SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	224395
Project Title:	Integrated Platform for Autonomic Computing
Project Acronym:	IPAC

Deliverable Number:	D1.1
Deliverable Type:	PU
Contractual Date of Delivery:	31 May 2008
Actual Date of Delivery:	9 June 2008
Title of Deliverable:	**Project Presentation**
Work-Package contributing to the Deliverable:	WP1
Nature of the Deliverable:	Report

Author(s): Vassileios Tsetsos, Sarantis Paskalis, Vassilis Papataxiarhis, Stathes Hadjiefthymiades, Eleftherios Fytros, Eleni Daoukou, Guiseppe D'Angelo, Jean-Dominique Decotignie, Christoforos Panayiotou, Panayotis Kikiras, Sigolene Pangaud

Abstract: This document presents general information and the main concepts of the IPAC project. It can serve as a public reference document for the project objectives, technical approach and work plan.

Keyword List: autonomic computing, embedded systems, middleware, system trials, contact info

(Co-ordinator)	Siemens A. E. Electrotechnical Projects and Products	SAE	Greece
Contractor	National and Kapodistrian University of Athens	NKUA	Greece
Contractor	Centre Suisse d' Electronique et de Microtechnique SA	CSEM	Switzerland
Contractor	CENTRO RICERCHE FIAT S.C.p.A.	CRF	Italy
Contractor	Hellenic Ministry of Defence	HMOD	Greece
Contractor	University of Cyprus	UCY	Cyprus

Table of Contents

Executive Summary

This document presents a general overview of the IPAC project. Some facts and figures about the project are listed, while there is also a brief introduction to the IPAC objectives, goals and technical approach. The technical descriptions that are included in this document are based on the initial approach and will be refined during the implementation of Work Packages WP2 and WP3.

A special section is devoted to the trials that are foreseen by the project consortium. These trials will demonstrate, under both real-world and laboratory conditions, the effectiveness, performance and usability of the project prototypes. The expected impact of IPAC is discussed near the end of the document, as reflected by the envisaged achievements.

Finally, contact details are provided for the Project Coordinator so that anyone further interested in the IPAC project can communicate with the implementation team.

1 Project Identification Information

1.1 Contract Number

224395

1.2 Project Name

Integrated Platform for Autonomic Computing

1.3 Project Acronym

IPAC

1.4 Work Programme Topic

ICT-2007.3.7: Networked Embedded and Control Systems

1.5 Project Logo

1.6 List of Participants

Participant Number	Participant name	Short name	Country	Contact Person
1 (Coordinator)	Siemens A. E. Electrotechnical Projects and Products	SAE	Greece	Eleni Daoukou
2	National and Kapodistrian University of Athens	NKUA	Greece	Sthathes Hadjiefthymiades
3	Centre Suisse d' Electronique et de Microtechnique SA	CSEM	Switzerland	Jean-Dominique Decotignie
4	CENTRO RICERCHE FIAT S.C.p.A.	CRF	Italy	Giuseppe D'Angelo
5	Hellenic Ministry of Defence	HMOD	Greece	Panayotis Kikiras
6	University of Cyprus	UCY	Cyprus	George Samaras

Partner	Expertise	Contributions to the project
SAE	integration procedures, wireless communication, networking, software engineering & development	Implementation of application creation component, implementation of IPAC applications, implementation of storage system for IPAC nodes
NKUA	software engineering, middleware development, research experience, autonomic networks	Implementation of application creation component, implementation of IPAC applications, IPAC middleware, knowledge engineering, simulations of information dissemination algorithms,
CSEM	real-time software, wireless communication, wireless sensor networks, sensing systems, protocols	wireless sensor networks, integration of communication components, hardware and system integration, integration of sensing elements, trials execution
CRF	manufacturing, process optimisation, micro-nano integration	requirements provision, trials execution, integration of sensing and communication elements
HMOD	crisis management, large scale deployment	requirements provision, trials execution
UCY	mobile computing, sensor computing, research experience, autonomic networks	middleware development, context awareness and personalization, sensor computing and querying, implementation of storage system for IPAC nodes

1.7 Total Cost

2,519,895.76 Euro

1.8 Commission Funding

1,650,000.00 Euro

2 Project Main Goals

IPAC aims at delivering a middleware and service creation environment for developing embedded, intelligent, collaborative, context-aware services in mobile nodes. IPAC is a very important development in the areas of embedded, mobile and pervasive computing. The main focus of the project will be in the development of an embedded middleware platform. The lightweight and flexible IPAC middleware stack will provide all services required for the deployment and execution of diverse applications in a collaborative nomadic environment. These services will be supported by novel knowledge and ontology engineering techniques, which will deal with interoperability, integration, and re-configuration/adaptation problems that are met in contemporary embedded platforms. Being collaborative, IPAC relies on short-range communications (e.g., IEEE802.15.4, DSRC, Bluetooth) for the ad hoc realization of dialogs between nodes. Being context-aware, IPAC relies on advanced sensing components thus, delivering a highly innovative application architecture.

IPAC is based on sophisticated information dissemination algorithms. Specifically, it relies on *rumour spreading* techniques. Rumour spreading involves the propagation of information within a certain network. Information is ducted only to immediate neighbours that are interested in specific content (*rumour*). Similar to such situation, is the spreading of an infectious disease in a population of humans (*epidemic spreading*). Therefore, IPAC incorporates recent research advances in the area of bio-inspired computing systems.

Nodes in IPAC, [hereinafter referred to as *communicating entities* (CE)], are logically divided into the following two categories:

1) *relaying nodes* (RN), transmit/receive and process info, but they have limited sensing and storage capabilities, and
2) *end nodes* (EN), which may receive or transmit, assess and, potentially, exploit the exchanged information.

Storage mainly refers to the caching of information transmitted by CEs. A CE always caches the received information for future relaying. When other CEs enter its proximity, it relays cached information in the form of *information spreading*.

The nodes specified and developed in IPAC have to be *non-selfish* with respect to information dissemination (a mobile incarnation of a peer-to-peer system). Specifically, the IPAC nodes operate in a *collaborative* fashion in order to diffuse contextual information and broader knowledge in their environment. A node tends to propagate an information message received by another node across the network. In case that such message appears to be usable for the node, it can process it. An information message that is of no interest to a CE has to be forwarded across the network for further processing (CE acts as RN); otherwise the CE can process it (acting as EN). The same path is followed for the dissemination of new applications or application components after their development thus contributing to the deployment and use of new embedded applications.

The platform assumes a central application creation environment. Applications may be preinstalled on the node or deployed on demand by the user. Nodes may be considered as sources of information stemming from various sensors mounted on them or from human user input. Such information may be disseminated from one node to another in the network, thus, catering for a distributed, autonomic information propagation platform. In order to have meaningful and controlled information dissemination, the spreading of information will be governed by space-time validity rules (directives). For instance, a message concerning congestion at a crossroad would be valid for less than one hour and within a radius of some kilometres.

IPAC targets real life applications in person-to-person computing, but also finds very interesting applications in the industry (e.g., intelligent transportation, industrial facility management). Target applications include traffic management, in-building guidance, industrial environment control, as well as crisis management. For instance, automobiles equipped with an IPAC device may forward information to one another regarding the road conditions (e.g., congestion, accidents, etc.) so as to improve circulation, and avoid potential accidents. Another potential application domain of IPAC is road advertising, or advertising in large commercial centres.

Some of the objectives targeted by the project are the following:

- Integration of multiple sensors into IPAC devices and their deployment during field trials
- Development and execution of different services on the IPAC infrastructure

- Trials deployment and execution
- Investigation of different SRC (Short Range Communication) technologies
- Preparation of hardware/software IPAC nodes prototypes
- Publication of scientific articles in international conferences and journals
- Attainment of B.Sc. and M.Sc. theses through project activities
- Creation of synergies (scientific and technical) with other projects
- Minimization of the time to market for IPAC nodes and services
- Strengthening of ICT research
- Contribution to relevant standardization efforts

3 Key Issues

IPAC pursues the development of a middleware platform for embedded devices with specific characteristics. Specifically, IPAC will integrate (bundle) techniques and algorithms for energy-efficient, autonomic node behaviour, advanced context awareness, embedded service/application modelling and efficient information dissemination. The IPAC contribution to these areas is studied in the following paragraphs.

3.1 Autonomic Computing in mobile ad-hoc environments

Autonomic computing is a new communication paradigm to assist the evolution of communication networks towards functional adaptability, extensibility and resilience to a wide range of possible faults. An autonomic system has four major characteristics: self-configuration, self-healing, self-optimizing and self-protecting (usually referred to as self-CHOP characteristics). Additionally, an autonomous system has several more characteristics (such as openness, self-awareness, self-management, self-adaptation, self-description), which enrich the total system's self-adequacy. These attributes of an autonomic architecture ideally match with the nature of mobile ad-hoc environments, where the state of the network, the number and type of users and the available resources vary continuously, in an unpredicted way and the need for network autonomy to maintain stability and efficiency is crucial and stronger than in any other case. Autonomic characteristics are directly associated with the quality factors typically used to evaluate the quality of a prototype, namely reliability, efficiency, maintainability, usability, functionality, portability, etc.

3.2 Reliable and efficient algorithms for information dissemination in autonomic ad-hoc environments

There is a multiplicity of approaches for the propagation of information in a network like *flooding*, *selective flooding*, *epidemical dissemination*, and *spatial gossiping*. However, flooding-based methods are not efficient approaches, as the same information may be circulated into the network, resulting in redundant transmissions and increased energy consumption (thus reducing the lifetime of nodes).

3.3 Embedded service/application modeling and provision

Modelling service logic can be easily achieved using high level programming languages like Java, C++ or other object oriented or procedural languages. Nevertheless, without overseeing the power of such general-purpose languages, none of them is tailored to the specific requirements of a particular application domain. Aiming to overcome this inefficiency, many organizations, vendors and consortiums have focused, during the recent years, on the specification of domain-specific service languages. Such languages are simple and easy to be used, thus enabling accelerated application development. Their simplicity lies in the fact that they define a minimum set of rules and directives and they are, in most cases, text-based and, consequently, easy to understand. The advent of the XML specification contributed a lot to the flourish of such languages as it enabled their systematic design and processing using off-the-shelf editors and tools.
Another great feature of the XML representation is that it can easily be presented in a visual manner, instead of the traditional textual form. This allows the easy development of visual service building tools. Many of the IPAC partners have significant experience in the development of such visual tools through their participation in relevant projects.

3.4 Collaborative Context Awareness

Context-awareness is a very important aspect of the emerging pervasive and autonomic computing paradigm. In order to render applications intelligent enough to support contemporary users everywhere/anytime and materialize the so-called ambient intelligence, information on the present context of the user has to be captured and processed appropriately. The efficient management of contextual information requires detailed and thorough modelling along with specific processing and inference capabilities. Mobile nodes that know more about the user context are able to function efficiently and transparently adapt to the current user situation. Several context models have been proposed in the relevant literature. Lately, considerable discussion takes place with regard to the issue of collaborative context awareness. By collecting contextual parameters from nearby collaborating nodes one can summarize a description of the current context that may be more accurate than the context recognized by individual nodes thus, realizing the idea of collaborative context awareness.

3.5 Progress beyond state of the art

IPAC will develop embedded middleware technology for the realization of innovative context-aware services by autonomous nodes. Supported by ad-hoc network infrastructures, IPAC will proceed to the study of situation- and context-aware services deployed by numerous and mostly dynamic network groups and communities. Context-awareness will allow autonomic nodes to sense and adapt to their environment, not only at the network level but also at the application plane. This adaptive behaviour will be aided by knowledge-based methods and technologies, in ways not studied by current or past research projects. Therefore, the key contribution of IPAC in this area is a novel embedded middleware and service provision platform that brings considerable intelligence to the device. None of the existing research projects or products, in academic or industrial environment, has achieved to implement all the self-CHOP characteristics, and additionally offer self-awareness and context-awareness, which are of major importance in many environments. IPAC will address all the self-CHOP requirements and thus provide a solid, efficient and future-proof platform.

IPAC targets mobile, embedded devices with energy consciousness. Therefore, sophisticated information propagation algorithms are needed that take into account the nature of the information considered and limit redundant transmissions. IPAC will adopt and optimize epidemic dissemination algorithms by taking into account several context-aware application requirements. The tuning of the parameters of the epidemical spreading algorithms will be done after careful study and simulations.

Within IPAC, the focus will be on the efficient delivery of embedded applications in a variable environment integrating heterogeneous technologies. IPAC will create a generic modelling language capable of representing a great variety of embedded applications. The project will go even further, delivering a visual editing environment that will support application creation through the automatic generation of the application logic specification from its corresponding visual representation. IPAC will facilitate the very rapid introduction/deployment of services/applications in the embedded systems domain. Rapid application development is a new concept in the area of embedded systems, and will revolutionize their future applications. The project's contribution lies both in the application development process but also in the deployment and spreading phase.

It is worth mentioning that IPAC will rely on ontological frameworks for controlling node and application behaviour. This capability which will be embedded in the IPAC middleware and the application creation environment is considered highly innovative and has not been seen in the embedded systems domain before.

Incorporating the collaborative behaviour of mobile nodes in context management, as addressed by IPAC, is a novel contribution to the pervasive computing research. In the IPAC project, an advanced collaborative context awareness model will be engineered and adopted. Such model will represent and manage context according to the applied domain (e.g., probabilistic and fuzzy sets-based context representation for sensors measurements and conceptual modelling through ontologies for user profiles and device capabilities).

Lastly, IPAC will research on the capabilities of different short range communication schemes, pursue their joint, seamless operation in a single computing unit and tune their parameters for optimized operation in the context of the IPAC environment.

Overall, IPAC will try to merge all the discussed techniques and algorithms in a single platform. Benefits will stem from the thorough study and optimization of the above areas

4 Technical Approach

4.1 Overview of IPAC Architecture and Description of Main Components

4.1.1 The IPAC Node Architecture

In IPAC the same middleware infrastructure can be used in many diverse ways and usage scenarios. As already mentioned, there are two types of nodes: end nodes (EN) and relaying nodes (RN). In the following paragraphs, the preliminary design of the generic IPAC node is described along with some indicative enabling technologies that will be examined, and possibly used. Finally, the flexibility and universality of the solution is shown through some example uses cases. An IPAC node has four main components/subsystems (see also Figure 1): the IPAC Middleware, the IPAC Embedded System (ES), the Short Range Communications Component (SRCC) and the Sensing Elements Component (SEC).

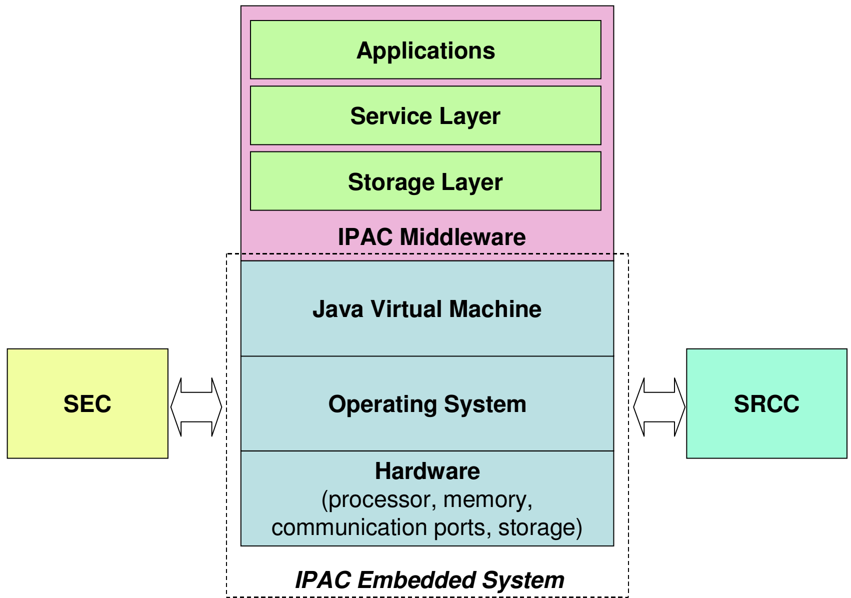


Figure 1. IPAC Node Architecture

IPAC Middleware

This middleware will be the core of the IPAC platform. It can be further decomposed to the service layer, the storage layer and the application layer. The storage component (flash memory, etc.) may store (cache) information to be forwarded when other nodes are in vicinity. This component is described in detail in the following section. Regarding the enabling technologies, IPAC middleware will be developed over a Java-based, lightweight container for dynamic software components. This framework will be responsible of handling the interactions between components, dynamic deployment and updating. A very reliable framework is OSGi (Open Source Gateway initiative). This framework fits the needs of pervasive computing environments by providing the aforementioned functionality. The adoption of OSGi for diverse embedded devices is rather limited for the time being (there are only some commercial OSGi distributions available for embedded devices). Hence, the IPAC middleware will adopt one of the following implementation approaches: a) implement a custom OSGi-like architecture or b) specialize and adapt an existing open OSGi implementation according to the requirements of IPAC.

Some details on the various middleware layers follow:

Applications The *application* layer is the runtime environment for applications' execution. IPAC applications may be preinstalled on the device (node) or the user may download them individually. Applications are designed and created in the Application Creation Component (ACC). Application types include road congestion condition notification, product advertising, infrastructure-less crisis management, etc.

Service layer. The service layer acts as a mediator of the core components which are integrated in an autonomic node. It consists of *modules* that manipulate and coordinate the data streams (sensor readings, messages, etc.) between the applications, the storage layer, the sensing elements and the SRC component. Some of these modules have a direct interaction with *sensors*. The main role of this kind of module-collector is to interact with the appropriate registered sensors and retrieve (collect) the available measurements or events (temperature, position, etc). The retrieved information is then processed and maybe forwarded to the upper layer where various applications can exploit it. Another type of module in the service layer interacts with the *SRC component*. The key role of this module is the interception and the assessment of messages that other nodes transmit. If such a message refers to an application installed in the upper layer, it is fed to the corresponding application. All running applications are registered according to their functionality (input, output, etc) in an *application registry*. In case there is no application that can use the message, such message is either forwarded to other nodes or stored to the cache for future forwarding, depending on the

rules specified by the owner of each device. An additional functionality of this module is the pushing of measurements or messages to the SRC component when the node moves near to other nodes. It is important to mention here that the messages arriving from the SRC component or the messages that are ready to be transmitted are “queue handled” in order to optimize the information management.
The management of the *cache memory (storage)* is vital for the overall functionality of the IPAC middleware. As mentioned before, a node may store locally produced or relayed measurements from sensors and messages arriving from the SRC component. All this information needs to be evaluated based on its creation time and place. Hence, the cache management module of the *service* layer needs to consider such spatio-temporal constraints whenever it caches or forwards information.
The sophisticated *information dissemination algorithms*, e.g., selective flooding, epidemic- and rumour-spreading, that are implemented in the service layer need to take care of redundant transmissions, as well as take into account the context of the communication environment. Various context parameters will be considered in order to re-configure the behaviour of a node. For example, the density of nodes may lead to different kind of topologies (e.g., connected-graph topology in case of dense node distribution or segmented topology otherwise), which should be handled by adjustment of certain configuration parameters (e.g., selection of a different SRC interface, increase in maximum number of re-submissions). In order to, facilitate context-modelling and re-configuration, a lightweight ontological infrastructure will be exploited (described in a following paragraph).
Storage layer. The storage layer consists of two segments: the *private* segment and the *public* segment. This discrimination serves the purpose of efficient information handling. The *private* segment is used by applications and/or other modules that run on the node and need data storage functions. Contrary to the private segment, the *public* segment is used for storing and relaying measurements and messages that are usable by installed IPAC applications but which may also be forwarded to other nodes in proximity. It should be mentioned that, the collaborative nature of a node is based on the efficient handling of the *public* segment of the storage component.

IPAC Embedded System

This is the device (along with the operating system and basic software) where the middleware is deployed and the sensor and communication boards are connected. Depending of the functional type of the node, the device may be *a custom embedded system* or a *Commercial Off-The-Shelf* (*COTS) mobile handset*. The latter may be a mobile/smart phone, a Personal Digital Assistant (PDA) that is very popular in modern car navigation systems, or any other similar end-user device. The main parts of such device are the processing unit, the storage unit, the operating system, a JVM and the communication ports. We should note before proceeding that the adoption of an open programming language as Java is critical for achieving IPAC’s independence from the underlying operating system and hardware. Moreover, this system will in most cases provide also a Wide Area Network (WAN) interface (e.g., Ethernet, GPRS). In such occasion, this interface can be used to connect the node to the Internet or other WAN infrastructure (this is depicted in Figure 3).
Regarding the ES, IPAC will explore several possible solutions:

1. *Handset-based IPAC ES (*PDA or mobile/smart phone*)*
2. *Custom embedded system:* In case the IPAC node is not based on an existing user device (i.e., it is a RN), we can adopt the following two approaches for the IPAC ES. Note that in case the device is a custom embedded system, it will have no user interface but can handle user input and output (if necessary) through a Bluetooth-enabled handset carried by the user.
    a. *Integrated embedded system with external OS and JVM:* The first follows the typical model of embedded systems, consisting of microprocessor, board and network interfaces. A representative example of this category is the Soekris board.
    b. *Java-based embedded system (Java support at chip-level):* The second approach is to use an embedded device with Java chip, capable of executing Java bytecode directly. There is no software operating system, but a micro-programmed real time kernel.

Many vendors can provide the above two solutions. The final design and implementation decisions will be made during the initial phases of the project.

Short Range Communications Component

One of the basic components in the IPAC architecture is the short-range communication (SRC) component. This component is responsible for the interchange of messages from one mobile node to another through the wireless medium. Contrary to long-range communications, SRC is well suited in the context of IPAC. IPAC applications will typically have a quite wide geographical scope, in which case infrastructure-based, long-range communication would require mobile nodes to transmit with rather high power, resulting in reduced lifetime for battery-supplied nodes. Through SRC, nodes need only reach their neighbour, which will in turn communicate with one of its neighbours, in the way to the destination node, and so on. The distance between neighbouring nodes may be typically in the order of metres, thus, rendering such multi-hop communication energy-efficient. Moreover, as the transmission power of the mobile nodes needs not be high, the mutual interference between (non-neighbouring) nodes can be kept to quite low levels. Lastly, in SRC due to the proximity (and possibly line-of-sight propagation) the achieved data rates may be quite high. Several interfaces will be explored: Bluetooth, WSN (Wireless Sensor Networks), and DSRC. DSRC offers faster access times and enhanced data rates, enabling the efficient information exchange between (moving) nodes. On the other hand, technologies, such as WSNs, exhibit the low battery consumption required for some application and will be enhanced towards higher responsiveness. Bluetooth is a ubiquitous communication technology and will “bridge” IPAC to existing user devices. Moreover, the use of the WiseNode RF module will be also investigated. This component will communicate with the middleware, through a communication port. It will also include all the electronic circuits necessary for the integration of the wireless interfaces. One, two or all interfaces may be active concurrently, depending on the application domain.

Sensor Elements Component
SEC is composed of sensors installed on the nodes measuring various environmental parameters like temperature, humidity, or detecting events like road congestion, icy road, etc. Such sensor readings are fed into the IPAC middleware for further processing, and are then exploited by local applications or forwarded to neighbour nodes. SEC is actually a board hosting various types of sensing elements, as well as a wireless interface in order to send/receive data in the context of a wireless sensor network. This node can be based on existing architectures (e.g., Berkeley motes) or not. The project will not put a significant effort in developing such boards, but will rather focus on integration and interoperability issues at the middleware layer. Some equipment will be provided by CSEM, which has already developed such WSN platform in the context of the WiseNet project (Figure 2).

Figure 2. WiseNet platform nodes

Flexibility in IPAC node configuration
The aforementioned components of an IPAC Node can be combined with many ways in different use cases. This is demonstrated in the following figures.

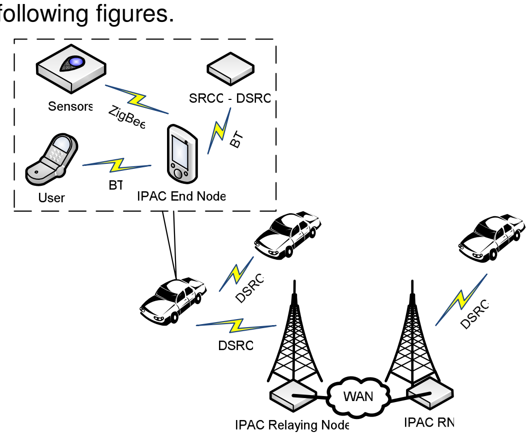


Figure 3. Automotive application

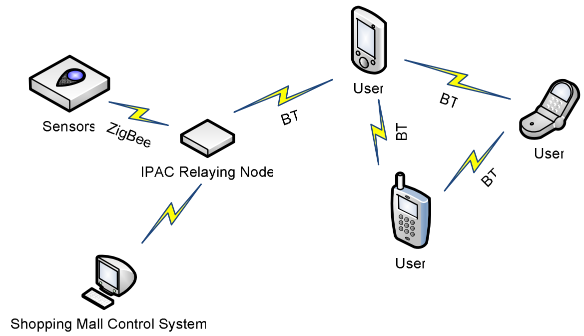


Figure 4. Pedestrian application

Lightweight knowledge-based framework for re-configurability and interoperability
Both context-aware re-configuration and interoperability between nodes with different features call for an intelligent system behavior, depending on the characteristics of each individual case (contextual information, node features, number of adjacent nodes, etc.). To meet these requirements, an architecture shift is necessary in the design of embedded systems middleware. Specifically, a new approach similar to the Knowledge Plane is called for. Such approach includes all the necessary components in order to create a distributed cognitive system, which is aware of its goals, limitations, and resources. The IPAC Knowledge Plane operates also as a broker since it is able to disseminate to all middleware layers the status of the SRCC, SEC and other key modules of the service layer. For example, it may give feedback to the dissemination algorithms about the physical-layer operation of peer nodes.
Specifically, some of the functions of this knowledge-based framework follow:

- model the possible situations (i.e., context) of the node/system,
- store the situation-information collected from the sources,
- reason over contextual data,
- identify possible conflicts in the system (e.g., through consistency checks),
- infer new information based on real-time observations,
- disseminate the inferred information to interested recipients.

Ontologies, which have gained much popularity due to the Semantic Web initiatives, are a good candidate technology for implementing such framework. The use of ontologies accommodates a model-driven design, which, in our view, can facilitate the node adaptation processes, through model consistency checking and reasoning.
The formalism used to represent the required ontologies has to be restricted in terms of expressiveness (i.e., be a small subset of first-order or predicate logic) in order to achieve the desired performance of reasoning process. This way, the model provides simplicity along with ease of management, while it supports basic reasoning services. Moreover, an efficient reasoning engine will be embedded in the middleware (e.g., based on a stripped-down version of a tableaux algorithm). Finally, special techniques will be investigated in order to achieve efficiency during ontology representation and serialization.

4.1.2 Application Creation Component

Within the scope of the IPAC project, the application creation environment refers to all tools, APIs and libraries, which assist the development of new applications for the IPAC node. Such applications, upon their creation, are uploaded to the IPAC node, and reside at the upper layer of the IPAC middleware stack (i.e., application layer), which serves as the runtime environment for their execution.
The IPAC application receives multiple inputs from different information sources. Input might come directly from the user (through the appropriate user interface), in the form of a short text or a control signal (e.g., press of a key). Moreover, it can receive input from a variety of sensors, integrated to the IPAC node and/or the SRC interfaces that may be supported (WSN, etc.). Next, the input data is evaluated through certain rules in order to provide a single output that feeds the appropriate communication elements, which will disseminate it further. In this perspective an IPAC application behaves similar to a MISO (Multiple Input Single Output) system, an assumption which can significantly simplify its modelling by enabling application development in a visual manner, as it is done already, by several systems design tools (e.g., matlab's simulink toolbox).

GUI tools, which support such type of application development, can be easily built using open source GUI frameworks like the Eclipse platform. Using these tools the developer has to define the expected inputs for his applications by selecting from a container the available/supported sensors and dragging them on his workbench. This sensor container will come in the form of a toolbox integrated with the visual editing environment. It will be extensible and capable of accommodating specifications for new sensors through an import option. Each sensor is represented as an icon, in the visual editor, with several outputs (one or more), backed by an XML specification, which describes these outputs. Other toolboxes will, also, be available enabling the developer to select other information sources and connect them to his system/application. The SRC elements (DSRC, WSN, Bluetooth) comprise iconized components, which will be, also, available for being placed on the application workbench. Use of such components as end points for the information flow will be mandatory if an outbound flow needs to be defined. Moreover, several components corresponding to flow control elements, pre-defined functions and middleware services will be available in separate toolboxes, enabling the application developer to choose from and use them inside his application.

Developing new IPAC applications will be easy and require the following simple steps (a mock-up screenshot can be seen in Figure 5):

1. Dragging input sources components such as sensor components, SRC components or other on the workbench and/or define their properties.
2. Dragging flow control, function and service components on the workbench.
3. Defining the information flow between the input components and the control/function/ service components by interconnecting the various icons on the workbench.
4. Assign the desired outbound information flow to the designated communication component(s).

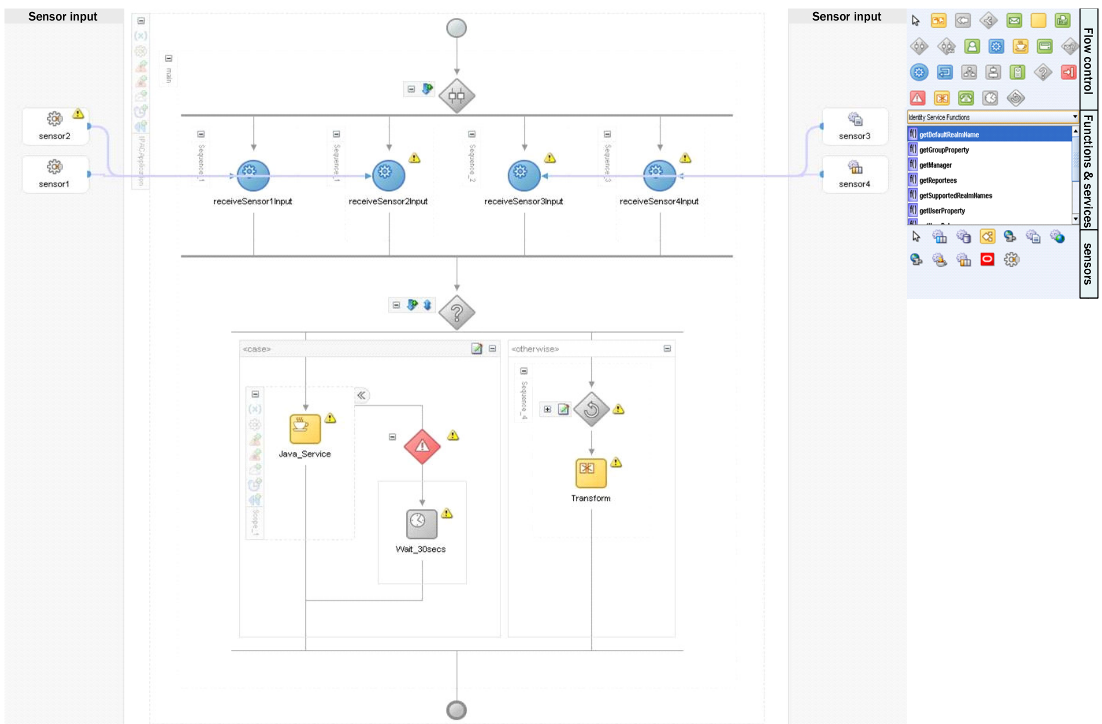


Figure 5. Sample visual application editing process

The diagram produced by the application developer in the editing window of the application creation environment, comprises the application specification, which will be backed up by an XML document, generated automatically. This XML specification constitutes the actual deployable application component, which will run inside the IPAC middleware. The application specification can either be compiled to produce an executable form of the application that will be deployed in the IPAC node or run in interpreted mode, in which case the XML itself can be used for the deployment.

In any case, the application creation process is an offline activity, meaning that no live connection to the IPAC node is required. In this perspective, the application creator can develop his applications, produce the corresponding deployable elements (e.g., an application archive) and publish it through any means (e.g.,

through a web portal). The IPAC node owner can download or buy such an application and use the interface provided by his node to install and run it. Runtime dependencies (e.g., availability of sensors) will be resolved during its installation and possibly prevent its completion.
The application creation environment will further support the IPAC application creation process by providing tools enabling developers to deploy and test their applications without the need to actually install it on the IPAC device. To achieve this functionality, the creation environment will integrate an emulated IPAC middleware stack and appropriate tools that will allow the execution of an application in emulated mode. The tools will be used for defining basic operations that are essentials for running the application in emulated mode, such as:

- Test data, flowing from the input components (sensors, user input, SRC elements) that have been defined in the IPAC application
- Runtime configuration parameters, which are otherwise provided from the IPAC device (e.g., by some policy*)*.

Moreover, the visual editor may provide feedback on the emulated execution of the application, by depicting in real time the flow of information between the various processing elements composing the application.
Each application will comprise three main parts: the *source application logic*, the *target application logic* and the *relay policy*. The first refers to the actual functionality of the application in End Nodes that act as information sources (e.g., when and what data will be disseminated). The second refers to the actions that a node will take upon reception of application-relevant information (e.g., how the received data will be used or presented to the user). The last one refers to the configuration parameters that should be set to the relay nodes so that they exhibit an "application-friendly" behaviour (e.g., in terms of QoS). Such parameters may be spatio-temporal validity values, priorities assigned to the application messages etc.

4.1.3 Simulations

Several simulations will be performed in the context of the IPAC project. Such simulations will focus on the comparison of the performance of the developed information dissemination algorithms versus certain parameter configurations. These parameters are related to the following characteristics of the studied environment:

1. the mobility behaviour of nodes: nodes may exhibit different degrees of mobility, from low mobility in case of pedestrians, to high mobility in case of vehicles,
2. the nodes connectivity: depending on the distribution of the nodes in the studied environment, as well as their radio capabilities, different network topologies may derive, e.g., scale-free power law network topologies and random homogeneous networks,
3. the nature of the disseminated information: the information disseminated may vary from highly dynamic changing contextual information, in the case of sensor readings, to fairly static context, in case of user profile information and preferences,
4. the spatiotemporal validity of the disseminated information: different rules may be defined for restricting the validity of information in spatiotemporal terms, thus, limiting the spread of information in the time and space horizon., and
5. the certain thresholds applied in the information spreading: also known as *epidemic thresholds*, such thresholds determine whether a piece of information will continue to be spread across the network or not.

The simulations in IPAC will also examine the *efficiency* and the *reliability* of the discussed dissemination algorithms over distributed sensors and information sources. The *efficiency* of the information dissemination algorithms will be assessed with regards to the energy consumption of sensing and relaying nodes of a specific network. *Reliability* of a dissemination algorithm relates to the proportion of the nodes that have been interested in the disseminated information (*rumour*) and have finally received relevant information. Hence, simulations results will highlight the reliability of the algorithms, as far as the mobility issues and spatiotemporal constraints are concerned. The performance of such algorithms will be examined for the case of (i) 2D-latice networks, (ii) random graph networks, (iii) Barabasi-Albert (BA) power-law networks (using the NS-2 or Omnet++ simulator), and (iv) top-down hierarchical networks topologies. As each network topology relates to different applications of the discussed algorithms, the simulation results will cover a wide range of possible applications.

4.2 Work Packages

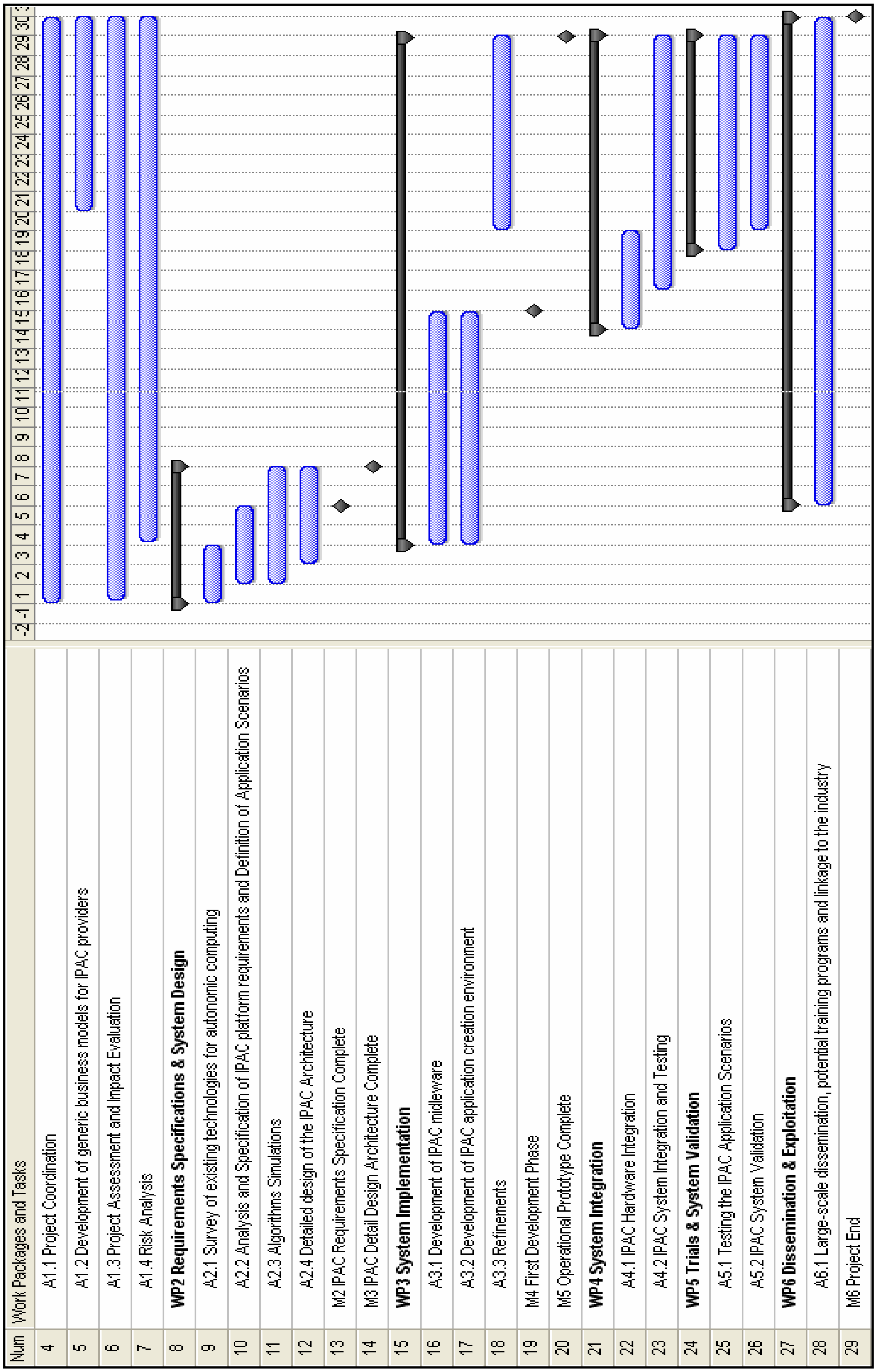
WP no.	Work Package Name	WP Description	Main Activities	WP Leader
1	**Project Management**	WP1 provides the management and overall coordination of activities, assessment, project impact evaluation and risk analysis. It ensures that objectives are met and represents the Project to the European Commission and the ICT community in general.	**A1.1** Project Coordination **A1.2** Development of Generic Business Models for IPAC Providers **A1.3** Project Assessment and Impact Evaluation **A1.4** Risk Analysis	SAE
2	**Requirements, Specifications & System Design**	WP2 will define the requirements and specifications of the targeted system. This includes both the application creation environment and the IPAC platform. The system design, which will be produced, will comprise the basis for the implementation phase that follows in WP3.	**A2.1** Survey of Existing Technologies for Autonomic Computing **A2.2** Analysis and Specification of IPAC Platform Requirements and Definition of Application Scenarios **A2.3** Algorithms Simulations **A2.4** Detailed Design of the IPAC Architecture	NKUA
3	**System Implementation**	WP3 constitutes the main work of the IPAC project. The work includes the development of the main components, IPAC middleware and Application creation environment, being used for the specification, creation and execution of IPAC applications. The work in this WP will be based in the results produced by activities A2.3 and A2.4 of WP2.	**A3.1** Development of IPAC Middleware **A3.2** Development of Application Creation Component **A3.3** Refinements	SAE
4	**System Integration**	WP4 concentrates on the integration of the elements produced in the WP2 and WP3. All subsystems previously developed or adopted will be glued together, in this WP, with the aim of producing an operational prototype of the IPAC platform.	**A4.1** IPAC Hardware Integration **A4.2** IPAC Overall Integration and Testing	CSEM
5	**Trials – System Validation**	Extensive testing will be carried out in order to validate the behaviour of the IPAC system and certify that the prototype meets the defined specifications.	**A5.1** Testing the IPAC Application Scenarios **A5.2** Platform Validation	CRF
6	**Dissemination & Exploitation**	WP6 package will perform dissemination and exploitation. IPAC will lead to the development of a platform which delivers application creation and runtime environment for autonomic computing. This concept should be disseminated following a dissemination plan.	**A6.1** Large-Scale Dissemination, Potential Training Programs and Linkage to the Industry	NKUA

4.3 Deliverables

Del. no.	Deliverable name	Deliverable Description
D1.1	**Project Presentation**	Description of the key project concepts and objectives. Summarization of the main components of the IPAC platform. Presentation of the consortium. Potential S&T approaches that will be investigated in the course of the project. Expected project results and impact.
D1.2	**Generic business models for IPAC providers**	Description of the IPAC enabled value chain (business roles). Description of several business models (e.g., revenue flows). Comparison of IPAC business models to existing business models in relevant domains.
D1.3a	**Risk Analysis**	Preliminary identification of risk factors and their dependencies. Outlines of contingency plans and risk prevention strategies. Initial version of the Risk Matrix.
D1.3b	**Risk Analysis**	Final version of the Risk Matrix with descriptions of all problems encountered in the project and evaluation of the corrective actions taken.
D1.4	**Periodic Progress Reports**	Collective project report from all the partners in the projects. They include work performed, milestones achieved, personnel labour, problems faced and actions taken.
D1.5	**Final Project Report**	The final administrative project report summarizing the individual achievements, breakthroughs, problems and lessons learned.
D2.1	**State of the Art**	Extensive survey of state-of-the-art technologies for the development of the IPAC platform. Survey of related European and International projects. Survey of state-of-the-art methodologies, algorithms and design approaches. Comparison of competitive approaches/ technologies.
D2.2	**Requirements Definition and System Specification**	Definition of functional and non-functional (e.g., QoS) requirements as resulted from the system description and D2.1. Description of application scenarios that should be supported. Specification of the projects trials (metrics, functionality to be tested).
D2.3	**Dissemination Information Algorithms Simulations**	Description and execution of algorithms for information dissemination. Setup and execution of the respective simulations. Estimation of optimization parameters for the application scenarios defined in D2.2.
D2.4	**IPAC Architecture defined and specified**	Detailed description of all aspects of the IPAC architecture. UML diagrams of various degrees of detail will be used. Detailed specification of all Application Programming Interfaces. Specification of all data and knowledge models involved.
D3.1a	**Development of the IPAC Middleware**	Implementation of the Application, Service and Storage layers. Implementation of the interfaces with the sensing and communication elements. Testing of all implemented functionality with sample application scenarios.
D3.1b	**Refinement of the IPAC Middleware**	Refinement of the developed middleware during and after the first trial
D3.2a	**Application Creation Environment Implementation**	Implementation of the Visual Editing Environment. Implementation of emulation software for testing off-line IPAC applications. Testing of all implemented functionality.
D3.2b	**Application Creation Environment Refinement**	Refinement of the Application Creation Environment after the first trial.
D4.1a	**IPAC Operational platform, Integration Report, industrial environment**	Hardware integration of the sensing components. Hardware integration of the communication components. Software integration of all IPAC software parts. Description of all integration tasks and detailed platform setup instructions. Extensive testing of the platform. Report of problems encountered during integration/testing along with the corresponding solutions.
D4.1b	**IPAC Operational platform, Integration**	Hardware integration of the sensing components. Hardware integration of the communication components. Software integration of all IPAC

	Report, emergency environment	software parts. Description of all integration tasks and detailed platform setup instructions. Extensive testing of the platform. Report of problems encountered during integration/testing along with the corresponding solutions.
D5.1a	**IPAC Trial Report in Industrial Environment**	Deployment and configuration description for the planned trial. Description of last-time optimizations/adjustments.
D5.1b	**IPAC Trial Report in Emergency Environment**	Deployment and configuration description for the planned trial. Description of last-time optimizations/adjustments
D5.2	**IPAC Platform Validation according to all Trials**	Description of the user feedback (information about general user acceptance and usability of the provided services). Detailed report on whether the project met its initial objectives, based on the trials results.
D6.1	**Dissemination and Exploitation Plan**	Presentation of dissemination and publicity activities (Web site, scientific publications, events/demos). Description of patenting activities. Description of partners' exploitation plans. Description of potential training activities.

4.4 Project Implementation Schedule



5 Trials

The IPAC consortium will pursue three different prototype implementations/installations and trials to validate the versatility and flexibility of the delivered system. We have chosen three different application domains for the trials to show that a very wide range of applications can be engineered through IPAC.

The first trial scenario involves the installation of IPAC in a FIAT industrial production line. The target audience of this trial is the production engineers of FIAT. IPAC nodes are mounted on the manufactured items, the machinery in each production stage but also carried by engineers and repairmen. Each component or subgroup in the production line and roaming technical managers carry an IPAC node. The personnel and the automobiles in the production line are considered as *mobile IPAC nodes*, whereas there will also be some *stationary (fixed) IPAC* nodes.

The fixed IPAC nodes and the personnel's IPAC nodes are sources and/or sinks of information and, therefore, play the role of EN nodes in the terminology of IPAC. Production items carrying IPAC nodes are simply RN, which relay information among production items and employees. The hop-by-hop information propagates either backwards in the form of feedback messages or in the forward direction in the form of alert-warning messages. Process quality issues and self diagnosis test results, originating from the fixed nodes, are diffused in the personnel ad-hoc network targeting the responsible persons.

Hence, the IPAC infrastructure will provide all the support for the communication of all nodes (machines, workers, technicians) that participate in the production process. The adoption of the IPAC platform in the FIAT industrial environment is primarily intended to simplify and expedite different segments of the production chain. It can be used either as a monitoring tool for the infrastructure (e.g., for the detection of alarming conditions in a process) or for monitoring the production per se (i.e., capture/relay quality measures of the production equipment or the produced items).

The second trial scenario involves the use of IPAC in a European training center for crisis management conditions. In such a situation IPAC can close the gap in communications by providing meaningful information fine grained to the needs and capabilities of the agencies rushing at the area of disaster.

Vehicles and personnel deployed in this training facility will carry IPAC with the purpose of optimally coordinating the overall activity. HMOD will perform trials using the IPAC prototype results in its Multinational Peace Support Operating Training Centre (MPSOTC) in Kristoni, Kilkis, near Thessaloniki http://www.mpsotc.gr/. The trials will focus on the deployment of IPAC infrastructure in simulated humanitarian operations in order to provide a much needed communication infrastructure.

The trial environment will simulate various scenarios common in a humanitarian crisis zone (such us relief force establishment, reception of refugees or NGOS's at disaster zone) where any other communication is non-existent (Figure 6). In this case, the IPAC infrastructure will assume the role of communication support between pedestrians, and vehicles, or between static check points and vehicles or pedestrians, of the operations workforce. Since some of the information transmitted will either be critical or confidential, IPAC security support will be tried out. Furthermore, the possibility of sensor integration will be investigated, in order to provide warning and alerts about possible dangers in the vicinity of the IPAC devices. Sensing will mainly provide the following information: GPS coordinates, wind speed and direction, fire/smoke detection, vehicle status, chemical contamination, vibrations, presence (e.g., RFID).

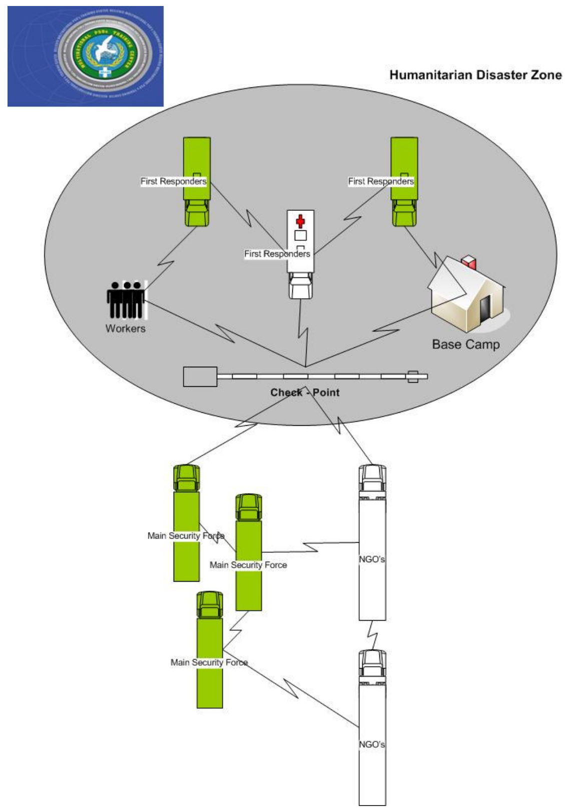


Figure 6. IPAC in humanitarian crisis zones

Thus, the target audience of this trial includes the personnel of the armed forced, security forces and public protection authorities. The adoption of the IPAC platform in the Kilkis training center for peace keeping missions is intended to support the relocation of civilians from disaster sites, the on-site rescue of civilians and the protection or critical assets.

Finally, a small-scale prototype development and laboratory trial is also planned. This trial will focus on ITS applications. Vision sensors (possibly supported by other sensors and sensor fusion mechanisms) will be mounted on cars. Such devices will be part of the IPAC architecture which will allow the exchange of road-alerts (e.g., warning on snow accumulating on the road floor) between the participating vehicles.
The optical sensors will be low-power, low-cost contrast cameras, featuring very high-dynamic range, but low-resolution capabilities. This approach allows pervasive use of optical sensors, not usually found in standard cameras, but also not forbidden by privacy laws. Indicative information provided by the sensors will be the following:

- Detection of markings (lanes and lanes curvature)
- Lane departure warning
- Presence of vehicles in front (one vision sensor) and behind the car (second vision sensor)
- Distance evaluation to precedent and next vehicles
- Potentially also, detection of their beams status
- Detection of external road conditions (fog and snow, tentatively ice or water if feasible) when possible under certain circumstances
- Detection of exteriors illumination levels and road curvature is possible for beams control

The size of the project does not allow for specific silicon release, but prototype solutions exist, and CSEM will provide with two RF enabled vision sensing nodes that can be operated on battery for short periods, when experimental operation is required. The size of the project does not allow also for other firmware development, but the capability of the vision sensor could permit to analyse scenes in various IPAC applications (either fixed – complementary information to mobile nodes network or either in mobile configuration); for example in industrial environments, detection of machines / human risks (hands, laser, or human detection in restricted areas or sub-zones..etc), detection of people presence for building management, shape recognition and verification, tracking of objects, smoke and fire detection...

6 Expected Achievements and Impact

6.1 Expected Achievements

IPAC seeks to produce an ICT system through close collaboration of EU research institutions, private companies and public authorities involved in the IT, automotive and manufacturing sectors. In this section we try to identify the expected achievements of the ICT research that is planned in the context of IPAC.

Industry
IPAC will contribute innovative solutions with significant market value to the networked embedded systems domain. The key impact is twofold
(a) on the networked embedded systems sector through the existence of a common reference platform for multipurpose devices. Manufacturers will gain the competitive advantage of offering a compatible multi-purpose embedded device
(b) on the services/applications sector through the existence of compatible devices, capable of executing the same application on a different machine. The existence of a reference network communication platform ensures the minimisation of the total cost of development and time to market of a new application to a fraction of their current values.

Universities/Research Centres
The IPAC project will bring together experts and researchers from different domains (geo-sciences, computer science, electrical engineering, etc.) in an effort to deliver an integrated platform for environmental risk management. The Universities and Research Centres will significantly benefit from this collaboration context, as they will expand their activities spectrum beyond their current research fields and improve their adopted processes and tools (e.g., by means of advanced computing infrastructure).

Employment
IPAC will improve the current status of EU industry in the product and services domain. New, advanced networked embedded devices will surface the high-technology market and new services will be delivered by a variety of companies and/or public bodies. This mobility is expected to improve EU-wide employment levels by developing new working opportunities with a long time perspective.

Citizens and community
IPAC fosters the development of cooperative networks and services, beneficial to the build-up of the community spirit between fellow citizens.

On the technical domain, IPAC will significantly boost the adoption of Short Range Communication technologies (SRC). The SRC standards are currently being processed in standards fora and industrial consortia. IPAC will act as a catalyst for their quick introduction and market establishment.

6.2 Impact

The impact on the business sector of embedded systems is expected to be profound. By providing the necessary infrastructure for networking between different systems and architectures, a network of these devices can be built. The provision of a generic programming interface will also enable the proliferation of multiple, diverse applications, which will have a much lower entry barrier (due to the existing communication infrastructure). The importance of IPAC to the growth of applications for embedded devices resembles the importance of the IP protocol for the Internet applications of today. By providing a universal networking architecture, tailored to the needs of embedded systems, new service paradigms open up, and existing ones are transformed into much larger frameworks. Some of the major impacts of IPAC are:

- It is projected that the common deployment platform built by IPAC will reduce the overall cost of development as well as the time to market for any new application to half.

- The paradigm of one application per embedded device will be transformed into a generic embedded device capable of performing multiple jobs and/or reconfiguring itself through the network with new capabilities and implementing new services.
- A new business sector for applications for embedded systems will be created as a spin-off from the existing embedded systems.
- The expected impact of IPAC is also spread beyond the strict limits of the embedded devices sector. The applications developed will have a profound impact on multiple human activities. Some example applications envisioned that make use of the common network infrastructure for embedded systems can be:
    - road safety (through embedded sensor networks both in automobiles and public roads)
    - manufacturing (both through quality control checks and improvised feedback to the actual manufacturing robots)
    - energy conservation (through power-efficiency mechanisms for energy consumption
    - large scale monitoring of buildings, landfields, roads, bridges, automobile traffic, forests, rivers, lakes, etc.
    - monitoring of core and metropolitan power, water, gas networks
    - targeted applications (e.g. museum guides, advertising)
    - crisis management (through the creation of ad hoc communication infrastructure)
- IPAC may also contribute to the better monitoring/control of large industrial facilities. This capacity of the IPAC nodes was also recognized by industrial partners of the consortium who plan to adopt IPAC as a rapid coordination mechanism between the different production line segments and the human resources.

The achievement of the aforementioned impacts will be evaluated both qualitatively and quantitatively by some *key performance indicators* such as the satisfaction of the participating companies, the number and quality of scientific publications, the interest of third parties for adopting/extending/integrating IPAC etc.

7 Coordinator Contact Details

For more details about the IPAC Project you can directly contact the Project Coordinator or visit the IPAC Web site:

Project Coordinator Name: **Eleni Daoukou**
Institution: **Siemens A. E. Electrotechnical Projects and Products**
Email: **Eleni.Daoukou@siemens.com**
Tel: **+30 210 6265618**
Fax: **+30 210 6265637**

Project Web Site: **http://ipac.di.uoa.gr**