

Facilitating nomadic interaction through a self-managed area-based support

GÓMEZ, Victor[†], MENDOZA, Sonia^{``}, GARCÍA, Kimberly^{```} and DECOUCHANT, Dominique^{````},^{``````}

[^] Instituto de Informática, Universidad de la Sierra Sur, Guillermo Rojas Mijangos S/N, esquina Av. Universidad, Ciudad Universitaria, C.P. 70805, Miahuatlán de Porfirio Díaz, Oaxaca, Mexico.

^{^^} Computer Science Department of Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Avenida IPN #2508, Col. San Pedro Zacatenco, Del. Gustavo A. Madero, C.P. 07360, Mexico city, Mexico.

^{```} Laboratoire LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75252 Paris Cedex 05, France.

^{````} Information Technologies Department, Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Avenida Vasco de Quiroga #4871, Col. Santa Fe, Del. Cuajimalpa de Morelos, C.P. 05348, Mexico city, Mexico.

^{``````} Laboratoire LIG –CNRS, 681 Rue de la Passerelle, 38400 Saint-Martin-d'Hères, France.

Received June 30, 2015; Accepted September 30, 2015

Abstract

Ubiquitous computing integrates network supports (from small sensors to powerful devices) into the people's working and domestic environments, which can be organized in self-managed areas. In this context, a self-managed area should provide useful information about the state of daily life objects (e.g., power failure of a refrigerator) without needing user intervention. Service discovery systems are essential to achieve this sophistication as they enable computing services to discover, configure and communicate with others, in order to facilitate the user's tasks. Although the solutions proposed by these systems cannot be easily reused in other contexts (e.g., support for human collaboration) their different solution approaches provide useful reference points for the design of future discovery systems. In this paper, we describe a service discovery system called SEDINU (Service Discovery for Nomadic Users), which provides a computational and communicational support for facilitating: 1) interaction between a nomadic user and the services provided by the current area where he is located, and 2) collaboration between nomadic users located in such an area. These kinds of interactions depend on a specific context defined in terms of users' roles, location and goals. The proposed system allows a nomadic user: 1) to select and execute a service as many times as he needs; 2) to interact and collaborate with other nomadic users; 3) to use proactive applications, in order to easily interact with the services offered by the current self-managed area, e.g., to gather information required by an area, such as user identification and profile; 4) to create ad-hoc networks within the current area, with the aim of transferring relevant information between nomadic users and organization employees, e.g., e-vouchers and equipment delivery approval.

Ubiquitous environments, cooperative work; nomadic users; service discovery, self-managed areas; ad-hoc networks

Citation: GÓMEZ, Victor, MENDOZA, Sonia, GARCÍA, Kimberly and DECOUCHANT, Dominique. Facilitating nomadic interaction through a self-managed area-based support. ECORFAN Journal-Spain 2015, 2-3: 161-179

[†] Researcher contributing first author.

Introduction

Internet users have recently observed important changes arising from the use of communication technology, mostly accessed by means of wired/wireless connections and mobile telephony, to satisfy social needs (e.g., multimedia message transmission [10]), welfare needs (e.g., home-health care applications [11][2]) and urban needs (e.g., GPS-based locative media applications [22][9]). The advantages of this communication technology become obvious in cooperative work environments, where applications and users easily need to access different services (e.g., Web browsing, printing, e-mail and database queries). In such environments, it is possible to assume that the connectivity provided by corporative networks not only offers a high bandwidth but also is reliable and continuous.

On the other hand, the increasing proliferation of heterogeneous but powerful mobile devices and the unstoppable progress of wireless networks allow considering the user [23] as a nomadic entity that evolves within a multi-computer and multi-user environment, where he employs several devices and smart applications to collaborate with other users anytime anywhere. However, when moving from one place to another, users can find substantial changes concerning: 1) the type of available services, 2) the communication technology, and 3) the network bandwidth and latency.

Wireless networks constitute the most suitable interconnection technology for the mobile nature of new devices [29]. In this way, users of smartphones can consult their e-mail anywhere; travelers can surf the Web from airports; tourists can use GPS to locate museums, restaurants or streets; scientists can interchange documents during a conference; people at home can transfer and synchronize information between smartphones and PCs.

However, ubiquitous environments also require efficient discovery systems to deal with: 1) the intrinsic dynamism of these environments, and 2) the growing number and diversity of devices that can provide and request services.

Some representative efforts have been realized, in order to develop service discovery systems [21]. In the academic field, we can mention the Intentional Naming System (Massachusetts Institute of Technology) [1] and the Ninja Service Discovery Service (University of California Berkeley) [12]. Likewise, software companies have integrated service discovery systems into their operating systems, e.g., Jini Network Technology (Sun Microsystems and Oracle) [18], Universal Plug and Play (Microsoft) [24] and Bonjour (Apple) [13]. In the opposite way, IBM Research (DEAPspace [17]), Salutation Consortium (Salutation [5]), IETF (Service Location Protocol [25]) and Bluetooth Special Interest Group (Bluetooth SDP [3]) propose service discovery systems that are independent from the operating system. Nevertheless, the reutilization degree of these solutions mainly depends on the way they were developed and provided [27][20].

Due to service registration and cancellation are dynamic operations, most of the discovery systems asynchronously notify their clients of the service availability [19]. However, the clients of these services are mainly programs, e.g., a printing service can be registered as a proxy object in a lookup server, which acts as a remote control. Therefore, these discovery systems do not support user-service interactions and even less user-user interactions. In addition, services are not provided following a user task-based organization, which supposes a more efficient, suitable and specialized support, but they are merely supplied according to a network organization, e.g., a network topology.

In this paper, we describe the SEDINU (*Service Discovery for Nomadic Users*) system, which facilitates user-service and user-user interactions within a specific context defined in terms of the user's roles, location and goals. More precisely, we are interested in defining a suitable support for ubiquitous environments characterized by users that move and collaborate within a large organization that includes several departments or administrative services. These organizational units serve as a basis for the definition of "self-managed areas", which define and administrate specific human tasks and provide collaborators with contextual services to carry out these tasks. A self-managed area provides nomadic users with services for: 1) activating other services (e.g., slide projection), 2) establishing planned collaborative sessions and ad-hoc network-based opportunistic interactions, and 3) performing specific tasks (e.g., collaborative production of diagnosis). The activation of these services depends on the nomadic user's current context.

This paper is organized as follows: after studying and providing a comparative analysis of related work (section 1), we define the "self-managed area/role attribution" and "hierarchical area organization" concepts and their principles (section 2). These concepts serve as a basis to describe the SEDINU functional schema and distribution architecture; we also present a human face recognition system that aims at facilitating the identification and location of known nomadic users (e.g., organization staff) in closed spaces (section 3). We validate our proposal by describing a use scenario that shows the functionality of SEDINU (section 4). Finally, we present the current research directions of this work.

Related Work

In this section, an analysis of the most important service discovery systems is presented.

We firstly describe their main characteristics and, at the end of this section, we make a comparison of the studied systems.

Service Location Protocol

The Service Location Protocol (SLP) proposed by IETF [25] defines a decentralized and extensible service discovery protocol for IP networks. The SLP infrastructure consists of three types of agents: 1) *user agents* perform the discovery of services that satisfy the needs of users' applications; 2) *service agents* announce the characteristics and location of the services, and 3) *directory agents* store information about the announced services.

SLP has two operation modes. In the former, a user agent sends multicast messages asking for a service, whereas service agents offering the service respond through unicast messages. In the latter operation mode, service agents register their services into a directory agent, where user agents look for a needed service.

SLP defines two modes for discovering directory agents. In the passive one, service and user agents listen to multicast messages, whereas directory agents periodically announce their own existence. In the active mode, service and user agents send multicast messages, with the aim of discovering directory agents. If one exists, service and user agents communicate with it by unicast messages, in order to register and look for services, respectively.

Ninja: Secure Service Discovery Service

The Secure Service Discovery Service (SSDS) [12] is a project of the UC Berkeley. SSDS is similar to other discovery systems but offers some scalability and security improvements.

SSDS is implemented in Java RMI but uses XML documents instead of Java objects for the description and location of services. The SSDS model consists of clients, services and Secure Discovery (SD) servers, which are integrated by a look up directory, a capability manager and a certification authority. An available service is announced by a SD server using periodic authenticated multicast messages, which contain the URL of the service.

Both the SD server and the clients can store information about the services in a cache memory. In addition, the system state can be completely built using multicast messages. This scheme facilitates the error recovery process by decreasing the user's manual intervention. SD servers are organized in a hierarchical way, so that new servers can be added dynamically. The server hierarchy is able to detect whether a server fails, in order to restart it or replace it.

Jini

Jini [18] is a service-oriented distributed architecture proposed by Sun Microsystems. Because Jini is based on Java, users must have a Java Virtual Machine (JVM) running on their device or must be represented by a JVM-enabled device. Jini defines three components: a) *services* are represented by Java objects that provide computing or control process on devices; 2) *clients* use services, so a service can be a client of another service; and 3) *directory services*, called Jini Lookup Services (JLS), serve as a means for clients and services to discover each other.

To register or look for a service, a JLS can be located by one of the following protocols:

1) the *Unicast Discovery Protocol* is employed when a JLS is already known, 2) the *Multicast Request Protocol* is used to look for a JLS or 3) the *Multicast Announcement Protocol* is employed by all the JLS to announce their availability. Jini implements a rent mechanism, which forces services to update their record periodically in a JLS, so that services can be located by clients.

A user looking for a service on the network first sends a multicast request to find a JLS. If one exists, the corresponding remote object is downloaded to the user's device; the user then employs this object to find the desired service. The service discovery process is made by matching the communication interface or by comparing Java attributes. If the JLS contains a valid service that implements the interface specified by the user, then a proxy of this service is downloaded to the user's device. From that moment, such a proxy is used to call the different functions offered by the service.

Universal Plug and Play

The Universal Plug and Play (UPnP) architecture [24], promoted by Microsoft, extends the original Plug & Play model for I/O devices to a dynamic environment, where multiple network devices offered by various suppliers can interact. UPnP uses a set of network protocols (TCP/IP) from the lower layers of the Internet model and allows devices to define their own APIs to implement these protocols, using the language or platform supported by such devices. UPnP also uses the Simple Service Discovery Protocol (SSDP) to discover IP-based services. SSDP can operate in the network with or without a directory or search service and works over open protocols and HTTP by means of unicast and multicast messages, respectively.

CROWDSOURCING

December 2015 Vol.2 No.3 161-179

When a service advertises itself on the network, it first sends an announcement message to notify its presence. In the case of multicast messages, the service sends this announcement to a reserved address. If a directory or search service is present, it can register this announcement, which contains two URLs: one identifying the published service and the other providing its description. When a client needs a service, the client can either directly access to the server through the URL provided in the service announcement or send a multicast search request. If the client discovers the service through this request, it can be directly treated by the service or by a directory or search service.

Bonjour

The Bonjour technology boosted by Apple is based on a previous work started at 1999 by the IETF Zeroconf group. The objective of this technology is to facilitate the creation of ad-hoc networks [13]. Bonjour uses DNS (Domain Name System) standard interfaces, servers, and package formats to look for services in a network. Bonjour provides service discovery support, using already existing DNS resource records, and performs queries that allow the user to get instances of a specific type of services.

Bonjour is based on a hierarchical structure of DNS services and interfaces for name resolution on distributed networks, such as Multicast DNS or LLMNR (Link Local Multicast Name Resolution). Given the type of service and the domain the client is looking for, Bonjour enables discovering name instances of the desired service that just use DNS queries.

DEAPSpace

The DEAPSpace system [17] was developed by IBM Research to efficiently work on single hop ad-hoc networks.

The main goal of this system is to provide information about frequent changes produced in the environment, considering the technical limitations of the devices using DEAPSpace. In this way, such devices are able to: 1) detect the presence of nearby devices, 2) share information of available services and 3) detect the unavailability of other devices.

DEAPSpace is based on the push technology [14] to allow all devices to keep a global view, which is periodically sent to all neighboring devices. The general view is updated when DEAPSpace receives the global view from each device. The main contribution of DEAPSpace is the definition of a service description format and a codification mechanism, which minimize the amount of data that need to be transmitted during the service discovery process.

Salutation

Salutation [5] is a service discovery system and a session manager developed by the Salutation Consortium. Salutation is an open and independent standard of operating systems, communication protocols and hardware platforms. Salutation solves problems concerning the discovery and access to services offered by a wide set of devices, which interact in an environment owning a high connectivity and mobility range.

The Salutation architecture defines three entities: Functional Units (FUs), Salutation Managers (SLMs) and Transport Managers (TMs). FUs define the most common services (e.g., printers, faxes and file storage) from the client point of view to guarantee interoperability. SLMs allow clients to discover network services and communicate with them.

TMs isolate SLMs from the transport protocol used to access them. In this way, to support a new transport protocol, it is just needed to implement a new TM entity, without modifying the SLM implementation.

The service discovery process can be performed using multiple SLMs. A SLM can discover some other remote SLMs and determine the registered services. Service discovery is carried out by comparing the required type of service, as specified by the local SLM, with the available type of services allocated in a remote SLM. Remote Procedures Calls are used to transmit: 1) the required type of services from the local SLM to the remote one, and 2) the response from the remote SLM to the local one.

Comparative Analysis

Figure 1 shows important characteristics of the previously analyzed service discovery systems. All the service discovery systems described in this section compare typical attributes (e.g., the type of service specified by a URL) or attributes of the communication interface (e.g., IP address and the port number of the service host) to check the existence and availability of the services requested by a client.

All the analyzed systems but Jini are language-independent. As Jini is totally dependent on Java, a running Java Virtual Machine is necessary on the involved devices.

Protocol	Language Dependent	Directory Based	Service Information Storage	Service Discovery Technique	Topology	Network type
SLP	✗	✓	Directory Agent (DA)	Centralized Service Repository	• Centralized • P2P	Subnetwork
Ninja SSDS	✗	✓	Lookup Service	XML Descriptions	• Client-Server	Hierarchical Service Structure
Jini	✓	✓	Lookup Service	Centralized Service Repository	• Centralized	Subnetwork
UPnP	✗	✗	Control Point	Requests Flooding	• P2P	Subnetwork
Bonjour	✗	✓	Hierarchic Directory	Centralized Service Repository	• Centralized • P2P	Intranet
DEAPSpace	✗	✗	Global View	Requests Flooding	• P2P	Mobile ad hoc 1-Hop
Salutation	✗	✓	Salutation Manager	Centralized Service Repository	• Client-Server • P2P	Any network Transport dependent

Figure 1 Comparative Analysis of Relevant Service Discovery Systems

Some service discovery systems use a directory to store information about services available on a network. Particularly SLP, Ninja SSDS, Jini, Bonjour and Salutation are based on a directory, which has different names according to the system (e.g., in SLP, it is called Directory Agent). To access services through a directory, the client first contacts the central directory to obtain the service description and then directly interacts with the service provider. On the contrary, UPnP and DEAPSpace do not use a directory. In this case, providers do not distribute the description of their own services to the rest of the network nodes but maintain it in a local cache of the device.

The service discovery technique used by the directory-based systems relies on a centralized service repository, excepting for Ninja SSDS, which uses XML-based descriptions. Systems that are not based on a directory send flooding requests to all the network nodes, in order to find the description of the requested service.

The topology of the system [21] defines how service descriptions and client queries are managed. The topology can be P2P, client-server or centralized. UPnP can be deployed in centralized or P2P topologies, whereas Jini operates inside a subnetwork. Ninja SSDS manages a client-server topology based on a hierarchical server structure. Depending on the transport protocol, Salutation can work through a client-server topology or a P2P topology inside any type of network.

All the analyzed systems provide support in terms of service location (e.g., Jini, UPnP and Bonjour) or network topology location (e.g., DEAPSpace and Salutation). Each one of these systems solves a different type of problems, such as service access control, device recognition, network connection or service identification. However, most of them are focused on domestic environments (e.g., locating multimedia services to receive a video call) or enterprise environments (e.g., determining available resources to fulfill a printing request). Therefore, these service discovery systems can not be used in other contexts such as collaborative environments.

Although the studied systems explore different aspects of service discovery in distributed systems, they do not offer a suitable solution to problems caused by the dynamic character of collaborative environments, e.g.,

The automatic detection of services offered by different areas of an organization or the creation ad-hoc networks to give support to user-service interactions and collaborations among users.

SEDINU Areas: Concepts and Principles

In this section, we introduce concepts and principles of areas in SEDINU, whose main goals are: 1) to discover nomadic users and services available in a self-managed area, and 2) to present them to the nomadic user depending on his current location, role, and goals. In section 2.1, we introduce the concept of self-managed area and the principle of role attribution. Then, in section 2.2, we explain how the different areas are hierarchically organized, following the well-known technique of interconnected and nested black boxes.

Self-Managed Areas

Nowadays, large organizations are divided into sub-organizations (e.g., administrative services, departments and sections) in order to be more efficient at providing users with specialized services. Sub-organizations are mapped to self-managed areas, which make the global organization administration easier, since resources (e.g., services, contextual information, workflows and roles) are managed and controlled in a distributed way. Thus, each self-managed area becomes an administration unit, which handles its own resources, while observing the general policies (e.g., security and invoicing) of the organization. The different areas exchange information for coordinating themselves, in order to assist nomadic users in achieving their partial and global goals within the organization. From a physical point of view, a self-managed area can be mapped to several buildings, a building or a part of a building.

Self-managed areas represent a hierarchy of sub-organizations, where higher levels provide general roles and activities, whereas lower levels offer specific ones. Figure 2 illustrates the principle of role attribution in a simple hierarchy, in which the areas A0 and A1 respectively denote the organization and a sub-organization. Each area defines a set of roles, e.g., A0 defines the roles R01 and R02, whereas A1 defines the role R1. Thus, when the nomadic user goes into the area A0 for the first time, the role R01 is attributed to him, e.g., at the institution entrance, the role “*Visitor ComputerScience*” is attributed to a computer scientist, whose goal is to give a lecture at the Computer Science Department. When he comes into the area A1, the role R1 is assigned to him, so that he can perform certain activities within the area A1. Finally, when the nomadic user has finished his task, he comes back from the area A1 to the area A0, which assigns the role R02 to him, i.e., he can play a role different from the one he previously obtained when he went into the area A0 for the first time.

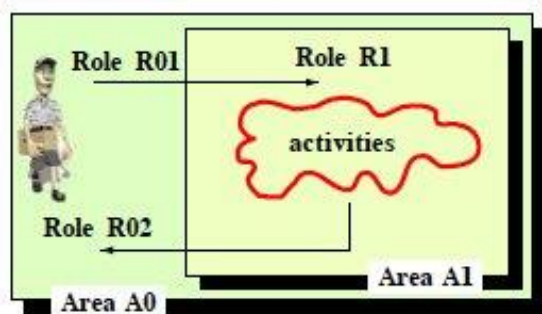


Figure 2 Role Attribution in Self-Managed Areas

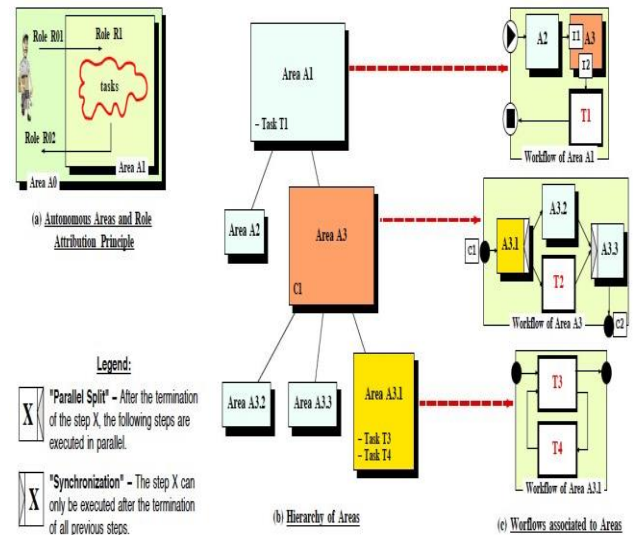


Figure 3 Principle of Hierarchical Self-Managed Areas

Hierarchy of Self-Managed Areas

As briefly introduced, large organizations are usually divided into several departments or units that are structured, following a well defined hierarchical relation. Our proposal consists in defining a hierarchy of areas that follows the organization structure: each organizational unit may be associated with a single self-managed area or several ones. Thus, a department can be represented by several hierarchically interconnected areas, e.g., a Physics Department can represent the main area, which includes the Nuclear Research Laboratory sub-area. This last one will define specific usage and administrative policies (e.g., entrance and exit procedures for people).

On the other hand, large organizations always need a sophisticated workflow system [6] for the management of their resources and tasks. However, defining and managing a global workflow system constitutes a complex and almost unrealizable objective, due to the particularities of the administrative policies in each organizational unit.

Thus, among other considerations, it appears interesting to take benefits from the area-based structuring to efficiently define and administrate the organization workflow system. In this way, a large organization management workflow is logically distributed among the involved sub-areas. Englobing areas provide general workflows, whereas englobed ones specify management workflows that implement particular administrative policies and tasks.

Thus, we define a hierarchical organization in which each self-managed area is in charge of defining and administrating its own component of the global workflow system. Each workflow component defines the connection points, i.e., the entrance and the exit points of this sub-workflow system. For instance, the central part of Figure 3 presents the workflow components of the areas A1 (the root area), A3 and A.3.1. As shown by the hierarchy of areas (left part of Figure 3), these three areas are linked by an “englobing-englobed” relations. Following this relation, the specific workflow of a sub-area constitutes an independent part of a more general workflow associated with an englobing area. The connection points C1 and C2 of the area A3 are respectively linked to the insertion points I1 and I2 of the area A1.

Such an area-based workflow system allows us to efficiently define the global workflow in different independent area sub-workflows that are managed in an autonomous way. More precisely, Figure 3 shows a hierarchy of self-managed areas, where the area A1 includes two sub-areas, A2 and A3. In turn, the area A3 includes three sub-areas, A3.1, A3.2 and A3.3. Each area manages its own environment: contextual information, services, roles and a sub-workflow.

The workflow of the area A1 defines a sequential control flow that successively executes the workflows of the areas A2 and A3 and terminates executing the task T1 within the area A1. The sub-workflow of the area A3 involves executing the sub-workflow of the area A3.1 followed by the parallel execution of: 1) the sub-workflow of the area A3.2 and 2) the task T2 within the area A3. The workflow of the area A3 terminates defining a join point that launches the sub-workflow of the area A3.3. The example we have developed highlights the principle of distributed and independent management of an organization workflow that is based on the well-known technique of interconnecting nested black boxes.

Thus, the simplicity and efficiency obtained by structuring and managing the global organization workflow attests one more time of the interest to structure and administrate an organization, following a hierarchy of self-managed areas.

Design and Implementation of SEDINU

Following the concepts and principles previously stated, in sections 3.1 and 3.2, we respectively describe the SEDINU functional schema and distribution architecture. The SEDINU components have been designed to allow the nomadic user to interact with other users and with the services available in the self-managed area where he is located. In section 3.3, we introduce a face recognition system, whose goal is to locate nomadic users in closed spaces, even if they are separated from their mobile devices. This system has been designed to efficiently complement a Wi-Fi signal triangulation-based system that locates mobile devices.

Because of the inherent limitations of these two location systems, it appears really interesting to merge their functionality, in order to provide users with freshly updated information that allow them to collaborate in an efficient and consistent way.

Functional Schema

The SEDINU system has been designed to allow the nomadic user located in a self-managed area to interact with: 1) other users (e.g., to cooperatively produce information) and 2) the services available on such an area (e.g., to request information). Interactions depend on a context-aware workflow, which considers the user's location, role and goals. Managed by the SEDINU system, each area has been designed as an autonomously administrated entity that relies on the *RBAC-Soft* system to communicate and coordinate itself with other self-managed areas. The *RBAC-Soft* system uses the RBAC (Role-Based Access Control) model [16] to manage the roles and resources defined by each self-managed area.

Figure 4 shows the steps followed by the SEDINU system: 1) to provide the nomadic user with services related to his current interaction context, and 2) to allow him to collaborate with other users located in the same area. First, the *Location Detector* (cf. section 3.3) determines the user's specific location within the current building (see #1 in Figure 4). Then, the *Location Detector* transmits this information to the *RBAC-Soft* system (see #2 in Figure 4), which processes it, in order to identify the corresponding self-managed area. Depending on his current location, the *RBAC-Soft* system determines: 1) the role attributed to the nomadic user and 2) the services that are associated with this role (see #3 in Figure 4).

Based on this contextual information, the *RBAC-Soft* system creates a workflow (or searches the already defined one) to guide the nomadic user towards the achievement of his goals (see #4 in Figure 4).

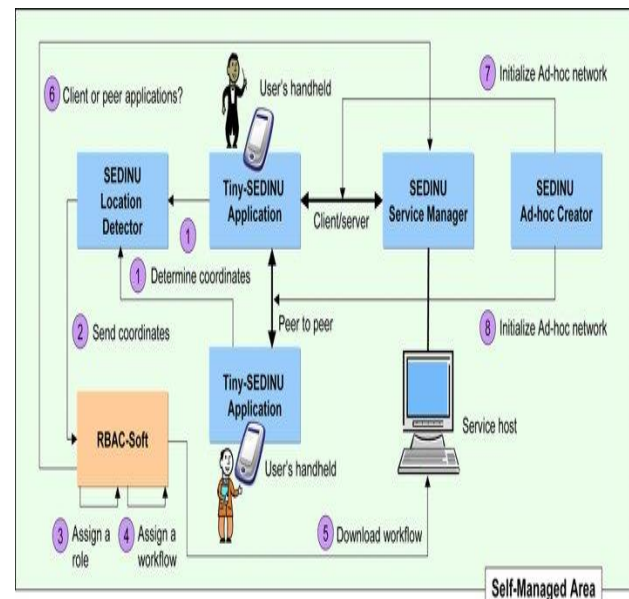


Figure 4 Functional Schema of the SEDINU System

Each activity of the workflow is associated to services that allow the nomadic user to perform his activity. Thus, the *RBAC-Soft* system uploads the workflow to the nomadic user's mobile device (see #5 in Figure 4) and asks the *Service Manager* to upload the needed applications to access the required services (see #6 in Figure 4).

As soon as the nomadic user has access to the services, he can select one of them from his mobile device. The set of available services is controlled by the corresponding workflow, for instance it can activate: 1) one service in the case of a single serialized task or 2) several services in the case of tasks whose execution order is irrelevant. Thus, the *Ad-hoc Network Creator* (see #7 in Figure 4) dynamically establishes an ad-hoc network between:

1) the *Tiny-SEDINU Application* (running on the user’s mobile device) and the *Service Manager*, or 2) two *Tiny-SEDINU Applications* running on the devices of the users who intend to collaborate (see #8 in Figure 4). This last kind of ad-hoc network facilitates direct information sharing (e.g., invoices or vouchers) and collaboration among users supported by a P2P (peer-to-peer) communication, instead of passing through a central server.

Distribution Architecture

The distribution architecture of the SEDINU services is centered on organizing the institution in autonomous areas that manage their own roles, services and workflows. That comes from the decision to make: 1) the area resources available and 2) the system reliable and easily manageable. Thus, when an area corresponds to a whole building or a part of a building, its services are concentrated on a host. On the contrary, when an area corresponds to several buildings (e.g., a department distributed on several cities) its services are replicated among the servers of the different buildings.

Distribution architectures mainly differ from one to another in three aspects: 1) the representation of the components on the participating sites (e.g., replica or proxy); 2) the number of instances of each component in the system, and 3) the possible mobility of components among sites.

The distribution architecture of the SEDINU system is integrated by three types of sites, which allocate different components. Sites of type A hold the *Tiny-SEDINU Application* (see

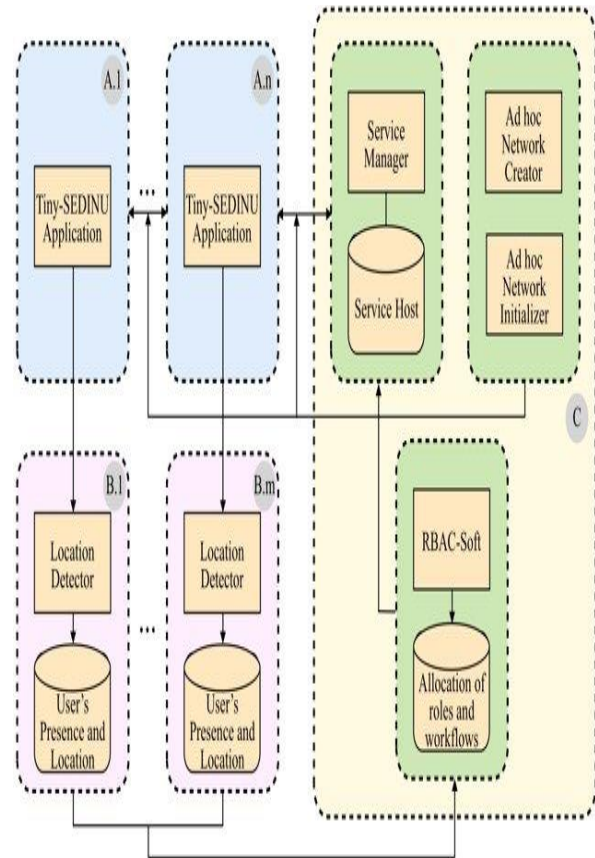


Figure 5 Distribution Architecture of the SEDINU System

Figure 5), which can play the client or peer role, since it allows a nomadic user to interact with services provided by a self-managed area or to interact with other nomadic users. At any moment, there can be multiple sites of type A, at least one per nomadic user.

On the other hand, there are two different types of sites that play the server role:

- Sites of type B store a *Location Detector* (e.g., the face recognition system) and information that it requires and produces (see Figure 5). It is possible to have multiple sites of this type, one per each set of Wi-Fi cameras located in the organization;

- Sites of type C allocate: 1) the *RBAC-Soft system*, which receives the nomadic user's location and relates it with a self-managed area; this system is also responsible for assigning roles and workflows to the user; 2) a *Service Manager*, which stores information about the services and workflows available at the self-managed area; and 3) the *Ad-hoc Network Creator*, which is in charge of initializing an ad-hoc network between two *Tiny-SEDINU Applications* or between a *Tiny-SEDINU Application* and a *Service Manager* (see Figure 5).

In an organization consisting of just one building, it can be placed a single site of type C, if the fault tolerance property is not a concern. However, if the SEDINU system is installed in an organization consisting of more than one building, it could be necessary to place a site of type C on each building. The replication of this type of sites is needed specially when buildings are physically distant from each other.

Face Recognition-based Location Detector

To provide a system with contextual capabilities such as bringing information closer to nomadic users and allowing them to easily find each other, it is mandatory to determine the location of the system users at anytime. People's nomadic characteristics expose the need of a mechanism capable of locating users in an "implicit" and "accurate" way; "implicit" because it is so tiring and annoying for a user to explicitly have to update his location every time he moves from one place to another inside a building, and "accurate" specially because, when providing support for indoors, it is important to detect whether a user is inside a specific office or in the adjacent one, in order to provide him with information or resources available at that specific place.

Many different ways have been proposed to locate people in closed spaces, but most of them involve a device that needs to be carried or worn out by the person wanted to be located. This constitutes a quite complex and heavy solution because any person can forget his device in any place different from where he is; devices can also be accidentally or intentionally interchanged with other users, making wrong assumptions about the user's current location or causing errors in the delivery of information.

To avoid this problem when locating well-known nomadic users, such as members of staff, the solution proposed in this paper consists in capturing and processing images of their own face rather than trying to detect their wireless devices. This solution appears satisfactory, since the user's face cannot be transferred to another person or left apart in another room. More precisely, we design and implement a face recognition system, which is focused on locating well-known users, in order to make information closer to them and to allow them to easily locate each other within the ubiquitous cooperative environment. In this way, we do not negate the real interest in locating mobile devices by triangulation of Wi-Fi signals [5] and GPS [4], since these solutions are useful when external persons (e.g., deliverymen, guests or visitors) have to be located within the organization.

Phases of the Human Face Recognition

The proposed computer vision-based system for the recognition of human faces needs to perform a learning phase before the effective real-time face recognition (known as the testing phase). The learning phase is carried out only once, and the testing phase takes place every time a human face is captured by a camera. In order to develop a relatively robust face recognition system, we have combined several techniques as described below.

The learning phase uses an object detector algorithm based on the Haar-like features principle [7] [28], which is able to differentiate a human face from any other object. This algorithm is used to create a picture database of the nomadic users that would be identified. This database constitutes the testing set of images and stores several pictures of each user's face in different positions, e.g., full-face portrait or profile, and customized with accessories, e.g., glasses or hat (see #1 in Figure 6). Once the database is completed, it is analyzed using the Eigenface method [26] and the Principal Component Analysis [15]. The information resulting from this analysis is then used to create classification model (see #2 in Figure 6), which includes information to differentiate one person's face from another.

Once the learning phase is achieved, the real-time face recognition phase can take place. Every time a human face is detected, a picture is captured (see #3 in Figure 6) and then analyzed according to the information produced by the Eigenface method (see #4 in Figure 6) during the learning phase. This information is assessed by the classification model (see #2 in Figure 6), which identifies the nomadic user (see #5 in Figure 6) by establishing a correspondence between the registered persons and the recently captured picture.

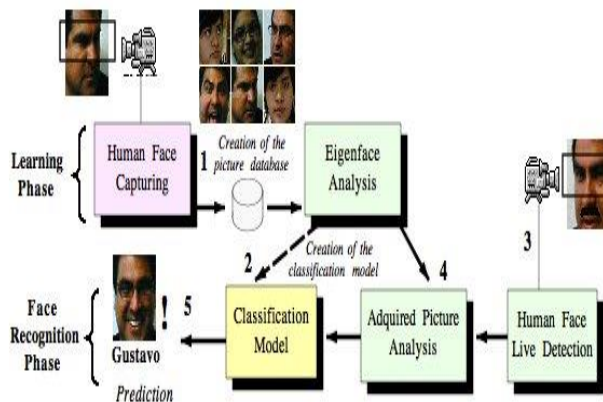


Figure 6 Learning and Testing Phases of the Face Recognition System

Implementation and Tests

Both the learning and the testing phases of the proposed face recognition system use the OpenCV (Open source Computer Vision) library, which aims at facilitating the implementation of computer vision applications. In particular, this library provides an object detector based on the already mentioned principle of Haar-like features [28], which consists in codifying light contrasts in some areas of the image, in order to obtain particular relationships among similar objects (human faces in this case). To perform this process, the object detector uses a classifier trained with some hundreds of positive and negative samples. Positive samples are images of the object that will be detected (i.e., a human face), and negative samples are arbitrary images.

The OpenCV object detector offers the following functions: 1) recognition of human faces in different sizes and 2) simultaneous detection of multiple human faces. The face size depends on the distance between the person and the camera. When a human face is detected, the camera takes a picture, from which the human face is extracted and resized to 245x245 pixels. The latter function is particularly useful in the real-time recognition phase, since if more than one person is in front of a camera, the OpenCV object detector is capable of simultaneously detecting each human face.

The Eigenface analysis is considered as an appearance-based method, whose goal is to create low-dimensional representations, called "eigenfaces", from the set of face images corresponding to the persons that will be identified. Such representations do not necessarily correspond to eyes, nose, mouth or ears.

Rather, they capture points (light contrasts) that produce significant characteristics among human faces that help to differentiate one registered person from the rest. Based on the calculated “eigenfaces” and the testing set of images, the most representative weights are obtained. These weights serve as input data to the LIBSVM (Support Vector Machine) library [8], which creates the mentioned classification model for determining whether the captured picture belongs to one of the registered persons. The hardware needed for the face recognition system to work is minimum: inexpensive webcams are installed at specific places (e.g., entrances to some offices, common places, hallways and corridors) within each building. Each camera is driven by a client hosted in a PC, which is responsible for capturing nomadic users’ pictures. Clients transfer these pictures to a multi-thread server, which is in charge of processing them to obtain a prediction about each nomadic user’s identity.

The current implementation of the face recognition system is capable of supporting a group integrated by up to 25 members. To illustrate how the human face recognizer works, we trained it to identify 25 collaborators. Thus, a database containing 25 pictures of each person was constructed. As mentioned above, these pictures were taken asking collaborators to move their face in different angles, varying their gestures and using different accessories each time.

Validation of SEDINU by a Use Scenario: Management of a Conference Lecturer

To validate our proposal and provide better understanding of how the SEDINU system allows us to organize and support nomadic interaction within a hierarchically structured organization, we describe a use scenario: the management of a conference lecture.

The prototype of the SEDINU system has been deployed across our institution. Thus, we have divided the test bed organization into five nested areas (see Figure 7), which are organized as follows: the first area (A1) represents the whole institution (root of the area hierarchy). Three sub-areas are included: the Entrance (area A2), the Computer Science Department (area A3) and the Restaurant (area A4). In turn, the Computer Science Department includes the Auditorium (area A5).

Bluetooth ad-hoc networks are established, in order to support proximity communications, e.g., transfers from a user’s device to an e-conference system or another user’s device. The institution area hierarchy is fully covered by a wireless network (Wi-Fi) that provides communication supports among different areas. In this environment, all the users employ mobile Wi-Fi devices (smartphone or tablet) to easily interact and exchange information.

To illustrate the functionalities of the SEDINU system, let us suppose that a computer scientist arrives at the institution (area A1) with the aim of giving a lecture at the Computer Science Department (area A3). First, the lecturer is registered at the Entrance (area A2) where he receives a mobile device running the *RBAC-Soft system*. This one assigns the role “*Visitor_ComputerScience*” to him and uploads a general workflow to his device, in order to guide him and support his activities within the institution (see #1 in Figure 7). Through an organization map and GPS, the SEDINU system shows him the way to reach the Computer Science Department (see #2 in Figure 7).

When the lecturer arrives at the Computer Science Department (area A3), our system determines the user’s current location.

Since the lecturer is a frequent visitor, his location is stimulated using two techniques that are complementary and can be fruitfully combined. The former refers to the face recognition system (c.f. section 3.3), which identifies and locates known people by using cameras installed at specific points (external public spaces, meeting rooms, corridors, secretary office). In this case, the goal is clearly to follow each user rather than to locate his mobile device, from which he may be regularly separated for a while. The latter technique consists in determining the user's device coordinates using Wi-Fi triangulation techniques [5].

The RBAC-Soft system relates the nomadic user's location to the self-managed area A3 and consequently it attributes the role "*Lecturer_ComputerScience*" to him. In a complementary way, the RBAC-Soft system uploads a specific workflow to the user's device, in order to guide his activities within the area A3. Relatively to the user's current environment (i.e., the Computer Science Department), the SEDINU system relies on a map and on the Wi-Fi technology to help him to localize the Auditorium (area A5) where his lecture has to take place (see #3 in Figure 7).

As soon as the lecturer arrives at the Auditorium, the SEDINU system determines his new location and sends it to the RBAC-Soft system, which associates his new location with the self-managed area A5 and then assigns him the role "*Lecturer_Auditorium*". The SEDINU system provides the lecturer with a dedicated "*Slide Projector*" service that allows him to present his slides (see #4 in Figure 7). Thus, a Bluetooth-based ad-hoc network is dynamically created between the lecturer's device and the projector service, in order to allow him: 1) to transfer his slides and 2) to control his presentation using the mobile device. During the lecturer's speech, all conference listeners (with the "*Listener*" role) can use the "*Slide Projector*" service and consult/annotate the slides. During the question session, all users (lecturer, chairman and listeners) can establish a fruitful question-answer collaborative session supported by the "*Slide Projector*" service and their personal devices.

When the lecture finished, the lecturer is invited to have a lunch at the institution restaurant. Thus, the chairman transfers an e-voucher from his device to the lecturer's device by a Bluetooth-based ad-hoc network (see #5 in Figure 7). Then, the SEDINU system provides the lecturer with relevant information to guide him to the restaurant (area A4), where he acts with the role "*Client*" (see #6 and #7 in Figure 7). There, the cashier verifies and charges the lecturer's e-voucher (see #8 in Figure 7). One hour later, the lecturer leaves the restaurant (area A4) and comes back to the entrance (area A2) with the role "*Exiting_Visitor*" always guided by the SEDINU system (see #9 in Figure 7). Finally, the lecturer gives back the mobile device (see #10 in Figure 7) and leaves the institution (area A1).

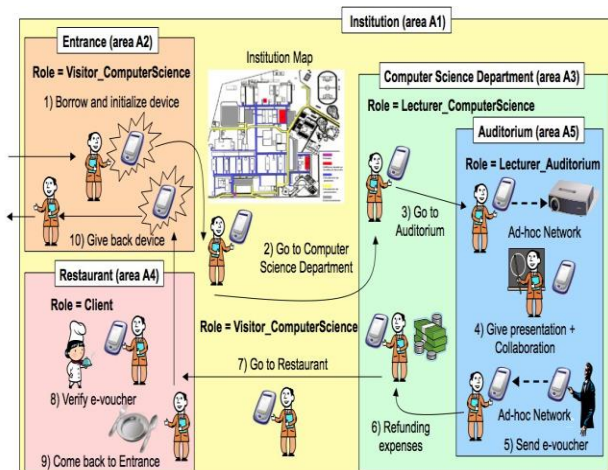


Figure 7 Scenario of the SEDINU System Use

Conclusion and Future Work

The main contribution of this paper is the definition of concepts and principles for the design and implementation of the SEDINU service discovery system. This proposal can be deployed across organizations, e.g., enterprises, institutions or governmental administrations, in order to facilitate: 1) the interaction between nomadic users and services provided by self-managed areas, and 2) collaboration among nomadic users under specific contexts. When a nomadic user enters to a building, the SEDINU system identifies and locates him or his mobile device, in order to determine the corresponding self-managed area where he is currently situated and to attribute him a role. Based on the user's current location and role, the SEDINU system determines the services available to him and then RBAC-Soft dynamically creates a workflow to guide his interaction or collaboration sessions. The applications required to interact with the available services or with other users are automatically downloaded to his mobile device. As soon as he selects a service, the SEDINU system can create an ad-hoc network: 1) between the user's device and the selected service host to facilitate user-service interaction or 2) among users' devices to support collaborative work.

Wi-Fi signal triangulation and GPS techniques allow us to respectively locate the user's mobile device in closed and open spaces within the organization. Since the user may be temporarily separated from his mobile device, we designed and implemented a face recognition system, which is able to identify the user's face within the organization buildings and to infer his location. However, the face recognition process includes a learning phase that needs the acquisition of several face pictures of the user.

Consequently, this face recognition system cannot be easily employed to manage unexpected users (e.g., visitors, guests or deliverymen) but better for well known and pre-declared ones (e.g., administrative staff, professors, or persons who attend a well planned event). Thus, the combination of these solutions (Wi-Fi triangulation, GPS and face recognition) is suited.

Improvements on the face recognition system are in development, including some modifications to reduce confusing predictions coming from changes in someone's appearance or variations in the environment lighting when a picture is taken. As this system has to be trained with a set of pictures from each user wanted to be identified, it learns to recognize the main features that differentiate a person from another. For example, if a person starts wearing a mustache, the face recognition system would be most likely to give a wrong prediction until a new learning phase with updated pictures would be performed. Consequently, it would be interesting to explore techniques that would allow this system to learn during the real-time recognition phase. In this way, it would not make a mistake when important changes in a user's appearance occur. In addition, drastic changes in the environment lighting can cause wrong predictions. For this reason, cameras have to be placed away from windows or other light changing sources.

The current version of the SEDINU system creates ad-hoc networks between mobile devices using the Bluetooth protocol, which offers several advantages: 1) it is a low-cost infrastructure; 2) users do not need a deep knowledge; 3) no network administrator is required; and 4) mobile devices have a Bluetooth connection. Therefore, the SEDINU system becomes compatible with several mobile devices.

As future extensions of this system, we will develop modules to create ad-hoc networks through ZigBee, which can be used by applications that do not require high data transmission. Moreover, the battery consumption of a ZigBee-enabled device is much lower than the one of a Bluetooth or Wi-Fi-enabled device. Thus, the SEDINU system can create ad-hoc networks able to control accesses to some restricted self-managed areas (e.g., a radioactivity laboratory), so only authorized users will be authorized to come into such areas. The SEDINU system constitutes the basis for the design of an elaborated and powerful platform able to support ubiquitous collaborative work.

References

- [1] Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., and Lilley, J. (1999). *The Design and Implementation of an Intentional Naming System*. In ACM SIGOPS Operating Systems Review, 33(5):186-201.
- [2] Arslan, P. (2016). *Mobile Technologies as a Support Tool for Health*. In Mobile Technologies as a Health Care Tool, SpringerBriefs in Applied Sciences and Technology Series, Springer International Publishing, pp. 11-41.
- [3] Ballmann, B. (2015). *Feeling Bluetooth on the Tooth*. In Understanding Network Hacks: Attack and Defense with Python, Springer Berlin Heidelberg, pp. 137-148.
- [4] Berman, K. J., Glisson, W. B., and Glisson, L. M. (2015). *Investigating the Impact of Global Positioning System Evidence*. In the 48th Hawaii International Conference on System Sciences (HICSS), IEEE Computer Society, Kauai, Hawaii, pp. 5234-5243.
- [5] Bhatti, N. (2014). *Service discovery protocols in Pervasive Computing: A review*. In the 17th International Multi-Topic Conference (INMIC), IEEE Computer Society, Karachi, Pakistan, pp. 205 - 210.
- [6] Boutamina, S., and Maamri, R. (2015). *A survey on context-aware workflow systems*. In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15), Djallel Eddine Boubiche, Faouzi Hidoussi, and Homero Toral Cruz (Eds.). ACM Press, Batna, Algeria, pp. 1-6.
- [7] Castrillón, M., Déniz, O., Hernández, D., and Lorenzo, J. (2011). *A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework*. In Machine Vision and Applications, Springer-Verlag, 22(3):481-494.
- [8] Chang, C.C. and Lin, C. J. (2011). *LIBSVM: A library for support vector machines*. In ACM Transactions on Intelligent Systems and Technology (TIST) Journal, 2(3):1-27.
- [9] Fitzgerald, E., and Adams, A. (2015). *Revolutionary and evolutionary technology design processes in location-based interactions*. International Journal of Mobile Human Computer Interaction, 7(1):59-78.
- [10] Grandhi, S. A. and Lanagan-Leitzel, L. K. (2016). *To Reply or To Reply All: Understanding Replying Behavior in Group Email Communication*. In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16). ACM, San Francisco, CA, USA, pp. 560-569.

CROWDSOURCING

December 2015 Vol.2 No.3 161-179

- [11] Gutierrez, F. J., Ochoa, S. F., and Vassileva, J. (2016). *Identifying Opportunities to Support Family Caregiving in Chile*. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16). ACM, Santa Clara, California, USA, pp. 2112-2118.
- [12] Herborn, S., Lopez, Y., and Seneviratne, A. (2005). *A distributed scheme for autonomous service composition*. In Proceedings of the First ACM International Workshop on Multimedia Service Composition, Singapore, pp. 21-30.
- [13] Klauck, R. and Kirsche, M. (2012). *Bonjour Contiki: A Case Study of a DNS-Based Discovery Service for the Internet of Things*. In the 11th International Conference ADHOC-NOW, Ad-hoc, Mobile, and Wireless Networks, LNCS Vol. 7363, Springer Berlin Heidelberg, Belgrade, Serbia, pp. 316-329.
- [14] Lagares-Lemos, A., Daniel, F., and Benatallah, B. (2015). *Web Service Composition: A Survey of Techniques and Tools*. In ACM Computing Surveys, 48(3):1-41.
- [15] Martínez Delgado, J. S., Mendoza, S., and García, K. (2013). *Flexible Bimodal Recognition of Collaborators in Pervasive Environments*. In Special Session Proceedings of the 12th Mexican International Conference on Artificial Intelligence (MICAI), IEEE Computer Society, México, D.F., Mexico, pp. 151-156.
- [16] Mitra, B., Sural, S. Vaidya, J., and Atluri, V. (2016). *A Survey of Role Mining*. In ACM Computing Surveys, 48(4):1-37.
- [17] Nidd, M. (2001). *Service discovery in DEAPspace*. IEEE Personal Communications, 8(4):39-45.
- [18] Oaks, S., and Wong, H. (2000). *Jini in a nutshell: a desktop quick reference*. O'Reilly & Associates, Inc.
- [19] Olbertz, J. H. (2015). *Service Availability and Discovery Responsiveness: A User-Perceived View on Service Dependability*, PhD Thesis, Humboldt Universität zu Berlin, Germany.
- [20] Olson, A. M., Raje, R. R., Devaraju, B., and Gallego, L. S. (2015). *Learning improves service discovery*. In Concurrency and Computation: Practice & Experience, John Wiley and Sons Ltd., 27(7): 1679-1694.
- [21] Raychoudhury, V., Cao, J., Kumar, M., and Zhang, D. (2013). *Middleware for pervasive computing: A survey*. In Pervasive and Mobile Computing, Special Section: Mobile Interactions with the Real World, Elsevier, 9(2):177-200.
- [22] Rodríguez-Amat, J. R., and Brantner, C. (2016). *Space and place matters: A tool for the analysis of geolocated and mapped protests*. In New Media & Society, 18(6):1027-1046.
- [23] Rossitto, C., Bogdan, C., and Severinson-Eklundh, K. (2014). *Understanding constellations of technologies in use in a collaborative nomadic setting*. In Computer Supported Cooperative Work (CSCW), Springer, 23(2):137-161.
- [24] Shi-Cai, Y., Wu, Y. Z., and Guo, R. N. (2013). *A UPnP-Based Decentralized Service Discovery Improved Algorithm*. In the Fifth International Conference on Computational and Information Sciences (ICIS), IEEE Computer Society, Shiyang, China, pp. 1413-1416.

CROWDSOURCING

[25] TaeKwon, M., Zafar, B., JaeTek R., KiHyung, K., and SeungWha, Y. (2010). *The efficient DA placement method in SLP*. In Proceedings of the 12th International Conference on Advanced Communication Technology. IEEE Computer Society, Gangwon-Do, South Korea, pp. 364-368.

[26] Turk, M., and Pentland, A. (1991). *Eigenfaces for recognition*. In Journal of cognitive Neuroscience, 3(1):71-86.

[27] Vara, M. I. and Campo, C. (2015). *Cross-Layer Service Discovery Mechanism for OLSRv2 Mobile Ad-hoc Networks*. In Sensors, 15(7):17621-17648.

[28] Viola, P., and Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features*. In Proceedings of the 2001 Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Kauai, Hawaii, pp. 511-518.

[29] Zhou, J., Liu, L., Deng, Y., and Huang, S. (2014). *A QoS Routing Protocol with Bandwidth Allocation in Multichannel Ad Hoc Networks*. In Wireless Personal Communications, Springer, 75(1):273–291.