WP6: "Applications"

D6.1 – Part A: Description of Application Scenarios and of the Services to be Provided

Status and Version:	Final	
Date of issue:	1.12.2006	
Distribution:	Public	
Author(s):	Name	Partner
	Nicola Bicocchi	UNIMORE
	Franco Zambonelli	UNIMORE
Checked by:		

Table of Contents

1 Introduction

1.1 Purpose and Scope

The goal of this document is to report on the work performed during the first year of CASCADAS related to the analysis of various application scenarios and their requirements, which eventually led to the selection of representative application scenarios. Requirements analysis had the twofold aim of *(i)* properly directing the activities of the various WP and of ensuring that all models and tools studied in CASCADAS – as well as the overall service framework which will be built from them – are grounded on the real needs of future autonomic communication scenarios and services, and (ii) selecting a set of representative application scenarios (i.e., expressing diverse requirements) as a basis for later demonstration activities.

1.2 Document History

Version	Date	Authors	Comment
0.01	15/11/2006	Franco Zambonelli, Nicola Bicocchi	Initial draft with proposed inclusion of M4 report on application requirements and definition of ToC.
0.2	22/11/2006	Franco Zambonelli	ToC finalized. Revision of Sections 1 to 5 and 7. Inclusion of Subsection 6.1. Still missing partner contributions for 6.2, 6.3, and 6.4
0.3	1/12/2006	Franco Zambonelli	Addition of pieces for Sections 6.2 and 6.3

1.3 Document overview

The document is structured as follows. Section 2 introduces and basic methodology that we have adopted to collect and organize requirements. Section 3 summarizes the various application scenarios identified and organize them into three main classes. Section 4 presents discusses the types of ACE-based services that one may expect to made available in the various classes of scenarios via the autonomic service framework. Section 5 details the list of requirements extracted from the scenarios and form the associated services, and present them organized on a "per WP" basis, i.e., by associating requirements to the WP which is more likely responsible for its accomplishment. Section 6 details the rationale under the selection of the CASCADAS application scenarios. Section 7 summarizes and sketches some lessons learned from the requirements collection work.

2 The Methodology

To guarantee ACE's functionalities in a large spectrum of scenarios, to properly guide the activities of the technical WPs, and to eventually release a sound and usable framework, it is necessary that all research and development activities are driven by meaningful requirements. To identify these requirements and to make sure that these are representative we have adopted the following methodology (Figure 1):

1. **Collection of application scenarios**. We asked all CASCADAS partners to send us some description of envisioned application scenarios.
2. **Scenarios analysis and synthesis of a possible taxonomy**. After all scenarios were submitted, we analyzed them and tried to organize them into a rationale, to identify commonalities and differences.
3. **Identification of services**. Considering that CASCADAS is aimed to produce a new infrastructure to build new generation communication services, we mines the above scenario to identify the specific communication services that are to be provided within each of them, and also to identify services common to different scenarios.
4. **Extraction of per scenario, per service requirements**. From the proposals we extracted basic requirements for each service.
5. **Synthesis of meaningful requirements organized in WP**. At the end, we organized all the meaningful requirements into WPs. For each requirement and each WP we highlighted the most relevant scenario.

In general, we expected (and indeed found) that diverse application scenarios share a ground of common characteristics, and can be grouped into a smaller set of “application areas”. Moreover, we expected (and found) that common services are to be provided in different area. At the same time, the same kind of services can express different requirements depending on the application area in which it is expected to apply. This fact is roughly represented in Figure 1.

As the last step, after the analysis and synthesis of requirement, we exploited such analysis towards the:

6. **Selection of representative application scenarios**. Such a selection has led to the identification of three application scenarios that will ground future research and demonstration activities.

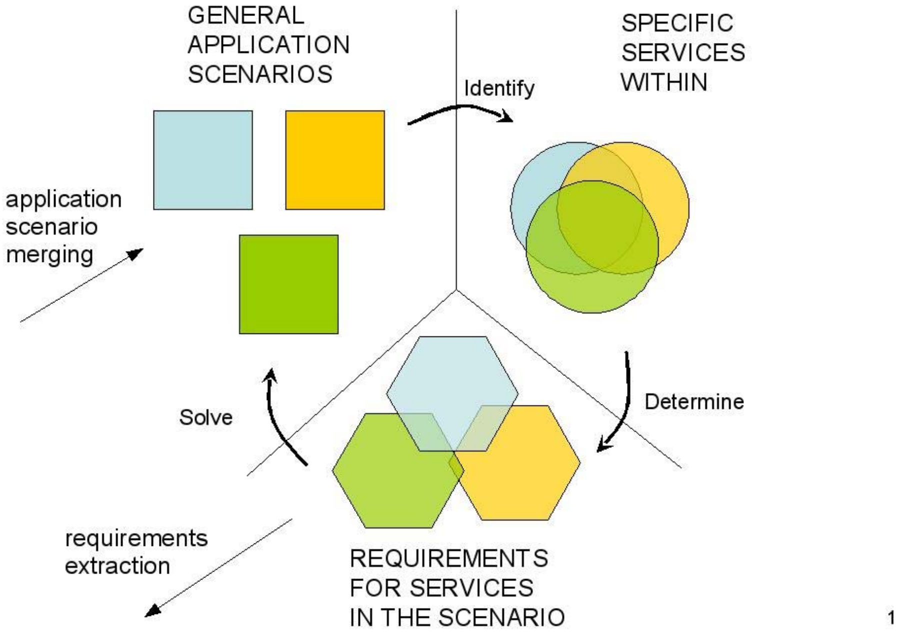

Figure 1. The methodology adopted for requirements extraction

3 Application Scenarios

In this section, we summarize the application scenarios collected from WP6 partners, and tries to organize them in a rational way.

The basic assumption in the description of the scenarios is that the exhibited functionalities are all expected to be realized in terms of ACEs and ACE-based services executed within the CASCADAS autonomic service framework.

3.1 Short Description of Collected Scenarios

We have collected some application scenarios from all participants. In this subsection, we all the received proposals.

Services for Pervasive Museums (UNIMORE). This proposal focuses on the problem of supporting the activities and the movements of tourists in a large museum. These activities include orienteering in the museum, retrieving information about pieces of art and history, as well as coordinating the visit tour in group of users, all of which to be supported by specific services. This may include the need of meeting at a particular place, distribute themselves according to specific spatial patterns, or simply move in the unknown environment without interfering with each other. This level of coordination requires dynamically acquiring some sort of context awareness on the part of the tourist: one can coordinate with other tourist only if it’s somehow aware of its context. Given these considerations, this scenario focuses on the problem of dynamically providing tourists with simple and effective contextual information to support their coordinated activities in an adaptive and autonomic way. The proposal assumes that tourists have a software agent (in the form of a user-level ACE) running on a wireless hand held device, such as a palm computer or cellular phone. The ACE collect information about what the user is doing and then gives the tourist suggestions on how and where to move, by relying on the services provided by embedded distributed ACEs. Indeed, the scenario also assumes that the museum has an adequate computer network and is densely enriched with sensor and actuator devices. Embedded in the museum walls is a network of computer hosts, each capable of communicating with each other and with the mobile devices located in its proximity via the use of a short-range wireless link. The scenario also assumes that the devices have a localization mechanism to find out where they are actually located in the museum. This localization mechanism could be implemented by some kind of GPS-like device or with less expensive local hardware that relies on the properties of triangulating radio or acoustic signals. This scenario and the associated coordination problems are of a very general nature, but the case study has many parallels to other scenarios, such as urban traffic management, forklifts activity in a warehouse (where equipped vehicles suggest to their drivers where to go), or software agents exploring the Web (where mobile software agents coordinate distributed researches by moving on various Web sites). The considerations based on this case study therefore have general applicability.

Smart Environments Supporting Independent Living (UU). This scenario describes some ideas to support the growing group of elders who still remain within their own homes and are not fully independent. This scenario introduces novel techniques for person-centric services in pervasive spaces. It proposes, how from a technical perspective, such services could be realized based on the emerging concepts of a distributed network of knowledge, facilitating dynamically composable and flexible service provision that engenders service continuity. This proposal assumes that individuals have a software agent running on a smart phone or a PDA. These devices (i.e., proper ACEs within) should be able to determine the user localization (using GPS or less expensive local hardware) and to interact with a wireless network that should be provided by the ambient. One of the challenges for future smart environmental infrastructures is the need for them to reason about their situation and to understand their own behavior. To do this they are required (both at the level of individual components and as a whole) to be introspective and reflective, and to feed back the results of these processes to be used to improve performance. While this provides the knowledge with which they can, eventually, manage and configure themselves it does also make them more self-aware or in short it makes them smarter. However, in order to get ‘smarter’, the environment, its entities and services need some form of properly represented, well correlated and widely accessible repository that leads to the concept of a knowledge network that is focus of a specific WP in in the CASCADAS project.

Cinema Grid Computing (BUTE). When we have a dense network of devices co-located in some physical space, one can think at exploiting them as a grid. For instance, the audience in a cinema is usually well-equipped with intelligent communication devices (Java-enabled mobile phones, PDAs etc.). Let us assume these devices run proper ACEs. Then, while the film runs and all devices are assumed to be idle, they can connect together to form a computing grid. A master assigns tasks and the grid solves it. At the end of the movie the results are collected and the grid is shut down.

Routing and Information System for Vehicles or People (ICL). This scenario consideres a distributed, autonomic system for vehicles routing aimed to minimize travel time using the knowledge to the current status of different areas leading to the destination. This assumes the presence of proper wireless enabled computing devices in cars and across streets, hosting on needed the execution of ACEs. Consider a situation where a person or vehicle, surrounded by other objects (fixed or moving), needs to move as fast as possible to a certain destination or requires obtaining information of the surroundings as it moves. A solution to this problem can be formulated with ACEs. First, ACEs need to be located on the area where mobiles move. The principal function of these ACEs is to monitor their surroundings and to provide information of other ACEs. Second, mobiles carry an ACE, which will interact with other ACEs (in other mobiles or in fixed locations) to provide the user’s application with the information required (e.g. calculate the lowest delay path). ACEs in mobiles can also provide monitoring, so that areas with no other ACEs can be included (temporarily) in the system. A direct application of the scenario described would create an autonomic environment to alleviate the roads congestion problem. The application would consist of ACEs in cars (for computing, sensing and visualization) and ACEs located at street intersections (for sensing). Sensors measure the congestion level of the streets, forwarding the information to cars nearby. Cars form a wireless ad hoc network that is used to distribute the information across the network. With the information available, computing devices in cars (e.g. PDAs) can map the status of streets in the city, so that faster routes can be calculated by avoiding congested areas. In addition, information about accidents and their location can be included, as well as weather conditions, which can be useful for long trips.

Road Fault Detector (BUTE). Strictly related to the above, consider a system to propagate information about road faults using local interactions and a central database. When driving a car, people often detect road faults: pits, ice/oil on the road, and other unexpected obstacles. In some cases, human interaction is not even needed to observe the faults: it can be derived from the measurement values coming from the wheels and suspensions (i.e., from simple ACE components associated to them), or they can be detected by monitoring the images coming from a simple video camera attached to the car. This road fault information can be very useful for the drivers who are coming behind. The goal of the scenario is to propagate the information observed. A car (driver) detects a road fault (pit, ice, oil on the road) and forwards it to the neighboring vehicles in order to warn them. Some components of the system even forward the relevant information to the road maintenance authority, too. This “urban mobility” scenario could exploit well the autonomic and self-organizing properties of CASCADAS's ACEs.

Tourist Information System and Friend Search (BUTE). An application in the area of tourist information systems can consider that people, using a handheld device, can connect to a tourist guide system of the city and receive information. This may include information about nearby objects and how to reach them, ‘what-to-visit- next’ suggestions, hints about public transport etc. The location of the tourist’s device can be determined either from network information (if the device is connected to a wireless network or a cellular network) or more accurately from its geographical coordinates (if the device is equipped with a GPS receiver). Since similar services like POI (Points of Interest) exist for publicly available GPS receivers, the novelty of this system is to provide not just the location but up-to-date information about the objects and transportation facilities (ongoing repair, actual events etc.). Using similar techniques, one could also use the system search friend in overcrowded places: mobile devices owned by each person build a multi hop ad-hoc network to be used to inject signals and gradients useful to find persons and things.

Behavioral Pervasive Advertisement (TI). Similar to the above in terms of goals, the basic assumption in this scenario is to rely on “behavioral targeting” to allow marketers to better grasp customer's needs and interest, mainly by tracking and monitoring users on the web. Behavioral Pervasive Advertisement (BPA) applies pervasive computing techniques and technologies to the nowadays emerging advertisement technique called behavioral advertisement (or behavioral targeting) which tries to ensure that advertisers reach the target audience in a more effective way. TI proposes to extend such technique to any communication context supported by Telecommunication operator services where user interests and needs can be grasped (e.g. tracking the preferred jeans shops a user goes into by means of GSM-based localization services). Moreover, exploiting the pervasive nature of CASCADAS based applications, BPA may provide customized contents and advertisement, not only during web navigation, but different channels may be personalized to the single user or to groups of users (e.g. digital screens in the road may autonomously provide advertisement customized to the profiles of the user moving in the zone of the screen). Behavioral Pervasive Advertisement could be a good challenge for CASCADAS due to the key role of communication and of proper ACEs to achieve its main goals.

Ubiquitous Grid Computing for Pervasive Services (BT). This scenario envisions an autonomic service deployment scenario featuring spontaneous service composition, load balancing and adaptation to changing demands patterns. The overall vision sits at the interface between Grid (or utility) computing and Service-Oriented Architecture (SOA). It is based on the hypothesis that the role of individual elements participating in a pervasive application is not entirely determined a priori by static properties, but also by their history and context, giving rise to plasticity. Our scenario is designed to emphasize the self-organizing abilities of a population of ACEs and is loosely based on a biological analogy. Each ACE, whether “atomic” or “complex” is regarded as an “agent” whose job is to manage a particular service, i.e. to achieve and maintain a target QoS by finding and consuming resources of the right type and in the right quantity to meet the demand for that service. The abilities of an individual ACE are constrained by its execution environment and access privileges, i.e. by the characteristics of the physical device on which it is running (desktop, server, PDA, database…) and its level of control on the device’s resources (bandwidth, CPU, memory, storage…). Whenever an ACE finds that the QoS for the service that it is managing is threatened (e.g. because of an increase in demand), it first attempts to resolve the situation by “growing”, i.e. by increasing its share of the local resources. If growth is not possible (because it is already using all of the local resources that it has access to), it attempts to “swarm”, i.e. to create a remote copy of itself on another physical device, with which it will subsequently be able to “share the burden” (either by explicitly notifying some service consumers of the existence of the new access point, or by transparently forwarding some service requests to its “twin”). The area in which this scenario is more focused is mainly dynamic resource management.

Self-organized Information Management (Fokus). The problem addressed in this scenario is that to enforce a semantically-oriented information retrieval approach that goes beyond the currently available keyword-based approach. Consider an information management service that deals with various types of information sources: documents, images, video data, audio material, life feeds, sensor data and measurements, and combinations of these types. Each of these sources is augmented with meta data: keywords, citations, staff and artist information, etc. The proposed solution is that information units are considered as active objects (i.e., ACE encapsulated entities in the CASCADAS context) that pro actively try to cluster into semantically similar groups which can be accessed by means of a single retrieval request. To realize self-organization, major features of CASCADAS project are to be exploited, and two mechanisms are proposed. *Movement:* Information units are driven by objective to be accessed as much as possible. Therefore, they tend to move itself into the neighborhood of “prominent” information units.

This can be realized by the exchange of tags, tag advertisement (tag blogging) and dynamic negotiations. *Topological Wrap:* Consider a query accessing two information units a and b establishing a semantic similarity between these documents and decrease of the distance between these documents (in opposite of an active movement). The area in which this scenario is more focused in semantic knowledge management. Within the scenario, one can imagine providing servicing for querying and for disseminating information.

Autonomic Content Distribution Service (NKUA). To some extent similar to the above, but stressing more on autonomic properties, this scenario envisions a P2P content networking application for the exchange of digitized content which operates in a totally uncontrolled environment and, thus, has to integrate an adaptive and autonomic behavior. For a comparison, consider a P2P node (e.g., KaZaA) and a CDN node (e.g., as it can be realized by an ACE). The P2P client: has no fixed infrastructure upon which it can rely, has to interact with previously unknown, potentially untruthful peers, has a very partial/inaccurate picture of its operating environment (basically only a handful of first step neighbors which it obtains from a bootstrap server). On the other hand, a CDN node: runs on meticulously maintained servers which communicate over highly provisioned (sometimes even dedicated) overlay links, can trust all the information it receives from other CDN nodes, and has a pretty accurate view of its operating environment, since it receives a continuous feedback stream from the management/reporting system of the CDN. Such an application can benefit from the autonomic aspects developed in the CASCADAS project. In particular, it can be able to support highly dynamics of the CDN networks, discourage free-riding and adversarial behaviors, dynamic optimization of resource usage, minimization of communication overhead.

Distributed Auctions (ICL). The scenario models future economies, which are expected to be organized around networked and completely automated transactions between enterprises, and between individuals and enterprises. Such systems are expected to carry a high number of short-lived electronic transactions operating at a high frequency and performed by ACEs participating in auctions with roles of bidder, buyers, and sellers. To succeed, auction participants will need to operate in an opportunistic way. The support of an autonomic communication network will ensure the delivery of uninterrupted economic services with a defined quality-of-service (QoS) to a large numbers of automated or live users. Such communications would require to be situated-aware and to self-adapt to network conditions to deliver the fastest response time and the best protection against failures. Auction services would require to be self-protecting from malicious users. Likewise, self-configuration would allow auctions to virtually move in the network to "place" participants in the most advantageous places and allow them maximise their reveneus. Operating at a global scale, the managing complexity of this application would scale up without control unless autonomic support, as it can be provided by CASCADAS tools, emerges.

3.2 Analysis of Scenarios and Synthesis

We have analyzed all the above scenarios and have tried to identify recurring characteristics in terms of background assumptions, services provided, and requirements, so as to define a sort of scenario taxonomy.

After the analysis of the proposed application scenarios we have clustered them into three main groups. These groups are mainly characterized by the spatial dimensions of the application domain but also from the kind of services that are required in each of them.

We propose that the scenarios can be classified along in three groups:

- ***Smart Homes (SH)*** refers to strictly pervasive applications, assuming some sort of computer-based instrumentation of the environment, and involving local-area people-to-people and people-to-environment interactions.
- ***Wide Area Network (WA)*** refers to wide area network computing. In this group we place large scale distributed applications, P2P networks, grid computing, content distribution systems, distributed auctions. All of them are more focused on the dynamic building of overlay services with autonomic behavior and fault tolerance than pervasiveness and local interactions.
- ***Urban Mobility (UM)*** refers to urban-scale applications. In this application scenario are present typical features of both the other two groups. We can find ad-hoc wireless sensors and instrumented environments, to support people-to-people (better: device to device) interactions and people-to-environment interactions, as well as medium-to-large scale P2P networks built over networks of ad-hoc devices to distributed and access information.

The differences, which directly reflect on the kind of services to be provided, are mostly related to the spatial scale of the applications. In SH scenario the most important aspects relate to local-range interactions between devices, the need for location-dependency, environment-awareness, and social awareness, and the need for cross layer networking (due to the need to interact with information about both embedded devices and the social level). Moving towards UM and WA, more importance is given to life cycle management of complex dynamic overlays, the dynamic management of their resources, semantic service interoperability, and access to contextual information related to network and computing resources. Security and the need for autonomic behavior are horizontal, playing a major role in all scenarios.

Given this taxonomy with can organize the aforementioned scenario proposals as follow:

- ***Smart Homes (SH)***
  - Services for Pervasive Museums (UNIMORE)
  - Smart Environments Supporting Independent Living (UU)
  - Cinema Grid Computing (BUTE)
- ***Urban Mobility (UM)***
  - Roaming and Information System for Vehicles or People (ICL)
  - Road Fault Detector
  - Tourist System Information and Friend Search (BUTE)
  - Behavioral Pervasive Advertisement (TI)
- ***Wide Area Network (WA)***
  - Ubiquitous Grid Computing for Pervasive Services (BT)
  - Self-organized Information Management (FOKUS)
  - Autonomic Content Distribution Service (NKUA)
  - Distributed Auctions (ICL)

4 Identification of Services

Considering that CASCADAS is aimed to produce a new conceptual and practical infrastructure to build services, we extracted from the general scenarios the specific services that are expected to be exhibited by them.

On the one hand, we identified a set of common general-purpose class of services, which are useful and expected to be present in most of the scenarios. Such sorts of "basic services" form the substrate over which more application-specific (or user-level) service can be build, and are listed in the next subsection. This list might be not complete but is representative enough to act as starting point for a comprehensive analysis of the requirements. Since each of the three classes of scenarios identified has both specific characteristics as well as common characteristics with the scenario classes that are "neighbors" in the proposed taxonomy (see Figure 2), it is not surprising that the same basic services are present in different classes of scenarios. However, at the same time, the same service in different scenarios may exhibit peculiar needs, and needs to be adapted depending on the scenario. From this, it clearly emerge the importance for the building block of these services (i.e., the ACEs) to be autonomic and adaptable over different contexts.

Beyond basic general-purpose services, the application scenarios lead to the identification of a large number "application dependent" services that can be somehow classified as "user level services". Considering the almost infinite number of these services that can be conceived, we extracted only some representative user level services examples from the proposed scenarios.

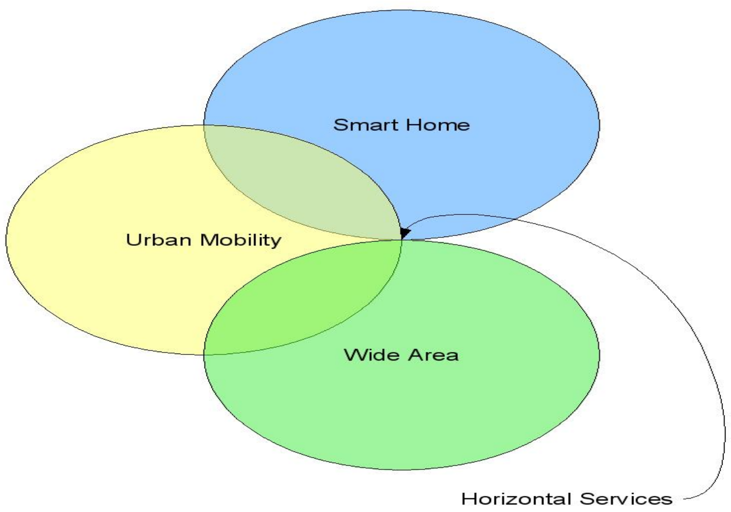

Figure 2. Services organized in main scenarios.

4.1 Basic Services

Let us now report about the classes/type of basic services identified. Each of these is identified by a name we assigned to it, as well as by the indication (in square brackets) of the scenario classes in which it is likely to be necessary.

[SH/UM] Ad-Hoc Routing Services. The main task of this service is guarantee that different sparse devices can form ad-hoc networks and route packets to a destination. Such a service can be use for local-scale networks, e.g., home networks, as well as in larger-scale geographically distributed networks (e.g., environmental sensor networks or ad-hoc networks). Depending on the characteristics of communication devices, it can be used both in single-hop mode as well as in multi-hop mode. Given the intrinsic dynamics of the scenario, the service should tolerate mobility of devices, the addition of new devices, the dismissing of some devices. Also, it should be able to adapt its behavior depending on the specific characteristics of the devices and on the patterns of use of the network. In other words, and more in general, it should exhibit adaptable and autonomic behavior.

Clearly, besides the general need of enforcing ad-hoc routing in an adaptable and autonomic way, the specific characteristics of the routing algorithms may require to vary in different scenarios, e.g., as far as addressing is concerned. In SH, it may be useful to route message to devices and people based on some identifiers for persons and objects ("send to washing machine"), also in considerations that not all devices will possibly have an IP. In UM scenarios it may be necessary to enforce geographical routing ("send message to location X"), or even a routing based on traditional network identifiers (e.g., the IP). Moreover, for several applications, different concepts of routing may be in need to be enforced, such as location-based multicast ("send message about a traffic jam to all cars in street X") or content-based multicast ("send message about a new show to all tourist in town interested in it"). In is also worth emphasizing that different types of routing services may require to be available at the same time in the same scenario, as different user level services may require different routing strategies.

[SH/UM] Localization Services. In general terms, localization services aim to produce and provide information about the position (logical or physical) of entities (users, objects, devices) in an environment. Such a service is of general importance to promote the enforcement of situation-aware services, being the position of things in space the primary basic information upon which to rely, in pervasive and urban scale applications, to identify the context in which situations occur. Indeed, all the scenarios that can be classified as SH or UM, implicitly assume the capability of somehow locating entities in space.

Clearly, depending on the specific scenarios, different concepts of positions may need to apply: location may refer to a room in a building (in the SH scenario I may need to know "where is person X"), a position in a room (in the SH scenario I may need to know "where are my glasses"), to a GPS position (in UM scenarios). Also, in general, any location service has be autonomic and adaptive and should guarantee that some results is always provided, i.e., that the best positioning system available is always used. For example, if GPS is not available, the localization service may autonomously decide to use an alternative, such as mobile phone cells trilateration. Analogously, inside a building, it could use some kind of RFID based localization if available, or rely on WiFi signal strength attenuation. To this end, the localization service must be context-aware in itself.

[SH/UM/WA] Context-aware Information Retrieval Service. This is a very wide spectrum service. In general, and in all presented scenarios, it is useful to have the availability of a service that enables retrieving information related to "something" (and object, an event, a person, some physical properties, a file, a computing or network resource) without knowing in advance how this something is named (if it is actually named at all), but simply knowing what are the expected characteristics of that something. In SH scenarios, this can be used to find persons of a specific class ("is there a security guard in here?"), or specific objects ("is there a toiled suited for disabled here?), or to detect unusual activities. In urban scenarios, this can be used to detect specific contingencies ("is there any road deviation abound"). In WA scenarios, this can be used to find resources or service suitable to perform a specific task ("is there any JPG compressor here?") or to find specific files, or to determining which auctions have been issued in which an ACE may be interested in participating.

Also in this case, the implementation of services of such kind should be very flexible, and should necessarily rely on a network of distributed ACEs capable of organizing and properly distributing knowledge and contextual information around. To tolerate the dynamics of the network, it should be able to work in any situations, possibly adapting the quality and accuracy of responses on the current situations. And it should also be able to accommodate the diverse needs and the diverse QoS requires by different classes of users (e.g., giving priority to requests coming from institutional actors than from normal users).

[UM/WA] Overlay Network Building and Maintenance Services. While routing and context-aware information retrieval have to rely on the definition of specific overlays of ACEs to realize their function, the general concept of overlay networks appears of a paramount importance to realize any kind of distributed applications in large dynamic networks. Indeed, whenever the network is large (as they can be both mobile ad-hoc networks of cars in a city and networks of world-wide-distributed peers) and dynamic (as in the presence of mobile nodes such as cars or ephemeral nodes such as peers), the only way to enable distributed coordination of activities is via the building of self-organizing overlay networks. Thus the basic general-purpose services aimed at bootstrapping, maintaining, and preserving in an autonomic way coherent overlay structures of ACEs over large dynamic networks are indeed necessary.

Such services should work in a transparent (traditional) way, but also in a sort of contextualized way, in which each of the devices involved in the formation of the overlay can user local information about the context (e.g., memory and storage capacity, computational power, network link speed, energy availability...) to maximize some utility function, optimize its position in the network, and properly self-adapt the structure of the local network upon changing condition or in reaction to faults and disconnections.

[SH/UM/WA] Aggregation Management Service. In many large networks, and for several specific application needs, it may be necessary to retrieve data distributed on or produced from different devices and produce aggregated values. Such a general service can be of use, in SH scenarios ("what is the overall number of persons in that wing of the museum"; "what is the average humidity in the Egyptian section?"), in UM scenarios ("what is the amount of cars in that section of the highway?"; what is the average load on the urban WiFI mesh?"), as well as in WA scenarios ("what is the number of connected peers?"; "what is the average computational load on the grid?"). And it can involve data related to the social aspects of the scenarios (people and their activities), data related to the ACEs activities themselves, data related to low-level network/device status, or a mix of both ("what is the average communication load per working person in the urban WiFi mesh?").

Clearly, for such aggregation services to work well, they must properly account for the system load and the network latency, as well as the specific characteristics of the devices in which the distributed algorithms run. Also, they should tune their degree of accuracy and freshness accordingly (to provide a best effort response in respect of specific constraints), and should of course be able to provide their service independently of any contingency or of any kind of network dynamics.

[UM/WA] Life Cycle Management Services (Grow, Swarm, and Shrink). Since many of the envisioned services for UM and WA scenarios (both the above ones and the user-level ones detailed in the following) cannot easily foresee the amount of resources needed to achieve their task, or the load conditions upon which they have to operate, or the specific structure and extent of the network in which they have to operate, the implementation of any distributed services must consider to handle available resources in an autonomic and situation-aware way, by replicating its components (i.e., the composing ACEs) on need, and have them start autonomously increase the quality of services by properly exploiting the enlarged resource availability as a collective (i.e., as a swarm). On the opposite, whenever such resources appear excessive, the distributed swarm of ACEs should properly shrink itself to accommodate the new need.

Clearly, such basic mechanisms for grow, swarm, and shrinking, are likely to be useful in a variety of the above described services, and as such must be general purpose, by adaptable to a large class of devices and networks situations, must be able to work in autonomy by dynamically acquiring all the necessary information required to take decision, and must be able to start operating in the new conditions without human direction.

[SH/UM/WA] Security Services. The consideration that security services are needed may appear trivial. However, the type of security services that one may need to enforce inj the stated scenarios may be very peculiar. In all the scenarios (SH,UM,AH), identification and authentication services may be necessary to acquire sensible information or sensible services. For instance, in SH, only authenticated users should be authorized to act on the heating system. In UM scenarios, only policemen should be authorized (for privacy reasons) to access distributed services and information detailing the characteristics of specific cars/drivers. In WA scenarios, specific sensible information may be in need to be made available on a P2P network, but readable only to authorized users. Not to mention the critical security issues that have to be faced in a scenario of distributed uactions. However, the distributed and highly decentralized structure of the scenarios considered, as well as the fact that they may involve the impossibility to access centralized servers and to execute complex algorithms (consider authenticating a simple note sensor!), suggest that traditional centralized and heavyweight service could not work in that context. In several cases, however, it may be the case that identification and authentication are not necessary to access services and information (e.g., anyone can be authorized to ask "what is the temperature around" or "what is the average traffic load on the highway"). Still, in these cases, the issue arises of establishing some ways to evaluating to what degree a service or some obtained information is trustworth. Thus, some services to compute, evaluate, and make available the trustworthiness of services and information in an open, decentralized scenarios must be provided in any case.

As an additional issue, in any scenarios, some basic services should be enforced not only to evaluate the trustworthiness of the involved services and actors, but also to identify adversarial or free-riding behaviors (e.g., components providing false information, fake services, or consuming an inappropriate number of resources) and mitigate them somehow while self-preserving the functional and non-functional characteristics of the system. In any case, all above kinds of security services are to be provided whenever possible in a lightweight and fully decentralized way, should tolerate a large-scale and dynamic network environment.

4.2 User-Level Services

Beside basic services there are a lot of other "user-level" services that can be identified from the scenarios. These services are typically built by exploiting the basic services, and by adding further logics and functionalities.

Considering the almost infinite number of these services, we extracted only a reasonable number of relevant ones form the proposed scenarios, and use them as a reference for requirements extraction. In this subsection, we detail only a few of these.

In SH scenario for example we can think about some services to find a person *"whereIs(Person(x))"* or an object *"whereIs(Object(x))"*, which can be simply realized by exploiting the basic localization services, possibly in couple with some kind of content-based access to distributed information to get the profile of specific people *"GetPeopleProfile(Person(x))"*. One can also think at a service to track movements of elders *"TrackMovements(Person(x))"*, which can be implemented by exploiting the basic localization service continuously by another ACEs which dynamically builds a profile of the movements.

Even more, one can think at a "*PredictMovements(Person(x))"* service which, based on the tracking of a person, and by interacting with ACEs that have tracked the same persons and/or persons with similar profiles, is able to probabilistically infer the next expected movement of a persons. In the extreme, such "*PredictMovements(Person(x))"* service could work in cooperation with ACEs associated to specific actuation devices (e.g., lamps, heating systems, etc.) to anticipate the needs of an elder person as he moves in the house (e.g., by swithing on lights and heating based on the expected predicted positions).

Another interesting example of this kind of services could be a sort of *"AlertCare(IfPersonLost(Person(x)))"*. This can be realized by composing by different of the above services: localization, tracking, and movement prediction. In fact, to establish if a person is lost, ACEs need information about its location, its past movements, and its predictable movements (to see if the current the position of the persons is outside of any meaningful/usual bounds. If the system cannot find anything to infer that his/her position is justified, an alarm is raised and, to deliver the message to proper actors, an ad-hoc routing service is invoked that also requires giving some high-priority to this message.

Clearly, the implementation of these services requires the capability for ACEs to establish dynamic interactions, and the capability of accessing and organizing a large space of common knowledge about present and past situations, and the capability of accessing a variety of different basic services.

Turning to the UM scenario we can find other types of services. Some of them have analogous in the SH scenario, although with peculiar characteristics, other are specific for UM. Services such as the *"whereIs(Person(x))",* or *"whereIs(Car(x))",* or *"TrackMovements(Person(x))"*, should exhibit the same overall behavior of their SH< counterparts, although exploiting slightly different types of basic services (e.g., GPS localization instead of WiFi one). Access to contextual information must be enforced at a urban scale, possibly by implementing proper overlay to enforce content-based information access, so as to enable services like a *"GetPeopleProfile(Person(x))"*. More user-oriented services can exploit all together information about the social aspects of the users and the current status of the environment so as to provide to a tourist a functionality of "*WhatToSeeNext(Location(current))"*, or so as to suggest to car drivers to "*IdentifyFastestPath(Location(current), Location(destination))"*, or simply to inform user about specific conditions occurring somewhere via a service "*UpdateStreetStatusMap(Data(x))".*

Analyzing in detail two services like *"WhatToSeeNext(Location(current))"* and *"IdentifyFastestPath(Location(current), Location(destination))"* one can clearly see the importance of some basic services: localization, access to local contextual information, context-based access to distributed information, possibly even past information (to calculate in an informed way the most likely fastest path). Services should account users profiles and preferences, i.e., should enforce social awareness, as well as aggregate traffic conditions. And. clearly, to properly deliver information and services in the presence of mobility (of users and cars), should provide context-based overlay networks and routing services suitable for large-scale and dynamic networks.

In the latest WA scenario, we find can find again services that are somewhat analogous to some identified of the UM scenario (e.g., "*GetPeopleProfile(Person(x))*" or *"WhatToSeeNext(Location(current))"*), but again with some peculiarities. For instance, the *"WhatToSeeNext(Location(current))"* service may refer to virtual locations in the cyberspace rather than to physical location. As far as sharing of computational resources is involved, we can identify services to submit to the grid network a task to execute *"SubmitTask(task_x))"*, or some services to try to optimize the execution of a *"Optimize(Task(x))"*, and possibly services to semantically compose together complimentary services *"ComposeService(Capabilities(x))"*. Clearly, it is also possible to conceive this services for execution in an large-scale ad-hoc network (e.g., at a urban scale) other than in WA networks. What we emphasize is that these services may require the support of basic services to dynamically distributed ACEs (e.g., grow and swarm),

As far as autonomic sharing of information is concerned, one can conceive services to enforce clustering of similar data *"ClusterData(Data(x))"*, services to disseminate some data in a network services *"Disseminate(Data(x))"*, and clearly services to query for data to semantically find and compose other composable services *"Query(Data(x))"*, also possibly requiring specific QoS parameters to semantically find and compose other composable services *"Query(Data(x), QoSparameters)"***.** All these services, to work properly, require basic building blocks for overlay network management, but also require access to contextual information related to the current status of the network and of the computational devices, and of course services for the life cycle management of ACEs. Once again, it is interesting to note how in different scenarios the same basic service could assume a different meaning and behavior in different scenarios. For example while in SH the context service should provide mostly spatial information about users, in WA scenario the context is mostly related with the internal state and capabilities of the system's components.

5 Requirements Analysis

5.1 General Requirements

In general, the analysis made in this document clearly outlines that all application scenarios shares a common ground of characteristics which, consequently, all services (whether basic of application-level ones) should properly take into account. These characteristics determine a set of common very general requirements that indeed motivates the whole CASCADAS project. Consequently our analysis confirmed and reinforced initial motivations of the project.

Such general characteristics – and the general requirements derives from them – include:

- **Heterogeneous Components.** In main scenarios the implementation of distributed communication services involve very diverse devices: from smart sensors and PDAs to dedicated multiprocessor hosts: from T3 network links to small capacity and unreliable GPRS connections. To cope with this problem are both required precise standards (to guarantee interoperability) and some sort of semantic to search and organize components. More in particular, this leads to the following general requirements (listed accordingly to their relevance for the various CASCADAS WPs)

- **WP1.** The ACE model should be sufficiently lightweight to be used also on low capabilities devices, and it define standard interfaces to guarantee basic interoperability between ACEs.
- **WP2.** The ACE model should provide methods to retrieve device and network specific information, and such information must enable ACE to adapt their actions accordingly specific actions.
- **WP3.** ACEs need to have sufficient logic/reasoning capability to turn the above device information into action (e.g., a change in behavior as battery level declines).
- **WP4**. Lightweight and standardized tools for identification, authentication, and trust evaluation must be provided.
- **WP5.** To reach a high level of integration, knowledge networks should provide standard semantic mechanisms to organize and compose heterogeneous information and heterogeneous services.

- **Dynamic and Unreliable Networks.** All considered scenarios, but mainly SH and UM, consider small devices interconnected with low bandwidth, unreliable networks links. In these networks, mainly overlays, nodes go up and down in an unpredictable manner. This leads to the following general requirements.
  - **WP1.** The ACE model should take in account dynamic network and should define ways to interact over unreliable network links (spontaneous networking).
  - **WP2.** There should be ways to monitor the status of the network (e.g., its topology), and of the devices within.
  - **WP3.** ACEs must be able to distributed information around accordingly to various means (e.g., anycast and probabilistic multicast such as gossiping), and both accordingly to the physical network or some logical overlay network.
  - **WP4.** All provided security mechanism should tolerate network dynamics as well as network disconnections and unreliable connections.
  - **WP5.** Information about the current status of the network should be organized and possibly continuously updated.

- **Decentralized Systems.** Mostly in UM and WA scenarios ACE's networks will be diffused over a large spatial scale in a very decentralized fashion. Because of this and considering the fact that a central control is not feasible WPs are asked to produce adequate tools to control, self configuration and security. This leads to the following general requirements.
  - **WP1**. ACEs does not necessarily have a clearly identifiable name/identifier, or a specific stakeholders, and must be able to interact and compose without centralized control and without relying on centralized information.
  - **WP2/WP3.** The (self-organized) structure of an ACE network providing a specific set of services is crucial for the organization of an accompanied supervision system. Thus both the organization principle has to be accessible in some machine-readable form (if there is a variety of

principles), and the actual network configuration has to be perceptible.
- **WP2/WP3.** ACE configurations have to be "effectable" in the sense that the supervision system has to have to means to modify the organization principles for this configuration in the case that the organization schema chosen by the configuration is not suitable.
- **WP3.** In addition to the points above about modifying the principles of organization, the ACEs of course must be able to exploit those principles. In other words perceive network organization, reason using internal logic + network data, decide, and act (by changing the network, by moving within it or by following some other behaviour which may or may not impinge on the rest of the network directly).
- **WP4.** Security mechanisms cannot in general rely on some sort of centralized authority.
- **WP5.** Distributed knowledge must be properly organized and aggregated in a fully self-organizing way, without any centralized control.

- **Large Scale.** ACEs should contemplate very large scale systems composed by a huge number of different devices in which services could be composed by a large number of components. This leads to the following requirements:
    - **WP1.** ACE model should guarantee self-similarity, i.e., multi-level and multi-scale composability of services. and consequently the ability to interact with a single ACE or a group of ACEs with the same interface.
    - **WP2.** Self-similarity is also necessary as a mean to achieve a scalable configuration is the major requirement.
    - **WP3.** Cloning functionality, whereby a new ACE is created which inherits identity and, if required, internal state from its parent, is necessary to enable dissemination and distribution of services in a large-scale network.
    - **WP4.** Any provided security mechanism should be inherently scalable.
    - **WP5.** Context-information must be organized in sorts of multi-level hierarchical distributed networks, and provided with different levels of granularity depending on points of interest and resources.

- **Security.** In distributed environments traditional security problems are integrated with other and less studied vulnerabilities typical in distributed systems. In CASCADAS, this implies that the security mechanism may have impact in nearly all WPs, other than in WP4, which translate in the following general requirements:
    - **WP1.** On need, ACEs must be provided with identification and authentication services, and must be able to evaluate actions based on trust measures.
    - **WP2.** The supervision must be concerned with attacks, misuse, etc. On the other hand, security mechanisms itself are – as being essentially software system – error-prone and have therefore to fulfill requirements on observability and effectability.
    - **WP3.** Self-organizaiton mechanisms, without being undermined in their basic "autonomous" nature, should in any case somehow tolerate observability and external control by supervision and security mechanisms.
    - **WP4.** ACE should be able to use traditional interfaces to access services hiding the complexity of the management of a distributed environment. ACE should transparently cope with networks of trust and reputation, and

unreliable and easily interceptable network links.
  - **WP5.** It may be necessary to protect knowledge networks from attacks, and it should be possible to associate "trust" measures to the knowledge carried on by them.

- **Strong and Unpredictable Services Usage.** In a distributed and decentralized environment, without a central control, ACE should be able to monitor themselves and quickly adapt to fast changing conditions. Smart system for QoS management are to be provided.
  - **WP1**. Composite ad distributed services must be built dynamically and must adapt to changing conditions of use.
  - **WP2.** There must be support for continuous system monitoring and nearly continuous interpretation of monitored data.
  - **WP3.** There must be support for automatic reconfiguration of activities and of interaction patterns, as well as for self-organizing growing and shrinking of aggregated services.
  - **WP4.** There must be support for dynamic instantiation of security mechanisms, as well as for mechanisms to recognize denial of services attacks in the form of unusual patterns of activities
  - **WP5.** To achieve a proper reconfiguration meaningful context information are to collected with high time granularity.

- **Situation Awareness.** In every service we described there is always a need to collect data to describe the system. These data are to be collected are synthesized in a sort of understandable knowledge, organized in networks. This leads to the following requirements:
  - **WP1.** ACEs (as a mean to implement knowledge networks) should be able to handle acquire and handle knowledge, other than being able to interpret this knowledge.
  - **WP2**. ACEs have to acquire self-awareness, by somehow being able to manage representations of itself and of the surrounding environment (whether internal or external and distributed in a knowledge network).
  - **WP3**. Knowledge should also enable representing the typical form of "environmental knowledge" useful to enforce biologically-inspired self-organization (e.g., gradients, pheromones, etc.).
  - **WP4**. Security mechanisms should properly adapt to the current status in a situation-aware way, by properly exploiting knowledge networks.
  - **WP5.** Knowledge networks should provide, at the end, a synthesized view of the physical world and of the network world which, at different granularity levels, could show relevant data patterns of examined systems and could enable self-reflection for ACEs.

5.2 Detailed "Per WP" Listing of Requirements

In addition to the above general requirements, we have performed a more fine-grained analysis of the envisioned basic-level and user-level services, so as to extract more specific requirements. In particular, in a sort of "virtual design" exercise, we have tried to sketch out a possible ACE-based implementation of the different services and then, in a sort of sort of “conceptual simulation”, we have tried to imagine the various (even challenging) operating conditions of these services.

From this analysis, we have extracted the most meaningful and recurrent requirements, and have organized them accordingly to their relevance to the various CASCADAS WP. Such requirements, together with already identified general ones, are reported below. We emphasize however the organization per WP and the choice of associating a requirement to a specific WP has not been easy (e.g., a requirements expressing what an ACE should to tolerate self-organization related both to WP1 and WP3), and this is may be arguable. What matter is that the requirement is there and thus must be somewhat accommodate within the CASCADAS framework. In the list of requirements, we have also reported the main application scenario in which each requirement should be most useful (a star “*” means the requirement applies to all scenarios), and the distinction between functional and non-functional requirement. However, also in these cases, the distinction is not always clear cut.

WP1 Requirements

ID	*Functional/ Non Funct.*	*Scenario*	*Requirement*
R_1_1	F	SH/UM	ACEs must tolerate very lightweight ACEs with lightweight interfaces, suitable for lightweight devices.
R_1_2	F	*	ACEs must support interoperability between different devices.
R_1_3	F	*	ACEs must support for interoperability between the network-level, the service-level (i.e., the ACE level), as well as the user level.
R_1_4	F	*	ACEs must tolerate execution over unreliable devices and unreliable network links
R_1_5	F	*	ACEs must support for dynamic and spontaneous aggregation and composition, even in absence of centralized control.
R_1_6	F	*	ACEs' aggregation model should support self-similarity, in which a group of ACEs can be accessed as a single entity.
R_1_7	F	*	Support for transparent aggregation from the outside
R_1_8	F	*	ACEs should be capable of handling both users level events as well as network level and device level events.
R_1_9	F	*	ACEs should be able to communicate with each other accordingly to various means (point-to-point, any cast and multi cast, local multi cast, probability multi cast).

ID	Functional/ Non Funct.	Scenario	Requirement
R_1_10	NF	*	ACEs should be able to implement complex and stateful communication protocols (e.g., negotiations).
R_1_11	NF	*	ACEs should support dynamic interfaces (i.e., should be able to dynamically adapt the provided functionalities).
R_1_12	NF	*	ACEs does not necessarily have a clearly identifiable name/identifier, or a specific stakeholders, and must be able to interact in anonymous way.
R_1_13	F	UM/WA	ACE should be able to include mechanisms to support the building and maintenance of various structures of overlay networks
R_1_14	F	UM/WA	Such overlay must be lightweight
R_1_15	F	UM/WA	Such overlays must be scalable
R_1_16	F	WA	ACE should support mechanisms for dynamic life cycle management (grow, swarm, & shrink...)
R_1_17	F	WA	ACEs should be able to provide services with various and tunable QoS, and also should support for QoS evaluation

WP2 Requirements

ID	Functional/ Non Funct.	Scenario	Requirement
R_2_1	F	*	There must generally be some support for autonomic control system in ACEs and in distributed ACEs aggregates
R_2_2	F	*	There must be general support for pervasive supervision tools
R_2_3	F	*	There must be support for fast reactions when something changes
R_2_4	F	*	There must be support for retrieving device specific information (for monitoring and evaluation), and provide access to device specific configuration functions (for effection/actuation)
R_2_5	F	*	Profiles for specific devices (i.e. meta-information) should

ID	Functional/ Non Funct.	Scenario	Requirement
			be made available in order to interpret monitored data and to turn abstract reconfiguration activities into device specific actions.
R_2_6	F	*	Network components (i.e. the "controlling ACEs") should at least provide a heard-beat function that allows to determine whether a specific network element is still functional.
R_2_7	F	*	There should be a way to monitor whether a new device enters or leaves a certain network segment, i.e., a method to keep track of the actual network topology.
R_2_8	F	*	The existence of situation dependent self-adaptation requires that an ACE configuration has to maintain an internal image of its own structure (probably in implicit/distributed and divided into small parts of knowledge within each of the configuration elements).
R_2_9	F	*	Pervasive supervision systems must protect themselves from attacks (or, which is the same, must be able to exploit the WP4 security mechanisms).

WP3 Requirements

ID	Functional/ Non Funct.	Scenario	Requirement
R_3_1	F	*	There must be support for dynamic system monitoring
R_3_2	F	*	There must be support for dynamic system reconfiguration
R_3_3	NF	*	Reconfiguration process must be fast and without service interruption
R_3_4	F	*	Monitoring activities must be lightweight
R_3_5	F	*	To enable self-organized adaptation of work and topologies, ACEs must be able to move from node to node.
R_3_6	F	*	Cloning functionality, whereby a new ACE is created which inherits identity and, if required, internal state from its parent, is necessary to enable self-organized dissemination and distribution of services in a large-scale network.

ID	Functional/ Non Funct.	Scenario	Requirement
R_3_7	NF	*	Self-organization mechanisms, without being undermined in their basic "autonomous" nature, should in any case somehow tolerate observability and external control by supervision and security mechanisms.

WP4 Requirements

ID	Functional/ Non Funct.	Scenario	Requirement
R_4_1	F	*	There must be support for the exchange of personal and confidential data over the network.
R_4_2	F	*	There must be support to control integrity of data.
R_4_3	F	*	There must be support to specify privacy levels for data and services.
R_4_4	F	UM/WA	There must be support for large-scale networks, and this any Mechanism should be inherently scalable to very large networks.
R_4_5	F	*	There must be support for mechanism to avoid resources exhaustion
R_4_6	F	*	Security mechanism should work without the assumption of centralized authorities/services available.
R_4_7	NF	*	Identification and authentication mechanisms should be lightweight and standardized.
R_4_8	F	*	There must be support to associate "levels of trust" to information and services.
R_4_9	F	*	Security mechanisms may be subject themselves to attacks, and must thus be subject to observability and external control.
R_4_10	NF	*	There must be support for dynamic instantiation of security mechanisms,
R_4_11	F	*	There must be support for recognizing adversarial and free-riding behaviors.
R_4_12	F	*	Security mechanisms should properly adapt to the current status in a situation-aware way, by properly exploiting knowledge networks.

ID	Functional/ Non Funct.	Scenario	Requirement
R_4_13	NF	WA	Support for identification of network faults and automatic reconfiguration

WP5 Requirements

WP	Functional/ Non Funct.	Scenario	Requirement
R_5_1	F	*	Knowledge networks must support for a virtual view of environment to facilitate the concept of interest to adapt to changing conditions
R_5_2	F	*	There must be support for distribution of knowledge across a dynamic network
R_5_3	F	*	There must be support for self-similar knowledge aggregation and for access to knowledge at different granularity levels
R_5_4	F	SH/UM	There must be support to represent and manage knowledge related to the user and the social level (user and social context profiling).
R_5_5	F	*	There must be support to represent and manage knowledge related to the ACE level (profiling of ACEs and of their dynamic and aggregated status)
R_5_6	F	*	There must be support to manage in an integrated (cross-layer) way user-level, ACE-level, and network-level, knowledge.
R_5_7	F	*	There must be support for construction and management of aggregated distributed knowledge.
R_5_8	F	*	To reach a high level of integration, knowledge networks should provide standard semantic mechanisms to organize and compose heterogeneous information and heterogeneous services.
R_5_9	NF	*	To achieve a proper reconfiguration meaningful context information are to collected with high time granularity.
R_5_10	NF	*	Knowledge networks should provide also for producing and organizing new knowledge, inferred from existing one (e.g., for the sake of prediction).
R_5_11	F	*	It may be necessary to protect selected sensible parts of knowledge networks from attacks (or, which is the same,

WP	***Functional/ Non Funct.***	***Scenario***	***Requirement***
			knowledge networks must be able to exploit the security services of WP4).
R_5_12	F	SH/UM	There must be support for spatial knowledge and spatial representation of situations.
R_5_13	F	WA	There must be support for semantic knowledge and shared ontologies to facilitate interoperability

6 Selected Application Scenarios

The above analysis makes it rather clear that, while several common requirements exists for very diverse application scenarios, specific application scenarios exhibit peculiar requirements as it is clear by a comparative analysis of smart home (or urban mobility) scenarios and wide area scenarios.

Accordingly, the analysis confirms what was already somewhat discussed in the initial project proposal, i.e., the need of focusing on two diverse application scenarios, one in the area of pervasive computing (which includes both smart homes and urban mobility scenarios) and one in the area of wide area computing. These scenarios represents extensions and refinements of the basic scenarios already identified in the earlier phases of the project and discussed in Section 3.

6.1 Rationale of the Selection Process

The process of selecting application scenarios has involved several threads of discussions to identify two applications scenarios in the general area of pervasive computing and wide area computing that were both representative enough and that could accommodate the interests and competences of the consortium members.

The scenario in the area of pervasive computing that have been selected is indeed a sort of new scenario, defined by merging together several ideas emerged from the analysis of individual scenarios (i.e., those presented in Subsection 3.1) in the area of smart homes and urban mobility. Such scenario, which consider a large exhibition center instrumented with pervasive devices and populated by a large number of visitor, is very representative in that it integrates into a single perspective the possibility of considering both services typical of smart homes and of urban computing. Within it, it has been decided to consider pervasive services related to both support for independent living (e.g., healthcare and behavior anomalies detection) and for pervasive content-sharing (e.g., pervasive advertisement and social serendipity).

With regard to wide area applications, distributed auctions have been selected in that they exhibit all the problems and requirements typical of a large class of wide area distributed applications. As in most distributed applications, distributed auctions require autonomic support in the form of self-adaptive communications between auction participants, and communications have to self-optimizing and self-healing features by means of adaptive routing on the ACE network and on the network infrastructure, which can be possibly highly dynamic and heterogeneous. Self-protection has compulsory to be implemented by a distributed detection of attacks and early dropping of malicious communications. As in content-distribution services, ACEs participating in auctions need to acquire and distributed information, needs to discover resources and auctions, and may possibly need to move over a networks and self-configure.

6.2 Services for Pervasive Applications

Services for pervasive applications focus on general a scenario enriched with pervasive devices, via which it is possible to provide both services for content-sharing between users and services to support independent living.

The process of selecting application scenarios has involved several threads of discussions to identify two applications scenarios in the general area of pervasive computing and wide area computing that were both representative enough and that could accommodate the interests and competences of the consortium members.

6.2.1 General Scenario

The general scenario falls under the general hat of "services for pervasive applications", within which it is foreseen to exploit two different classes of services: (i) smart environments services for supporting independent living; (ii) behavioral pervasive content sharing services. In practice, such kinds of services, though serving different purposes, all represents instances of either people-to-environment coordination or of people-to-people coordination. Let us however start by sketching the main characteristics we envision for the general scenario.

First, the scenario considers an environment which is densely enriched with sensorial and computational capabilities. In particular, sensor networks as well as RFID tags are embedded in the environment, and can act both as sources of environmental data as well as a sort of environmental computing infrastructures. Sensors can interact with each other in an ad-hoc way, with some users nearby, and can possibly (not necessarily) be connected to some "sink-server" that can be used to collect data. RFID tags can be accessed by nearby users and devices to read and/or store data.

Second, humans populate the scenario, live in it and interact with each and with each other. It is expected that users carry with them some kind of mobile devices (e.g., smart phones and/or PDAs). Via such portable devices, users can be given access to the information produced by nearby sensors and tags, can possibly interact with each other in an ad-hoc way (e.g., via bluetooh or WiFi), and can possibly access to the Internet via some wireless connection.

Most observers agree that the above scenario will become increasingly common in a few years. Because of the ever-growing diffusion of portable devices and of the incredible success of sensor network technologies. Very soon, all our everyday environments, our homes, our cities, will exhibit exactly the above characteristics. It is also rather clear that the above general scenario includes (and expressed the same challenges and requirements of) all those applications that have been classified under the Smart Homes and Urban Mobility hats in the earlier sections of this document.

To better ground the discussion, let us focus on a more specific scenario, which (i) on the one hand, has greater probabilities of becoming reality early than others, (ii) on the other hand, expresses very challenging requirements, which is good for testing CASCADAS results in extreme situations.

Why this can become reality soon? Imagine a large exhibition center which host very successful and crowded exhibitions. An exhibition center of this type (similarly to a stadium or a huge museum) is definitely the place where it can be already convenient to deploy pervasive infrastructures around (sensors, lots of WiFi connections, sensors, tags, location systems, etc.). In fact, exhibition centers may afford the costs of deploying such infrastructures if this enables to provide good services to visitors and (accordingly) attract a higher number of persons and get higher revenues. Also, it can afford the costs of providing each and every visitor the necessary mobile devices to access such services.

Why this is very challenging? Developing and making available high-quality services in that scenario is very challenging and definitely calls for property of autonomicity and situation-awareness. Whatever infrastructure is installed, it has to face incredible bursts of usage (hundred thousands of persons all in a place), and incredible load unbalances (during the interval of an exhibition, most people move to bars) some of which totally unpredictable (a specific exhibition at a place has a great success and attract more people than expected). Such unbalance affects computational load of all devices as well as communication load of network connections. Clearly, this requires the capability of exploiting all resources at the best, enabling both WiFi connections, ad-hoc connections between users and devices, as well as connections through embedded sensors. Also, the amount of data to be dynamically processed can be very huge, and this include not only a large amount of personal profiles and of data coming from embedded sensors, but also a large amount of data describing the state of the network and of the computational devices. Last but not least, the presence of a high-number of diverse and decentralized devices, spread where also common visitors are, calls for the capability of tolerating failures of portions of the overall computation/communication infrastructure. Despite all of these problems, all "user" services should be provided at the better, without experiences significant degradation and properly reacting and adapting in an unsupervised to current network and environmental status.

From a more user-oriented viewpoint, the services to be provided in the above scenario have to satisfy a large number of possible very diverse visitors needs, from children to elderly, from students to professionals, and it is necessary that services takes into account such diverse needs via personalization and dynamic adaptation. All of the above which may call not only for the capability of dynamically reacting to contingencies and user needs but also, whenever possible, of inferring and predicting situations. Inferring situations implies being able to extract from incomplete information available some high-level understanding of what is happening. For instance, recognizing the needs of a user from its current behavior, recognizing that some attack is occurring in the network, or recognizing some unusual situations (e.g., an elderly having some health or cognitive problem) from partial information only. Predicting situations implies the capability of being able to analyze the past and current status (of environment, users, and network), and probabilistically predict what is likely to happen in the future. This can be a service in itself to users, and can be of great use to achieve better adaptivity of the network and faster reactions to contingencies.

6.2.2 Behavioral Pervasive Content Sharing

In general terms, behavioral pervasive content sharing refers to all those services that enable people in an environment to share experiences and digital content based on their own personal preferences and/or current behavior, as well as to those services that enable the environment to adapt to the current users' needs and preferences. Let us now sketch some examples of services of this kind that can be provided within the pervasive exhibition scenario.

The living diary. This is personal user-centric applications, which aims at exploiting the pervasive devices embedded in an environment to produce a sort of digital self-composing diary. When a user moves in an environment (as in an exhibition) with a PDA and enriched with peripherals to access embedded devices (e.g., sensors and RFID tags) and to produce situational information (e.g., a GPS), this can generate a lot of elementary information about the context (i.e., knowledge atoms). On this base, a specific service (i.e., an ACE), can be in charge of collecting all contextual information that is gathered from the environment, and organize such information into a sort of “knowledge networks”, reporting in an organized way all the facts, and events (e.g., people met and objects encountered) having occurred (see Figure 3). The collection of knowledge atoms describing the situations a user is in and has been, stored in PDAs or portable devices and possibly downloaded to some server on need, acts as an historical memory for the user. A user, on need, can then exploit a specific ACE to browse its living diary. An another example, in an exhibition center, it is possible to think at some exhibition-specific ACE that, by browsing in the past the living diary of a user to detect detailed preferences and habit, can act as a personalized guide to the exhibition. To some extent, we can classify this as an example of a “people-to-itself” coordination services.

As simple as it can be, the living diary introduces key problems related to situation-awareness (by definition, since the very goal of the application is to analyse and correlated situational knowledge) autonomicity (the living diary should be composed in background, and ACEs should do their best to compose it in a meaningful and as much complete as possible way, independently of what specific devices and connections are available at a given time), and security (for obvious reasons, a user must fully trust his own living diary).

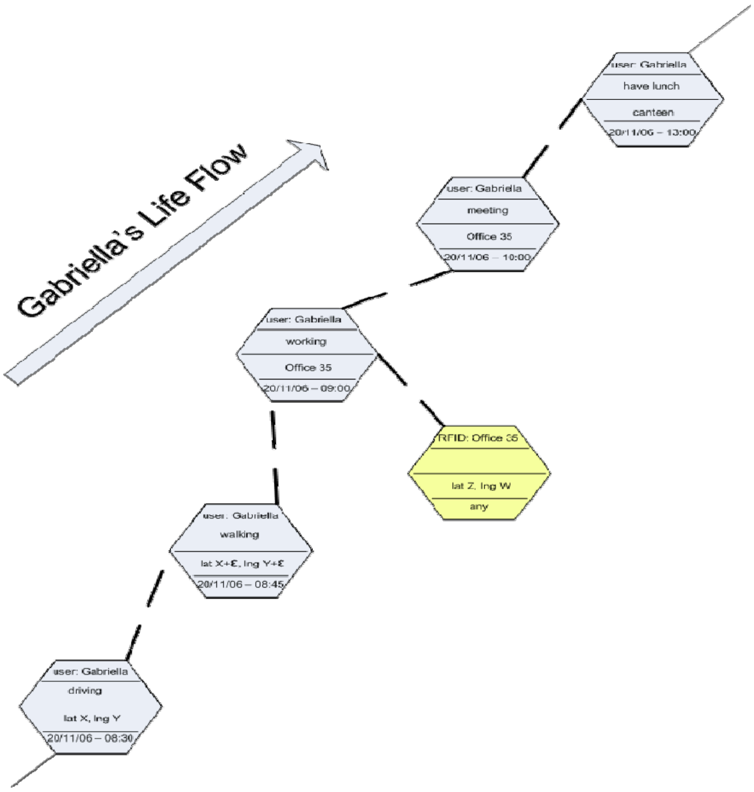

Figure 3. Gabriella's living diary.

Social Serendipity at the Blogcafé. Imagine that, in the exhibition scenario, there exists a sort of open place for people to meet or simply to rest (a cafeteria for example). We assume that such cafeteria is enriched with pervasive and wireless devices as the rest of the exhibition. When a user enters the cafeteria, some devices (e.g, an RFID reader) can recognize that such user has entered, so that the cafeteria is always aware of who's in it. At this point, a user can decide to share with other users (either directly in an ad-hoc way or by mediation of some server of the cafeteria) his own personal profile, or even his living diary.

By sharing (portions of) the living diaries, and by allowing group of ACEs to navigate and analyze the living diaries of the persons in the cafeteria, one can think at discovering relations between persons, common interests, or common past events. As a trivial example, ACEs can identify that two persons have just assisted to the same show in the exhibition, and can decide to signal this fact to them so as to promote socialization and exchange of experiences. More in generally, ACEs can cooperatively explore the past life of users to users and discover facts and social affinities that users' could have never discovered otherwise. In the example of Figure 4, ACEs could discover that Gabriella and Alberto already met in the past during a meeting. This can be classified an a service supporting people-to-people coordination.

In addition to the problems already identified for the living diary (situation-awareness, autonomicity, security and trust), this services introduces additional challenges related to the need for group of ACEs to cooperatively explore the living diaries of a possibly very large number of persons, and to the need of doing this in a self-organized, adaptive, and resource-effective way. Not to mention the fact that, since living diaries may not necessarily fully reside on users' PDA but may be stored somewhere remotely, there may also be the need of coordinating the activities of ACEs distributed at a larger-scale.

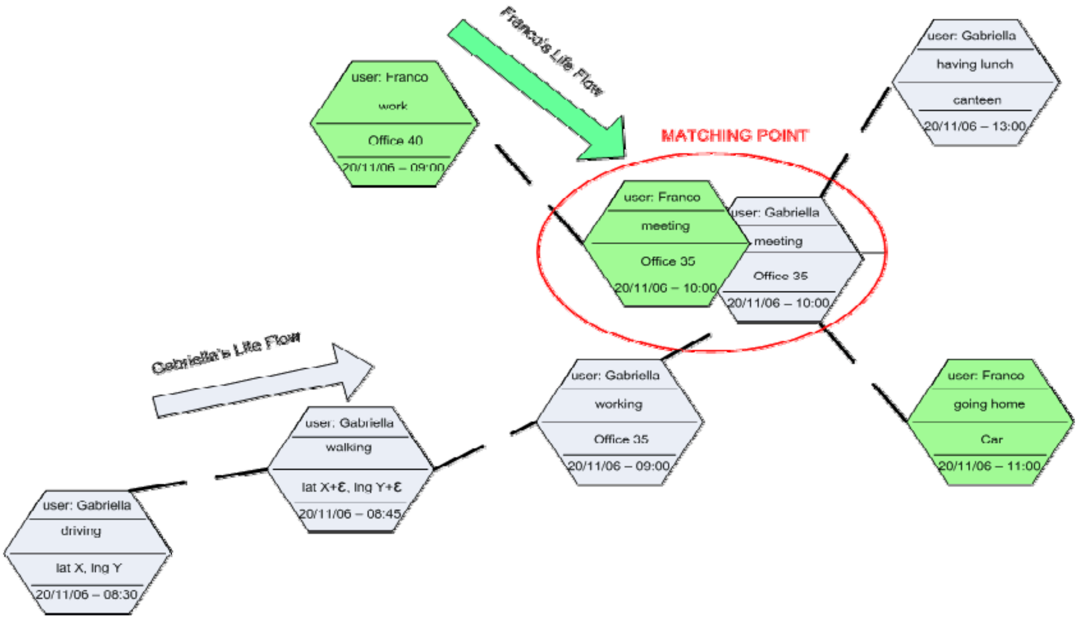

Figure 4. Social serendipity at the Blogcafé.

Behavioural pervasive advertisement. Behavioral advertisement (or behavioral targeting) generally tries to ensure that advertisers reach the target audience in a more effective way. In the envisioned scenario, this may results in specific advertisement being directed in a personalized way to individual PDAs, and/or in wall displays at the exhibition to show those ads that, at a given moment, would better fit the surrounding audience (see Figure 5). With this regard, in the exhibition scenario, we can imagine that users' profiles can be gathered either via reading of some personal RFID or by accessing some personal profile and living diary stored somewhere else. ACEs can be devoted to analyze this information, matching it against a database ads profiles, and decide what to do. In this service, one can envision that not all the information necessary to identify user preferences is necessarily made public by the user (e.g., the personal profile of a user can only mention the fact that he is a student of 23 and nothing else). Thus, ACEs may need to gather additional information to perform a reasonable match between user preferences and ads. To this end, one may think at having additional ACEs that proactively explore external sources of information to gather the additional information that may be needed to perform a better match. For instance, one could think at having them access some knowledge repository or exploit the Google API to discover that, probabilistically, a 23-years student prefers to be advertised about an incoming visual show rather than to a poetry reading. This can be classified an a service supporting people-to-environment coordination.

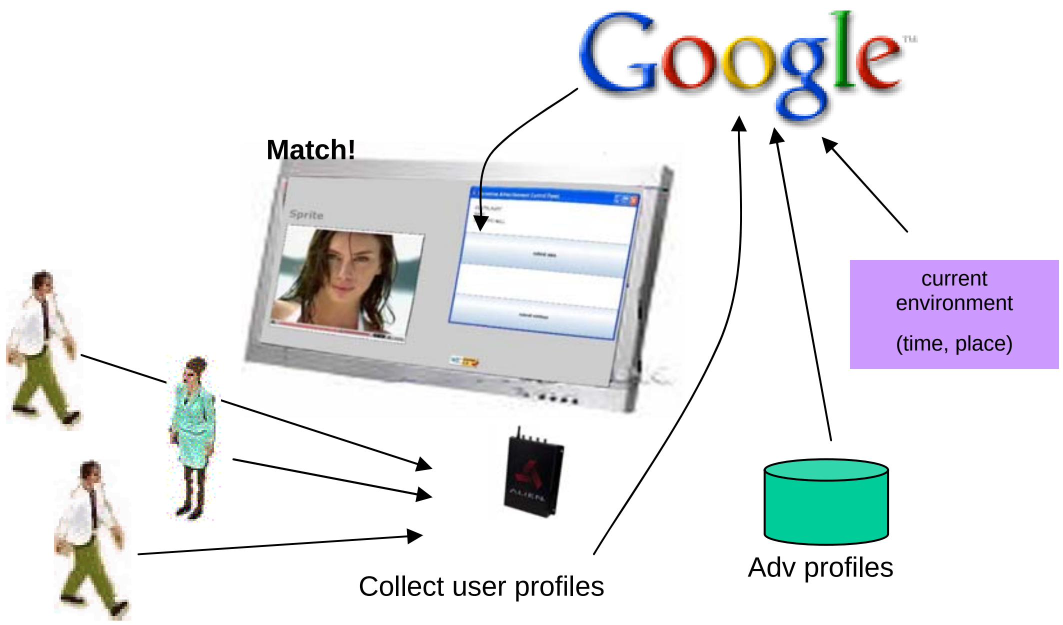

Figure 5. Behavioral advertisement.

In addition to the problems exhibited by also the other services, behavioral pervasive advertisement introduces greater challenges as far as analysis of knowledge and dynamics of the scenarios are involved. In fact, with regard to the former point, one must account that the system should be able to automatically self-tune the frequency and the diversity of advertisements, and should be able to deal with a possibly highly varying number of different users and different profiles. Also, such service should be able to self-monitor its performance so as to self-adapt to reach increased levels of performances (i.e., increased accuracies in matching).

6.2.3 Supporting Independent Living

As the population continues to grow, society is faced with the challenge of supporting those within the community who still remain within their own homes and are not fully independent. Independence can be regained by a number of stakeholders providing support – healthcare providers may visit the home to administer medication and monitor vital signs, close members of the family may be responsible for the reminding of simple daily activities and neighbours may offer the comfort of security by being in close proximity. It has become more and more widely appreciated that the application of technology within the home environment can provide, to a certain extent, a degree of independence which may have previously been provided by one of the aforementioned stakeholders. The major benefits of deployment of technology in such a manner is the potential to extend the duration a person remains in their own home and thus avoids institutionalisation. Although technology can promote independence and indeed impact positively upon a person's quality of life, detrimental effects can also be witnessed if reliance upon the technology reaches a level where the person will not leave their own home in fear of losing the support once outside of the home. Taking this into consideration it becomes necessary to ensure the technology has a degree of portability and continuity of service to ensure its support both inside and outside of the home environment. This requires the continuum of services from within the home to places like supermarkets, hospitals, exhibitions, and other places where people are likely to visit.

To avoid reducing the net impact of devices such as cognitive prosthetics (i.e., healthcare technology that helps a person with cognitive impairment to function more independently) introduced within the living environment requires ensuring a continuum of service i.e., the service provided is not lost once the person leaves their home. If we consider the typical requirements that a typical ageing person with minor cognitive impairment would exhibit, then a key requirement would be spatial orientation reinforcement i.e., continuation of support from their network of family/friends/carers when they leave the home. An example of when such a service would be required would be in instances of the person not being able to find their way home or to remember why they have left the house, for example, to attend an appointment. From a technical perspective addressing the problem of spatial reinforcement is largely complex for a number of reasons. In the first instance the service must be delivered to available stakeholders at a number of levels of granularity. For example, provision of service with those in close proximity perhaps in the same building or street, or those in the same town or alternatively those who may be simply 'available' to offer support but may be located within a larger geographical area. A secondary issue for consideration is are the people within proximity able to support the person in need? For example a person who requires some advice as to why they have left the house may only be able to receive support in this instance by a family member who has access to the person's daily agenda. On the other hand, if the stakeholders' prosthetic raises an 'emergency suggestion for intervention' message then the appropriate stakeholder who can be contacted should be made aware of the situation.

If we assume in the first instance that communication between the person's cognitive prosthetic and the stakeholder's prosthetic can be established the problem becomes one of information management within a networked environment addressing the dynamic positioning of the person and stakeholders. This suggests that the requirements of the technical service should provide for the following: situation awareness – to identify the positioning of the person concerned and relevant stakeholders; self organisation and adaptation – to self-identify the best remedy in instances of alarm, and adapt this depending on contingencies,

Services to be provided. Entering in more details about the specific services to be provide it quite difficult, in that the number of such services is potentially very large. However, and without the ambition of being complete, they may be categorized as follows.

Domain Dependent Services. Domain dependent services include services for cognitive reinformencent. This include: services for helping people to remember, such as “GoTo(location)”, which can be classified as of people-to-environment coordination; services for social reinforcement (aiming to help people to maintain social contact inside and beyond their own family), such as “Call(Person(x))”, “WhereIsPerson(y)”, “WhoIsClosestPerson(x)”, which can be classified as forms of person-to-person coordination; services for functional reinforcement (to help perform daily life activities), such as “EatLunchAt(Time(x))”, which can be classified as a form of person-to-environment coordination; and services for spatial reinforcement (in order to improve and extend a person’ mobility), such as “AlertCare(IfPersonLost(Client(x)))”, “WhereAmI?”.

Operational Services. These include networking services, that facilitates social spheres in an ad hoc manners, services that we have already somehow introduced in the BLogcafé example, and communication services, such as being able to communicate anytime, anywhere, with anything, an issue that is more properly analyzed into different scenarios.
Management Services. This is overarching or master control of all of the aforementioned services within this specific application domain.
Also in these cases, and beside their simplicity, the realization of these services introduces several key problems, making them a useful application case study to test CASCADAS results. ACEs realizing these services must be interoperable, must acquire spatial-awareness, must be light-weight and resource-efficient, and are not allowed to fail. Performances of the system needs to be continuously improved via a process of self-monitoring and self-supervision. For services to act properly, they must be highly cognitive and must be able to acquire and process meaningful situational knowledge, both at the global and at the local level. Security is critical, in that one cannot tolerate failures or misbehaviors due to attacks.

An Example Scenario. Consider a person, suffering from mild dementia, who has to go regularly to a health clinic at certain time intervals to retrieve specific treatment. The potential problem facing persons suffering from mild dementia is that they may not only forget the time and the place of the scheduled visit but upon arrival they may also forget the actual purpose of the visit. If a pervasive support environment could be established which would extend from the person’s home to their intended destination (beyond the home) and could be supported via a simple mobile device, such as a PDA, this could be used to guide the person from a spatial perspective i.e. in providing directions and in providing cognitive reinforcement i.e. the purpose of the visit. For intelligent services to achieve such behaviour it is necessary that individual components of autonomous smart environments become context-aware and exhibit self-management capabilities in order to achieve their objective.

This type of autonomous, situation-aware communication is depicted as an example in the Figure 6, where a person-to-environment scenario is visualised. In this scenario a virtual orb surrounds a person that moves within a smart environment (in this case the health clinic) and as such not only senses its immediate surroundings but may also interact with individual services provided throughout the smart environment. Such communication is not necessarily limited to the 'smart building' itself but could also communicate with any intelligent component thereof and of course individuals (family/friends/careers) that could provide specific help and guidance or even raise alarms if necessary.

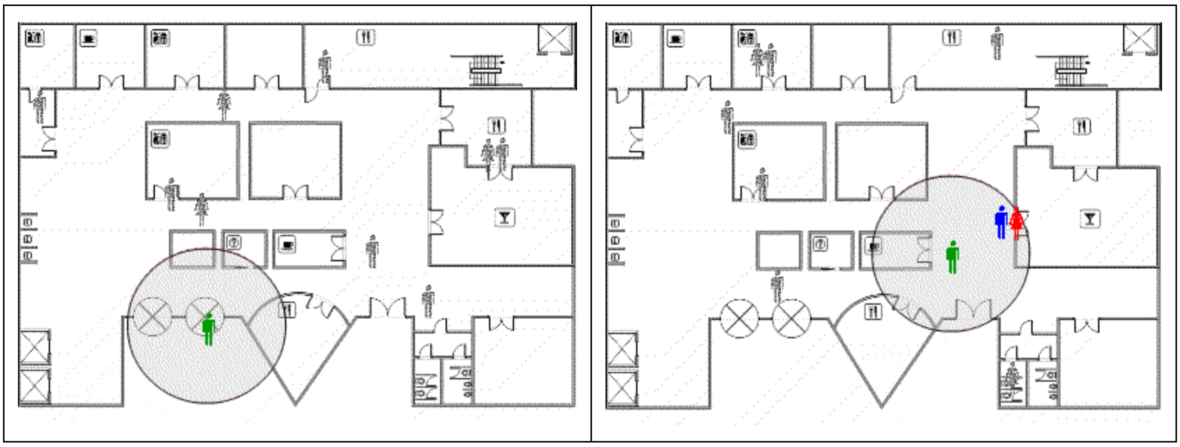

Figure 6: Examples of People-to-Environment Interaction with a smart environment.

Within this scenario the notions of spatial orientation reinforcement can be further exemplified; stakeholders within close proximity to the person can be assessed to gain an appreciation for the level of support they may be able to offer. Consider a scenario such as depicted in Figure 7, where people-to-people interaction is visualised. In this instance the person is represented as the green icon in the centre of the virtual orb with stakeholders who can offer any form of support as icons of various other colors.

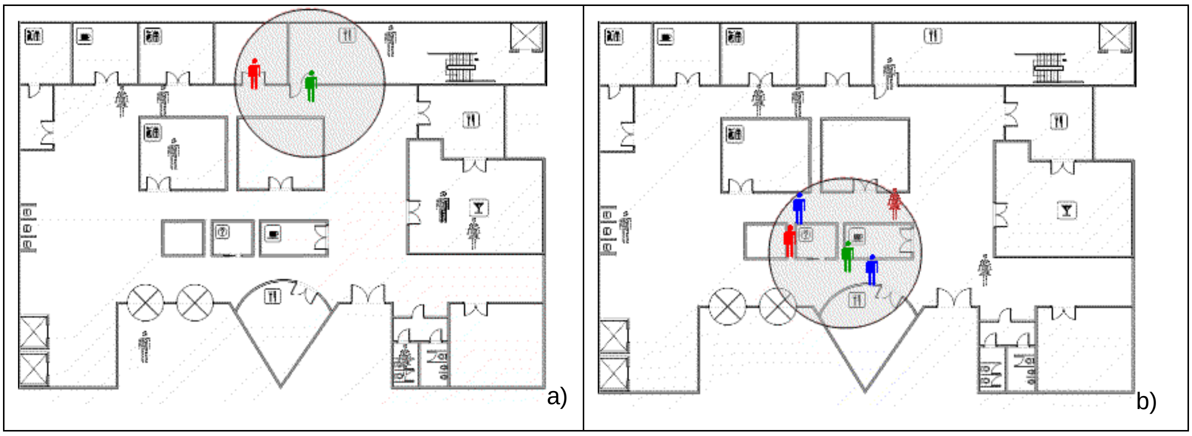

Figure 7: People-to-People Interaction (a) Person can obtain support from one stakeholder (b) Person can obtain support from four stakeholders.

Firstly, a service may have identified a stakeholder in the near vicinity who can provide orientation reinforcement, which is in this case visualised by a red icon as shown in figure 7 (a). As the person moves the service detects that there are now other types of stakeholders who can provide support in instances of required cognitive and orientation reinforcement as visualised by colors blue and brown, respectively. This is represented in figure 7 (b) which identifies individual stakeholders even if there are not directly visible by the person concerned. The system may also has identified a number of other stakeholders who may be likely to move into the proximity of the person, but at present, have not been considered by the system to be in a position to offer assistance (coloured as light grey and as yet outside the virtual orb).

Similar concepts could also be applied with in the home environment of a person, as well as within other more global pervasive environments such as towns, country's etc, as well as – of course – in adopted the exhibition scenario.

6.3 Distributed Auctions

Consider a likely scenario in the future global economy organized around networked auctions between enterprises and between individuals and enterprises. In this, auctions will be completely automated by being conducted by software entities acting on behalf of the user. Auctions might have very short duration in time, with some of them possibly finishing within seconds. In this scenario, delays as small as few milliseconds might cause considerable losses. Moreover, the evolutionary nature of the market often creates situations where the same good is auctioned by more than one auction, some of which under more convenient conditions (consider for instance holiday packages, where the same conditions have different prices based on the offering agency).

Autonomic distributed auctions will allow management of networked auctions in such a scenario in an autonomic fashion. The software entity will manage many concurrent auctions on behalf of the user by autonomically deciding what to do and when, to the extent of fulfilling the overall user aim. In addition, failures by infrastructures, or even unwanted delays in the auctioning process will be avoided by autonomically moving in the background network. For instance, consider a company whose business is in the fast changing market of technology. If the company sells technology goods such as, for instance, mp3 players and game consoles, it is desirable for its warehouse to be able to sustain a fast turnover of goods, where older models will be replaced by newer ones upon release. Autonomic auctions will be an effective way to manage goods in the warehouse, selling older ones and acquiring newer ones without human supervision. The process of acquiring new wares through auctions will be conducted by the software entity in charge of management by acting autonomically based on knowledge of the environment the entity interacts with.

The aforementioned scenario will require a robust communication infrastructure capable of providing differentiated levels of quality of service to its users and to deliver, in a reliable and timely manner, a large number of data packets. This scenario will benefit greatly from an autonomic communication network able to operate in an opportunistic way and that offers self-* features by exploiting situation awareness. Autonomic Distributed Auctions is considered an excellent use-case to demonstrate the CASCADAS framework. In fact, the key role played by autonomic communications allows increase the potential of electronic auctions in terms of number of auctions to be handled concurrently, and efforts required in supervision of the system.

The scenario consists of three main types of users:

- Buyers (B): users that search for an item, and willing to bid for auctions on items of interest.
- Sellers (S): users owning items under auction. Sellers start auctions by advertising items on one or more Auction Centers.

- Auction Centers (AC): users acting as meeting point for buyers and sellers. Typically they contain a representation for a particular sub-system of networked auctions, visualized as an *Auction Web Page* (AWP) containing the list of items currently auctioned along with a description of auction terms and conditions.

Roles can be undertaken on a per-auction basis, and are not mutually exclusive in the context of multiple auctions. For instance, a user can act as seller for one or more auctions, buyer for other auctions, and AC for another, set of auctions. Obviously, fairness will be addressed whereas conflicts of interest might arise.

Buyers and sellers announce their goods (or services) by advertising them on one or more ACs with an indication of the auction model required for the transaction. A seller, for example, may announce the she is offering a good G, for an initial price P, from time T1 and to be sold under an English auction model. Previous agreements between the seller and the auction centre would allow the transaction to take place with the proper security. Likewise, buyers would require a certain level of authentication to proceed. Buyers will find the good by consulting the centre on demand. Interested buyers will contact the seller and a distributed auction will take place in an automated or semi-automated way, hold by the seller. Prices updates will be reflected in the centers to inform possible additional participants in the auction.

The application models auction models through a set of parameters, which can be set in such a way to define auction rules for the most widely accepted and used auction models. Among others, the most famous are:

- *English auction*: the seller starts the auction with a low price, and bids are aimed to increase the initial price. Also called *Open-outcry* auction.
- *Dutch auction*: the seller starts the auction with a high initial price, which is decreased until one of the bidders is willing to accept to pay the auctioneer's price.
- *Japanese auction*: bidders are forced to bid at each round to remain in the competition for the correspondent item. When a bidder decides not to bid, it is automatically excluded and it is not allowed to bid in future. The auction finishes when only one bidder remains.
- *Sealed-bid first-price auction*: all bidders simultaneously submit bids so that no bidder knows the bid of any other participant. The highest bidder pays the price they submitted. Also known as *Sealed-High bid auction* and *First-Price Sealed-Bid auction* (FPSB).
- *Sealed-Bid Second-Price auction*: identical to the Sealed-bid first-price auction, except the winning bidder pays the second highest bid rather than its own. Also known as *Vickrey auction*.
- *All-pay auction*: all bidders must pay their bids regardless of whether they win the prize. The highest bidder wins the prize.

These auctions will be modeled through parameters defining basic rules:

- Price: initial price of the item.
- Bids: states defines the tendency of bids to increase or decrease the initial price.
- Rounds: defines the rounds the auction is based upon (either single or multiple).

- Form of bids: defines the public or private nature of bids. This latter choice allows to model a category of auctions called *closed auctions*.
- Occurrence of transactions: defines the way transactions between auctioneer and bidders are conducted. *Periodic* or *Continuous*. When set to this latter value, auctions are said to belong to the category of *continuous auctions*.

In addition, through other parameters the application will be able to model variations of such auctions:

- Reserve: defines the reserve price, i.e. the minimum price for which the item can be sold.
- Expiration: defines auction expiration. Allows to capture behavior of all types of timed auctions.
- *Ordering mechanism*: allows to define the way bids are carried out. Either *order-book*, to indicate asynchronous bids, or *clocked*, to indicate that bids need to be proposed within well defined interval times. Setting of this parameter to the former value allows to capture behavior of a category of auctions named *call auctions*.
- Items Number: defines whether the item under auction is single or multiple (in which case it might be split).

The system will contain immediate support for a set of pre-defined auction models, i.e. a combined setup of the aforementioned parameters that will reflect rules and behavior complying to well known auction models. However, whereas auction rules and behavior do not match any known auction model, the user will be able to setup a customized combination of parameters so as to be capable of handling (either imposing, as a seller, or interpreting, as a buyer) rules eventually specified by the auctioneer.

On the architectural side, a physical user will be represented, in the auction platform, by an *Auction Application* (AA), composed by an *Auction Strategy* (AS) and an *Auction Logic* (AL) layer (see Figure 8). The former will drive auctioning aspects tied to strategy, such as timing and selection. For instance, in an auction where the AA is acting as seller this layer will typically decide the right moment to start an auction, the right market (i.e. the right ACs) for advertising the item, and the right bidder to sell the item to. On the other hand, when acting as buyer the AS will decide the right time to participate to an auction, the whose AWP contains a higher number of items of interest, the sight seller to place a bid to, and so forth.

The AL will be in charge of handling auction aspects tied to the actual per-auction behavior. When acting as a seller, the AL will be in charge of make sure that bidders participating to the auction respect the auction rules. On the other hand, when behaving as a buyer the AL will be responsible for interpreting auction rules correctly (eventually adapting to rules which are not previously known) and, after consultation with the AS, place the actual bids accordingly.

Underneath the AA, the per-user architecture foresees the presence of an *Auction Unit* (AU), which represents the physical user in the auction platform. The AU will provide the autonomic support throughout the whole lifecycle of the object and will encapsulate the ACE.

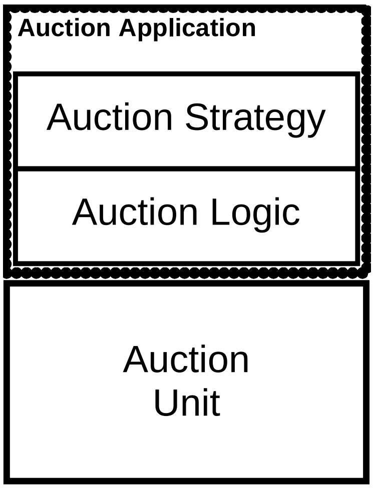

Figure 8. The Auction Architecture.

Autonomic support will be provided in the form of self-adaptive communications between auction participants. Communications will offer self-optimizing and self-healing features by means of adaptive routing on the ACE network. A distributed detection of attacks and early dropping of malicious communications will implement self-protection. Auction participants will be allowed to virtually move (migrate to a different ACE) to gain differential advantage. In order to migrate a service, the target location will need to be determined from available information (e.g from knowledge networks). Relevant information will be passed to the target, such as auction participants and auction states, and any residual communication would need to be handled (e.g. by redirection). Virtual movements are expected to be managed via self-configuration.

7 Conclusions

CASCADAS is a project driven by the ambition to define a new general-purpose approach to support the design, development and execution of advanced communication services for model application scenarios. By promoting autonomic behavior and by supporting situation-awareness and adaptability to situations, CASCADAS can increase the level of satisfaction in services usage and can dramatically reduce the cost of managing services life cycle. Clearly, for the activities of CASCADAS to be meaningful and well grounded on the real needs of users, developers, and managers, we have to make sure that all research goals of CASCADAS represents real needs of the actors involved and, viceversa, that all needs are properly addressed in CASCADAS.

To this end, the first activity undertaken in WP6 as part of task 6.1 has focused on analyzing in details the application scenarios that could potentially take advantage of CASCADAS researches, has tried to identified a large set of relevant services that one may be in need of develop in such scenarios, and has tried to extract meaningful requirements from such services. Eventually, such requirements have been listed and organized with regard to their relevance for the activities of the scientific work packages. The above analysis activity, performed with a rigorous methodological approach, has enabled to: *(i)* re-enforce the initial motivations of the project by emphasizing that all scientific research activities are indeed motivated by important requirements; *(ii)* identify in detail what are the expected functionalities and tools – as they emerge from the real needs of modern applications – that the various research activities should strive to achieve and deliver.

The detailed analysis of the requirements has also made it quite easy to subsequently identify, as a second thrust of activities within task 6.1, what application scenarios were representative enough to act as a common basis for demonstrating the power and effectiveness of CASCADAS research outcomes and, at the same time, were complex enough to challenge them.

As a final word, we are confident that the result of the analysis work summarized in this document will strongly support CASCADAS researches and will contribute to make CASCADAS a project that – despite its inherent foundational nature – will always be strongly driven by real-world application needs.

8 References

The full description of all the above described application scenarios can be found on the CASCADAS Wiki Web site.