A Programmable Context-aware Voice Service

Kerry Jean and Alex Galis
University College London {kjean, agalis@ee.ucl.ac.uk}

Abstract: This paper presents a programmable context-aware voice (CAV) service, which enables VoIP services to cope in crisis situations. To prevent the upsurge in traffic normally associated with a crisis from degrading the quality of service for privileged users like emergency workers, the CAV service prevents non-privileged users from accessing the network during a crisis. This service is built upon the CONTEXT platform, an innovative middleware designed for the creation, deployment and management of context-aware services. The platform consists of a programmable layer, a context-aware service engine and a policy-based service layer. The voice services in the CAV are provided by a session initiation protocol (SIP) platform called Siptrex. The result is an easily customised, flexible and scalable context-ware service, which suppresses, non-essential traffic during crisis situations allowing greater bandwidth for essential traffic.

1 Introduction

In recent years there has been a huge increase in the use of IP networks to carry voice traffic. These VoIP networks can have difficulty coping during crisis situations e.g. terrorist bombs. During a crisis there is a huge strain on the network due to the great increase in voice traffic, both essential (emergency workers) and non-essential (public). The non-essential traffic degrades the quality of service for the emergency workers as it consumes scarce network resources. This paper presents a voice service that uses context-awareness [1] and programmable network technology [2] to deal with crisis situations. This context-aware voice (CAV) service classifies users into privileged users (emergency services, doctors, hospitals) and non-privileged users (eye witnesses, public) and restricts network access to only privileged users during a crisis situation. As more network resources become available to privileged users a better quality of service can be provided. The CAV service monitors for crisis conditions and enables the network to respond appropriately when one occurs.

This research was carried out as part of an EU funded project called CONTEXT [3]. This project created the CONTEXT platform [4] which uses programmable networks, a context infrastructure and policy-based service management (PBSM) for the creation, deployment and management of context-aware services (CAS). The CAV service was built on top of the CONTEXT platform and uses a session initiation protocol (SIP) [5] based VoIP platform called Siptrex [6] to provide voice services.

The next section, the background, provides an overview of the technologies used to create the CAV service. Then the service realisation and interactions are detailed. The service evaluation is then presented and the paper ends with the conclusions.

2 Background

2.1 Context and Context-awareness

Context consists of the implicit and explicit information of an entity, which can be used to characterise it. An entity can be a person, physical or computational object, service or application [1]. Typical examples of context are: (1) user location information (e.g. building, city); (2) social context (e.g. wife, boss); (3) personal preferences (e.g. favourite websites); (4) user's behaviour (e.g. task, habits); (5) device and network characteristics (e.g. network elements, bandwidth). This context can be used to enhance a service or application by personalising it, enriching it or making it more responsive to changes in its environment or situation [1]. When a service makes use of its context to provide relevant information and/or services to the user, it becomes context-aware. Context-awareness is characterised by the ability of a service or system to react to its changing environment. Context-awareness enables a new class of computing and network services [7]. These services can help users find nearby services or devices, decide the best devices to use, enable systems to react appropriately in certain situations, etc. CAS can be developed through the creation of a context infrastructure on top of a programmable network [4][8].

2.2 Programmable/Active Networks

In programmable or active networks, the packets contain code, which can be executed by active routers. Active or programmable routers are network nodes that execute the code contained in active packets [9] This code can be used to make the network nodes more intelligent and programmable. There exist two main approaches to realise active networks: *programmable nodes* and *encapsulation.* In the first approach, the user injects programs into the programmable node separately from the actual packet using existing network packet formats and providing a discrete mechanism for downloading programs to the active nodes [2]. In the *capsule* approach, programs are encapsulated in active packets. Here the active node has a built-in mechanism to load the encapsulated code and an execution environment to execute it [2].

3 The Context-aware Voice Service

3.1 CAV Service Realisation

The CAV service is a programmable network service that uses context-awareness to deal with crisis situations in voice networks. This service terminates all non-essential voice traffic when a crisis occurs and from then on, only priority users are allowed access to the network. The CAV service works as follows. A context computational object (CCO), the CH_Publisher, monitors the network, computes context and publishes it to the Context broker. A service execution condition evaluator (SICE), the SIP_SICE, subscribes to this information and determines when a crisis has occurred. It queries the Context broker to determine when the calls to 911 have satisfied a crisis condition. When a crisis occurs, the SIP_SICE informs the PBSM and a service level object (SLO), the CH_Main, is deployed to the nodes in the crisis area. CH_Main, a programmable application, begins executing by terminating all non-essential calls to and from the crisis domain. It then takes over call admission control from the SIP servers and only allows privileged users access to the network. When the crisis is over CH_Main stops executing, relinquishes call admission control and the CAV service reverts to its default state.

The CAV service is realised through the CAV service code (the SIP_SICE, CH_Publisher and CH_Main), the Siptrex and CONTEXT platforms and the SIP and Context brokers. The CONTEXT platform is used to create, deploy and manage the CAV service. The brokers serve as interfaces between the programmable network and the service code. The Siptrex platform is a SIP based VoIP platform that provides voice services for the CAV service. The CONTEXT platform and brokers are described below.

3.1.1 CONTEXT Platform and DINA Brokers

Figure 1 illustrates the high level architecture of the CONTEXT platform [5]. It consists of a distributed service execution environment (EE) on top of a transport layer. The EE is composed of two layers, the service support layer (SSL) and the active application layer (AAL).

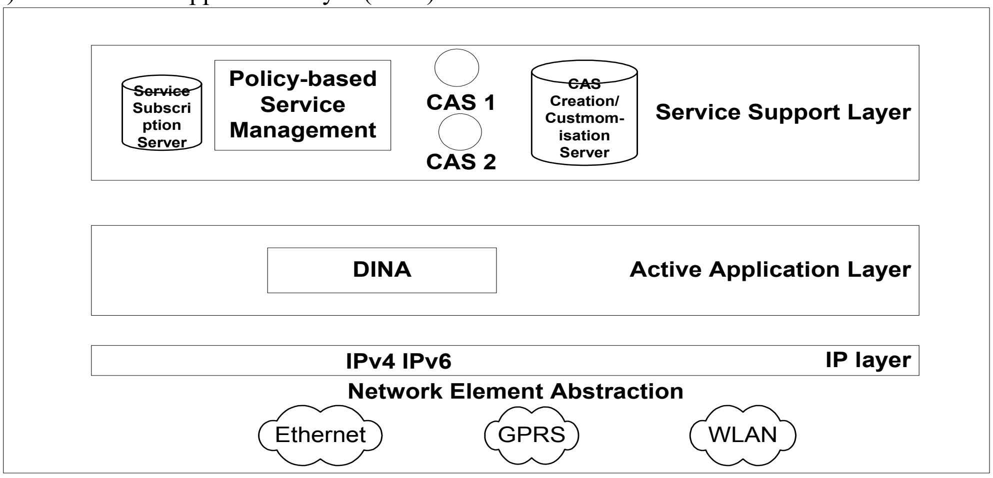

Figure 1: CONTEXT framework architecture

The SSL consists of two main subsystems, the CAS creation and customisation subsystem and the PBSM subsystem. The former enables context-aware service creation and customisation. The PBSM contains the policy management infrastructure including the policy manager, the policy repository as well as the code distributors and code execution controllers. Policies are used for service creation, deployment and operation.

The AAL provides the programmable network functionality. A programmable platform, DINA was developed, based on concepts used in ABLE [10][11]. DINA is a programmable middleware, which can be attached to routers to make them programmable routers [9][4]. It enables the deployment, control, and management of programmable services over networks entities. DINA provides interfaces (brokers) that can be used by the active services to retrieve information and perform network configurations. DINA is easily extended through the addition of new brokers. The following are two DINA brokers created for the CAV service.

- SIP Broker: This broker provides to the programmable entities the ability to obtain information and manage SIP components. The SIP broker performs three primary functions; provide information about the SIP servers, sessions and users; terminate voice sessions; and delegate call admission control to SLOs.
- Context Broker: This broker implements the mechanism for enabling SLOs to access the necessary context information from the various context sources. It provides the methods that enable context producers to publish their context information and context consumers (SLOs) to retrieve context.

3.2 CAV Service Interactions

3.2.1 Service Creation and Deployment

The CAV service is created using the CAS creation and customisation subsystem of the CONTEXT platform. The Siptrex, CONTEXT and DINA platforms are first installed. Then the Context broker, SIP broker and CH_Publisher are installed on the DINA nodes. The CAS code generator generates and customises the service code and policies for the CAV service. The result is the customised CH_Main and SIP_SICE, customised policies and customisation parameters for the CH_Publisher. The SIP_SICE along with the generated policies and customisation parameters is distributed to the code execution points (DINA nodes).

3.2.2 Crisis Detection

The CH_Publisher registers to publish the 911 call context to the Context broker. If a crisis condition is detected, the SIP_SICE is notified by the CH_Publisher through the Context Broker. Then the SIP_SICE notifies the PBSM which then distributes CH_Main to the specified DINA nodes. Figure 2, illustrates the interactions among the components of the CAV service involved in crisis detection.

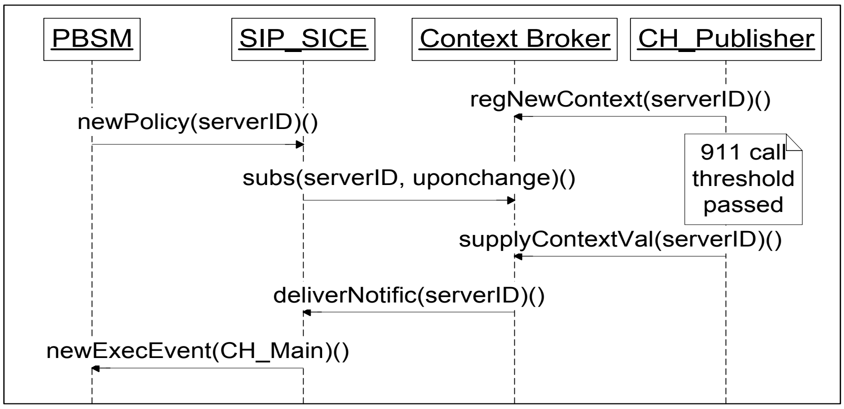

Figure 2: Sequence of interactions for the crisis detection stage

3.2.3 Service Execution

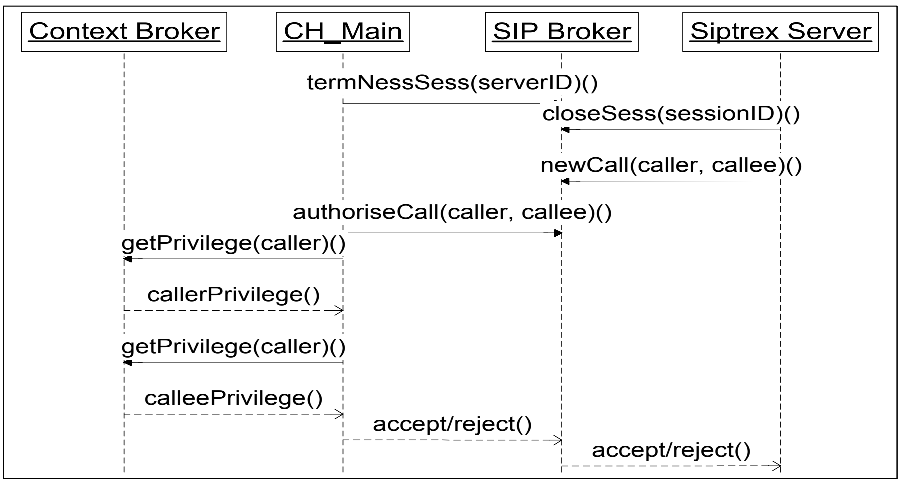

Figure 3: Sequence of interactions for the service execution stage

Figure 3 illustrates the interactions among components involved in the service execution stage. On arrival at the DINA node, CH_Main begins its execution by instructing the SIP broker to terminate all ongoing non-essential calls to and from the crisis domain. Then CH_Main is delegated call admission control. From this point on, all new sessions to be established must be authorised by the CH_Main allowing only privileged callers to make calls. When the crisis is over, CH_Main stops and relinquishes call admission control.

4 CAV Service Evaluation

4.1 Service Testing and Results

The CAV service was tested as follows. With an ongoing call between two non privileged users, five calls were made to the 911 SIP phone in less than five seconds (this was the definition of a crisis). This crisis was detected and CH_Main deployed and started executing on the DINA node. Calls were made between privileged users. Other calls were then made between non privileged users. After a while an event was raised signalling the end of the crisis situation. Then calls we made between privileged as well as non privileged users.

As soon as the crisis was detected and the CH_Main began to execute, the call between the non privileged users was terminated. While the calls between privileged users went though, the calls between non privileged were not authorised and did not go through. After the crisis was over all calls in the system were authorised. The service tests were all successful and hence the CAV works as was designed within the limits evaluated.

4.2 Service Extensibility and Flexibility

The CAV service is very flexible and extensible due to its context-awareness and policy-based infrastructure. More complex privilege allocation and logic to deal with it can be introduced as well as other call services. This can be done by altering the service logic found in CH_Main to allow several permutations of privilege and access control to be used. Some users could be allowed video calling, others voice only calls, some short message service (SMS) calls while blocking others, all according to their privileges. This can be achieved just by changing the user context published to the Context broker and the call admission logic in CH_Main. Changes to the service can be introduced during the service customisation phase. The definition of a crisis situation could be changed (e.g. 10 calls to 112 in 5 seconds) or a totally new crisis situation can be defined (e.g. 30 dropped calls in one minute).

4.3 Service Scalability

As the Siptrex and CONTEXT platforms are scalable, the CAV service can also be easily scalable. To allow scalability, much use is made of the modular design of the CAV service and the policy-based infrastructure. Additional Siptrex servers can be accommodated by using a policy defined naming scheme both for the individual servers and the context associated with each server. Different crisis conditions could be defined for each Siptrex domain. These parameters can all be defined during the service customisation phase.

5 Conclusions

The CAV presented is a programmable context-aware service. It uses context, programmable network technology and PBSM to enable a network to deal with the huge increase in voice traffic during a crisis situation. It is easily customisable, scalable and extensible due to the use of context and a policy-based infrastructure. The CAV service could easily be developed further through more complex privilege allocations and more complex logic to deal with them. Also the services provided by the CAV can be extended to provide video calls and instant messaging in which case it could be possible to define privileges that allow users to either have video calling, voice only calls or short message service (SMS) calls according to their user privileges.

6 References

[1] Dey, A. and Abowd, G., "Towards a better understanding of context and context-awareness". Proceedings of Workshop on the What, Who, Where, When and How of Context-Awareness, The Hague, Netherlands. April, 2000.

[2] Tennenhouse, D. L. and Wetherall, D. J., "Towards an Active Network Architecture" Computer Communication Review, Vol. 26, No. 2, April 1996.

[3] CONTEXT project website, http://context.upc.es/index.htm.

[4] CONTEXT Consortium, "Context project deliverable D2.2: CONTEXT Architecture: Solution for provisioning and delivery of context aware services" ed. UCL, 2004.

[5] Sinnreich, H., Johnston, A., "Internet Communications Using SIP," John Wiley & Sons, New York, 2001.

[6] Siptrex website, www.siptrex.net.

[7] Ebling, M., Hunt, G. and Lei, H., "Issues of Context Services for Pervasive Computing," Proceedings of Workshop on Middleware for Mobile Computing, Heidelberg, Germany, 2001.

[8] Sygkouna, I. et al., "Context-Aware Services Provisioning on Top of Active Technologies," IFIP 5th International Conference on Mobile Agents for Telecommunication Applications (MATA 2003), Marrakech, Morocco 8-10.10, 2003.

[9] Denazis, S. G., Galis, A., "Open Programmable & Active Networks: A Synthesis Study" IEEE IN 2001 Conference, Boston, USA, 6- 9 May 2001, ISBN 0-7803-7047-3.

[10] Raz, D. and Shavitt, Y., "Towards Efficient Distributed Network Management," Journal of Network and Systems Management, September 2001.

[11] ABLE: The Active Bell Labs Engine http://www.cs.bell-labs.com/who/ABLE/