WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* 2009; **9**:445–467
Published online 30 September 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.689

# Next generation grids and wireless communication networks: towards a novel integrated approach

Romano Fantacci[1*,†], Marco Vanneschi[2], Carlo Bertolli[2], Gabriele Mencagli[2] and Daniele Tarchi[1]
[1]*Department of Electronics and Telecommunications, University of Florence, Firenze, Italy*
[2]*Department of Computer Science, University of Pisa, Italy*

## Summary

One of the most promising trends for next generation networks is to consider an integrated approach to the communication infrastructure and the processing layer. In particular, the introduction of broadband and reliable wireless networks allows the interaction of a huge number of devices all creating a single network. On the other hand, the grid paradigm is considered as one of the most promising approach for pervasive and dynamic applications. Aim of this paper is to present a novel integrated approach between grid paradigm and wireless networks by highlighting the main advantages of their cooperation. In particular, it will be shown here how a wireless heterogeneous network can be exploited for implementing a pervasive and dynamic grid (mobile grid) and, on the other hand, a mobile grid allows the optimization of the communication infrastructure. The integrated approach can be an effective method for solving applications, such as emergency management, where a huge amount of data derived from a wireless infrastructure needs to be processed efficiently and adaptively, and the traffic flow in the wide area wireless networks needs to be coordinated and optimized. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: next generation networks; mobile grid; wireless networks; adaptivity; context-awareness; programming model

## 1. Introduction

One of the most important topics for next generation networks paradigm is a joint approach to networking in terms of scalable applications. On one side, the mobile computing offers the opportunity to have a huge number of devices disseminated all over an area. On the other side, the grid paradigm for co-operative and high-performance enabling platforms is evolving from the first generation infrastructures for scientific computing to systems for more dynamic and pervasive applications. Historically the grid approach has been used for fixed general-purpose computers connected by high-speed networks. The introduction of broadband wireless technologies and the improvement in computational capabilities of portable devices allows now to interconnect also mobile devices with high speed structures and to use them for processing parts of distributed applications.

An increasing number of critical applications requires the existence of novel distributed, heterogeneous, and dynamic ICT platforms composed of a

*Correspondence to: Romano Fantacci, Department of Electronics and Telecommunications, University of Florence, Firenze, Italy.
†E-mail: romano.fantacci@unifi.it

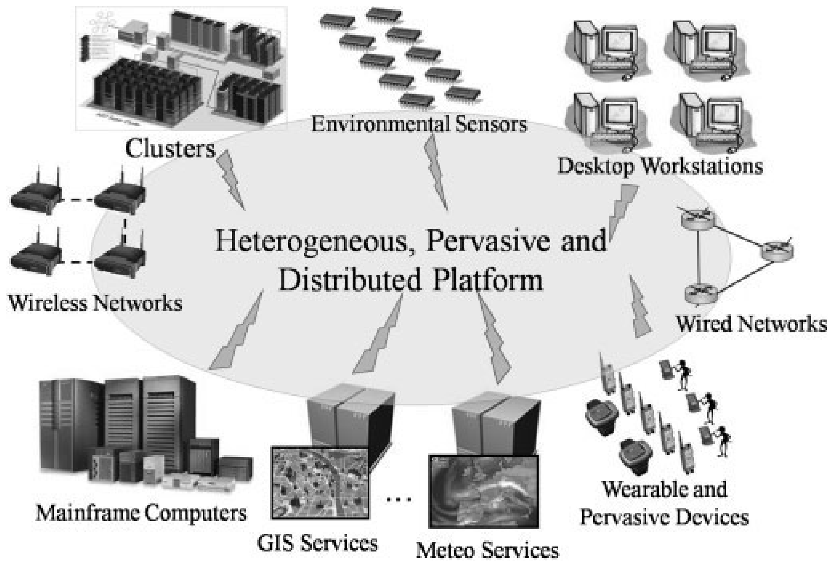



Fig. 1. Heterogeneous, pervasive, and distributed platform.

variety of fixed and mobile processing nodes and networks. Notable examples of such applications are (but are not limited to) risk and emergency management, disaster prevention, homeland security, and i-mobility. These platforms are characterized by the full virtualization of ubiquitous computational resources, data and knowledge bases, embedded systems, PDA and RFID devices, wearable computers, and sensors, through fixed mobile and *ad hoc* networks, as illustrated in Figure 1.

Wireless-based platforms, enabling the robust, flexible, and efficient cooperation of mobile components, including both software components and human operators, are of special interest. This cooperation implies the development, deployment, execution, and management of computations that, in general, are *dynamic* in nature. Dynamicity concerns the number and the specific identifications of cooperating components, the deployment and composition of the most suitable version of software components, processing and networking resources and services, i.e., both the *quantity* and the *quality* of the application components to achieve the needed quality of service (QoS). The specification and requirements of QoS itself are varying dynamically during the application according to the user intentions and to the *context* situation, for example, the information produced by sensors and services as well as the monitored state and performance of networks and nodes.

For a QoS-driven approach to complex distributed applications, we need to fully understand and to realize large and very large-scale platforms, that is, composed of a very high number (from thousands to millions) of heterogeneous nodes, supported by architectures, programming tools, and development environments that are not yet available at industrial level. The general reference point is the Grid paradigm [1–5] which, by definition, aims to enabling the access, selection, and aggregation of a variety of distributed and heterogeneous resources and services. However, though notable advancements in recent years have been achieved, grid technology is not yet able to supply the needed *software technology* with the requirements of high *adaptivity*, *ubiquity and mobility*, *proactivity*, *self-organization, scalability and performance*, *interoperability*, as well as *fault tolerance, and security*, of the emerging applications running on a very large number of fixed and mobile nodes connected by wireless networks [6,7].

The *mobile grid computing* paradigm addresses this framework of problems from a scientific and a technological point of view.

The main aim of this paper is to present an innovative approach that integrates the mobile grid paradigm and the wireless communication technology, in order to study and to experiment new QoS-driven architectural models for large scale distributed platforms characterized by high heterogeneity, mobility, and dynamicity. The strong integration of Computer Science and of Communication Technology, which characterizes our proposal, is also in line with



DOI: 10.1002/wcm

some notable recent trends like GENI [8] and Clean Slate [9].

The integrated approach is studied and developed in the context of the Integrated Systems for Emergency (In.Sy.Eme.) management research project.[‡]

The paper is structured as follows. In Section 2, the mobile grid approach is introduced by focusing on its principles and then by introducing the novel approach of the mobile grid architecture and programming model. In Section 3, the effect and how the communication infrastructure needs to be thought for enabling a mobile grid architecture and programming model is discussed; in particular, the attention has been focused on three different effects of the mobile grid approach on the communications: the data gathering, the localization techniques, and the data communications. In Section 4, the main feature of the integrated approach between the mobile grid and the wireless communication technology are outlined. Finally, in Section 5 Conclusions are drawn.

## 2. The Mobile Grid Approach

With the considerations of Section 1 in mind, our approach to the mobile grid paradigm is based on a novel combination of the disciplines of *pervasive computing* and of *high-performance and adaptive programming models*. The objective is the definition and realization of an architectural model and a software technology, for the applications of large-scale heterogeneous and mobile platforms, based upon:

- an *adaptive and context-aware programming model*, for designing applications viewed at several layers, that is, the abstract computation layer, the management layer, and the context layer;
- a set of tools and run-time support services that implement the programming model on top of large scale mobile platforms with the features outlined above, that is, the *mobile grid abstract machine* (GAM) that implements the software infrastructure of the platform, thus subsuming the traditional middleware functionalities according to a more efficient and user-oriented approach.

The definition of the architectural and programming model is based on the following features:

- QoS-driven approach at all the application layers,
- explicit, yet structured, visibility of the application layers in such a way that 'performance contracts' can be established dynamically between entities at the various layers,
- on demand adaptivity of functionalities and of performance,
- context awareness of the application at all application layers.

In order to understand how a structured approach to the architectural and programming model can be defined, we start from an analysis of the fundamental requirements of the applications of interest. We explicitly consider emergency management applications, however, the analysis applies also to other applications of large-scale distributed and mobile platforms.

### 2.1. Application Requirements

#### 2.1.1. An emergency management example

Let us consider a schematic view of an emergency management application, as exemplified by fluvial flood management.

During the 'normal' behavior, several parameters are periodically monitored and acquired through sensors, and possibly by other *services* (notably, *meteorology*, *GIS*): current value and variation of flow level, surface height, density of flow in each spatial coordinate, water density, wind speed, atmospheric temperature and pressure, characteristic parameters depending on latitude, meteorological conditions, and so on. A *forecasting model* is periodically applied for specific geographical areas and for widest combinations of these areas. An example is MIKE 21 hydrodynamic model [10], based on mass and momentum partial differential equations to describe the flow variation at surface. Their discrete resolution requires, for each time slice, the resolution of a very large number of linear (tri-diagonal) systems, each one of large size. Thus, we have to deal with a typical *data and computation intensive problem*, which requires suitable high-performance solutions. Parallel techniques are available (e.g., according to the data-parallel cyclic reduction method [11,12]), which make it possible to achieve reasonable response times in a scalable manner.

Sometimes, the forecasting model, and/or the direct intervention of human operators, may signal abnormal

[‡]FIRB In.Sy.Eme. project, Integrated Systems for Emergency. http://www.unifi.it/insyeme



DOI: 10.1002/wcm

situations that could lead to a flood. Alternatively, a human operator can ask the execution of the model for a specific area with specific parameters, and/or detects an abnormal situation.

Moreover, it could be necessary

- to acquire new data and to instantiate the proper level of knowledge, through *filtering, fusion, and classification* techniques,
- to execute complex *simulations*, on almost the complete state space, to acquire a reliable and updated view of the context,
- to run further models, such as *data mining, profiling, and decision support systems*, in order to plan rescue intervention, assistance courses, people organization, and so on,
- to perform *high quality graphic and visualization activities*.

It is important to notice that the same *networking and communication strategies* could require powerful resources and high-performance computations, as exemplified by current trends towards network processors [13,14].

The computations mentioned above could be executed by central servers, if available and connected, especially during the 'normal' behavior. However, it may be convenient or necessary to exploit decentralized resources, including mobile and wearable devices, and network processors too, according to their computing, storage, and communication capabilities.

In many mentioned cases, performance is a critical parameter, concerning the response time, throughput, or both. Let us assume a situation in which the network connection of the human operator(s) with the central servers is down or unreliable: to manage the potential crisis in real time, we can think to execute the models and the visualization tools on a set of decentralized resources (and possibly a subset of available central resources) whose interconnection is currently reliable. The parameter of the models could be available and updated, otherwise the same resources have also to run predicting models for the parameter values too. The *logical interconnection structure* of the application has to be mapped onto the physical communication infrastructure in the most efficient way as possible: the *networking strategy* to be adopted has to run jointly and consistently with the forecasting model and the data prediction model, as well as with all the other computational activities.

#### *2.1.2. Processing + Communication integration for a true user-centric approach*

Referring to the examples above, it is clear that there is a complex problem in dynamic allocation of software components to processing and communication resources. Some resources may have specific constraints in terms of storage, processing power, power consumption: the same version of the software components may be not suitable for them, or even may be impossible to run it. Thus, the application must be designed with several levels of *adaptivity* in order to be able to cover several resource availability situations and several QoS dynamic requirements. For example, we wish to be able to restructure the *same* parallel component according to alternative versions differing in response time, or throughput, or storage size, or power consumption, and so on.

The complexity of dynamic management of software components and processing-communication resources introduces several scientific and technological problems which require novel approaches and solutions, since such problems cannot be solved by means of current grid computing and pervasive computing technologies or by some marginal extensions to them.

The correct approach is to define the architectural and programming models *starting from the user viewpoint*.

Transformations from raw data to user-oriented information are implemented through a sort of *'interleaved chain' of processing and communication phases*. In some applications, or their parts, communication may be the prevalent activity (e.g., data streaming control and management), while, in other cases, intensive processing is done on statically available or locally generated data. In the most general and useful cases, communication and processing activities must be properly and dynamically interleaved in order to achieve the needed QoS of data and services.

In several phases, activities may be *data- and/or computation-intensive* (complex mathematical models for forecasting, simulations, filtering/fusion, data-mining, decision support strategies, and so on). This characteristic requires high-performance capabilities.

These issues have the following consequences on architectural and programming principles for mobile grids:

I. the project platform is a virtually unique system, composed of all the available processing and communication resources, according to a novel



DOI: 10.1002/wcm

*integrated approach to computation and communication*;

II. the programming model is based on the concepts of *components*, of flexible composition of components, and of *high-performance* components;

III. system resources are managed dynamically, and software components are deployed dynamically onto the system resources. *Adaptivity* and *context-awareness* are fundamental features of the architecture and of the programming model;

IV. to design adaptive and context-aware strategies, a *proper visibility of the system application layers* is necessary (with sensible differences compared to traditional, general purpose systems based on the concept of 'opaque layers') [15]. This principle is strongly related to adaptivity and context-awareness (III), because the application management strategies are often fired by 'contracts' that are dynamically established between entities at different layers;

V. *portability*, which of course is a general feature of any component technology, in our case must address explicitly the issue of dynamic deployment and adaptivity of application components onto several and extremely different and heterogeneous resource kinds (general servers, dedicated nodes, mobile devices, and so on).

These principles will be discussed in the next Sections.

## 2.2. Mobile Grid: Architectural and Programming Issues

In this Section, we position our view of mobile grid with respect to the trends in grid technology, pervasive computing, and hardware technology trends.

### *2.2.1. Next generation grids*

Till now grid technology has exploited a very limited amount of its potentials, mainly in the scientific application domain and some *e*-business and *e*-government domains. In general, current grid platforms are not predisposed to a uniform approach to distributed and parallel computing, are homogeneous or heterogeneity is limited, adaptivity is scarcely exploited (limited to node reservation/scheduling in large jobs), integration with advanced networking technologies is low. Most notably, the commercial grid technology, or some large scientific infrastructures (e.g., EGEE in Europe) are extremely limited from the viewpoint of high-level, high-performance, adaptable, and context-aware programming models.

Many scientific advances have been done in recent years about programming technology, notably in the European projects CoreGRID [4,5,16] and GridComp [17]. Moreover, international working groups on next generation grids [2] have clearly indicated that one of the most promising trends of grid computing is just in the direction of pervasive, ubiquitous, and mobile applications. The key distinguishing points are programmability, heterogeneity, uniformity of distributed and high-performance parallel computation, system wide adaptivity and autonomic behavior, context-awareness, and strong integration of processing and communication technologies.

### *2.2.2. Pervasive computing*

Pervasive computing [18,19] is an area that can sensibly contribute to the advancement of next generation grids. This paradigm is centered upon the creation of systems characterized by the a multitude of heterogeneous, mobile, and ubiquitous computing and communication resources, whose integration aims to offering seamless services to the user according to their current needs and intentions. Moreover, the pervasive computing area has stimulated several research results on *context-awareness* [20] which are of special interest in view of mobile grid future development.

Currently, the pervasive computing is used mainly in applications, for example, smart house, smart office, smart hospital, and so on, that are much simpler than the ones we are interested on. In particular, the issue of programming models has neglected the aspects of large composition of high-performance components, high-performance and adaptivity for data and computation intensive services, uniform approach to communication and computation.

Current pervasive computing projects favor the infrastructure approach based on some Middleware [21–23].

### *2.2.3. Programming model- versus pure middleware-based approach*

Many grid infrastructures and pervasive computing infrastructures are based on a middleware, offering a set of services, without relying on a programming model but merely offering the possibility to link services to applications [24] (e.g., according to the Web Service model).



DOI: 10.1002/wcm

Existing solutions to adaptivity for pervasive and mobile computing extend the original definition of middleware [25] with concepts that can be used to support adaptivity. In particular, computational reflection [26] can be used by applications to get information about its implementation and possibly modify it to adapt to the actual conditions of the context. This is, in more general terms, the mechanism through which context awareness is expressed.

Middleware-based solutions span from extensions to traditional ones (e.g., OpenCorba [27] and dyanamicTAO [28]), which add some reflection mechanisms, to *ad hoc* solutions for mobile applications. The general idea behind the latter solutions is to include adaptive mechanisms inside a middleware and to offer them as services to the applications. For instance, in Odissey [29] the resource control and management is completely demanded to the middleware level. An application can request the middleware to monitor a given set of resources, by passing it some kind of 'confidence' values: the middleware notifies the application of the violation of these values. In the case of notification, an application can request the middleware to modify the policy of usage of the related resources.

Similar adaptation logics can be expressed in the CARISMA middleware [30]. Whenever an application requests for a service, it can express a policy which is used by the middleware to select proper resources. The policy dictates which service is to be used (or which configurations of a same service) depending on the monitoring of resources.

A different approach is described in [31]. In this case applications are built up of several services, whose implementation cannot be modified. Adaptation happens in the way in which whose are composed. Services composing an application can be dynamically modified to meet general rules, which are specified as *laws* (i.e., general constructs) and instantiated to coordination contracts.

The discussion of the previous sections has shown that the approach based on middleware is not adequate for mobile grid applications, mainly because:

- it cannot allow the programmer to design suitable QoS adaptive and context-aware strategies at the various application layers,
- it is not predisposed to express high-performance computations running across all the system nodes,
- it poses serious programmability and portability problems, which implies high designing and development costs.

By replacing the old-fashion OS-like view—according to which the application development occurs directly on top of the middleware—by the view centered upon the programming model, we wish to stress the programming model-based approach to system design, and, at the same time, to minimize the amount and variety of functionalities that are present in the underlying GAM: that is, these functionalities must be limited just to the support for the programming mechanisms and tools used to build adaptive and high-performance applications.

Our approach is more similar to [26], in which adaptation mechanisms are completely mapped at the application level, while the supporting environment only delivers resource information through reflective mechanisms. In fact, we are investigating how to integrate these concepts into the mobile grid programming model of In. Sy. Eme.

#### 2.2.4. Multiprocessor on chip and network on chip technology

Recent advances in hardware component technology have stressed the transition from very sophisticated superscalar/multithreaded CPUs, with a limited amount of processing cores, to multicore chips with a parallel internal architecture (shared/distributed memory) at core level [32], where core have a simpler architecture (e.g., in-order pipelined execution) and characterized by much lower power consumption compared to the current workstation processors.

The mobile grid architectural and programming model we propose is suitable to exploit this technological revolution in computing and communication structures.

Multicore technology offers very powerful perspectives not only to general-purpose parallel machine, but mainly to embedded systems, mobile, and wearable devices: in the next future they will be equipped with multiprocessor on chip, or architectures based on several multicore chips. This trend will allow mobile software components to be executed with much higher performances than today, thus allowing mobile grids to exploit their true potentials for large scale, heterogeneous, distributed and high-performance applications, without assuming the presence of traditional central servers as a critical resource. This important trend is also studied in the context of interconnection structures, giving rise to high-performance, low power consumption networks on chip [33]. In the near term, large-scale ICT architectures will be built by assembling multiprocessor on chip and networks on chip.



DOI: 10.1002/wcm

One of the most significant limitations of multicore technology is the lack of suitable programming tools for the machine-independent design and development of parallel programs too:[§] if parallelism in computer architecture will become the rule, instead of the exception, then parallel programming tools and environments must exit from the current phase of limited applications and must be considered at the same level of commodity software development frameworks.

### 2.3. Mobile Grid Programming Model

Our mobile grid approach is focused on the existence of a high-level, high-performance programming model, and related development tools. Applications are expressed entirely on top of this level.

As shown in Figure 2, the programming model is supported by the level denoted by GAM: it includes the kind of functionalities to support the preparation, loading, and execution of the applications, which should be provided by the middleware tools and services.

#### 2.3.1. *Layered view of the applications*

In the *level structure* of the system, shown in Figure 2, the GAM implements the programming mechanisms provided by the model, in a way that is invisible to the programmer. Traditionally (i.e., in a general-purpose system conceived as a sort of 'job shop'), a level structure is accompanied by a system view based on *opaque layers,* where an entity at some layer adopts a 'best effort' strategy in a fully independent way with respect to the entities at the other layers.

However, this view is not suitable for the applications of mobile grids: here, the requirements for dynamic QoS management (through adaptivity and context-awareness) imply that the programmer must have a proper *visibility* of the system layers of the application. Notice that the term 'layer' refers to the application, while the term 'level' refers to the system architecture: at some system levels we have a proper view of one or more application layers.

For example, in the mobile grid approach, at the programming model level the user has not only the abstract view of the computation components, but also a certain view of the way in which the system resources are managed to achieve the required QoS dynamically.

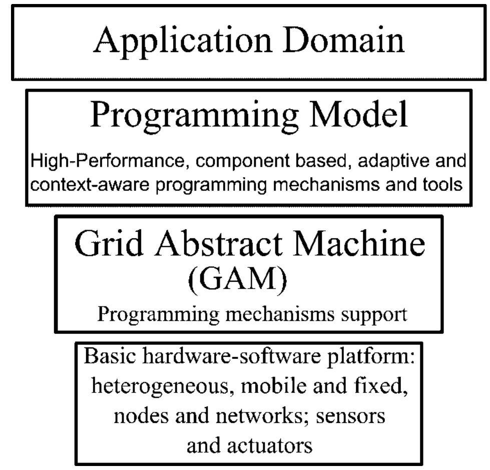



Fig. 2. System levels and programming model.

In the design of a mobile grid, the choice of the programming mechanisms is a critical issue and it is strongly influenced by the proper view of system resources and QoS policies. An entity A at a certain application layer is not simply defined according to independent, best effort strategies; instead its behavior is conditioned/fired by the requirements of entities B at different layers. For example, to be performed with the required QoS, B needs to establish a certain 'contract' with A dynamically, thus A must be programmed to have the proper visibility of B.

Of course, this concept of layer visibility must be not in contrast with the principles of structured and modular design of systems and applications, otherwise it could lead to too low-level and tricky programming mechanisms.

In respect of the principles I–V stated in Section 2.1.2 our proposal is the layered model of applications illustrated in Figure 3.

The model consists of three application layers,[¶] from top to bottom:

- application structuring by means of *high-performance components composition* (called, the RED layer),

[§]Massimiliano Meneghin. PhD Thesis Proposal. University of Pisa.

[¶]We denote these layers as application layers to distinguish them from network layers (see Section 3).



DOI: 10.1002/wcm

- network of *application managers* (the BLUE layer),
- *context interface* (the GREEN layer).

At the RED layer, the application designer has a very abstract view of the application itself, just concentrating on the data structures and algorithms for expressing his/her problem, without an explicit view of resource management, adaptivity, and context awareness.

The BLUE layer is the heart of the application as far as adaptivity and context awareness strategies are concerned. At this layer, the application is viewed as a set of *managers* that are responsible for implementing such strategies dynamically. For this reason, managers must interact with the GREEN layer to acquire knowledge about the system context, and they must interact with the RED components in order to configure/reconfigure them according to the needed/required QoS. Moreover, the managers interact with each other, in order to have a global view of the context and of the computation, that is, in order to achieve a global QoS starting from separate, partial views. In Figure 3, a 'partitioned' scheme is illustrated, in which each manager has its own partial view of the context and its own part of computation to control. In fact, this partitioned model is the one currently studied. Fully shared model views, or intermediate partitioned-shared views, will be investigated and compared in successive phases of our research.

Managers must be *programmable* explicitly: we are defining how to design applications in a layered fashion. Thus, the manager functionality corresponds to a specific *construct, or set of constructs*, in the programming model. These constructs will allow the

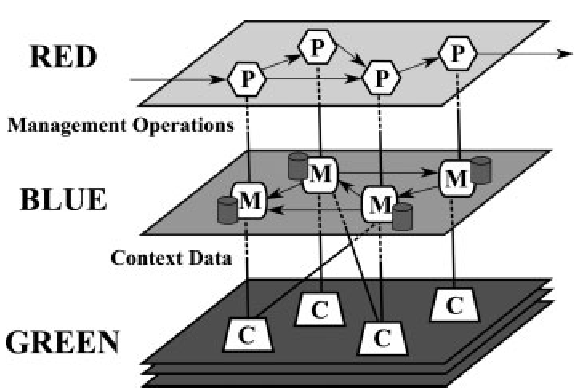



Fig. 3. Layered model of applications.

application designer to explicitly program adaptivity and context-awareness strategies, by taking into account the information sent by the context interfaces (events, state changes, monitoring of processing, and communication resource), and by providing to restructure the application layer according to the functional and non-functional properties of the application.

The implementation of the manager mechanisms is done at the GAM level. For example, if a mechanism consists in the allocation of a node to a RED component, the implementation of node allocation to that component is performed by the GAM: that is, the manager mechanism is the programmer-visible part of this functionality, while the GAM implementation is the invisible part.

The *context* is modeled and expressed as a proper layer (GREEN) of the application. It is used to spread information about the environment to be controlled (sensors, actuators, the same computing nods and communication lines) and about application modules (performance measures). It consists of a set of programming constructs for dealing with context objects and events in a structured view. All the system resources are properly abstracted at this layer: for example, the availability, connectivity, performance, load, memory occupancy, power consumption, and so on, of node and networks. That is, the implementation of the sensors, nodes, and networks is completely defined at the basic architecture level, and their applicative view is defined by a proper abstraction in the GREEN layer.

Dynamic properties of the context are monitored and are known to the managers by programmable interfaces, notably by specific programmable events. For example, the manager have some mechanisms to denote certain context events from sensors or nodes, to test their presence, to acquire related information, and to fire some programmed management actions on some RED components. Also the behavior of monitoring functionality can be abstractly defined at the GREEN level to be used by the managers.

The RED components operate on data which are present in some storage medium or on data dynamically acquired, filtered and classified by sensors, networks, and other virtualized resources.

Some remarkable features of this layered model are the following:

(1) *Integration of processing and communication activities*. At the BLUE layer the management of all the resources is programmed. Resources



DOI: 10.1002/wcm

include *networks* and the entire related infrastructure for transmission, synchronization, and communication control. In a sense, the BLUE layer includes the 'control plane' of network management, which, in our model, is performed through information exchange with the GREEN layer. Data streaming and algorithms for network planning are implemented at the RED layer, under the control of the BLUE layer. In other words, our approach integrates processing and communication activities according to a uniform model. This allows the designer to conceive applications as 'processing—communication chains' (see Section 2.1.2), without a fixed, *a priori* distinction between processing and communication; in fact, he/she can design novel, efficient strategies in which the processing and communication activities are combined and merged in the most suitable way for the application QoS. The same programming mechanisms are used, at the RED–BLUE–GREEN layers, both for processing and for communication. The flood management example above shows clearly this integration and uniformity. Section 4 will discuss this feature in detail.

(2) *Context-awareness*. Recently, this issue has been studied in the area of pervasive computing, and some formal programming models exists that exploit such property [34]. Other approaches are based on ontologies to express context entities and inference rules [35,36]. Our layered model of applications is context aware by definition: the GREEN layer allows the designer to give a suitable abstraction of the context, and the BLUE layer manages components and resources according to the context information. We are evaluating the possible exploitation of some results in the literature above. However, our model needs a more general and powerful approach to the formal description of context awareness for several reasons: variety of context information (from sensors, from services, from node, and networks monitoring), high-performance requirements, and dynamic deployment.

(3) *Dynamic deployment of application components*. Software components can be dynamically deployed (not only on central servers, but also) on decentralized nodes, embedded systems, wearable and mobile devices. In different phases of the application, and according to the context situation (normal monitoring and forecasting, detection of potential alarm events, crisis management, and so on), components and associated data can be dynamically created, modified, adapted, and deployed onto nodes with different characteristics.

(4) *Manager cooperation and robustness at the BLUE layer*. As previously said, we are studying various organization forms for the cooperation of managers, starting from the partitioned approach. The manager must also possess the property of robust and reliable behavior, which can be realized according to suitable schemes of replication [37,38] and/or according to a programming model approach to fault tolerance [39].

### 2.3.2. High-performance adaptive components

Our approach is based on the definition and realization of a programming model with the following features:

1. applications are expressed as compositions of *high performance components*,
2. a uniform approach is followed for distributed and parallel programming: in general *components* exploits internal parallelism and are executed in parallel with each other,
3. the strategies to drive the dynamic adaptation of applications are expressed in the same high-level formalism of the programming model.

A systematic approach, taking into account both high-performance and adaptivity, must be based on the existence of a rigorous *cost model* (performance model). This issue represents one of the main features of the so-called *structured parallel programming approach* [40]. Notably, in the *skeletons* approach [41] any parallel application is structured as a composition of known parallel paradigms (i.e., the programming skeletons) with known cost models. For example, pipelines, farms or divide, and conquer are typical task-parallel (stream-parallel) paradigms, while map, reduce, prefix, scan, stencil are typical data-parallel paradigms. In structured parallel programming, a *coordination language* is adopted that acts as a metalanguage used to compose, possibly already existing, code modules expressed in any standard language (C, C++, Java, Fortran. MATLAB, etc.).

Another important feature of structured parallel programming relies in its capability to support a complete set of *fault tolerance* mechanisms and strategies [39].



DOI: 10.1002/wcm

In our programming model we adopt structured parallel programming as a way to specify the strategy for structuring and for restructuring a component or a composition of efficient and robust components.

An intensive experience with structured parallel programming and adaptivity in parallel programs have been done, in recent years, with the *ASSIST* programming environment, a research product of the Department of Computer Science at University of Pisa [42]. ASSIST has been used successfully in the design of complex applications, notably image processing, earth observation systems, computational chemistry, MPEG parallel encoder, data mining, knowledge discovery and profiling. Beyond the 'classical' skeletons, the ASSIST programming model contains several features (graphs, parallel modules, and external objects) that sensibly increase flexibility and expressive power, including the possibility to design adaptive program structures [43]. These features can be properly merged with one or more standards for component technology, as it has been done in the Grid.it national project [44] and in the CoreGRID European network of excellence [4,5,16], and in the GridComp European project [17].

### *2.3.3. ASSISTANT*

*ASSISTANT* (*ASSIST with Adaptivity and Context Awareness*) represents the framework in which we study the issues of mobile grid programming models in the context of the *In.Sy.Eme.* project. It is based on the ASSIST experience, which is the starting point to define a new programming model to realize high performance computations with adaptive and context-aware behaviors.

In ASSISTANT an application can be described according to the three layers of Figure 3, each representing a different view of the whole application. In the RED layer we define the functional behavior of applications as a composition of ASSIST *parallel modules*. The BLUE layer represents the management part of applications and it implements adaptive behavior of RED parallel modules: this requires a new set of programming constructs, thus available to the programmer, to design adaptive and context-aware reconfigurations for the associated parallel modules. We distinguish two kinds of reconfigurations:

(1) a *transformation of the parallel executable version of the program*, such as a different degree of parallelism and/or a different data distribution and partitioning,
(2) *alternative versions of the sequential algorithms for the same functionality*, that is, a different implementation of the same sequential component.

We speak of *non-functional adaptivity*, or *performance adaptivity*, in case (1), and of *functional adaptivity* in case (2). In both cases, components can be defined and deployed statically or dynamically.

The formalism to explicitly program the managers is a set of *adaptation policy rules* which correlate particular context situations (obtained interpreting context interface data) with corresponding functional and non-functional reconfigurations. The program of a manager is composed of a set of relations between events and adaptation policy rules.

**Example.** We describe a simple adaptive application which performs a sub-part of a forecasting computation, that is, a flood forecasting model as MIKE 21 [10] briefly introduced in Section 2.2. The computation is applied to a specific geographical area of interest, assuming that a set of users requested it. We consider a stream parallel computation (Figure 4) structured as a pipeline of two ASSISTANT adaptive parallel modules. The first module, called *filtering module*, performs filtering of input data. It obtains input data from context providers, such as sensors (flow level, surface height, density of flow, wind speed, atmospheric temperature and pressure, etc.). The filtering module also solves the differential equations (e.g., mass and momentum equations in Mike 21) by means of a numerical method. In the case of MIKE21 the result is a large set of tridiagonal systems, which are passed to the second module. The *tridiagonal solver module* applies a direct method to solve these systems. The results are sent to a third set of modules (not shown in the figure), which post-process them for client visualization.

The size $N$ of the system of equations is a dynamic value: higher values of it induce a better discrete spatial representation and so the precision of the forecasting solutions. In Figure 4, we can observe the three-layered view of this simple application in which the two parallel modules have their own manager. In ASSISTANT we can program each module above at the RED and BLUE level. In this application the main reconfiguration activities are:

(1) for the *filtering module* the manager can dynamically restructure the size $N$ to target different



DOI: 10.1002/wcm

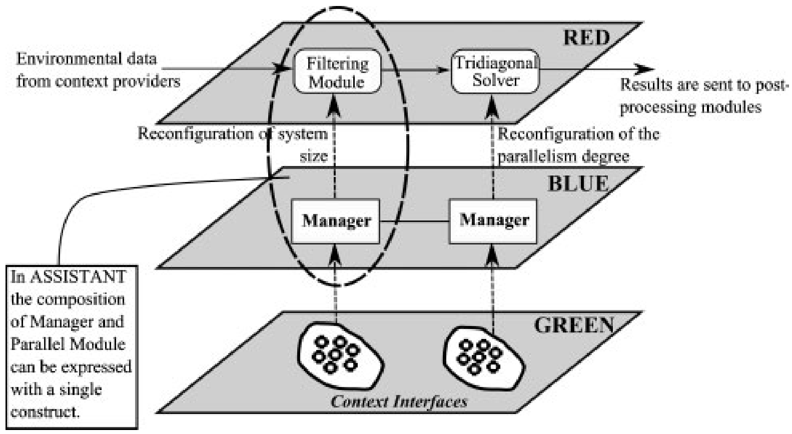



Fig. 4. Example of adaptive, high-performance application at several layers.

accuracy values of the forecasting model as a different grain of the spatial discretization of the interested area;

(2) the *tridiagonal solver module* executes a parallel implementation of a specific direct method to solve tridiagonal linear systems. The manager associated to it can restructure the module computation to target a different performance to respect some QoS contract. For instance, the user may indicate a specific maximum completion time for the forecasting computation over the interested area. The required completion time can be achieved, for instance, by upper bounding the maximum service time to solve each single system. The module manager can dynamically modify the parallelism degree of the solver module to increase its performance in order to target the requested service time.

The two managers must communicate to implement global reconfigurations over the whole application. For example, assume that the manager of the second module detects an abnormal emergency status in which the system solver must be executed using only the available mobile nodes (suppose that the central servers are not anymore reachable): in this case such manager can ask the first one to reconfigure the filtering module in order to reduce the forecasting precision. The goal is to produce systems of equations with a lower size. For instance this happens when the mobile nodes on which we perform the tridiagonal solving computation feature small memory sizes.

According to the definition of adaptation we described above, the point (a) is a function reconfiguration activity, whereas the point (b) is a performance adaptation activity.

A variant of the described behavior can consist in a different strategy for data acquisition. If the limited connectivity of the operator implies limited availability of updated data too, then the manager of the filtering module has to fire some predictive model for data and execute the computation in a proper RED component.

**A benchmark for adaptive behavior.** To demonstrate the adaptation features offered by our novel programming model, let us consider the following general benchmark consisting on a data- and compute-intensive computation (say, a known function $G$) on streams of input arrays $C[N]$, $D[N]$ and output array $E[N][N]$:

```
for i: = 0–N − 1 do
  for j: = 0–N − 1 do
    E[i][j]: = G(C[i], D[j])
```

The ASSISTANT implementation at the *RED layer* is a parallel module (parmod) able to work either in a task-parallel (farm) modality or in a data-parallel (map with replicated $D$ and scattered $C$) modality. The application manager (*BLUE layer*) reconfigures the module to switch from the task-parallel modality to the data-parallel modality and/or modifying the parallelism degree. The benchmark has been performed on a 32-workstation cluster (*Pianosa,* Pentium III 800 Mhz) at the Computer Science Department in Pisa. The initial $N$ value is set to 5000 integers so every $E$ matrix requires about 95 MB of virtual memory occupation. Figure 5(a) shows the worker virtual memory occupation: the functional reconfigurations are executed when we wish to migrate the application processes onto nodes with low-memory


*Wirel. Commun. Mob. Comput.* 2009; **9**:445–467
DOI: 10.1002/wcm

capacity (map version). In Figure 5(b) the adaptation of the average service time is illustrated in a situation where a maximum value of the service time is required by the user.

## 3. The Mobile Grid Effect on Wireless Communications

In this section, the vision of jointly considering the mobile grid model and the wireless network scenario will be introduced by focusing on the advantages of their joint approach and the challenges to be considered, by pointing out the attention on different elements of the communication system. In particular, we will focus on three different problems:

- data gathering from sensors and devices disseminated on the environment,
- localization technique allowing the system a more precise communication with specific devices,
- communication infrastructure used for dispatching processing units to the devices able to process them.

In Figure 6 the considered scenario, where a Mobile grid and the wireless communication networks operate in an integrated way, is shown.

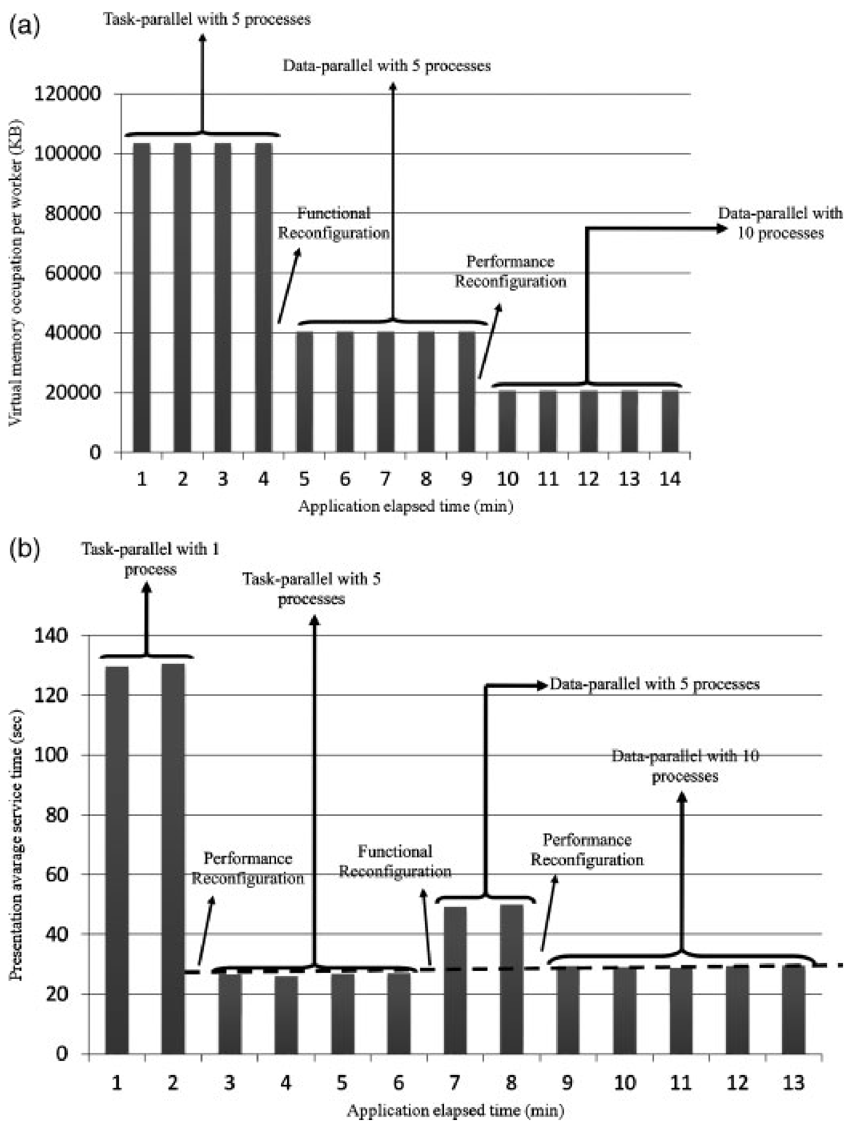



Fig. 5. Benchmarking results about ASSISTANT adaptivity. (a) Memory occupation results. (b) Average service time result.




DOI: 10.1002/wcm

### 3.1. Sensing and Data Gathering

The network scenarios on which the mobile grid network is focused on are characterized by the coexistence and cooperation of different heterogeneous technologies that include broadband wireless access, *ad hoc* networks and finally sensor networks to provide an efficient environment monitoring. This is of particular importance for our aims because sensor networks are often generating huge amount of data that needs to be processed before being human readable and interpreted by operators.

Wireless sensor networking (WSNing) represents an inherently disruptive approach specifically designed to detect events or phenomena, collect and process related data, and transmit sensed information to final users in a distributed way [46–49]. Although Wireless Sensor Networks (WSN) exhibit several features common to wireless *ad hoc* or mesh networks, as self-organizing capabilities or short-range broadcast communication with a multihop routing, they present additional constraints in terms of limitations in energy, transmit power, memory, and computing power. This implies that nodes constituting the WSN can participate to the mobile grid as a source of data while, mainly due to its processing limitations, can participate only in a limited way as a distributed part of the processing functionalities; this particular function will be better explained at the end of this section. By recalling the schematization introduced in Section 2.3.1, the WSN can be associated with the GREEN layer due to their capabilities of interacting with surrounding environment in terms of measurements.

Further, the operative conditions usually require cooperative efforts of sensor nodes in the presence of frequently changing topology due to fading and node failures or nodes mobility. A typical WSN is comprised of the following basic components:

- a set of distributed or localized sensors;
- an interconnecting network usually, but not always, wireless-based;
- a central point of information clustering;
- computing resources at the central point (or beyond) to handle data correlation, event trending, status querying, and data mining.

These elements allow a system administrator observing and reacting to events and phenomena in a specified environment. Presently WSNs have been largely focused on dense, small-scale homogeneous deployments to monitor a specific physical phenomenon. Nevertheless, the integration of multiple heterogeneous sensor networks operating in different environments could provides the ability to monitor diverse physical phenomena at a global scale.

In addition, such remote integration will make the infrastructure able to query and fuse data across multiple, possibly overlapping, sensor networks in different domains. Network links to back-end monitoring and collection systems may be intermittent due to weather or other problems, while in-network data storage is limited, leading to important observations being missed. If we think the WSN as a part of the aforementioned wireless mobile grid network, differently from other mobile or fixed devices a WSN is constituted by small devices with a very little computational capacity; this implies that the WSN nodes can be exploited to the a direct interpretation and processing of the data monitored. In that sense WSN are of great importance within a grid network mainly because they are one of the main parts for implementing it because the WSN is a generator of a huge amount of data to be interpreted and processed; on the other hand a WSN needs a more powerful data networks both in terms of bandwidth and processing power for processing data incoming from the WSN. However, a WSN is often constituted by several nodes with elementary capacity but fully connected among them. In that sense a WSN can be modeled as a processing cloud where each element has only a little computational capability but the whole network can perform a distributed computation. In particular, one of their most valuable contributions is toward a preprocessing of the data measured within the WSN before sending to the structures considered for processing purposes. This consideration, however, can be severely limited by certain circumstances in which the WSN operates,

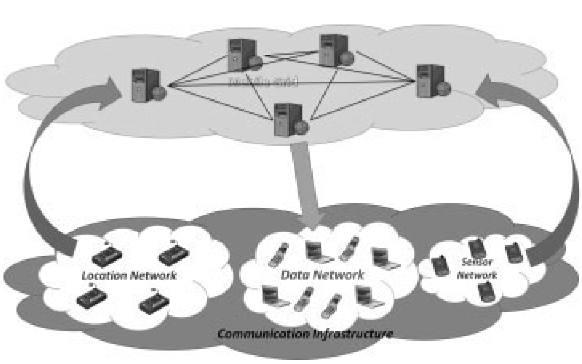



Fig. 6. Interconnection of the mobile grid with the communication infrastructure.



DOI: 10.1002/wcm

for example, limited battery power or sparse nodes in a harsh environment. In that sense the WSN can belong also the RED layer as an active component of the mobile grid.

### 3.2. Localization Techniques

The other important topic to be considered for implementing an efficient mobile grid network is concerning the ability of the system of localizing communication nodes and devices within the operative scenario [50–53]. This is important in a mobile grid mainly because this allows the system finding and sending directly to a certain device the amount of data to be processed as extensively introduced in the previous section. This characteristic is known also as geocasting [54]. This feature can be exploited at the BLUE application layer of the grid architecture for managing the component at the RED application layer. Indeed, the RED application layer can be seen as corresponding on those devices that needs to be located for a better exploitation. In order to meet the typical requirements of a mobile grid network, a localization protocol must be:

- robust to node failures;
- insensitive to measurement noise;
- low error in location estimation;
- flexible in any terrain.

Concerning the latter requirement, since devices forming a mobile grid networks can indifferently be both in indoor and outdoor environments an optimal localization approach has to be able to dynamically adapt to the channel features. Currently, two types of localization techniques address these challenges: beacon based and relative location based. Both techniques may use range and angle estimations for sensor node localization *via* received signal strength (RSS), time of arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA).

Localization methods suitable for applications in mobile grid network seem to be those based on beacons with a field containing the known sender position. This *ad hoc* inspired localization system requires that few nodes know their location (anchor nodes), for example, by means of satellite systems, like GPS or Galileo [51]. This allows nodes to discover their location through a two-phase process: ranging and estimation. During the ranging phase, each node estimates the distance from its neighbors. The estimation phase then allows neighbors to use the range estimated in the ranging phase and the known position of the anchor nodes to estimate their locations. To overcome the typical limitations of the range-based localization schemes, due to the outage of the underlying localization process, many range-free solutions have been proposed, basing on the location information hop-by-hop relayed from the source to the sink. These solutions estimate the location of sensor nodes or user terminals by exploiting the radio connectivity information among neighboring nodes or the sensing capabilities that each sensor node possesses, while in any case the sensor nodes must collaboratively work together to assist each other. Due to the specific characteristics of these approaches, the range-free localization can be divided into anchor-based schemes, which assume the presence of sensor nodes in the network with known position, and anchor-free schemes, which require no special sensor nodes for localization. The range-free localization schemes eliminate the need of high-cost specialized hardware on each sensor node. Since radio propagation characteristics vary over time and are environment dependent higher calibration costs for the anchor-based localization schemes are imposed. Basing on the fact that each type of localization protocols offers different capabilities, future sensor network applications are expected to relay on a combination of localization techniques.

### 3.3. Wireless Communication Infrastructure

#### *3.3.1. Physical layer impacts*

Whenever the behavior of a single network needs to be considered, one of the most important topics involves the physical network layer aspects of a telecommunication network, because they represent the requirement for guaranteeing a defined reliability level for the received data also under bad propagation conditions. This is of fundamental importance in wide wireless networks due to specific operative scenarios in order to allow efficient communications among users and support the implementation of a mobile grid processing structure to manage heavy computational duties or data processing. An efficient approach at physical layer permits a fast deliver of processing units to the grid elaborating devices.

One of the most important problems is devoted to the implementation of techniques able to achieve a reliable communications when devices are moving



DOI: 10.1002/wcm

within the environment; the definition of suitable techniques able to remove or limit the problems due to the non line of sight (NLOS) propagation and users mobility within the operative scenario have to be considered. In this context, efficient schemes for channel parameters acquisition and transmission/reception with MIMO systems (like WiMAX) [55] have to be envisaged. In particular, the MIMO approach allows the users to use at one's best the potentialities offered by the orthogonal frequency division multiple access (OFDMA) technique in terms of flexibility in the subcarriers allocation and modulation and power selection in transmission [56].

The aim of such approaches is to allow the system a significant performance increase respect to the traditional systems in terms of throughput and reliability of the transmitted information, that are essential requirements when the network has to support a mobile grid architecture. The advantages deriving from the use of the MIMO technique will be fundamental for defining also the efficient resource allocation techniques, considered in the following section.

In particular, the MIMO approach can be considered in terms of beamforming capabilities as well as spatial multiplexing. In a transmission system where beamforming is implemented by a MIMO scheme, the same signal is transmitted coherently through different antennas. Vice versa, if the antennas that form the array transmit incoherently, that is, independent coded signals, it is possible to exploit the fading independence among paths, hence obtaining at the receiving end a gain proportional to the number of transmitting/receiving antennas. This is particularly important for our aims because an efficient exploitation of MIMO techniques allow transmitting the resources to a particular node (i.e., a component of the grid architecture) that the BLUE application layer knows to have all the requested capabilities.

An extension of this concept is the cooperative use of antennas of different terminals (cooperative MIMO). This class of techniques extends the classical multiantenna schemes where multiple antennas were located in a single device to the exploitation of multiple devices for allowing a relaying of communication. In that sense the multiple antennas are spread over multiple devices. This concept permits to extend the coverage of actual networks and to improve performance. This could be useful for our aims because it makes it possible to manage components of the grid architecture also outside of the previous coverage area. This approach is consistent with the needs of a mobile grid architecture where the co-ordination of operators, systems and devices of different types in high ubiquity, heterogeneity, and adaptivity is foreseen.

#### *3.3.2. Resource allocation*

Whenever traffic belonging to mobile grid needs to be considered, it is of high importance the definition of suitable techniques for resource management in wireless broadband scenarios in order to allow the management of different traffic types with different QoS requirements and to guarantee to the distributed processing structure the possibility of an efficient management of the QoS in all the communication protocol layers and a high 'on-demand' adaptivity of the performance and functionalities in relation to the application context and the operator needs.

In particular, it is of crucial importance the development of optimization strategies jointly for physical and scheduling at MAC layer in order to maximize the system throughput for a predefined QoS level. This approach, named channel-aware scheduling, performs the channel allocation based on the awareness of the communication channel quality, by allowing a specific user to exploit specific transmitting resources with the aim of achieving certain target (e.g., maximize the throughput or minimize the error probability).

A particular attention is focused on the orthogonal frequency division multiplexing (OFDM) technology because it has the capability of mitigating frequency-dependent distortion across the channel band and simplifying the equalization in a multipath fading environment. The basic principle of OFDM is parallelization: by dividing the available bandwidth into several smaller bands that are called subchannels, the transmitted signal over each subchannel may experience flat fading. The properties associated with OFDM have led to its consideration as a candidate for high rate extensions to third-generation communication systems as well as for fourth-generation mobile communication systems.

To have more flexible and higher efficient OFDM systems, the adaptive OFDM schemes are adopted to maximize the system capacity and maintain the desired system performance. In particular, in an OFDM wireless system, the inherent multi-carrier nature of OFDM allows the use of link adaptation according to the behavior of the narrow-band channels: the bit-error probability of different OFDM subchannels, transmitted in time-dispersive channels, depends on the frequency-domain channel transfer function. This could be very useful in a mobile grid implementation



DOI: 10.1002/wcm

because allow reaching the different nodes (i.e., components of the grid) with a higher reliability that means a lower delay introduced by the communication infrastructure to the grid procedures.

Furthermore, in a multiuser OFDM wireless network, the given system resources are shared by several terminals and, in particular, in an OFDMA system disjunctive sets of subcarriers are allocated to different users, to provide a flexible multiuser access scheme [57]. The channel characteristics for different users are almost mutually independent in multiuser environments; the more attenuated subcarriers for a user may result not to be in a deep fade for other users, therefore an OFDMA wireless network may benefit from multi-user diversity dynamically assigning subcarriers with the best frequency response to the users. The design of link adaptation strategies can benefit of a cross-layer optimization approach (i.e., the joint de sign of the MAC and PHY network layers), that attempts to dynamically match the requirements of data link connections to the instantaneous available physical layer resources and to the channel state information [58–60].

The transmission parameters are selected at the physical layer to match the wireless channel, but they must take into account also the information coming from the higher levels, in particular from the MAC layer. Furthermore, in a multiuser OFDMA wireless network where the given system resources are shared by several terminals an adaptive subcarrier allocation strategy can significantly increases the system capacity by exploiting the multiuser diversity: the channel characteristics for different users are almost mutually independent; the more attenuated subcarriers for a user may result not to be in a deep fade for other users. Subcarrier allocation strategies dynamically assign subcarriers with the best frequency response to the users.

The OFDMA gives more flexibility when used within a grid environment allowing the managing of the components among the different nodes with a higher reliability.

One of the main problems that we have to consider is regarding the complexity of the system. The OFDMA-based systems are one of the most promising ones for respecting the requirements of the users, aiming on the next future to access to multimedia content in a mobile scenario. As highlighted before an OFDMA system allows a high flexibility at the cost of considering several parameters: on one hand each user suffers of different propagation conditions on the other hand each user can have different requirements in terms of QoS parameters (e.g., due to different traffic type the users want to use) [61,62]. The cross-layer approach needed to optimize the problem is faced in the literature by considering simplified approaches or by introducing simplified algorithm for approaching the optimal solution. In our case, the mobile grid could help in this task by allowing a fast parallel computation at the RED layer. In particular, by solving the problem with the use of specific algorithms could be possible to approach the optimal solution whose optimality degree could depend on the computation power of the devices used.

Aim of the optimization of resource allocation for a mobile grid network is to design and analyze the resource allocation techniques able to satisfy certain QoS requirements, such as fairness, delay, throughput, for each traffic type.

Due to brevity it is not possible to show here some numerical results, however, the interested reader could refer to [58–62].

### *3.3.3. Heterogeneous networking*

As mentioned earlier, one of the main advantages for implementing a mobile grid is to exploit a large amount of resources. From the communication infrastructure point of view a resource is an element of it, and belong, from the mobile grid point of view, to the RED application layer. The modern approach to wireless communications is constituted of several different devices each one with different capabilities and functionalities for the user that can be also connected with different communication protocols (Figure 7). This implies that one of the main problems to be solved in order to set up a mobile grid architecture over a wireless communication infrastructure is to exploit as much as possible devices, and so exploit at the best the heterogeneous networking techniques. Suitable solutions have to be considered, from both architectural and control algorithms point of view, allowing having a self-configurable network that allows the QoS management. When considering wide area networks, implemented with several heterogeneous communication standards, the time elapsed time during setup and the update of the network connections can be very long if those operations are implemented by using classical algorithms. The possibility of exploiting a mobile grid allows parallelizing the algorithms and so reducing the processing time. This is of fundamental importance in those environments where a fast setup of the network is needed. For example, let us think to an



DOI: 10.1002/wcm

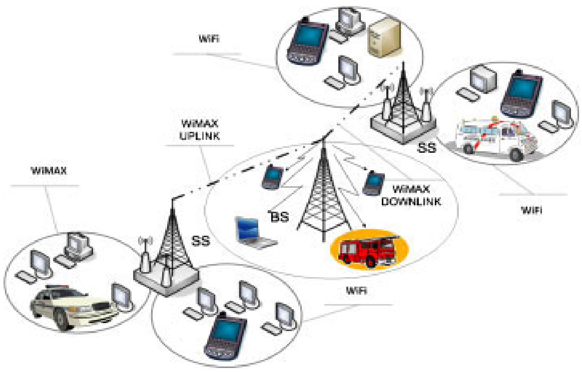



Fig. 7. A heterogeneous networking scenario including WiFi and WiMAX connections.

emergency scenario where multiple heterogeneous nodes need to form a network in a fast deployable way. In a normal context, this phase needs an amount of time that increase exponentially with the number of nodes to be connected. The RED layer of the mobile grid allows in our case to speed up this phase for achieving a fully connected network in a small amount of time.

Moreover, by recalling the resource allocation problem in OFDMA network introduced in Section 3.3.2, in case of heterogeneous network we have to face with a more complex problem due to the higher number of users and devices, with different communication characteristics. In this case, the optimal solution could be very difficult to be approached by exploiting a classical optimization technique even if simplified by using iterative algorithms; once again the mobile grid infrastructure could be exploited for optimizing the resource allocation in a more flexible and fast manner.

Beside the presence of second and third generation mobile networks (GSM, GPRS, EDGE, UMTS, and HSDPA), widespread all over the world and able to guarantee a coverage of a whole country, nowadays (or at least in the next future) it is possible to consider radio systems belonging to the IEEE 802.x family (e.g., IEEE 802.11, IEEE 802.16) that can be seen as a complement to the previous ones in not covered zone (digital divide) as well as a solution to be easily implemented in case of damage or malfunctions of the primary telecommunication infrastructures. This means that a large number of devices can be considered as a component of large whole wireless network and exploited for any grid application. Beside the popular IEEE 802.11 technology, it is easily to forecast that networks based on the IEEE 802.16 technology will be widespread in a short time. In an initial phase, it is reasonable to consider the use of IEEE 802.16 networks limited to backhauling connections between 2G and 3G base stations while point-to-multipoint (PMP) connections will be more used for fixed nodes where the wired connections cannot be deployed. Even if the MAC network layer of the standard IEEE 802.16 is designed for satisfying the QoS requirements, some issues are still open especially when a mesh topology is considered. Moreover, mainly because IEEE 802.16 connections will be used for high-speed data communications it is expected that devices using this type of connections will have a higher processing power.

One of the main advantages that can be achieved would be a seamless integration between heterogeneous networks [63]; the interworking optimization issue of different networks has been considered in the past when wireless access was not widely used. With the introduction of the modern wireless networks, the possibility of use jointly more than one of them has been often suggested in the literature, as for resource optimization as for wider area coverage. This could be interesting for supporting mobile grid applications allowing the management of the components of the grid architecture also considering the network type to which each one device is connected. In particular, a suitable approach for the interconnection among a WiFi and a WiMAX network is to resort to bridging solutions by taking into account two main goals: traffic priority and implementation issues.

As highlighted before, the WSN can participate better to the mobile grid if they are connected with other broadband connections; there must be a way for a monitoring entity to gain access to the data produced by WSN, or for exploiting the components of the WSN by the mobile grid. By connecting the sensor network to an existing network infrastructure such as the global Internet, a local-area network, or a private intranet, remote access to the sensor network can be achieved. Given that the TCP/IP protocol suite has become the *de facto* networking standard, not only for the global Internet but also for local-area networks, it is of particular interest to look at methods for interconnecting sensor networks and IP core networks. Sensor networks often are intended to run specialized communication protocols, for example, IEEE 802.15.4 or Zigbee, therefore an all-IP-network will not be viable, due to the fundamental differences in the architecture of IP-based networks and sensor networks. It is envisaged that the integration of sensor networks with the Internet will need gateways in most cases. A proxy server at the core network edge is able



DOI: 10.1002/wcm

to communicate both with the sensors in the sensor network and hosts on the TCP/IP network, and is thereby able to either relay the information gathered by the sensors, or to act as a front-end for the sensor network. It is also envisaged that sensing devices will be equipped with interfaces to wireless access networks such as 2/3G and WLAN enabling total ubiquitous connectivity.

The interested reader could refer to Reference [63] for some numerical results about the QoS aware interconnection of WiFi and WiMAX systems.

## 4. Features of the Integrated Approach

In this section, we outline some specific and relevant issues concerning the utilization of the layered model for mobile grid applications (Section 2.3.1, Figure 3) in the proposed integration of wireless communication strategies with computation strategies.

As said (Section 2.1.2), in our model a mobile grid application is conceived and implemented according to an explicit *composition of processing and communication activities*, without relying on a fixed, predefined scheme. The networking and communication strategies are considered *first-class citizens of applications*, in order to have complete control of QoS at all the systems levels. This view is applied at all the application layers: RED, BLUE, GREEN. Specific mechanisms are available in each application layer *to explicitly program communication strategies* and their proper composition with other computation activities of the whole application. This is needed to dynamically adapting the applications to changes in the context and user intentions.

### 4.1. RED Layer

Notable instances of processing-communication integration at the RED layer are:

1. several communication strategies can be explicitly programmed according to *high-performance parallel and distributed algorithms*, especially for scheduling, resource allocation, and heterogeneous networks. These algorithms, whenever facing with complex networks where multiple parameters have to be considered, could imply the solution of complex functions in a small amount of time, as highlighted in Section 3. In particular, when resource allocation in a multiuser environment has to be considered, the optimization problem needs high computational capabilities for achieving the optimal solution. Traditionally, some of these strategies are implemented, once for all, in commercial dedicated nodes/device through a best effort approach. Instead, more flexible and scalable solutions could be found by rendering such strategies explicitly programmed according to the application needs. We believe that this is a very promising research area, recently encouraged by first implementations in network processors [13,14]. Moreover, these high-performance networking strategies are/must be adaptive themselves, thus several versions may exist that are to be selected dynamically according to the management policies of the BLUE layer;
2. some components at the RED level are represented by the same devices taking parts to a computation, notably a WSN considered as an abstraction of a distributed computation of many elementary processes (Section 3). Such WSN components cooperate with other application components (e.g., filtering, fusion, and classification) in a way that is strictly related to the application semantics in order to optimize QoS. In the ASSISTANT example of Section 2.3.3, the data stream in input to the filtering module can be the proper RED abstraction of a sensor device or of a WSN.

Thus, an interesting and critical variety of communication strategies have to be explicitly programmed at the RED layer. For this purpose, we introduce the concept of *smart stream* in the ASSISTANT programming model, based on the following rationales:

1. in an ASSIST parallel and distributed program, components (parallel modules—called *parmod*s—or internally sequential modules) cooperate through typed *streams* of values;
2. streams are implemented according to a variety of algorithms and inter-process communication mechanisms (e.g., asynchronous channels, barriers, and collective communications) whose choice depends on many architectural aspects and, notably, on the underlying communication infrastructure. Thus, several run-time versions of streams must be provided in an ASSIST implementation. In a static application, the most suitable version is determined at compile-time. However, in a highly dynamic environments, like a mobile grid application, this choice must be driven at run-time: in our model, by the BLUE layer managers that allow the



DOI: 10.1002/wcm

RED components to select the most suitable stream implementation version;
3. the *smart stream* concept corresponds to an explicitly programmed component able to execute a set of communication strategies to be selected at run-time. In practical implementations, some default smart streams can be rendered available to the programmer. In general, smart stream is a more flexible concept and is configured according to the specific semantics of the application.

For example, the application segment described in Section 2.3.3 is programmed as a *pipeline of three stages*, as shown in Figure 8.

The first and the last stages are the filtering and the tridiagonal solver, and the intermediate stage is a smart stream programmed according to the specific needs of the data stream (large sets of tridiagonal systems) from the filtering to the solver modules. Several versions exist of this smart stream, depending on the characteristics of the node and network resources allocated to this stream, the required bandwidth and latency, the size $N$ of the system of equations, the parallelism paradigm selected for the filtering and for the solver module. These versions are explicitly programmed at the RED layer of ASSISTANT and are driven by the respective managers at the BLUE layer. For instance, switching from a 'normal connectivity' situation to a situation in which the operator connectivity is limited to a certain geographical area only, the smart stream version is changed to one that exploits at best the network and internetwork resources of such area.

## 4.2. BLUE Layer

At this layer, some notable features of the processing-communication integration are:

1. the integration between heterogeneous networks, in order to drive the dynamic selection of the network type according to the context events and to the RED application needs. As noted in Section 3.3 different networks can have very different communication characteristics; at the BLUE layer the manager can exploit the selection of the network type most useful for the processing needs among the available ones;
2. the dynamic selection of alternative versions of RED layer smart streams (see Section 4.1), as well as the allocation of suitable node and network resources, in particular, taking into account QoS issues of bandwidth, latency, availability, and fault tolerance of the RED components w.r.t. the specific needs of the application;
3. the special requirements for bandwidth, latency, availability, and fault tolerance of the BLUE components themselves (Managers) and their co-operation. That is, the BLUE layer must be able to schedule the available network types and configuration in such a way to take into account not only the requirements of the RED components, but also of the BLUE components themselves with proper priorities. As highlighted in Section 3.3, modern approach of wireless communication networks is to allow the management of different traffic types at the same time on the same network; this can be exploited for managing RED layer and BLUE layer traffic in a more suited way.

In ASSISTANT, these features are implemented by the manager mechanisms.

In the example of Figure 8, the co-operation of the three managers aims to allocating the processing and networking/internetworking resources in the operator's area so that the best QoS (e.g., service time) of the whole pipeline computation is achieved. In particular, the degree of parallelism of the filtering and solver modules is dynamically established to balance their service times according to the ratio between the computation time and the communication time obtained with the selected version of the smart stream module.

## 4.3. GREEN Layer

This is the layer in which the processing-communication integration is more evident, yet it must be defined in such a way that the GREEN components and their relationships with the BLUE components are *programmable*. In particular, we are interested in programming *context interfaces* related to semantics, quantity and quality of information and events:

1. from WSN and other devices (Section 3.1),
2. from monitoring devices and services (Section 3.2),
3. from networks performance measurements (Section 3.3).

ASSISTANT will contain specific mechanisms for these programmability issues.



DOI: 10.1002/wcm

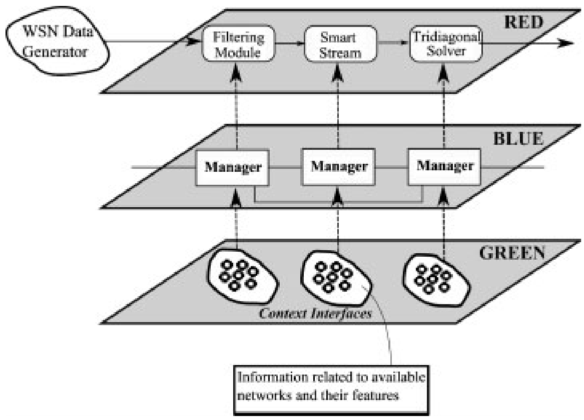



Fig. 8. Smart streams in the example application.

Notice that mobile devices and sensors are abstracted in two ways:

- as sources of context information, to be properly abstracted at the GREEN layer and used by the BLUE layer managers,
- as sources of data to be properly abstracted as streams of the RED layer components of the application, for example, the input stream of the computation of Section 2.3.3.

## 5. Conclusion

Next generation networks are considered one of the most interesting research and application field for the next future: in particular, integration between the communication infrastructure and the processing layer is one of the most promising approaches. This paper has presented a novel integrated approach between grid paradigm and wireless networks by highlighting the main advantages of their integration. In particular, it has been shown how a wireless heterogeneous network can be exploited for implementing a mobile grid, and, on the other hand, a mobile grid allows optimizing and interconnecting the several parts of the communication infrastructure in a novel way.

We have described and exemplified the features of our approach to mobile grid architectural and programming model. We are currently working on a first version of the ASSISTANT model. First experiences and demonstration have been done, and are been doing, through emulation in the existent ASSIST environment and other commodity software frameworks (Java, C++). Other experiences have been done in collaboration with the Italian Space Agency on Earth Observation Systems for disaster management [45]. This first set of results is quite encouraging to pursue the objective of a novel environment for mobile grid applications.

For what concerns the communication infrastructure it has been highlighted how it can be partitioned into three main parts when interacting with a mobile grid showing, in particular, how each part participates to the whole computation performed by the system.

## Acknowledgments

This work has been supported in part by MIUR-FIRB Integrated System for Emergency (In.Sy.Eme.) project under Grant RBIP063BPH. The overall approach to emergency management technologies has been planned, in the In.Sy.Eme. project, in collaboration with Francesco Tisato, University of Milano Bicocca, coordinator of the Workpackage on Domain Abstractions.

## References

1. Foster I, Kesselman C. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc.: San Francisco, CA, U.S.A., 2003.
2. NGG3, Future for European grids: GRIDs and service oriented knowledge utilities. *Vision and Research Directions 2010 and Beyond*. Next Generation GRIDs Expert Group, January 2006.
3. Travostino F, Mambretti J, Karmous-Edwards G. *Grid Networks: Enabling Grids with Advanced Communication Technology*. Wiley: New York, 2006.
4. Priol T, Vanneschi M (eds). Towards next generation grids. In *Proceedings of the CoreGRID Symposium*, 27–28 August 2007, Springer, Rennes, France.
5. Priol T, Vanneschi M (eds). From grid to service and pervasive computing. In *Proceedings of the CoreGRID Symposium* 25–26 August 2008, Springer, Las Palmas de Gran Canaria, Spain.
6. *IEEE 802.11*, 1999 Edition—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999 (and successive modifications and amendments).
7. McKnight LW, Howison J, Bradner S. Guest editors' introduction: wireless grids-distributed resource sharing by mobile, nomadic, and fixed devices. *IEEE Internet Computing* 2004; **8**: 24–31. DOI 10.1109/MIC.2004.14
8. Global Environment for Networking Innovations (GENI): Establishing the GENI Project Office (GPO) (GENI/GPO). http://www.nsf.gov/pubs/2006/nsf06601/nsf06601.pdf, 15 July 2008.
9. The Clean Slate Research Program. http://cleanslate.stanford.edu/index.php, 14 July 2008.
10. Warren IR, Bach HK. MIKE 21: a modelling system for estuaries, coastal waters and seas. *Environmental Software* 1992; **7**(4): 229–240.
11. Buzbee BL, Golub GH, Nielson CW. On direct methods for solving poisson's equations. *SIAM Journal on Numerical Analysis* 1970; **7**(4): 627–656. doi:10.1137/0707049

12. Gander W, Golub GH. Cyclic reduction—history and applications. In *Proceedings of the Workshop on Scientific Computing*. Luk FT, Plemmons RJ (eds). Springer Verlag: New York, 1997.
13. Carlstrom J, Boden T. Synchronous dataflow architecture for network processors. *IEEE Micro* 2004; **24**(5): 10–18. doi:http://dx.doi.org/10.1109/MM.2004.57
14. Venkatachalam M, Chandra P, Yavatkar R. A highly flexible, distributed multiprocessor architecture for network processing. *Computer Network* 2003; **41**(5): 563–586. doi:http://dx.doi.org/10.1016/S1389-1286(02)00450-4
15. Maes P. Concepts and experiments in computational reflection. *SIGPLAN Notices* 1987; **22**(12): 147–155. doi:http://doi.acm.org/10.1145/38807.38821
16. Dünnweber J, Gorlatch S, Campa S, Danelutto M, Aldinucci M. Adaptable parallel components for grid programming. *Integrated Research in Grid Computing*, CoreGRID series. Springer: Berlin, 2007. ISBN: 0-387-47656-3.
17. Aldinucci M, Campa S, Danelutto M, Vanneschi M, Dazzi P, Kilpatrick P, Tonellotto N. Behavioural skeletons in GCM: autonomic management of grid components. In *Proceedings of Euromicro PDP 2008: Parallel Distributed and network-based Processing*, IEEE, Toulouse, France, February 2008.
18. Weiser M. The computer for the 21st century. *SIGMOBILE Mobile Computing and Communications Review* 1999; **3**(3): 3–11. doi:http://doi.acm.org/10.1145/329124.329126
19. Satyanarayanan M. Pervasive computing: vision and challenges. *IEEE Personal Communications* 2001; **8**: 10–17.
20. Dey AK, Abowd GD. Towards a better understanding of context and context-awareness. In *Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness 2000*, ACM Press, New York.
21. Mascolo C, Capra L, Emmerich W. *Middleware for Mobile Computing*. In *Advanced Lectures on Networking. Lecture Notes in Computer Science*, Vol. 2497, Gregori E, Anastasi G, Basagni S (eds). Springer: Berlin, 2002; 20–58.
22. King J, Bose R, Yang HI, Pickles S, Helal S. Atlas: a service-oriented sensor platform: hardware and middleware to enable programmable pervasive spaces. In *Proceedings 31st IEEE Conference on Local Computer Networks* 2006; pp. 630–638.
23. Roman M, Hess C, Cerqueira R, Ranganathan A, Campbell RH, Nahrstedt K. A middleware infrastructure for active spaces. *IEEE Pervasive Computing* 2002; **1**(4):74–83, doi:http://dx.doi.org/10.1109/MPRV.2002.1158281.
24. Mokhtar SB, Preuveneers D, Georgantas N, Issarny V, Berbers Y. Easy: efficient semantic service discovery in pervasive computing environments with QoS and context support. *Journal of Systems and Software* 2008; **81**(5): 785–808. doi:http://dx.doi.org/10.1016/j.jss.2007.07.030
25. Emmerich W. Software engineering and middleware: a roadmap. *Proceedings of the Conference on The Future of Software Engineering ICSE '00*, ACM, New York, NY, USA, 2000; 117–129. doi:http://doi.acm.org/10.1145/336512.336542.
26. Adorni M, Micucci D, Tisato F. RA3: enabling adaptivity via a reflective architecture, In *Proceedings of the IASTED International Conference on Software Engineering as part of the 26th IASTED International Multi-Conference on Applied Informatics*, Innsbruck, Austria, February 2008, pp. 214–219.
27. Ledoux T. OpenCorba: a reflective open broker. In *Reflection*, Vol. LNCS 1616. Springer: Saint-Malo, France, 1999; 197–214.
28. Kon F, Roman M, Liu P, Mao J, Yamane T, Maes L, Cambpell R. Monitoring, security, and dynamic configuration with the dynamicTAO reflective ORB. In *International Conference on Distributed Systems Platforms and Open Distributed Processing* (Middleware), ACM/IFIP: New York, 2000; pp. 121–143.
29. Noble B. System support for mobile, adaptive applications. *IEEE Personal Communications* February 2000.
30. Capra L, Emmerich W, Mascolo C. CARISMA: context-aware reflective middleware system for mobile applications. *IEEE Transactions on Software Engineering* 2003; **29**(10): 929–945.
31. Andrade LF, Fiadeiro JL. An architectural approach to auto-adaptive systems. *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*. IEEE Computer Society: Washington DC, U.S.A., 2002; pp. 39–444.
32. Gschwind M, Hofstee HP, Flachs BK, Hopkins M, Watanabe Y, Yamazaki T. Synergistic processing in cell's multicore architecture. *IEEE Micro* 2006; **26**(2): 10–24.
33. Bjerregaard T, Mahadevan S. A survey of research and practices of network-on-chip. *ACM Computer Survey* 2006; **38**(1): 1. doi:http://doi.acm.org/http://doi.acm.org/10.1145/1132952.1132953.
34. Henricksen K, Indulska J. Developing context-aware pervasive computing applications: models and approach. *Pervasive and Mobile Computing* 2006; **2**(1): 37–64.
35. Serrano JM, Serrat J, Galis A. Ontology-based context information modelling for managing pervasive applications. In *Proceedings of the International Conference on Autonomic and Autonomous Systems* 2006.
36. Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review* 2003; **18**(3): 197–207. doi:http://dx.doi.org/DOI:10.1017/S0269888904000025
37. Romano P, Rughetti D, Quaglia F, Ciciani B. APART: low cost active replication for multi-tier data acquisition systems. In *Proceedings of 7th IEEE International Symposium on Network Computing and Applications* (NCA) September 2008, IEEE Computer Society Press, Cambridge, MA, U.S.A.
38. Bertolli C, Coppola M, Zoccolo C. The co-replication methodology and its application to structured parallel programs. *CompFrame '07: Proceedings of the 2007 symposium on Component and framework technology in high-performance and scientific computing*, ACM, New York, NY, U.S.A., 2007; pp. 39–48. doi: http://doi.acm.org/10.1145/1297385.1297393
39. Bertolli C. Fault tolerance for high-performance applications using structured parallelism models. *PhD Thesis*, University of Pisa, 2008.
40. Vanneschi M. PQE2000: HPC tools for industrial applications. *IEEE Concurrency* 1998; **6**: 68–73.
41. Cole M. Bringing skeletons out of the closet: a pragmatic manifesto for skeletal parallel programming. *Parallel Computing* 2004; **30**(3): 389–406. doi:http://dx.doi.org/10.1016/j.parco.2003.12.002
42. Vanneschi M. The programming model of ASSIST, an environment for parallel and distributed portable applications. *Parallel Computing* 2002; **28**(12): 1709–1732.
43. Vanneschi M, Veraldi L. Dynamicity in distributed applications: issues, problems and the assist approach. *Parallel Computing* 2007; **33**(12): 822–845. doi:http://dx.doi.org/10.1016/j.parco.2007.08.001
44. Aldinucci M, Campa S, Coppola M, *et al*. Components for high performance grid programming in grid.it. In *Proceedings of the International Workshop on Component Models and Systems for Grid Applications*, Getov V, Kielmann T (eds). CoreGRID, Springer: Saint-Malo, France, 2005; pp. 19–38.
45. Guerriero L, Intelligente F, Lorè VA, Milillo G. Analysis of algorithms for SAR data processing: an approach in the time domain. *Geoscience and Remote Sensing Symposium*, Vol. 5, 2005; pp. 3337–3340.
46. Hać A. *Wireless Sensor Networks Designs*. John Wiley & Sons: Chichester, West Suxxex, England, 2003.
47. Akyildiz F, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *IEEE Computer Networks* 2002; **38**: 393–422.



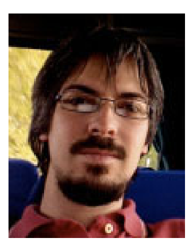
DOI: 10.1002/wcm

48. Ilyas M, Mahgoub I. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press: Boca Raton, USA, 2005.
49. Chiti F, Fantacci R. Wireless sensor network paradigm: overview on communication protocols design and application to practical scenarios. *EURASIP Newsletter* 2006; **17**(4): 6–27.
50. Chiti F, Fantacci R, Menci S, Zappoli A. Cooperative localization protocols for wireless sensor networks. In *Proceedings of the IEEE GLOBECOM '07*, 26–30 November 2007; pp. 1048–1052.
51. Tsui JB-Y. *Fundamentals of Global Positioning System Receivers, a Software Approach*. Wiley & Sons: New Jersey, USA, 2005.
52. Sichitiu ML, Ramadurai V. Localization of wireless sensor networks with a mobile beacon. In *Proceedings of IEEE MASS 2007*, 25–27 Oct. 2004; pp. 174–183.
53. Pathirana PN, Bulusu N, Savkin AV, Jha S. Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing* 2005; **4**(3): 285–296.
54. Ko Y-B, Vaidya NH. Geocasting in mobile ad hoc networks: location-based multicast algorithms. In *Proceedings of IEEE WMCSA '99*, 25–26 February 1999; pp. 101–110.
55. Nuaymi L. *WiMAX Technology for Broadband Wireless Access*. J. Wiley: Chichester, England, 2007.
56. Marabissi D, Fantacci R, Papini S. Robust multiuser interference cancellation for OFDM systems with frequency offset. *IEEE Transactions on Wireless Communications* 2006; **5**: 3068–3076.
57. Song G, Li Y(G). Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks. *IEEE Communications Magazine* 2005; **12**(12): 127–134.
58. Marabissi D, Tarchi D, Genovese F, Fantacci R. Adaptive modulation in wireless OFDMA systems with finite state modeling. In *Proceedings of IEEE GLOBECOM '07*, 26–30 November 2007; pp. 5210–5214.
59. Marabissi D, Tarchi D, Fantacci R, Balleri F. Efficient adaptive modulation and coding techniques for WiMAX systems. In *Proceedings of IEEE ICC '08*, 19–23 May 2008; pp. 3383–3387.
60. Marabissi D, Tarchi D, Fantacci R, Biagioni A. Adaptive subcarrier allocation algorithms in wireless OFDMA systems. In *Proceedings of IEEE ICC '08*, 19–23 May 2008; pp. 3475–3479.
61. Fantacci R, Tarchi D, Bardazzi M. Performance evaluation of the MAC protocol in IEEE 802.16 systems with data and VoiP traffic scheduling. *Wireless Communications and Mobile Computing*, in press. DOI: 10.1002/wcm.649
62. Tarchi D, Fantacci R, Bonciani E. On the ranging and scheduling of data traffic in OFDMA mobile environments. In *Proceedings of IEEE WCNC 2008*, 31 March–3 April 2008; pp. 1916–1921.
63. Fantacci R, Tarchi D. Bridging solutions for a heterogeneous WiMAX-WiFi scenario. *Journal of Communications and Networks* 2006; **8**(4): 369–377.


## Authors' Biographies

**Romano Fantacci** born in Pistoia, Italy, graduated from the Engineering School of the Università di Firenze, Florence, Italy, with a degree in Electronics in 1982. He received his Ph.D. degree in Telecommunications in 1987. After joining the Dipartimento di Elettronica e Telecomunicazioni as an Assistant Professor, he was appointed Associate Professor in 1991 and Full Professor in 1999. His current research interests are digital communications, computer communications, queuing theory, satellite communication systems, wireless broadband communication networks, *ad hoc*, and sensor networks. He has been involved in several European Space Agency (ESA) and INTELSAT advanced research projects. He is the author of numerous articles published in prestigious communication science journals. He guest edited special issues in IEEE Journals and magazines and served as symposium chair of several IEEE conferences, including VTC, ICC, and Globecom. Professor Fantacci received the IEE IERE Benefactor premium in 1990 and IEEE COMSOC Award Distinguished Contributions to Satellite Communications in 2002. He is currently serving as Associate Editor for *Telecommunication Systems, International Journal Communununications Systems, IEEE Transactions on Communununications*, and Area Editor for *IEEE Transactions on Wireless Communications*.

**Daniele Tarchi** was born in Florence, Italy, in 1975. He received the M.Sc. degree in Telecommunications Engineering and the Ph.D. in Informatics and Telecommunications Engineering from the University of Florence, Italy, in 2000 and 2004, respectively. He is now an Assistant Professor at the University of Florence, Italy. His research interests are in both data link and physical layers, with particular interests to resource allocation algorithms in wireless networks, link adaptation and adaptive modulation, and coding techniques, and MAC protocols for broadband wireless access. He has been involved in several national projects (Insyeme, rescue, pattern, and women) as well as European projects (Nexway, Newcom, Satnex, COST289). He is currently serving as Associate Editor for *IEEE Transactions on Wireless Communications* and has been reviewer of several technical papers submitted to journals and magazines.



**Carlo Bertolli** graduated magna cum laude in Computer Science at the Computer Science Department, University of Pisa, in 2004 with a second level curriculum on Computing Technologies and High Performance Enabling Platforms. Currently is young researcher of the Integrated Systems for Emergency (In.Sy.Eme.) project and research collaborator of the High Performance Computing group at the Department of Computer Science, University of Pisa. His research interests are in the area of high performance parallel computing with a special attention to model-driven approaches. During his Ph.D. thesis, entitled 'Fault Tolerance for High-Performance Applications using Structured Parallelism Models', he has investigated models and mechanisms to support fault tolerance for high performance applications. In the context of the CoreGRID network of excellence he has also investigated performance analysis issues for parallel computation in



DOI: 10.1002/wcm

presence of failures. He is author of scientific papers, in conferences and journals, in the area of parallel distributed computing.

**Gabriele Mencagli** graduated magna cum laude in Computer Science at the Computer Science Department, University of Pisa, in 2008 with a second level curriculum on Computing Technologies and High Performance Enabling Platforms. Currently he is a research collaborator in the MIUR-FIRB Integrated System for Emergency (In.Sy.Eme) project at the High Performance Computing Laboratory, Department of Computer Science. His research interest is in the area of high performance computing, parallel programming environments, and platforms. Presently he is dealing with the definition of a parallel programming model for high performance and context-aware applications continuing the research started with his graduation thesis.

**Marco Vanneschi** is full professor of Computer Architecture at the Department of Computer Science of the University of Pisa since 1983, where he is responsible of the High Performance Computing Laboratory. His current research and teaching activity is in the area of high performance computing, parallel processing models and programming environments, parallel and distributed computing platforms, and grid computing. In this area, he has co-ordinated several national projects and collaborations with national companies, including the PQE2000 Project in High Performance Computing, the ASI-PQE2000 project of Italian Space Agency on high performance earth observation applications, the SAIB project of MIUR-Athos Origin on high performance customer relationship management, the CNR strategic projects on grid computing, and the MIUR-FIRB project Grid.it. He has been and is member of several international committees, steering committees and working groups, as well as member of the editorial boards of international conferences and journals in the area of Parallel Computing and High Performance Computing. He is author of more than 200 scientific papers published in international journals and conferences and of three books on computer architecture and parallel programming, and he is scientific editor of eight international books, including the most recent books (co-editor Thierry Priol, INRIA) 'Towards Next Generation Grids' (2007) and 'From Grid to Service and Pervasive Computing' (2008).



DOI: 10.1002/wcm