A Proactive Approach for Information Sharing Strategies in an Environment of Multiple Connected Ubiquitous Devices

Remus-Alexandru Dobrican
Computer Science and Communication Research Unit
University of Luxembourg
Luxembourg, Luxembourg
remus.dobrican@uni.lu

Denis Zampunieris
Computer Science and Communication Research Unit
University of Luxembourg
Luxembourg, Luxembourg
denis.zampunieris@uni.lu

***Abstract*—The current trend in Ubiquitous Computing is to develop smaller, smarter and adaptive devices that are continuously connected to the Internet through wireless technologies. This offers a great opportunity for cooperation between these devices. Although much related work has been performed on collaboration techniques between smart devices, most of the work is not suited for large-scale networks, where their number can increase significantly. Following this context, we investigate how information is exchanged and handled in a distributed way in a Collaborative Ubiquitous Environment. To be perceived as collaborative and proactive, each single component has to coordinate with the others in smart way. In order to achieve these key features, we introduce the concept of Global Proactive Scenarios (GPASs). To support the deployment of these scenarios, we propose a specially designed intelligent architecture for ubiquitous systems that allows them to be context-aware, proactive and self-adaptive. Furthermore, we provide an example of joint effort for illustrating the full potential of a network of interconnected proactive systems.**

Keywords—ubiquitous computing; smart devices; collaborative; proactive; context-aware; self-adaptive.

I. INTRODUCTION

Parallel advances in hardware and software technologies [1] have led to the development of smarter and smaller devices that are capable of providing better and highly customizable services to their users [2][3][4]. These technological advances begun to introduce new information sharing techniques and new computational models. The increasing computing power that is nowadays assigned to objects, from home appliances to wearable technology, has made it possible to continuously collect information about the surrounding environment. The users' behaviour, activities and preferences are also taken into account for improving their experience and their quality of life.

The tendency of integrating smart objects in our daily lives has contributed to the appearance of multiple Ubiquitous Environments. Such environments include classrooms, university campuses, playgrounds, office buildings, homes, hospitals or even entire cities. It is essential that components of such systems are able to interact [5]. They should contain efficient mechanisms for supporting distributed algorithms that will allow them to interact, share information and perform joint actions. This new research direction is leading to a re-evaluation and remodelling of ubiquitous systems.

Collaborative ubiquitous environments raise a series of new questions and challenges in terms of development and management. First, in order to achieve their full potential, collaborative systems should aim at providing mechanisms capable of achieving the best results with the available shared resources. Second, they should be in a continuous adaptation process rather than being static. Should there be a new kind of dynamic software applications that would provide highly innovative services to the user or should the existing applications be adapted to cope with the new technological advances? Another aspect to be taken into account is the quality of the interaction between the users and their mobile and embedded devices. The user's privacy is also a major challenge when designing ubiquitous systems that exchange information. Future ubiquitous systems and applications will be characterized by their ability to detect changes in the context and to adapt their behaviour according to these changes in order to align with the user's needs. They will not only allow users to interact with shared objects but also to communicate and coordinate their actions [6].

The contribution of this paper is threefold. First, we propose a flexible architecture capable of supporting a complex collaboration between embedded proactive devices. Second, we investigate which type of devices could support the proposed architecture and how would they be grouped while being spread across multiple networks. And third, to stimulate the future development of intelligent applications and services on top of ubiquitous devices, we introduce a new methodology for sharing knowledge between mobile and embedded devices.

The rest of this paper is organized as follows. Section 2 discusses related research in the area of Ubiquitous Computing, Proactive Computing and collaborative information sharing techniques. In section 3, we present a middleware-based architecture for ubiquitous systems capable of supporting multiple type of cooperation in large-scale networks. Section 4 analyses which types of devices and gadgets could integrate the proposed architecture and how they could interact while scattered across various networks. Section 5 introduces the notion of Global Proactive Scenario together with its objectives and effects. Section 6 shows how GPASs are designed and implemented. Key properties that may emerge from this structure are highlighted and briefly discussed in section 7, and section 8 concludes the paper.

II. State of the Art

The research fields of Ubiquitous Computing and Proactive Computing are very broad. While Ubiquitous Computing aims at integrating new technology into the objects/devices we use in our daily lives, Proactive Computing concentrates more on the scenarios that may occur when using these objects. We only focus on the previous work that connects these two research fields. Also, we investigate the techniques that were developed for facilitating the cooperation between smart devices.

A. Ubiquitous Computing

Weiser came up with the idea of Ubiquitous Computing [7] in the late 80's. In his opinion, computer systems were to become invisible to the eyes of the user in order to not disturb and distract him/her in his/her daily tasks. A various range of devices such as smartphones, wearables, tablets, laptops and RFID-based systems are now part of so called Ubiquitous Environments. The increasing need of collaboration between Ubiquitous Environments has been noticed and discussed in [8]. In [9], the authors notice that ubiquitous computing has opened the doors for a new kind of applications, where devices collaborate on behalf of their users.

B. Proactive Computing

Proactive Computing is quite a new area of research in which the purpose is to study how software systems can take measures by themselves to cope with foreseen situations and events. Tennenhouse introduced the concept in 2000 [18] together with the intention to move from human-centred computing to human-supervised computing. Proactive systems should be able to act on their own initiative and make decisions for their users [19].

The need of having software that handles data proactively has led to the development of the first Proactive Engine in 2006 [20]. It was conceived as a complex mechanism on top of which Proactive Rules could operate. A Proactive Rule performs an analysis of particular data or events and performs actions, like, for example, sending notifications to users.

The Proactive Engine was initially developed to overcome the limitations of a Learning Management System (LMS), e.g. Moodle™. Such limitations included restricted interaction and collaboration between the users of the LMS or the fact that the LMS was not taking any actions for preventing unwanted situations like missing the deadlines for inscribing into the exams, missing assignments or even assisting users with the use of an LMS. Experiments conducted on an LMS showed that using a Proactive Engine can improve its assignment system [21], can increase the online participation of the students [22] and can raise the chances of students to perform better in their final exams [23].

Previous work focused mainly on making single systems proactive and not having multiple proactive systems exchanging information. The idea of including a Proactive Engine on mobile devices was brought up only recently [24]. The initial architecture was modified and the new structure, called the Local Proactive Engine (LPE), was enhanced with interconnected parts like a Context-Manager, a Rules Engine, a local database for storing information and a Notification Manager. These parts can be useful only in the case of mobile devices; otherwise they are too complex to run on top of smaller wearable devices that have limited computing capabilities.

C. Architectures and Information Sharing Strategies for Collaborative Ubiquitous Systems

Cross-device collaboration approaches have been developed for different purposes in various fields such as Computer Supported Collaborative Learning (CSCL) [10][11], Ambient Intelligence (AmI) [12], Urban Public Transport Systems [13] and many more. Most of these techniques, however, support only a limited range of devices that would engage in collaborative actions, mostly smartphones and tablets. They do not explore all the smart devices and gadgets, with computational capabilities, that can be found in a ubiquitous environment.

In [14], the authors described an example of cross-device collaboration between portable and embedded devices. They do not provide a clear example of how the inter-device communication is realized nor if the collaboration process can be realized among other intelligent devices. However, they realize that an appropriated mechanism is needed for providing personalized services to the user. Similarly, a middleware that can act proactively and provide an interaction mechanism for Ambient Intelligence embedded devices is proposed in [15] for taking control of all its distributed components.

Another device collaboration system, based on a context-aware architecture, for content sharing between heterogeneous devices is proposed in [16]. Context is captured at an internal and at an external level for both, the devices and their users. This structure is able to anticipate the user's intention and therefore to select personalized user-centred services. The Device Collaboration Server is the central point of the whole environment, where all the data is accumulated and analysed. It is also in charge of the cooperation between all the devices. The limitation comes from the fact that RFID tags attached to all the devices are used only for capturing location information. Other information regarding the user's preferences and behaviour is not taken into account as the devices are missing a more complex architecture, designed to provide more information. A multi-level architecture modelling approach for collaborative ubiquitous systems is presented in [17] as a framework for developing and implementing cooperating applications. The framework is divided into three layers: the application level, the collaboration level and the middleware level. A model is proposed for each level, together with possible implementation techniques. Nevertheless, the suggested architecture captures only instant context information, causing possible false interpretations by the system, due to a lack of past historical data.

A methodology on how to make Proactive Engines exchange information in various networks of wearable devices and ubiquitous systems has not yet been established. Moreover, the effects of such cooperation are yet to be explored.

III. A Proactive Architecture For Ubiquitous Devices

In this chapter, we propose a middleware-based architecture for Ubiquitous Devices called Proactive Middleware Architecture (PMA). This architecture is intended to be a robust platform for creating standardized means to share information between ubiquitous devices. It is also a starting point for developing smart applications that will be able to perform cooperative reasoning. Such an architecture is able to capture relevant *context information*, support *distributed reasoning*, act *proactively*, provide *adaptation* mechanisms and allow multiple *collaborative actions* at the same time.

Context information is typically obtained from sensors and actuators embedded into the smart devices, but can also be acquired from applications or from the user's behaviour. It comes under various forms [25] such as physical context, e.g. noise level, temperature or traffic conditions, as computing context, e.g. network connectivity or communication bandwidth, as user context, e.g. user's location, profile or social status, or as time context, e.g. time of the day or season of the year. The proposed middleware architecture makes, under certain restrictions, the context information of each device available to the other devices. Hence, each PMA is a collaborative context information management system.

Distributed Reasoning is an essential feature for context-aware ubiquitous systems. The most complex part of distributed collaborative systems is reaching a common decision, which requires mutual agreement between all the participants. This is hard to achieve, especially in very large networks, where the number of devices equipped with a PMA can rapidly increase. Distributed Reasoning is thus necessary for determining the quality and validity of the data gathered by individual devices, for creating other meaningful context information, derived from local contextual information available on each device, and for triggering adaptation measures according to each specific situation.

Acting proactively requires having strategies for anticipating special events that may appear on the system. For collaborative actions it may be important to anticipate possible situations where joint actions would be beneficial. For example, two devices which detect that two users have common interests, are working on the same task and their location is close by, would spontaneously propose them to collaborate and exchange information.

Adaptation can be either behavioural or architectural. We focused more on providing our framework with behavioural adaptation or the ability of a device to modify its behaviour in order to adapt itself to changes in its operational ubiquitous environments. In distributed collaborative systems, adaptation to external context is necessary as it is richer in information that the local context of each device.

Collaborative activities including but not limited to joint supervision of common tasks, resource sharing and distributed processing will facilitate work on the smart devices. These collective actions often involve the communication, coordination and cooperation of the all the smart devices involved in the collaboration process. PMA supports the parallel execution of multiple collaboration actions at the same time.

The biggest advantage of such a middleware-based architecture is that users do not need to reason when and how to coordinate, and how to support each other in case of unexpected events or situations. The Proactive System does it silently, in the background, without the user's explicit involvement. However, the user can check and modify at any time the state of the system.

A. The structure of Ubiquitous Devices

As illustrated in Figure 1, the PMA's internal structure is composed of several interconnected components such as: the Rules Engine, a Local Database, a Rules Queue and a Package of Proactive Rules. These components compose the Middleware Layer where the data is gathered and processed.

The Rules Engine is a mechanism in charge of executing Proactive Rules. The Proactive Rules, queued in the Rules Queue, which are triggered iteratively by the Rules Engine. The time between two consecutive Iterations can vary and can be changed for increasing the performance of PMAs. Proactive Rules are divided into multiple packages, depending on their purpose. These packages include Adaptation Rules, Collaboration Rules and Coordination Rules. Each device is initially equipped with a basic set of packages and can be extended with other packages of Proactive Rules.

B. Storing information on PMAs

PMAs are capable of storing information on each device in the local database. They can keep an index of the other devices that were involved in an information exchange process and can store previous successful or unsuccessful collaboration. Keeping track of other devices can be useful in certain cases because they can be organized afterwards into groups either by

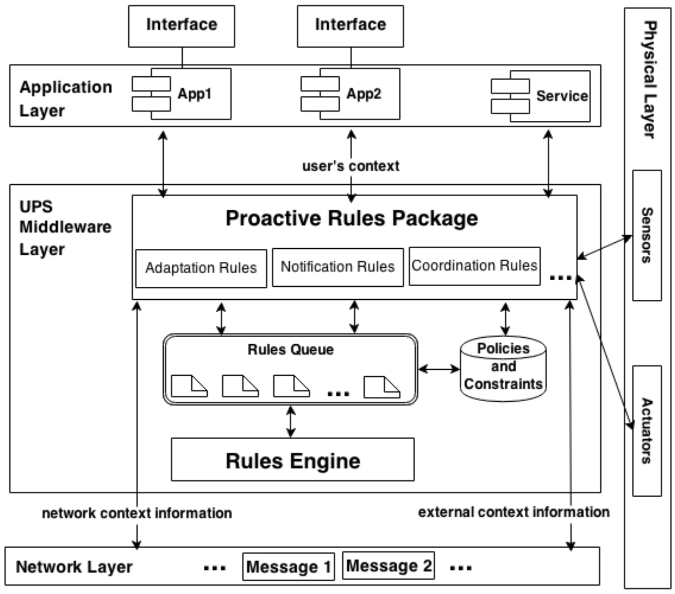

Fig. 1. A platform architecture for Smart Devices

the system itself or by the user. Policies and constraints, specific to each smart device, are also kept in the database and are checked by Proactive Rules before performing an action. It is enough for a device with a PMA to engage into a collaborative action only once for sharing its ID with the other devices and for obtaining a list of IDs that were used in the collaboration process. This list of IDs of the previous collaborations can be shared with the occasion of other cooperation tasks between the devices.

C. *Acquiring network and external context information*

The Network Layer is responsible for providing network context and external context, which is in fact information coming from the other devices. Smart devices equipped with PMAs use messages to exchange information, regardless of the communication protocol used in their networks or the network topology. For example, smartphones and tablets would exchange messages over Wi-Fi, while wearable devices such as smart watches or smart glasses use Bluetooth technologies.

D. *Interaction with local applications and services*

Devices that use such a middleware-based architecture represents a great opportunity for other applications, running on top of smart devices, to make use of its functionalities. These applications can activate Proactive Rules through the middleware's API for getting relevant local or external context information.

IV. NETWORKS OF DEVICES WITH PMAS

In the following section we investigate which kind of devices and gadgets would support PMAs and how would they be organized in various networks. Since PMAs are a particular category of distributed systems, they can be deployed on multiple devices, scattered across many different networks. This offers a great opportunity for collaboration, sharing and joint working. Providing a framework that facilitates collaborative tasks, where a collaborative task usually includes both individual and shared actions, will allow smart devices to benefit from the general knowledge gathered by each device and to make smarter decisions.

A. *Smart Things*

Generally speaking, smart things can be either real or virtual. The term *thing* refers to something that can be perceived, seen or felt by humans, while the term *smart* encompasses multiple properties such as context-awareness, self-adaptiveness, proactivity, self-governance and cooperativeness. Smart things can be equipped with various levels of intelligence. For instance, a coffee mug with a basic RFID tag attached to it would only have a limited capacity of communicating and computing, while a smart thermostat would be able to learn from the behaviour of its users, could be controlled remotely and even adjust itself according to the ambient conditions around it. Smart things can be classified into various categories based on their appearance and their functionality [26]. We only focus on smart devices or smart objects that can already support the proposed middleware-based architecture.

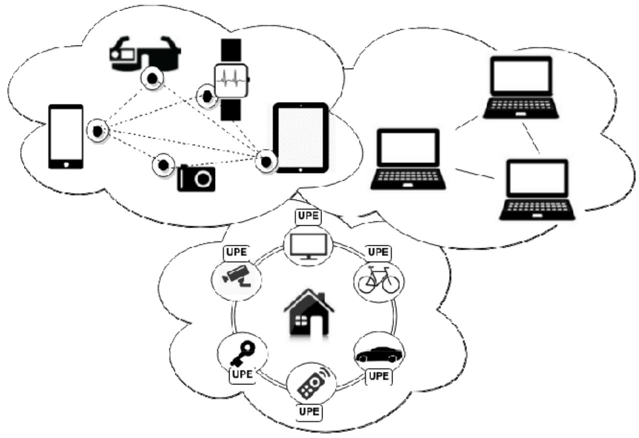

Fig. 2. Multiple connected networks of Smart Devices

B. *Smart Devices*

Smart devices are systems equipped with processors and software that gives them enough computational power to perform complex actions. They are not only ingenious systems capable of fitting in our pockets; they are smart because they are connected. Unlike traditional devices, which were assisting humans in doing a job, smart devices are acting on their own with minimal intervention from their users. They are characterized by the ability to run many applications in parallel and to collaborate with other intelligent systems. Smart devices are either highly developed equipment such as smartphones and laptops, or common objects like keys, watches, glasses, known as wearables.

1) *Wearables*

Wearables are a particular type of smart devices. Wearable technology is aiming at embedding computers into all the objects we use in our daily lives such as clothes, watches, footwear and headwear.

2) *Intelligent Systems in a Smart Home*

A Smart Home is an example of ubiquitous environment containing lots of connected smart systems and devices such as intelligent bulbs, security cameras, climate control systems, smart remote controls and smart keys.

3) *Ubiquitous Proactive Devices (UPDs)*

This category of devices includes all those objects that are capable of integrating a PMA into their structures such as smartphones, tablets, smart watches, smart glasses, smart cameras, vehicles, bicycles, smart televisions and the list may continue. Users can interact with these UPDs either through their touch screen or by special gestures or movements.

C. *Interconnecred UPDs*

As shown in Figure 2, there are many possibilities of finding smart devices organised into various networks. Even more, these networks can cooperate and become part of a bigger global network enhanced with ubiquitous intelligence. Networks differ in terms of size, from small-sized networks with only a couple of UPDs to large-sized networks with hundreds or thousands of UPDs. Users could build their own networks of UPDs, where he/she could use specific applications and services. In this kind of personal networks, devices would benefit from simpler communication and cooperation algorithms. Also, security and privacy aspects would be much easier to be handled.

In order to explore the possible benefits of these kind of networks and for assuring a smooth co-operation between the above smart devices we introduce Global Proactive Scenarios (GPASs).

V. Global Proactive Scenarios (GPASs)

GPASs are information sharing mechanisms with the following characteristics: they can detect unexpected events, they dynamically collect information from devices with an integrated PMA, they provide strategies for cooperative reasoning and they support collective decision making processes.

In contrast with traditional distributed systems GPASs provide the possibility of achieving distributed reasoning between UPDs by the concurrent execution of various Proactive Rules. The main idea of GPASs is to look for the minimal set of relevant information to perform a local or collective task. And this has to be done in an efficient matter as there is abundant information coming from all the sensors in a ubiquitous environment. A balance needs to exist between the proportion of retrieved data and the precision of processed data. In practice, it is hard to achieve this balance. To address this problem, GPASs activate local Proactive Rules on devices that have an integrated PMA for retrieving and processing data coming from their embedded sensors. After it is processed, only the relevant data is exchanged between the devices.

GPASs are suited for large-scale networks, where the number of smart devices involved in a collaboration process is increasing very fast, because GPASs support multiple strategies for sharing information. These strategies include approaches where, for example, only one UPD has a global overview of all the information and the other UPDs have only partial information, or, where each UDP has a global view of what is happening. Establishing a certain strategy depends on the purpose of the application being developed on top of the UPDs.

GPASs consist of various types of Proactive Rules like Adaptation Rules, Cooperation Rules, Coordination Rules, Communication Rules and Notification Rules. Proactive Rules are divided into multiple categories according to their purpose. Adaptation Rules are in charge of modifying or adapting the behaviour or/and the Graphical User Interface (GUI) of different devices or applications. Notification Rules are used for sending personalized messages, hints or alerts to the users. Communication Rules contain the logic for ensuring that data is sent and received by the UPDs. Cooperation Rules contain reasoning algorithms for supporting collaborative actions, while Coordination Rules are specifying actions that help distributing the knowledge accumulated by the UPDs.

In Figure 3, a Coordination Proactive Rules is presented in pseudo-code. It is part of a GPAS and it is in charge of checking if the adaptation actions were performed locally by the UPD. If yes, the user is notified about the outcome and a Communication Rule is created for informing the other UPD of the actions that were taken.

```
                    Proactive Rule R[i]
data acquisition
    ids [] = adaptationActionsIDs []
    deviceReceiverID = initiatorID
    commandID = receivedCommandID
activation guards
    return checkDB(adaptationWasSuccesfull(ids[]))
conditions
    return userWasNotNotified(msg, fullText)
actions
    sendNotificationToUser(msg, fullText)
rules generation
    if(activationGuard)
        activate CommunicationRuleRx(okMsg,
        commandID, deviceReceiverID)
    else
        cloneRule (R[i])
```

Fig. 3. An example of a Coordination Proactive Rule

A major advantage of our approach for information sharing is that new GPASs can be elaborated to meet new user requirements. The starting point would be to clearly define which situation would the GPAS respond to. Then, to develop a set of Proactive Rules that would accomplish the goals of the GPAS and that would be in charge of specific actions. A concrete example of designing and implementing GPASs is provided in study case in the next section.

A. Triggering GPASs

The two most common ways of triggering GPASs are either the occurrence of a foreseen event or the interactions of users with the screens of their devices. Foreseen events are detected by Proactive Rules, which are acting as background services with a clear purpose of monitoring the environment of each device. For example, running low on battery on a smartphone is a foreseen event. If the battery level drops under a certain percentage, specific action are taken, e.g. the luminosity of the screen of the smartphone is reduced to a minimum level. When interacting with mobile devices or embedded devices, the user's input can divided into explicit or implicit instructions. Explicit instructions refer to precise and clear input from the user, while implicit instructions point out commands which are not directly expressed but they are implied. For instance, if a user wants to share his/her location with a group of friends, he/she will perform an explicit command by pressing a certain button on the screen of his/her device.

B. The Objectives of GPASs

The design of GPASs is aiming at numerous objectives. With the help of GPASs, the local information available on a smart device is distributed to the other UPDs, context is captured at various levels, expected situations are detected and treated accordingly, resources shared and general knowledge is gathered for achieving a ubiquitous intelligence capable of taking smart decisions. From the user's point of view, only the effects of GPASs are noticeable and not how they are activated or how they manage to exchange information. Thus, it will not be necessary for the users to have advanced technical skills for using smart devices, instead they will be able to focus more on their tasks and achieving their goals.

C. *The Effects of using GPASs*

GPASs enable devices to perform a wide range of group tasks and to benefit from a collective intelligence, which is far more effective than the knowledge of each single device. They can change the state of the ubiquitous systems as well as being affected by the system's state changes. The effects of GPASs can be measured by developing test cases where the performance of the UPDs and frequency of triggering Proactive Rules can be evaluated.

Several research question are rising from the new perspective of having device equipped with a distributed middleware architecture capable of executing GPASs. One of the possible questions would be if enhancing smart devices with UPEs will lead to ubiquity. Another one would be what opportunities would open having multiple interconnected networks of UPDs capable of acquiring, analysing and sharing large amounts of information, and what are the challenges of such an approach. Understanding how collaborative techniques such as GPASs would benefit entire organisations or institutions like universities or schools is still an ongoing research topic.

VI. Case Study

In this section, we introduce a study case for illustrating better how GPASs are used for solving different situations. We analyse the role of the GPAS in the study cases, together with its objectives and consequences it has on the outcome.

A. *Basic solution: Automatically turning off the sounds of a smartphone by using local Proactive Rules*

This study case is related to the problem of attending a meeting and forgetting to disable the sounds of the smartphone. This situation is quite common and can have unpleasant results. For example, in a company, a sales meeting takes place with a very important client. While the sales department starts to present the product to the client and suddenly the smartphone of one of the company's employees start to ring. This shows, among others, a lack of respect to the client and can even lead to serious consequences. This situation may also appear during presentations at most of the conferences organized around the world. The UPDs are able to detect that a meeting is scheduled with the help of local Proactive Rules and, as a consequence, they can then take proactive measures for turning off the sounds and alarms. First, a Proactive Rules would be in charge of acquiring relevant context information. More explicit, it will check local applications such as the calendar, schedule or planner for determining if the user has a meeting. If a meeting is detected, two other rules get activated: a Proactive Rule for checking that special constraints are not violated and an Adaptation Rule for switching off the sounds of the phone. There are special cases, where the user wants to have his/her phone turned on all the time, e.g. an employee waiting for phone call from the hospital, where he's wife is about to give birth. User's preferences are a very important matter and UPDs are taking them in account when performing automated tasks.

B. *Extended solution: Using a GPAS*

In many situations, context information like a scheduled meeting in the calendar application is not available on the targeted devices. This information can be however obtained externally, from other UPDs. In order to benefit from having networks of UPDs that are able to share their local knowledge, we propose a GPAS which is in charge of acquiring external context information.

It is enough for one UPD to know precise information about the meeting, like from what hour does it start, on which day and who is supposed to attend that meeting, to start sharing it. The UPD which has this additional information will automatically trigger a GPAS for alerting the devices of the other participants about the meeting. The GPAS will send commands to activate local Proactive Rules in charge of disabling the sounds of each smartphone and of checking that the preferences of each user do not prevent the proactive actions. The status of the users, "*in a meeting*", can be shared with other applications and, if the user allows it, received phone calls will be replied with a specific message, saying the user is in a meeting and he/she cannot answer. The owners of the smartphones attending the meeting will thus avoid unpleasant interruptions. They will have their devices in their pockets, in silent mode, which will permit them to stay focused on the main subject of the meeting.

VII. A Logic sharing Process over a network of UPDs

GPASs contain techniques for achieving collaboration and coordination between multiple UPDs. To achieve dynamic data distribution, UPDs are designed to keep an open communication channel. This allows them to receive commands or information requests from other UPDs. UPDs support both types of reasoning: synchronous and asynchronous. If UPDs are online at the same time they can support real-time activities. Asynchronous reasoning takes more time and a couple of extra steps but it is useful in large networks of UPDs, where the number of UPDs entering and leaving the network changes very fast.

Raw data is pre-processed locally on each UPD before it is sent over to the other smart devices. Initial input coming from different sensors can contain incorrect recordings, missing values and can be inconsistent. Even after the data is locally analysed by a UPD it can still contain errors. With the help of GPASs and their distributed methods for analysing data the majority of these local errors can be avoided.

A UPD is called *Initiator* if it begins the cooperation process and it triggers a GPAS. On the other hand, *Receivers* are those UPDs that participate in a cooperation process but which do not start the GPAS. UPDs can have multiple role at the same time, in different collaborative tasks. A UPD can be the Initiator of a task and, simultaneously, be a Receiver in two other GPASs. *Receivers* can also become *Initiators* of other GPASs by external requests from other *Initiators*.

Figure 4 shows a sequence diagram with a possible execution of a GPAS between three devices: the Initiator and two Receivers. The Initiator activates the GPAS as consequence of an expected event, e.g. the location change of the UPD. The IDs of the Receivers targeted to be part of the collaboration process are extracted from a list found in the Initiator's local database, and then, a Communication Rules is activated for sending these particular commands to the Receivers. In this case, the commands send over to the Receivers contain specific instructions for initiating two Proactive Rules, Rx and Ry. An example of the content of a possible message exchanged between the Initiator and the two Receivers is given in Figure 5. The instruction type is specified in the first field, in the second one a list with Proactive Rules to be executed is given, in the third one information is given about the UPDs involved in the collaboration process, in the fourth one multiple parameters are provided and in the fifth field the Initiator's ID is provided.

When the Receivers get the commands, they check if they do not contravene their local constraints and policies. In case they do not, Receivers activate Proactive Rules Rx and Ry. Then, a message containing an acknowledgement is sent over to the Initiator. The messages may arrive at different times as the Receivers may take longer periods of time to process the commands or because they are located on different networks. The Initiator has to wait for the acknowledgement messages before sending other requests to the Receivers. If these messages do not arrive in a given period of time, the Initiator will resend the initial commands. After the acknowledgement messages, the Initiator sends a request to the Receivers for specific information. Once the Receivers process the request and relevant information is found, the responses are forwarded to the Initiator. This is an example of a collaborative action performed by the Initiator and the two Receivers. It contains the coordination and communication process between the three devices.

At that moment where the Initiator has all the information it needed from the Receivers it will decide if it takes or not proactive measures. This measures can refer to local adaptation of the behaviour of each devices, global adaptation of the interface of all the devices or even notification messages sent to the user to let him/her know that a successful collaboration process took place.

The execution of the GPAS in Figure 4 is only a basic example designed to illustrate, in a couple of steps, how information is exchanged between three smart devices. In complex cases, the number of UPDs would be much higher as well as the number or steps required to reach a solution in response to an event. Also, GPAS may contain Proactive Rules for asking for the user's input in order to get information that is otherwise not available. For instance, finding a solution for a meeting at a certain hour for the employees of a company can involve the activation of multiple GPASs. These GPASs would start multiple rounds of negotiation and check if the final solution is according to the personal preferences of each employee.

VIII. PROPERTIES OF UPDS

Interesting properties like Context-Awareness and Self-Adaptiveness emerge from having interconnected UPDs scattered across various networks and exchanging useful information. These properties, combined together in a distributed ubiquitous environment of smart devices, lead towards a global computational intelligence that will extend the computing capabilities and the knowledge base of each individual UPD.

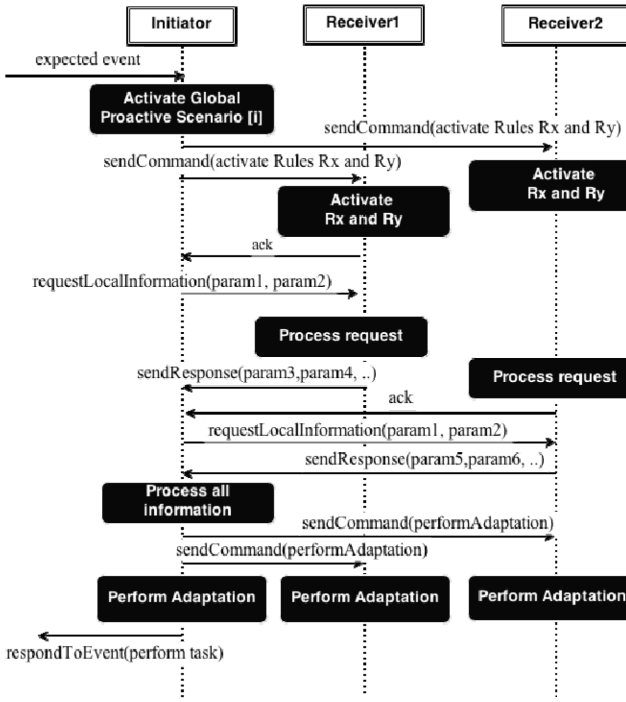

Fig. 4. A possible execution of a GPAS

Context-awareness refers to the ability of a UPD of detecting various situation, events and context changes that may occur. This is a key property for ubiquitous systems where the abundant data coming from their sensors can provide valuable knowledge about the surrounding environment and their users.

Self-adaptiveness on UPDs focuses more on the ability to change their initial behaviour based on new circumstances and on adapting to the changing context. When triggering GPASs, UPDs need to remodel their local routines and execute Proactive Rules that will require a certain degree of adaptability. For example, a UPD, running on a smartphone, is required to compute and send its location to another UPD that wants to perform a collaborative task. A Proactive Rule is activated for retrieving the coordinates and another one for sending them to the UPD that initiated the request. Computing and sending the location of the smartphone calls for a certain degree of adaptability from the receiving UPD, which needs to process the new tasks in parallel with its local tasks.

```
{   'instruction': 'activate rules',
    'rules': ['R001', 'R002'],
    'receivers': ['ID1','ID2'],
    'parameters': ['param1','param2'],
    'sender ID':'ID'   }
```

Fig. 5. An example of JSON-based message exchanged between UPDs

UPDs are being proactive by acting in behalf of the users in certain situations which occur, instead of waiting, each time, for the user's explicit command. GPASs are examples of distributed techniques, used for helping the users by assisting them in their daily tasks and guiding them through the challenges that may appear as a result of using smart devices.

These new properties will not prevent classical properties that a collaborative system should have like *privacy*, *security* and *trust*. To avoid privacy issues and increase the trust of the users, security policies and constraints are stored locally on each system. Access to sensible context information can be made available only with the implicit consent of the user.

IX. Conclusions

The huge potential ubiquitous environments have, where information is abundant, is now ready to be explored. The current lack of important and advanced properties for smart devices linked to Proactive Computing represented our motivation for proposing a middleware-based architecture for coordinating and running GPASs. Also, the increasing number of networks of interconnected smart devices represents a big opportunity for using a middleware-based architecture capable of performing joint actions, sharing resources and creating a global ubiquitous intelligence. The proposed architecture solves the problem of how to enhance smart devices with collaboration and cooperation techniques. GPASs enrich the computational possibilities between smart devices and open a new perspective on how smart devices exchange information and perform collaborative activities. Having smart devices being able to cooperate in a proactive manner benefits not only single users but entire communities.

References

[1] J. Sen. "Ubiquitous Computing: Applications, Challenges and Future Trends", book chapter in *"Embedded Systems and Wireless Technology: Theory and Practical Application*", Chapter No. 1, pp. 1-40, CRC Press, Taylor & Francis Group, USA, 2012.

[2] A.L. Tuán, H.N.M. Quoc, M. Serrano, M. Hauswirth, J. Soldatos, T.G. Papaioannou, and K. Aberer, "Global Sensor Modeling and Constrained Application Methods Enabling Cloud-Based Open Space Smart Services", *Proc. of the 9th Inernational Conference on Ubiquitous Intelligence and Computing/Autonomic and Trusted Computing*, pp.196-203, 4-7 Sept. 2012.

[3] T. Nakajima, T. Yamabe, and M. Sakamoto, "Proactive ambient social media for supporting human decision making", *Proc. of the 8th Inernational Conference on Ubiquitous Intelligence and Computing*, vol. 6905, pp. 25-39, 2011.

[4] C.-Y. Lin, C.-T. Chang, M.-T. Chen, Z. Ruan, A. Hsueh, Y.-Y. Chang, and R.-H. Hwang, "NuNote: An Augmented Reality Social Note Posting Service", *Proc. of the 8th Inernational Conference on Ubiquitous Intelligence and Computing*, vol. 6905, pp. 94-108, 2011.

[5] V. Serrano, and T. Fischer, "Collaborative innovation in ubiquitous systems", *Journal of Intelligent Manufacturing*, vol. 5, pp. 599-615, 2007.

[6] I. H. Dos Santos, A. Raposo, P. G. Rodrigues, R. P. Souza, and W. G. do Amaral, "A collaborative environment for offshore engineering simulations", *Collaboration and Technology* , pp. 191-206,, 2011.

[7] M. Weiser, "The computer for the 21st century.", *Scientific American* 265, no. 3: 94-104, 1991.

[8] G. Sancho, I.B. Rodriguez, T. Villemur, and S. Tazi, "What about collaboration in ubiquitous environments?", *Proc. of the 10th Annual International Conference on New Technologies of Distributed Systems (NOTERE)*, pp. 143-150, IEEE, May 2010.

[9] M. A. Kabir, J. Han, A. Colman, and J. Yu, "SCaaS: A Platform for Managing Adaptation in Collaborative Pervasive Applications", in *Proc. of On the Move to Meaningful Internet Systems(OTM 2013)* Conferences, pp. 149-166, Springer Berlin Heidelberg, January 2013.

[10] D. Spikol, M. Milrad, M. Svensson, and O. Petersson, "Mobile Collaboration Tools and Systems to Support Ubiquitous Learning", in *Proc. of the 4th Annual Conference on Collaboration Technologies*, 2008.

[11] P. Byrne, "MUSE-Platform For Mobile Computer Supported Collaborative Learning", Doctoral dissertation, University of Dublin, Trinity College), 2011.

[12] C. Maciel, P.C. de Souza, J. Viterbo, F.F. Mendes, and A.E.F. Seghrouchni, "A Multi-Agent Architecture to Support Ubiquitous Applications in Smart Environments", in *Procedings of the 5th International Workshop on Collaborative Agents – Research & Development, CARE for Intelligent Mobile Services*, 2014.

[13] J. G. Vieira, T. Galvão, J. F. e Cunha, P.M. Costa, and J. Pitt. "Smart Mobile Sensing for Measuring Quality of Experience in Urban Public Transports", *2nd Workshop on Smart Mobile Applications*. 2012.

[14] N. Kohtake, K. Matsumiya, K. Takashio, and H. Tokuda, Smart device collaboration for ubiquitous computing environment", in *Workshop of Multi-Device Interface for Ubiquitous Peripheral Interaction, Fifth International Conference on Ubiquitous Computing (UbiComp2003), USA*, October, 2003.

[15] G. Cabri, L. Ferrari, L. Leonardi, and F. Zambonelli, "Connecting Ambient Intelligent Components via Dedicated Middleware: an Agent Approach", in *Workshop on Ambient Intelligence-Agents for Ubiquitous Environments* (AAMAS 2005), Utrecht, The Netherlands, July 2005.

[16] S. Y. Kwon, C.Y. Choi, and J.H. Lim, "Device Collaboration System Based on Context-Aware Architecture in Smart Environment", in *Future Information Technology*, pp. 256-263, Springer Berlin Heidelberg, 2011.

[17] I. Bouassida Rodriguez, G. Sancho, T. Villemur, S. Tazi, and K. Drira. "A model-driven adaptive approach for collaborative ubiquitous systems", in *Proc. of the 3rd workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing*, pp. 15-20. ACM, 2009.

[18] D. Tennenhouse, "Proactive Computing". *Communications of the ACM*, vol. 43, issue 5, pp. 43-50, 2000.

[19] A. Oulasvirta, and A. Salovaara, "Six modes of proactive resource management: a user-centric typology for proactive behaviors", in *Proc. of NordiCHI*, ACM Press, pp. 57-60, 2004.

[20] D. Zampunieris, "Implementation of a Proactive Learning Management System", in *Proceedings of E-learn*, AACE Press, pp. 3145-3151, 2006.

[21] D. Shirnin, S. Reis, and D. Zampunieris, "Experimentation of Proactive Computing in Context Aware Systems: Case Study of Human-Computer Interactions in e-Learning Environment", in *Prooceedings of CogSIMA*, IEEE, pp. 269-276, February 2013.

[22] R. Dobrican, and D. Zampunieris, "Supporting collaborative learning inside communities of practice through proactive computing", in *Proc. EDULEARN13*, pp. 5824-5833, 2013.

[23] S. Coronado, and D. Zampunieris, "Continuous proactivity in learning management systems", In *Proceedings of EDUCON 10*, IEEE Publishing, vol.1, 201-205, 2010.

[24] R. Dobrican, and D. Zampunieris, "Moving Towards a Distributed Network of Proactive, Self-Adaptive and Context-Aware Systems", in *Proc. of the 6th International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE)*, pp. 22-26, IARIA, 2014.

[25] G. Chen, and D. Kotz, " *A survey of context-aware mobile computing research*", Technical Report TR2000-381, Vol. 1, no. 2.1., Dept. of Computer Science, Dartmouth College, 2000.

[26] J. Ma, L. T. Yang, B. O. Apduhan, R. Huang, L. Barolli, and M. Takizawa, "Towards a smart world and ubiquitous intelligence: a walkthrough from smart things to smart hyperspaces and UbicKids", *International Journal of Pervasive Computing and Communications*, vol. 1, pp. 53-68, 2005.