Work 41 (2012) 4250-4257
DOI: 10.3233/WOR-2012-0719-4250
IOS Press

Designing dynamic distributed cooperative Human-Machine Systems

A. Lüdtke[a,*], D. Javaux[b], F. Tango[c], R. Heers[d], K. Bengler[e], C. Ronfle-Nadaud[f]
[a]*Transportation,OFFIS Institute for Information Technology, Escherweg 1, 26121 Oldenburg, Germany*
[b]*Next Step Solutions, Rue Daussoigne-Mehul 20, 4000 Liège, Belgium*
[c]*Centro Richerche di Fiat, Strada Torino 50, 10043 Orbassano, Italy*
[d]*Visteon Innovation & Technology GmbH, Visteonstraße 4-10, 50170 Kerpen, Germany*
[e]*Technical University of Munich, Boltzmannstraße 15, 85747 Garching, Germany*
[f]*Ecole Nationale de l'Aviation Civile 7, avenue Edouard Belin 31055 Toulouse Cedex 4, France*

Abstract. The paper presents a new approach to the development of cooperative human-machine systems in the Transportation domain which is currently researched in the European project D3CoS. A necessary precondition for the acceptance of cooperative human-machine systems with shared control is the confidence and trust of the user into the system. D3CoS tackles this important issue by addressing the cooperative system as the object and the target of the system development process. This new perspective, along with corresponding innovative methods, techniques and tools, shall allow the identification of optimal task and authority sharing approaches supported by intuitive human-machine interaction and user interfaces at an early stage of system development. This will support powerful teamwork between humans and machines or between machines and machines that is transparent, intuitive and easy to understand. The paper describes the research dimensions for the development of the methods, techniques and tools as well as first results.

Keywords: cooperative system development and evaluation, reusable concepts for advanced human machine interaction, design patterns for adaptive task and resource allocation

1. Introduction

Transportation based on cars, aircraft and ships is a key factor in modern human societies. Human operators have historically been in charge of the two main facets of transportation: vehicle control and traffic control. Technological innovations have progressively allowed the introduction of advanced automated assistance systems leading to a complex and efficient interplay of humans and automation which has also been shown to lead in many cases to new types of human errors, incidents and sometimes accidents. It has been recognized that further automation alone cannot solve the problem and the crucial issue is how to achieve an adequate level of human-machine cooperation with shared authority.

The paper describes the objectives, research dimensions and first results of the European research project D3CoS (Designing Dynamic Distributed Cooperative Human-Machine Systems) where we develop affordable Methods, Techniques and Tools (MTT) which address the specification, development and evaluation of cooperative systems from a multi-agent perspective where human and machine agents are in charge of common tasks. A critical idea of D3CoS is that the object of the System Development Process is the cooperative system as a whole (i.e., a multi-agent system), not only its components.

D3CoS puts emphasis on cross-domain applicability of the MTTs focusing on the commonalities between four demonstrator transportation domains: Manned Aircraft, Unmanned Aerial Vehicles (UAV), Automotive and Maritime. Corresponding cooperative systems pose common engineering challenges peculiar to the inherent (kinetic and potential) energy, time and space-based characteristics of transportation.

*luedtke@offis.de

The MTTs are developed in three feedback cycles. In each cycle (1) industrial requirements for the MTTs are collected/refined, (2) the MTTs are developed/improved, (3) the MTTs are applied to develop demonstrator applications in the four domains, and (4) feedback is collected as well as processed.

2. Methods, Techniques and Tools for Cooperative Systems

In D3CoS a cooperative system (cf. ① in Figure 1) is defined as a set of agents, which can be either human②[a] or machine②[b] agents. These agents basically interact and communicate③ with each other, thus creating a cooperative network. Then, tasks④ are assigned to the cooperative system and allocated⑥ to the agents, who achieve them by using their own as well as distributed resources⑤. Each agent has access⑦ to specific usually limited resources. The cooperative system operates on one or more controlled objects⑧, either a vehicle or traffic. That means, that the cooperative system together with the objects it controls is immersed in an environment⑨ (e.g., weather, communication, infrastructure).

Moreover, each of the components of the cooperation framework can be static or dynamic. In addition, the agents in the system may change (some of them leaving, others incoming) or the capabilities of agents may change (e.g. performance of human agents deteriorates under stress). Also, tasks may change (e.g., an aircraft has to deviate to avoid weather) as well as the available resources (e.g., the engines may fail).

The development of cooperative systems in the described framework is supported in D3CoS by five complementary research dimensions to address major general aspects in human-machine cooperation:

a) definition of a generic cross-domain applicable framework for cooperative systems,
b) modeling and simulation of cooperative systems including human modeling and simulation,
c) reference designs and design patterns for intelligent multi-modal human-machine interfaces,
d) reference designs and design patterns for state inference and dynamic system adaptation,
e) operational architectures for communication and coordination within cooperative systems.

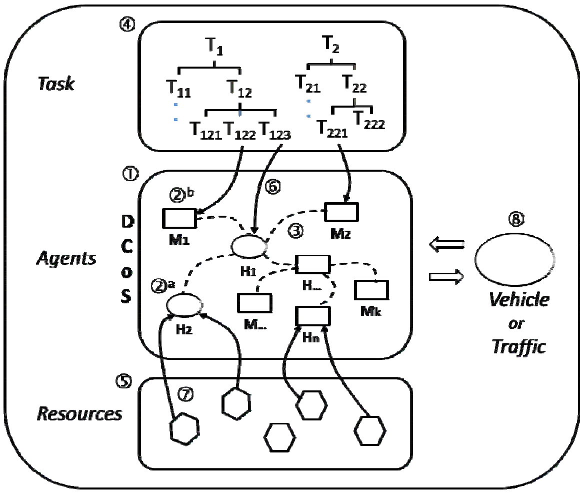

Fig. 1: Generic cooperative system framework

2.1. Generic Cross-Domain Cooperative System Framework

The challenge is to highlight the commonalities between the domains (with a cooperative system framework such as the one in Figure 1), understand the common methodological, theoretical and technical challenges they pose, and address them by providing appropriate MTTs such as those detailed below, organized into a common, generic and reusable DCoS Development Methodology supporting the development process at all steps (requirements capture, specification, development, evaluation).

D3CoS aims at supporting the industry and system engineers to pay improved attention to the intrinsically cooperative nature of the human-machine systems they are (implicitly) designing. D3CoS suggests that human-machine systems, especially cooperative ones, should be an object of design, and that specific, new methods should be used. These methods will enable the designers to efficiently and systematically analyze and decompose the cooperative system into its relevant subcomponents (tasks, agents, resources) and systematically specify and develop these components and the possible interaction strategies between them according to different optimization criteria.

2.2. Modeling and Simulation of DCoS

The challenge is to provide modeling and simulation tools for designers that allow to build up models of a cooperative system in order to analyze relevant quality and safety criteria of the overall performance already in early phase of the development life-cycle.

Existing experimental simulation platforms allow to simulate interaction between human and machine agents in order to analyze potential human errors of operators. Human agents are simulated using executable cognitive models. In order to tackle evaluation of cooperative systems D3CoS will extend such platforms by considering distributed cognition [1] allowing to analyze different cooperation modes with regard to safety, efficiency and effectiveness of the overall team performance. Distributed cognition extends the concept of traditional cognition beyond the processes taking place in individual minds (or cognitive agents in general, including machine ones). Reusing existing simulation tools is a key requirement for the platform.

2.3. Reference Designs and Design Pattern for intelligent multi-modal human-machine interfaces

The challenge is to provide reusable interaction principles and interaction modalities for cooperative systems that enable human and machine agents to cooperate on a higher semantic level (e.g. to communicate intentions of machines and humans) to support a deeper level of understanding and to guarantee real cooperation instead of the traditional master-slave relation.

Two aspects of multimodality are taken into account: (1) to support multimodal interaction (input/output of information) between human and machine agents and (2) to rely on multimodal information to assess human agents' status and availability. For (1) new forms of interaction are investigated that benefit from the observation of human-human interactions, to derive an understanding of the use of multiple modalities for communicating commands, but also more sophisticate information such as plans, intentions or for supporting negotiation. In (2) multimodality is used to provide a reliable estimation on the availability of the human partner in a cooperative framework. Insights from the observation and modeling of natural human-human communication and cooperation are used and generalized to human machine cooperation. Research and development on these two aspects (1) and (2) will allow to integrate multimodality in the panels of MTTs we will produce, and in particular in the form of reusable reference designs and design patterns involving multimodality as an appropriate solution for sophisticated communication and interaction design problems.

2.4. Reference Designs and Design Pattern for state inference and dynamic system adaptation

The challenge is to provide cross-domain reusable metrics and measurements for inference of the current and future DCoS state as well as algorithms for dynamically deriving adaptations of the overall the system to guarantee optimal authority sharing and task allocation at each point in time.

An issue of major importance for a cooperative system is the ability to adapt to hazardous events. This presuposes to implement situation tracking tools to diagnose the state of a cooperative system and possibly modify the cooperation modes and interactions among the agents. Different techniques have been developped in the artificial intelligence domain such as particle petri nets [2], support vector machine [3], hidden Markov Model [4] or fuzzy inference engine [5]. Nevertheless, the efficiency of these formal methods relies mostly on the metrics and the measurements that have to be monitored. As long as human agents are part of a cooperative system, the question is not trivial. In such heterogeous systems, a first metric to be considered is the occurrence of conflicts as is classically used in distributed artificial intelligence and is relatively easy to formalize [6][7]. Moreover, from a cognitive point of view, the occurrence of such conflicts is a remarkable precursor of loss of situation awareness [8][9]. A second metric is based on the monitoring of the human operators' psychophysiological and ocular activities as long as they reflect workload, stress and attentional demand [10]. An analysis of these metrics and measurements will allow to determine tasks and authority allocation regarding the efficiency of the agents and determine a series of reference designs and design patterns that capture efficient, reusable solutions for state inference and overall cooperative system adaptive reconfiguration (cooperation modes, tasks and resource allocation, authority sharing)

2.5. Operational DCoS Architecture

The challenge is to provide cross-domain reusable architectures that allow real-time interaction between all agents including exchange of all relevant parameter for necessary for optimal authority sharing and task allocation.

Therefore, existing architecture approaches will be taken into account, within the different domains and addressing different concepts (e.g. service-oriented architectures and agent-based architectures). These architectures can follow the autonomic paradigm (i.e. should have the self-healing characteristics), they should support real-time computation and guarantee robustness. Cooperative applications are heterogeneous, complex and dynamic. The underlying infrastructure is constituted by a lot of independent and heterogeneous resources, whose interaction is highly complex (concerning the computational power, communication, data storage, sensors and SW applications). Thus, innovative paradigms for system and application design are needed. In particular, an autonomic computing paradigm must have the following features:

- Protect itself from possible dangerous behaviours (of other applications or users)
- Repair itself in case of a bug/error or a malfunctioning
- Configure itself, in order to react to external environmental changes
- Keep its own performances at certain optimal level (as much as possible)

The high mobility of cooperative devices together with the diversity and variability of their computational resources or energy power consumption raises some specific needs regarding communication protocols. Several research approaches on this topic are of interest for the communication architecture envisioned in D3CoS. Mobile ad hoc networks for example, which are self-configuring, self-healing networks [11][12]. Networks aware of the needs of their agents in terms of communication and able to adaptively provide them with (or inform of the impossibility to provide) a certain quality of service are also interesting (e.g., the cognitive networks of [13]). We investigate these concepts and operationalise them to provide a relevant communication architecture for D3CoS cooperative systems.

3. General Cooperative System Development Process

The results of the five research dimensions described above are used as the technological basis of a new methodology for supporting the cooperative system development process, based on three methodological steps (cooperative system composition, interaction, interfaces) combined with the traditional industrial phases for System Development (requirement capture, specification, development and evaluation). A preliminary structure for a cooperative system methodology is presented in Figure 2. Overall, the methodology shall take into account and combine methodological steps required to create a cooperative system as well as industrial best-practice development processes:

- DCoS composition: (i) Requirement Capturing: System designers and/or evaluators study the tasks and resources assigned to the DCoS, the environment in which it will operate, the different families of scenarios it has to deal with. (ii) System Specification: They then determine the agents of which it should be composed, which functions, role or tasks should be assigned to them, and the cooperation modes between them. (iii) System Development: The machine agents (Embedded Systems) are then developed. (iv) Evaluation&Test: The DCoS can then be evaluated in terms of its capability to satisfy the requirements previously captured.
- DCoS Interaction: (i) system designers and/or evaluators define (Requirement Capturing) which agents need to interact with which agents, with which protocol and thanks to which communication modalities (channels), for all required cooperation modes (e.g., task sharing patterns, authority structure,...) identified in step 1. (ii) System Specification: specify technical and socio-technical (e.g., procedures) solutions to cope with the requirements. (iii) System Development: develop technical and socio-technical (e.g., procedures) solutions to cope with the requirements. (iv) Then, once again, the whole system can be evaluated (Evaluation&Test) with regard to the satisfaction of these requirements.
- DCoS Interfaces: (i) System and interface designers define (Requirement Capturing) the information that must be exchanged through the interfaces, based on the interactions between machine and human agents specified in step 2. (ii) System Specification: specify the devices and physical interfaces that support these cooperative interactions. (iii) System Development: implement the devices and physical interfaces that support these cooperative interactions. (iv) Evaluation&Test System evaluators then assess and test the satisfaction of these.

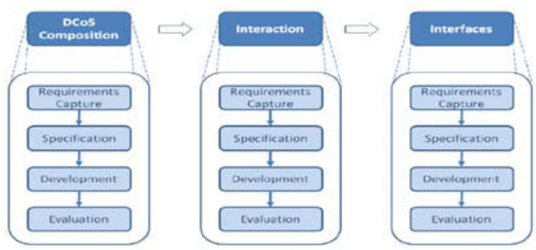

Fig. 2: Generic cooperative system development process

4. Demonstrators

One key objective of the D3CoS project is to apply the MTTs to develop cooperative system demonstrators in order to assess the added value of the MTTs for industrial development of cooperative systems in four domains: Manned Aircraft, Unmanned Aerial Vehicles (UAV), Automotive and Maritime. The demonstrators will be developed iteratively in the three project cycles. In each cycle we will follow the common development process as outlined above in Section 3.

Thus, the demonstrators will address two perspectives specific to the Transportation domain: on-board perspective and traffic perspective.

On-board perspective: a specific DCoS architecture will be used for the demonstrators resorting from the on board perspective (Manned Aircraft, Automotive, Maritime). The architecture is a peculiar instantiation of the generic cooperative system architecture depicted in Figure 1: a vehicle (e.g., a car, an airplane, a ship) is controlled by a cooperative system consisting of human and machine agents, all on board the vehicle: (a) set of one or two human agents a and zero, one or more machine (b) agents are in charge of the main control tasks. They are assisted by specialized “assisting” machine agents (c) a state assessor, a conflict solver, a task & resource allocator and an assistant system (which effectively assists the main human and machine agents when needed or useful). The task dynamically assigned to the cooperative system is to move the vehicle from origin to destination, possibly following a specific itinerary, in fully safe conditions, and with possible performance objectives (deadlines, fuel consumption). The agents exploit resources such as engines or power units, fuel, navigation infrastructure to perform the tasks. The vehicle and the agents are immersed in a typically complex and semipredictable environment (e.g., weather, other vehicles).

Traffic perspective: the DCoS architecture for the demonstrators resorting from this perspective (UAV, Automotive, Maritime), again a peculiar instantiation of the generic cooperative system architecture depicted in Figure 1, is based on a cooperative system consisting of multiple vehicle agents, possibly complemented by an (optional) control agent, external to the vehicle (e.g., air traffic control). The vehicles are either machine agents (e.g., an unmanned aircraft) or simpler cooperative systems made of human and machine agents (e.g., a car) (and therefore resorting from the on board perspective). The object under control is the flow of the vehicles itself, or traffic. The task assigned to the cooperative system is to fluidize the flow and allow each vehicle to pursue its own objectives (itineraries and performance), in fully safe conditions, using resources such as a space, time, and communication and geo-localization infrastructures. The vehicles (and possible external control agent) are immersed in a complex and semi-predictable environment (e.g., weather, uncontrolled traffic).

4.1. Manned Aircraft

The Manned Aircraft cooperative system resorts from the on-board perspective. It incorporates human pilots, aircraft (including its cockpit subsystems like the auto flight system and flight management system) and air traffic controller in flight tasks and serves as a demonstrator for the added value of the MTTs relevant for aircraft cockpit development throughout the D3CoS project. The main features to be tackled within this system are twofold: definition of a suitable cooperative system supporting the detection and resolution of authority sharing conflicts immanent in relevant Aeronautics scenarios and provision of human-machine interaction concepts to guarantee a shared understanding between involved agents. In order to realize these features innovative cooperative machine agents will be developed as parts of the cooperative system like dynamically adaptive checklists, task reallocation mechanisms that make use of workload measurement and aim for global situational awareness (as an important precondition for future virtual co-pilot and single pilot concepts) or procedural deviation warnings.

4.2. Unmanned Aerial Vehicle

The UAV cooperative system resorts from the traffic perspective. It incorporates several UAVs (a UAV swarm) and a remote human ground operator who defines the mission, performs surveillance tasks (akin to an optional traffic control agent) and is able to intervene if necessary. The UAV cooperative system serves as a demonstrator for the added value of the MTTs relevant for UAV swarm development and deployment.

4.3. Automotive

The Automotive demonstrator will consist of two sub-demonstrators. The first demonstrator is based on In-Vehicle Information Systems (IVIS) and focuses on the on-board perspective, taking into account the sharing of tasks between driver, passengers and machine agents realizing the IVIS functionalities. The second demonstrator is based on Advanced Driver Assistance Systems (ADAS) and focuses on the traffic perspective, taking into account the cooperation between two or more driver-vehicle systems. For both demonstrators we will develop innovative cooperative system applications with machine agents which include functionality for cooperation with other agents (e.g. the driver) like dynamic authority sharing and adaptive multi-modal HMI interface solutions. Their functionality will be based on highly innovative systems developed by D3CoS project partners in previous research projects that will be extended with cooperative functionalities in D3CoS.

4.4. Maritime

The Maritime demonstrator focuses on the safe berthing of ports traffic supported by Vessel Traffic Service and the Port Authority. The main agents involved are the human agents: officer of the watch, master, pilot, VTS-operator and the machine agents: autopilot / track-pilot and the engine control-system. VHF radio, AIS (Automatic Identification System) and GPRS (2G mobile phone) are needed as resources for communication, additionally RADAR and ECDIS (Electronic Chart Display and Information System) are needed as resources for navigation. The demonstrator will include AVTMIS (Active - Vessel Traffic Management Information System) machine agents acting in port and between ship and port that perform berth allocation and path planning and that show the safe distance to other participating vessels. Because it involves human and machine agents cooperating on board on a ship, the Maritime demonstrator resorts from the on-board perspective. Because it aims at improving traffic at ports and safe berthing, by distributing traffic management between multiple vehicles (ships), it also resorts from the traffic perspective.

5. Requirements for Method, Techniques and Tools

In the first phase of the project requirements for the development of the MTTs have been collected from the industrial partners in D3CoS. The MTTs are intended to support engineering of cooperative systems in four steps of industrial engineering: (1) system requirements collection, (2) system specification, (3) system development and (4) system evaluation. The requirements have been collected for these four engineering steps separately.

The following list shows some examples of requirements for MTTs intended to support the industrial engineering step system requirements capturing:

- MTTs shall allow to prioritize system requirements by their relative importance
- MTTs shall address functional capabilities, performance levels, data structures/elements, constraints and limitations
- MTTs shall address the identification and profiling of the intended users
- MTTs shall engage the intended users and subject matter experts for their input
- MTTs shall support simultaneous definition of requirements by multi actors, e.g. OEM (Original Equipment Manufacturer) and supplier.
- MTTs shall be able to interact with standard commercial requirements capturing tools

The next list shows some examples of requirements for MTTs intended to support the industrial engineering step system specification:

- MTTs shall allow to visualize and crosscheck information flows within a DCoS system in accordance to defined requirements
- MTTs shall allow to identify inconsistencies within a cooperative system specification
- MTTs shall support specification of progressing system design layers (system components are decomposed and refined by adding further details) in order to support changes during development and to keep track on interdependent requirements and changes (as one change in an agent may have an impact on another agent)
- MTTs shall be able to interact with standard commercial specification tools

The next list shows some examples of requirements for MTTs intended to support the industrial engineering step system development:

- MTTs shall provide the developer with a graphic environment to drag and drop the key elements of a cooperative system (such as agents, resources and tasks)
- MTTs shall provide the developer with some hints and guidelines according to the standards available for the domain he/she is working on
- MTTs shall allow for an easy manual re-allocation of tasks to different agents to simulate the changes in workload and the respective effects
- MTT shall allow the developer to seamlessly develop and integrate the DCoS in a single platform

The final list shows some examples of requirements for MTTs intended to support the industrial engineering step system evaluation:

- MTT shall allow to simulate different DCoS cooperation modes to evaluate their effects
- MTTs shall precisely define the criteria for the evaluation of the performance of the total DCoS
- MTTS shall allow to measure different criteria for DCoS performance evaluation with the criteria being on the system level
- MTTs shall allow the integration of external system hardware/software as black box algorithm for simulation purposes
- MTTs should support DCoS simulation for performance evaluation with human performance models in the loop instead of real human subjects

6. Summary and Future Work

Thanks to the three methodological steps (DCoS composition, Interaction, Interfaces) and the four industrial phases (Requirements Capture, Specification, Development, Evaluation) D3CoS will provide methods, techniques and tools (MTTs) for the development of highly innovative cooperative human-machine systems.

The D3coS work plan relies on three cycles, preceded by a preparatory phase.

During the preparatory phase, the operational requirements for each demonstrator, as well as common requirements (cross-domain) are collected from the industrials using scenarios and use-cases. The MTTs requirements for each of the development phase (requirement, specification, development and evaluation) are listed and described. They are based on expert knowledge, pre-existing data or literature studies. A first version of the MTTs, preliminary to the demonstrator development, is made by adapting pre-existing results.

In the first cycle of D3coS, the industrial requirements for the demonstrator will be refined and the MTTs will be developed or improved in conformance with the preparatory phase. Low-fidelity simulator applications will be realized for the four demonstrators. An evaluation will be done by virtual pilots or virtual operators. Then feedback will be collected and used to improve the demonstrator's requirements. Evaluation and experiments will be also used to assess the MTTs (version 1).

Cycle2 and Cycle 3 will apply exactly the same pattern. The whole process will produce more and more improved versions of the demonstrators.

At the end of the project (end of Cycle 3), four demonstrator applications will be realized and during three successive cycles, the progressive benefit of the D3CoS MTTs, with regards to reduced cost of DCoS development, achievement of DCoS development cycles and achievement of cross-domain applicability and reusability of Embedded Systems devices will have been demonstrated.

Acknowledgement

The work described in this paper is funded by the ARTEMIS Joint Undertaking under the number 269336-2 (www.d3cos.eu).

References

[1] J. Hollan, E. Hutchins and D. Kirsh, Distributd Cognition: Toward a New Foundation for Human-Computer Interaction Research, ACM Transactions on Computer-Human Interaction, 7(2), 2000, pp. 174-19.

[2] C. Lesire and C. Tessier, Particle Petri nets for aircraft procedure monitoring under uncertainty. Applications and Theory of Petri Nets, 2005, pp. 329-348.

[3] R. Picard, Toward machine emotional intelligence: Analysis of affective physiological state, IEEE Trans. Pattern Anal. Mach. Intell., 10(23), 2001, pp. 1175–1191.

[4] J. Scheirer, R. Fernandez, J. Klein and R. Picard, Frustrating the user on purpose: A step toward building an affective computer, Interacting Comput., 14, 2002, pp. 93–118.

[5] D. Kulic and E. Croft, Anxiety detection during human–robot interaction, Proc. IEEE Int. Conf. Intell. Robots Syst., 2005, pp. 389–394.
[6] J. Rushby, Using model checking to help discover mode confusions and other automation surprises. Reliability Engineering & System Safety, 75(2), 2002, pp. 167-177.
[7] F. Dehais, C. Lesire, C. Tessier and L. Christophe, Method and device for detecting piloting conflicts between the crew and the autopilot of an aircraft, WO/2010/000960 Airbus Patent, 2010
[8] D. Woods and N. Sarter, Learning from automation surprises and going sour accidents. Cognitive engineering in the aviation domain, 2000, pp. 327–353.
[9] F. Dehais, C. Tessier and L. Chaudron, GHOST: experimenting conflicts countermeasures in the pilot's activity, International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003.
[10] M. Causse, J. Pastor, J.-M. Sénard and J.-F. Démonet, Monitoring cognitive and emotional processes through pupil and cardiac response during dynamic vs. logical task, Applied Psychophysiology and Biofeedback, 35, 2010, pp. 115-123.
[11] C. K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2001.
[12] C. Perkins, Ad Hoc Networking, Addison-Wesley Professional, 2008.
[13] F.H.P Fitzek and M.D. Katz (eds), Cognitive Wireless Networks, Springer, 2007.