Context Dissemination for Autonomic Communication Systems

Nadeem Akhtar, Klaus Moessner, and Ralf Kernchen

Centre for Communication Systems Research,
University of Surrey, Guildford, UK
{n.akhtar, k.moessner, r.kernchen}@surrey.ac.uk

Abstract. Autonomic Communication is a new communication paradigm that has been proposed as a way to design new self-organizing, self-healing, self-optimizing, self-protecting and evolvable networks. The motivation comes from the problems created by the unstructured and haphazard growth of the Internet. Among the many guiding principles of Autonomic Communication is context-awareness. In this paper, we discuss architecture for context dissemination in Autonomic networks based on the Autonomous Decentralized Community Communication for information dissemination.

1 Introduction

The rapid and often chaotic growth that the Internet has seen over the past few years has resulted in an extremely complex network. Furthermore, the introduction of newer communication technologies, services and applications is leading to a growing patchwork of interconnected networks. Therefore, the management of the increasingly unwieldy Internet is becoming extremely difficult with each passing day. It is in this context that the Autonomic Communication paradigm has been proposed that aims to create self-organizing, self-managing and context-aware autonomous networks in order to meet the diverse demands and challenges confronting the Internet and to allow for a scaleable and manageable growth.

Context-awareness is an important property of Autonomic systems [1]. Research in this area is rather diverse and broad because the notion of context has many different connotations and the nature of context varies a lot depending on the application scenario. According to Moran and Dorish [2], context refers to physical and social situation in which computational devices are embedded. Anind Dey *et al* [3] define context as any information that can be used to characterize the situation of an entity where an entity is a person, place, or object that is relevant to the interaction between a user and an application. Most of the earlier work in the area of context awareness has focused on human-machine interactions. In contrast, Autonomic systems can also be deployed for business-business, human-human and machine-machine communications. Therefore, the notion of context acquires a much broader meaning.

For Autonomic systems, context needs to be considered as a dynamic process in which context is generated as a consequence of continuous interactions between

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/ 978-3-540-32993-0_29

I. Stavrakakis and M. Smirnov (Eds.): WAC 2005, LNCS 3854, pp. 237 – 242, 2006.

users, networks, network elements and the physical environment itself. This gives rise to the notion of *context state* which can be described as the instantaneous view of the context. The context state consists of a set of information elements corresponding to the parameter set that characterizes the system under consideration. Depending on the specific application that requires context awareness, whole context state may not be useful or relevant and only a subset of the state maybe needed. Therefore, the context sub states are constructed out of the overall 'global' state. The lowest level of granularity is the so-called feature context that corresponds to individual information elements. Fig. 1 illustrates how the context hierarchy is organized.

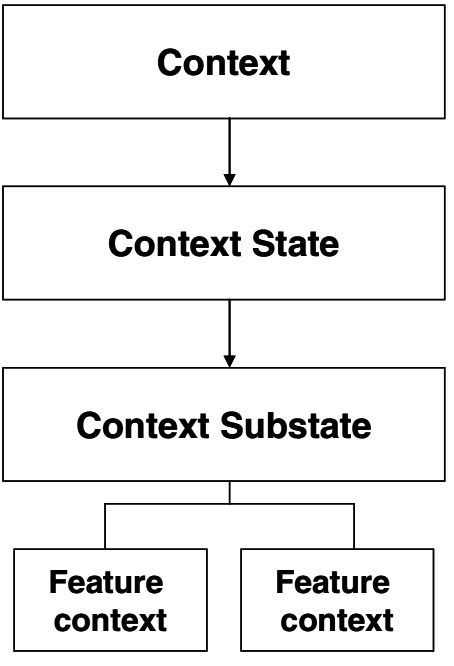

Fig. 1. Context has to be regarded as a function over time, hence it can be seen in different states and sub states which again can be described as context features forming a "context hierarchy"

Considering existing context aware systems and applying them to Autonomic Communications puts a set of requirements on the structure of such an approach. The extensible distribution of the network has to be taken into account. Further the characteristics of such a system have to be researched in general. We identified three main challenges in designing context aware systems in this area.

Some sort of context representation, often referred to context abstraction [4], is necessary in order to provide a common language for communication between entities of such a system. For Autonomic Communication Systems in particular this has to support and follow the distributed character of these networks. Apart from common characteristics for context-aware systems the distributed context representation has to consider certain network parameters and possible dependencies and relations between

different entities as well as relevance of context items for certain levels of context based decision.

Another important factor for the successful decision process is the collection of the context itself on the very deepest information layer: the entity itself. This data has to be reprocessed in order to distinguish between network or entity-level relevant information. This needs to be done to know which information has to be sent to higher level decision units in the network or can be kept in the entity itself.

Context dissemination is the third part of the design problem. A distributed context dissemination mechanism is required for Autonomic systems. Furthermore, for dynamic network, function and service composition/decomposition, the mechanism should be fast and efficient. Security is also a very important component of the dissemination mechanism.

Each of these needs to be researched in detail with reference to Autonomic systems. In this paper, we focus on the context dissemination problem. In the following, we consider the Autonomic Decentralised Community Communication and investigate its suitability as an efficient and scalable context dissemination solution for Autonomic systems.

2 Context Dissemination

Context dissemination is an important component of the overall context awareness design. A distributed, efficient, secure and scalable architecture needs to be designed for this purpose. In the following, we present an overview of information dissemination architecture for community communication and then investigate its suitability to the requirements of context dissemination in an Autonomous Communication System.

2.1 Autonomous Decentralized Community Communication

The Autonomous Decentralized Community Communication (ADCC) system proposed in [5] is designed to provide an infrastructure for information dissemination in order to help end-user groups communicate and share information efficiently. This system is based on a decentralized architecture and application-level multicast is used to distribute information from a particular source of information to all the other members of the community. The ADCC system was originally developed to meet the growing demand for real-time content delivery in the Internet and aimed to provide a scalable dissemination infrastructure. We propose to use it for context dissemination between network elements of an Autonomic Communication system. In the following, we first provide an overview of ADCC and then describe how it can be used for dissemination of context in an AC system.

2.2 ADCC Overview

The basic idea of the ADCC system is to enable members of a ‘user’ community to exchange information that is of ‘interest’ to all the members while ensuring their autonomy at the same time. In ADCC, a community consists of members that may

have individual objectives but share common interests and have similar information requirements. No distinction is made between senders and receivers of information and thus, any member can be a source as well as sink with respect to information flow. The community members organize themselves into a logical network. To form such a network, member nodes maintain information about immediate neighbors in a table and share this information with other nodes. Nodes autonomously decide to join or leave the community based on their own requirements and each member has to carry out the same set of network duties.

Since there is no central co-coordinating entity in ADCC, the responsibility of construction and maintenance of the community is shared by all the members. This self-organized community is created with three main objectives. Firstly, the resulting network must support efficient broadcast. Secondly, the traffic is distributed evenly to avoid hotspots. Finally, the network must provide redundancy so that the effect of node failures can be mitigated.

In the ADCC system, the nodes that constitute a community network are organized in a regular 2-dimensional graph: G = {V, E}, where V is the set of nodes and E is the set of edges. The graph G consists of a set of *n* Hamiltonian Cycles (HCs) that are all edge-disjoint with each node having *2n* neighbors. Each such cycle connects all the nodes in the graph and each node is traversed only once. The advantage of HC lies in localized impact of a node joining or leaving of a node.

When a node, say X, wishes to join the community, it first has to discover at least one existing member. Node X sends a join request to the discovered node Y. The latter has to find out the *2n* neighbors so that the former can join the *n* Hamiltonian Cycles. The join request is forwarded by node Y to all nodes that are within $O(log_{2d}M)$ distance, where M is the current number of nodes in the network. The nodes that receive this message then decide whether to accept the request or not. A repair mechanism is in place in case there is a node failure. Keep-alive messages are used to keep the neighbor information fresh and when a node failure is detected, the fault tolerance algorithm comes into play to repair the network.

Communication within the network is multilateral. When a node receives new information that needs to be shared with other members, it is sent to the neighbor nodes which then send it to their own neighbors while ensuring that it is not sent to the sender itself. This process is repeated until every member of the community has received the information. ADCC uses a hybrid pull-push model as well as a request-reply-all model. In the first case, when a member has new information to share, it forwards it to the neighbor nodes as described above. In the second case, when a member wants to find a specific piece of information, it sends a request to neighbor nodes. If a particular neighbor does not have the desired information, it forwards the request to its own neighbors but if the neighbor does have the information, it replies to the sender with its results. This is disseminated to all members of the community, thus updating the whole community.

The performance of the ADCC system has been compared with traditional techniques such as sequential unicast and peer-to-peer methods. It has been shown that the mean communication cost of the ADCC approach is much lower compared to the other two while its mean delay is significantly less than the unicast approach. Furthermore, the cost of community construction and maintenance increases

logarithmically as the number of member nodes increases. Results also indicate that the load on the physical links connecting the nodes in the network is distributed more evenly across the network.

2.3 ADCC for Context Dissemination

The ADCC system described above has been designed for information dissemination between members of a community in a co-operative, efficient and scalable manner. In an Autonomic Communication system, network elements are inherently members of one or more groups on the basis of physical or logical proximity, similar functionality, need for same set of configuration data etc [1]. The notion of *entity-group* has been mentioned in the context of Autonomous Communication. It refers to a type of group communication where network elements can join and leave a group based on a set of *membership rules.* Within a group, the behavior of an entity is dictated by its *group-behavior* definition. This results in a programmable and controllable group. Thus, we see that there are similarities between the idea of a community in ADCC and the notion of group in AC. Furthermore, no central controller is required to create a community and nodes can join and leave the network as and when they wish. Furthermore, the network heals itself when a node leaves the network or when one or more nodes fail. Thus, the ADCC community network is self-organizing and self-healing thereby making it an attractive choice from the AC point of view.

In an AC network, context could mean many different things. It may include information about the networking environment, the physical environment itself and even rules and policies could be treated as context. Thus, it is very important that these different types of information are described in such a way that it is easy to distribute them to interested parties. In ADCC, a *content code* is used to describe the information that is being sent out. Also, a *characterized code* is used to indicate further details of the content. This approach is well-suited for the purpose of context dissemination in the AC network.

ADCC implements a multilateral communication between community members for information flow using the bilateral links between members. Once again, it does not require any central source of information and any member can send information and all members are receivers. In the AC system, many scenarios are possible. For example, there maybe a controller node that is responsible for information collection and dissemination. Alternatively, these responsibilities maybe distributed throughout the network. Finally, it may not be required to send all new information to all the group members. The ADCC system is flexible to cater to these different requirements.

Network elements may require regular context updates as well as instant updates based on the type of context. In some scenarios, a specific type of information maybe requested by a particular element. In the ADCC system, the two communication protocols used are *hybrid push-pull* and *request-reply-all*. They could be used for context dissemination as well. In the second model, when a node replies to an information request from another node, the reply with the desired data is sent to all members. This is done to avoid further requests for the same data. While this works very well when all the members have more or less the same information requirements but it may not be efficient when this is not the case. Thus, there is a trade-off which needs to be studied further for the communication dissemination scenario.

3 Conclusion

Context awareness is an important requirement of AC systems. Context dissemination is an important component of a context-aware system. In this paper, we have reviewed the Autonomous Decentralized Community Communication system and discussed its suitability as the basis of context dissemination architecture in AC network. The ADCC was primarily designed to provide an efficient and scalable infrastructure for data dissemination on the web. It supports cooperative information exchange in a loosely-connected network consisting of members with similar information requirements. We observe that ADCC is a suitable choice for context dissemination. The community network in ADCC is self-organizing and self-healing which is extremely important from the AC perspective. Furthermore, the way content is coded in ADCC can serve as the starting point for describing context and addressing it to the right destinations. The communication model in ADCC is flexible for request-based context delivery as well as periodic and instantaneous context updates. Further work is required for a more thorough and detailed investigation of the ADCC with respect to context dissemination, especially the analysis of communication models as well the coding of context information.

References

1. Smirnow, M.: Autonomic Communication: Research Agenda for a New Communication Paradigm. Fraunhofer FOKUS White Paper (2004)
2. Moran, T.P., Dorish, P.: Introduction to the Special Issue on Context-Aware Computing. Special Issue of Human-Computer Interaction, Volume 16 (2001) 87-95
3. Dey, A.K., et al: Understanding and Using Context. Personal and Ubiquitous Computing Journal, Volume 5 (1), (2001) 4-7
4. Dey, A.K., Abowd, G.D, and Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, vol. 16, (2001) 97-166
5. Ragab, K., et al: Autonomous Decentralized Community Communication for Information Dissemination. IEEE Internet Computing, Vol.8, No.3, (2004) 29-36