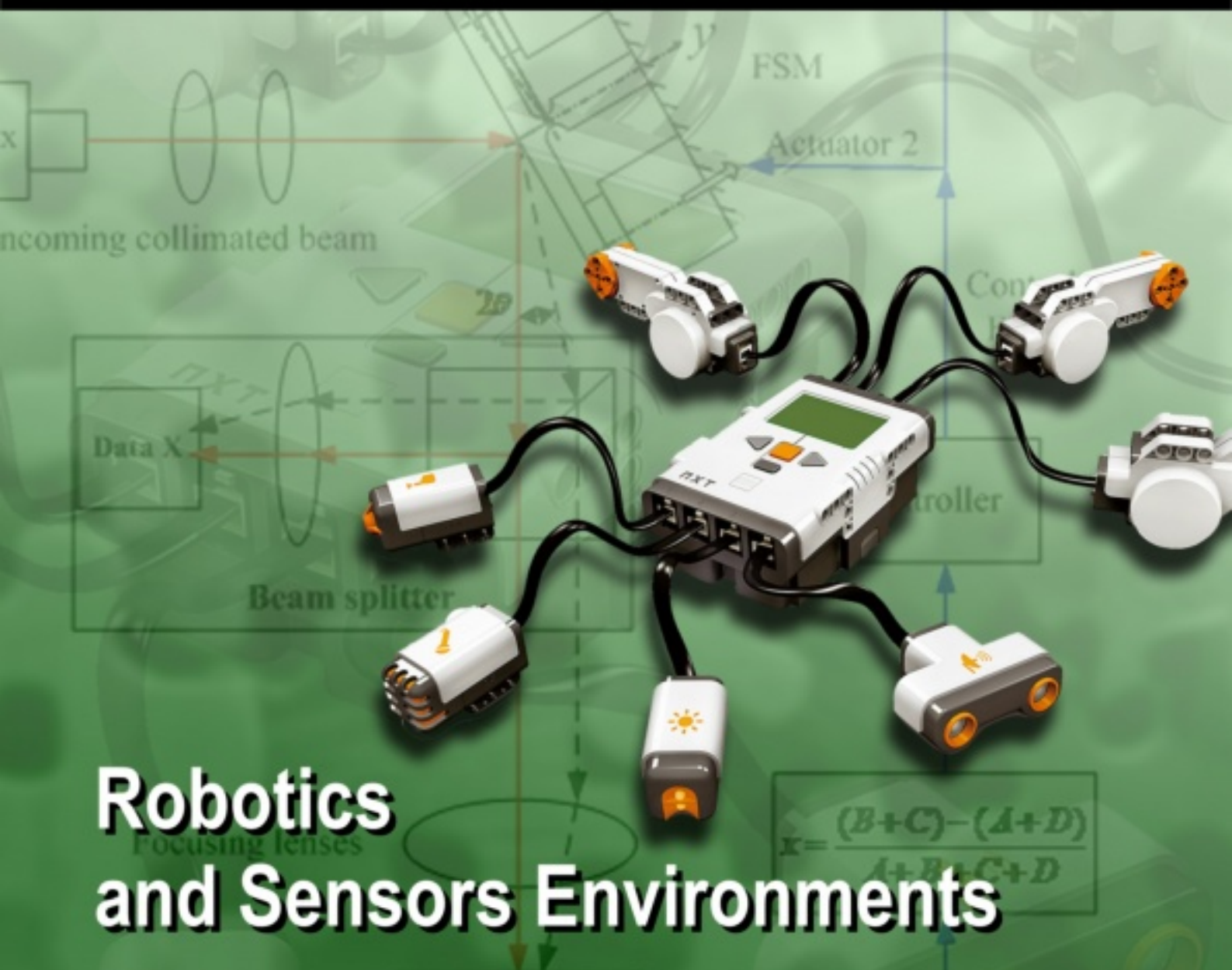


ISSN 1726-5749

SENSORS & TRANSDUCERS

3^{vol. 5}
Special
/09



Robotics and Sensors Environments

International Frequency Sensor Association Publishing



Guest Editors: Dr. Pierre Payeur and Dr. Emil M. Petriu, University of Ottawa, Ottawa, ON, Canada

Editor-in-Chief: professor Sergey Y. Yurish, phone: +34 696067716, fax: +34 93 4011989, e-mail: editor@sensorsportal.com

Editors for Western Europe

Meijer, Gerard C.M., Delft University of Technology, The Netherlands
Ferrari, Vittorio, Università di Brescia, Italy

Editor South America

Costa-Felix, Rodrigo, Inmetro, Brazil

Editor for Eastern Europe

Sachenko, Anatoly, Ternopil State Economic University, Ukraine

Editors for North America

Datskos, Panos G., Oak Ridge National Laboratory, USA
Fabien, J. Josse, Marquette University, USA
Katz, Evgeny, Clarkson University, USA

Editor for Asia

Ohyama, Shinji, Tokyo Institute of Technology, Japan

Editor for Asia-Pacific

Mukhopadhyay, Subhas, Massey University, New Zealand

Editorial Advisory Board

Abdul Rahim, Ruzairi, Universiti Teknologi, Malaysia
Ahmad, Mohd Noor, Northern University of Engineering, Malaysia
Annamalai, Karthigeyan, National Institute of Advanced Industrial Science and Technology, Japan
Arcega, Francisco, University of Zaragoza, Spain
Arguel, Philippe, CNRS, France
Ahn, Jae-Pyoung, Korea Institute of Science and Technology, Korea
Arndt, Michael, Robert Bosch GmbH, Germany
Ascoli, Giorgio, George Mason University, USA
Atalay, Selcuk, Inonu University, Turkey
Atghiaee, Ahmad, University of Tehran, Iran
Augutis, Vyantas, Kaunas University of Technology, Lithuania
Avachit, Patil Lalchand, North Maharashtra University, India
Ayesh, Aladdin, De Montfort University, UK
Bahreyni, Behraad, University of Manitoba, Canada
Baoxian, Ye, Zhengzhou University, China
Barford, Lee, Agilent Laboratories, USA
Barlingay, Ravindra, RF Arrays Systems, India
Basu, Sukumar, Jadavpur University, India
Beck, Stephen, University of Sheffield, UK
Ben Bouzid, Sihem, Institut National de Recherche Scientifique, Tunisia
Benachaiba, Chellali, Universitaire de Bechar, Algeria
Binnie, T. David, Napier University, UK
Bischoff, Gerlinde, Inst. Analytical Chemistry, Germany
Bodas, Dhananjay, IMTEK, Germany
Borges Carval, Nuno, Universidade de Aveiro, Portugal
Bousbia-Salah, Mounir, University of Annaba, Algeria
Bouvet, Marcel, CNRS – UPMC, France
Brudzewski, Kazimierz, Warsaw University of Technology, Poland
Cai, Chenxin, Nanjing Normal University, China
Cai, Qingyun, Hunan University, China
Campanella, Luigi, University La Sapienza, Italy
Carvalho, Vitor, Minho University, Portugal
Cecelja, Franjo, Brunel University, London, UK
Cerda Belmonte, Judith, Imperial College London, UK
Chakrabarty, Chandan Kumar, Universiti Tenaga Nasional, Malaysia
Chakravorty, Dipankar, Association for the Cultivation of Science, India
Changhai, Ru, Harbin Engineering University, China
Chaudhari, Gajanan, Shri Shivaji Science College, India
Chen, Jiming, Zhejiang University, China
Chen, Rongshun, National Tsing Hua University, Taiwan
Cheng, Kuo-Sheng, National Cheng Kung University, Taiwan
Chiang, Jeffrey (Cheng-Ta), Industrial Technol. Research Institute, Taiwan
Chiriac, Horia, National Institute of Research and Development, Romania
Chowdhuri, Arijit, University of Delhi, India
Chung, Wen-Yaw, Chung Yuan Christian University, Taiwan
Corres, Jesus, Universidad Publica de Navarra, Spain
Cortes, Camilo A., Universidad Nacional de Colombia, Colombia
Courtois, Christian, Université de Valenciennes, France
Cusano, Andrea, University of Sannio, Italy
D'Amico, Arnaldo, Università di Tor Vergata, Italy
De Stefano, Luca, Institute for Microelectronics and Microsystem, Italy
Deshmukh, Kiran, Shri Shivaji Mahavidyalaya, Barshi, India
Dickert, Franz L., Vienna University, Austria
Dieguez, Angel, University of Barcelona, Spain
Dimitropoulos, Panos, University of Thessaly, Greece
Ding Jian, Ning, Jiangsu University, China
Djordjević, Alexandar, City University of Hong Kong, Hong Kong

Donato, Nicola, University of Messina, Italy
Donato, Patricio, Universidad de Mar del Plata, Argentina
Dong, Feng, Tianjin University, China
Drljaca, Predrag, Intersema Sensoric SA, Switzerland
Dubey, Venketesh, Bournemouth University, UK
Enderle, Stefan, University of Ulm and KTB Mechatronics GmbH, Germany
Erdem, Gursan K. Arzum, Ege University, Turkey
Erkmen, Aydan M., Middle East Technical University, Turkey
Estelle, Patrice, Insa Rennes, France
Estrada, Horacio, University of North Carolina, USA
Faiz, Adil, INSA Lyon, France
Fericean, Sorin, Balluff GmbH, Germany
Fernandes, Joana M., University of Porto, Portugal
Francioso, Luca, CNR-IMM Institute for Microelectronics and Microsystems, Italy
Francis, Laurent, University Catholique de Louvain, Belgium
Fu, Weiling, South-Western Hospital, Chongqing, China
Gaura, Elena, Coventry University, UK
Geng, Yanfeng, China University of Petroleum, China
Gole, James, Georgia Institute of Technology, USA
Gong, Hao, National University of Singapore, Singapore
Gonzalez de la Rosa, Juan Jose, University of Cadiz, Spain
Grael, Annette, Goteborg University, Sweden
Graff, Mason, The University of Texas at Arlington, USA
Guan, Shan, Eastman Kodak, USA
Guillet, Bruno, University of Caen, France
Guo, Zhen, New Jersey Institute of Technology, USA
Gupta, Narendra Kumar, Napier University, UK
Hadjiloucas, Sillas, The University of Reading, UK
Hashsham, Syed, Michigan State University, USA
Hernandez, Alvaro, University of Alcalá, Spain
Hernandez, Wilmar, Universidad Politecnica de Madrid, Spain
Homentcovski, Dorel, SUNY Binghamton, USA
Horstman, Tom, U.S. Automation Group, LLC, USA
Hsiai, Tzung (John), University of Southern California, USA
Huang, Jeng-Sheng, Chung Yuan Christian University, Taiwan
Huang, Star, National Tsing Hua University, Taiwan
Huang, Wei, PSG Design Center, USA
Hui, David, University of New Orleans, USA
Jaffrezic-Renault, Nicole, Ecole Centrale de Lyon, France
Jaime Calvo-Galleg, Jaime, Universidad de Salamanca, Spain
James, Daniel, Griffith University, Australia
Janting, Jakob, DELTA Danish Electronics, Denmark
Jiang, Liudi, University of Southampton, UK
Jiang, Wei, University of Virginia, USA
Jiao, Zheng, Shanghai University, China
John, Joachim, IMEC, Belgium
Kalach, Andrew, Voronezh Institute of Ministry of Interior, Russia
Kang, Moonho, Sunmoon University, Korea South
Kaniusas, Eugenijus, Vienna University of Technology, Austria
Katake, Anup, Texas A&M University, USA
Kausel, Wilfried, University of Music, Vienna, Austria
Kavasoglu, Nese, Mugla University, Turkey
Ke, Cathy, Tyndall National Institute, Ireland
Khan, Asif, Aligarh Muslim University, Aligarh, India
Kim, Min Young, Kyungpook National University, Korea South
Sandacci, Serghei, Sensor Technology Ltd., UK

- Ko, Sang Choon**, Electronics and Telecommunications Research Institute, Korea South
- Kockar, Hakan**, Balikesir University, Turkey
- Kotulska, Malgorzata**, Wroclaw University of Technology, Poland
- Kratz, Henrik**, Uppsala University, Sweden
- Kumar, Arun**, University of South Florida, USA
- Kumar, Subodh**, National Physical Laboratory, India
- Kung, Chih-Hsien**, Chang-Jung Christian University, Taiwan
- Lacnjevac, Caslav**, University of Belgrade, Serbia
- Lay-Ekuakille, Aime**, University of Lecce, Italy
- Lee, Jang Myung**, Pusan National University, Korea South
- Lee, Jun Su**, Amkor Technology, Inc. South Korea
- Lei, Hua**, National Starch and Chemical Company, USA
- Li, Genxi**, Nanjing University, China
- Li, Hui**, Shanghai Jiaotong University, China
- Li, Xian-Fang**, Central South University, China
- Liang, Yuanchang**, University of Washington, USA
- Liawruangrath, Saisune**, Chiang Mai University, Thailand
- Liew, Kim Meow**, City University of Hong Kong, Hong Kong
- Lin, Hermann**, National Kaohsiung University, Taiwan
- Lin, Paul**, Cleveland State University, USA
- Linderholm, Pontus**, EPFL - Microsystems Laboratory, Switzerland
- Liu, Aihua**, University of Oklahoma, USA
- Liu Changgeng**, Louisiana State University, USA
- Liu, Cheng-Hsien**, National Tsing Hua University, Taiwan
- Liu, Songqin**, Southeast University, China
- Lodeiro, Carlos**, Universidade NOVA de Lisboa, Portugal
- Lorenzo, Maria Encarnacio**, Universidad Autonoma de Madrid, Spain
- Lukaszewicz, Jerzy Pawel**, Nicholas Copernicus University, Poland
- Ma, Zhanfang**, Northeast Normal University, China
- Majstorovic, Vidosav**, University of Belgrade, Serbia
- Marquez, Alfredo**, Centro de Investigacion en Materiales Avanzados, Mexico
- Matay, Ladislav**, Slovak Academy of Sciences, Slovakia
- Mathur, Prafull**, National Physical Laboratory, India
- Maurya, D.K.**, Institute of Materials Research and Engineering, Singapore
- Mekid, Samir**, University of Manchester, UK
- Melnyk, Ivan**, Photon Control Inc., Canada
- Mendes, Paulo**, University of Minho, Portugal
- Mennell, Julie**, Northumbria University, UK
- Mi, Bin**, Boston Scientific Corporation, USA
- Minas, Graca**, University of Minho, Portugal
- Moghavvemi, Mahmoud**, University of Malaya, Malaysia
- Mohammadi, Mohammad-Reza**, University of Cambridge, UK
- Molina Flores, Esteban**, Benemérita Universidad Autónoma de Puebla, Mexico
- Moradi, Majid**, University of Kerman, Iran
- Morello, Rosario**, DIMET, University "Mediterranea" of Reggio Calabria, Italy
- Mounir, Ben Ali**, University of Sousse, Tunisia
- Mulla, Imtiaz Sirajuddin**, National Chemical Laboratory, Pune, India
- Neelamegam, Periasamy**, Sastra Deemed University, India
- Neshkova, Milka**, Bulgarian Academy of Sciences, Bulgaria
- Oberhammer, Joachim**, Royal Institute of Technology, Sweden
- Ould Lahoucine, Cherif**, University of Guelma, Algeria
- Pamidighanta, Sayanu**, Bharat Electronics Limited (BEL), India
- Pan, Jisheng**, Institute of Materials Research & Engineering, Singapore
- Park, Joon-Shik**, Korea Electronics Technology Institute, Korea South
- Penza, Michele**, ENEA C.R., Italy
- Pereira, Jose Miguel**, Instituto Politecnico de Seteбал, Portugal
- Petsev, Dimitar**, University of New Mexico, USA
- Pogacnik, Lea**, University of Ljubljana, Slovenia
- Post, Michael**, National Research Council, Canada
- Prance, Robert**, University of Sussex, UK
- Prasad, Ambika**, Gulbarga University, India
- Prateepasen, Asa**, Kingmoungut's University of Technology, Thailand
- Pullini, Daniele**, Centro Ricerche FIAT, Italy
- Pumera, Martin**, National Institute for Materials Science, Japan
- Radhakrishnan, S.**, National Chemical Laboratory, Pune, India
- Rajanna, K.**, Indian Institute of Science, India
- Ramadan, Qasem**, Institute of Microelectronics, Singapore
- Rao, Basuthkar**, Tata Inst. of Fundamental Research, India
- Raouf, Kosai**, Joseph Fourier University of Grenoble, France
- Reig, Candid**, University of Valencia, Spain
- Restivo, Maria Teresa**, University of Porto, Portugal
- Robert, Michel**, University Henri Poincare, France
- Rezazadeh, Ghader**, Urmia University, Iran
- Royo, Santiago**, Universitat Politècnica de Catalunya, Spain
- Rodriguez, Angel**, Universidad Politécnica de Cataluña, Spain
- Rothberg, Steve**, Loughborough University, UK
- Sadana, Ajit**, University of Mississippi, USA
- Sadeghian Marnani, Hamed**, TU Delft, The Netherlands
- Sapozhnikova, Ksenia**, D.I.Mendeleyev Institute for Metrology, Russia
- Saxena, Vibha**, Bhabha Atomic Research Centre, Mumbai, India
- Schneider, John K.**, Ultra-Scan Corporation, USA
- Seif, Selemeni**, Alabama A & M University, USA
- Seifter, Achim**, Los Alamos National Laboratory, USA
- Sengupta, Deepak**, Advance Bio-Photonics, India
- Shankar, B. Baliga**, General Monitors Transnational, USA
- Shearwood, Christopher**, Nanyang Technological University, Singapore
- Shin, Kyuho**, Samsung Advanced Institute of Technology, Korea
- Shmaliy, Yuriy**, Kharkiv National University of Radio Electronics, Ukraine
- Silva Girao, Pedro**, Technical University of Lisbon, Portugal
- Singh, V. R.**, National Physical Laboratory, India
- Slomovitz, Daniel**, UTE, Uruguay
- Smith, Martin**, Open University, UK
- Soleymannpour, Ahmad**, Damghan Basic Science University, Iran
- Somani, Prakash R.**, Centre for Materials for Electronics Technol., India
- Srinivas, Talabattula**, Indian Institute of Science, Bangalore, India
- Srivastava, Arvind K.**, Northwestern University, USA
- Stefan-van Staden, Raluca-Ioana**, University of Pretoria, South Africa
- Sumriddetchka, Sarun**, National Electronics and Computer Technology Center, Thailand
- Sun, Chengliang**, Polytechnic University, Hong-Kong
- Sun, Dongming**, Jilin University, China
- Sun, Junhua**, Beijing University of Aeronautics and Astronautics, China
- Sun, Zhiqiang**, Central South University, China
- Suri, C. Raman**, Institute of Microbial Technology, India
- Sysoev, Victor**, Saratov State Technical University, Russia
- Szewczyk, Roman**, Industrial Research Institute for Automation and Measurement, Poland
- Tan, Ooi Kiang**, Nanyang Technological University, Singapore
- Tang, Dianping**, Southwest University, China
- Tang, Jaw-Luen**, National Chung Cheng University, Taiwan
- Teker, Kasif**, Frostburg State University, USA
- Thumbavanam Pad, Kartik**, Carnegie Mellon University, USA
- Tian, Gui Yun**, University of Newcastle, UK
- Tsiantos, Vassilios**, Technological Educational Institute of Kaval, Greece
- Tsigara, Anna**, National Hellenic Research Foundation, Greece
- Twomey, Karen**, University College Cork, Ireland
- Valente, Antonio**, University, Vila Real, - U.T.A.D., Portugal
- Vaseashta, Ashok**, Marshall University, USA
- Vazquez, Carmen**, Carlos III University in Madrid, Spain
- Vieira, Manuela**, Instituto Superior de Engenharia de Lisboa, Portugal
- Vigna, Benedetto**, STMicroelectronics, Italy
- Vrba, Radimir**, Brno University of Technology, Czech Republic
- Wandelt, Barbara**, Technical University of Lodz, Poland
- Wang, Jiangping**, Xi'an Shiyou University, China
- Wang, Kedong**, Beihang University, China
- Wang, Liang**, Advanced Micro Devices, USA
- Wang, Mi**, University of Leeds, UK
- Wang, Shinn-Fwu**, Ching Yun University, Taiwan
- Wang, Wei-Chih**, University of Washington, USA
- Wang, Wensheng**, University of Pennsylvania, USA
- Watson, Steven**, Center for NanoSpace Technologies Inc., USA
- Weiping, Yan**, Dalian University of Technology, China
- Wells, Stephen**, Southern Company Services, USA
- Wolkenberg, Andrzej**, Institute of Electron Technology, Poland
- Woods, R. Clive**, Louisiana State University, USA
- Wu, DerHo**, National Pingtung University of Science and Technology, Taiwan
- Wu, Zhaoyang**, Hunan University, China
- Xiu Tao, Ge**, Chuzhou University, China
- Xu, Lisheng**, The Chinese University of Hong Kong, Hong Kong
- Xu, Tao**, University of California, Irvine, USA
- Yang, Dongfang**, National Research Council, Canada
- Yang, Wuqiang**, The University of Manchester, UK
- Ymeti, Aurel**, University of Twente, Netherlands
- Yong Zhao**, Northeastern University, China
- Yu, Haihu**, Wuhan University of Technology, China
- Yuan, Yong**, Massey University, New Zealand
- Yufra Garcia, Alberto**, Seville University, Spain
- Zagnoni, Michele**, University of Southampton, UK
- Zeni, Luigi**, Second University of Naples, Italy
- Zhong, Haoxiang**, Henan Normal University, China
- Zhang, Minglong**, Shanghai University, China
- Zhang, Quintao**, University of California at Berkeley, USA
- Zhang, Weiping**, Shanghai Jiao Tong University, China
- Zhang, Wenming**, Shanghai Jiao Tong University, China
- Zhou, Zhi-Gang**, Tsinghua University, China
- Zorzano, Luis**, Universidad de La Rioja, Spain
- Zourob, Mohammed**, University of Cambridge, UK

Contents

Volume 5
Special Issue
March 2009

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Foreword

Pierre Payeur and Emil Petriu 1

An Omnidirectional Stereoscopic System for Mobile Robot Navigation

Rémi Boutteau, Xavier Savatier, Jean-Yves Ertaud, Bélahcène Mazari 3

Movement in Collaborative Robotic Environments Based on the Fish Shoal Emergent Patterns

Razvan Cioarga, Mihai V. Micea, Vladimir Cretu, Emil M. Petriu..... 18

A Multiscale Calibration of a Photon Videomicroscope for Visual Servo Control: Application to MEMS Micromanipulation and Microassembly

Brahim Tamadazte, Sounkalo Dembélé and Nadine Piat..... 37

A Study on Dynamic Stiffening of a Rotating Beam with a Tip Mass

Shengjian Bai, Pinhas Ben-Tzvi, Qingkun Zhou, Xinsheng Huang..... 53

Towards a Model and Specification for Visual Programming of Massively Distributed Embedded Systems

Meng Wang, Varun Subramanian, Alex Doboli, Daniel Curiac, Dan Pescaru and Codruta Istin 69

Feature Space Dimensionality Reduction for Real-Time Vision-Based Food Inspection

Mai Moussa CHETIMA and Pierre PAYEUR 86

Design and Analysis of a Fast Steering Mirror for Precision Laser Beams Steering

Qingkun Zhou, Pinhas Ben-Tzvi and Dapeng Fan..... 104

Neural Gas and Growing Neural Gas Networks for Selective 3D Sensing: a Comparative Study

Ana-Maria Cretu, Pierre Payeur and Emil M. Petriu..... 119

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal's webpage with preparation instructions: <http://www.sensorsportal.com/HTML/DIGEST/Submission.htm>

Towards a Model and Specification for Visual Programming of Massively Distributed Embedded Systems

Meng WANG, Varun SUBRAMANIAN and Alex DOBOLI

Department of Electrical and Computer Engineering, State University of New York at Stony Brook,
Stony Brook, NY 11794-2350, USA
Tel.: +1-631-632-1611, fax: +1-631-632-8494
E-mail: adoboli@ece.sunysb.edu

Daniel CURIAC, Dan PESCARU and Codruta ISTIN

Faculty of Automatics and Computers, "Politehnica" University Timisoara,
Timisoara, 300223, Romania
Tel.: +40-256-403227, fax: +40-256-403214
E-mail: daniel.curiaac@aut.upt.ro

Received: 30 January 2009 / Accepted: 24 February 2009 / Published: 23 March 2009

Abstract: Massively distributed embedded systems are rapidly emerging as a key concept for many modern applications. However, providing efficient and scalable decision making capabilities to such systems is currently a significant challenge. This paper proposes a model and a specification language to allow automated synthesis of distributed controllers, which implement and interact through formalisms of different semantics. The paper refers to a case study to illustrate the main capabilities of the proposed concept. *Copyright © 2009 IFSA.*

Keywords: Cyber-physical systems, Networks of sensing systems, Goal-oriented programming

1. Introduction

Massively distributed embedded systems are rapidly emerging as a key enabling concept for many modern applications in infrastructure management, environmental monitoring, energy conservation, healthcare, homeland security, manufacturing, and many other [1-6]. This is due not only to sensing and electronic devices becoming extremely cheap and small in size (thus, deployable in large numbers) but also to the potential of having significantly superior decision making quality if related problems

are tackled together instead of separately. Moreover, this helps improving the robustness of the systems as experience has shown that many disasters occur due to unaccounted correlations between the sub-systems. Providing reliable and efficient decision making capabilities to massively distributed embedded systems currently represents a main challenge [7].

The envisioned decision making paradigm is different from existing approaches, which either focus on centralized control or on local control [7], many times using ideas inspired from social or biological systems. Centralized control is well understood and reliable but does not scale well for large systems. In contrast, local control works well for large systems but with the exception of simple situations, its overall performance is hard to be captured. We argue for a distributed decision making mechanism in which parameterized control procedures implementing different strategies are dynamically introduced or removed depending on the specific optimization goals and operation conditions of the application. The controllers operate according to different models depending on the nature of their decision making. For example, reactive controllers use physical inputs for local control but operate under the constraints set by more abstract decision making procedures, which analyze broader situations based on aggregated (abstracted) data and procedures. The proposed decision making mechanism is flexible, scalable, and predictable. A key component of the approach is a suitable model and specification notation for massively parallel applications.

This paper presents a novel control model and the related specification for developing massively distributed embedded applications. We believe that a main challenge in providing scalable descriptions for such applications is due to the wide variety of interactions that emerge among the composing sub-systems, some of which are hard to anticipate a-priori, or might change their importance dynamically during execution. The proposed solution is to describe the nature of interactions that might occur among modules while leaving the task of optimally implementing these interactions to the compiler and execution environment.

More specifically, the proposed model defines the operation goals of each sub-system (e.g., the criteria to be maximized or minimized during execution) and the physical capabilities of a module to achieve a certain goal (such as its maximum processing speed, highest bandwidth, etc). Different interaction types are introduced depending on the way the sub-systems influence each other's goals and capabilities. The specification is compiled into a network of Decision Modules (DMs), which use reactive models to control decisions at the physical level, and more abstract formalisms, such as Task Graphs and Markov Decision Processes, to perform broader, more strategic decisions. The multi-semantic DMs interact through (i) constraints by which the strategic modules restrict the reactive DMs to guarantee satisfaction of the global goals and (ii) feedback offered by the reactive DMs about the feasibility of the constraints.

Section 2 summarizes the related work. Section 3 defines the distributed control model and Section 4 introduces the main language constructs. Section 5 illustrates the design of an application. Finally, conclusions are offered.

2. Related Work

Developing concurrent processing systems is known to be a difficult and error-prone activity [8, 9, 10]. Existing programming models for concurrent processing differ depending on the granularity of the supported parallelism and communication [11]. Traditionally, concurrent threads synchronize and communicate through low level mechanisms, like semaphores, critical regions, send-receive primitives, etc. Interrupts are the main hardware support for communication [12]. More recent work suggests coordination models, in which interfaces define the proper cooperation and communication between threads [11-13]. Coordination models describe a common tuple space for placing and

retrieving data (e.g., Linda [14]), adaptive parallelism by need-based activation and deactivation of threads (like Piranha [15]), manipulation of the tuple space including aggregation (i.e. Bonita [16]), constrained based communication (i.e., Law-Governed Linda [17]), moving objects from one space to another (e.g., Objective Linda [18]), and dynamic transformation of the tuple space objects (like in Gamma [19]). Compositional models, a special class of coordination models, include constructs for integrating components into bigger programs, such that the properties of the components are preserved. The integration constructs include logic clauses, like parallel AND operations in Strand [20], and higher-order functions [21, 22]. Interface-based design [23-26] is based on black-box models defining the system properties of the component interfaces, like arrival rate, latency, and capacity of the shared resources.

The concept of Visual Programming (VP) was arguably proposed in the 80s [27], however, it is only recently that its advantages for embedded applications became apparent. VP languages have been proposed for applications like managing smart oilfields [4, 28], vehicle tracking [29, 30], contour finding [29], environmental monitoring [2], etc. Region Streams [29] is a functional macro-programming language for sensor networks. Specification is based on successive filtering and functional processing of data pools. The data model is based on continuous data streams sampled from the environment and groups of nodes defined by their specific interests in space and over time. Language constructs enable aggregation of the data streams from a region and application of a function to the streams in a region. Abstract Task Graphs [2] is also a functional specification in which tasks sample from and place data into pools. There is no other interaction type between tasks. Channels are filters for associating only specific data from a pool to a task. Tasks are executed periodically or when input data is available. Semantic Streams [5] implements a query-based programming paradigm that fits well applications in which sensor networks operate as large distributed databases. Queries formulated as logic programs are converted by the compiler into a service graph for the network. Data sensing is modeled as streams. Other constructs include filtering by specifying properties of the streams, defining regions and sub-regions of the physical space, and performance requirements (e.g., quality of service). Kairos [30] proposes a set of language-independent extensions for describing global behavior of sensor networks controlled centrally. The extensions assume shared memory to allow any node to iterate through its neighbors and address arbitrary nodes.

Similar to [5, 29], the data model of the proposed specification language is based on data pools and continuous data streams to the modules. However, it differs in that it focuses on optimal decision making in massively distributed environment and not on algorithmic descriptions. Therefore, we argue that the language is orthogonal to the existing notations as it concentrates on the interactions between groups of nodes, or nodes and environment, and less on the behavior of the individual nodes.

3. Goal-Oriented Model

Cyber-physical Systems (CPS) are defined by a large set of interacting sub-systems. Each sub-system has well defined functionality and performance requirements, however, the interactions are dynamic, and their nature and intensity changes during execution. Fig. 1 illustrates a simplified CPS for intelligent traffic management. The goal of the system is to optimize the traffic flow of a region by adjusting the traffic-light characteristics to the specifics of the traffic flow. The example comprises of four sub-systems, the sub-system represented by the moving cars, the two data collection sub-systems (based on video cameras and sound-based tracking), and the sub-system formed by the traffic-light of the region. Each sub-system has a well defined functionality.

- The image-based tracking system collects video images of traffic to get information like car position and speed, distance between cars, average number of cars passing through a region, size of car clusters, and so on. Video images have a certain precision in time and coverage: they are

acquired at certain time intervals, and they might offer only a partial field-of-view covering. The sub-system can also detect events, like accidents or certain “special” cars, like police cars.

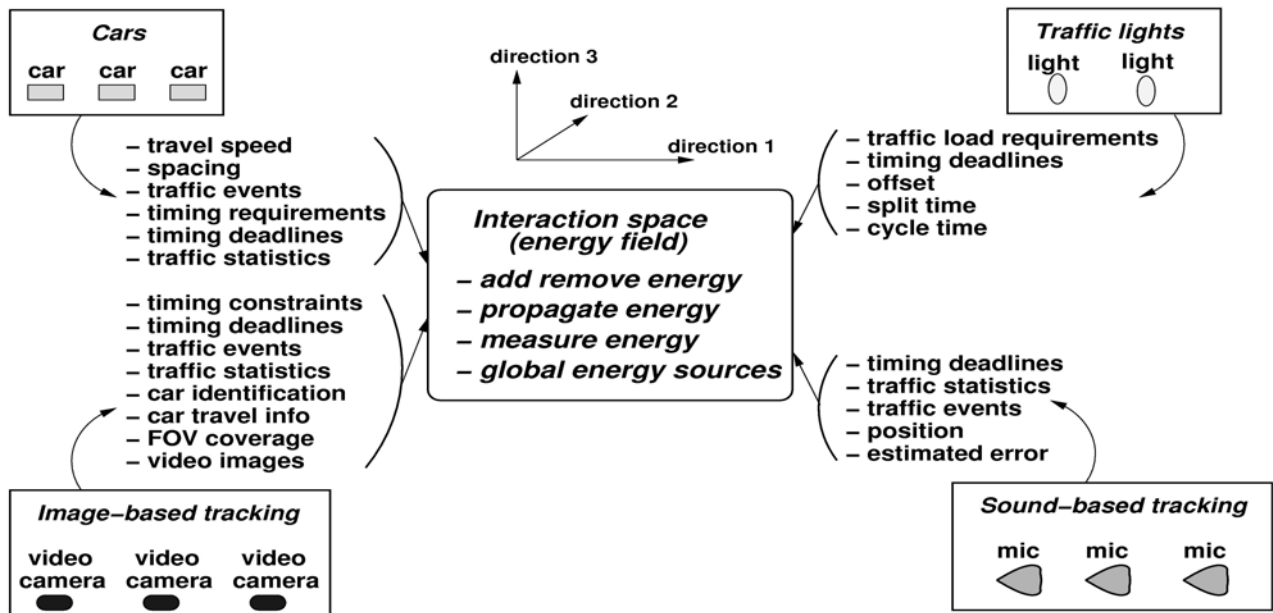


Fig. 1. Interactions in Cyber-Physical Systems.

- The sound-based tracking system follows the moving of specific cars, such as police cars, fire trucks, and ambulances. Having a tracking alternative can be very useful if the field of view to a vehicle is lost, or if the video-based tracking is too slow to meet the necessary timing constraints of the application. The system computes the position of a tracked car with a given precision and at certain intervals of time. The system also computes traffic statistics, like average speed, and detects traffic events, such as stopped cars.
- The moving cars form a group of mobile embedded nodes. The mobile nodes can interact with each other both directly through a wireless, ad-hoc network, and indirectly by sharing the same physical space at a certain time. The node dynamics is described by attributes like speed and distance to neighboring cars.
- The traffic-lights subsystem includes all traffic-lights of a region. The traffic-lights controllers are connected in a network to better coordinate their parameters offset, cycle and split times.

3.1. Sub-system Modeling: Functionality and Performance

Each CPS sub-system is defined as shown in Fig. 2.

The figure illustrates the main characteristics of the proposed distributed control concept and the related specification language. The wide geographical areas from which data is sampled define pools of heterogeneous data (e.g., images, temperature, humidity, number of moving vehicles, etc.). The association of data acquisition to the physical area is defined graphically, as shown in the left figure by the larger and smaller rectangles. For each defined area, the user specifies the goals that must be achieved by the distributed application, such as maximizing the traffic flow through the area, or keeping the pollution level below a preset limit.

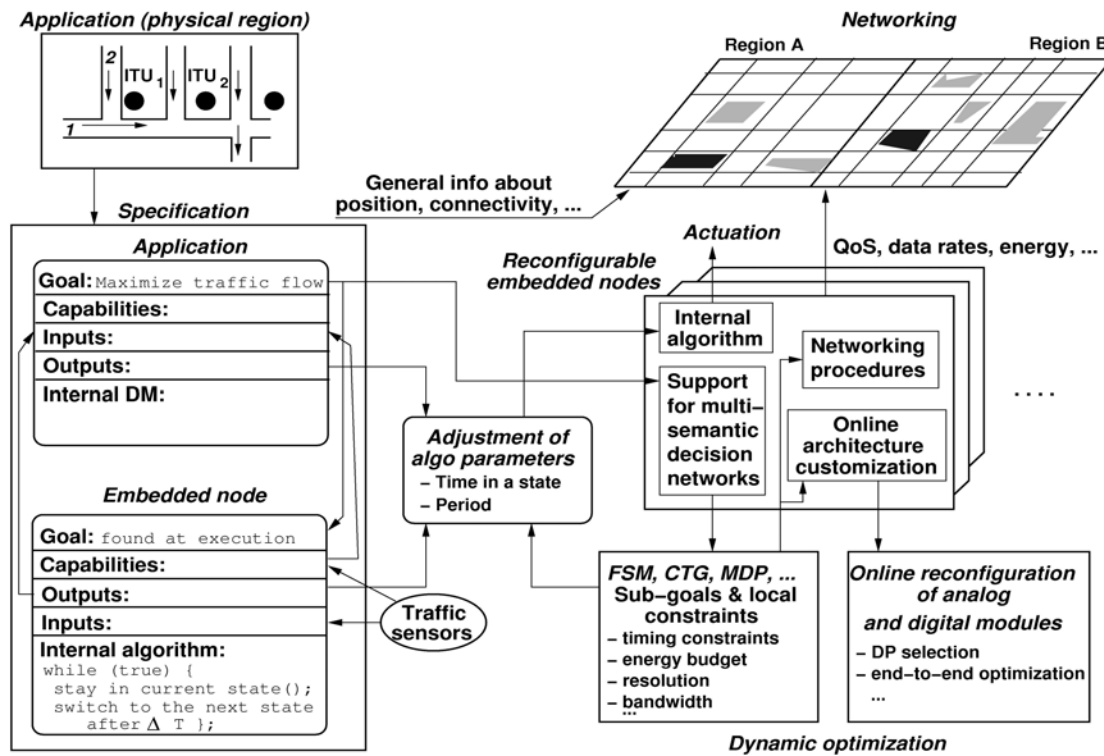


Fig. 2. Sub-system description for distributed decision making and control.

The execution platform is a massively distributed network of embedded controllers. Each controller node comprises hardware for sensing and actuation as well as processing, storage, and communication. In addition, hardware is reconfigurable, so that the available pool of resources at a node can be configured to meet different performance trade-offs, like variable processing speed, energy consumption, resolution, storage space, and so on. Hence, the model assumes that the functionality (algorithm) of each node is well defined and fixed but the node's performance is parametric and dynamically changing at execution.

The following three aspects are the core of the proposed model:

- *Separation of algorithmic aspects from goals:* Algorithms describe the ontology of the application, and, over time, remain the same for all nodes in the network. In contrast, goals define optimization criteria, performance, safety, etc., depend on the specific execution platform and conditions, and change dynamically in time. While building algorithms is arguably more efficiently done by humans, finding the parameters for optimal execution is cumbersome but can be automated.
- *Description of interdependent, heterogeneous sub-systems:* Global goals in large applications are likely to transcend different sub-system types. Also, the significance of the various related components can change over time. This invalidates static interaction schemes between sub-systems.
- *Avoiding explicit descriptions of synchronizations and data transfers:* Explicit specification of inter-node communication reduces scalability and reusability. It is hard and unreliable to attempt capturing all possible interactions between the components of massively large scale applications. Instead, the design environment ought to identify the best interaction schemes between components, so that the overall goals as well as the goals of the modules are met.

The proposed goal-oriented model comprises of separate Decision Modules (DMs) that operate to optimize a well-defined set of goals while the overall goals of the application are also being optimized. Each module executes a set of parameterized behaviors (algorithms) for which the parameters are automatically computed based on the information provided through the goal-oriented descriptions. DMs interact with each other only in small numbers but can perform global decisions by using data

aggregated from large regions and by strategic decision methods (e.g., based on Markov Decision Processes). Fig. 3 illustrates a network, which comprises different decision making models (Finite State Machines, Markov Decision Processes, etc.).

Similar to other specification languages for sensor networks, the proposed data model is based on data pools associated to regions and groups. Modules sample inputs from and generate outputs to a data pool for region. Regions represent continuous collections of tokens, such as a geographical area. Groups are discrete collections of tokens. Regions and groups can be associated to a specific physical area of the environment, or can be described by their defining properties.

Efficient and robust decision making must be guaranteed for hard-to-predict conditions. The overall behavior should be capable of autonomously meeting performance requirements, e.g. real-time constraints, bandwidth limitations, energy constraints, speed requirements, precision, etc. The underlying decision making model (DMM) is based on a multi-semantic decision making networks that represents the overall application performance at different levels of abstraction and using different evaluation formalisms. The low levels employ reactive models, e.g., Finite State Machines, which are capable of tackling unexpected situations. The upper levels use less flexible models but with more predictable performance, such as Data Flow Graphs and Task Graphs. Thus, the semantic hierarchy offers a smooth transition from the fully reactive behavior at the embedded node level and the deterministic operation at the application level. The number of abstraction levels and the decision making models for each level depend on the application.

The interaction between the different semantic models is achieved through top-down and bottom-up constraint transformation along the entire multi-semantic network: the top-down mechanism constrains the lower decision making levels by bounds introduced for their goals. As long as the low level operation stays within the bound it can be guaranteed that the overall performance is satisfied. The bottom-up mechanism signals when constraints (imposed by the upper levels) are cannot be satisfied by the lower levels. Fig. 3 shows a semantic hierarchy with three models, the bottom layers represent reactive behavior, and the top layers offer a performance predictive description of the system as Task Graphs (TGs) with data dependencies and Markov Decision Processes (MDPs).

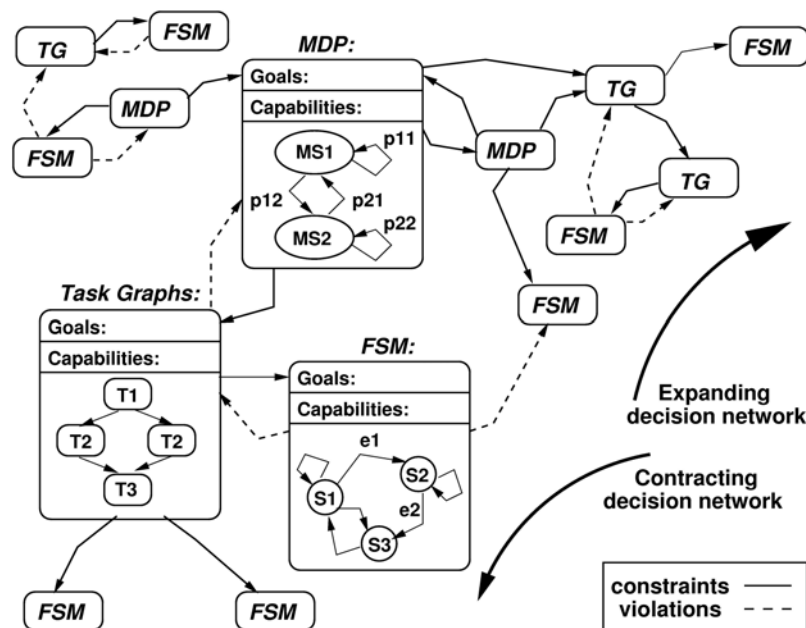


Fig. 3. Specification and distributed control concept.

The scheduling results of TGs are used to compute timing constraints for the reactive behavior of the bottom level DMs. As long as their reactive operation stays within these constraints, the overall timing requirement can be guaranteed. Bottom-up constraints express the amount of performance “violation” occurred at the physical level, and ought to be considered when re-computing the decision at the upper levels. The propagation of top-down and bottom-up constraints is performed continuously at runtime.

The networked decision making in Fig. 3 operates as follows. The reactive DMs employ input sampling and event driven decision making mechanisms, such as Finite State Machines (FSMs). The reactive behavior switches from one task to another depending on the occurrence of events. For example, the system switches from S_1 to S_2 , if event e_1 occurs. Each embedded unit executes its own controller. Hardware reconfiguration offers the capability of selecting online task implementations with different parameters, like speed, energy, memory, communication bandwidth, etc. Several individual controllers might decide to collaborate by building collectively a shared description that defines how the individual controllers use jointly any shared resources to achieve common objectives, such as the access over time (schedule) of vehicles accessing intersections. For example, the shared description can be a Conditional Task Graph (CTG) [31], which includes decisions specific to different operation conditions, such as various traffic loads. Note that CTG decision making is at the level of small local areas rather than individual controllers. Next, the CTG descriptions of the correlated areas are used to build description covering broader areas, such as those at the Markov Decision Process level. This step distributes the overall goals into goals for the individual sub-systems using information from the Markov Decision Process level.

The execution environment determines the structure of the distributed multi-semantic decision network as well as the node’s performance parameters. The network expands and shrinks depending on the nature of the application. Section 5 presents a case study for different decision making procedures.

3.2. Emergent Interactions

The sub-systems interact in complex ways. Interactions include not only exchanging data between the sub-systems, like in traditional concurrent tasks, but also modifying the goals, constraints, and other parameters of the participating sub-systems. For example, the travel speed of cars sets the timing constraint of the image-based tracking sub-system, so that certain field of view coverage is guaranteed. The traffic characteristics (e.g., average speed and average number of passing cars) determine the timing parameters of the traffic-lights, like cycle time, offset and split time. The timing deadlines for “special” cars, like police cars and fire trucks, act as hard timing constraints for both the image-based and sound-based tracking sub-systems.

Many interactions emerge for specific conditions. For example a high traffic load triggers the need to optimize the timing parameters of the traffic-lights, however, if the traffic load is very low then the default, periodic timing is sufficient, and hence there is no need of establishing interactions between the traffic-light and image-based tracking sub-systems. Also, the sound-based tracking sub-system needs to interact with the other sub-systems only in special conditions, such as if there is no direct sight of view or if the timing requirements are tight. In general, two sub-systems interact with each other whenever the activities of one have a significant impact on the behavior of the other sub-system. The significance of the impact is dynamic as it changes in time and space. Moreover, the identity of the interacting sub-systems is hard to predict a-priori due to the large number of potential interactions and the likelihood of adding new sub-systems to an existing CPS application. Incremental changes to the CPS functionality should be performed without recompiling the entire application, and without stopping the execution of the existing sub-systems. Adopting a conservative approach in implementing emergent interactions can lead to excessive communication overhead due to excessive synchronization and data exchange between sub-systems.

We suggest the concept of *interaction space* as the main mechanism for defining and implementing emergent interactions between sub-systems. Interaction spaces are inspired from energy fields: interaction spaces define procedures for (i) adding and removing energy from a field, (ii) propagating energy, and (iii) measuring the energy without changing the status of the space. In addition, global energy sources define overall interaction needs for the sub-systems. The routines of each CPS sub-system have defined a “footprint” in terms of their impact on the interaction space. The footprint can change dynamically. Emergent interactions between sub-systems are identified by following the footprint of the sub-systems, such as a sub-system that generates significant amounts of energy interacts with another sub-system that is sensitive to the energy. Once an interaction is identified, dedicated communication channels are set-up between the sub-systems to provide the needed synchronization and data exchange with a reduced communication overhead. Fig. 1 illustrates the concepts. The space has multiple orthogonal directions, which may or may not correspond to the Cartesian directions. The directionality of interactions results from the way energy is added (input) and removed (output) from the space. The influence of each action ripples through the space along each direction depending on a propagation resistance along that direction. The influences propagate more if the resistance is lower. The propagation resistance can change dynamically depending on the impact of the sub-system that adds or removes energy. Also, the influence can be eliminated in case a sub-system must have only a local influence.

A possible representation of an interaction space can be based on electrical fields. For our example, the cars entering the region can be interpreted as being analogous to electrical charge, thus adding energy to the electrical field corresponding to the interaction space. The cars leaving the region represent removing of energy from the field. The two tracking sub-systems measure the energy of the interaction space. Each route of the region defines a direction of the space. The traffic characteristics along a route, e.g., the average speed of cars, define the propagation resistance along that direction. The propagation resistance changes dynamically for each car or group of cars depending on the moving characteristics of the cars. Each traffic-light behaves as a switch that controls the moving of the charge particles along the existing routes, hence control the propagation resistance along directions. This representation can be used to identify emergent interactions between separate sub-systems and also between the components of the same sub-system. For example, if the traffic flow increases significantly along a certain route then the related energy increase affects a certain area of the region. The traffic-lights in that area must be coordinated with each other as they control the propagation resistance. Another example refers to the emerging interactions between video-based and sound-based tracking system. The coverage dropping of the video-based sub-system is described as a steep increase in the propagation resistance along the route. The resistance increase can be compensated through a similar decrease if the sound-based tracking sub-system can provide the necessary tracking. Hence, an emerging interaction is established between the two sub-systems. Note that the interaction space representation defines only a qualitative view of a CPS application, hence it is not used for quantitative reasoning, such as for detailed decision making and control at the level of the individual sub-systems.

4. Main Specification Constructs

The main constructs of the proposed goal-oriented specification notation are illustrated in Fig. 4.

The basic specification entity is a Decision Module (DM), which corresponds to local points, physical areas, zones, or larger regions. As explained in Section 3, each DM executes specific application-specific algorithms for sensing, processing, and actuation. This part is internal to each DM and is specified using an existing programming language, like C++ or Java. The focus of the proposed specification notation is on describing the local and global goals and the nature of interactions between modules. The compiler automatically produces the actual interaction mechanism, such communication mechanism and parameters. For simplicity, the presentation is descriptive instead of a formal one.

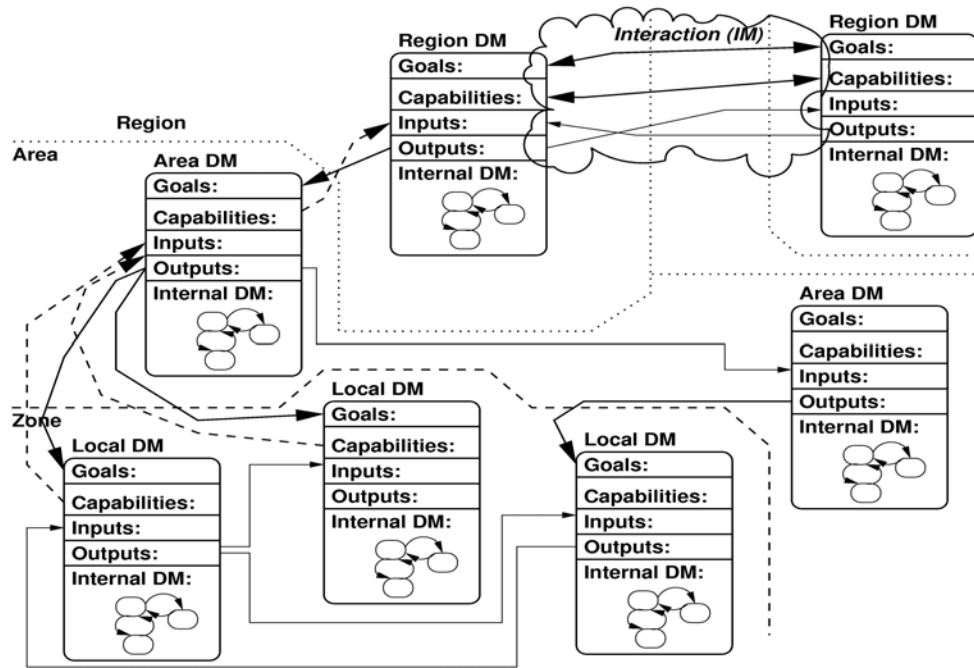


Fig. 4. Specification for scalable decision making.

Fig. 4 illustrates that a DM has four main parts: inputs, outputs, capabilities, and goals:

- **Inputs:** DM inputs either sample the pool of data associated to a physical region or are interaction data coming from other DMs. The acquisition semantics of an input can be continuous time, discrete time, or event driven. Moreover, inputs can refer only to certain facets of physical signals, e.g., voltage, current, phase, and frequency of a signal. The user can also specify aggregation of signals over time and space (integrals for continuous inputs and sums for discrete signals) and rate of change (sensitivities) with respect to time, space, or other signals (derivates for continuous inputs and differences for discrete inputs).
- **Outputs:** DM outputs relate to the outputs produced by the algorithm of a module. Outputs include physical actuation and control signals.
- **Capabilities:** Each module has a limited set of physical capabilities, such as the highest amount of service requests it can service, local memory, processing speed, energy resources, communication distance and bandwidth, and so on. These capabilities affect the ability of a node to maximize its local goals, and also percolate in influencing the quality of the optimal value that is reached for the overall application. For example, the low processing speed of a node or lack of memory might restrict the amount of alternatives that are locally analyzed, and affect the quality of the optimization process conducted by the node. The user can define attributes, like the upper and lower bounds and average value in time and space of a capability.
- **Goals:** The part indicates the goals to be optimized by each DM. Goals can be expressed either as maximizing or minimizing a cost function, or as a constraint satisfaction requirement, in which the goal expression must either exceed or fall below a threshold value. The cost function is defined over outputs and capabilities.

Components interact through interactions. The interactions are identified dynamically based on the interaction space concept discussed in Subsection 3.2. The following kinds of interactions can be established between DMs in the proposed notation:

- **Collaborative interactions:** Collaborative interactions are set up between DMs, which have non-conflicting (non-competing) goals. Modules interact with each other through inputs and outputs,

e.g., one module produces outputs to the pools that serve as inputs to another module. The goals and capabilities of the modules are not affected by this kind of interactions.

- *Competing interactions*: Such interactions are between DMs with competing goals, like optimizing the goal of one DM affects adversely the optimization of the other DM. Similarly to collaborative interactions, modules interact through their inputs and outputs but cannot change their goals or capabilities.
- *Guiding interactions*: Guided interactions are between DMs at consecutively higher levels in the semantic hierarchy. DMs at upper levels generate outputs that are used to set the goals of the modules at lower levels. This way the first modules are steering the goals and hence the behavior of the latter modules.
- *Enabling interactions*: Enabling interactions are information transfers from lower (reactive) semantic levels to the upper levels. A lower DM transmits information about its capabilities to an upper DM, so that the second can use this knowledge during a decision making that might affect the goals set for the lower DM through guided interactions. Enabling interactions are the information links through which upper DMs acquire knowledge about the actions that are ongoing in the real world.

In the proposed specification notation, the nature of interactions is dynamic and does not explicitly state the DMs participating to it. Instead, the user describes the conditions (thresholds) under which DMs start to interact with each other, like the change of an input, exceeding a threshold value, exceeding certain capabilities, etc. Interactions are described using Interaction Modules (IMs). IMs define any data transformation (such as data aggregation, filtering, etc.) that is needed if DMs of different formalisms are interacting. Similar to DMs, IMs have goals and capabilities. Moreover, the user must specify for each of the four interaction types, the formal mechanism (TGs, MDPs, etc.) used to resolve the interaction.

5. A Case Study and Insight into Specification Compiling

This section offers insight onto how the proposed model is used for efficient decision making and implementation. The discussion is based on a case study on coordinated traffic management in urban areas. The section discusses decision making strategies at different levels of abstraction, such as the traffic-light level, zone level, and area level. Different models and methods are used in each case. The section also highlights the interactions between these levels.

The traffic-light management sub-system has the goal to maximize the total number of vehicles passing through the managed area while the delay of a car is kept below the preset limit T . This constraint guarantees that cars on the less traveled directions are not indefinitely delayed by the vehicles moving on busier directions. Fig. 5a presents a simple street network. Arcs indicate the traffic flow. Traffic-lights (shown as small, black bubbles in the figure) are positioned at each intersection. The traffic-light parameters, e.g., cycle time, split time and offset, must be controlled by the sub-system to maximize the traffic flow.

The traffic management sub-system has a hierarchical structure so that decision making scales well over large geographical areas. The lower structures use semantic models that are based on physical parameters, such as the vehicle speed and inter-vehicle distance. The upper levels employ semantic models that provide an aggregated, more strategic view of the traffic flow. The hierarchy is as follows:

- Each traffic-light is controlled by a local DM called *Intersection Traffic Unit* (ITU). ITUs have reactive behavior based on FSMs with parameters optimized for the maximum traffic flow.
- Neighboring ITUs form *Zone Traffic Coordination Units* (ZTCUs) that coordinate the ITUs. Using information from the individual ITUs, ZTCUs compute the ITU parameters that offer the best

overall traffic flow for the set of coordinated ITUs. The ITU interactions are emergent: ITUs interact with each other only if cars are passing through the intersection.

- The ZTCUs of an area are coordinated by the higher-level DM called *Area Traffic Control Unit* (ATCU). ATCUs formulate a global, aggregated view about the traffic flow, and calculate the ZTCU parameters that optimize the traffic flow of the entire area.

Fig. 5 and Fig. 6 illustrate the first two lower levels of the multi-semantic, hierarchical decision making model: the FSM of the ITU and the Conditional Task Graph (CTG) of the ZTCU. The figures also show the interaction between the two levels. Each ITU implements a FSM with three states, *Red*, *Yellow*, and *Green*. DMs are executed by each embedded node. Assuming cyclic traffic signals, the arcs in Fig. 6b define the state transitions, which are taken after the FSM spends $T_{i,j}$ time units in the current state (the time is called split time). The time intervals are specific to each ITU and transition, and are fixed by the ZTCU to optimize the traffic flow through the zone. The schedule of the traffic signal repeats after a period called cycle time. The neighboring ITUs interact with each other to decide the delay between the monitored intersections (the delay is called offset). The decision making methods of the ITUs and ZTCUs are discussed next.

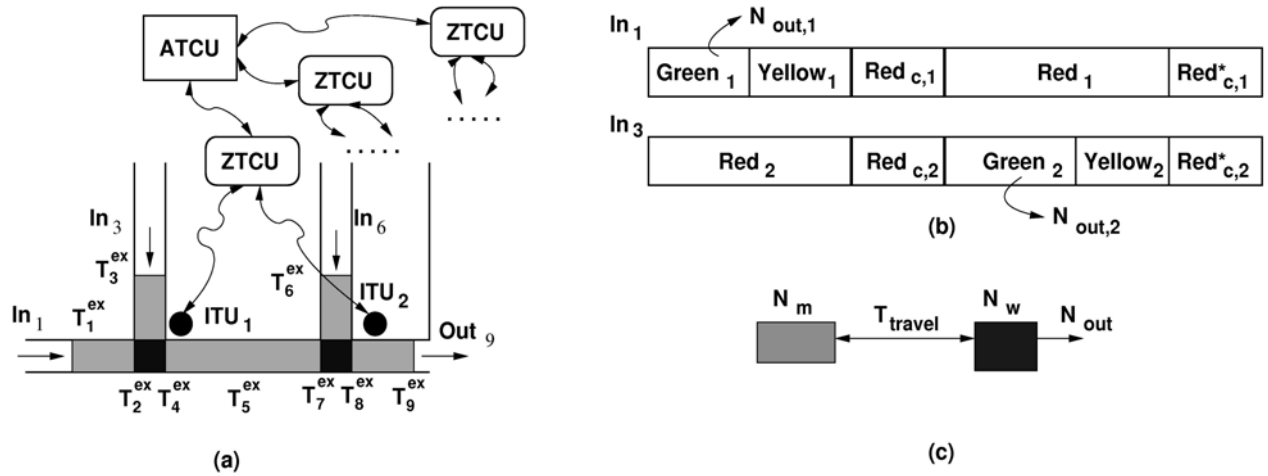


Fig. 5. ITU-level behavior: a) simple street network, b) time behavior of ITU and c) traffic parameter estimation.

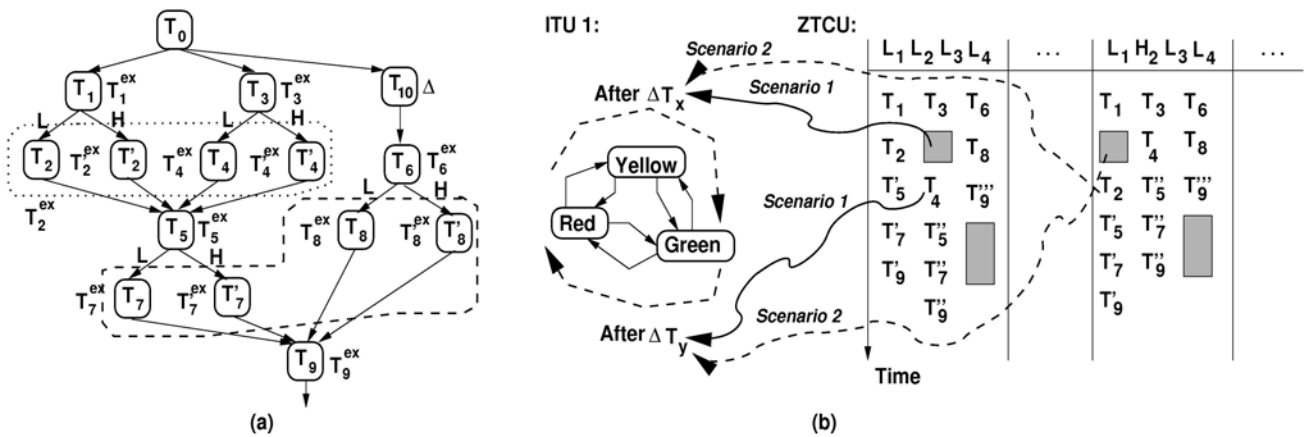


Fig. 6. FSM – CTG interaction: a) CTG for traffic in Figure 5(a) and b) ZTCU-level CTG scheduling table and interaction with ITU-level FSM.

Fig. 5 illustrates the physical-level behavior of the traffic-light sub-system. The figure indicates the sequence of states along the directions In_1 and In_3 in Fig. 5a. Fig. 5b shows the time behavior of the ITUs (for each traffic-signal). N_1 vehicles are passing along direction In_1 as long as the traffic-light is in state $Green_1$, and N_2 vehicles are passing along direction In_3 as long as the traffic light is in state $Green_2$. The discussion assumes that no cars are passing if the current state is either Red_i or $Yellow_i$. The buffer states $Red_{c,i}$ and $Red^*_{c,i}$ were introduced to guarantee that no vehicles are moving through the intersection during the switching to the states $Green_i$ along any of the two directions.

The moving of the vehicles is characterized by two main parameters, the average time T^{ex}_i required to traverse the route portion I , and the number of vehicles $N_{out,j}$ that move along direction j through an intersection. Using Fig. 5c, we estimate the number of vehicles that are moving along Direction In_1 and are going through the following intersection during the next state $Green$ of the traffic-light. The same reasoning can be applied for all other directions. Value N_w is the number of vehicles waiting in the queue of the next traffic-light. Value N_m is the number of cars arriving at the intersection along the observed direction. Time T_{travel} is the time that cars need to move between the two consecutive traffic-lights. Time Δ is the delay between the traffic-lights.

There are two scenarios: (i) if the N_m cars reach the next intersection before the N_w cars pass through, and (ii) if the cars reach after the waiting queue was cleared. In the first case, some of the N_m cars experience delays, while in the second case the in-coming cars are not delayed. The two cases are analyzed next. The time needed for all N_w vehicles to pass through the intersection is equal to $T_{through} = N_w/n_L + \Delta$, where n_L is the number of vehicles passing per unit of time after vehicles were stopped. The first case corresponds to the conditions $T_{travel} \geq T_{through}$ and $T_{through} \leq T_{Green}$. Otherwise, the second case is performed. The two cases are analyzed next.

Case (i): The value $N_{left} = N_w - n_L (T_{travel} - \Delta)$ is the number of cars that are still in the queue when the pool of N_m cars reaches the intersection. Two situations can be identified. First, if $n_{In} > n_L$, where n_{In} is the number of cars arriving along the observed direction in a unit of time then all N_m cars are delayed. Hence, the number of cars passing through the intersection is $N_{out} = \min (N_w + N_m, n_L T_{Green})$. The number of cars remaining in the waiting queue, hence representing the value N^{next}_w for the next traffic-light period, is equal to $N^{next}_w = \max (0, N_w + N_m - n_L T_{Green})$. Second, if $n_{In} < n_L$ then only some of the N_m cars are delayed in the waiting queue of the traffic-light, while the following cars pass through without delay before the traffic-light changes states. The time required to clear the waiting queue is equal to $T_{clear} = N_w/(n_L - n_{In})$. If $T_{clear} \geq T_{Green}$ then only a fraction of the waiting cars can pass through before the traffic-light changes states. Hence, $N_{out} = \min (N_w n_L/(n_L - n_{In}), n_L \times T_{Green})$ and $N^{next}_w = N_w + N_m - N_{out}$. If $T_{clear} < T_{Green}$ then the number of passing-through cars is equal to $N_{out} = N_w n_L/(n_L - n_{In}) + \min (n_H (T_{Green} - T_{clear}), N_m)$. Value n_H is the number of cars passing through per unit of time, if the cars are not placed in the waiting queue at the intersection. The cars remaining in the waiting queue for the next period are $N^{next}_w = N_w + N_m - N_{out}$.

Case (ii): The time required to move the N_w vehicles through the intersection is equal to $T_{clear} = N_w/n_L$. If $T_{clear} \geq T_{Green}$ then only a fraction of the N_w vehicles are moving through, hence, $N_{out} = n_L T_{Green}$. The number of cars that are placed in the waiting queue for the next traffic-light period is equal to $N^{next}_w = N_w - n_L T_{Green} + N_m$. If $T_{clear} < T_{Green}$ then $N_{out} = N_w + \min (N_m, n_H (T_{Green} - T_{delay} + \Delta))$, and $N^{next}_w = N_w + N_m - N_{out}$.

The analysis shows that the ZTCU-level behavior experiences multiple modes of operation depending on the traffic conditions. To represent the conditional behavior, the ZTCU functionality is expressed as a Conditional Task Graph (CTG) [31], which describes the activities performed by the moving vehicles as they pass through the zones controlled by the ITUs. The activities are selected depending on conditions such as the traffic load. Hence, CTGs capture all situations in which a car can be involved. The analysis method assumes that one car enters the corresponding zone from each of its

directions. For example, in Fig. 5a, one car enters along directions In_1 , In_2 , and In_3 . The related activities include passing through the intersection, moving to the next intersection, and so on. We assumed single-lane roads. This implies that two cars coming through the intersections ITU_1 and ITU_2 cannot move simultaneously.

Fig. 6a presents the CTG for the zone in Fig. 5a. There is a parallel thread for each direction of the route. The graph presents the activities for moving a car through the zone. Tasks T_1 and T_2 describe the traveling of a vehicle entering the zone from the left, tasks T_3 and T_4 model the moving of a vehicle coming from the top-left road, tasks T_6 and T_7 represent a car moving from top right, and so on. Each of the activities is characterized by a typical execution time T_k^{ex} that is estimated based on the data collected locally by the sensors at the traffic-lights, e.g., the video-based and sound-based tracking sub-systems. The execution times differ for different traffic scenarios, such as slow traffic (branch L_i) and heavy traffic (branch H_i). The semantics of CTGs states that one and only one of the two branches L_i and H_i originating at a conditional node is executed for each traversal of the graph. This semantics defines a conditional (multimode) behavior. Certain road sections (shown as dark areas in the figure) act as shared resources for two or more cars, while other sections are dedicated resources. In the graph, the tasks sharing the same resources are grouped in the two clusters marked with dashed and dotted lines. In addition, the execution time Δ of the dummy task T_{10} defines the offset time between the two traffic-lights. Note that the parallel threads through the CTG interact through the number N_w of cars waiting before a traffic-light. These cars were “placed” in the waiting queue by the previous thread. Hence, finding fast schedules requires not only minimizing the total execution time of the current flow but also controlling the queue lengths.

ZTCU computes the optimal scheduling of the tasks T_i to maximize the number of cars that move through the zone over a period of time, and/or minimize the overall time taken to cars to move through the zone. Different traffic conditions determine the values of the task attributes, like the execution time T_i^{ex} . The schedule lengths are used to compute the optimal cycle and split times of each ITU, and also find the delay values between successive traffic-lights. This step is discussed later in the section.

Each schedule corresponds to traffic conditions described by the conditional behavior of the CTG. For example, if the traffic is slow on Segment 1 then the branch labeled L is taken after task T_1 is completed, hence task T_2 is performed. If the traffic load is high then branch H is selected, and task T_2' is pursued. Tasks T_2 and T_2' can never be performed simultaneously as they correspond to mutually excluding conditions (the traffic intensity cannot be simultaneously low and high). This generates 16 different situations for the case in Fig. 5, such as schedule $S_{(L,L,L)}$ with the total time $T_{(L,L,L)}^{tot}$ for the situation in which the traffic load is light for all three intersections. The concrete task scheduling for CTGs can be computed with scheduling algorithms similar to the one in [31].

Fig. 6b presents the interaction between the FSMs of the ITUs at each traffic-light and the CTG scheduling at the ZTCU. The structure of the scheduling table computed online by the ZTCU is shown in the figure. The table includes the schedules for the scenarios that can emerge in traffic. The table in the figure presents two situations: one in which traffic is light along all directions, and the second in which the traffic along Direction 2 is high, and it remain low along the others. The boxes in gray indicate idle times along a direction. The information in the CTG scheduling table is mapped down onto the FSMs of each individual traffic-light. For example, for Scenario 1 (in which the traffic intensity is low along all directions), the time required for performing task T_3 plus the following idle time generate the timing constraint ΔT_x , which states that after that amount of time the FSM must be in state *Green*, thus allowing traffic to progress along Direction 2. Similarly, for Scenario 2, the time for task T_1 plus the following idle time define the constraint ΔT_y , which states that the traffic signal must be in state *Red* after time ΔT_y , which allows traffic to progress along Direction 1. As long as each traffic signal meets its local timing constraints (ΔT_x and ΔT_y), the behavior at the ZTCU-level meets the traffic schedules specified in the ZTCU table.

Fig. 7a introduces the more abstract decision making level, called Area Traffic Coordination Unit (ATCU). Its procedure is based on Markov Decision Processes (MDP). The figure also presents the connection of ATCU to the lower-level CTGs. For an estimated number of vehicles that pass through the area, each ZTCU calculates the optimal activity schedules for the predicted traffic scenario. If the timing delay T^{ZTCU}_i of the ZTCU schedule i exceeds the timing constraint for traversing the area then the traffic-light controllers are adjusted so that the new traffic scenarios produce schedules that meet the constraint. Figure 7(b) offers a pictorial description of Continuous-Time Markov Chains (CTMCs):

$$CTMC = (S, A, A(i), p, K, r) \quad (1)$$

where S is the set of states, A is the action set, and $A(i)$ are the actions associated to state i in S . $p(i,j,a)$ is the transition rate if action a is chosen for transitioning from state i to state j . K is the number of reward criteria, and $r_k(i,a)$ is the reward rate if action a is selected for state i . x_i is the steady-state probability of state i .

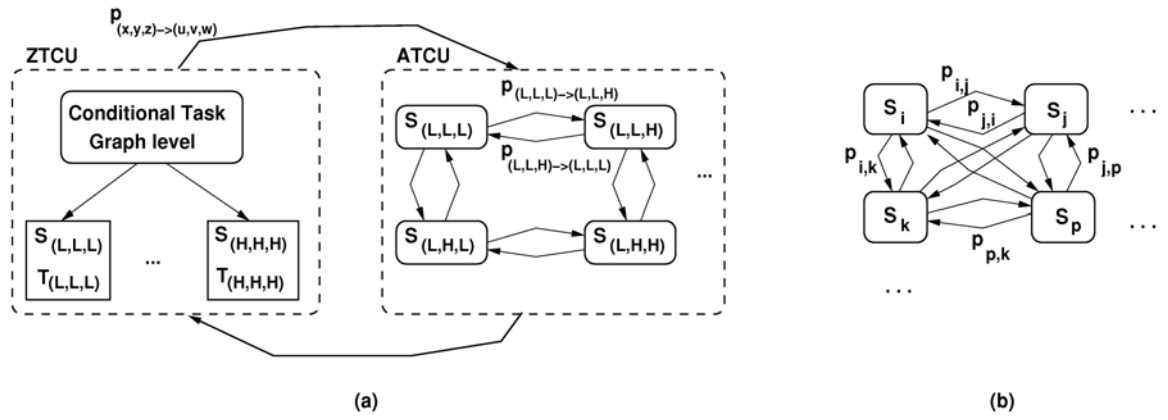


Fig. 7. CTG - MDP interaction: a) interaction between ZTCU-level CTG and ATCU-level MDP and b) MDP definition.

CTMC are employed at the ATCU-level to control the moving of vehicles over time so that the traffic load is maximized. The interaction between ZTCUs models the fact that the number N_{out} of outgoing vehicles for a ZTCU is equal to the number N_m of ingoing vehicles for the next ZTCU. Hence, for the situation in Figure 8 (a), $N_{out,1} = N_{m,2}$ and $N_{out,2} = N_{m,3}$. In addition, the behavior of each ZTCU is expressed as a separate CTMC, as presented in Fig. 8 b.

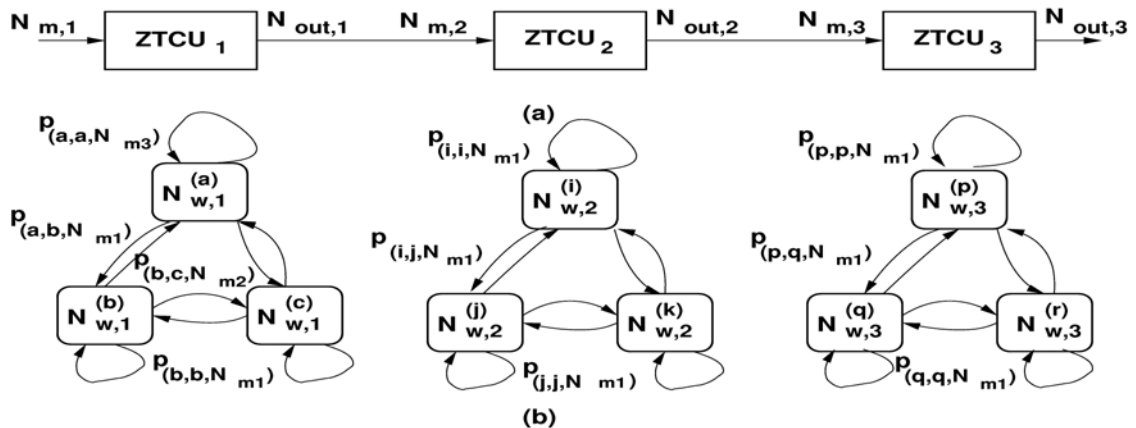


Fig. 8. MDP-level decision making: a) sequence of three ZTCUs and b) interaction of the related MDPs.

The states of the CTMC correspond to the number $\langle N_{w1}, N_{w2}, \dots, N_{wk} \rangle$ of vehicles waiting in the k queues of the zone monitored by each ZTCU. The transition rate $p_{ij, N_m, k}$ between any pair of states i and j depends on the number $N_{m, k}$ of vehicles entering the zone. Hence, the values $N_{m, k}$ act as actions for the CTMC. The transition rate is equal to the frequency of having N_m incoming vehicles. The reward criteria are the execution time required for the vehicles to reach the end of the zones (the execution time corresponds to the lengths of the schedules computed by the ZTCU) and the number N_{out} of vehicles that exit the zone. The following set of equations is formulated for each ZTCU I :

$$\sum_{\forall N_m} p(i, i, N_m) x_{i, N_m} - \sum_{j \in S_I \forall N_m} p(i, j, N_m) x_{i, N_m} = 0, \forall j \in S_I, \quad (2)$$

$$\sum_{i \in S_I} \sum_{N_m} x_{i, N_m} = I, \forall i \in S_I, \forall N_m \quad (3)$$

$$x_{i, N_m} \geq 0, \forall i \in S_I, \geq 0, \forall N_m \quad (4)$$

The set of equations must be solved so that the total number of vehicles output at the last ZTCU ($ZTCU_3$ in Figure 8(a)) is maximized:

$$\max \sum_{i \in I} \sum_{N_m} N_{out}(i, N_m) x_{i, N_m} \quad (5)$$

The steady-state probabilities are computed by solving the set of equations. The probabilities are utilized to calculate the percentage of time that each alternative is used at the ZTCU-level.

6. Conclusions

Massively distributed embedded systems are rapidly emerging as a breakthrough concept for many modern applications. However, providing efficient and scalable decision making capabilities to such systems is currently a significant challenge. The paper proposes a model and specification language to allow automated synthesis of distributed controllers, which implement and interact through models of different semantics. Scalability of descriptions is realized through defining the nature of interactions that can occur among decision modules while leaving to task of optimally implementing these interactions by the execution environment. The notation defines the operation goals of each sub-system (e.g., the criteria to be maximized or minimized during operation) and the physical capabilities of a module to achieve a certain goal. Different interaction types are introduced depending on the way sub-systems influence each other their goals and capabilities.

The paper refers to a case study to illustrate the model and the related decision making steps. Compared to similar work, the proposed model and specification notation differs in that they focus on optimal goal satisfaction in massively distributed systems and not on algorithmic descriptions. Therefore, the language is orthogonal to existing notations as it concentrates on the interactions between groups of nodes, or nodes and environment and less on the behavior of the individual nodes. This simplifies specification and helps scalable decision making.

References

- [1]. C. Batini, E. Bertini, M. Comerio, A. Maurino, G. Santucci, Visual languages and quality evolution in multichannel adaptive information systems, *Journal of Visual Languages & Computing*, Vol. 18, Issue 5, 2007, pp. 513-522.
- [2]. A. Bakshi, V. Prasanna, J. Reich, D. Larner, The Abstract Task Graph: A methodology for architecture-independent programming of networked sensor systems, in *Proceedings of End-to-End, Sense-and-respond Systems, Applications and Services*, Seattle, Washington, 2005, pp. 19-24.
- [3]. L. Camara, E. Jungert, A visual query language for dynamic process applied to a scenario driven environment, *Journal of Visual Languages & Computing*, Vol. 18, Issue 3, 2007, pp. 315-338.
- [4]. R. Soma, A. Bakshi, V. Prasanna, A semantic framework for integrated asset management in smart oilfields, in *Proceedings of the IEEE Symposium on Cluster Computing and Grid*, Rio de Janeiro, Brazil, May 2007, pp. 119-126.
- [5]. K. Whitehouse, F. Zhao, J. Liu, Semantic streams: a framework for declarative queries and automatic data interpretation, Technical Report, *Microsoft Research*, MSR-TR-2005-45, 2005.
- [6]. A. Bakshi, V. Prasanna, A. Ledeczi, Milan: A model based integrated simulation framework for design of embedded systems, *ACM Sigplan Notices*, Snowbird, Utah, 2001, pp.82-93.
- [7]. K. Passino, Biomimicry for Optimization, Control, and Automation, *Springer*, 2005.
- [8]. E. Lee, The problem with threads, *IEEE Computer*, Vol. 39, Issue 5, 2006, pp. 33-42.
- [9]. H. Sutter, J. Larus, Software and the concurrency revolution, *ACM Queue*, Vol. 3, Issue 7, 2005, pp. 54-62.
- [10]. N. Zeldovich, A. Yip, F. Dabek, R. Morris, D. Mazieres, F. Kaashoek, Multiprocessor support for event-driven programs, in *Proceedings of the USENIX Annual Technical Conference (USENIX' 03)*, San Antonio Texas, June 2003.
- [11]. G. Papadopoulos, F. Arbab, Coordination models and languages, *Advances in Computers - The Engineering of Large Systems*, Academic Press, Vol. 46, 1998, pp. 329-400.
- [12]. E. Lee, Cyber physical systems: Design challenges, Technical Report No. UCB/EECS-2008-8, *University of California at Berkeley*, 2008.
- [13]. G. Papadopoulos, A. Stavrou, O. Papapetrou, An implementation framework for software architectures based on the coordination paradigm, *Science of Computer Programming*, Vol. 60, Issue 1, 2006, pp. 27-67.
- [14]. S. Ahuja, N. Carrieri, D. Gelernter, Linda and friends, *IEEE Computer*, Vol. 19, Issue 8, 1986, pp. 26-34.
- [15]. D. Gelernter, D. Kaminsky, Supercomputing out of the recycled garbage: Preliminary experience with Piranha, in *Proceedings of the ACM International Conference on Supercomputing*, Washington DC, 1992, pp. 417-427.
- [16]. A. Rowstron, A. Wood, Bonita: A set of tuple space primitives for distributed coordination, in *Proceedings of the International Conference on Systems Sciences*, Waikoloa, Hawaii, Vol. 1, 1997, pp. 379-388.
- [17]. N. Minsky, J. Leichter, Law-governed Linda as a coordination model, in *Object-Oriented Models and Languages for Concurrent Systems*, LNCS 924, *Springer Verlag*, 1994, pp.125-145.
- [18]. T. Kielmann, Designing a coordination model for open systems, in *Proceedings of International Conference on Coordination Models, Languages and Applications*, LNCS 1061, *Springer Verlag*, 1996, pp. 267-284.
- [19]. J.-P. Banatre, D. Le Metayer, Gamma and the chemical reaction model: Ten years after, in *Coordination Programming: Mechanisms, Models and Semantics*, *World Scientific*, 1996, pp.1-39.
- [20]. I. Foster, Compositional parallel programming languages, *ACM Transactions on Programming Languages and Systems*, Vol. 18, Issue 4, 1996, pp. 454-476.
- [21]. M. Cole, Algorithmic Skeletons: Structured Management of Parallel Computation, *MIT Press*, 1989.
- [22]. D. B. Skillicorn, Towards a higher level of abstraction in parallel programming, in *Proceedings of Programming Models for Massively Parallel Computers*, Berlin, Germany, 1995, pp. 78-85.
- [23]. L. de Alfaro, T. Henzinger, Interface-based design, *Engineering Theories of Software-Intensive Systems*, NATO Science Series: Mathematics, Physics, and Chemistry, Vol. 195, *Springer*, 2005, pp. 83-104.
- [24]. T. Henzinger, B. Horowitz, B. C. Kirsch, Giotto: A time-triggered language for embedded programming, *Proceedings of IEEE*, Vol. 91, Issue 1, 2003, pp. 84-99.
- [25]. T. Henzinger, S. Matic, An interface algebra for real-time components, in *Proceedings of the Annual Real-Time and Embedded Technology and Applications Symposium*, San Jose, California, 2006, pp. 253-266.
- [26]. E. Wandeler, L. Thiele, Real-time interfaces for interface-based design of real-time systems with fixed priority scheduling, in *Proceedings of International Conference on Embedded Software (EMSOFT)*, Jersey City, New Jersey, 2005, pp. 80-89.
- [27]. W. Johnston, Advances in dataflow programming languages, *ACM Computing Survey*, Vol. 36, Issue 1,

2004, pp. 1-34.

- [28].C. Zhang, A. Bakshi, V. Prasanna, ModelML: a markup language for automatic model synthesis, in *Proceedings of the IEEE Conference on Information Reuse and Integration*, Las Vegas, Nevada, 2007, pp. 317-322.
- [29].R. Newton, M. Welsh, Region Streams: Functional macroprogramming for sensor networks, in *Proceedings of the International Workshop on Data Management for Sensor Networks*, Toronto, Canada, 2004, pp. 78-87.
- [30].K. Gummadi, O. Gnawali, R. Govindan, Macro-programming wireless sensor networks using Kairos, in *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, 2005, pp. 126-140.
- [31].P. Eles, A. Doboli, Z. Peng, P. Pop, Scheduling with bus access optimization for distributed embedded Systems, *IEEE Transactions on VLSI Systems*, Vol. 8, No. 5, 2000, pp.472-491.

2009 Copyright ©, International Frequency Sensor Association (IFSA). All rights reserved.
(<http://www.sensorsportal.com>)

International Frequency Sensor Association



International Frequency Sensor Association (IFSA) is a professional association, created with the aim to encourage the researches and developments in the area of quasi-digital and digital smart sensors and transducers.

IFSA Membership is open to all organizations and individuals worldwide who have a vested interest in promoting or exploiting smart sensors and transducers and are able to contribute expertise in areas relevant to sensors technology.

More than 500 members from 62 countries world-wide including ABB, Analog Devices, Honeywell, Bell Technologies, John Deere, Endevco, IMEC, Keller, Mazda, Melexis, Memsis, Motorola, PCB Piezotronics, Philips Research, Robert-Bosch GmbH, Sandia Labs, Yokogawa, NASA, US Navy, National Institute of Standard & Technology (NIST), National Research Council, etc.



For more information about IFSA membership, visit
<http://www.sensorsportal.com>

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In addition, some special sponsored and conference issues published annually.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 6-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: <http://www.sensorsportal.com/HTML/DIGEST/Submission.htm> Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit: http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2008.pdf

**e-Impact Factor 2008:
205.767**



Subscription 2009

*Sensors & Transducers Journal (ISSN 1726-5479)
for scientists and engineers who need to be
at cutting-edge of sensor and measuring
technologies and their applications.*

*Keep up-to-date with the latest, most significant
advances in all areas of sensors and transducers.*

**Take an advantage of IFSA membership
and save **40 %** of subscription cost.**

Subscribe online:

http://www.sensorsportal.com/HTML/DIGEST/Journal_Subscription_2009.htm

e-mail: editor@sensorsportal.com

tel. +34 696 06 77 16