computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

FROM DISTRIBUTED MEASUREMENT SYSTEMS TO CYBER-PHYSICAL SYSTEMS: A DESIGN APPROACH

Domenico Luca Carnì [1], Domenico Grimaldi [1], Francesco Lamonaca [2], Libero Nigro [1], Paolo Francesco Sciammarella [1]

[1] Dept. of Informatics, Modelling, Electronics and System Eng., University of Calabria, 87036 Rende – CS, Italy
{dlcarni, d.grimaldi, p.sciammarella}@dimes.unical.it, libero.nigro@unical.it
[2] Dept. of Engineering, University of Sannio, 82100 Benevento – BN, Italy, flamonaca@unisannio.it

Abstract: Recently the technological advances have allowed the introduction of new standards to model and monitor control systems. The classical distributed measurement systems (DMSs) have been enriched with the use of a large number of new measurement devices, increasing the amount of produced data. New paradigms for their management are needed, that coupled with actuators, enable the integration of DMSs into more complex systems, known as Cyber-Physical Systems (CPSs). CPSs application field includes different areas such as health care, power management in smart micro grids, and, more in general, management and execution of time-dependent event-driven systems.

This paper proposes a framework that allows the modelling, analysis and the implementation of CPSs. In particular, the proposed framework introduces an original feature called model continuity that offers the possibility of using a same model for both simulation phase and real-time execution phase. *Copyright © Research Institute for Intelligent Computer Systems, 2017. All rights reserved.*

Keywords: Distributed Measurement Systems, Multi-agent systems, Cyber-Physical Systems, Model continuity.

1. INTRODUCTION

Nowadays new technologies such as Internet-of-Things [1] coupled with sensors miniaturization, offer the possibility to enrich Distributed Measurement Systems (DMSs) with a large number of measurement devices, that can be located everywhere and easily controlled, through internet connection. This has led not only to increase the productivity of the industries [2], or ameliorate the management of smart micro grids [3], but has made available a huge amount of measurements data.

To handle, process and use this new huge amount of data in an efficient way, new paradigms are needed. Big data, Industry 4.0 and health care, are just few examples of research areas, that offer new sophisticated techniques to upgrade the classical data management systems used in DMS, leading to new kind of systems, known as Cyber-Physical Systems (CPSs) [4]. CPSs incorporate and extend DMSs. CPSs integrate a computational part, called cyber, constituted by algorithms implemented through software, with physical elements, such as sensors and actuators, that not only acquire data from the environment, but also try to change it, using a general communication network [5, 6]. The main challenge of CPSs arises from system integration [7], due to components heterogeneity and their interactions. Among the emerging issues, there is the tying together two different notions of time: discrete, for the computing units of the cyber-part, and Newtonian, associated to the physical process that needs to be supervised, because it can lead problems, such as system reaction loss to events of interest [8], or time constraints not satisfied [9].

To overcome this issue this paper proposes a framework that offer a methodology to model, analyze and implements CPSs, based on multi-agent programming paradigm [10]. The proposed architecture includes a unique feature called model continuity, that consists in the possibility of transitioning a same model from simulation down to real-time execution, changing only the engine and other software entities used to manage time. This allows to firstly evaluate the system behavior, later the impact of the data transmission delays of the network, on the fulfillment of the time constraints, avoiding problems and timed inconsistencies, due to model approximations introduced during the transition from one development phase to other [11].

This paper is organized as follows. Section 2 introduces the basic concepts of the adopted agent-based architecture, emphasizing details for its use in CPSs development, concerning the interfacing between the cyber and physical part. Section 3, presents a case study, in the context of real-time monitoring and management of a smart micro-grid. Finally, in Section 4 the conclusions are drawn.

2. AGENT-BASED CONTROL FRAMEWORK

The proposed agent-based control framework is founded on the main concepts that are: actor, message and action.

Actors are thread-less agents with an internal state and behavior patterned as a finite state machine (Fig. 1). To communicate each other, actors use asynchronous message passing. Messages are pure software entities used only to promote sociality among actors.

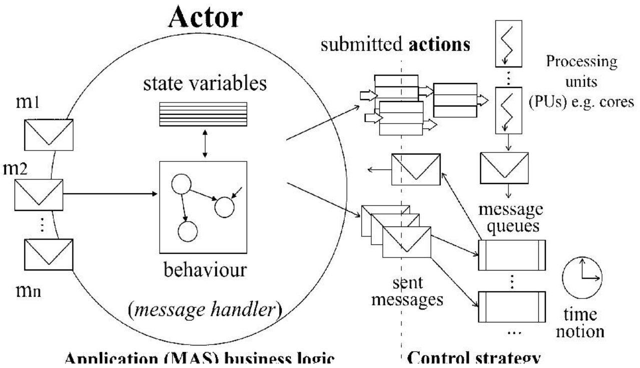

Fig. 1 – Basic actor organization

When an actor receives a message, it can:

- reply by sending another message to another known actor;
- change its status or update someone of its variables;
- submit actions.

Actions are tasks which consume time, real or simulate according to the software lifecycle phase, constituted by a self-contained code block, that not interact with the internal data of the submitter actor, avoiding consistency problems. Actions and messages are conceptually and practically different entities, in fact:

- actions are executed in a parallel way, according to a physical parallelism, exploiting the concept of processing units (that can be core, thread, fake object, etc.);
- messages are processed one at a time in an interleaved and atomic way, according to a cooperative concurrency. They are not task, so it is assumed that their processing does not consume time.

A set of actors allocated on a same computing node, constitutes a Logical Process (LP), and are managed by a control structure called control machine. A control machine is in charge of:

- to supervise the message exchange among actors;
- to give a global notion time to the overall system;
- to assign the processing units available for actions execution.

To support model continuity, only actions and control machine, since hybrid components that include explicitly time, need to be reified according to specific simulation or real-execution phase, associated with the software development. The framework implemented by using Java and Jade [12], offers libraries with all the different implementation of the components [11, 13].

2.1 GATEWAY

The framework offers also a middleware layer called environmentGateway or Gateway, that connects actions related to the cyber part, with the measurement devices and instrumentation of the physical part. It exposes the basic read/write operations useful to connect the application layer with devices, hiding all the implementation details. Moreover, sensors and actuators have been interfaced through the use of commercial-off-the-shelf (COTS) hardware, such as Arduino or Raspberry Pi, each one managed by a thread of an In/Out layer.

In previous work [13, 14] has been used the RS232 standard interface as a communication network. In this paper, instead, a new module is proposed to extend the gateway component, with a component that makes wireless communication available, using Internet and the Message Queue Telemetery Transport (MQTT) protocol [15]. The introduction of this feature allows to fully exploit the potential of the Internet connection, both to better distribute the system in the environment and also to put in communication devices that refer to different standards.

2.2 MQTT GATEWAY EXTENSION

The MQTT is a Client-Server publish/subscribe messaging transport protocol, designed for networks with low bandwidth and high-latency, as Internet. It is widely used for real-time data transmission, due to: (i) the messages exchanged, and (ii) the reduced size of header and payload.

MQTT consists of three main components:

- a publisher client, that is the source of the required data;
- a subscriber client, that is the receiver of the transmitted data;

- a broker server, used as a mediator, that connects the clients and allows the data exchange.

Each publisher client produces data, assigns them a topic label and sends them to the broker. Each subscriber client registers itself to the broker, specifying the topic name of the information requested. When the broker receives an incoming data, it checks the topic and sends it to all registered clients waiting for the given data [16].

The framework Gateway component implements a module based on the MQTT protocol, because, through the use of Internet and wireless technologies, it:

- introduces greater flexibility in the application design, allowing the space displacement of the different measurement devices and actuators;
- hides the complexity of the underlying communication network, offering a standard interface and methods to send data in a simply way.

Furthermore, the protocol provides three different levels of quality of service (QoS) to the programmer, to set, according with the application goals, an efficient communication, i.e. small delays in sending data packets, or reliable communication, which consists of the certainty that the data reaches the receiver, exploiting an acknowledgement mechanism.

Fig. 2 shows the use of the publish/subscribe mechanism, clarifying that cyber and physical parts act as publisher and subscriber, because the data flow is bidirectional and both produce and receive data, that are commands for actuators or measured values for sensors.

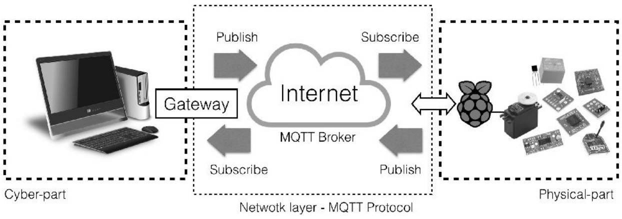

Fig. 2 – Functioning of MQTT protocol

3. CASE STUDY

The proposed case study consists in the optimization of the power management in a smart micro-grid, acting as a home automation system.
The system components are:

- a local producer of energy, e.g. a photovoltaic panel;
- a collection of appliances or power loads.

The goal is to schedule commands and connect or detach loads to the power grid, in a given time interval, optimizing the consumption of the generated power.

The system inputs are:

- the power available reference, that acts as a threshold curve, generated by the provider;
- the power request by each load, that is modeled as a consumption curve.

The generated output is the algorithm scheduler decision used for loads activation/deactivation.

3.1 DATA CONFIGURATION

The details and properties associated to the different loads are summarized in Table 1.

Table 1. Load parameters.

Priority π	Time [s]/ Power Request [W]/ Active Lamps	Scaling factor	Cyclic load	Relays x Lamp
4	90/500/2 85/750/3	1	Yes	(1x2) (1x1)
3	50/500/2 170/1000/4 40/500/2	2	No	(1x1) (1x1) (1x2)
2	220/1000/4	2 or 4	Yes	(1x1) (1x1) (1x2)
1	100/750	1	No	(1x1) (1x2)

Loads are modelled through:

- a priority level π, ranging from 1 to 4, where 4 is the highest;
- a consumption curve, modeled as a stepwise function, that indicate the power request in a given time interval;
- a scaling factor that indicates if the load power request can be modularized;
- a Boolean value which specifies whether the load needs to restart its execution after the end.

Other parameters refer to the execution of the real experiment. In particular, each load will be modeled not using real appliances, but through the use of lamps. So Table 1 shows also:

- the total number of active lamps that models each load;
- the number of relays used to drive the lamps.

The generated power signal available has been modeled through a stepwise function, whose features are summarized in Table 2.

Table 2. Generated power signal steps.

Time interval [s]	Power [W]	Time interval [s]	Power [W]
20	400	15	2280
30	800	35	2330
20	1100	10	2390
40	2050	15	1760
130	2600	10	1540
10	700	5	1030
5	1000	15	470

3.2 METHODOLOGY TO DEVELOP A CPS

To develop the system, the proposed methodology includes four different phases:

- a modeling phase in which the different components of the system are modeled as a set of interacting agents;
- a property analysis phase or simulation phase, used to check the properties and the system behavior, simulating the connection with physical devices;
- a preliminary execution phase that is an intermediate stage between simulation phase and real-execution phase, in which time is not simulate but real. It is useful to estimate the impact of the messages processing overhead on the system behavior, verifying if the time constraints continue to be satisfied, including network delays. The physical devices are not yet connected, but simulated through the use of software components;
- a real-time execution phase in which all the hybrid components of the model are reified in their real part, so time is real and the physical devices are connected with the cyber-part.

According to model continuity all these phases are sequentially executed, using the same model developed in the first phase, changing only the control structure and the actions kind and body-code.

3.3 SYSTEM EXAMPLE

According to CPS paradigm, the system assembled consists in two different parts: a cyber-part and a physical part.

The cyber part is constituted by a set of load agents, each one maps the functioning of a specific physical load, a scheduler agent that implements the algorithm useful to select a subset of load for the execution, and a gateway useful to interface the two part.

Since in simulation and preliminary execution phases loads are not connected yet, the physical part is modeled through the use of an agent that provides step by step the sample of the generated power, and a set of agents that simulate loads On/Off functioning.

For experimentation purposes, in order to test the model continuity, the proposed case study has been assembled in a context of academic laboratory. Physical entities have been connected with the cyber part, making an overall system as shown in Fig. 3.

An Arbitrary Waveform Generator Sony/Tektronix AWG2021, driven by LabVIEW, through the General Purpose Interface Bus (GPIB), has been used to generate an analog signal, that models the generated power reference. To carry out the digital conversion of the signal, an Arduino board has been added to the system.

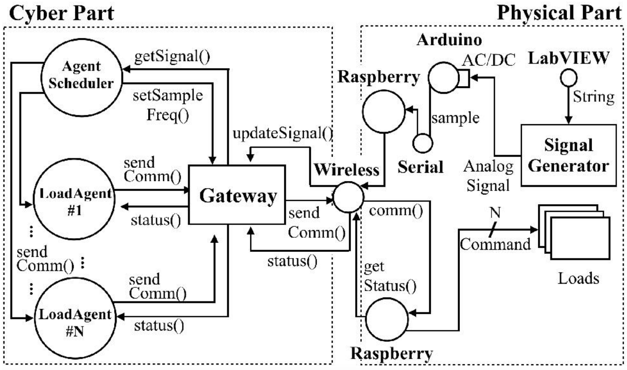

Fig. 3 – Overall system configuration

Two Raspberry Pi3 Model B were also included. One of them, connected through serial communication with the Arduino board, is in charge to send the sample of the power signal to the cyber-part, the other is used to manage load, using the pins of its GPIO. The data exchange entrusted to the Gateway MQTT module, exploiting wireless technologies, allows to obtain a greater system design flexibility. In fact, it enables a physical separation between the part related to the acquisition and sampling of the power signal, and the other related to load management and monitoring, potentially placing them in different environments. The Digital Oscilloscope Tektronix TDS220, as shown in Fig. 4, is connected in parallel to the output channel of the AWG2021 in order to visualize the trend of the output signal, evaluating the correctness of its generation.

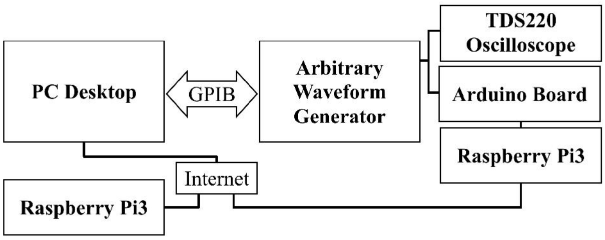

Fig. 4 – Measurement stand

3.3 LOADS SUBSYSTEM

The loads subsystem is composed by: (i) 2 relay board with 8 voltage channels characterized by rating of 10A at 250 V and 125 V AC, and 10A at 30 V and 28 V DC, managed by 8 digital pins adapted to work with the Raspberry output operating voltage, and (ii) 12 Standard lamps 250 W.

A single load is constituted by a pre-established connection of a finite number of lamps. The digital pins of the Raspberry board feed the digital pins of the relay modules, allowing the turning on and off of the loads.

3.3 SCHEDULER ALGORITHM

The goal of the scheduler agent is to find a valid schedule, that selects a subset of loads ready for execution. They are chosen so that their total power required, does not exceed the generated power threshold produced by the photovoltaic panel.

The scheduler agent works with a period of 1s, using both predictive and reactive mechanism.

The first one uses a predicted signal associated to the generated power, and calculates the schedule in an offline way before the execution. The other technique, instead, reads the power value at the beginning of each sampling period, finding the schedule in an on-line way, according to the received data. In particular, if an incoming data is the same of the predicted one, the scheduler uses the predicted schedule without execute a new computation, else it applies the reactive mechanism, calculating a new schedule in a run-time way.

Selected loads must satisfy two properties: they have the highest priority level and they must minimize the difference between the generated power and the consumed one. The algorithm uses a greedy technique to maximize the size of the selected loads, based on the scaling factor. In fact, by modulating the power required by the single loads, it is possible to use the non-used power to feed other loads. Finding the optimal schedule is a time-consuming task, that could have a great impact on system responsiveness [14]. Depending on the number of loads to be handled, the scheduler could introduce a considerable delay during the computation. This is a dangerous situation, because, in a general case, if the cyber part is not sufficiently reactive to intercept an event, postponing its processing, the system may run abnormally respect the required conditions. To overcome this issue, the proposed algorithm works as follow: if the computation requires more than a sampling period, it ends its execution at the end of the sampling period, applying the best schedule founded in that given interval.

Later the available schedule is translated into coded strings, associated with the commands, which are forwarded towards the periphery by the gateway.

3.3 SIMULATION RESULTS AND EXPERIMENTAL RESULTS

The prototyped agent model was investigated in simulation, then in preliminary and real-time execution. To switch from one phase to another only the control machine, with the associated time notion, and actions, need to be replaced. The remaining part of the model is unchanged.

Experiments were performed on a time window of 400s using a Macbook pro Intel Core i5, 2.9 GHz, 16Gb, OS X Sierra.

By switching the different phases, the results are the same, although the real execution introduces a delay estimated in 4s, due to Arduino serial communication configuration.

The LabVIEW software loads the power signal samples in the AWG2021 memory, and configures it so as to generate the signal with sampling frequency equal to 10Hz and amplitude in the range [0÷5] V.

Fig. 5 shows a scenario in which the scheduler works only in a pure predictive case. It is supposed that both signal, generated power and predicted generated power, are the same. Furthermore, the total consumption curve keeps as near as possible to the generated one, without going beyond it.

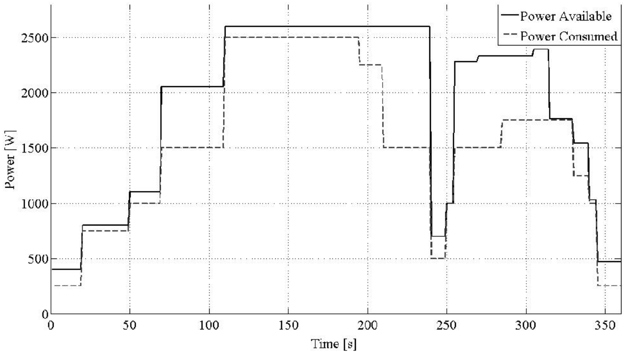

Fig. 5 – Experimental results using only predictive mechanism

Fig. 6 shows a different scenario in which the predicted generated power is not always the same of the real generated one. At 150s, in fact, samples begin to differ, and for this reason the schedule applies not only the predictive mechanism, using also the reactive until the end.

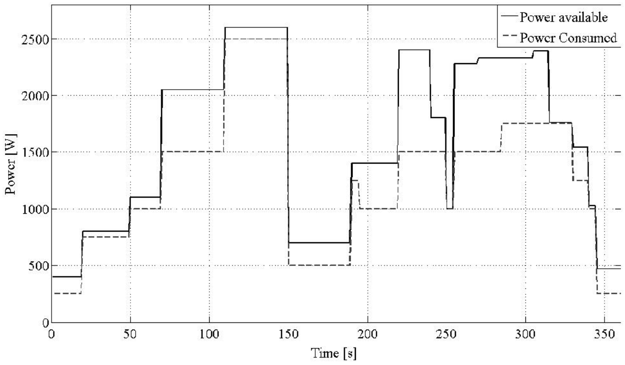

Fig. 6 – Experimental results using predictive and reactive mechanisms

The preliminary execution was useful, to estimate the drift introduced in the communication among agents. The drift describes the delay between the theoretical and real time delivery of the messages. Its upper bound found, estimated in 123 ms, is registered only at the beginning of the experimental simulation. In addition to the communication delay of the cyber-part associated to message exchange cost, there is also a delay concerning the data exchange with the physical part, due to MQTT protocol functioning and internet networking. The experimentation was carried out synchronizing the Raspberry clock with the one of a desktop computer with Ubuntu 17.04, using the ntp protocol [17]. The upper bound delay was estimated into a maximum of 67 ms.

4. CONCLUSIONS

This paper proposes a lightweight distributed agent-based methodology to develop CPS, as an extension of DMS. The framework uses the agent paradigm programming, to enrich the properties of the DMS with the ability to carry out actuations in the environment, where the measurement devices are placed.

Agents map measurement nodes and are managed by a control structure that is in charge of giving a global notion time to the overall system and to dispatch messages. Messages are pure software entities, that constitutes the basic element of communication. They are used to promote sociality, information exchange and to coordinate and negotiate the different measurement procedures.

A software component, called Gateway, is also present in the framework and acts as middleware, to enable communication between the cyber part and the physical part of the CPS.

A unique feature of this agent-architecture is the support of model continuity [11, 13, 14], that consists in the transitioning a same model from simulation down to real-time execution.

To test the validity of the proposed framework, a case study concerning the power management in a smart micro grid, is proposed.

On going activity aims to optimize the runtime agent infrastructure to improve the performances by avoiding Jade and developing a lighter middleware layer. The goal to achieve consists in the possibility of introducing more strictly constraints, moving towards a hard-real time execution and simulation, using the framework for systems with faster dynamics than the one proposed in the example.

5. REFERENCES

[1] E. Spanò, S. Di Pascoli, G. Iannaccone, “Internet-of things infrastructure as a platform for distributed measurement applications,” *in Proceeding of the IEEE Instrumentation and Measurement Technology Conference (I2MTC 2015)*, Pisa, Italy, May 11-14, 2015.

[2] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, “Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination,” *Computer Networks*, Vol. 101, 158-168, 2016.

[3] G. Rohbogner, U.J.J. Hahnel, P. Benoit, S. Fey, “Multi-Agent Systems Asset for Smart Grid Applications,” *Computer Science and Inf. Sys. (ComSIS)*, Vol. 10, pp. 1799-1822, 2013.

[4] J. Lee, H.A. Kao, S. Yang, “Service innovation and smart analytics for industry 4.0 and big data environment,” *Procedia Cirp*, Vol. 16, pp. 3-8, 2014.

[5] E.A. Lee, “Cyber Physical Systems: Design challenges,” in *Proceedings of the International Symposium on Object/Component/Service Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363-369.

[6] P. Leitão, A.W. Colombo, S. Karnouskos, “Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges,” *Computers in Industry*, Vol. 81, pp. 11-25, 2016.

[7] J. Sztipanovits, X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, S. Wang, “Toward a science of cyber–physical system integration,” in *Proceedings of the IEEE*, Vol. 100, pp. 29-44, 2012.

[8] F. Lamonaca,, A. Gasparri, E. Garone, D. Grimaldi, “Clock synchronization in wireless sensor network with selective convergence rate for event driven measurement applications,” *IEEE Transactions on Instrumentation and Measurement*, Vol. 63, pp. 2279-2287, 2014.

[9] T. Sanislav, L. Miclea, “Cyber-physical systems – Concepts, challenges and research areas,” *Journal of Control Engineering and Applied Informatics*, Vol. 14, pp. 28-33, 2012.

[10] S. Abras, S. Pesty, S. Ploix, M. Jacomino, “Advantages of MAS for the resolution of a power management problem in smart homes,” in *Proceedings of the 8th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAMS’10)*, Springer, pp. 269-278, 2010.

[11] F. Cicirelli, L. Nigro, “Control centric framework for model continuity in time-dependent multi-agent systems,” Concurrency and Computation: Practice and Experience, Vol. 28, No. 12, pp. 3333-3356, 2016.

[12] F. Bellifemine, G. Caire, D. Greenwood. *Developing Multi-Agent Systems with JADE*, John Wiley & Sons, 2007.

[13] D.L. Carni, F. Cicirelli, D. Grimaldi, L. Nigro, P. F. Sciammarella. "Agent-based software architecture for distributed measurement systems and cyber-physical systems design," in *Proceedings of the IEEE Int. Instrumentation and Measurement Technology Conference (I2MTC 2017),* Turin, Italy, May 22-25, 2017.

[14] F. Cicirelli, L. Nigro, P.F. Sciammarella, "Agents+Control: A Methodology for CPSs," in *Proceedings of the IEEE/ACM 20th Int. Symp. on Distributed Simulation and Real Time Application*, London, England, September 21-23, pp. 45-52, 2016.

[15] Mosquitto, An Open Source MQTT v3.1/v3.1.1 Broker - https://mosquitto.org/

[16] S. Lee, H. Kim, D.K. Hong, H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," in *Proceedings of the International Conference of Information Networking (ICOIN)*, Bangkok, Thailand, January 28-30, pp. 714-717, 2013.

[17] D.L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on communications*, Vol. 39, No. 10, pp. 1482-1493, 1991.

Domenico Luca Carnì, *received the master's degree in computer engineering and the Ph.D. degree in systems and computer engineering from the University of Calabria, Rende, Italy, in 2003 and 2006, respectively.*

In 2006, he joined the Department of Informatics, Modeling, Electronics and Systems, University of Calabria, as an Assistant Professor of Electric and Electronic Measurements. His current research interests include the measurement on telecommunication systems, characterization of digital to analog and analog to digital converters, digital signal processing for monitoring and testing, virtual instrumentation, and distributed measurement systems.

Domenico Grimaldi*, is a Full Professor of Electronic Measurement with the Department of Computer Sciences, Modeling, Electronics, and System Science, University of Calabria, Rende, Italy, where he has remained in a variety of research and management positions.*

He is responsible for the Laboratory and for processing the measurement information. He was responsible for the research unit in the frame of the National Projects PRIN, FIRB, and international projects INTERLINK supported by the Italian Ministry for University and Research. He was responsible for the Tempus Project and Leonardo da Vinci Project supported by the European Union, and the PARCO Project supported by Calabria Region. He is the delegate of the Rector of the University of Calabria for the safety. He has authored or co-authored more than 250 papers published in international journals and conference proceedings. His current research interests include the characterization of measurement transducers, digital signal processing for monitoring and testing, distributed measurements and synchronization, and measurement for medical application. The results of his research have led to a number of awards.

Prof. Grimaldi is a member of the Italian Group of Electrical and Electronic Measurements. He is member of the Editorial board of International Journal of Computing, Associate Editor of IEEE Transactions on Instrumentation and Measurement, and of Metrology and Measurement Systems.

Libero Nigro *is a full professor of computer engineering at Unical, DIMES, where he teaches object-oriented programming, software engineering and real-time systems courses. He is the responsible of the Software Engineering Laboratory (www.lis.dimes.unical.it).*

His current research interests include: software engineering of time-dependent and distributed systems, real-time systems, Petri nets, modeling and parallel simulation of complex systems, distributed measurement systems. Prof. Nigro is a member of ACM.

Francesco Lamonaca, *received the M.S. degree in computer science engineering and the Ph.D. degree in computer and system science from the University of Calabria, Rende, Italy, in 2005 and 2010, respectively, and the doctorate equivalent degree in science and engineering science from the Université Libre de Bruxelles, Brussels, Belgium, in 2010 and 2011, respectively. He is currently an Associate Professor of Electronic Measurements with the University of Sannio, Benevento, Italy. He has authored and co-authored more than 150 papers published in international journals and conference proceedings. His current researches interests include measurement for medical use, characterization of human tissue by thermal analysis, digital signal and image processing for health parameters monitoring, noninvasive monitoring and testing, synchronization of networking measurement instruments and sensors, and distributed measurement systems. Dr.*

Lamonaca is a Senior member of the Institute of Electrical and Electronic Engineers (IEEE), and the Italian Group of Electrical and Electronic Measurements.

***Paolo Francesco Sciammarella** was born in Avellino, Italy, in 1990. He received the Bachelor's degree first in Computer Science Engineering in 2012 and then the Master's degree in Automation Engineering in 2015, both at University of Calabria, Italy.*

Since 2015 he is Information and Communication Technologies Ph.D student at University of Calabria. His research interests include modeling, simulation and implementation of cyber-physical systems, Petri nets and Multi-Agent systems.