1.	Record Nr.	UNINA9910483545803321
	Titolo	Programming Multi-Agent Systems : 7th International Workshop, ProMAS 2009, Budapest, Hungary, May10-15, 2009.Revised Selected Papers / / edited by Lars Braubach, Jean-Pierre Briot, John Thangarajah
	Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2010
	ISBN	1-280-38819-6 9786613566119 3-642-14843-3
	Edizione	[1st ed. 2010.]
	Descrizione fisica	1 online resource (XII, 285 p. 57 illus.)
	Collana	Lecture Notes in Artificial Intelligence, , 2945-9141 ; ; 5919
	Altri autori (Persone)	BraubachLars BriotJean-Pierre ThangarajahJohn
	Disciplina	006.3
	Soggetti	Artificial intelligence Computer networks Software engineering Computer programming Computer simulation Artificial Intelligence Computer Communication Networks Software Engineering Programming Techniques Computer Modelling
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Bibliographic Level Mode of Issuance: Monograph
	Nota di bibliografia	Includes bibliographical references and index.
	Nota di contenuto	Communication Models -- Programming Multiagent Systems without Programming Agents -- Elements of a Business-Level Architecture for Multiagent Systems -- A Computational Semantics for Communicating Rational Agents Based on Mental Models -- Formal Models -- Multi-Agent Systems: Modeling and Verification Using Hybrid Automata -- Probabilistic Behavioural State Machines -- Golog Speaks the BDI Language -- Organizations and Environments -- A Middleware for

Sommario/riassunto

The earliest work on agents may be traced at least to the ?rst conceptualization of the actor model by Carl Hewitt. In a paper in an AI conference in the early 1970s, Hewitt described actors as entities with knowledge and goals. Research on actors continued to focus on AI with the development of the Sprites model in which a monotonically growing knowledge base could be accessed by actors (inspired by what Hewitt called “the Scienti?c Computing Metaphor”). In the late1970sandwellinto 1980s, controversyragedinAIbetweenthosearguingfor declarative languages and those arguing for procedural ones. Actor researchers stood on the side of a procedural view of knowledge, arguing for an open s- tems perspective rather than the closed world hypothesis necessary for a logical, declarativeview. In the open systemsview,agentshad armslength relationships and could not be expected to store consistent facts, nor could the information in a system be considered complete (the “negation as failure” model). Subsequent work on actors, including my own, focused on using actors for general purpose concurrent and distributed programming. In the late 1980s, a number of actor languages and frameworks were built. These included Act++ (in C++) by Dennis Kafura and Actalk (in Smalltalk) by Jean-Pierre Briot. In recent times, the use of the Actor model, in various guises, has proliferated as new parallel and distributed computing platforms and applications have become common:clusters,Webservices,P2Pnetworks, clientprogrammingonmulticore processors, and cloud computing.