

# SYSTEM DESIGN AND INTEGRATION IN PERVASIVE APPLIANCES

Manfred Glesner, Tudor Murgan, Leandro Soares Indrusiak,  
Mihail Petrov and Sujan Pandey

Darmstadt University of Technology, Darmstadt, Germany

INVITED PAPER  
MIDEM 2003 CONFERENCE  
01. 10. 03 - 03. 10. 03, Grad Ptuj

**Abstract:** ubiquitous or pervasive computing environment needs to offer an important amount of essential features like proactivity, transparency, ease-of-use, high-level performance and energy management, cyber foraging and surrogate request support, location and context awareness, scalability, and so forth. Such environment-correlated attributes pose significant requirements on the employed hardware platforms. This work highlights the emerging hardware design paradigms supposed to comply with those strict requirements. Prior to this, we give a brief insight in the main evolutionary steps of pervasive computing, analyse the primary characteristics of calm technologies, and point out emerging hardware architectures and design methodologies. As a case study, an approach for the integration of reconfigurable hardware and computer applications is discussed.

## Sistemska načrtovanje in integracija v pervazivnih sistemih

**Izvleček:** Povsodno in prodorno računalniško okolje naj ponudi pomembno število bistvenih lastnosti, kot so proaktivnost, transparentnost, enostavnost uporabe, visok nivo storitev in energijskega obvladovanja, odvisnost od lokacije in vsebine itd. Takšni atributi odvisni od okolja postavljajo pomembne zahteve za delovanje strojnih platform. V prispevku opisujemo prihajajoče načrtovalske zglede, ki naj bi ugodili vsem tem strogim zahtevam. Pred tem podamo kratek pregled pomembnih razvojnih korakov prodornega računanja, analiziramo osnovne lastnosti novih tehnologij ter opozorimo na prihajajočo arhitekturo strojne opreme in načrtovalske metodologije. Kot primer smo prikazali pristop k integraciji rekonfiguracijske strojne opreme in računalniške aplikacije.

### 1. Introduction

Mark Weiser imagined the forthcoming ubiquitous computing systems as specialized elements of hardware and software, connected by means of both wired and wireless technologies /1/. Eventually, such elements should gracefully melt into the environment and become so ubiquitous that no one will notice their presence. By weaving themselves in an indistinguishable and diffuse fashion into the everyday life, pervasive technologies allow users to focus on tasks rather than tools /2/.

On one hand, as technology shrinks and the maximum die size enlarges, integrating complete systems of continuously increasing complexity becomes possible /3/. Moreover, new materials like organic semiconductors and plastic lasers, progress in communication technologies (especially wireless ones), as well as enhanced pattern recognition capabilities due to increasingly performant sensors offer the possibility to develop platforms and appliances only dreamt of a couple of decades ago.

On the other hand, technology improvements bring with them several challenges and drawbacks regarding increased delay, higher time-of-flight, strict requirements in clock and power distribution, higher noise, associated capacitive and inductive effects, increased leakage, rela-

tive power consumption, and heat dissipation /4/. In order to cope with those problems, a multitude of design paradigms like reconfigurable architectures, platform based design, IP reuse, orthogonalization of concerns, and communication abstraction emerged during the last years.

In a world composed of advanced communication components, highly sophisticated sensors, smart pens and tabs, such gadgets and the design thereof have to comply with a multitude of characteristics discussed in the sequel like proactivity, transparency, ease-of-use, energy management, cyber foraging, context awareness, scalability, and so forth.

### 2. Computing ages evolution

The term of ubiquitous computing, more recently also called pervasive computing, is subject to an impressive amount of redefinitions and variations and thus, a terminology labyrinth emerged. Terms like calm technology, ambient intelligence, proactive computing, invisible computing, disappearing computing, augmented reality, sentient computing, smart dust depict one and the same thing stated by Weiser in his seminal paper /1/: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indis-*

*tinguishable from it. ... A good tool is an invisible tool. ... The tool does not intrude on your consciousness; you focus on the task, not the tool."*

## 2.1 The "Third Wave" of Computing

The first age of computing was the so-called *Main-frame Era*. At that time, each mainframe was shared by a group of people. The deserved group gradually reduced and a transition era marked by minicomputers fused with the second age of computing, that of the personal computer which allows a one-to-one connection user-PC. However, the PC is too attention demanding, too complex and hard to use by untrained or little trained users, isolating thus the user from people and activities. Consequently, the human became a peripheral of the computer and not vice-versa. Some drawbacks have been partly remedied through Internet and distributed computing. This second transition era is alleviating the evolution towards the third age of computing, the age of calm technology, the ubiquitous or pervasive computing era, in which the nowadays dominating desktop metaphor receives only a secondary importance and every person is served and attended by a gradually increasing number of embedded processors. Figure 1 depicts the evolution of the computing ages envisioned by Weiser.

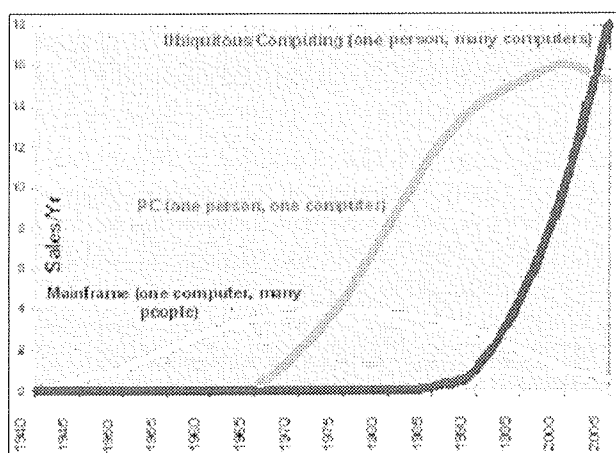


Figure 1: Evolution of computer ages (Source: Mark Weiser, <http://www.ubicomp.com>)

## 2.2 Related Research Fields

Satyanarayanan observed two distinct earlier steps in the evolution of pervasive computing, namely distributed systems and mobile computing /5/ as represented in figure 2. Thus, the technical problems induced by pervasive computing can be classified in two main categories: those related to the previous evolution steps which have been already identified - the solutions thereof can be easily mapped or adapted to the new needs; and those introduced by the ubiquitous computing paradigm requiring different solutions /2/.

Distributed systems emanated from the overlap of personal computers and local area network, belonging thus to the previously mentioned first transition era. Some of the

fundamental issues addressed in this field are quintessential for pervasive computing: remote communication, fault tolerance in transactions, remote information access (distributed file systems and distributed databases), and security (authentication and privacy) /5/.

The requirements of distributed systems enhanced with the appearance of a new degree of freedom, i.e. mobile clients. Consequently, some key constraints posed by mobility arose: unpredictable network quality variation and local resource restriction entailed by weight, size, and battery power consumption constraints. The research issues in mobile computing have been focusing onto the following areas: mobile networking, mobile information access, adaptive application support, high-level energy saving techniques, and location sensitivity /5/.

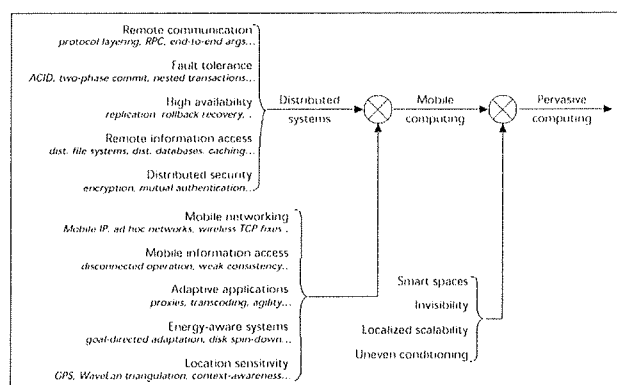


Figure 2: Related fields to pervasive computing (Source: M. Satyanarayanan /5/)

For a user not being acutely aware of the employed technologies, pervasive computing needs to subsume mobile computing. Nevertheless, the emphasis of pervasive computing lies in four further additional research directions identified by Satyanarayanan /5/: smart space use, localized scalability, invisibility and uneven conditioning masking.

A smart space can be a local area network, an infrastructure-enhanced room, or even a body area network - the so-called wearable computing environment /6/. While such smart spaces enhance their functionality, the degree of interaction with users increases, and thus easy scalable spaces are required. Not only must scaling be invisible, but also must the use of such spaces be user transparent, or at least minimally distracting. Finally, masking uneven conditioning refers to reducing the amount of variation seen by the users. Thus, the personal computing space counteracts against the low-capable space.

## 3. System design challenges

Since the late 1980s, when the first ubiquitous computing project started in the Electronics and Imaging Laboratory of the Xerox Palo Alto Research Center /7/, various pervasive computing projects based on several scenarios emerged /8, 9, 10/. Early industrial ubiquitous comput-

ing projects included the Active Badge from Olivetti and later Xerox PARC, the Active Bat – an ultrasonic-based location system of AT&T's emerged out of Olivetti's project, the CoolTown Project of Hewlett Packard, the Pathfinder GPS watch of Casio, or Lancaster's handheld city tour guide system. Lately, numerous universities have also initiated pervasive computing projects.

### 3.1 Features and Requirements

Every user will be immersed in a personal computing space, which is accompanying her or him everywhere. This smart personal space, actually a set of interacting intelligent devices is an invisible interface with the surroundings. Such a personal space should support some missing features like proactivity, user-intent recognition, context-awareness, inter-layer knowledge exchange, and seamless task motion among platforms /5/.

In order to shift the desktop metaphor in the background, appropriate physical interactions have to be defined. Implicit inputs like GUIs are replaced or at least enhanced by handwriting, tactile, speech, and motion recognition. Data will be send to or acquired by a collection of devices and sensors that can process it in various ways. With an augmenting number of devices in a personal smart space, it gets increasingly difficult to provide ease-of-use. For an easy-to-use multi-modal user interface it is crucial to find the right balance between proactivity and transparency. A non proactive gadget will be "dumb", while a highly proactive one might get annoying.

A pervasive device entering into a smart space must be able to detect services and potential surrogates. Thus, surrogates – resources belonging to the infrastructure of the smart space – can temporarily assist a portable unit and dynamically augment the resources of a wireless device.

Open research points deal with discovery of and level-of-trust setting up with surrogates, balanced surrogate loading, low-intrusive integration of surrogates. Additionally, adaptation is highly necessary whenever major disequilibria between the demand and supply of resources appear.

Among other features, context awareness is fundamental for developing minimally intrusive pervasive appliances. Context awareness builds upon location awareness, a problem tackled for example by services as GPS.

Privacy and security are already painstaking problems in distributed systems. Furthermore, due to the intrinsic interaction of personal computing spaces with other personal spaces or the fixed infrastructure, those issues are of increased complexity in pervasive computing /5/.

Smart spaces can deploy themselves into user transparent or at least user-friendly environments if crucial features like reliability, availability, maintainability, and scalability are present /11/.

Last but not least, it is worth noticing that the overall power consumption increases because of proactivity and self-tuning. Due to the continuous pressure to make devices smaller, lighter, and more independent, stringent requirements on battery capacity is posed /2/. Low power circuitry is hereby insufficient, and high-level power management techniques like energy-aware tuning (applications switch to less power hungry modes of operation when possible) and memory management, or user intend deduction and remote task execution, i.e. the use of computational surrogates to increase battery life.

### 3.2 Impact on Layering

The impact on layering of pervasive appliances is of paramount importance. The merging of information from different layers for proactivity, adaptation, and dynamic power-performance management seem to require much more information exchange between layers in ubiquitous computing environments than in typical systems up to present /5/. The layer decomposition is a very tedious task, and the classical information hiding principle might not prove to be an efficient approach regarding ubiquitous computing.

Based on a simplified OSI layer model, one can generally talk about three super-layers /12/: the abstract layer – dealing with services, environment and soft-ware issues; the resource layer – comprising middle-ware, network control, and protocols; and the physical layer – focusing on flexible platforms, electronic design automation, low power issues.

Although the hardware appears to become more reliable, the software seems to grow more fragile while getting more powerful /2/. Furthermore, novel services have to be provided by an ubiquitous computing environment /13/. New services have to be informed about the local context of the user through a collection of sensors. Moreover, those services have to be enabled to affect the environment through a set of actuators.

In an intelligent space, communication and networking build the heart of the pervasive computing environment /14,15,16/.

Generally, the identified key communication challenges refer to the heterogeneity and topology of the networks, to short-lived and intermittent connectivity, and to the evolution and up-grade of long-lived systems /15/.

There are two fundamental approaches for designing a communication network, which must be carefully analysed for pervasive computing environments: infrastructured – with base stations, and infrastructure less – mobile ad-hoc networking. The latter seems to be very attractive for smart spaces, but nevertheless, the challenges regarding routing, security, reliability, QoS, and so on, increase in complexity.

## 4. Hardware design issues

As previously mentioned, pervasive computing is technology driven. For almost forty years, silicon integration has actually followed a rule that started as a speculation, i.e. Gordon Moore's Law /17/.

### 4.1 Orthogonalization of Concerns

Driven by the simultaneous demand of both performance and flexibility, platform-based design and reconfigurable architectures are getting an increasing interest in recent times /18,19/. Thus, a remarkable variety of different platforms have been proposed to trade energy-efficiency, cost and performance (see /20/ for a survey).

The SIA Roadmap /3/ predicts for the future 50 nm technology the integration of more than 4 billion transistors on a single chip running at 10 GHz and operating below 1V. Under these circumstances, one of the most important limiting factors for system performance, die area, and power consumption will be generated by the on-chip interconnect networks. Not only functional modules, or so-called IP blocks, have to be reused, but also the communication architecture and the interfaces between such blocks must be standardised. Therefore, orthogonalization of concerns /18/ and communication centric design /21/ emerged as possible solutions to fill the design gap.

Abstracting the physical interconnections through Networks-on-Chip (NoC) based design, will offer the possibility to cope with problems like distributed traffic monitoring and control, unavoidable data failures on the physical layer, synchronization, scalability, re-use, reliability, global asynchronous and local synchronous (GALS) communication /22/.

### 4.2 Interconnection Structures of Reconfigurable Architectures

Driven by the simultaneous demand of both performance and flexibility, reconfigurable architectures (see /20/ for a survey) are getting an increasing interest in recent days. For example, in the DSP context, the capacity of these devices to select the more appropriate data path for computing a task represents a significant advantage in terms of performance and power efficiency. Different platforms have been proposed to trade energy-efficiency, cost and performance. In /23/ a template for a heterogeneous reconfigurable architecture is proposed and used to instantiate a wireless base-band. A generic platform is presented in figure 3.

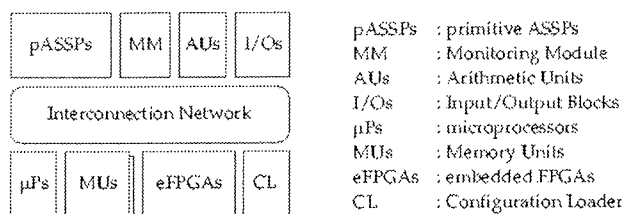


Figure 3: Reconfigurable template /30/

In both fine- and coarse-grained reconfigurable devices, the programmable interconnect architecture has a decisive influence in the total area, performance and power consumption of the system /24/.

To achieve high computational performance and flexibility, the different composing processing elements have to be richly interconnected. But since powerful interconnections imply large chip area and consume more energy, estimation procedures working at high level of abstraction are required to select the sufficient communication structure for a given application domain.

Several interconnection strategies for coarse-grained reconfigurable architectures have been reported. According to the degree of shareness, communication architectures can be classified into shared or dedicated structures. Dedicated architectures are generally composed by point-to-point like structures devoted to provide high performance communication of closed units. They use to be restricted to the first level of neighbours (e. g. *DReAM*, *KressArray*) or they can include a second level as in *MATRIX*. The shared interconnection structures can be divided into three categories: multi-bus (e.g. *RaPID*), crossbar (e.g. *PADDI-2*), and mesh; either regular (e.g. *DReAM*, *KressArray*, *MATRIX*, *Garp*) or irregular (*Pleiades*). A crossbar is the simplest way of interconnecting a given number of units and it guarantees full and arbitrary connectivity among elements. Thus, the mapping of any given network is attainable but the area requirement is of order  $O(n^2)$ . Consequently, for large crossbars, wire size dominates chip area.

The mesh structure reduces the total number of switches needed by limiting the connections to fixed distances. This type of network requires a smaller amount of area, but it makes distant communication slower and more expensive. This enforces a detailed performance analysis of the architecture before implementation /25/.

Another important communication issue is the connection between the reconfigurable computing elements and the host computer. There are two main approaches: either the reconfigurable unit is integrated into the processor core as a functional unit (e.g. *PRISC*, *CHIMAERA*); or the reconfigurable unit is placed as a coprocessor close to the host unit, sharing the cache unit (e. g. *Garp*, *REMARC*, *MorphoSys*). Although theoretically, it is also possible to place the reconfigurable unit as an attached processing unit outside the processor, the low performance of this solution made this solution impractical.

Further on, in /26/ the possibility of integrating switching network concepts within a reconfigurable architecture has been analysed. At the physical layer, the communication architecture is based on a simple point-to-point structure conforming a 2-D mesh of routers. The functionality of the data-link and network layers defines a packet protocol that can be used as a virtual connection between all the processing units of the configurable architecture.

### 4.3 Power Estimation and Optimisation

In portable appliances, batteries are a significant source of size, weight, and mechanical inflexibility [27]. There is still a significant need to improve and optimise batteries even though the energy of the human body is envisaged to be used for personal computing spaces. Nevertheless, power consumption must be reduced due to rapidly increasing thermal dissipation issues (packaging costs).

Benefiting from the non-uniformity of the workload in various signal processing applications, several dynamic power management policies have been developed [28]. Nevertheless, the integration of on-line power, performance and information-flow management strategies based on traffic monitoring in (dynamically) reconfigurable templates has yet to be explicitly tackled [29]. Similar strategies will be an in-separable part of pervasive devices [30].

A hierarchy of increasing complexity stochastic data models that effectively exploits the knowledge about the excitation of the system was developed in [31] order to estimate the power consumption in digital circuits by modelling the effects of the high level signal characteristics on the power consumption. The hierarchy of models includes a Gaussian ARMA model for linear systems, a cyclic multiplexing of Gaussians for folded and multiplexed architectures, a general uncorrelated model for memoryless non-linear architectures, and a general correlated model for non-linear circuits with reconvergent paths. Such techniques can also be integrated in the design flow of reconfigurable platforms.

## 5. Ubiquitous access to reconfigurable hardware

An approach for the integration of reconfigurable hardware and computer applications based on the concept of ubiquitous computing is presented in [32]. The goal is to allow a network of reconfigurable hardware modules to be transparently accessible by client applications. The communication between them is done at the API level, and a Jini-based infrastructure is used to provide an interface for the client applications to find available reconfigurable hardware modules over the network. A DES-based cryptography system was implemented as a case study.

The aim is to reduce the integration overhead of reconfigurable hardware modules and computer systems. Such overhead can be reduced by raising the level of abstraction of the integration architecture, allowing the communication to be done via message passing, as proposed in the object-oriented paradigm. By using this approach, each reconfigurable hardware module can be seen by the rest of the system as an object. Thus, it should be reconfigured and used through method calls. This can make a significant difference for the system designer, because he/she can abstract the internal details of the reconfigurable module - a typical result of the encapsulation feature of object-oriented systems - and can design the whole sys-

tem communication at the API level. In such approach, all the subsystems depending on the reconfigurable hardware module can call a configuration method to set up the desired functionality, and then call methods to pass the data to be processed and receive the results. Figure 4 depicts such possibility.

In order to cope with the demands of the current application scenarios - where the computation is performed by several interconnected appliances - one also has to support the integration of reconfigurable hardware modules into distributed computer systems. We can expect that each subsystem could be in a different location, connected to the others by a heterogeneous network. So, the approach should allow those subsystems to interact with any number of encapsulated reconfigurable hardware modules. The minimum infrastructure to do so comprehends distributed resource localization and remote method invocation. The first technique provides means for the distributed objects to locate other objects according to their needs, while the second one is responsible for the common dialect used by the objects to exchange messages once they have established communication.

Many of the applications of reconfigurable hardware can benefit from the proposed approach. For instance devices, which were already deployed - such as an ad-hoc network of sensors - could be located and upgraded by method calls if they have encapsulated reconfigurable hardware supported by an infrastructure for localization and remote method invocation. Another application scenario would be the use of reconfigurable hardware modules as accelerators for specific computational tasks. For instance, a mobile device, which needs to decode a stream of data and does not have the computational power to do so could use the resource localization feature to search for a reconfigurable hardware module which is able to decode the data stream.

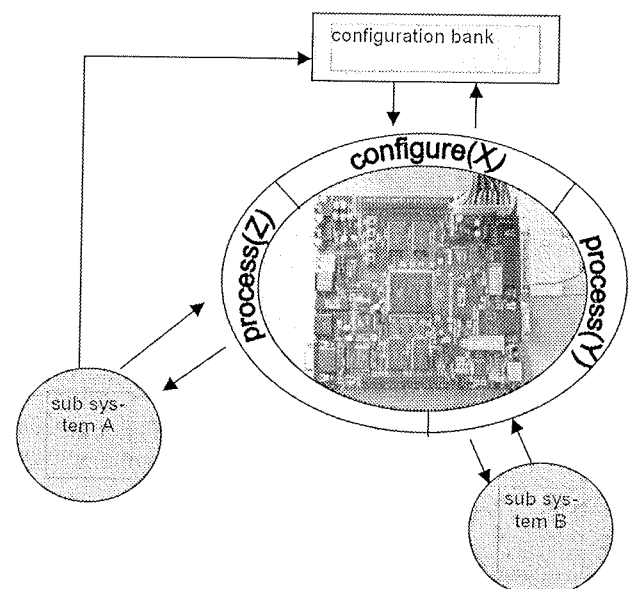


Figure 4: Reconfigurable hardware encapsulation

A third application could be found on the use of reconfigurable hardware as a prototyping platform. Let's imagine a system design specification done in the functional level. Once that specification fulfils the functional requirements, it should be submitted to successive synthesis steps in order to be implemented as a physical entity. Reconfigurable platforms are often used as an intermediate stage within such process, allowing system designers to verify the correctness of their designs prior to the final implementation. Our approach could provide a simpler way to integrate the functional specification with the prototyping platform, in such a way that they can interoperate. This would allow a mixture of simulation and emulation in the functional level, because one could synthesize and implement part of the functional specification in the reconfigurable hardware and still be able to perform the functional simulation, as the rest of the specification would communicate with the prototype in the same way it did before with the functional description.

Similarly to the prototyping platform scenario, a distributed IP core validation system could benefit from the proposed approach. In simulation systems a designer access remotely an IP core so it can be simulated together with the rest of the design. The approach could provide a layer between the IP core repository and the client, so the cores could be accessed seamlessly, without a previous connection to a predefined server. Another advantage would be the possibility of simulating an actual core implemented in a reconfigurable module, instead of the simulation models.

## 6. Conclusions

This work discussed the main features of pervasive appliances, described the evolution of computing ages, and highlighted the primary requirements and constraints posed especially on the hardware platforms and the design thereof.

Moreover, an approach for the integration of reconfigurable hardware and computer applications based on the concept of ubiquitous computing was discussed. The goal was to reduce the integration overhead of reconfigurable hardware modules and computer systems.

Even though pervasive computing is rather driven by technology, it is not to be successful only if technology does, but also if it can support and ease social life. By becoming a larger part of human existence, pervasive devices might promote but also inhibit social relationships [33], thus system designers should be ready to deal additionally with such non-technical constraints.

## 7. References

- /1/ Weiser, M.: The Computer for the 21st Century. *Scientific American*, 265(3):66-75, September 1991
- /2/ Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7):74-84, July 1993
- /3/ Allan, A.; Edenfeld, D.; Joyner, W. H. Jr.; Kahng, A. B.; Zorian, Y.: 2001 Technology Roadmap for Semiconductors. *IEEE Computer*, 35(1):42-53, January 2002
- /4/ Sylvester, D.; Keutzer, K.: A global wiring paradigm for deep sub-micron design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 242-252, February 2000
- /5/ Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, August 2001
- /6/ Starner, T.: Wearable Computing: No Longer Science Fiction. *IEEE Pervasive Computing*, pages 86-88, January-March 2002
- /7/ Weiser, M.; Gold, R.; Brown, J. S.: The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal*, 38(4): 693-696, 1999
- /8/ Want, R.; Schilit, B.: Expanding the Horizons of Location-Aware Computing. *IEEE Computer*, pages: 1-4, August 2001
- /9/ Want, R.; Pering, T.; Boriello, G.; Farkas, K.: Disappearing Hardware. *IEEE Pervasive Computing*, pages 36-47, January-March 2002
- /10/ Abowd, G. D.; Mynatt, E. D.; Rodden, T.: The Human Experience. *IEEE Pervasive Computing*, pages: 48-57, January-March 2002
- /11/ Hennessy, J.: The Future of Systems Research. *IEEE Computer*, pages: 27-33, August 1999
- /12/ Ciarletta, L.; Dima, A.: A Conceptual Model for Pervasive Computing. Intl. Workshop on Parallel Processing, Toronto, Canada, 21-24 August 2000
- /13/ Kindberg, T.; Fox, A.: System Software for Ubiquitous Computing. *IEEE Pervasive Computing*, pages: 70-81, January-March 2002
- /14/ Boriello, G.: The Challenges to Invisible Computing. *IEEE Computer*, pages: 123-125, November 2000
- /15/ Boriello, G.: Key Challenges in Communication for Ubiquitous Computing. *IEEE Communications Magazine*, pages: 16-18, May 2002
- /16/ Estrin, D.; Culler, D.; Pister, K.; Sukhatme, G.: Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, pages: 59-69, January-March 2002
- /17/ Boekhorst, F.: Ambient intelligence, the next paradigm for consumer electronics: how will it affect silicon? IEEE International Solid-State Circuits Conference. Digest of Technical Papers. pages: 28 -31, vol.1, 3-7 Feb. 2002
- /18/ Keutzer, K.; Newton, A. R.; Rabaey, J. M.; Sangiovanni-Vincentelli, A.: System-Level Design: Orthogonalization of Concerns and Platform-Based Design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 19(12): 1523-1543, December 2000
- /19/ Rabaey, J.: Reconfigurable Processing: The Solution to Low-Power Programmable DSP. In Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, pages 275-278, April 1997
- /20/ Hartenstein, R.: A Decade of Reconfigurable Computing: A Visionary Retrospective. In Proc. of the DATE Conference, pages: 642-649, 2001
- /21/ Sgroi, M.; Sheets, M.; Mihal, A.; Keutzer, K.; Malik, S.; Rabaey, J.; Sangiovanni-Vincentelli, A.: Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design, In Proc. of the DAC, pages: 667-672, 2000
- /22/ Benini, L.; De Micheli, G.: Networks on Chips: A New SoC Paradigm. *IEEE Computer*, pages: 70-78, January 2002
- /23/ Zhang, H.; Prabhu, V.; Rabaey, J.; et al. A 1-V heterogeneous reconfigurable DSP IC for wireless baseband digital signal processing. *Journal of solid state circuits*, pages 1697 - 1704, November 2000.
- /24/ Rabaey, J.: Reconfigurable Processing: The Solution to Low-Power Programmable DSP. In IEEE Intl. Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 275-278,

- April 1997
- /25/ Zhang, H.; Wan, M.; George, V.; Rabaey, J.; Interconnect Architecture Exploration for Low-Energy Reconfigurable Single-Chip DSPs, IEEE Computer Society Workshop on VLSI'99, April 1999, Orlando, Florida
- /26/ Michael B. Taylor, Jason Kim, et al. The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *IEEE Micro*, 22, March-April 2002.
- /27/ Hahn, R.; Reichl, H.: Batteries and power supplies for wearable and ubiquitous computing, The Third International Symposium on Wearable Computers, Digest of Papers, pages: 168 - 169, 18-19 Oct. 1999
- /28/ Benini, L.; De Micheli, G.: System-Level Power Optimization: Techniques and Tools, *ACM Trans. on Design Automation of Electronic Systems*, 5(2): 115-192, April 2000
- /29/ Benini, L.; De Micheli, G.: Powering Networks on Chips: Energy efficient and reliable interconnect design for SoCs. In Proc. of the 14th International Symposium on System Synthesis, pages: 33-38, 30 Sept.-3 Oct. 2001
- /30/ Murgan, T; Petrov, M; Garcia Ortiz, A.; Ludewig, R.; Zipf, P.; Hollstein, T.; Glesner, M.; Oelkrug, B.; Brakensiek, J; Evaluation and Run-Time Optimisation of On-Chip Communication Structures in Reconfigurable Architectures, In Proc. of the Intl. Conference on Field Programmable Logic and Applications, Lisbon, Portugal, 1-3 September 2003
- /31/ Garcia Ortiz, A.; Stochastic data models for power estimation at high-levels of abstraction, Ph. D. Thesis, Darmstadt University of Technology, June 2003
- /32/ Indrusiak, L.; Lubitz, F.; Reis, R.; Glesner, M.; Ubiquitous access to reconfigurable hardware: application scenarios and implementation issues. In Proc. of the Design, Automation and Test in Europe Conference and Exhibition, pages: 940-945, March 3-7, 2003
- /33/ Dryer, D. C.; Eisbach, C.; Ark, W. S.: At what cost pervasive? A social computing view of mobile computing systems. *IBM Systems Journal*, 38(4): 652-676, 1999

Manfred Glesner, Tudor Murgan, Leandro Soares  
Indrusiak, Mihail Petrov and Sujan Pandey  
Darmstadt University of Technology  
Karlstr. 15, D-64283 Darmstadt, Germany  
{glesner, murgan, lsi, pmihail, spandey}  
@mes.tu-darmstadt.de

Prispelo (Arrived): 15.09.2003 Sprejeto (Accepted): 03.10.2003