Towards an Agent Model for Future Autonomic Communications

Francesco De Mola[1], Raffaele Quitadamo[1]

Abstract — **The continuous growth in ubiquitous and mobile network connectivity, together with the increasing number of networked computational devices populating our everyday environments (e.g., PDAs, sensor networks, tags, etc.), call for a deep rethinking of traditional communication and service architectures. The emerging area of autonomic communication addresses such challenging issues by trying to identify novel flexible network architectures, and by conceiving novel conceptual and practical tools for the design, development, and execution of "autonomic" (i.e., self-organizing, self-adaptive and context-aware) communication services. In this paper, after having introduced the general concepts behind autonomic communications and autonomic communication services, we analyze the key issues related to the identification of suitable "component" models for autonomic communication services, and discuss the strict relation between such models and agent models. On this basis, we try to synthesize the key desirable characteristics that one should expect from a general-purpose agent model for autonomic communication services.**

Index Terms— **Autonomic Communication, Services, Self-organization, Self-adaptation, Multiagent Systems**

I. INTRODUCTION

OUR everyday world is increasingly being populated with a wide variety of new communication technologies and computing devices. On the one hand, several wireless and ad-hoc communication solutions are being deployed with the potential of ensuring us 24/7 ubiquitous connectivity to the Internet and to the surrounding devices. On the other hand, devices such as sensor networks [6], RFID tags [18], cameras, GPS and other location systems [9], will enable us to dynamically acquire information and interact with the physical world.

The above scenario opens up the possibility for a wide range of brand new applications (e.g., on-line monitoring of the world [6] and enhanced social experiences [13]), as well as for enhancing the quality and effectiveness of current communication services (via context-awareness and dynamic personalization [5]). However, it also introduces a dramatic increase in complexity and a number of novel design issues, challenging current communication and distributed computing paradigms, and making it difficult to deliver the promised benefits in truly usable and economically feasible ways.

The complexity we are talking about is due to several factors, there included:

- *Heterogeneity of involved components*. The range of newly introduced network and computing technologies is already wide and it is expected to grow consistently.
- *Dynamism of network scenarios and applications.* As connectivity is becoming ubiquitous and mobile, and as computers are getting embedded in our everyday objects, the resulting network becomes highly dynamic in terms of topology and usage patterns.
- *Decentralization and unreliability*. The highly decentralized and embedded nature of the involved components, makes it hard (whether not impossible) to enforce some forms of direct control over their configuration and their activities.

Other than from the above sources of complexity, additional challenges are introduced by the need of exploiting in full the potentials of the new scenarios and put them at the service of users. This implies identifying suitable models and tools by which *innovative services* can be designed, developed and deployed, and by which existing and new services can be made more flexible and *dynamically adaptable*, i.e. able to properly react to the dynamics and unreliability of the scenario without suffering from any malfunctioning, and able to increase user satisfaction by adapting their behaviour to the current context (physical and/or social) of users and to their own individual needs.

The emerging inter-disciplinary research area of autonomic communication [1, 14] attempts to overcome the limitations of current communication models and architectures in addressing complexities and issues raised by modern network scenarios. In particular, autonomic communication broadly relates to the study and development of novel semantic communication models [5], novel adaptive and evolvable architecture for network components [3], as well as novel paradigms and tools for the design, development, and execution of *autonomic* (i.e., self-organizing, self-adapting, and context-aware) *communication services* [12].

---

[1] Dipartimento di Ingegneria dell'Informazione – Università di Modena e Reggio Emilia - 41100 Modena, Italy, email: `{demola.francesco, quitadamo.raffaele}@unimore.it`

In this paper, we specifically focus on autonomic communication services with the goals of: *(i)* analysing the key issues related to the identification of novel software engineering approaches and of a novel "component" model for the design and development of autonomic communication services (Section II); *(ii)* eventually, trying to synthesize the key desirable characteristics that one should expect from a general-purpose component model for autonomic communication services and the contributions that can come from the agent community (Section III). The key message we hope to get home is that current researches in software agents and multi-agent systems have the potential for playing a major role in inspiring and driving the identification of such model, and more in general for influencing and advancing the whole area of autonomic communication.

II. AUTONOMIC COMMUNICATION SERVICES

Autonomic communication generally refers to all those research thrusts involved in a deep foundational re-thinking of communication, networking, and distributed computing paradigms, to face the increasing complexities and dynamics of modern network scenarios. The ultimate vision of autonomic communication researches is that of a networked world, in which networks and associated devices and services will be able to work in a totally unsupervised – i.e., autonomic [10] – way, being able to self-configure, self-monitor, self-adapt, and self-heal. To some extent, the idea is to consider networks as sorts of immense organisms, and by conceiving components within as parts of these organisms, able to prosper and autonomously survive contingencies [12]. On the one hand, this will enable to effectively have networks capable of dynamically adapting their behaviour to meet the specific needs of individual users. On the other hand, this will enable to dramatically decrease the complexity – and the associated costs – currently involved in the effective and reliable deployment of networks and communication services.

A. *Scenarios of Autonomic Communication*

The need for re-thinking communication and distributed computing paradigms directly derives from the novel characteristics exhibited by modern and emerging network scenarios. Traditional communication and distributed computing paradigm were conceived to target a now obsolete perspective of computer networks: wired networks of (rather homogeneous) medium/high-end computers and routers. In such scenarios, network disconnections and failure of components are considered exceptions, and network and system managers are always assumed to be able to act on the system for re-configuration and fault-recovery. However, as stated in the introduction, modern network scenarios more and more include a large number of very heterogeneous components (from low-end computer-based sensors, to PDA, laptops, and workstations), interacting over a variety of wireless channel (from WiFi, to Bluetooth and ZigBee), and in the presence of mobility (of both devices and users exploiting them). There, failures of components and network disconnections are the norm rather than the exception, and the possibility for network and system managers to intervene in the system is challenged by the intrinsic decentralization and complexity of the scenario.

Figure 1. An urban scenario of autonomic communication

Just to reach a better understanding of what such scenarios could look like, imagine what our cities will be in the next few years (see Figure 1). First, a variety of computer-based sensors will be spread around in every street, crossing, squares, and within buildings. We can already find a variety of simple sensors around in our cities (e.g., to measure traffic intensity and pollution), but the future will see these sensors become wireless-enabled, and dramatically increase in density and diversity. It is not unrealistic to think that – say in ten years – it will be possible to determine in real time how many free benches are there in a specific park or how long is the queue at the nearest post office. Second, wireless-enabled computing devices will be worn by each and every person and will be embedded in any cars. Such devices, beside the capability to access to the Internet, will also be able (via ad-hoc wireless communications) to directly interact with each other and with sensors around, and localize themselves via GPS or other means. After all, smart phones with GPS, Bluetooth, and cameras, are already a reality. Third, all of these devices will be able to mobilize data from and to the Internet, based on a variety of communication channels, from WiFi, to UMTS or satellite communications.

The heterogeneity of components and network technologies involved in the above scenario is very evident, as it is the fact that the resulting network is highly dynamic (due to both the unreliable nature of sensors and the ephemeral and mobile nature of wearable and car-embedded devices) and highly decentralized (no system manager could enforce a strict control over dispersed sensors and over personal devices). This factors clearly justify the efforts of autonomic communication researches towards the identification of: (i) innovative ways of modeling communication, suitable for dense dynamic networks of wireless devices and overcoming the limitations of traditional point-to-point Shannon-oriented communication models; (ii) the definition of innovative flexible architecture for network devices, suitable to tackle dynamics and decentralization via dynamic re-configuration; (iii) the identification of innovative models and tools for the design, development, and execution of autonomic communication services.

B. Towards Autonomic Communication Services

In general terms, a communication service is a functionality that is made available within a network to access and exploit the network resources. IP datagram routing, DNS, socket-based point-to-point communication, cryptographic tools, web services and P2P data delivery services, can all be considered communication services. The definition applies independently of the fact that such services can be conceived to act either as "user-level" services or as "infrastructural" ones, to be put at the service of other services.

In the sketched scenario, a variety of communication services can (should) be put in place to properly access and exploit the available network and computing resources, some of which of an innovative nature.

At the more infrastructural level, we could think at localization services that, by exploiting GPS, WiFi signal strength, or whatever localization tool is available, are able to provide the location of users, cars, or devices. Also, we could think of a variety of routing services, able to deliver data and messages across the network, from more traditional routing services offering delivery to a specific network ID, to more advanced routing services capable of delivering messages at specific locations of the network (for which the routing service has to exploit the available localization services) or to multicast messages at specific groups of nodes or users.

Shifting to the user level, the presence of sensors, ubiquitous and mobile connectivity, localization services, opens up an incredible range of possibilities for the deployment of highly innovative and useful services. By properly exploiting sensors, localization services, and proper routing services, one could think of making available to users various services to query the physical world and obtain any kind of information about the surrounding situation (there included other users), and possibly to integrate this information with information dynamically downloaded form the Web. As another example, one could think at elaborated services to alleviate roads congestion problems. This would imply devices in cars (for computing, sensing and visualization) to interact with devices in streets and crossings (for sensing the current traffic situation and communicate it to cars). Cars could also interact with each other (the same as sensor could) and form wireless ad-hoc network that can be used to properly forward information across the town. The overall service could then exploit all this available information to map in real-time the status of streets in the city, and calculate on-the-fly faster routes for users that avoid congestion areas or areas that are likely to become congested soon.

Whatever communication service one can think of, and whether at the infrastructure level or at the user level, it is clear this will be generally realized in terms of some software components (though it may be the case that some components can be directly encoded in hardware) [11, 12]. Such components will act as access points to the service, and will be able to provide the service either in autonomy or by interacting with each other in the distributed network environment, as it should necessarily happen for all those services, like routing, which are of an intrinsic distributed nature.

However, in modern network scenarios, the only possible way to effectively develop and deploy services is by making them autonomic, (i.e., capable of self-organization, self-monitoring, self-healing) and flexibly adaptable (to meet very diverse situations and diverse user needs). For instance, a localization service should always be able to provide information on a best-effort basis in any situation, without rigidly requiring the availability of specific localization devices and rather exploiting a variety of heterogeneous localization devices. As another example, a routing service should guarantee message delivery in very dynamic and mobile networks, without requiring manual reconfigurations, and should possibly tune quality-of-service depending on the specific needs of the user/application exploiting it. These needs induce specific requirements on what a proper autonomic and adaptive component model for autonomic communication services should be, and also forces abandoning traditional (i.e. stack layered) communication service architectures.

C. Requirements for Autonomic Communication Services

The need for communication services to fit the complexities of modern network scenarios by becoming autonomic and adaptable, calls for an underlying component model capable of satisfying a set of requirements. In particular:

- *Autonomicity.* A component model for autonomic communication services implies the capability of components (at the individual level, or at an aggregate social level, or both) to support self-preservation and self-healing of some specific functional and/or non-functional properties independently of contingencies, just like a living organism is able to maintain its internal balances [10].
- *Dynamic Adaptation to Changes.* A component model requires the capability of tolerating dynamic self-reconfiguration of components, and of their composition and interaction patterns, without requiring any a priori information and/or human intervention.
- *Situation-Awareness.* To achieve autonomic behaviour and adaptivity, a component model for autonomic communication services must be necessarily aware of what's happening around.
- *Generality.* It is expected that next-general autonomic communication services will involve several components executing on a variety of heterogeneous devices and interacting via a variety of communication technologies.
- *Scalability.* Given the possible very large scale of the target network scenarios, the component model should be based on design principles that can be

practically applicable to small systems as well as very large systems, and should promote organizing services according to patterns that exhibit scalable performances (or quality of service).

At this point of the discussion, the reader will probably already think that the requirements for such envisioned component model can simply and directly be mapped into an agent-based model, and that the ecology of autonomic communication services can be considered as a sort of complex agent society. This is true only to some extent and the rest of the paper will better unfold and analyze such an issue, keeping in mind the above mentioned requirements throughout the discussion.

III. TOWARDS A MODEL FOR AUTONOMIC COMMUNICATION SERVICE AGENTS

In this section, we claim that an agent model can be the most suitable answer to the challenging requirements of autonomic communications. Nonetheless, past agent models do not fulfill all the requirements discussed earlier and thus we stated that such a model should exhibit some peculiar features that we try to discuss in the remainder of the paper.

A. Agents as Autonomic Service Providers

Taking into account the intrinsic dynamicity and complexity of the above scenario and its requirements, it clearly emerges that autonomic communication services cannot be modeled and implemented as "passive components", like in a standard service-oriented architecture. Rather, autonomic communication services should be modeled and implemented by "active" autonomous components, exposing their service and integrating (at the component or at the system level) features of autonomicity, self-adaptation, and situation-awareness, in a scalable and general way. Accordingly, at this point, we can state that the search for a novel autonomic component model for autonomic communication services corresponds to the search for a proper "service agent" model.

In general, we envision that the nodes of an autonomic communication architecture should host some sort of *agent/service execution environment* on top of the operating system (see Figure 2), to act as a general flexible support for the execution of service agents. The execution environment should tolerate the hosting of both very simple reactive agents and of more heavy-weight "intelligent" self-adaptive agents. Furthermore, it is likely that such environment will have to host also other kinds of "artifacts", such as tuple spaces, resources, channels and so forth. The execution environment should be as *thin* as possible: it should provide only the minimal set of basic services to agents (e.g., agent creation and cloning, capability to perceive local events), so as to make it possible to run it even on small resource-constrained devices, like sensors or smart-phones. Upon the distributed set of execution environments, agents of different types can execute, reproduce themselves, and interact with each other. Whenever a specific autonomic communication service is needed, users (or other agents) can provide it, "injecting" the proper service agents in the network. Any type of communication service, from infrastructural ones to user-level ones, is realized by specific service agents deployed in the infrastructure, without any pre-defined "layering". Rather, the idea is that of an "ecology" of distributed agents, in which different species of agents, from ant-based to intelligent ones co-exist, each providing specific services either as a species or as individuals, and interacting with each other so as to gather what services they need from each other.

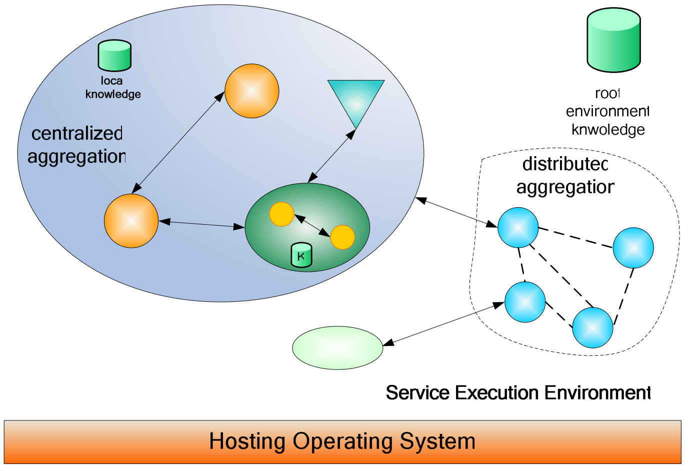

Figure 2. Aggregation of autonomic service agents

In this general scenario, satisfying the requirements of autonomicity, self-adaptability, situation-awareness, scalability, and generality, requires defining a service model and associated tools to support:

- different forms of spontaneous *self-aggregation* by service agents, to enable both multiple distributed agents to collectively and adaptively provide a distributed communication service and a service agent to properly exploit other services on need;
- some ways to *enforce control* in the ecology of service agents;
- *self-similar* forms of aggregation, capable of reproducing nearly identical structures over multiple scales, and achieving software engineering scalability;
- suitable models for the *organization of situational* information and their access by agents, promoting more informed adaptation choices by agents and advanced forms of stigmergic interactions.

These issues are analyzed in the following of this section.

B. Self-Aggregation as an Adaptation Mechanism

In an "ecology" of self-adaptive service agents executing on a very thin and bare environment, *self-aggregation* is the key mechanism to build and exploit complex communication services. Self-aggregation is clearly an autonomic adaptation process, in that it must occur on need and without direct human intervention: whenever some changes occur in the surrounding environment, some simple communication services can decide to form a coalition that can better handle the new unforeseen situation or provide an improved service.

Enabling self-aggregation in our agent model implies rethinking traditional integration architectures both from an *architectural* and a *behavioural point of view*.

Let's consider, first of all, the *architectural* viewpoint, i.e. how our service agents should be designed to support aggregation formations and to accomplish the discussed requirements of autonomic communications. A TCP socket can be seen, for instance, as a composition of layered services, e.g. the IP routing service and the Ethernet data link service. This type of composition can be defined a *containment*, because an outer component (i.e. the TCP service) encapsulates one or more inner components (i.e. the IP service) and uses their services. Every service request delivered to the outer component is forwarded to an inner one and, while negligible in many simple aggregations, the overhead introduced by this forwarding can be significant in resource-constrained devices and/or when more complex aggregations are formed. Autonomic communication services overcome the limits of layering, proposing a flattened model, in which composing service agents means rather *combining/fusing their service interfaces into a new negotiated interface*. This avoids the overhead of forwarding messages/function calls across the several layers of the aggregation.

Moreover, our flattened service model means also that agents should be allowed to participate in more than one aggregation, e.g. because their service can be shared among different clients. This requires that our service agent should be more than simply a "service provider" with a fixed published interface. We envision a new concept of *interface* that is far more flexible than classical software component interfaces. When a service agent participates in an aggregation, its interface should be updated or, better, it must expose a new interface: the "aggregate service interface". Such interface is expected to be the same in every agent participating in that aggregation and its provided operations are negotiated and constructed according to the aggregation strategy and the requirements of the new complex service. As a consequence, in our agent model service agents are considered sort of "polyhedral components", capable of exposing several interfaces as different service access points. Let's consider, for example, the case of a service agent participating in more than one aggregation: it has to dynamically choose the right interface to expose, depending on the access point from where it is accessed.

Moreover, each provided interface can even change depending on internal reorganizations of the aggregate service, e.g. due to adaptation to environmental happenings, like the failure of one aggregated service. Handling multiple and dynamic interfaces requires a sort of *interface negotiation mechanism* among service agents: in place of fixed interfaces, the interface negotiation mechanism defines and requires service agents to support universally known "introspection facilities" by which support for other services can be ascertained at runtime. Clients of a service agent use these well known services to obtain mutually agreeable interfaces.

Autonomic communication services are very often located on different network nodes and the aspect of the *physical distribution* of services should be taken into account, even though the aggregated service appears *conceptually unique*. For example, services provided by self-organizing swarm agents are usually distributed by their own nature (e.g. some localization service whose agents are distributed across many sensors in the environment), and do not respond to a central controller or supervisor; nonetheless, these agents should ideally work as a unique service, which can be invoked from many scattered access points. This leads as a consequence that our agent model should transparently support "at least" both centralized and distributed service aggregations (see Figure 2) and we said "at least", because there might be intermediate or hybrid solutions between these two extremes. *Centralized aggregations* are those where many service agents, locally available or instantiated at runtime, are combined into a new complex service. In this case, the "aggregator agent" is the new centralized access point to the service and is not physically distributed. It exposes a single communication interface to other services all over the network. *Distributed aggregation* raises more challenging design issues, because, in this case, several agents decide to join together into an aggregated service, but they still preserve their physical distribution in the network. In other words, they all agree on a common "aggregate service interface", but there is no aggregator agent exposing it; every single participant instead is considered "access point" to the aggregate service and exposes the same interface as all others.

Besides architectural design choices, self-aggregation needs effective algorithms and tools to work in dynamic and open environments, without human intervention. From a more *behavioural* standpoint, service agents are expected to support different aggregation techniques, which are an active research area of AI. Several coalition formation algorithms have been proposed for task allocation problems [16, 15] and, although we are not interested here in one particular algorithm, we state that autonomic self-aggregation will likely draw much inspiration from such research work. Therefore, each service agent in our model must include a proper *aggregation interface* (through which the agent can be involved in new aggregations, leave broken coalitions and so on) and such interface should be as much general as possible, to support a wide range of coalition formation algorithms. Finally, we must recall that autonomicity should be enforced at all levels of aggregation and this requires proper mechanisms to control/supervise the behaviour of the aggregated components. Such issues are the subject of the next Subsection.

C. *Enforcing Control for Self-management*

As already highlighted, one of the key driving principle of the autonomic communication vision is that services should be *self-managing*. The fundamental problem when trying to enable autonomicity (at all levels of service aggregation, from primitive service to complex ones) consists in *establishing some kind of control over service agents*, in order to constantly guarantee an optimal overall functionality, protect against malfunctioning parts and so forth. The IBM proposal for building autonomic components [10] is based upon the introduction of the so-called "autonomic manager" (see Figure 3), which is an intelligent software entity that monitors the activity of its managed resource, and can take corrective actions in a sort of continuous control loop. Nonetheless, control and supervision at the individual level does not guarantee an autonomic behaviour of the entire system: in an aggregation of service agents, where every member constantly monitors and regulates its own essential variables (i.e. in a local loop), it often occurs that the selfish nature of each component does not result in an optimal outcome of the aggregated service. Applied to our highly dynamic, open and distributed scenarios, the problem of enforcing control and achieving an adequate level of self-management is even trickier.

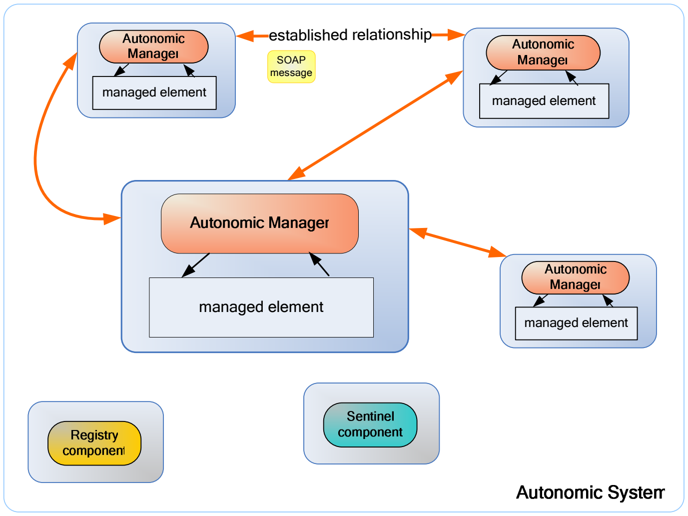

Figure 3. Autonomic components in the IBM perspective

To this purpose, some more traditional design patterns introduce a special "sentinel element", in charge of supervising the behaviour of each autonomic element and avoid dangerous or incorrect actions: many autonomic component frameworks (e.g. [11]) adopt this pattern, since they continuously monitor deployed components and influence their behaviour injecting proper "adaptation rules". These rules are interpreted by the component and translated in corrective actions on its internal parameters, leading hereafter to a modified behaviour of the same component. Other more agent-inspired solutions rely on the "cooperation capability and sociality" of autonomic managers: exchanging information with each other and orchestrating their actions, these intelligent individual controllers can ensure an *autonomic behaviour* of a composed service. All such approaches are essentially coupling traditional monitoring and resource management with artificial intelligence techniques for planning and knowledge management, as well as multi-agent systems negotiation ones.

Nevertheless, we argue that these approaches, though still feasible and valid, will prove to be increasingly unsuitable for many autonomic communication scenarios, like they have been presented so far. Autonomic communication services are expected to be pervasive and to run on even small wearable devices and, having autonomic managers logically separated from their managed services, can produce heavyweight service agents. In fact, designing each single service agent, with the rational capability to react to all possible contingencies planning proper corrective actions, may end up in a cumbersome service architecture. It can be stated that the limitations of the discussed autonomic self-management derives from its being inspired by traditional human-based management, where we usually have the controller and the controlled entity. Self-organization approaches support instead the biologically inspired idea that "a system should be able to self-manage by its own very nature and not by external intervention of other "non-self" entities" [19]. Self-organizing systems exhibit an intrinsic self-managing capability, which only indirectly depends on the behaviour of their individual little agents, but rather descends for the combination of their local interactions. Service emergence helps avoiding intelligent elements, with planning and knowledge management capabilities to react to unforeseen environmental changes, and produces simpler and more lightweight architectures. As autonomic computing design patterns have their drawbacks, current approaches to self-organization are likewise limited, e.g. because they can implement only a limited set of self-managing functionalities, but often fail in accounting the diverse and complex requirements of autonomic communications.

The agent model we are sketching in this paper should be thus as much flexible and general-purpose as possible. It should still allow both traditional autonomic managers and self-organizing approaches, but here we deem crucial to introduce also an innovative vision of self-management [19] tailored to the peculiarities of autonomic communications. Since most autonomic communication usage scenarios will be dramatically distributed, often without any clearly identifiable stakeholders, the only solution to enforce some forms of control over them, and to have the self-management features of each individual system coexist with more decentralized forms of self-management, will be that of populating the ecosystem with additional "manager components". In an environment where every single service, even the most basic one, is provided by a service agent, it is reasonable to assume that self-management should be enforced by means of some first-class elements, injected on demand into the self-organizing system. These "manager agents" will have to live inside the system and interact with other self-organizing service agents, to monitor their execution and possibly influence their emergent behaviour. This brings as a consequence that the knowledge management and planning capability, previously placed as a possibly heavyweight burden on every single component, is now "externalized" and made distributed across the various deployed manager agents. It must be pointed out that some of these ideas have been already experimented and formalized in MAS research: the idea of Electronic Institutions (EI) and *norm-aware agent societies* have been proposed as a model to specify the kinds of interactions among software agents using norms (e.g. obligations, permissions, etc.). In [2] norms are explicitly represented and managed via rules and a team of "administrative (institutional) agents" is deployed in the distributed architecture, to ensure normative positions are complied with and updated by individual agents. Experiences from this and other research on norm-based systems will be of paramount importance to formalize our "ecology-like" autonomic communications service model.

D. Robustness and Generality with Holonic Agents

Given the importance of self-aggregation in our model, the combination of primitive services into new complex ensembles must be fully *scalable*, i.e. the software design principles should be applicable to small systems, as well as very large systems, possibly made by huge numbers of heterogeneous nodes and service components. In our service agent model, all agents should at least expose a common set of basic functionalities, i.e. a "common interface", besides their specific peculiar operations. Aggregate service agents will be services in their turn and will thus have the same basic shared interface. Applying the *self-similarity* principle means that "individual components self-organize and self-aggregate so as to reproduce nearly identical structures over multiple scales" [4].

From a software engineering point of view, having the same structural and organizational principles in force at different scales facilitates the management of services: e.g. if service agent A decides to aggregate with service agent B, it can at least rely upon the shared common interface to negotiate the aggregation and agree on the new aggregated interface to expose. Self-similarity helps to achieve the key requirement of *generality*: this feature is fundamental to better handle design complexity in an environment where possibly thousands of heterogeneous software agents can be hosted. It would allow "diving" into specific sub-systems whenever necessary, without having to modify abstractions and tools to work at finer levels of granularity.

From a more architectural standpoint, self-similar structures are known to be intrinsically *robust*: it is more than desirable that the combination of some autonomic communication services brings to entities that are robust and capable of adapting to changes in the environment (e.g. a wireless link goes down, but the service will find alternative paths to deliver the message). Many biological systems exhibit such properties, thanks to their being organized in hierarchical and *self-similar* structures at different scales.

A successful agent model for autonomic communication services should therefore support self-similar aggregation and MAS research has already explored some important applications in this direction, introducing the promising idea of *holonic agents* [7, 8] (mainly applied to manufacturing scenarios). The term *holon* was originally introduced by the philosopher Arthur Koestler in order to name recursive and self-similar structures in biological and sociological entities: a lot of systems in nature can be seen as either "whole" or "part" of a larger system; for example, a human individual is on the one hand a composition of organs, consisting of cells that can be further decomposed, and on the other hand he (or she) may be part of a group which in turn is part of the human society. According to Koestler, a holon is a structure that is stable and coherent and that consists of further holons that function similarly. Koestler defines a *holarchy* as a hierarchy of self-regulating holons which function

a. as autonomous wholes in supra-ordination to their parts,
b. as dependent parts in sub-ordination to controls on higher levels,
c. in co-ordination with their local environment.

Therefore, it is clear how *self-similar service aggregations* are endowed with the properties that are intrinsic in this holarchy definition. Building holonic service compositions enables the construction of very complex systems that are efficient in the use of resources, highly resilient to disturbance (both internal and external) and adaptable to changes in their surrounding environment. Holarchies (i.e. service aggregations) are *recursive* in the sense that a holon (i.e. an agent) may itself be an entire holarchy that acts as an autonomous and cooperative unit in the first holarchy. The *stability* of holonic service aggregations stems from holons being self-reliant units, which *have a degree of independence and handle circumstances and problems on their particular level of existence* (i.e. the local execution environment of the aggregator agent), without asking higher level holons for assistance. Holons can also receive instructions from and, to a certain extent, be controlled by higher level holons. The self-reliant characteristic ensures that holons are able to survive disturbance, while the subordination to higher level holons ensures the effective operation of the larger whole.

Like holons, self-similar aggregate agents would participate in further aggregations/holarchies or would simply exist as new available services, but always as self-reliant units: hiding their internal complexity under a self-similar interface, they can react to changes in the environment and *adapt* to different situations, transparently re-organizing their internal structure.

E. Organizing Situational Data into Knowledge Networks

Another essential requirement for autonomic communication services is their capability to perceive their surrounding context and consequently adapt and improve their behaviour. Information about the context is expected to be increasingly important to enable *situation-awareness* in next generation communication services.

Nowadays, several mechanisms exist to produce situational data from the environment (e.g. intelligent sensors or monitoring mechanisms) and such knowledge is expected to become a dramatic amount in the near future. In our vision, this huge amount of information cannot be fully managed or internalized by every single service agent: it would require a significant knowledge management capability that we consider an avoidable burden in our agent model. Our basic idea is that situational data should be somehow scattered part in the environment (e.g. in a shared tuple space) and part across the different service agents. In further details, we envision that when service agents decide to form aggregations, they share their pieces of context knowledge with the other participants, forming a sort of "aggregated situational knowledge". This knowledge, scattered among aggregate agents, will be thus organized in a hierarchical fashion among all the running service agents: in a few words, one agent could own a piece of knowledge about the local context and, by joining an aggregation, it would integrate its information within the aggregated knowledge. The aggregated entity, being self-similarly part of another aggregate or of the Service Execution

Environment (see Figure 2), would perform the same integration in turn. Therefore, the global knowledge would be dynamically built by the various service agents that join and leave the system during the execution.

Moreover, situational data should be elaborated and any relationships between such information properly represented and correlated according to well-defined ontological constructs. We expect that the bulk of this sort of continuous "knowledge analysis and elaboration" phase will be performed mainly out of the service agents that will access and use it. It would be advisable to have special "knowledge manager" agents injected in the environment, in charge of properly analyzing and correlating such diffused situational data. Distributing such analysis activity among different actors helps achieving better scalability and reduces the reasoning capability that a service agent should have (we do not want a heavyweight rational service agent).

The final conceptual outcome of the above knowledge organization and analysis phases is the formation of so-called *knowledge networks*, in which all information about individual contexts are properly represented, organized and correlated, and around which semantically-enriched stigmergic interactions among individual agents can take place. Distributing such knowledge in the environment and hierarchically among agent aggregations, service agents can self-organize their activities using "cognitive stigmergy" approaches [17]. As anticipated earlier, the distributed knowledge network is expected to play the part of a *high-level intelligent and dynamic environment*, useful in particular for those self-organizing services that use the environment as a mediator for their local stigmergic interactions. *Self-adaptation* and *self-organization* would be driven by more sophisticated application-level knowledge data, other than simple pheromones value to react, and this will enable more robust and adaptive configuration patterns (e.g. the knowledge network can be used to enforce a more semantic control over a set of swarm agents). In addition, scattering context information among aggregate agents allows to make services situation-aware with different degrees of granularity: locally relevant situational data are consumed in place, while components are allowed also to reason about more global situational data, interrogating the distributed dynamic knowledge network: service components can "navigate" through the available knowledge hierarchy to attain, on demand, the degree of contextual awareness they require.

IV. CONCLUSIONS

The continuous growth in ubiquitous and mobile network connectivity, together with the increasing number of networked computational devices populating our everyday environments, call for a deep rethinking of traditional communication and service architectures. In this paper, we have focused on communication services, and have analyzed the key characteristics and features that a proper innovative component model for the effective development and deployment of autonomic (i.e., self-organizing, self-adaptive, self-healing) communication services should exhibit.

The results of our analysis can be simply summarized as follows:

- Such new component model should be general-purpose, able to enforce autonomic behaviour in both the forms of self-adaptation and self-organization, able to handle "situatedness" in complex knowledge environments, and should tolerate scalable forms of dynamic aggregation.
- Multiagent systems researches can play a major role in the definition of such component model and, more in general, in the advance of the autonomic communication research area. Nevertheless, as this paper envisioned, their scope should be limited by a clear and suitable component model, tailored to the requirements of autonomic communications. Such a model is the aim of the CASCADAS project in the future years.

V. ACKNOWLEDGEMENTS

Work supported by the European Commission within the Integrated Project CASCADAS (IST-027807) "Component-ware for Autonomic, Situation-aware Communications, And Dynamically Adaptable Services" (FET Proactive Initiative, IST-2004-2.3.4 Situated and Autonomic Communications).

REFERENCES

[1] The Autonomic Communication Forum, www.autonomic-communication-forum.org.

[2] J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodriguez and C. Sierra, "Engineering Open Environments with Electronic Institutions", in Journal on Engineering Applications of Artificial Intelligence, 18(2):191-204, 2005.

[3] I. Carreras, I. Chlamtac, H. Woesner, C. Király, "BIONETS: BIO-inspired NExt generaTion networks" in the Proceedings of the 1st IFIP Workshop on Autonomic Communications, LNCS No. 3457, Springer Verlag, 2004.

[4] S. Dill, R. Kumar, K. Mccurley, S. Rajagopalan, D. Sivakumar, A. Tomkins, "Self-Similarity in the web", ACM Tran14tions on Internet Technology 2(3):205-223, 2003.

[5] S. Dobson, "Putting meaning into the network: some semantic issues for the design of autonomic communications systems", in the Proceedings of the 1st IFIP Workshop on Autonomic Communications, LNCS No. 3457, Springer Verlag, 2004.

[6] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.

[7] K. Fisher, "Holonic multiagent systems - theory and applications", in the Proceedings of the 9th Portuguese Conference on Progress in Artificial Intelligence (EPIA-99), LNAI Volume 1695, Springer Verlag, 1999.

[8] A. Giret and V. Botti, "Holons and agents", in Journal of Intelligent Manufacturing, vol. 15, pp. 645-659, Kluwer Academic Publishers, 2004.

[9] J. Hightower, G. Borriello, "Location Systems for Ubiquitous Computing", IEEE Computer, 34(8):57-66, Aug. 2001.

[10] J. Kephart, D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, 36(1):41-50, January 2003.

[11] H. Liu, M. Parashar, "Component-based Programming Model for Autonomic Applications", in the Proceedings of the 1st International Conference on Autonomic Computing, New York, NY, USA, 2004.

[12] A. Manzalini, F. Zambonelli, “Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision”, 1st IEEE Workshop on Distributed Intelligent Systems, Prague (CZ), June 2006.

[13] A. Pentland, “Socially Aware Computation and Communication”, IEEE Computer, 38(3):63-72, March 2005.

[14] The European Commission, “The Situated and Autonomic Communications Initiative”, http://cordis.europa.eu/ist/fet/comms.htm.

[15] T. Sandholm and V. Lesser, “Coalitions Among Computationally Bounded Agents”, in Artificial Intelligence, Special Issue on Economic Principles of Multi-Agent Systems, Vol. 94, N. 1, pp. 99-137, 1997.

[16] O. Shehory, S. K. Sycara and S. Jha, “Multi-agent Coordination through Coalition Formation”, in Lecture Notes in Artificial Intelligence, n. 1365, pp. 143-154, Springer Verlag, 1997.

[17] L. Tummolini, C. Castelfranchi, A. Ricci, M. Viroli, A. Omicini, “Exhibitionists and Voyeurs do it better: A Shared Environment Approach for Flexible Coordination with Tacit Messages", Environments for MultiAgent Systems. LNAI 3374, Springer-Verlag, January 2005.

[18] R. Want, “An Introduction to RFID Technology”, IEEE Pervasive Computing, 5(1):25-33, Jan.-March 2006.

[19] F. Zambonelli, “Self-management and the Many Facets of Nonself”, in IEEE Intelligent Systems, Vol. 21, N. 2, pp. 50-56, 2006.