Enhancing Service Discovery in ad hoc communications

Mohamed Khedr, Ahmed Karmouch
Multimedia and Mobile Agent research Lab,
School of Information Technology and Engineering, University of Ottawa
161 Louis Pasteur St. Ottawa, ON, K1N 6N5 Canada
mkhedr, karmouch@site.uottawa.ca

Abstract-- **The paper describes the Context based service discovery protocol (CBSDP), a service discovery protocol build over the Ad hoc Context aware network architecture (ACAN), it is designed to solve the problem of discovering services in a highly dynamic ad hoc communications.**

CBSDP is a distributed system that provides the user with the ability to obtain services and information according to the context (location, time, situation, users, etc), it will automatically detects the available services through the existence sensors and context processing agents.

The paper starts with a discussion of the traditional services discovery concepts, then the ACAN architecture. The complete design of the context based service discovery protocol is presented with the messaging and cooperation between the different agents in the environment. A service ontology build using Protoge2000 is used as the common language for understanding the different services available in the environment. The paper concludes with the implementation of the protocol over the 802.11 Wavelan network

I. INTRODUCTION

With the growth of the Internet, networking and mobile devices, services are showing up everywhere in the user's life, at work, at home or when traveling. This brings up problems like unfamiliar user interfaces, unavailable services or unable to find the service specially when a user changing terminals and network access points or in ad hoc communications.

This task is addressed by different organizations like the like SLP (Service Location Protocol), Jini, UPnP (Universal Plug and Play), and Salutation. In a service discovery environment, services advertise themselves, supplying details about their capabilities and information, and give users the possibility of finding the necessary information to search and browse for services, choose the right service (with desired characteristics), and use this service.

These current service discovery protocols have some disadvantages in common, they cannot enable services in a spontaneous networking environment, and they cannot provide service initial configuration and no service composition.

Our proposed protocol, on the other hand, takes in consideration these requirements in its deign. It was implemented over ACAN system that uses the context captured by sensors as the main key for setting up a network, obtaining information about users in the environment and provides useful information to the users like the available services.

ACAN architecture consists of three layers: the mobility layer, the ad hoc active network layer and the ad hoc application layer. The ad hoc active network layer represents the configuration and connectivity of the nodes in the network, while the ad hoc application layer consists of a

The CBSDP overcomes the problem of enabling services in spontaneous environment through the use of the sensors that captures the available devices and a context processing agent that process the sensor data, interpret it to context specifications and deduce the available services. A directory agent will use these discovered services to build a directory that users can consult when in need of a service.

The problem of service initialization is solved through two processes, the first one is the use of the ACAN system that has the ability of self configuring an ad hoc network and the second is through the use of a service agent that represents each service in the environment and works on its behalf. The last problem of service discovery is solved through the implementation of a composite service ontology that acts as a common language and as a way of composing different services that has direct relation to each other.

Following this introduction to our work we will present an overview of ACAN in section 2. We will provide the complete design of the CBSDP in section 3, the messaging model between the different agents in the system. and the Service Ontology design using the Protoge2000. related work will be presented in section 4, conclusion and future work in section 5.

II. ACAN architecture

In the last couple of years, a number of research projects that deals with context awareness and service discovery have emerged. Among them are the context aware tool kit in Georgia Tech[1], the Ninja project[2] and the Centaurus project[3]. Unlike our model each of the above projects dealt either with context or service discovery individually.

Flexibility, scalability and adaptability are the main characteristics of the ACAN architecture. The layered architecture of the ACAN provides the flexibility required in the system by distributing the tasks and functionalities on the different layers of the model, each layer has certain functions to fulfill, hides the processing details from other layers and only provides abstraction of the results to the other layers.

Adaptation is achieved by the context aware mechanism embedded in the system where tasks performed by the ACAN are constant and specific but the operations and functionalities that provide these tasks are dynamic, active and change according to the context fed to the system by the sensors [4].

This leads us to define context and context awareness; we take their definition from [1].

Context: is any information that can be used to characterize the situation of an entity where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application..

Context awareness: is the ability to use context information. A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current Context in use.

Protocols designed to be used in the ACAN architecture are extensible, distributed and light weighted which are the main requirements of scalability, ACAN can add or remove services dynamically and accepts new users, suspend or remove existing users and provide the network parameters in an efficient dynamic way.

The ACAN architecture consists of three layers: the mobility layer, the ad hoc active network layer and the ad hoc application layer as shown in figure 1.

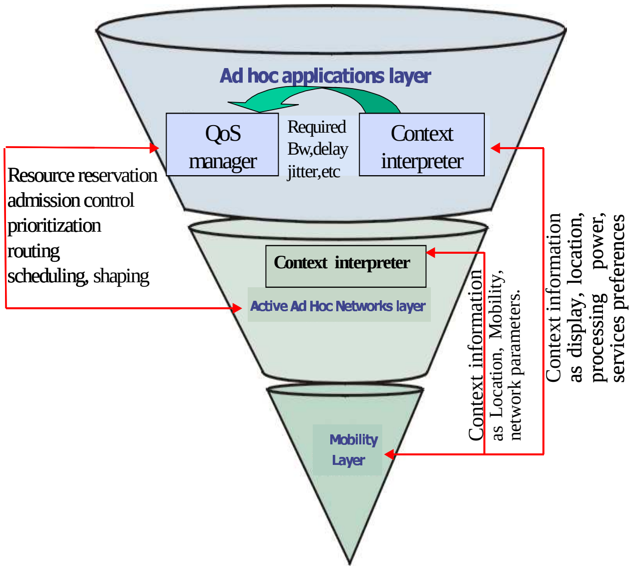

Figure 1. ACAN Layer Architecture

The mobility layer is network independent, *p*hysical sensors reside in this layer gathering information about the current environment and users in the system. The ad hoc active network layer represents the connectivity and reachability of the nodes in the environment. According to the context provided from the sensors in the mobility layer, figure (2), it is responsible for the auto configuration of network such as addressing scheme, routing technique, QoS levels and mobility management. The Ad hoc application layer benefits from the context gathered from the mobility layer, figure (2), and from the connectivity established by the ad hoc active network layer to deploy application spontaneously, discover services and users in the environment and adapt its requirements according to the current situation as shown in figure 2.

Our new protocols are designed to greatly simplify the tasks of creating and maintaining context-aware systems by distributing the weight of context-aware computing onto the ad hoc application layer and active ad hoc network layer. Using this distributed model ACAN infrastructures could make it easier to develop robust applications even on a diverse and constantly changing set of devices and sensors.

Our new protocol stack, figure 3, is needed to provide a common communication and interaction model for service discovery, user presence, mobility, auto configuration of the ad hoc network, setting security levels, setting QoS in the network and setting service level of agreements between the ad hoc network and the other domains.

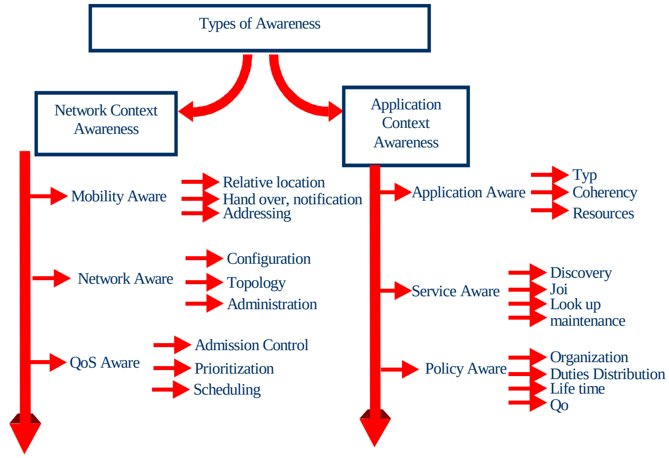

Figure 2. Types of context

The mobility layer, consists of the underlying physical layer and the mobility aware protocol, in general it will be a heterogeneous environment consisting of different communication media. These media will provide the communication infrastructure upon which the ad hoc network and the associated application will be implanted over it.

The mobility aware protocol will provide the mechanism for supporting mobile users and mobile nodes with requirements like power saving, smooth handoff, registering and proximity.

The Ad hoc active network consists of network auto configuration protocol that is needed to set up the configuration of the network automatically without the help from administrators. It will be responsible for providing addressing scheme to all the nodes in the system, name resolution algorithms, network QoS parameters and management. Network based sensors will provide the context information needed for ongoing session in the conference.

The ad hoc application layer contains the security aware protocol, the user QoS protocol, SIP and the context based service discovery protocol that will be discussed in the coming section.

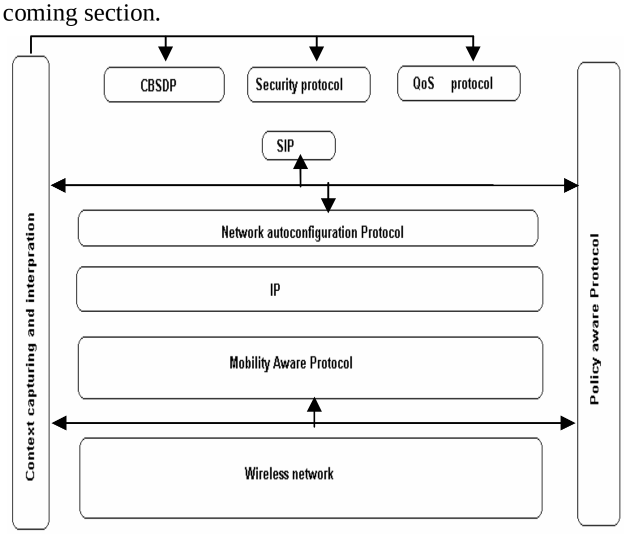

Figure 3. ACAN protocol stack

As seen from figure 3, a policy aware protocol is needed that is vertically with other protocols, policies describe the rules that must or may be used by the other protocols in order to force a certain requirement or achieve a certain goal. Policy aware protocol will provide the rules for user policies, rules for the network layer and rules for context capturing and interpretation.

III. Context Based Service discovery Protocol

Services are referred to devices that provide certain functionalities such as a printer, projector , a network device or referred to applications like a web page , a power point presentation.

The Context Based Service Discovery Protocol is build on top of the ACAN architecture and has the goal of offering the services and associating it to the users according to their current context and the system context.

CBSDP extends the existing work in service discovery by:

- Discovering, maintaining and associating services according to context in an ad hoc application that require auto service discovery.
- Providing a common ontology for the different services that behaves as a basic communication between the agents in the environment.
- Providing QoS to users.

1 CBSDP model design

The CBSDP is composed of five main components: sensors, context agents, service agents, directory agents and user agents.

The CBSDP has sensors embedded in the environment and with the responsibility of acquiring the surrounding devices and services.

This acquired context will be provided to a context agent that will interpret it to extract the context specifications as location, time, users, values, etc.

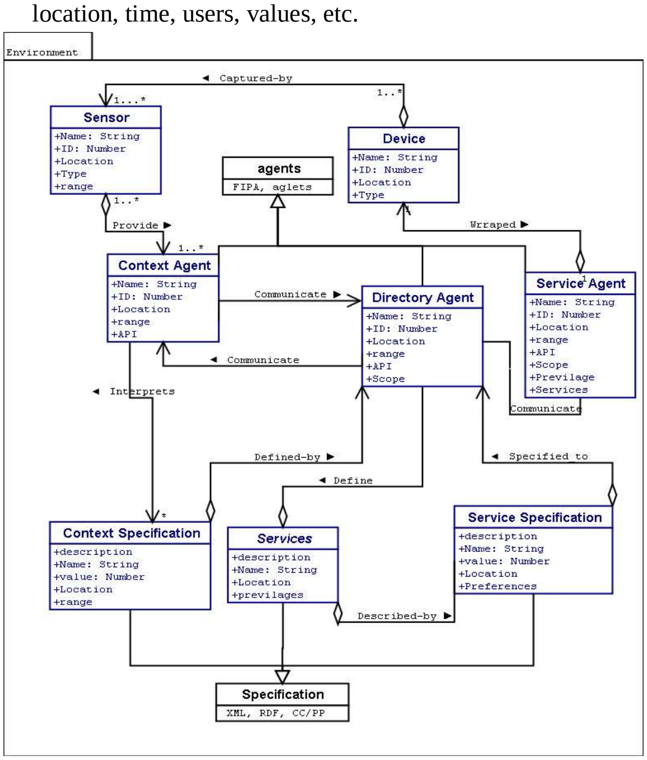

figure 4 CBSDP model

A directory agent will use the interpreted context to build a directory of the available services and the associated services attributes along with the service agent representing each of these services. Figure (4) summarizes the model of the CBSDP.

The CBSDP consists of three phases: the discovery phase, the associating phase and the maintaining phase.

In the discovery phase the sensors will be capture the available devices presenting services and resources needed by the application or the user. the context agent will communicate with the sensors using a group address assigned to the sensors during the setup of the ACAN network. The sensors will aggregate the captured context and deliver them to the context agent through the messaging sequence shown in figure 5

The context agent then uses the service ontology to interpret

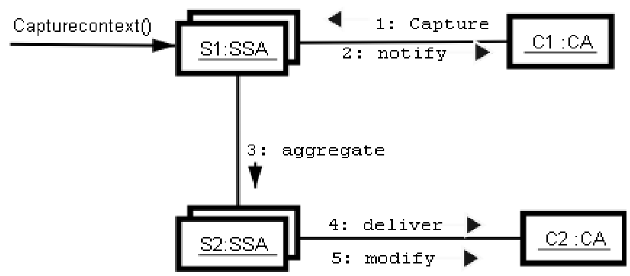

Figure 5. message sequence during discovery phase

the context information into meaningful information. The context agent may use a directory agent in case of a large network or it may be sufficient ot use only the service agent in smaller networks, this is decided based on the configuration of the ACAN network which is not discussed in this paper. The Directory agent upon receiving the information form the directory agent and with the help of the service ontology will define the different services in the environment and the attributes associated with each service along with the service agent representing each service as illustrated in figure 6

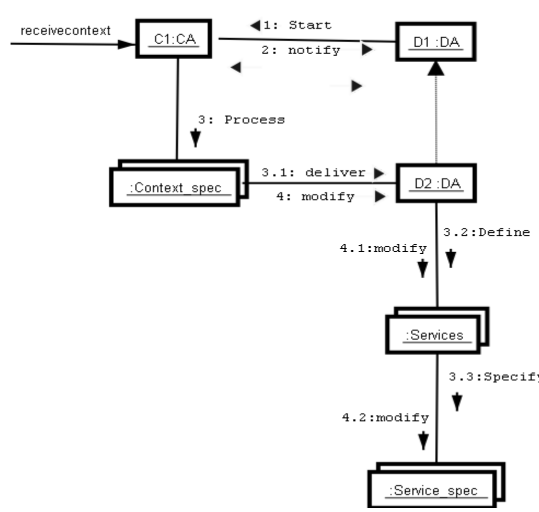

Figure 6. message sequence during discovery phase

In the association phase, when a user or an application request a service from the service agent or from the directory agent, the directory agent will lookup this service and associate it to the requested user.
The maintain phase is the dynamic changing in the discovered services according to the captured context, determined by the sensors and the service agents and informed to the directory and context agent with the modify, error message in figure 5, 6 and 7

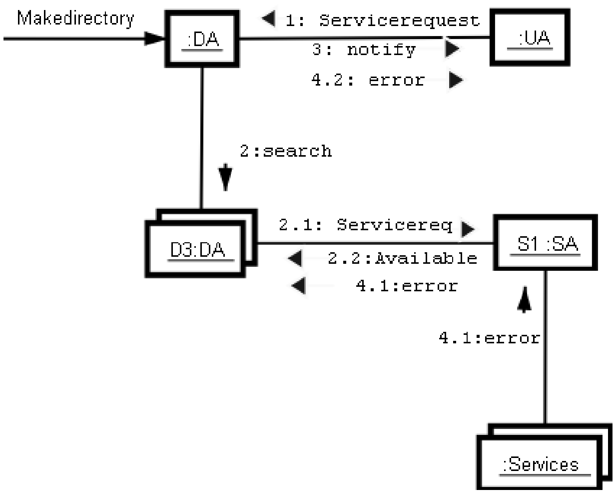

Figure 7. message sequence during association phase

2 Service Ontology

Rather than using user interfaces and/or different diverged attributes for defining available service, the CBSDP uses Ontology as a common vocabulary for entities to share information in a domain. Service ontology is designed for sharing a common understanding of the structure of information among users and the agents in the CBSDP protocol. The service ontology was designed using Protégé [5] developed in Stanford university
It is a hierarchy of classes the represents the general services that could be found in the environment and subclasses representing the services more specifically. Each class contains slots that represent the attributes of the services. It is a extensible ontology, meaning more services or attributes can be easily added to the ontology to increase the knowledge of the agents with the system.
The service ontology mainly consists of eight classes presenting services like audio, video, context, printing, quality of service and networking. Each of these classes is further subclasses to specific services like speakers, projectors, physical and software sensors, etc.
For each service a list of most common attributes were defined in a an attribute-value tuple. These classes are transformed to an RDF file used by the agents for Service descriptions and queries leveraging the flexibility and semantic-rich content of this self-describing file format.

IV. Related Work

The Cooltown project by HP labs[6] provides services through web pages and can be accessed via the web page. The Cooltown try to provide the services directly to the users while the CBSDP aim is first to define a common understanding of the services and then to provide the services to the users in an automatic way according to Context.
INS [7] form an application-level overlay network to exchange service description and construct a local cache based on these advertisements. Services are known to other clients by service dissemination through out the network.
Services periodically advertise their intentional names to the system to describe what they provide.
JINI [8] provides a framework for spontaneous distributed computing over java's RMI. JINI does not address how resources discovery will work in a dynamic environment or when service fails.
Berkeley secure service discovery [2] extends the concept with secure authenticated communication .
CBSDP differs from JINI and SDS that it handles dynamisms and failure through the service updating using captured sensors. While it is more applicable for mobile and ad hoc environments than the INS architectures it does not need the periodic advertisement or the service dissemination
The main difference between the context shadow project[10] and our system is that the context shadow uses the blackboard architecture where entities are represented by context servers while CBSDP is build over the ACAN benefiting from the auto configuration and the extensibility of the system.

V. Conclusion

In this paper we described the need for a context based service discovery where applications discover and use existing services according to the current context. Our design goals were scalability, interoperability and simplicity. We presented the design, the communication model and the service ontology of the CBSDP.
Also we showed that available service location protocol are not sufficient to enable network services in a spontaneous networking environment which requires mechanisms to decide which nodes should host services to determine how a service's initial configuration and databases are initialized and to ensure a service's availability and database consistency in face of partitions and merges [9]. All these problems are solved in our model by the simple context based model of the service discovery protocol. As a conclusion ACAN simplifies the implementation of an ad hoc network in a flexible, adaptable and scalable manner.
Currently we are simulating the system using PtolemyII. As future work we will also enhance the QoS protocols like the DiffServ and the MPLS to work with the ACAN architecture. Intensive experiments are underway to validate the ACAN architecture and the associated protocols.

References:

1. Dey and G. Abowd. "Towards a Better Understanding of Context and Context-Awareness", 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999.
1. S. Czerwiniski, et al, " An architecture for a secure service discovery", ACM/IEEE, Mobicom'99.

2. L. Kagal, et al, " Centaurus: a Framework for intelligent services in a mobile environment", IWSAWC, April 2001.
3. M. Khedr, A. Karmouch, " ACAN-Ad Hoc Context Aware Network", the Canadian Conference of Electrical and Computer Engineering, CCECE'02, May 2002.
4. http://protégé.standford.edu
5. Caswell.D., Debaty,P. "Creating Web representations for places",2nd international symposium on handheld and ubiquitous computing, HUC2K 2000.
6. William W. et al, " The design and implementation of an intentional naming system", the 17th ACM symposium on operating system principles, SOSP'99.
7. http://java.sun.com/products/jini.
8. L. Feeney et al, "Spontaneous networking: an application–oriented approach to Ad Hoc networking", IEEE Communications magazine, June 2001.
9. Patrik Werle, et al, "A Ubiquitous Service Environment with Active Documents for Teamwork Support**",** Third International Conference Atlanta, Georgia, USA, Sept. 2001