A User-Centric Privacy Framework for Pervasive Environments

Susana Alcalde Bagüés[1 2], Andreas Zeidler[1],Carlos Fernandez Valdivielso[2], Ignacio R. Matias[2]

[1]Siemens AG, Corporate Technology
Munich, Germany.
[susana.alcalde.ext|a.zeidler]@siemens.com
[2]Public University of Navarra
Department of Electrical and Electronic Engineering
Navarra, Spain.
[carlos.fernandez|natxo]@unavarra.es

Abstract. One inherent feature of pervasive computing environments is the need to gather and process context information about real persons. Unfortunately, this unavoidably affects persons' privacy to a large degree. Each time today a citizen uses his cellular phone, his credit card or surf the web, he is leaving a trace that is stored for some reason. In a pervasive sensing environment, however, the amount of information collected is a) much larger than today and b) might be used to reconstruct personal information with great accuracy. The question we address in this paper is to *control* dissemination and flow of personal data across organizational, as well as personal boundaries, i.e., to potential addressees of privacy relevant information. This paper presents the *User-Centric Privacy Framework* (UCPF). It aims at protecting a user's privacy based on the enforcement of privacy preferences. They are expressed as a set of constraints over some set of context information. To achieve the goal of cross-boundary control, we introduce two novel abstractions, namely *Transformations* and *Foreign Constraints*, in order to extend the possibilities of a user to describe his privacy protection criteria beyond the current expressiveness ussually found today. *Transformations* are understood as any process that the user may define over a specific piece of context. This is a main building block for obfuscating or even plainly lie about the context in question. *Foreign Constraints* are an important complementing extension because they allow for modeling conditions defined on external users that are *not* the tracked individual, but may influence disclosure of personal data to third parties. We are confident that these two easy-to-use abstractions together with the general privacy framework presented in this paper constitute a strong contribution to the protection of the personal privacy in pervasive computing environments.

1 Introduction

Pervasive computing involves merging technology into the everyday life to such an extent that computer environments will be integrated into people's ongoing needs, practices, values and goods. Technology becomes invisible and seamlessly interconnected. Users will be provided with services and information in an anywhere, anytime fashion.

This vision also entails a pervasive sensing of personal information, often in real time, such as identity, location and activity (in the following simply called context information). Privacy issues are some of the main concerns about pervasive computing: without explicit control by the individual what data is disclosed when, how, to whom and under what constraints, the vision of Mark Weiser [20] ultimately can fail.

In this paper, we describe a User-Centric Privacy Framework which aims at protecting a tracked individual's privacy. The respect of privacy preferences is not an easy issue, since they depend on a user's *wishes* which are variable by nature. Privacy preferences commonly are expressed as a set of constraints to control the flow of information from the sender to the recipient. So far, constraints only affects the tracked individual or/and the service's features [14] [11], e.g. time, service, activity or location constraint. However, often one may wish to express not only constraints over the own context but also over the context of the recipient, or other *external* users, like in this trivial example: 'Bob wants to reveal his activities to his wife only if they are in the same city'. In this case not only the tracked individual's context has to be considered but also the recipient's context. A situation which cannot be catered for today's privacy frameworks.

Our framework is based on the use of policies to define and to enforce user's privacy constraints. Another important feature in this approach is the possibility to enrich policies by transforming context information. So far, policies languages cannot be used to express the obligation of transforming data before publishing. Policies are widely classified in the literature [17] [5] [10] as either *authorization* or *obligation* policies. Authorization policies are used in the context of privacy to permit or deny the deliver of a piece of context information (absolute decisions). An obligation policy involves a future promise linked to the fact of disclosing information. There could be many situations in which to fulfill the user's privacy preferences includes the deliberate modification of a piece of context information, e.g. to reduce the precision in a tuple of coordinates.

Policy-based privacy frameworks have been implemented in, e.g. [3] [12] [14] [18]. In general, such frameworks define policies in a non-semantically enriched language, which we consider to fall short in many cases. In order to be able to express the richness of the user's privacy preferences, we adopt and extend a semantic policy language, namely the *Rei declarative policy language* [10] [9], applied mainly to facilitate effective agent communication and access control. In order to cater for situations in which we have to *transform* data from one data set into another, we introduce *SeT*, a policy language for creating *transformation policies*.

The remainder of this paper is structured as follows: We motivate our approach by introducing a simple application scenario in the Section 2. Section 3 discusses relevant related work in some detail, followed by the introduction of our user-centric privacy framework and the complementing SeT policy language in Section 4and 5, respectively. Finally, Section 6 concludes this paper and indicates the directions of future work.

2 Scenario

We want to illustrate the concept of *Foreign Contraints* and *Transformation* in privacy preferences by introducing the following example scenario.

Ana is an employee of a home health-care organization (the HHCO). As all the nurses in that company, Ana allows her employer to track her working times. The HHCO wants to improve its service by informing the patients of the nurses' estimated time of arrival, a service very much appreciated. Additionally, Ana is subscribed to a traffic information service (TIS), which provides real-time information about the traffic conditions found en-route to the next patient. Also, another location-based service (LBS) is used whenever her car needs to be refilled at the nearest petrol station approved by her company.

On the other hand, privately Ana is used to synchronizing her calendar electronically with her husband Bob. Arguing that knowing about Bob's activity will help her to organize the daily life better, Bob agrees to give Ana additional information about his current activity, e.g, in a meeting, driving, etc, but only when both are in the same city.

Obviously, even in this simple scenario, various privacy issues are tackled. It is important for an organization like the HHCO to respect its customers wish to keep the actual identity undisclosed to someone other than the HHCO. Therefore, the location of the customer visited by the nurses has to be obfuscated as much as possible. The problem of privacy disclosure occurs when Ana interacts with some *implicitly untrusted* service like the LBS for finding a petrol station. In the case of the TIS, we assume that the HHCO states that the nurse's identity is not revealed, since the location has to be disclosed. Hence, both interactions with an external service implies that the original coordinates are *transformed* into a different data set than the original one. In the first case to decrease the accuracy and in the second case to meet a parameter K in an *anonymity set* within K users. This concept is detailed in Section 3 below.

Target	Context	Constraints	Transformation	Service
Ana	Location	- Working time - Authorized personal	N/N	HHCO's LBS
Ana	Location	- Working time - Unauthorized personal	Spatial Obfuscation 500 meters	External LBS
Ana	Location	- TIS - Undisclosed Id	Data/identity abstraction en *K*	TIS
Bob	Activity	- Id = Ana - Same city than Bob	N/N	Current activity

Fig. 1. Scenario Rules.

Summarizing, services that use context information in the above scenario are: the HHCO's LBS, the TIS, an LBS that informs of the nearest petrol station, and the Bob's activity service. The free distribution of the Ana's location and Bob's activity are restricted by a set of constraints as shown in the Figure 1. The delivery of a nurse's location to some external LBS or TIS includes the necessity to transform coordinates. The use of foreigns constraints is illustrated in the situation in which Bob decides only

to reveal about his activity to his wife when both are in the same city, which means not only to consider Bob's but also Ana's actual location before delivering any information.

3 Related works

There are different methods to address privacy protection, mainly: policies, anonymization and obfuscation techniques. None of them achieve the goal of total protection of the user privacy integrity, thought. In order to interact meaningful with a pervasive environment, it will always be necessary to give up some amount of privacy. The goal is to control how much privacy is disclosed for what reason.

According to [4], a privacy policy is an assertion that a certain amount of information may be released to a defined entity under a certain set of constraints. We classify privacy policies from the point of view of defining *service privacy practices* or *user privacy preferences*.

A well-know approach of privacy policies stems from the World Wide Web Consortium (W3C), which standardizes the *Platform for Privacy Preferences (P3P)* [3]. P3P enables web sites to express their privacy policies and compare them with the user's privacy preferences, which, in turn, can be specified by using *A P3P Preference Exchange Language (APPEL)* [13]. The policies are transferred to the user's browser and then matched to his personal preferences there. However, as stated in [4], P3P has not been tailored to the specific requirements of pervasive applications. *PawS*, a privacy awareness system for ubiquitous computing [12], extends P3P to cover aspects of pervasive applications. In PawS, when a user enters in an environment in which services are collecting data, a *privacy beacon* announces privacy policies of each service. A users *privacy proxy* then checks these policies against the user privacy preferences. If the policies agree, the services can collect information and users can utilize the services. If the policies do not match, the system notifies the user, who then can choose not to use the service in question or, in some cases, simply physically can leave the area in which the collection of information occurs. Both define privacy practices for services which are not the scope of this work.

While APPEL [13] provides a good starting point for expressing privacy preferences, it cannot support the richness of expressions necessary for the evaluation of user criteria in real-world application domains. In [14] such requirements are implemented as system components called *validators*. The features of validators are described without defining a concrete implementation language and they need a centralized location provider to enforce them. Another approach is the *Confab system* [8], where a complex data structure is elaborated to represent contextual information, the basic context atom called a *context tuple*, which is equivalent to a web page. Information is captured, stored, and processed on the end-user's computer. This gives end-users a great level of control and choice over what personal information is disclosed but fail in flexibility to share context information.

Anonymization mechanisms technically hide the identity of a tracked user with respect to emitted context data so that she is not identifiable within a set of other tracked subjects, constituting the *anonymity set* [15]. We can distinguish between techniques of data and identifier abstraction. In a data abstraction, anonymity can be ac-

complished by cloaking data, e.g., by reducing temporal and/or spatial accuracy, so that data of different targets cannot be distinguished. In [7], cloaking is based on the formal model of k-anonymity [19]. For enforcing k-anonymity, a trusted context provider is needed, which has global knowledge about a group of targets. In identifier abstraction, pseudonyms are associated with context data. However, this approach suffers from the obvious problem that pseudonyms can be uncovered by statistical attacks. For this reason, in [1], pseudonyms are dynamically changed into *mix zones* to avoid linking different pseudonyms of a target together. In [6], a formal model for obfuscating location information is given. In contrast to anonymization techniques, which have the objective of hiding targets' identities, the identity is supposed to be known. Instead, position accuracy is reduced as far as application requirements can still be adhered to.

So far Transformations have been only considered by *Geopriv* in the draft proposal for expressing privacy preferences for location Information [16]. In this approach transformations are part of authorization policies. The evaluation of the authorization policy is done by the location server which executes the transformation to ensure minimal disclosure of location information.

4 User-Centric Privacy Framework

In this section we present the User-Centric Privacy Framework (UCPF). We propose a user-centric approach to safeguard user privacy preferences. Several publications in the area of privacy protection, such as the well-known P3P [3], are concerned of how to define service side privacy practices, in such situations the user can only check the service's privacy criteria and either accept or deny the service. Our approach, like the Confab system [8], seeks to give as much control as possible to the tracked user, in our case without assuming that the sensing technology is attached to the user's computer device. A privacy framework should give users explicit control over what data are disclosed when, how, to whom, and under what constraints.

We introduce two novel concepts, *Transformations* and *Foreign Constraints*, to give users more freedom to define their privacy preferences. Basically, we define Transformations as any process that the tracked user may define over a specific piece of context information, and Foreign Constraints as the context information of *external* users which must be taken into account before forwarding privacy relevant information. An example is the current location of Ana for delivering information about Bob's activity to her.

The integration of these features in the UCPF poses the following requirements; Any transformation on a piece of context information should be only known by the tracked individual. Transformations should be automatically applied when the selected policy is enforced. The UCPF should act as point of enforcement for Foreign Constraints, this implies that before delivering context information to a third party, Foreign Constraints must be retrieved, evaluated and the enclosed policy has to be enforced by the UCPF. If all constraints are satisfied, the UCPF should also act as the context provisioning proxy.

In order to address the requirements stated above we introduce *Sentry*. Sentry is a major architectural building block within the UCPF. An Sentry instance manages the context disclosure of a tracked entity to third parties. To do so, it gathers and filters

collected data about an entity and - based on the set of Foreign Constraints and Transformations - disseminates suitable data to third party services as a context proxy.

4.1 Role Model

The data flow along the processing chain in a service that is fed with context information, involves different autonomous entities like people, companies or organizations, we refer to as actors, see [11]. In order to understand how the UCPF works in the processing chain, we distinguish the following roles; *Target* is the tracked individual, in the scenario Ana is the target for the HHCO's LBS and the TIS. Based on Ana's location *service providers* compile locations for users of this services, which are the *recipients* of the information (respectively the HHCO and Ana). An intermediate role is played by the *context provider*, in our guiding scenario this is the mobile provider as the location information provider, who is also responsible for collecting, caching and managing location information of Ana and transferring them, accordingly.

The actors along the processing chain can be classified as *trusted* and *untrusted*. Trusted parties are those entities that have any contractual and usually legally binding relation in a way that it protects the exchange of the target's context information, such as between the HHCO and its mobile provider. Untrusted parties, on the other hand, are such third parties that are not under the control of the target, like the TIS.

Using policies entails to include a policy distribution architecture [21], introducing the following four roles: the *policy repository*, the policy management tool, the *policy decision point* (PDP) and the *policy enforcement point* (PEP). From the point of view of the target's privacy only trusted parties should act as the PEP, while untrusted parties should only receive context information from the PEP.

The most favorable solution, from the point of view of privacy, is that the context provider adopts the role of the PEP, since we assume that there is a contractual relationship between the HHCO and its location provider, it is also the solution followed in [14], [18]. However, this solution is not applied when a policy includes Foreign Constraints to be enforced. For example in our scenario, Bob's activity is disclosed only when Ana is in the same city. Since the Ana's mobile provider, the Bob's mobile provider and Bob's activity provider might be three different entities, Bob's activity provider is not able to enforce a policy constrained by the physical position of Bob and Ana. The use of Foreign Constraints pose the problem of introducing a separate entity to act in the role of a trusted PEP. As a solution we leverage our UCPF and introduce a special component called *Sentry*.

The UCPF is located between trusted and untrusted entities and thus can act as a mediator, see Figure 2. Additionally, the Sentry component is part of the UCPF and acts in the role of a trusted PEP for the tracked target. The Sentry is attached to the target infrastructure. In this environment context providers forward raw data to the *Sentry* where the data is filtered and/or transformed, see also section 4.2. The Sentry is provided with mechanisms to query and retrieve Foreign Constraints as needed. In our example scenario, Bob is requested to reveal his activity to Ana. In order to reach a decision what to do, Bob's Sentry has to consider the Foreign Constraints regarding Ana. Therefore, it queries Ana's Sentry. This particular instance of a Sentry is registered as the Context Provisioning Service for Ana. Therefore, Bob's Sentry can obtain a reference via the

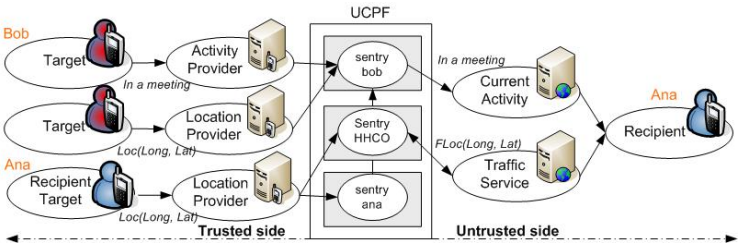

Fig. 2. Dataflow in the UCPF.

UCPF's service registry. After querying the position of Ana, Bob's Sentry decides on a suitable course of action. In this model, each target has his own Sentry.

4.2 Architecture Overview

The Figure 3 shows a sketch of the UCPF architecture. The upper layer is the Privacy Layer that implements all the functions of policy distribution architecture [21], additionally incorporates two functionalities: a) the capability of transforming context information and b) a central distribution point for context information under the target's privacy criteria. Sentry is a component integrated in the Privacy Layer in charge of the PDP and the PET roles, see Section 4.1, and acting as the context provisioning proxy.

The UCPF architecture provides the means to publish the Sentry as a service that offers target's context information in a UCPF's service registry. In order to respect target privacy preferences, all third party services (service providers, other sentry services, etc) should retrieve a target context only from the appropriate Sentry.

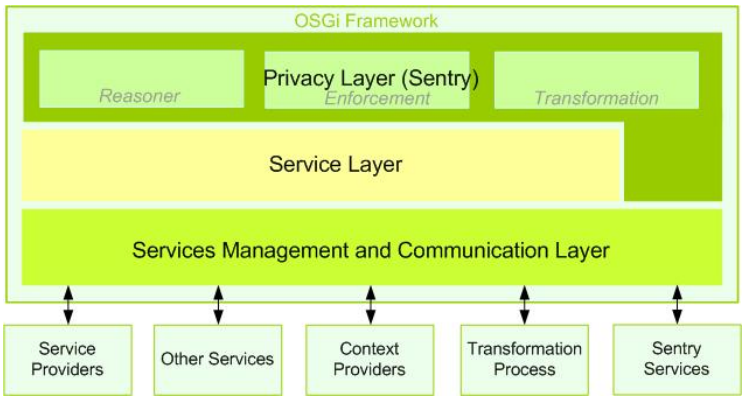

Fig. 3. Architecture Overview.

One of the key features of this framework is the possibility to interact with a wide variety of services. The UCPF architecture manages and intercommunicates different

entities such as services providers, other services (that do not use context information), context providers, transformation process, and sentry services (to retrieve Foreign Constrains). That infrastructure demands flexibility to add and remove services without disturbing amount each other from the trusted and untrusted side of the processing chain, see Figure 2, therefore this framework should offer the possibility of remote deployment and management of services.

A first prototype of the UCPF is implemented on the top of the Open Service Gateway Initiative (OSGi) framework. The OSGi provides a strong environment for multiple Java-based components to work on a single Java Virtual Machine (JVM). The OSGi also adds the capability to manage the life cycle of the software components from anywhere in the network.

The UCPF supplies a secure ambit to enforce authorization, and transformation policies where Transformations are only knows by the target's Sentry, and also supplies an environment where any service may retrieve target context information respecting the privacy criteria.

5 SeT language / Transformation Policy

The language choose to define policies in the UCPF is Rei [9]. Rei is a policy specification language in OWL-Lite that allows users to develop declarative policies over domain specific ontologies. Rei is used to describe positive and negative permissions and obligations of entities in the policy domain. In order to define transformation policies the Rei policy language is extended, we have created the SeT language to define transformation policies. The Figure 4 shows the most important classes and properties of SeT. We use the OWL-S [2] to describe a transformation and its corresponding inputs, therefore SeT imports Rei and OWL-S ontologies.

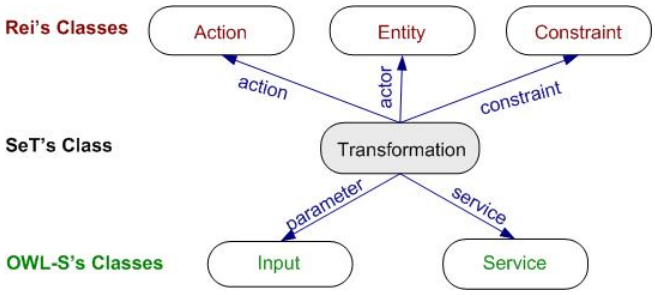

Fig. 4. SeT language's classes and properties.

Each transformation available in the UCPF is specified and registered as a service in the OSGi framework. Transformations can be seen as services offered by the UCPF. We model Transformations as processes with the *Process Ontology*, one of the three main parts of the OWL-S. The goal is to automatically applied a transformations process when a transformation policy is enforced.

We consider in our first prototype as Transformations: spatial obfuscation, identity an data abstraction in K users, and the use of white lies. White lies involves enforcing Transformations to calculate virtual context data which will be deliver instead of the real context. Each transformation has a different set of inputs that are specified with SeT.

6 Conclusions and outlook

In this paper, we presented our User-Centric Privacy Framework (UCPF) which aims at protecting a user's privacy based on the enforcement of privacy preferences. They are commonly expressed as a set of constraints over some set of context information. We introduced two novel abstractions, namely *Transformations* and *Foreign Constraints*, and showed how they can be used to extend the expressiveness of privacy policies. Transformations are an important tool for enforcing a certain kind of privacy preferences, like obfuscation and introducing uncertainty. Foreign Constraints on the other hand are a truly novel concept for formulating preferences, which include the actual context of the recipient or other *external* users, over some privacy relevant information.

Both concepts are founded on the generic UCPF framework and the *SeT language* for formulating and subsequent enforcement of privacy preferences. The UCPF provides a secure environment where Transformations will be only known by the user, allowing for the integration of obfuscation, anonymization, and *white lies* in the policy domain. So far, policies cannot be used to express the need of transforming data before accepting and releasing context information to a service. Currently, in parallel to the refinement of the concepts of Transformations and Foreign Constraints, we are working on the further extension of the SeT language.

We designed the UCPF in a way that *Sentry*, as part of the UCPF, can act as a context provisioning proxy for third party services. This way a user's context is accessible by external entities but only under the control of the tracked person's privacy preference.

Part of the ongoing and future work is to complete the implementation of our UCPF prototype, which currently is in an early stage. We are also planning to use the UCPF to present the concept of *white lies* as potential privacy mechanism. White lies involves Transformations which calculate virtual context data which will be deliver *instead* of the real context data. In future work we plan to analyze the use of static transformations (for the same inputs the same output) and dynamic transformations that may generate different outputs for the same set of inputs, inhibiting the use of reverse-engineering techniques to a large degree. An important issue in further works will be to explore ways to provide policy conflict resolution in a hierarchy of sets of policies within the same and different instances of Sentries.

In summary, we believe that the approaches presented in this paper add significantly to the field of privacy protection of individuals in pervasive computing environments. Obviously, parts of the approaches presented here are work-in-progress and need further investigation, e.g. conflict free formulation of policies or white lies. However, the foundations of Transformations and Foreign Constraints are sound and together with the generic UCPF build are strong foundation for personal privacy protection.

References

1. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.
2. T. O. S. Coalition. Owl-s: Semantic markup for web services., November 2004.
3. L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification. W3C Recommendation, Apr. 2002.
4. J. R. Cuellar. Location information privacy. In B. Sarikaya, editor, *Geographic Location in the Internet*, pages 179–212. Kluwer Academic Publishers, Norwell, MA, 2002.
5. N. Damianou, N. Dulay, E. Lupu, and M. Sloman. Ponder: A language for specifying security and management policies for distributed systems, 2000.
6. M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive*, pages 152–170, 2005.
7. M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, May 2003.
8. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSYS 04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189. ACM Press, 2004.
9. L. Kagal. Rei ontology specifications, ver 2.0, 2004.
10. L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks.*, September 2003.
11. A. Küpper. *Location–based Services — Fundamentals and Operation*. John Wiley & Sons, Aug. 2005.
12. M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In *In Proceedings of the 4th International Conference on Ubiquitous Computing*, pages 237–245. LNCS No. 2498, Springer-Verlag, September 2002.
13. M. Langheinrich, L. Cranor, and M. Marchiori. Appel: A p3p preference exchange language. W3C Working Draft, Apr. 2002.
14. G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.
15. A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In H. Federrath, editor, *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA*. Springer, 2001.
16. H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, and J. Polk. A document format for expressing privacy preferences for location information. draft-ietf-geopriv-policy-08.txt, February 2006.
17. M. Sloman. Policy driven management for distributed systems. *Journal of Network and Systems Management*, 2:333–360, 1994.
18. E. Snekkenes. Concepts for personal location privacy policies. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce, Tampa, Florida, USA*, pages 48–57, New York, NY, USA, October 2001. ACM Press.
19. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.
20. M. Weiser. The computer for the twenty-first century. scientific american, pp. 94-10, September 1991.
21. R. Yavatkar, D. Pendarakis, and R. Guerin. Rfc 2753 - a framework for policy-based admission control, January 2000.