica dinamica differenziale. Date una precondizione e una postcondizione di un sistema, formalizziamo due nozioni di sicurezza quantitativa: sicurezza quanti-

tativa in avanti e sicurezza all'indietro, che esprimono rispettivamente (1) quanto è forte la postcondizione più forte del sistema rispetto alla postcondizione specificata, e (2) quanto forte la precondizione specificata rispetto alla precondizione più debole. Introduciamo due nozioni, *robustezza* forward e backward, per caratterizzare la robustezza di un sistema contro gli attacchi ai sensori come perdita di sicurezza. Per ragionare con robustezza sono state sviluppate due distanze di simulazione, che caratterizzano rispettivamente i limiti superiori del grado di perdita di sicurezza in avanti e all'indietro causata dagli attacchi dei sensori. Verifichiamo le due distanze di simulazione esprimendole come formule di logica dinamica differenziale. Mostriamo un esempio di veicolo autonomo che deve evitare una collisione.

I dati di scansione pubblicati di recente su Shodan mostrano come migliaia di sistemi di controllo industriale (ICS) siano direttamente accessibili da Internet. In particolare, i componenti altamente sensibili, come i controllori logici programmabili (PLC), sono potenzialmente vulnerabili rispetto a diversi tipi di attacchi. Daltro canto, per realizzare attacchi cyber-fisici non banali, lattaccante deve possedere un grado sufficiente di comprensione dei processi fisici allinterno dellICS bersaglio. Negli articoli [25, 2] esploriamo la fattibilità della progettazione di strategie di offuscamento per impedire all'attaccante di comprendere il comportamento del processo fisico all'interno di un ICS accedendo ai registri di memoria del PLC. Proponiamo due strategie generiche di offuscamento per le memorie PLC, che coinvolgono registri di memoria, codice della PLC e processi fisici simulati controllati dai PLC offuscati. Quindi misuriamo l'efficacia delle strategie di offuscamento proposte in termini di potenza, resilienza e costo su un caso di studio non banale.

Infine, negli ultimi anni il Dr. Merro ha lavorato allo sviluppo di honepot per sistemi industriali. Nel contesto industriale, le honeypot rappresentano contromisure efficaci sia per difendersi da tali attacchi sia per scoprire nuove strategie di attacco. Negli ultimi anni, le honeypot per sistemi ICS hanno compiuto progressi significativi nellemulare fedelmente le reti OT, comprese le interazioni dei processi fisici. Negli articoli [29, 26] proponiamo HoneyICS, una honeynet ad alta interazione, sensibile alla fisica, scalabile ed estensibile per sistemi industriali, dotata di un sistema di monitoraggio avanzato. Abbiamo esposto la nostra honeynet su Internet e condotto esperimenti per valutare l'efficacia di HoneyICS. In particolare, l'articolo [24] presenta uno studio latitudinale su un set di dati comprendente interazioni sia IT che ICS raccolte da un'istanza di HoneyICS esposta su Internet per tre mesi. Lo studio si concentra su tre aspetti ortogonali di tali interazioni: livello di interazione, origine delle interazioni e modelli di interazione/attacco. I nostri risultati fanno luce sull'impatto delle diverse scelte nella configurazione di una honeynet in termini di attrattiva e comportamento catturato.

3 Attività didattica

Il Dr. Merro ha maturato un'esperienza ventennale nell'insegnamento universitario. Insegnamento tenuto prevalentemente presso il corso di laurea specialistica in Informatica, il corso di laurea triennale in Informatica e il corso di laurea magistrale in Ingegneria e scienze informatiche dell'Università di Verona.

3.1 Incarichi didattici

Il Dr. Merro è o è stato responsabile dei seguenti insegnamenti:

- *Fondamenti di linguaggi di programmazione e specifica* (modulo di 9CFU), 66h, laurea magistrale in Ingegneria e scienze informatiche, Università di Verona (2021-);
- *Network Security*, (6CFU) 48h, laurea magistrale in Ingegneria e Scienze Informatiche, Università di Verona (2021-);
- *Linguaggi* (modulo di 6CFU all'interno del corso di Fondamenti), 56h, laurea magistrale in Ingegneria e scienze informatiche, Università di Verona (2010-2020);
- *Sicurezza delle reti*, 56h, laurea magistrale in Ingegneria e Scienze Informatiche, Università di Verona (2013-2020);
- *Programmazione di rete*, 60h, laurea specialistica in Informatica, Università di Verona (2003-2013);

- *Linguaggi concorrenti e mobili*, 60h, laurea specialistica in Informatica, Univ. di Verona (2003-2009);
- *Programmazione II*, 60h, laurea triennale in Informatica, Università di Verona (2009-2010);
- *Informatica di base*, 24h, laurea specialistica in Informatica, Università di Verona (2006-2009);
- *Interazione Uomo-Macchina*, 20h, Facoltà di Letteratura e Filosofia, Univ. di Verona (2004-2005);
- *Teoria degli Automi*, 20h, University of Sussex, UK, (2000-2002).

3.2 Supervisione di tirocini, tesi di laurea triennale, specialistica e magistrale

Il Dr. Merro ha supervisionato (e anche revisionato) numerosi tirocini (in qualità di tutor aziendale), tesi di laurea specialistica, tesi di laurea triennale e tesi di laurea magistrale. Alcuni di questi lavori hanno dato luogo a pubblicazioni scientifiche apparse in atti di convegno internazionali o su riviste internazionali, a dimostrazione dello stretto legame tra l'attività scientifica e quella didattica del Dr. Merro.

- L'articolo [38] estende e generalizza alcuni dei risultati apparsi nella tesi di laurea magistrale di Andrei Munteanu.
- L'articolo [47] estende e generalizza alcuni dei risultati ottenuti durante il tirocinio e la tesi di laurea triennale di Alice Dal Corso; un lavoro preliminare a riguardo era stato precedentemente svolto durante i tirocini di Marco Campion e Giacomo Annaloro.
- L'articolo [49] estende e generalizza i risultati ottenuti durante il tirocinio e la tesi di laurea triennale di Luca Battisti.
- L'articolo [50] contiene un'analisi semantica di tre protocolli crittografici; in particolare approfondisce l'analisi del protocollo LiSP iniziata nella tesi di laurea specialistica di Mattia Tirapelle.
- Gli articoli [52, 13] estendono e generalizzano alcuni dei risultati ottenuti durante il tirocinio e la tesi di laurea triennale di Francesco Ballardin.
- L'articolo [53] estende e generalizza alcuni risultati apparsi nella tesi di laurea specialistica di Davide Benetti; la tesi di laurea di Benetti ha ottenuto nel 2010 il terzo premio Clusit (Associazione Italiana per la Sicurezza Informatica).
- L'articolo [57] estende e generalizza risultati della tesi di laurea specialistica di Corrado Biasi.

4 Incarichi istituzionali

- *Coordinatore del Collegio dei Docenti* del Dottorato in Informatica di Verona (2016-2022);
- Membro del Consiglio della *Scuola di Dottorato dell'Università di Verona* (2020-2022);
- *Referente per l'Assicurazione della Qualità* del Corso di laurea magistrale in Ingegneria e scienze informatiche (2016-2019);
- Membro del Consiglio della *Scuola di Dottorato in Scienze Naturali e Ingegneristiche* (2016-2020);
- *Presidente della Commissione Didattica* della Laurea in Informatica ed Informatica Multimediale, Università di Verona, (2006-2013);
- Membro della *Giunta del Consiglio di Dipartimento* di Informatica di Verona, (2009-15, 2018-22);
- Membro della *Commissione Didattica del GRIN* (2008-2011);
- Vice presidente del *Corso di Laurea in Informatica* della Facoltà di Scienze MMFFNN dell'Università di Verona (2006-2012).

Verona, 11 Febbraio 2025

Firma