Policy-driven Mobile Agents for Context-aware Service in Next Generation Networks

Kun Yang, Alex Galis, Chris Todd

University College London, Department of Electronic & Electrical Engineering, Torrington Place, London WC1E 7JE, UK
Email: {kyang, agalis, ctodd}@ee.ucl.ac.uk

Abstract. In order to bring together the different advantages of various wireless technologies and wired networks, solution from service's perspective is critical. And this kind of integrated services should be context-aware in order to automatically adapt themselves to the changing environment. This paper proposes an all-policy based context-aware service method for next generation networks (NGN). A thorough consistency is expected to be achieved by this all-policy method where policies are well planned to cover from context representation through services down to the underlying networks. The efficiency of context-aware service provisioning as a result of policy decision is maximized by the introduction of mobile agents. A context-aware service scenario called Super-mother is explored to exemplify this methodology and policy-based context-aware service system architecture. This paper presented part of the work ongoing in European Union IST project CONTEXT.

1 Background and Rationale

One of the main characteristics of next generation networks (NGN) is the convergence of both fixed networks and wireless networks. And this convergence has attracted plenty of researches aiming to bring together the higher speed of wired networks and wider coverage of wireless networks (typically represented by GPRS/UMTS). As the hardware and lower-level structures and protocols of wireless networks get mature, the demands from higher-level applications and services relevant to integrated networks are rapidly growing, especially when wireless LAN (WLAN) technology becomes increasingly popular for providing IP connectivity and 3G is undergoing deployment stage. This paper tends to contribute this literature from the service's perspective by providing context-awareness to the integrated services operating on NGN. We believe the essence of NGN services is the ability of being *context-aware*. Furthermore, the provisioning of context-aware service is facilitated by policy-driven mobile agents.

Bearing a wider scope than *location*, *context* refers to the physical and social situation in which computational devices are embedded [1]. *Context-aware service* (CAS) is more flexible and autonomous so as to respond accordingly to the highly changing computing environments such as location, terminal size, and network features etc without disturbing end user. For example, a cell phone will always vibrate rather than beep during a meeting, if the system can know the location of the cell phone and the meeting schedule. While most of the researches on context-aware computing focus mainly on the human-computer interface (HCI) [1, 2], this paper tends to tackle the context awareness from the perspective of networks, i.e., *network-centric* context-aware services. The networks include both wired IP networks and wireless networks.

To facilitate the provision of context-aware services, apart from an appropriate infrastructure to gather, manage, and disseminate context information to services, the design and development of a context model that takes into account the service and network management is even more important. This context model serves as the basis for CAS system infrastructure. This paper explores the applicability of *policy* for the representing of context in context-aware service and the applicability of mobile agent for the deployment of these context-aware services.

The reason why policies are employed for context modelling is partly because we want to take into consideration the implementation of context-aware services in the underlying networks where policy-based network management (PBNM) is widely regarded as a promising means. Policies are seen as a way to guide the behaviour of a network or distributed system through high-level declarative language in PBNM field, which has been the subject of extensive research as a new network management method over the last decade [3, 4]. As many research works have shown, PBNM technology can relieve network administrator from the burden of configuring every single device manually and it is more flexible since administrator can reconfigure network elements by just giving or changing policies.

On the other hand, in order to provide context aware services, contextual information need to be represented, stored, and maintained. Context information are usually complex, changing, layered and related to each other, which means the use of context information needs complex decision making. Policy-based method fits well to these features of context. Policy-based context modelling can comply very well with the underlying PBNM thanks to the common policy-based schema.

The integration of policy and mobile agents for the automated and flexible management of IP networks has been shown in [5]. This paper tends to take the method one step further by extending it to the provisioning of context-aware service, as depicted in Figure 1.

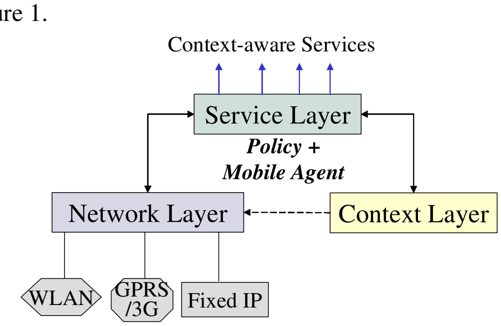


Figure 1: All-policy CAS Method in NGN Environment

The content of the paper is structured as follows. After the scenario description in Section 2, Section 3 discusses the requirement of context modelling and context-aware service system in term of context definition, context classification and policy specification. Then policy-based context modelling and CAS system architecture are presented in Section 4 and 5 respectively. Before conclusions and future work in Section 7, Section 6 presents the implementation of scenario mainly by means of workflow.

2 Scenario Description

A typical network-centric context-aware service called *TEANU* (Transparent Enterprise Access for Nomadic User) is described, which represents a typical service in NGN.

This service tends to provide a means for nomadic user to maintain the *secure* access to her enterprise network *transparently* after the user has registered for context-aware services from context-aware service provider.

Consider Katherine, a middle class graphic designer with 3 kids. Katherine works from home a few days a week, using her home network that is connected to the office network. On the due day of the project she was working on in the last few weeks, Peter, her 9 years old son, complained that he does not feel well, and his situation degraded to the point that she had to take him to the local public hospital.

Waiting in the unavoidable lines she tries to send her work to the office, using her laptop and a cellular modem. Unfortunately, the bandwidth of the cellular network is insufficient, and transferring the 10GB file will take about 7 hours, far passing the deadline.

However, luckily enough, she had subscribed to this new *TEANU* service that detected bandwidth problem and tried to resolve them. In this scenario the *TEANU* service found out that there was an appropriate wireless LAN (WLAN) access point in the university, and as such it dynamically switched the network connection to this faster wireless LAN. At the same time, a secure tunnel was established automatically between the hospital and her office network. Katherine was then connected to the local network so as to deliver the file faster. She noticed happily the significantly increased file transferring process, and her work was submitted successfully before the deadline. The whole scenario is called *super-mother*. This scenario will be used throughout the paper for exemplifying this all-policy based CAS over NGN.

3 Requirement Analysis

Due to the heavily overloaded use of the term *context* with a wide variety of meanings, a clarification of context definition in the context of this paper is the most important requirement for further context modelling and management. The context also needs to be classified so as to guide the design and implementation of context classes in context information model.

3.1 Context Definition

In the work firstly introducing the term *context-awareness* [1], B. Schilit *et al.* refer to context as location, identities of nearby people and objects, and changes to those objects. They claim that the important aspects of context are: *where you are, who you* *are with, and what resources are nearby*. Dey *et al.* give their definition of context as follow: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [2]. There are plenty of other definitions of context which more or less share the same meaning.

We feel Dey's context definition shouldn't limit the entities to just these that exist only *between user and application*. This constraint, as continuously implied by Dey's later work and most of other context-related work carried out in this field, indicates the origination of context (and context-awareness): Human-Computer Interface (HCI). This paper extends the scope of context research into networks, especially NGN by considering *network-centric context-awareness*. The context definition in this paper is as follow:

Context is any information, obtained either explicitly or implicitly, that can be used to characterize one certain aspect of an entity that involves in a ***specific*** *application or* ***network service****. An entity can be a physical object such as a person, a place, a router, a 3G network gateway, a physical link, or a virtual object such as IPsec tunnel, SNMP agent.*

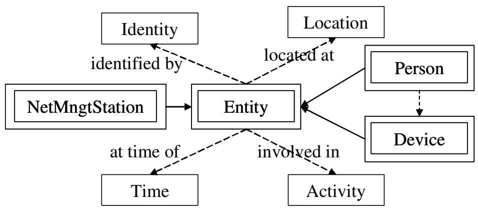
Please note that the collection of context can be done explicitly or implicitly, and an entity can be any network element, wired or wireless.

3.2 Classification of Context

A classification of context types will help application designers decide the most likely pieces of context that will be useful in their applications. Previous definitions of context seed our development of context types. Schilit *et al.* [1] list the important aspects of context as where you are, who you are with and what resources are nearby. Schilit *et al.* didn't consider time this critical factor. Ryan *et al.* suggest context types of location, environment, identity and time [2].

In this paper, sharing the same idea as proposed by Dey [6], we use *location*, *identity*, *time*, and *activity* as basic context types for characterizing the situation of a particular context entity, as depicted in Figure 2.

This context classification clearly answers the questions of who (*Identity*), where (*Location*), when (*Time*), and what (*Activity*) for a specific context entity (*ContextEntity*). This contributes to the horizontal classification of context information. The vertical classification of context information is about context entity itself, which can be categorized as: *persons* including end users with different roles, service provider administrator, network administrator, or *devices* such as 3G mobile handsets, fixed IP routers, WLAN access point, etc, or virtual entities like *network management station*. Furthermore, an object-oriented design of these context types is also carried out in this paper, which is absent in Dey's work.



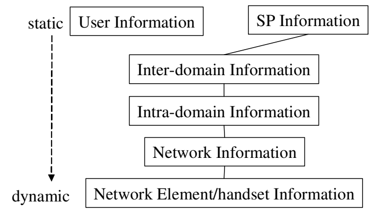


Figure 2: Classification of Context **Figure 3: Network-centric Context Information**

As far as network-related context information is concerned, its categories follow the logically hierarchical structure of network as depicted in **Error! Reference source not found.**. The higher the context information sits the more static they are.

3.3 Policy Specification

Here policy specification language is rather informal in the sense that it is a kind of high-level English-like declarative language used by administrator to add and update management policies. Policies serve as the containers of context information. As suggested by the Internet Engineering Task Force (IETF) Policy Framework Working Group [4], which has been investigating policies as a means for managing IP-based networks for many years, policies take the following rule-based format:

```
IF {condition(s)} THEN {action(s)}
```

It means *action(s)* is/are taken if the corresponding *condition(s)* is/are true. Policy condition can be in both disjunctive normal form (DNF, an ORed set of AND conditions) or conjunctive normal form (CNF, and ANDed set of OR conditions).

A typical context-aware service scenario can be represented by the following policy, which forces the mobile handset only vibrates rather than beeping during meeting time.

```
IF (location == meetingRoom) and (time within meetingSchedule) THEN MobileVibratingOnly
```

By this means, any policies related to the CAS can be defined. The execution of these policies eventually triggers the action of classes in policy information model, which are further wrapped by mobile agents travelling around networks.

These rule-based policies are further represented by XML (eXtensible Mark-up Language) due to XML's built-in syntax check and its portability across the heterogeneous platforms.

4 Policy-based Context Modelling

Context modelling addresses in this paper the issues of how to represent the contextual information in a way that can help bridge the gap between application that uses context information and the deployment of context-awareness. Particularly, an object-oriented (OO) information model should be adopted.

The pioneering work in this area was carried out by Schilit in his PhD thesis [7]. The model of context used in this work was extremely simple, with context information being maintained by a set of environment variables. The most relevant work concerning context modelling is the work lately carried out by K. Henricksen *et* *al*. [8]. This model overcomes problems associated with previous context models such as their lack of formality and generality. As its many precedents, contextual information related to network and network management is not fairly taken into account. Another shortcoming of this work is that all its discussion is on the graphical model while leaving the object-oriented context model unmentioned; therefore a big gap between their context modelling and real implementation of context-awareness is still left unfilled. This is probably due to the lack of a clear picture of the context-aware system that will use these contexts.

A policy-based context information model is designed in this paper based on the IETF PCIM (Policy Core Information Model) and its extensions [9], which are followed by most of the work carried out in policy-based management field.

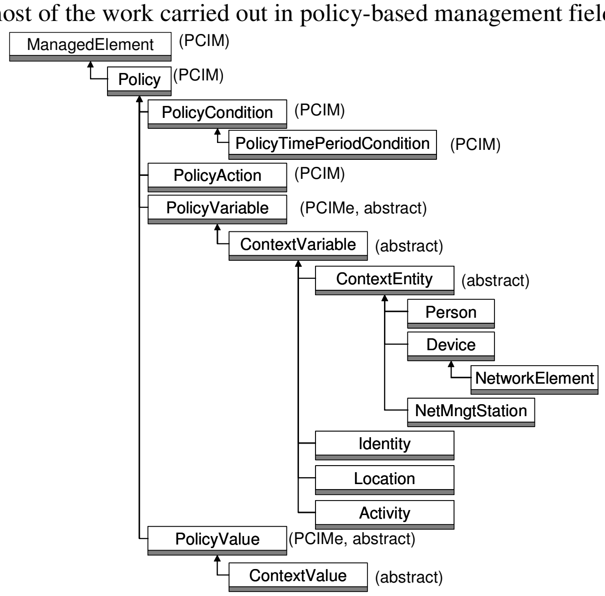


Figure 4: Class Inheritance Hierarchy in Policy-based Context Information Model

Figure 4 depicts part of the inheritance hierarchy of our information model representing context in network-centric CAS. It also indicates its relationships to IETF PCIM and PCIM extension (simplified as PCIMe in Figure 4). Some of the actions are not directly modelled due to the space limitation. Please note that apart from the policy condition and policy action, context information rooting from *ContextVariable* is mainly reflected in this figure.

Context information is expressed in the policy conditions in the form of elementary Boolean expressions of the form: <*Context Variable*> *MATCH* <*Context Value*>. The relationship "MATCH", which is implicit in the model, is interpreted based on the context variable and the context value. Therefore, the modeling of context information starts from *ContextVariable* and *ContextValue* which inherit two abstract classes available in PCIMe: *PolicyVariable* and *PolicyValue* respectively.

As discussed in classification of context, every context entity should at least be described by four basic features, i.e., identity, location, activity and time. The first three of them are modelled in this class hierarchy as shown in Figure 4. The time feature of context can be directly expressed by *PolicyTimePeriodCondition* which is available as a direct child of *PolicyCondition* in the IETF PCIM.

Now let's have a look at the modelling of network-related context information as described earlier in **Error! Reference source not found.**. Since the final fulfilment of network-centric context-aware service is on network elements (including mobile handsets), network elements play a vital part in CAS providing. As such, *NetworkElement* class is defined explicitly as a child of *Device* class. Network information and intra-/inter-domain information are usually maintained by network management stations at different levels or domains. Therefore, network management station (*NetMngtStation*) is defined as a kind of entity that has all the four basic context types. User and Service Provider information can be easily presented by the *Person* class which is actually the child of *ContextEntity* class therefore no separate class for this purpose is needed.

ContextValue is used for modeling context values and constants used in context-related policy conditions. Many extensions of the abstract class *PolicyValue*, as already defined in IETF PCIMe, provide a list of values for basic network attributes which are used directly for context purpose.

5 CAS System Architecture based on Policy and Mobile Agents

Mobile agent based CAS system architecture, as depicted in Figure 5, has been designed, whose components are organized in term of PBM structure as proposed by IETF Policy Framework Group, which gains wider popularity.

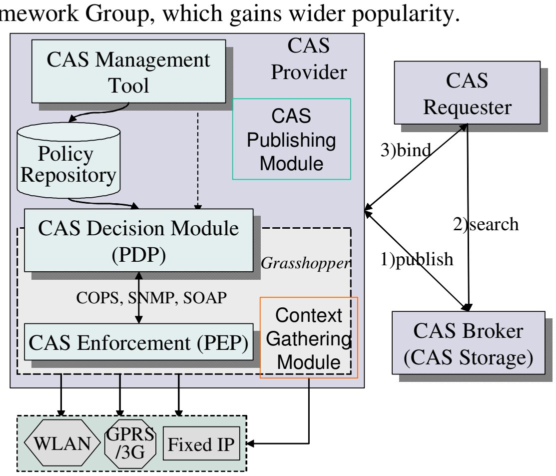


Figure 5: CAS System Architecture based on PBM and Mobile Agents

As illustrated in Figure 5 from top down, the PBM system for CAS management mainly includes four components: policy management tool, policy repository, Policy Decision Point (PDP) and Policy Enforcement Point (PEP). Policy management tool serves as a policy creation environment for the administrator to define/edit/view policies in a high-level declarative language. Depending on the login name, CAS Management Tool also provides management GUI for network management (mainly

IP VPN) and context information management. After validation, new or updated policies are translated into a kind of object oriented representation or so-called information objects and stored in the policy repository. The policy repository is used for the storage of policies in the form of LDAP directory. Once the new or updated policy is stored, signalling information is sent to the corresponding PDP, which then retrieves the policy and enforces it on PEP.

After a CAS is mature for deployment, CAS provider can publish it to CAS storage, which can be owned by brokers, so that it can be searched for by any CAS requester like Katharine.

Mobile agent technology is explored in this architecture as an enabling tool for flexible service delivery. Mobile agent platform is based on the *Grasshopper*. Grasshopper [10] has been designed in conformance with the Object Management Group's Mobile Agent System Interoperability Facility (*MASIF*). Furthermore, it is a commercial product with extensive documentation and future development under way. However it also provides the version for non-profitable use. In this paper, the mobile agents are fully guided by the policies. One big advantage with mobile agents is that functions can be dynamically provisioned *on the fly*. This feature matches the requirement of mobile user or the features of wireless networks very well.

6 Implementation of *Super-mother*

The test-bed for exemplifying the context-aware integrated service *TEANU* scenario as described earlier in this paper is depicted in Figure 6. The experiment is to simulate the scenario which begins from the point when Katharine enters into the hospital where WLAN is available and ends at the point when the data from the Katharine's laptop begins to transport through an IPsec tunnel between hospital WLAN Access Point and ingress router of her office network.

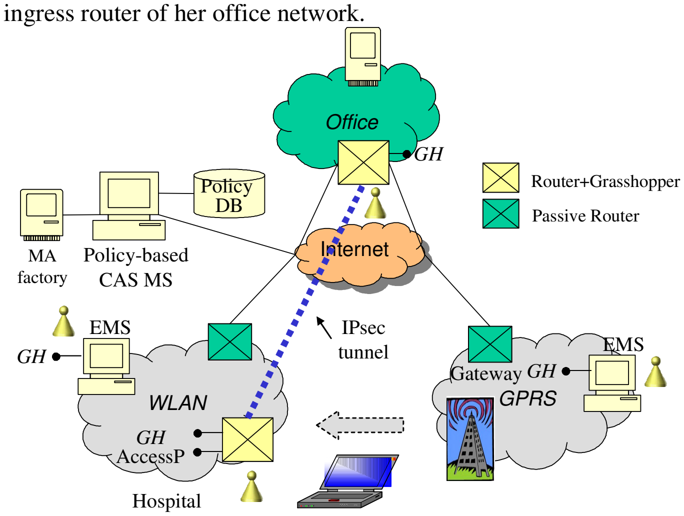


Figure 6: Test-bed for TEANU Super-mother

Policy-based CAS Management Station (MS), belonging to the Context-Aware Service Provider (CASP), is used by service administrator to management context information, services and networks. EMS (Element Management Station) is introduced for the management of *ABLE* execution environment, access point, etc. Therefore a two-tiered management infrastructure is built up. *MA factory* in Figure 6 serves as the repository of mobile agent classes.

The workflow of this *super-mother* scenario is as follow:

1) When customer (Katharine) subscribes, she should provide CASP with her WLAN card MAC address so as to get access to Access Point (AP).

2) CASP sends this MAC address to all APs that involve in this CAS so that AP can sense the arrival of this specific WLAN card (representing customer himself) and report it to the CASP. These APs are usually given by customer during service subscription according to her selection of possible venues where this service is likely to be applicable. To send customer's MAC address to APs in advance can protect the CASP from being frequently disturbed by unregistered WLAN users.

3) Location sensor (AP in this case) serving as PEP sends a signal to CASP if Katharine enters into the covered scope of AP.

4) Based on the policies in CAS management system and all the relevant context information (such as Katharine's ID, role in the company, etc), CAS Decision Module decides policies to be sent to WLAN domain, and initiate the corresponding mobile agent existing in the MA factory which sends the policy to EMS because AP is not policy-sensitive.

5) After policies arrive at the EMS, EMS begins to configure the access control table of AP to allow Katharine to get access to the AP. Then Katharine's uploading traffic will be automatically switched from GPRS connection to WLAN connection.

6) When the first packet from Katharine's laptop arrives at the AP, its destination address will be checked by Address Checking daemon (a stationary agent) running on the machine where AP is installed. This daemon can find that the packet is to be delivered to her office. Based on the policy that if the packets from Katharine are destined to her office then secure tunnel should be set up for this transferring, IP VPN is going to be set up between hospital and her office. To make the customer side as slim as possible (or without disturbing the user by installing any supporting software such as *Grasshopper* platform or IP VPN software), the secure tunnel is set up between two routers rather than between enterprise router and customer's laptop. Here in this scenario, the machine on which AP is installed also serves as a router.

7) The Address Checking daemon then reports this VPN setup requirement to CAS Management Station (via EMS), which, after further decision making, sends mobile agent to Access Point machine which has *Grasshopper* platform preinstalled to set up one end of IP VPN tunnel in WLAN domain. After this, the mobile agent travels from AP machine all way to ingress router of her office to set up the other end of the tunnel. Then this mobile agent travels back to the management station to report the result. At the same time, customer traffic goes through this secure tunnel.

When uploading is finished (this can be sensed by significant drop of traffic), the TEANU service can switch itself to another level of services to save money.

7 Conclusions and Future Work

This paper provides context-aware service over NGN using the integration of policy-based management and mobile agent technology. This CAS can adapt itself to the change of complex environment dynamically according to the policies resulting from customer-provider contract (or so-called SLA, Service Level Agreement) and rules cast by network administrator for proper use of the networks. The CAS deployment is fully based on mobile agents. Based on the context model and CAS system architecture, a concrete context-aware service scenario *Super-mother* demonstrated the applicability of all-policy method for context-aware services over NGN.

Further refinement of policy-based context model, especially on introducing more WLAN and 3G specific actions, is the main future work. And these actions will be carried out by mobile agents.

Acknowledgements

Part of the work presented in this paper describes part of the work undertaken in the context of the EU IST project CONTEXT (IST-2001-38142-CONTEXT) [11]. The IST programme is partially funded by the Commission of the European Union.

References

[1] B. Schilit, M. Theimer. "Disseminating Active Map Information to Mobile Hosts". *IEEE Network*, 8(5): 22-32, 1994.

[2] A. K. Dey and G. D. Abowd. "Towards a better understanding of context and contextawareness". Proceedings of Workshop on the What, Who, Where, When and How of Context-Awareness, April, 2000.

[3] M. Sloman. "Policy Driven Management For Distributed Systems," *Journal of Network and System Management*, vol. 2, no. 4, Dec. 1994, pp. 333–60.

[4] IETF Policy Framework Working Group web page: http://www.ietf.org/html.charters/policy-charter.html

[5] K. Yang, A. Galis, T. Mota and S. Gouveris. "Automated Management of IP Networks through Policy and Mobile agents". *Proc. of 4th Int. Workshop on Mobile Agents for Telecommunication Applications*: 249-258, Barcelona, 2002.

[6] A. K. Dey. "Providing Architectural Support for Building Context-Aware Applications". PhD thesis, Georgia Institute of Technology, November 2000.

[7] B.N., Schilit. *A Context-Aware System Architecture for Mobile Distributed Computing*, PhD thesis, 1995.

[8] Karen Henricksen, et al. "Modeling Context Information in Pervasivev Computing System". *Proceedings of Pervasive 2002*, LNCS 2414, pp. 167-160, 2002.

[9] B. Moore. "Policy Core Information Model Extensions". IETF-RFC 3460, IETF Policy Framework Working Group, Jan. 2003.

[10] Grasshopper website: http://www.grasshopper.de, 2002

[11] European Union IST Project CONTEXT (IST-2001-38142-CONTEXT) web site: http://context.upc.es/