FoxyTag

Michel Deriaz and Jean-Marc Seigneur

Abstract. This paper presents the first steps towards the design of an application called FoxyTag, allowing a driver to post a virtual tag on a speed camera in order to notify other drivers. A trust mechanism allows the automatic computation of the trustworthiness of a given tag. This results in a trustworthy architecture, freely accessible by anyone that owns a mobile phone and a GPS.

1 Introduction

Today, there are many ways to get informed about the positions of speed cameras. However existing solutions have drawbacks, like outdated or missing information, textual descriptions instead of coordinates (which could be used by a GPS device), or are very expensive. Our solution handles and solves most of these issues.

We chose the topic of speed cameras for two reasons. First, it is very context dependant. For example the heading of the car can be used as a filter to avoid alerts concerning speed cameras on the opposite direction. Another example is the precision of the tag's position that depends on the current speed of the car, the precision of the GPS device and on the current environment (tunnel, city centre...). Second, many European countries are currently installing lots of new speed cameras, which provides by the same way a big community of potential testers for our application.

We propose an architecture that allows any driver to easily signal a speed camera or to signal that a former one has been removed. This is done by posting virtual tags at the critical points, so that other drivers can be alerted on time. Our trust mechanism is then responsible to link together trustworthy people and exclude malevolent ones. So the more a user participates, the more he will get trustworthy information.

2 Related Work

The design of FoxyTag is based on the use of spatial messaging (virtual tags) and the quality of the service relies on a trust engine. There are many approaches in defining trust and there are many spatial messaging systems. In this section we describe the most representatives, and we give also a summary of some other speed camera warning systems.

2.1 Computational Trust Overview

In the human world, trust exists between two interacting entities and is very useful when there is uncertainty in result of the interaction. The requested entity uses the level of trust in the requesting entity as a mean to cope with uncertainty, to engage in an action in spite of the risk of a harmful outcome. There are many definitions of the human notion trust in a wide range of domains, with different approaches and methodologies [20]: sociology, psychology, economics, pedagogy… These definitions may even change when the application domain changes. However, McKnight and Chervany have convincingly argued that these divergent trust definitions can fit together [21].

Interactions with uncertain result between entities also happen in the online world. So, it would be useful to rely on trust in the online world as well, as in our speed camera collaborative alerting application. A computational model of trust based on social research was first proposed by Marsh [13]. Trust in a given situation is called the *trust context*. Each trust context is assigned an importance value in the range [0,1] and utility value in the range [-1,1]. Any trust value is in the range [-1,1). A computed trust value in an entity may be seen as the digital representation of the trustworthiness or level of trust in the entity under consideration. The trustcomp online community [22] defines *entiTrust* as a non-enforceable estimate of the entity's future behavior in a given context based on past evidence. The EU-funded SECURE project [18] represents an example of a trust engine that uses evidence to compute trust values in entities and corresponds to evidence-based trust management systems. Evidence encompasses outcome observations, recommendations and reputation. Figure 1 gives an overview of a trust engine. The decision-making component can be called whenever a trusting decision has to be made. The Entity Recognition (ER) module [14] bridges the gap between identity management and reputation by recognizing the entities involved in the interactions with attack resilience and privacy protection considerations. The decision-making of the trust engine uses the trust module to dynamically assess the trustworthiness of the requesting entity and evaluates the risk involved in the interaction based on the available trust and risk evidence in the evidence store.

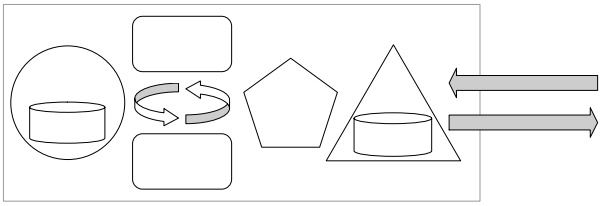

Figure 1. Overview of a Trust Engine

A common decision-making policy is to choose (or suggest to the user) the action that would maintain the appropriate cost/benefit. In the background, the evidence manager component is in charge of gathering evidence (for example, recommendations, comparisons between expected outcomes of the chosen actions and real outcomes…). This evidence is used to update risk and trust evidence. Thus, trust and risk follow a managed life-cycle.

2.2 Spatial Messaging

Spatial messaging, also called digital graffiti, air graffiti, or splash messaging, allows a user to publish a geo-referenced note so that any other user that attends the same place can get the message. For example, let us consider the community of the Mt-Blanc mountain guides. The members would like to inform their colleagues about dangers in specific places or about vacancies in refuges. One guide can publish a geo-referenced message that informs about a high risk of avalanches, and any other guide that attends the same place will get the warning, and comment it if necessary. Spatial messaging is a kind of blog, in which editors and readers share the same physical place. Many other examples of hypothetical scenarios can be found in [1]. The following academic projects focus on spatial messaging.

2.2.1 E-Graffiti

E-Graffiti [3] is a spatial messaging application that allows a user to read and post geo-localized notes. These notes can be either public or private, meaning that only the set of people defined by the author are able to read the note.

E-Graffiti has been designed to study the social impacts on spatial messaging. 57 undergraduate students were given a laptop with E-Graffiti for a semester. All their activity has been logged and studied. And the results are far from encouraging. At the end of the semester, it came out that a user logged into the system only 7.6 times in average (std dev: 12.6), and that actually most of the user stuck to initial test messages. Another disappointment was that most of the posted notes were not related to their position. For example, a number of people posted notes to advertise a website. The system was designed so that the user could only get messages available at his current position, but it was possible to post a new message at any place from anywhere.

Technically, the position of the user is determined by the wireless access point to which the device is connected. The precision is therefore limited to the building in which the user is.

2.2.2 GeoNotes

GeoNotes [4] has more functionalities than E-Graffiti. While posting a note, the user can choose how he is going to sign it (for privacy reason the user can write any text he wants as a signature), decide whether people are allowed to comment it, and decide whether anyone can remove this message. For the readers, the graphical interface of the application provides some interesting functionalities like showing all the neighboring messages or sort them according to different criteria. Inspired by the E-Graffiti evaluation, GeoNotes discarded the remote authoring of tags as well as the possibility to "direct" notes to certain users.

The main interest of the GeoNotes authors seems to be the navigation problems in the virtual messages space. How to find a specific note? How to select only relevant messages? One answer of these questions consists in giving to the readers the possibility of ranking the notes. Each user maintains also a friends list, which can be used as a filter. But the trust and security aspects have not been taken into account. It is easy to usurp someone's identity and post funny notes. An analysis of a GeoNotes

log made during a real-use study showed that 6% of the messages have been signed using someone else's identity.

2.2.3 ActiveCampus Explorer

ActiveCampus Explorer [5] goes a step further by displaying also where other users are. Every user holds a PDA and its location is determined by comparing the signal strength of different wireless access points. Thus, the system knows the position of all its users, and communicates this information to the all of them that are close together. Like E-Graffiti and GeoNotes, it is also possible to tag objects.

2.2.4 Socialight

Socialight [17] allows a user to post some data to a specific place, intended for himself, for his friends, or for everybody. Meta-data containing keywords and geographical coordinates are attached to the posted data, in order to facilitate searches. Tags are called Stickyshadows and can be viewed with some specific mobiles phones (and equipped with a positioning system) via the Socialight Mobile application, or by browsing the Socialight website. A nice feature they provide consists in showing Stickyshadows on maps.

2.2.5 Context Watcher

Context Watcher [16] is a mobile phone application written in Python for Nokia Series 60 based on the MobiLife framework [19]. The first version of this application already uses the notion of confirmed buddy for security and trust purposes. It may be expected that the following version will use the MobiLife trust engine as specified in the MobiLife framework for other purposes.

2.2.6 Concluding remarks

It seems that academic projects like E-Graffiti or GeoNotes didn't reach a critical mass of users. We believe also that the lack of success is related to the lack of interest... in publishing notes just for publishing notes! Spatial messaging would probably have more chance to emerge if we focus on specific communities, with real problems that could be solved by this concept, rather than imposing the system to students without giving them any good reason to use it. Another reason could also be that it is impossible, in some systems, to post anonymous messages.

2.3 Speed Camera Services

As the number of speed cameras increases on European roads, we find more and more services that help the driver to avoid expensive pictures. We will talk neither about illegal means (for the majority of European countries), like the radar detectors provided by RadarBusters [6], nor about non-technical means like phone centrals providing vocal information. We will concentrate here only on information systems that inform drivers about speed camera positions, which is completely legal according to the law of most European countries.

2.3.1 SmartSpeed

SmartSpeed [7] is an application running on Windows Mobile that informs the driver about dangerous zones, traffic jams, and speed cameras. Working with all NMEA compatible Bluetooth GPS, the program compares the current position with the "events" to come and informs the user through a voice synthesizer. Maps and "events" files can be downloaded in advance, and a GPRS connection allows the user to get recent information. An interesting functionality allows any user to send a new event to the server, which will in turn inform all the users. A typical use consists in signaling mobile speed cameras to other drivers. Even if presented differently, it is clearly a way of doing spatial messaging.

The light version a SmartSpeed is relatively cheap (30 € including free updates for one year) if you possess already a smartphone and a Bluetooth GPS. However, messages sent by other users to signal mobile speed cameras are not verified and are available only for one hour. And users are not really motivated to post such messages since they have nothing to gain in signaling a new "event". SmartSpeed seems more adapted to signal fix speed cameras than mobile ones.

2.3.2 Coyote

Coyote [8] is an independent system sold as a little box containing a GPS. When the driver approaches a speed camera, Coyote informs him orally about the remaining distance to this camera. To signal a new speed camera (or a new position for a mobile one), the user can simply press once the button on the top of the box. To signal a speed camera on the opposite direction, the user presses twice the button. This information is then sent to the server thanks to an included GPRS card, where a human operator verifies (previous messages of that user, comparison with other users, using another speed camera information service...) the plausibility of the information before broadcasting it to all users.

Despite it is very simple to use, Coyote remains an expensive system (699 € for 2 years with unlimited use and including communication fees) that not everybody can afford. And if there are too few users, then the chance that **you** are the first that discover a speed camera (by being flashed!) is high...

2.3.3 Natel-Futé

Natel-Futé [9] is a system that informs its users about traffic jams and mobile speed cameras via SMS. The user gets only textual information about the positions of the speed cameras. It is therefore not possible to be automatically informed when we approach a critical point, like it is done by SmartSpeed or Coyote. And the price is quite expensive too, since you have to count with about 170 € for one year.

2.3.4 InfoRad

Autonomous and easy to use, InfoRad [10] beeps when the driver enters a "risky area". All the risky areas, materialized with a speed camera, are stored in the on-board database. It works thus only with fix speed cameras and it is not possible to signal a new one to other drivers. It allows however a user to add its own risky areas for personal use. Their website provides time-to-time updates of risky areas. The device with an unlimited access to their database costs about 200 €.

2.3.5 POIplaces

A POI (Point Of Interest) is a geo-referenced item that presents a particular interest, like a restaurant, a fuel stations, or a car park. Written in standard formats, POI lists can be used by most navigation systems. POIplaces [11] is a website where people can share their own POIs. One successful topic is speed cameras. In the same way than for restaurants or fuel stations, users can download for free the list of all speed cameras. Their navigation system can then be configured to emit a sound when they approach a POI.

Free for everyone that owns already a GPS and a navigation device, this solution is however far from perfect. Since everybody can publish its POIs without any control, the speed cameras database is incomplete (lots of speed cameras are missing), redundant (several POIs for the same speed camera), incoherent (speed cameras have been found in a forest...), and mobile speed cameras are not taken into consideration.

2.3.6 GpsPasSion

GpsPasSion [12] provides active forums about the different topics of the GPS world. Some of them are specialized in the speed cameras domain and aim to collect information about their positions. They provide time-to-time an update of their POIs file that can be freely downloaded. Compared to POIplaces, the list is smaller (lots of speed cameras are missing), but is more consistent since they check the information before updating their list. Members that submit new positions have also access to a list containing the preferred places for mobile speed cameras.

3 Towards an Efficient Speed Camera Tagging System

We saw in section 2.3 - "Speed Camera Services" that there is no ideal solution for a reasonable price. Systems like Coyote are very expensive. Natel-Futé provides only textual information instead of coordinates. Some systems are more specifically (like Smartspeed) or even exclusively (like InfoRad) designed for fix speed cameras. And finally some suffer from inconsistent, missing, incomplete and untrustworthy data (like POIplaces or GpsPasSion). We propose to use trust to build an application that informs drivers about speed cameras helped with the knowledge in computational trust acquired by the group members who participated to the SECURE project [18] introduced above in Section 2.1.

In order to validate our ideas for the development of trusted spatial messaging, we developed GeoVTag. GeoVTag [1][2] is an application for reading and posting trusted spatial messages. It is designed to run on a java-enabled mobile phone coupled with a Bluetooth GPS. The architecture is centralized. Each server manages vTags (virtual tags) of a specific subject and each of them is identified by a different URL and works independently from the others. Any user can obtain anonymously all the vTags in his neighborhood just by queering the server. To become a member, and therefore be able to review vTags (add comments to an existing vTag) or be able to create new vTags, a user has to register. The registration process allows you to choose a pseudonym and returns a key pair that will be used each time to reconnect to the server. The registration process is done so that it is impossible, even for the server, to

make a link between a pseudonym and the corresponding real-life identity. The process guarantees also that each pseudonym is unique, so that it can be used as a unique identifier.

Technically, posting virtual tags is quite simple. During a vTag edition, the application gets also the GPS position and adds it as a meta-data. The whole is then sent via Internet (using the HTTP protocol) to a vTag server. For the vTag reading, the principle is similar. The mobile user sends its current position to the server (still using the HTTP protocol) which returns all the available vTags at the given position and its neighborhood. The size of the neighborhood is specified during the request.

GeoVTag serves as the vehicle for further research in trust for spatial messages. GeoVTag provides the generic framework for the development of trust mechanisms and models for spatial messages. A reference implementation, called GeoVTagRI, will be sufficiently generic to suits to several completely different services. For example, we can imagine a tourist that comes to our country. He sees on a flyer the URL of a vTag (virtual tag) server supplying useful information in his mother tongue, like information about places he is visiting, or what people with the same cultural background than him think about the different neighboring restaurants. If the potential user seems interested by this service, he will probably accept to add the URL at his server list, but it is much more unlikely that he accepts to add new software on his mobile phone for each new service.

The second and main advantage over other spatial messaging systems is however the trust engine that will allow the computation of the trustworthiness of a vTag. Each vTag contains different trust values computed according to the current context, the marks given by reviewers, the reputation of the author, and the friends list of the reader. A user can then easily determine how reliable a given vTag is.

Based on the GeoVTag platform and in order to validate the trust ideas, we designed an advanced speed camera application allowing to signal and being informed about speed controls: The FoxyTag application.

4 The FoxyTag Application

In this section we present our model which aims to solve most of the issues described in 2.3 - "Speed Camera Services".

4.1 Usage Scenario

Our model, built on top of GeoVTag, works as follow: A driver runs the client application (or simply "client" in the rest of the text) on his mobile phone. Coupled to a GPS or to another positioning device, the client knows permanently its position, expressed in latitude and longitude. According to its context (position and heading), and to the trust relations with his friends, the driver gets a personalized list of all the speed cameras he is susceptible to cross within the next few minutes. The client is then responsible to warn him when he approaches a critical point. We call "protected zone" the zone that is covered by the speed camera list. Time-to-time, the client connects to the server to check if there are changes in the protected zone. It contacts

also the server when it gets ready to quit the protected zone, so that the server defines a new one and provides the corresponding speed camera list.

When a driver gets a false alarm (he is alerted about a speed camera that does not exist), or when he crosses a speed camera, he signals it to the server, which can then update the user's trust relationships. The more a user contributes, the more his trust relationships will evolve and become precise: the approach is based on computational trust as presented in subsection 2.1. In return the user beneficiates of more accurate lists.

4.2 Initiating a First Connection

To simplify the trust model and to avoid that the system becomes a Sybil attack [15] victim, we impose that a single user owns only one pseudonym. There are many means to achieve this goal. It will depend on the identity scheme used and the GeoVTag model will be generic enough to accommodate any of these schemes by following the ER (Entity Recognition) approach [14]. One of them consists simply in sending a reverse billing SMS (there are many phone company partners that provide such a service) containing a password. Then, each time the user connects to the server, he will identify himself with his pseudonym and the corresponding password.

4.3 Exchanged Messages

Since Internet communications through a mobile phone are still quite expensive, our model tries to minimize them. A client sends only five different messages (we do not take into consideration the messages exchanged to establish the connection):

- **CPZ, latitude, longitude**: CPZ is an abbreviation for Check Protected Zone. The message indicates to the server that the driver is currently at the position defined by the given latitude and longitude, and that he wants the protected zone to be checked. Checking is done in the following way: if there are changes in the current zone, or if the user is close to the border of the zone, then a new protected zone is computed. Otherwise, the server just answers that the current protected zone is still OK.
- **CAN, latitude, longitude, heading**: CAN is an abbreviation for cancel. The message indicates that there is no, or not anymore, any speed cameras at the given position and for the given heading. The heading is important, it avoids that a user cancels a speed camera in the opposite direction.
- **MSC, latitude, longitude, heading**: MSC is an abbreviation for Mobile Speed Camera. The message indicates that there is a mobile speed camera at the given position and for the given heading.
- **FSC, latitude, longitude, heading**: FSC is an abbreviation for Fix Speed Camera. The message indicates that there is a fix speed camera at the given position and for the given heading.
- **OTC, latitude, longitude, heading**: OTC is an abbreviation for Other Type of Camera. The message indicates that there is a camera that does not measure the

speed of the driver (like the ones used to record registration numbers) at the given position and for the given heading.

Headings are usually positive numbers between 0 and 360 (0° = North, 90° = East, 180° = South and 270° = West), but it is possible to add a minus sign to indicate that the measurement has been taken from the opposite direction (for example a driver sees a mobile speed camera in the opposite direction). The server will therefore compute the correct heading, and, during trust computation, take into account that the precision of the positioning is not optimal.

A message from the server is either an acknowledgement, to say for example that there is no change in the protected zone, or a list of speed cameras as well as information about the space covered by the new protected zone.

4.4 Defining Protected Zones

As a first approach, we thought that the ideal shape and size of a protected zone can be computed mathematically according to the context. For example, we guess we are in a city if the speed is low. We would then choose a small (the speed camera density is high) circle (we are susceptible to change often the heading). On the other hand, we guess we are on a highway if the speed is high. We would then choose a thin (the chance that you change suddenly your heading is low) and long (the speed camera density is low) rectangle. But we were wrong. The speed and the heading is only a part of the context. For example, on a highway speed cameras usually form a line along the road (since highways are most of the time in the countryside), so a wide zone contains not necessarily more speed cameras than a thin one. Computing the ideal zone according to the context is far more complex than we could initially think, and this issue is therefore out of the scope of this paper.

Since mathematics could not help us to define the best protected zones, we set up dozens of different hypothetical scenarios and deduced experimentally the best parameters. It came out that a circle with a diameter of 12 km suits in most of our scenarios. All the numbers presented in the rest of this paper are also deduced experimentally.

When the user connects to the system (time t_0), the server defines a protected zone as a 12-km-diameter circle centered at the user's current position (see Figure 2). Every 5 minutes, or when the distance between the driver and the center of the circle is higher than 5.5 km, the client connects to the server and sends him a CPZ (Check Protected Zone) message. For example, in Figure 2, a driver follows a given path. At time t_1, the driver is at 5.5 km from the center (or he reached the internal circle in the figure). The client connects to the server which computes a new protected zone. To do that, the server draws a line between t_0 and t_1, prolongs it for 5.25 km, and defines the end of this line as the center of the second circle (see Figure 3). When the driver reaches t_2 (5 minutes after t_1), the client sends a CPZ message. Since there is no change in the protected zone, the client keeps the same one. And finally, the process repeats when the driver reaches the internal circle of the second protected zone, at time t_3.

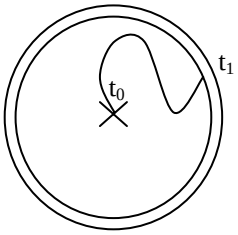

Figure 2

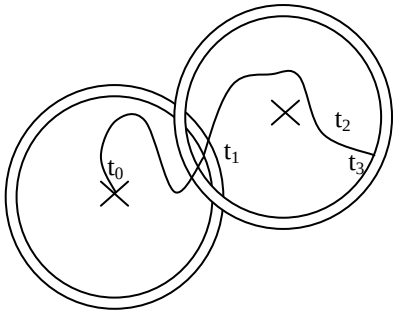

Figure 3

If a user requests two new protected zone within a minute, the second one is centered on the driver's current position. This could happen for example when the driver changes its heading just after the computation of a new protected zone, or if the driver goes round of a point that is between two protected zones.

4.5 Specific Rules

Some additional rules must be obeyed by the client in order to behave in a similar way as the others, and therefore get the same chances in building good trust relationships. The first rule concerns speed cameras inside shadow zones (zones that prevent the client from knowing its current position, like in a tunnel). The rule says that if a speed camera is in a shadow zone, then the warning (message signaling the presence of a speed camera) must be posted at the shadow zone entry. Technically, the application remembers always the last position it gets from the GPS, and uses it as the position of the shadow zone entry when the GPS becomes unable to make a fix. The second rule concerns speed cameras close to shadow zone exits. In order to leave enough time between the moment the device acquires again a positioning signal and the time the driver crosses a speed camera, the second rule imposes to use the coordinates of the shadow zone entry when posting a new message within ten seconds after exiting the shadow zone.

5 Future Work

In the previous sections we presented the global view of our model. However we did not talk about how the trust relationships are built, and how they are used to set up personalized lists. These issues are out of the scope of this paper and are precisely the ones we are going to focus on in future work. For now we consider it as a black box containing six main modules (see Figure 4). The first is a history database, containing previous messages of the driver. This information is used for the computation of the next protected zone and for building the trust relationships. The second is a speed camera database, containing all the information about the positions of the speed

camera. The third is an ID database for the phone numbers, pseudonyms and corresponding passwords. The fourth is a geo toolkit, containing tools to compute geo-related information (distances, precision of positioning, contextual information...). The fifth is a security tool box used to authenticate and protect communications. And finally, the sixth is the trust engine which builds the trust relationships and that uses them to personalize the speed camera lists sent to the drivers.

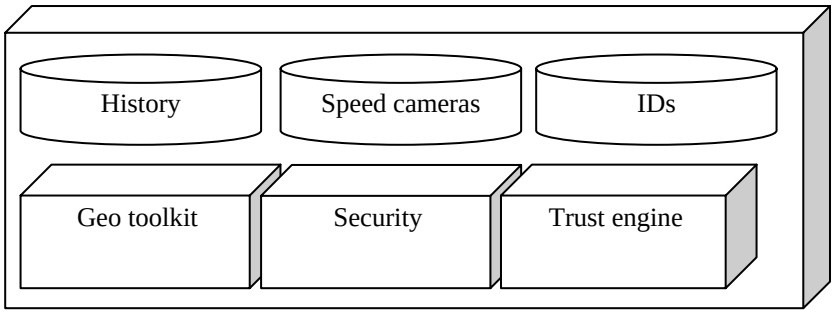

Figure 4

The validation process will be divided in two parts. The first will take place in the field. It consists in driving in the city of Geneva (Switzerland), and logging the driver's behavior. Log files will be studied and compared with other sources, like the ones provided by POIplaces [11] or GpsPasSion [12]. This validation aims to prove that our application works in real-life situations.

The second will focus on the trust engine. We will build a simulator, define a community of users (including malevolent ones), and observe how trust relationships are built, and make sure that a person how behaves honestly receives in counterpart trustworthy information. We envision also, in a later time, to provide the application on the Internet and profit from a large community of real users. Since we can initially feed our database with information found on other sites, we can provide an application that is usable even if the community is small; we do not need to reach a certain threshold or users.

6 Conclusion

We presented a first move to the design of the FoxyTag application, aiming to post virtual tags on speed cameras in order to inform other drivers. To keep the information trustworthy, we use a trust engine. We saw in the related work that other systems rely on human interaction, thus improving the price of the system. Our model can be implemented in a trustworthy and cheap solution.

However, the speed cameras topic is just an example. We chose it because there were already available data and because we have potentially a big community of volunteers to test it. Our results will also be applicable in other domains like road

traffic management (a traffic jam could be signaled in advance), or in domains that have nothing to do with roads and cars, like position-aware outdoor games.

References

[1] Michel Deriaz. Trust and Security for Spatial Messaging. In this collection of papers
[2] Michel Deriaz. GeoVTag. In this collection of papers
[3] Burrell, Jenna, Gay, Geri K. (2002): E-graffiti: evaluating real-world use of a context-aware system. In Interacting with Computers, 14 (4) p. 301-312
[4] Persson, P., Espinoza, F., Fagerberg, et al.: A Location-based Information System for Public Spaces, in Höök, Benyon, and Munro (eds.) Readings in Social Navigation of Information Space, Springer (2000)
[5] William G. Griswold, Patricia Shanahan, Steven W. Brown, et a.: ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing. IEEE Computer 37(10): 73-81 (2004)
[6] Website: http://www.radarbusters.com/
[7] Website: http://www.smartspeed.fr/
[8] Website: http://www.moncoyote.com/
[9] Website: http://www.natel-fute.ch/
[10] Website: http://www.gpsinforad.co.uk/
[11] Website: http://poiplace.oabsoftware.nl/
[12] Website: http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=21763
[13] S. Marsh., "Formalising Trust as a Computational Concept", PhD Thesis, University of Stirling, 1994.
[14] J.-M. Seigneur, "Trust, Security and Privacy in Global Computing", PhD Thesis, Trinity College Dublin, 2005.
[15] John R. Douceur. The sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.
[16] Website: http://www.lab.telin.nl/~koolwaaij/showcase/crf/cw.html
[17] Website: http://socialight.com/
[18] Website: http://secure.dsg.cs.tcd.ie/
[19] Website: http://www.ist-mobilife.org/
[20] J.-M. Seigneur. In this collection of papers.
[21] McKnight, D., and Chervany, N. L., “The Meanings of Trust”, MISRC 96-04, University of Minnesota, 1996.
[22] Website: http://www.trustcomp.org/