A Middleware for the Support of Autonomic Behaviour of Home Networking Communities

Adriano Fiorese[1,2], Paulo A. F. Simões[1], Fernando P. L. Boavida Fernandes[1]
[1]CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
[2]DCC, Department of Computer Science, University of the State of Santa Catarina - Brazil
E-mail:{fiorese,psimoes,boavida}@dei.uc.pt

***Abstract*—Nowadays more and more people connect to the Internet through some kind of home network. In addition to connecting to the Internet, user networks are being used to establish communities that produce and share content, giving strength and meaning to the expression "community networking". This is being enabled by a variety of technologies, of which the WirelessMan^{tm} technology is only an example. In order to deal with the increasing interest for and complexity of community networks, suitable management techniques must be in place, with emphasis on techniques that make networks and network services as autonomic as possible. In this work we present a proposal for a middleware with the aim of easing the development of autonomic applications and services, that can make up a home network community.**

I. Introduction

We are currently experiencing a rapid increase in the number of networking communities. People who share some kind of interests join with the objective of making services and content sharing a reality. This phenomenon is being enabled and fostered by the widespread availability of a variety of information and communications technologies. This set of technological resources and people sharing interests is generally known as "network community". When the network community is mainly made up of home networks, it is called "home network community". Home networking communities scale according to the number of people who join them and may evolve to the point of not only producing content but also generating and supporting an economic cycle. The management of such communities is a very hard task due, on one side, to the complexity of the system or systems that support the community overlay and, on the other, to the dynamicity of the content services being supported, which may be a handicap to the use of community networks. Hence, a possible solution is to deploy autonomic behaviour in the community, so that it can manage itself.

New technologies for network communication like WirelessMan^{tm} (see IEEE 802.16), known as WiMax, are starting to be deployed in order to overcome the last miles between end-user communities and Internet service providers. The deployment of this technology with quality of service guarantees, as being developed in the scope of the FP6 IST 034622 WEIRD project, will bring new development opportunities including content production and distribution to/from remote user communities.

The autonomic behaviour in a system is related to its capacity of self-management. Some authors among them [2] and [3], identify four fundamental characteristics of autonomic systems: self-configuration, self-healing, self-optimization and self-protection. In a network community this means capacity to optimize the use of the shared resources and to configure and expand itself based on common characteristics shared by its members. Moreover, at the same time, a network community should use mechanisms to protect itself from outside and inside attacks or from information misuse, and to avoid the disruption of the community as a consequence of any kind of fault that may occur.

The communities can be built on a series of software technologies, like the service agents approach explained in [5]. In this article the author presented a hierarchic architecture abstraction, based on Smart Environments, consisting of an environment leader module and of environment members, which are service agents implemented using the Jini framework (see http://www.sun.com/software/jini/), with mobility capabilities, i.e., mobile service agents.

As the current work is in its beginning, in this text we will present some ideas on how to address the introduction of autonomic behaviour in services which will be used in home networking communities. One of these services can be useful in keeping every single node in the home community updated with some version of codec used to produce video content, for instance. This kind of service can be classified as a self-configuration service, as it provides the capacity to determine which nodes in the community are not updated, access the codec's repository and deploy de appropriated codec, installing it without manual intervention. With examples such as this one in mind, the proposal that is described below aims to develop a middleware that eases the construction of autonomic network communities, content services and applications.

II. Related Work

Among others Xiangdong et al. already proposed a middleware for the support of autonomic computing, named AUTONOMIA [4]. AUTONOMIA is a framework for developing applications or systems that can have self-healing and self-configuring properties. This framework uses self-healing and self-optimizing handlers that monitor the applications with the purpose of handling faults and optimising performance. The handlers are, essentially, mobile agents based on Java/Jini. They are - through the Autonomic Middleware

Services, AMS - the core of autonomic management services in AUTONOMIA. The user can develop an application using the Application Management Editor, AME, the Component Repository, CR, and Resource Repository, RR. To do that the user selects the components from a well-defined library of components that are registered in CR and selects how those components are interconnected as well. At this point, AME is used for the creation of a template that is supposed to specify all the management and control attributes associated with the application. Further, the application attributes will be monitored by the self-healing handlers and self-optimizing handlers, both of which are mobile agents supported by AMS.

III. Design guidelines for the proposed middleware

As a starting point of the proposed research, we will explore and extend some ideas from autonomic personal computing [1] in order to develop a middleware for the development of autonomic applications. Our hypothesis is that a group of applications presenting separate autonomic behaviour, developed to form a home network community, can deliver autonomic behaviour to the entire home network community by generalization. To accomplish this, the proposed middleware has to encapsulate some or all of the basic characteristics of self-CHOP, according to [1], in autonomic elements, AE, composed of one or more classes, which will be used by the developers to build their own applications by derivation (object oriented programming paradigm). AEs will be responsible for providing the communication skills between the autonomic elements that make up an application in a home network community scenario. Additionally, AEs will be able to monitor themselves as well other AEs, in order to provide self-healing capabilities. To do this, AEs will resort to local data persistence for fault recovery, and to some election strategy to determine which AE will perform some basic distribution of tasks and will provide high availability information to the whole application. With all of these characteristics, developers will be free to concentrate on the functionality of content applications.

Communication between AEs is crucial, being at the basis of self-healing characteristics, as it allows the exchange management information. Key management information encompasses status-monitoring information in the case of recovering a broken down AE with the start of another AE in another location in the network, or simply transferring the execution to the same AE previously replicated at another location. Both approaches will be developed, studied and compared.

In the case of self-configuration, for instance, the group of AEs can elect one AE to be the group's manager so that, having in mind self-optimizing, it can coordinate load balancing. This can be accomplished using idle AE replicas used for self-healing.

The approach being taken in the proposed middleware is considerably different from previous work in general and, specifically, from the AUTONOMIA approach. In the latter case, mobile agent functionality is explored in order to support mobile service agents, which implement the services in a home network community [5]. AUTONOMIA thus uses mobile agents as autonomic middleware service providers, and offers a collection of ready-made components to application developers.

On the other hand, the proposed approach is quite different, as it is based on the concept of application virtualisation. Although one application looks like one single block, it is composed of as many AEs as necessary to deploy the service in the home network community. The location of the various AEs that make up the application are transparent to the users of the application and, in fact, this location is irrelevant because the grouped AEs behave as a standalone application. This development approach is, thus, quite similar to the approach taken in grid application scenarios, where several distributed components cooperate in order to implement a globally available application.

IV. Conclusion

Autonomic computing, autonomous content distribution and integrated content service infrastructures are some of the most important areas of research for the next few years. Although partial solutions to some of the issues posed by content networks and services are beginning to emerge, application developers are still faced with numerous low-level details. The work proposed in this paper has the objective of easing the task of autonomic application developers, by using a middleware that provides suitable abstractions. The key design guidelines of such middleware were presented, as well as its distinguishing factors in relation to previous work. Future work will address the development of the proposed middleware and the assessment of its functionality and capabilities, both in controlled lab environment and in real world scenarios, such as remote communities, medical applications, environment monitoring and fire control and prevention.

Acknowledgment

This work was partially funded by IST FP6 0384239 - CONTENT Network of Excellence.

References

[1] D. F. Bantz, C. Bisdikian, D. Challener, J. P. Karidis, S. Mastrianni, A. Mohindra, D. G. Shea, and M. Vanover, "Autonomic personal computing," *IBM Systems Journal*, vol. 42, no. 1, pp. 165–176, 2003.

[2] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM Systems Journal*, vol. 42, no. 1, pp. 5–18, 2003.

[3] J. O. Kephart, "Research challenges of autonomic computing," in *ICSE '05: Proceedings of the 27th international conference on Software engineering*, ACM: Association for Computing Machinery and SIGSOFT: ACM Special Interest Group on Software Engineering. ACM Press, 2005, pp. 15–20.

[4] D. Xiangdong, S. Hariri, X. Lizhi, C. Huoping, Z. Ming, S. Pavuluri, and S. Rao, "Autonomia: an autonomic computing environment," in *IEEE International Performance, Computing, and Communications Conference, (IPCCC 2003)*, 2003, pp. 61–68, iNSPEC Accession Number: 7755808.

[5] F.-C. Yang, "Design and implement of the home networking service agent federation using open service gateway," in *KIMAS 2003. International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2003.* Boston, USA: IEEE Computer Society, 2003, pp. 628–633, iNSPEC Accession Number: 7906840. [Online]. Available: http://ieeexplore.ieee.org/iel5/8803/27847/01245112.pdf