Revised version (1) of paper to appear in the proceedings of the *ACM Conference on Electronic Commerce (EC'01)*, 14-17 October 2001 Tampa, Florida, USA

Concepts for Personal Location Privacy Policies

Einar Snekkenes
Norwegian Computing Center
P O Box 114, Blindern
N-0314 Oslo, Norway
+47 22 85 26 10
einar.snekkenes@nr.no

ABSTRACT

A Location Based Service (LBS) is a service where knowledge of the location of an object or individual is used to personalise the service. Typical examples include the E911 emergency location service in the US and 'Where is the nearest xx' type of services. However, since these services often may be implemented in a way that exposes sensitive personal information, there are several privacy issues to consider. A key question is: "Who should have access to what location information under which circumstances?"

It is our view that individuals should be equipped with tools to become in the position to formulate their own personal location privacy policies, subject to applicable rules and regulations.

This paper identifies concepts that may be useful when formulating such policies. The key concept is that of an observation of a located object. An observation typically includes the location, the identity of the object, the time the observation was made and the speed of the object. The idea is that the individual should be able to adjust the accuracy at which these observations are released depending on parameters such as the intended use and the identity of the recipient.

We provide fragments of a language for formulating personal location privacy policies and give some small examples illustrating the kind of policies that we have in mind.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General – *Security and protection (e.g., firewalls).* K.4.1 [**Computers and Society**]: Public Policy Issues – *Privacy.* C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *Wireless communication.*

General Terms

Management, Security, Legal Aspects.

Keywords

GSM, EDGE, GPRS, UMTS, PCS, iMode Privacy policy

EC'01, OCTOBER 14-17, 2001, TAMPA, FLORIDA, USA.

1. Introduction

In many countries, the cellular phone has become an item that the society cannot do without. For some individuals it has become like an essential item of clothing, it's always in a pocket, handbag or fixed to an item of clothing. In the very near future, one would expect that these devices in many situations could be located with an accuracy of less than 1 meter! With similar technology, which is available today, it has been shown that one can obtain a spatial accuracy of around 2-meter [3]. This technology and its widespread deployment may produce many new business opportunities. According to SUN [14], one can conceive new location based services (LBS), including m-commerce targeting consumers, businesses and government. The types of information that will be part of these services include positions, events, distributions, service points, routes, overview information, directions etc. For example:

- Where is the nearest visible landmark?
- Where are my dispatch trucks?
- What is the current traffic pattern?

There are several military applications relating e.g. to logistics and the location of submarines.

Location technology comes in two flavours: tracking and positioning. Using tracking technology, the position is computed by some external entity (e g the network operator). The term positioning is often used when referring to technology allowing the located object to compute its location by itself (e g GPS).

In the case that positioning technology is used in conjunction with local mapping software, privacy may not be an issue. However, because of the widespread use of relatively cheap cellular phones, many of the location services will be based on tracking technology, where the provider of the LBS will be different from the location provider. In this setting, clearly location privacy is worth some considerations.

Several studies show that many individuals are concerned about personal privacy[5]. According to Robinson[11], privacy is the top remaining issue for LBS. In some countries there is legislation that requires consent for processing sensitive personal information (such as location data). This includes the 'Personal Data Act' ('Personopplysningsloven') in Norway[8].

We appreciate that location privacy may be a difficult issue. For example, asking the owner of a cellular phone if he is concerned about location privacy, we most certainly would hear a 'yes'. However, if we ask the same individual if he usually takes any measures to preserve his location privacy e g by taking the battery out off his phone to prevent tracking, the answer most likely would be 'no'. This may be because

- The cost of privacy (a few key presses and not being reachable in real time) is judged to exceed the perceived benefit of increased privacy.
- Individuals are not aware of the fact that the network operator may be able to track their cellular phone.

The key issue we explore in this paper is

"Who should have access to what location information under which circumstances?"

The aim of this paper is to identify concepts which may be useful when constructing tools for letting individuals formulate the personal location privacy policy they feel is most appropriate.

The general setting of our work is that of some entity (e.g. cellular phone network operator) which obtains fairly accurate location data. This data is then 'sanitised' (to reduce its accuracy) before it is released to the entity that delivers the LBS. We believe that individuals may be more willing to approve the release of sensitive information if there are reasonable mechanisms for enforcing the need-to-know principle.

In some sense, attitudes, privacy rules and regulations can represent barriers to the deployment of LBS. If one can devise a way of improving trust and establishing informed consent, e g by offering a location privacy policy language for controlling information release relating to location observations, one may achieve the following:

For the individuals: One place to define the personal privacy policy, and one place to maintain this policy.

For service providers: Aspects relating to consent is handled 'once and for all', reducing barriers.

The remainder of this paper is organised as follows. We first give a brief overview of some related work, highlighting the relationship with the work presented in the remainder of the paper. We then give an overview of the context where our work may be useful. Having identified the context in which the personal location privacy policy would be used, we identify and explain some concepts that may be useful when expressing personal location privacy policies. To make the ideas somewhat more concrete, we then define fragments of a language for expressing policies. We illustrate the use of the language by means of several examples. Finally, we suggest some issues that deserve further work.

2. Related work

Privacy issues surface in many situations.

- Information regarding individuals that is communicated between third parties (e g medical records).
- Information that is destined for me (e g incoming phone calls).
- Information that I may emanate, both implicitly and explicitly (e g signals from my cellular phone)

In this paper, we focus on the latter. The control of flow of information to users (personal reachability) in a mobile context is addressed e g in [9].

There is some commercial interest in sending out location based advertising. This can be implemented in several ways. An

implementation based on broadcast and recipient anonymity is more of a reachability issue [9]. However, if location data is used either in the routing or selection/generation of the advert, we have a privacy issue inside the scope of our work.

There are several papers addressing anonymity and privacy in a network exposed to threats from traffic analysis, see e.g. [6][2][12][13][1]. We consider a slightly different setting, where all sensitive routing is performed on a 'trusted' dedicated cellular phone network. Our focus is on the specification of policies governing the limited disclosure of location information.

In [7], Leonhardt and Magee describe a system that implements location privacy policies based on Lampson's access matrix and the Bell and LaPadula (BLP) security labels. In their paper, they define the set of subjects to include individuals and the set of objects to include both individuals and locations. A particular cell contains rights corresponding to operations such as 'testForColocation'. By assigning security levels to both subjects and objects, the releasability of information can be specified using the traditional 'dominates' relation. By associating 'high' ('low') security levels with accurate (inaccurate) location specifications, it seems possible to formulate 'partial information release' policies. The paper also discusses anonymity policies, and suggests that the identity of a located object can be specified as a point in an identity hierarchy. In [7], the focus is on privacy policies formulated by some third party (e g x is allowed to check if y is at location z). We will be looking at privacy issues from the view of an individual (x is allowed to view my location).

In a sense, our work combines the access matrix concepts and the 'reduced release' BLP policies that can be constructed using the MAC lattice operators, but does so without explicit reference to security labels. In addition to spatial accuracy of location information, we introduce the concept of temporal accuracy, and allow the policy decisions to depend on the current time. We also introduce the concept of roles (service provider, location provider, service consumer, service requestor, and service initiator) and the purpose of information usage.

Within The World Wide Web Consortium (W3C), work on Privacy is being managed as part of W3C's Technology and Society domain. The W3C has several privacy-related activities, including P3P [15][16][4].

The P3P initiative assumes a setting in which there is some merchant or service provider that communicates with some user. During the 'initialisation' phase, the merchant defines how he will handle and use information collected from the users browser. This policy is translated to XML e.g. using some policy editor. This policy file is then installed on the server together with a policy reference file specifying where the policy file can be found. The user specifies her privacy requirements to her web browser. When the user accesses the web server, she first gets the servers P3P policy file. Her web browser matches this policy with her preferences, disconnecting or giving some form of notice if the received policy is unacceptable.

It may be informative to briefly discuss differences and similarities between P3P and our approach. Our approach is different to P3P in several ways.

- P3P lets service providers offer the user a policy that the user has to accept as is, or disconnect from the server. Our approach is to offer the user some means of controlling the accuracy of the data released to the service provider.
- The dynamic aspect of P3P includes that of click streams. The dynamic aspect of a location privacy policy includes the movement of located objects in space.
- With respect to policy enforcement, using our approach, the responsibility is split between the location provider (reduce accuracy) and the service provider (restrict usage to stated purpose). In P3P, the policy enforcement seems to be at the discretion of the merchant.

P3P focuses on privacy relative to a particular activity (access of a service through a WEB browser), most likely over a limited period of time. Consider a LBS, where a group of friends have agreed to mutually disclosing their location through the weekend. In a P3P setting, the collection of sensitive data is restricted to the time the user is actually using the browser. With respect to LBS, sensitive data may be collected not just when the user is making a phone call, but all the time the phone is on. Location privacy thus is more of 365×24 issue, suggesting that breach of location privacy may be considered very invasive.

There are also several similarities between P3P and our ideas. In both cases, individuals specify their own privacy requirements, and individuals may be willing to release some sensitive data in order to obtain a better service. Both approaches are concerned with the following issues:

- Who is collecting the data?
- What is the purpose of the collection?
- Who are the data recipients?

The scope of the P3P initiative is somewhat wider, in that it also includes issues such as dispute resolutions.

Some LBS technology vendors include some form of privacy support. For example, in WISE 2.0, XMARC INC offers a Secure Profiler[17]. It is claimed that it

- Delivers privacy protection for users and their profiles.
- Allows users to set-up secure groups, manage the members of the group and provide user access to group location information.
- Allows groups to be set up with geographic boundaries including automatically generated alerts for boundary violations.

3. Architectural context

The application we have in mind is that of a 'Universal' location service, where location data typically is produced' by the GSM/ GPRS/ UMTS network operator. We assume that the location provider has some interest (e.g. for compliance with privacy laws) in letting subscribers have some say in what location data is released and to whom.

There is at least one country (Norway), where a government organisation (Brønnøysund Register Centre, http://www.brreg.no/english/) runs a (free) public register service, where individuals can register that they don't want to receive direct marketing or sales phone calls or mail. (http://www.brreg.no/oppslag/reservasjon/index.html). In some sense, this offers individuals of Norway a privacy policy in the sense of [9]. In this paper, we build on this idea. We consider a setting including the following entities:

- Personal Location Privacy Policy: Statement of what can be released to whom and when. Each located object will have associated a policy.
- Policy custodian directory: Where to find the policies.
- Policy custodian: Where the policy is stored and possibly also enforced. The set of permitted operations may include read, write, modify, and query etc. depending on the identity of the requesting entity.
- Location provider: Entity providing the location data. Any release of data should be subject to the policy.
- Service provider: Entity that is combining location data with other data to produce some service.
- Service consumer: Entity to which the service is presented for consumption.
- Service initiator: Entity that would like and/or accept that the service is produced.
- Service requestor: Entity that makes a request to the service provider for the service to be produced.
- Located object: The entity whose location data will be required to deliver the service.
- Owner of located object: The entity that owns the located object.

Our idea is that there should be established some (free central public) register (the Policy custodian directory) at some well known address, that for each located object contains a pointer to the location where a personal location privacy policy for that located object is stored. A location provider would then be obliged to receive 'release approval' from the policy custodian before any location data could be released. In many cases, it may be convenient to make the network operator the custodian. As the policy itself might be sensitive, some access control measures to the policy might be required. Clearly there are several ways to ensure that release of location data is in accordance with personal policy. The following represents one possible sequence of events:

- The location data requestor (e g the service provider) sends a location query to the location provider. The request may include several located objects.
- The location provider must identify the currently applicable personal location policy by contacting policy custodian directory and then the policy custodian. For performance/scalability reasons, one may want to do this periodically rather than on request. In many cases, it may be acceptable to let policies have e.g. a one-week expiry period.
- The location provider forwards the request to the custodian, which responds as appropriate.
- Depending on the response, the location provider responds to the location data requestor with the location data according to the response received, which will be in accordance with the privacy policy.

The figure below illustrates by means of an example a possible order in which the various requests can be made.

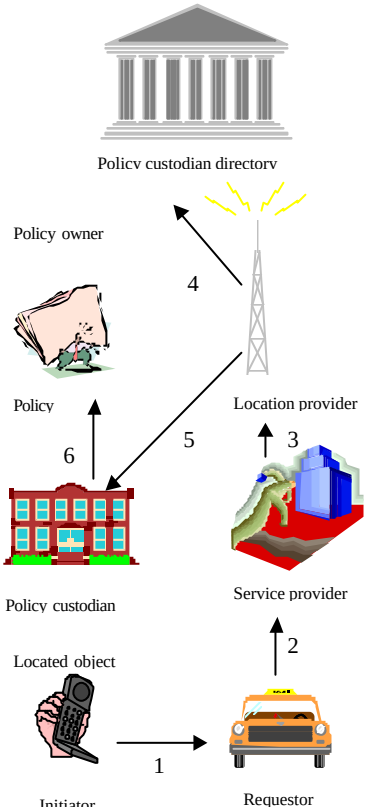

Figure 1. Service request: Sequence of interactions

One interesting issue worth some consideration is that of policy enforcement. Many entities may have access to sensitive location data. How are we to know that all entities will comply? One possibility could be to use cryptographic techniques, and in

addition, including in the requests for information, some (legally binding) form of 'promise to conform'.

4. Policy concepts

One can view location privacy policies at several different levels:

- User interface
- Logical structure
- Machine representation

This paper focuses on concepts relevant to the logical structure of the policy. By policy, we mean a specification of what location data can be released to whom and when.

Privacy is closely related to the ability to control the flow of information. Traditional 'yes'/'no' type access control models seem to be too crude for our needs. The purpose of the policy is to specify what data can be released. The central piece of information to release is that of an observation. An observation includes the following elements:

- The time the observation was made.
- The location the located object was observed at.
- The speed the located object was observed to have at the time of the observation.
- The identity of the located object being observed.

For simplicity, when specifying location, time etc, we ignore the issue of 'units' and restrict our attention to discrete sets, just assuming that there is a 1-1 function that can be defined to perform a 'reasonable' mapping. A summary of our notation is given in the appendix.

We can model observational accuracy using lattice structures. Before we describe how we intend to model identity, location etc, it may be useful to give a brief description of the key features of a lattice.

Intuitively, a lattice may be viewed as a collection of points, where some of the points are connected with straight lines and where no lines are horizontal. There is a unique element 'on the top' ('on the bottom') which can reach directly or indirectly each and every other point when traversing strictly down (up).

A lattice is a partially ordered set where we have a unique top and bottom element and where each pair of elements have a least upper bound and a greatest lower bound. For the purpose of this paper, the intended 'semantics' of these is given below.

a = b : This should be interpreted as 'a is less defined than b'.

lub(a,b) : Least upper bound. If a, b are elements of a lattice, this term denotes the element of the lattice which is no less defined than the elements a, b, but not more defined than 'absolutely' necessary. We can define the greatest lower bound (glb) analogously.

T : The top element, i.e. the element denoting the 'most' defined value.

B : The bottom element, i.e. the element denoting the 'least' defined value.

We now give the lattices corresponding to degree of accuracy of time, location, speed and identification. If an observation was made at time t1, and the current time is t2, (t1= t2), we can 'sanitise' the temporal aspects of this observation by instead disclosing one member of the following set:

$TL(t1,t2) = \{x \mid t1 \in x \wedge x \subseteq \{t \mid t = t2\}\}$ // The elements of the lattice.

$T = \{t1\}$

$B = \{x \mid x = t2\}$

$x = y = y \subseteq x$

$glb(x,y) = x \cup y$

$lub(x,y) = x \cap y$

With respect to spatial accuracy, we restrict our attention to geometric locations. We can model spatial accuracy as presence inside a (some collection of) 3 dimensional space. The smaller the volume of this space, the more accurate the observation will be. If we let a triple (x, y, z) denote some body at position (x, y, z) having volume 1, we can model the decreased spatial accuracy by means of some 'bigger' body, enclosing (x, y, z).

$SL((a,b,c)) = \{s \mid (a,b,c) \in s\}$ // The elements of the lattice.

$T = \{(a,b,c)\}$

$B = \mathbf{N} \times \mathbf{N} \times \mathbf{N}$

$x = y = y \subseteq x$

$lub(l1,l2) = l1 \cap l2$

$glb(l1,l2) = l1 \cup l2$

Let F be the body ('volume') enclosing my office. Then if I've been observed at location (a, b, c), which is outside my office, we can model this as the set $B \setminus F$.

The specification of the identity of a located object i, is modelled as a set, where the *precise* identification of i, is modelled as the singleton set {i}. Complete anonymity, which is a non-observability, is modelled as the empty set. Anonymous existence is modelled as **N**.

$IL(i) = \{x \mid i \in x \wedge x \subseteq \mathbf{N}\}$ // The elements of the lattice

$T = \{i\}$

$B = \mathbf{N}$

$x = y = y \subseteq x$

$lub(i1,i2) = i1 \cap i2$

$glb(i1,i2) = i1 \cup i2$

Partial identification of individuals can then be done by defining sets corresponding to sex, occupation, nationality, employer etc.

The set of all identity specifications for all individuals, can then be defined as follows:

$$\text{Identity} = \bigcup_{i \in \mathbf{N}} \text{IL}(i) \quad // = \text{set of subsets of } \mathbf{N}$$

where $\bigcup$ denotes the operator for distributed union.

An observation will also include a speed component. Assuming we have an observation of the speed of the located object having some value between v1' and v2', then we can define a lattice as follows:

$$\text{VL}(v1',v2') = \{s \mid \{v \mid v1' = v = v2'\} \subseteq s\}$$

$$T = \{v \mid v1 = v = v2\}$$

$$B = \mathbf{N}$$

$$v1 = v2 = v2 \subseteq v1$$

$$\text{glb}(v1,v2) = v1 \cap v2$$

$$\text{lub}(v1,v2) = v1 \cup v2$$

If the observation was made at time t1, the current time is t2, the located object was observed at location (a, b, c), the located object had identity i, the speed was known to be between v1 and v2, then the set of releasable observations form a lattice consisting of the following elements:

$$\text{OBS}(t1,t2,(a,b,c),i,v1,v2) = \text{TL}(t1,t2) \times \text{SL}((a,b,c)) \times \text{IL}(i) \times \text{VL}(v1,v2)$$

The top, bottom, ordering and bounds can be defined in the obvious way.

The set of all releasable observations can the be defined as the set OBSERV, where

$$\text{OBSERV} = \bigcup_{t1,\, t2,a,b,c,i \in \mathbf{N}} (\text{OBS}(t1,t2,(a,b,c),i,v1,v2))$$

Having characterised the concept of an observation, and shown how different degrees of accuracy can be modelled as a lattice, we now take a closer look at the motivation for why somebody would like to use location data, that is, purpose of use. The establishment of the intended usage of the query response may be somewhat difficult to formalise. In some countries, lawful processing of personal information must be based on consent relative to a specified purpose of usage. In some cases, the actual purpose may change over time. Here are some examples of 'purpose' specifications (not necessarily acceptable from a legal point of view):

- Collection of data for resale to the highest bidder.
- Anonymous statistics
- Validation and follow-up on agreements and legal obligations such as
    - Speed limits
    - Parking
    - Insurance policy requirements (distance driven, car usage etc)
- Provision of information on second hand cars available from local garages.
- Targeted information on changes to local transport services.
- Identification of people in the same area having similar interests.
- Generation of 'You are here' type of maps for delivery to located object only.

One possible way of formalising 'purpose' could be to make informal description of some suitable set of 'purposes'. The ordering of these elements can then be defined formally, introducing new elements such that the set forms a lattice. For example, the top (bottom) element could be defined as 'Any use' ('Will not be used').

Relative to a single located object, there are essentially two kinds of queries to consider:

- Where in ... are you?
- Are you at ...?

In the first case, the query is a request for some arbitrary number of bits of information, where as in the second case, we are requested to provide a single bit of information. However, in both cases, the motivation for making the query is to establish a relationship between the identity of an individual, a location and the time when the relationship was observed.

Then, to summarise, the context of a query consists of the following:

CurrentTime : **N**

ServiceProvider,

ServiceConsumer,

ServiceInitiator,

ServiceRequestor: Identity

Purpose : Lattice

QueryType : Information | Confirmation

QueryExpectation : Observ

The location provider provides raw location data. The observation made by the location provider is a function of the identity of the located object and the current time.

$$\text{Observations} = \text{LocatedObject} \rightarrow \text{Time} \rightarrow \text{Observ}$$

Having specified the structure of the information to be released, the time has come to take a closer look at the policy.

There will be one policy associated with each located object. A decision on what information to release depends on the context in

which the location query was made. Each located object will have associated a policy. The set of policies can then modelled as a family of functions characterised by the following signature:

Policies = Observations →

LocatedObject →

Context → Observ

Each located object has associated an owner. The policy of a located object may include reference to the location of all located objects being owned by the owner of the located object. The ownership of a located object is uniquely defined for each located object.

Ownership = LocatedObject → Owner

If we have some syntactic representation P of some policy, we may define its 'meaning' with some function eval(.) where eval[P] ∈ Policies.

Using the ideas described above, we now show how to introduce mandatory policies. By a mandatory policy, we mean a policy that defines some minimum requirement. In our case, this translates to some ceiling on the accuracy of the releasable observations. Using the concepts sketched above, there are at least two ways to accommodate this.

Let P1 be the policy defining the mandatory policy, that is, the maximum accuracy of information releasable in any given situation. Let P2 be a discretionary policy. Then we may define the policy

P1 JOIN P2

as follows:

eval(P1 JOIN p2) =

? b l c . glb(eval(P1)(b)(l)(c), eval(P2)(b)(l)(c))

If we interpret a policy as a rule for reducing the accuracy of observations, we can feed the 'output' resulting form the evaluation of the mandatory policy as the input to the more discretionary policy as follows:

eval(P1 FILTER P2) =

? b l c. eval(P2)(

? l2 t. eval(P1)(b)(l2)(c +CT t))(l)(c)

where c + CT t denotes the context c, where the field in c corresponding to CurrentTime has been replaced by the value t. The idea is as follows: When somebody (say at t1) approaches the component enforcing P2, there is likely to be one or more requests to the component enforcing P1 (say at t2), then it is t2 rather than t1 which is to be used when deciding what can be released

5. Fragments of a language for formulating location privacy policies

Having identified the basic concepts, we now show how policies can be defined. In practise, I may own several located objects including a car, a boat, a moped, a cat, a dog, a cellular phone etc. My willingness to release the whereabouts of my car may depend on whether or not I'm in the car. For example, if the car is in motion, and I'm not in it, it may be acceptable for me to let the insurance company or private security firm know this. However, if I'm in the car, I don't want them to know where I'm driving. Assuming that there is a very high probability that I always bring my cellular phone with me, this kind of policy will require the ability to relate the location of several of my located objects. Consequently, I may want to define one privacy policy for each of the located objects I own, with the opportunity to include the location observation of all located objects I own in all of the policies.

For each of the located objects, there will be a 'default' release policy, specifying the location information I'm willing to release in all unspecified situations. For example, we have the 'paranoid' location specification (B(.),B(.),B(.)) and the naive location specification (T(.), T(.),T(.)). The remaining aspects of the policy will consist of an ordered list of pairs, each pair consisting of a guard and a response. The intended meaning of the tuple being that the guard specifies the situation when the response part can be released.

Fragments of a language are specified using a BNF like notation below.

<Policy> ::=

<Owner Id>

(<Located Object>

<Observation>

// The default response

(<Guard> <Observation>)+)+

The response of a query will be some observation. An observation can either be

- the observation provided by the environment (typically the location data provider, or some more 'mandatory' policy),
- some version of the observation having reduced accuracy (i.e. lower down the lattice)
- the greatest lower bound of two observations,
- some observation constructed from a quadruple of temporal, spatial and identity specifications (or possibly a lie?) or
- the observation expected by the entity making the query.

For readability, we let _ on the right hand side denote recursion.

<Observation> ::=

RawObservation(<Located object>)(<Time>)

| DowngradeObs(_,**N**)

| glb(_,_)

| (<Temporal spec> <Spatial spec>

<Identity spec> <Speed spec>)

| QueryExpectation

There will be situations when the decision on what information to release depends on the observation associated with e.g. the ServiceRequestor. When defining the semantics of 'RawObservations', it will be convenient to let this denote the raw location data from the location provider in those cases that the requestor and located object have the same owner, and the observation subject to 'filtering through' the appropriate security policy in all other cases. The downgrading operator, takes some observation (for parts of observation, see below), and returns some less defined element. The second parameter specifies the size of the accuracy reduction.

The guard can be constructed using standard logical operators and various tests involving temporal, spatial etc. aspects of observations.

<Guard> ::= _ ^ _

| ~ _

| <Temporal test>

| <Spatial test>

| <Identity test>

| <Speed test>

| <Purpose test>

| <Observation test>

| <Query test>

The various atomic tests can be constructed using the lattice orderings, equality and various 'shorthand' and comparison predicates. For example, it may be useful to have some predicate over pairs of locations, specifying that the located objects are in the vicinity of each other. Essentially, the purpose of these predicates is to make policies more succinct.

<Temporal test> ::=

<Temporal spec> (= | =) <Temporal spec>

| <Temporal pred> <Temporal spec>+

// e g 'IsSunday(.) etc

<Spatial test> ::=

<Spatial spec> (= | =) <Spatial spec>

| <Spatial pred> <Spatial spec>+

// e g 'IsInOslo(.) etc

<Identity test> ::=

<Identity spec> (= | =) <Identity spec>

| <Identity pred> <Identity spec>+

// e g 'IsMyFriend(.)' etc

<Speed test> ::=

<Speed spec> (= | =) <Speed spec>

| <Speed pred> <Speed spec>+

// e g 'IsWalkingSpeed(.)' etc

<Purpose test> ::=

<Purpose spec> (= | =) <Purpose spec>

| <Purpose pred> <Purpose spec>+

// e g 'IsAcceptablePurpose(.)' etc

<Observation test> ::=

<Observation> (= | =) <Observation>

<Query test> ::=

(Confirmation | Observation) = QueryType

The terms in the various tests above are constructed from the 'variables' offered by the context, constant terms, the glb(.) operator on the lattices, projection operators on observations and the downgrading operator.

<Temporal spec> ::= glb(_,_)

| <CurrentTime> // From context

| <Any reasonable temporal constant ...>

| <Observation> .Temporal

| DowngradeTemporal(_,**N**)

<Spatial spec> ::= glb(_,_)

| <Any reasonable spatial constant ...>

| <Observation> .Spatial

| DowngradeSpatial(_,**N**)

<Identity spec> ::= glb(_, _)

| <Any reasonable identity constant ...>

| <Observation> .Identity

| ServiceProvider // In the current context

| ServiceConsumer // ...

| ServiceInitiator // ...

| ServiceRequestor // ...

| DowngradeIdentity(_,**N**)

<Speed spec> ::= glb(_,_)

| <Any reasonable speed constant ...>

| <Observation> .Speed

| DowngradeSpeed(_,**N**)

<Purpose spec> ::= glb(_,_)

| <Any reasonable purpose constant>

| Purpose

To illustrate how the concepts and language described in the previous chapters can be used we offer some examples. We first include some policies, which specify that information can be released to particular individuals.

"I will let my insurance company know the location of my car when it's driven without me in it. The location of my car or my cellular phone should not be visible in any other situations"

Owner: JohnSmith

Located object: John's cellular phone

Default response : (B,B,B,B)

Located object: John's car

Default response : (B,B,B,B)

ServiceRequestor = {MyInsuranceCompany}

∧ ~(IsDrivingSpeed(RawObservation(John's cellular phone)(CurrentTime).Speed))

∧ IsDrivingSpeed(RawObservation(John's car)(CurrentTime).Speed)

?

RawObservation(John's car)(CurrentTime)

"A request for my location in situations when I initiate the service should return location data which is as accurate as possible. My friends and relatives should know whether or not I'm in London."

Owner: JohnSmith

Located object: John's cellular phone

Default response : (B,B,B,B)

ServiceInitiator = {JohnSmith}

?

RawObservation(John's cellular phone)(CurrentTime)

ServiceRequestor = ServiceConsumer

∧ ServiceInitiator = ServiceConsumer

∧ IsFriendOrRelative(ServiceConsumer)

∧ QueryType = Confirmation

∧ (London) = (RawObservation(John's cellular phone)(CurrentTime).Location)

?

(CurrentTime, London, John Smith, B)

John Smith would allow his wife to know if he's in his office or not

Owner: JohnSmith

Located object: John's cellular phone

Default response : (B,B,B,B)

ServiceRequestor = ServiceConsumer

∧ ServiceInitiator = ServiceConsumer

∧ ServiceConsumer = {John Smith's wife}

∧ QueryType = Confirmation

∧ (John's office) = (RawObservation(John's cellular phone)(CurrentTime).Location)

?

(CurrentTime, John's office, John Smith, B)

On Friday night, my friends are allowed to see where I am, and at what speed I'm moving.

```
Owner: JohnSmith
Located object: John's cellular phone
Default response : (B,B,B,B)

ServiceInitiator = ServiceConsumer
∧ IsJohnsFriend(ServiceConsumer)
∧ IsFridayNight(CurrentTime)
        ?
(RawObservation(John's cellular phone)(CurrentTime))
```

If I'm looking for a taxi, I'm perfectly happy to let the taxi drivers in the vicinity know where I am. They are however not permitted to get my identity.

```
Owner: JohnSmith
Located object: John's cellular phone
Default response : (B,B,B,B)

ServiceInitiator = {John Smith}
∧ ServiceConsumer = ServiceRequestor
∧ IsTaxiDriver(ServiceConsumer)
∧ (Provide Taxi service) = Purpose
∧ InTheVicinity (
   (RawObservation(John's cellular phone)
                                  (CurrentTime).Location),
   (RawObservation(ServiceInitiator)(CurrentTime).Location))
        ?
((RawObservation(John's cellular phone)(CurrentTime).Time),
(RawObservation(John's cellular phone)(CurrentTime).Location),
(Somebody),
(B))
```

Note, that for this to have the desired effect, taxi drivers should have fairly liberal privacy policies if they are to have any chance of obtaining John Smith's location.

There may be an interest in services showing if the beach is likely to be crowded and what clubs or restaurants have the most visitors. Considering that there will be scenarios when this kind of statistics may give rise to serious breach of privacy, some individuals may prefer to be invisible. If location providers faithfully implement location privacy policies, individuals will have the opportunity to decide if they want to be invisible, anonymous or distinguishable.

6. Implementation issues

It may well be the case that a naive implementation of our architecture will be terrible inefficient and that it will not scale particularly well. We can improve performance by using policy caching, e.g. by accepting that policy enforcement is relative to some policy which was in effect say during the last 48 hours rather than the current policy. The use of 'policy push' to the location provider may also improve performance.

With respect to communication overhead, the architecture requires very little (if any) extra communication on the inherently wireless channels (i.e. to the terminals).

It seems that the most likely potential bottleneck with the proposed architecture may be that of evaluating policies, i.e. deciding what information to release the service providers. However, we would expect that this evaluation is linear in the size of the policy, or at least could be made linear by placing minor restrictions on the policies. If we assume that the complexity of evaluating a single policy is constant in all factors except the complexity of the policy itself, the architecture ought to be scalable, and the performance can be improved by evaluating different policies in parallel.

A close co-operation (or integration) between the policy custodian and the location provider may improve performance.

In countries, where access to location data requires explicit consent, assuming our approach is legally acceptable, it may enhance service performance if the service provider is able to obtain location data before it receives the request for a service.

7. Conclusions and further work

We have identified some concepts that may be useful when formulating personal privacy policies. Fragments of a language intended for formulating personal privacy policies has been presented. Several examples illustrating the use of the language has been given.

There are several issues that deserve further study. Recognising that it is difficult to get the average cell-phone user to formulate privacy policies, one should carry out surveys and user trials investigating attitudes and desires with respect to personal location privacy in conjunction with LBS. One should also investigate the relationship between attitudes and actions, as it is not obvious that individuals may act in accordance with their expressed views.

In the paper we have assumed that raw location data is available through the function 'RawObservation()()'. The language fragments proposed do not easily allow the formulation of policies involving the 'inverse' of 'RawObservation(a)(t)' relative to t. That is, return the set of times consistent with a given observation. It may be worth considering how to enhance the policy language with features making this inverse available. As pointed out by one of the referees, the service provider may then be provided with these times at various degrees of accuracy. Then one could construct applications that could answer questions such as

- Is this the first time I'm here?
- When was the last time I was here?

This could then be done without having to trust the service provider to record large quantities of sensitive location data.

It would be very useful to investigate how the use of personal privacy policies could contribute towards the reduction of legal barriers. Along similar lines, it would be useful to have identified what would constitute an acceptable 'purpose', that is, identify the 'purpose' lattice. We have indicated how one may combine privacy policies, but we have not discussed how to handle conflicting policies (government, employer and individual). Clearly, there will be situations when a decision on what information to release will depend on past queries. How should this be formulated?

8. ACKNOWLEDGMENTS

The anonymous referees provided several helpful comments and suggestions. Ragni Ryvold Arnesen, Demissie B. Aredo, Hans Jacob Rivertz and Christian Hauknes made several helpful comments.

9. REFERENCES

[1] Ateniese, G., Herzberg, A., Krawczyk, H., and Tsudik, G. On Traveling Incognito. In journal of Computer Networks (31) 8, pp. 871-884, 1999.

[2] Ateniese, G., Herzberg, A., Krawczyk, H., and Tsudikc, G. Untraceable Mobility: On Travelling Incognito. Computer Networks and ISDN Systems, April 1999

[3] Bahl, P., and Padmanabhan, V. N. "Radar: An in-building rf-based user location and tracking system," In Proceedings of the IEEE Infocom 2000, Tel-Aviv, Israel, vol. 2, Mar. 2000, pp. 775–784.

[4] Berthold, O., and Köhntopp, M. Identity Management Based On P3P. In proceedings of "Workshop on Design Issues in Anonymity and Unobservability", july 2000. Also available at http://www.koehntopp.de/marit/publikationen/idmanage/BeKoe_00IdmanageBasedOnP3P.pdf

[5] Fox, S. The Internet Life Report. Trust and Privacy Online: Why Americans Want to Rewrite the Rules. The Pew Internet & American Life Project. August 20, 2000. http://www.pewinternet.org/reports/pdfs/PIP_Trust_Privacy_Report.pdf

[6] Lee, C., Hwang M., and Yang W. Enhanced privacy and authentication for the global system for mobile communications. Wireless Networks Volume 5, No. 4 (Aug. 1999). Pages 231 - 243

[7] Leonhardt, U., and Magee, J. Security considerations for a Distributed Location Service. Journal of Network and Systems Management, 6(1):51-70, March 1998

[8] Norwegian parliament. 'Act of 14. April 2000 No. 31 relating to the processing of personal data (Personal Data Act)'. http://www.personvern.uio.no/regler/peol_engelsk.pdf

[9] Rannenberg, K. How much negotiation and detail can users handle? Experience with security negotiations and the granularity of access control in communications. In "Proceedings of 6th European Symposium on Research in Computer Security, France, October 2000", LNCS 1895, Springer, Editors F. Cuppens, F., Deswarte, Y., Gollmann, D., and Waidner, M.

[10] Reservasjonsregisteret. The Brønnøysund Register Centre. http://www.brreg.no/oppslag/reservasjon/index.html

[11] Robinson, T. Location is everything. Internet week online, tuesday September 12, 2000. http://www.internetwk.com/lead/lead091200.htm

[12] Spreitzer, M., and Theimer, M. Providing location information in a ubiquitous computing environment. In Proceedings of the 14th ACM Symposium on Operating System Principles, volume 27 of ACM SIGOPS, pages 270-283, 1993.

[13] Spreitzer, M., and Theimer, M. Scalable, secure, mobile computing with location information. Communications of the ACM, 36(7):27, 1993.

[14] Sun Microsystems. JAVA Location Services. Java™ Location Services: The New Standard for Location-enabled e-Business. http://www.mapinfo.com/community/free/library/java_location_svcs_whitepaper.pdf http://www.jlocationservices.com/company/ImageMatters/javalocationServices.html

[15] W3C. P3P and Privacy on the Web FAQ. http://www.w3.org/P3P/P3FAQ

[16] W3C. Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/

[17] XMARC INC., WISE 2.0, http://www.xmarc.com/news_events/2001/press_air-xmarc.htm

Summary of notation

N	The set of natural numbers
=	Partial order
lub(a,b)	The least upper bound of a and b
glb(a,b)	The greatest lower bound of a and b.
$\bigcup$	Distributed union
$\cup$	Union
$\cap$	Intersection
\	Set difference
$\subseteq$	Subset
$\in$	Set membership
$\Rightarrow$	Logical implication
$\wedge$	Conjunction
{v | …}	The set of v's such that …
$A \rightarrow B$	The collection of partial functions from A to B.
$A \times B$	The cartesian product of A and B
? x. E	Lambda abstraction