1. Record Nr.   UNINA9910144204803321

   Titolo   Software Engineering for Multi-Agent Systems II : Research Issues and Practical Applications / / edited by Carlos Lucena, Alessandro Garcia, Alexander Romanovsky, Jaelson Castro, Paulo S.C. Alencar

   Pubbl/distr/stampa   Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2004

   ISBN   1-280-30702-1
                        9786610307029
                        3-540-24625-8

   Edizione   [1st ed. 2004.]

   Descrizione fisica   1 online resource (xii, 278 p.) : ill. ;

   Collana   Lecture Notes in Computer Science, , 0302-9743 ; ; 2940

   Disciplina   005.1

   Soggetti   Software engineering
                        Computer networks
                        Computer programming
                        User interfaces (Computer systems)
                        Artificial intelligence
                        Software Engineering/Programming and Operating Systems
                        Software Engineering
                        Computer Communication Networks
                        Programming Techniques
                        User Interfaces and Human Computer Interaction
                        Artificial Intelligence

   Lingua di pubblicazione   Inglese

   Formato   Materiale a stampa

   Livello bibliografico   Monografia

   Note generali   Based on the 2nd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS 2003) held May 2003, Portland, Ore.

   Nota di bibliografia Includes bibliographical references at the end of each chapters and index.

   Nota di contenuto   Requirements Engineering -- The Agent at the Center of the Requirements Engineering Process -- Lexicon Based Ontology Construction -- Multi-agent Systems and Security Requirements Analysis -- Software Architecture and Design -- Separation of Concerns in Multi-agent Systems: An Empirical Study -- Architecting the Design of Multi-agent Organizations with Proto-frameworks -- A

Basic Taxonomy for Role Composition -- Modeling -- Object-Oriented Modeling Approaches to Agent-Based Workflow Services -- Using the MAS-ML to Model a Multi-agent System -- Software Engineering Challenges for Mutable Agent Systems -- Dependability -- Improving Exception Handling in Multi-agent Systems -- On Manageability and Robustness of Open Multi-agent Systems -- Security Mechanisms for Mobile Agent Platforms Based on SPKI/SDSI Chains of Trust -- MAS Frameworks -- Farm: A Scalable Environment for Multi-agent Development and Evaluation -- Role-Based Approaches for Engineering Interactions in Large-Scale Multi-agent Systems -- Evaluating Agent Architectures: Cougaar, Aglets and AAA.

Sommario/riassunto	Advances in networking technology have revitalized the investigation of agent technologyasapromisingparadigmforengineeringcomplexdistributedsoftware systems. Agent technology has been applied to a wide range of application - mains, including e-commerce, human-computer interfaces, telecommunications, and software assistants. Multi-agent systems (MASs) and their underlying t- ories provide a more natural support for ensuring important properties such as autonomy, mobility, environment heterogeneity, organization, openness, and intelligence. As a consequence, agent-based systems are likely to provide new - proaches to dealing with the complexity of developing and maintaining modern software. However, developing robust large-scale agent-based systems will - quire new software engineering approaches. There are currently many methods and techniques for working with individual agents or with systems built using only a few agents. Unfortunately, agent-based software engineering is still in its infancy and existing software engineering approaches are unable to cope with large MASs. The complexity associated with a large MAS is considerable. When a huge number of agents interact over heterogeneous environments, various phenomena occur which are not as easy to capture as when only a few agents are working together. As the multiple software agents are highly collaborative and operate in networked environments, they have to be context-aware and deal with - vironment uncertainty. This makes their coordination and management more di?cult and increases the likelihood of exceptional situations, such as security holes, privacy violations, and unexpected global e?ects. Moreover, as users and softwareengineersdelegatemoreautonomytotheirMASs, andputmoretrustin their results, new concerns arise in real-life applications.