IPSJ Journal Vol. 52 No. 7 2065–2086 (July 2011)

Regular Paper

Defining and Investigating Device Comfort[⋆1]

Stephen Marsh,[†1] Pamela Briggs,[†2]
Khalil El-Khatib,[†3] Babak Esfandiari[†4]
and John A. Stewart[†1]

Device Comfort is a concept that uses an enhanced notion of trust to allow a personal (likely mobile) device to better reason about the state of interactions and actions between it, its owner, and the environment. This includes allowing a better understanding of how to manage information in fine-grained context as well as addressing the personal security of the user. To do this, it forms a unique relationship with the user, focusing on the device's judgment of user in context. This paper introduces and defines Device Comfort, including an examination of what makes up the comfort of a device in terms of trust and other considerations, and discusses the uses of such an approach. It also presents some ongoing developmental work in the concept, and an initial formal model of Device Comfort, its makeup and behaviour.

1. Introduction

Mobile devices, such smartphones, laptops, tablets, or a development of some other form factor, are growing in both popularity and capability. A large amount of computing power is coupled with hitherto unheard of sensing capabilities, to produce platforms that really can divine where they are on the planet, to an extent who is using them, and what their users are doing with them.

If we are to follow the postulation of the power of anthropomorphisation so aptly demonstrated in Ref. 31), we may come to the conclusion that people really do think of their devices in a way at least approaching that of a relationship. Indeed, the relationship the owner has with the device can be achieved by, for example, simple customization, moving things around on the screen, changing things like ringtones, and so on, as well as through potentially deeper sense-act capabilities on the part of the device. However, we do not see a great deal in the way of a concrete examination of how the device relates to its owner. The device-owner relationship is much deeper than that displayed in owner-device relationships because, at the crux, the device is what is doing the work. In the Ambient Intelligence (AmI), or Ubiquitous Computing (UbiComp), world, this work has often included sharing information with others around it in an effort to make things 'easier' or more convenient for the owner, as with location-based services [12),33)].

Other than things like lock codes and simple security measures, very little has been done to enhance or understand the device-owner relationship. We move toward a situation of a problem of increased capabilities tied to an almost non-existent conception of personal (information and action) security, bar the simplest of protections. Work ongoing in UbiComp remains a pointer in the direction of the applicability of, for example, context modelling and contextual reasoning [4),6)] for mobile devices in specific situations, such as driving [40)] or virtual meeting rooms [11)], but does not as yet stretch to this inherent *bi-directional* relationship. Whilst this is possibly because UbiComp has moved from general to more specific goals and applications, it nonetheless remains a loss.

Context and contextual modelling are undoubtedly vital to the success of ubiquitous computing, or the dream of truly responsive mobile technologies. Security is another vital aspect of the puzzle [24)], and trust has indeed been postulated as a potential aid in this environment [13)]. However, rather than simply extend into the mobile space models of, for instance, webs of trust, and how to trust *other people*, we conjecture that a largely *local* approach to trust 'management' can produce interesting results because it can to some extent model and facilitate this bilateral relationship between device and user (how to trust *the other person* or other devices). While interesting formal models of trust do exist, we do in fact persist in the belief that the relationship can be based on, and reasoned about,

†1 Communications Research Centre, Canada
†2 Northumbria University, UK
†3 University of Ontario Institute of Technology, Canada
†4 Carleton University, Canada

⋆1 The primary author of the work is an employee of the government of Canada and prepared the work in connection with their official duties. As such, the work is subject to Crown Copyright and is NOT assigned to the IPSJ. The undersigned acknowledges, however, that the IPSJ has the right to publish, distribute and reprint the Work in all forms and media.

via relatively straightforward models of trust with simple rules. We believe this for two main reasons: the first is that a simpler model is inherently more easy to explain to users, as we have mentioned above; the second is that, at the local level, trust requires no more than simple answers to two questions [20]: how much do you have, and how much do you need? ⋆1

Device Comfort is an attempt to merge relationship with (at least information, and up to and including personal) security in mobile devices. It represents a two-pronged approach encompassing a subset of Persuasive Technology [7] that we have begin to refer to as 'Annoying Technology,' and a conceptualization (and ongoing implementation) of the enhanced trust relationship between the device and the user. This latter, is based on a wide range of considerations, some of which are traditionally seen in ubiquitous computing methodologies, including location, goal, or task, and some of which are more in the domain of traditional computational trust, such as a simple underlying trust model, coupled with that of regret management and trust for information handling [23].

Device Comfort has grown from an understanding of the Biometric Daemon [2] which at its outset aimed to provide a more secure information environment. Device Comfort also owes much of its efficacy to the concept of local, emergent behaviour postulated in Artificial Life work (see e.g., Refs. 15), 16), 25)) – the concept that even simple rules can lead, with interaction with other such rules, to interesting behaviour that, regardless of its initial simplicity, can be seen as complex or contextually intelligent. There is however less intelligence in it than there is intuition.

This paper discusses the concept of device comfort, provides a working definition, and examines the components of device comfort that we have identified to date. It illustrates some of the components via a conceptual extension of our extant trust formalization [20),23].

It is important to acknowledge the fact that we approach this topic somewhat controversially by referring to how a device may 'feel,' not to mention Device Comfort as a phenomenon. The point, of course, is that we may see echoes of tamagotchis in some aspects of Device Comfort, and an often unashamed use of anthropomorphisation in particular through the paper. There are a couple of reasons for this – the explanation aspect is one, and the other is that indeed the device is not a tamagotchi, but then it is not a phone or a laptop or some other device either, since the Device Comfort model is defining a new kind of reasoning and behaviour on the part of the device that may be seen as an amalgam of electronic pet or companion, mobile computer, phone, and personal information repository. We acknowledge that speaking 'as if' the device really is comfortable or feels something is apt to make some uncomfortable, but argue that it has extraordinary explanatory power at the same time. Not to mention that it's helped immeasurably in guiding the ongoing research.

Device Comfort is an ongoing research project, and a great deal of work is currently exploring and defining the questions the work brings up as much as its efficacy. It's also based on a conceptual vision, and the paper reflects this. Thus, the paper is arranged as follows: the first part is a discussion the 'big picture' of Device Comfort, while the second part discusses the practical approach we are taking to lay the groundwork for comfort, and enable mobile devices to become active participants (agents and actors) in the world, through reasoning about their users, rather than just objects. More specifically, the next section discusses why we should care about the concepts discussed here, in the context of security and behaviour. Following this, we define device comfort and explore the three components thereof. We next discuss aspects of device comfort, before entering into an exploration of the groundwork we have laid in order to facilitate extending mobile devices as Comfort devices. Following a look at related work, we conclude with a brief re-exploration of some of the benefits of the device comfort model and of potential future work in the physical (device) domain.

2. Why Should We Care?

A great deal of work has been put into making devices work well for their owners. Human Computer Interaction, interface design, increasing capabilities, industrial design and more, all contribute to a user experience that is hopefully rewarding and efficient in packages that even a few years ago would have seemed virtually impossible to attain. Rather than specific examples of devices, it's

⋆1 We do accept that these questions hide a multitude of sins, and the purported simplicity is in the eye of the beholder. However, we also believe that if the beholder is the user of a device, this simplicity is quite enough.

possibly useful to think in terms of 'ecosystems' for devices, or, at a more prosaic level, operating systems. Thus, at the time of writing, we may see Apple's iOS, Google's Android, and to a lesser extent systems such as Intel/Nokia's Meego and Microsoft's Windows Phone 7. Examples of devices that run within these ecosystems abound, and it isn't instructive to list them here. However, in general, all of these devices are mobile computers, some with more capabilities than others, but all with extensive computing power and all with customizability and design to ensure user 'satisfaction' from their experience in interacting with the device.

And, indeed, users do seem largely satisfied. At least, they keep buying the devices, and they keep using them (and they're not cheap). The convergent device has enabled people to communicate in more ways, at more times and in different places than ever before, and distance is truly not an object. The relationship that the owner has with their device has been enhanced and encouraged by each manufacturer and extensively examined by researchers and designers.

There are, unfortunately and inevitably, downsides. Any one of these devices is a shiny gadget that is tempting to those ⋆1 who don't have and want one (or have one and want the next best thing). Quite apart from the need for developers to 're-learn' secure programming methodologies, we can observe attacks on users, thefts and subsequent loss of device (and information). Vendors such as F-Secure are now marketing the capability to wipe stolen phones of all their data with a single message, and there are other techniques for locking stolen phones, or disallowing the swapping of SIM cards to address some of these problems.

Cyber-bullying is increasingly a problem, both physical and 'virtual.' Some of this bullying and related antisocial behaviour has resulted from the actions of the owner of a device themselves, making too much information public, sexting, and so forth ⋆2. It has also recently been acknowledged that mobile devices are increasingly capable of being used as attack tools in their own right ⋆3 – something that isn't exactly a surprise. Also, notwithstanding laws against it in many places, the use of such devices when one should be concentrating on other tasks (driving for one) remains an issue. Consider that privacy is of utmost primary concern in any ubiquitous, mobile, or reputation-based environment [1),17)].

In many of these downsides, we can see a particularly promising pattern, that *the device itself may be key to addressing the problems.* Indeed, we conjecture that if the device were capable of reasoning in a more nuanced and informed manner about what it was doing, and who was requesting the action, where it was, and other contextual niceties, many of these problems may be mitigated. That's not to say traditional security methodologies don't work, or won't work, merely that if we can leverage some of the sense-act powers of the device, we should try to make them work for us as much as the capabilities of the device sometimes work to our detriment, or encourage us to do so.

We believe that the answer lies in the *relationship* that the device has with its owner and the environment the device is in. It's an enhanced notion of location, trust, and action/task management that we believe can lead to a more secure, user/owner sensitive system that is socially adept and technologically savvy. This relationship is what we call Device Comfort.

3. An Overview of Comfort

Comfort, as defined in Merriam-Webster online, is "a feeling of relief or encouragement . . . contented well-being, . . . a satisfying or enjoyable experience." We see no reason why this definition should not serve as a useful starting point for the relationship that we conjecture should exist between a device and its owner. As we have stated above, there is a large body of work on the relationship the owner has with the device. There is relatively little to none at all about how the device feels about its owner, or more particularly how the device feels about what its current user requires of it.

The feeling of comfort that the device should be able to draw from its current context should ultimately allow it to make decisions on behalf of, or give advice to, its legitimate owner (who sometimes is not the same as its current user) that can mitigate some of the threats extant in mobile technologies. It's possible, and relatively straightforward, to imagine a device which simply refuses to perform as required (for instance, if it has been stolen). It's much more nuanced to have a device that aims to, for example, make the user think twice about an action

⋆1 At least, those with a different moral code. . .
⋆2 Witness, http://www.athinline.com/
⋆3 See for example, http://www.reuters.com/article/idUSTRE5AG5BQ20091118

('are you sure you want to send this picture?'). Device comfort can allow the nuanced as well as the 'sledgehammer cracking a nut' approach.

Device comfort has to be seen as a dynamic phenomenon – the more capabilities a device has for, e.g., sensing its environment, the more we can integrate into comfort, and our groundwork for this sensing is discussed further below. However, the current aspects of Comfort are plainly necessary whatever the future brings. Currently we see device comfort as a measure based on reasoning about an amalgamation of the following:

- The user (if they can be identified, or not, and who they are) –
- Most importantly, enhanced trust reasoning about the user, and the ongoing *relationship* with respect to trust that the device has in the user (and/or owner);
- The current task (for instance, making a call, sending text, sending pictures, email, etc.)
- The current location (which virtually all mobile devices can determine with some accuracy)
- A Comfort Rulebase (which can be specified by the owner or their representative)

We will examine these in more detail in the following sections.

4. The Big Picture of Device Comfort

Device Comfort, as described above, is a value judgment on the part of the device based on reasoning about (currently) three specific components: User (Owner), Task, and Location. In this section, we briefly show how this can be calculated in a simple manner.

Each of these components has an effect on device comfort. As with the idea of Basic, General, and Situational Trusts (see Ref. 20)), any device has a Basic Device Comfort level with which it can start any deliberation regarding a particular unknown context task, location, etc.). Each component in context acts with this Basic level to ultimately produce a Situational Device Comfort, in the following manner:

Basic Device Comfort combined with Device-Owner Relationship Trust is used to calculate General Device Comfort. The General Device Comfort is a value that can be taken to some extent out of any other context to produce an overall picture of the device at a given time when no task is being undertaken (other than, e.g., background processing). While Basic Device Comfort is not in itself particularly dynamic, it's possible to imagine General Device Comfort fluctuating to some extent across time. It is also possible to inject some Location considerations to General Device Comfort in order to take into account where the device is regardless of the task undertaken. We discuss this further in Ref. 22).

When specific tasks are initiated, it is possible to use General Device Comfort, combined with Location and Task Components, to determine Situational (or Contextual) Device Comfort. It is this Situational Device Comfort which enables the device to determine if the current action in context is 'sensible' or not, or makes it uncomfortable or not. And with this knowledge, it can take some form of action.

In other words:

- If Situational Device Comfort > Comfort Threshold, then the device can be said to be 'comfortable' and perform the action.
- If Situational Device Comfort < Discomfort Threshold, then the device can be said to be 'uncomfortable' – see below.
- Otherwise, the device is 'ambivalent' and can justify its ambivalence if queried.

In the first and last cases, a simple interface icon (green or white dot in status bar, for instance) is potentially all that is necessary to get the message across (we are investigating different models).

In the second case, of 'discomfort,' more is needed. As an aside, it is interesting that the negative areas are more worthy of consideration than the positive areas – not surprisingly, we find the same situation in trust reasoning – when you don't have enough trust, or don't feel comfortable enough, what do you do? As with trust in certain circumstances, the answer is usually 'think about it a bit more and then decide,' and not 'forget it.'

A more formal discussion, including definitions of Basic, General, and Situational Device Comforts, follows in Section 5.7. First, we explore some of the ways in which user behaviour can affect Device Comfort.

5. Thinking about the User

In many ways, this is both the most straightforward and the hardest of the components of comfort to reason with and about. It's relatively straightforward because, with the right assumptions and data, it's possible to build up a quite detailed picture of users, their likes and dislikes, behaviours and requirements, and from this build a trust picture for that user – both of how and whom they trust as well as how the device trusts them. It's difficult because at the outset getting those assumptions correct is a challenge, and the relationship is just that – something that builds over time and cannot necessarily be rushed.

The Biometric Daemon approaches the problems by acknowledging the fact that a relationship has to be developed – owners have to grow with their Daemon, building the relationship over time and with use. Moreover, identifying the owner is as straightforward as ensuring that their (behavioural as well as physical) biometrics match what the Daemon knows. While we can't necessarily assume today's devices are capable in all these respects, certainly behavioural biometrics are not beyond their scope. However, building a relationship is something of a challenge except in its simplest forms.

That said, we consider that there are several states in the relationship, in some respects not altogether unlike what may be found in nature. These have a direct effect on the level of trust the device has in the owner. These are:

- Imprinting
- Nurturing
- Growth
- Repair
- Use

At any time, the device is in one (or perhaps more in complex contexts) of these states, as discussed below. It should be noted that we see this component of device comfort as being extremely closely tied to trust – emphatically the trust the device has in the owner, but also the trust the owner has in the device – while the latter is hard for the device to ascertain, we can also safely estimate it with regard to the Comfort equation, as we see below.

The Use state is technically very close to the Task component of device comfort. Below we briefly cover some aspects of it that are relevant to the trust relationship.

5.1 Relationship Trust

The device-owner/user relationship in these states is largely informed by reflection on the part of the device. That is to say that the device is doing the reasoning about how it sees the relationship (and potentially how it perceives the owner feels, but this is at best an inexact science). We call this Relationship Trust.

Relationship trust is an amalgamation of:

- Current level of trust in owner:
  $TD = T_{\text{Device}}(\text{Owner})^{\text{Device}}$
- Current estimation of owner's trust in Device
  $TO = T_{\text{Owner}}(\text{Device})^{\text{Device}}$
- If applicable, Situational Trust and Cooperation Threshold estimates for the current situation (see e.g., Ref. 23))

For the purposes of the current discussion, we bypass the Situational Trust and Cooperation Threshold, and we have, at any time t:

$$\text{RelTrust}_{\text{Device}}(Owner)^t = \frac{TD + TO}{2} \quad (1)$$

Here, we have a value for the device-owner relationship in any given situation that is relatively straightforward to obtain.

For the purposes of device comfort, the TD value is the most important of the trust values here, because it is this which is influenced and enhanced or damaged by the states of the device-owner relationship. That said, we can perhaps in the absence of other information reduce RelTrust to the TD trust value. However, the more information we have, the better the final results.

5.2 Imprinting

In its initial state, the device knows next to nothing of its owner. Here, the first hours or days of its existence are spent in an imprinting mode, which basically enables it to build a very strong model of trust and behaviour (biometrics and associated identifiers – see also Ref. 37) for an examination of owner identification) with its owner. There should be no need to imprint more than once assuming that:

- The owner doesn't lose the device (or have it stolen), and if such a thing does happen,
- There is a viable, up to date backup available for a new set of hardware.

However, it will be possible for the owner to enter a new imprinting mode (effectively wiping out extant trust models) if needed – for instance if modes of behaviour change for nurturing, or perhaps in the case of acquiring a second hand device (legitimately!).

More formally:

$$TD^{(t^0)} = f(C_I, T(Imprint), Sec(Imprint))$$

Where:
C_I is a predefined imprint constant.
$T(Imprint)$ is related to the length of time take to imprint in general, we might expect that the longer time taken, the better the imprint and the higher the initial TD.
$Sec(Imprint)$ is related to the security aspect of the imprint behaviour - more secure biometrics or behaviours that are subjectively harder to observe or replicate, for instance, may increase TD initially. However, this is beyond the scope of this paper, and we don't go further into it here.

5.3 Nurturing

Following Imprinting, the device is capable of moving between several states. Perhaps the most important of these is Nurturing, which relates to reassurance. Here the device is continually nurtured in the unique manner established by the owner at imprinting – for our purposes, this is most likely via behavioural biometrics.

In this phase, the trust the device has in its owner is effectively reinforced (as is its Comfort), and depending on the level it started at, is either maintained or returned to the imprint level of trust, but not usually increased. Bear in mind that, as per the Biometric Daemon, the trust value will periodically decrease without nurturing behaviour.

The Nurturing mode can be entered at any time, on initiation by the device or the owner. The device can initiate, for example, where its level of Comfort has dropped below the initial imprint level of trust (TD^{t^0}).

More formally:

if $TD^{(t-1)} > TD^{(t^0)}$ **then**
  $TD^{(t)} = TD^{(t-1)}$
else
  $TD^{(t)} = TD^{(t^0)}$
end if

5.4 Growth

The Growth and Nurturing phases are closely linked. In the Growth phase, however, unlike Nurturing (which maintains Comfort via reassurance), the explicit aim is to enhance trust (and hopefully as a result, comfort).

This is relatively straightforward and requires that the owner behave according to type, or in a benevolent fashion. It may simply be a case of the user using the device 'properly' (responding to comfort warnings, for instance.

The Growth stage is interesting because it is at this stage the device is trying to ensure its Comfort whilst still 'protecting' the owner's information (and by extension the owner). These two things are not the same, all the time. So, while it may be the case that the owner wants to accomplish a specific task, the device may feel uncomfortable performing it. The result is that the owner may be upset, and this may damage the relationship (and device comfort).

More formally:

$$TD^{(t)} = TD^{(t-1)} + f(C_G, ComfortS^{(t-1)})$$

Where:
C_G is a predefined growth constant.
$ComfortS$ is Situational Device Comfort.

5.5 Repair

Transgressions in behaviour and action are not uncommon. When something goes wrong, relationships can suffer. The device may have, for instance, shared information it should not have, or perhaps the user has carried through a task

regardless of the Comfort level of the device. The result is a lower level of trust one the part of one or both of the parties.

We don't discuss the device increasing owner trust in it here (we've covered some of that in Ref. 23)). Repairing device trust in owner is at the high level similar to the processes involved in the growth stage. However, the symptoms of trust needing repair should be obvious to the owner/user – continuous 'discomfort' messages, for instance. The device is, of course, in a difficult situation because the basic operations of it should be maintained regardless of how if perceives comfort or trust, so to an extent the legitimate user can simply ignore the process and carry on. However, as we discuss further below, we conjecture that a comfortable device can potentially give better advice or protection to the user because it is able to focus on the reason for any discomfort more closely than a continually untrusting device.

Device capabilities notwithstanding, for the user, we might say that simply behaving in a fashion that responds to warnings is often enough. Taking a leaf from the Daemon, the initiation of a nurturing phase on the part of the user may also help repair damaged trust.

More formally:

$$TD^{(t)} = TD^{(t-1)} + TD^{t^0}/C_R$$

Where:
C_R is a predefined repair constant.

5.6 Use

Consider an example from Ref. 18) where an employee wishes to access a secured site provided by her employer. Ordinarily, the exchange of credentials would be enough to establish the 'trusting' relationship. Introducing the device allows the negotiation to be further automated with concern for situational aspects inherent in the negotiation. For instance, the employee may be in an untrustworthy location (country) which makes the device uncomfortable enough not to share credentials with the employer's system. The strength of the model lies in the ability of the device to search for comfort in a given situation that would allow it to share this information (perhaps by reassurance from the user).

In a different kind of use, the device should be able to give advice based on trust/comfort levels based on what is being done (see below) or with whom. Thus, as in Ref. 23), some people or other devices are trusted more than others, and this affects the amount of and quality of information available to them.

More formally:

$$TD^{(t)} = TD^{(t-1)} + f(ComfortS^{t-1}, ComfortS^{t}, C_U)$$

Where:
$ComfortS$ is Situational Device Comfort, as will be seen below.
C_U is a predefined Usage constant.
Note that the outcome of the formula may well be negative, shrinking the level of TD. For the purpose of this work, we use the following:

$$f(ComfortS^{t-1}, ComfortS^{t}, C_U) = \frac{(ComfortS^{t} - ComfortS^{t-1})}{C_U}$$

5.7 Continuing a Formal Exploration

Device Comfort, as we currently define it, is reasoning based on several parameters, some of which are founded on the sensing capabilities that we currently identify on extant devices. Since we much expect these capabilities may be augmented in the future, we accept that the model below and through this paper with be subject to change, likely for the better, as devices become more capable.

With this in mind, there are three types of Device Comfort:

- Basic Device Comfort:

  $$ComfortB = f(ComfortS^{(t-x)}, TD)$$

  In Basic Comfort, we use knowledge of previous behaviour as represented by the trust (TD) of device in user, as shown above – this analysis, including acknowledgment of behaviour aimed at strengthening the relationship, inherently takes into account what has come before. In specific, Basic Comfort expresses the level of comfort a device will assign to completely unknown situations, much as with Basic Trust [20]. We use an analysis of a series of time, the length of which (x) is dependent on sensitivity to previous experiences, thus:

$$ComfortB^t = \frac{\Sigma_{tt=t-x}^{t-1}(ComfortS^{tt} * TD^{tt})}{x} \quad (2)$$

Where x is the length of time back which we wish to analyze, and could be either user- or pre-defined.

- General Device Comfort

  $$ComfortG = f(Loc_D, Soc_D, Sec_D, ComfortB, TD^{(t-y)})$$

  General Comfort takes into account the basic level of comfort expressed above, but also some measures of context that do not include task, which is a situation-specific parameter. It is used by the device to give an indication of comfort at any given time that is unaffected by the current use of the device, and can be used, for instance, to determine responses to events initiated off device, as well as in the calculation of Situational Comfort. In the above function, there are:
  - Loc_D: a value based on the *Location* the device perceives it is in. See Section 7.4.
  - Soc_D: a value based on the *Social Situation* the device perceives it is in. See Section 10.3.
  - Sec_D: a value based on the *Security Situation* the device perceives it is in. See Section 10.1.

  Each of these will be further discussed in the sections referred to, below. We will thus revisit the calculation of General Device Comfort in Section 11.
- Situational Device Comfort

  $$ComfortS = f(ComfortG, Task_D)$$

  The Situational Device Comfort measure is used at specific points in time and takes into account context at that instant, which is General Comfort and Task. The $Task_D$ value is a comfort level calculated from the current task list, and is shown in Section 6.

5.8 'Help! This Device is Being Stolen!'

It probably goes without saying that we are concerned with the theft, or loss, of mobile devices, particularly given their capabilities for storage of and release of private, personal information. The RelationshipTrust model, based as it is on the Biometric Daemon concept, specifically intends to defend against this problem. If the device is not Nurtured properly when required (because the current user does not know the proper way to nurture it), it will simply refuse to operate because its RelationshipTrust is compromised. It alternatively may operate in a limited sense (if your friend borrows the phone, for instance) until you get it back and reassure it.

6. Considering the Task

Mobile devices are very capable – they can be used for email, web access, playing music, taking (and forwarding) photographs, recording voice or music, texting, MMS, GPS navigation and more. The more capable and adaptable devices are effectively only limited by the imaginations and skills of their users and programmers (along with obvious considerations of speed and hardware ability).

They are also largely capable of determining which task(s) is (or are) being undertaken at any moment – at the very least which applications are running, and what state the device is in – which we can extrapolate usably to task. This capability is useful because there are levels of device comfort associated with task. It's fairly straightforward to consider some simple rules, for example:

(1) If a task has never been done before, Comfort is negatively affected.
(2) If a task has never been done before *in this context*, Comfort is negatively affected.
(3) If a task has not been done for some time, Comfort is negatively affected.
(4) if a task has been done repeatedly and recently, Comfort is positively affected.
(5) If a task is proscribed or flagged, Comfort is negatively affected.

And so forth.

In this list, some of the items are in no need of explanation or justification. However, the second and last items bear some further elucidation.

6.1 I've Never Done This Before (Task In Context)

Tasks are contextually sensitive. That is to say, they are very much sensitive to items such as location, as we show below. However, they are also sensitive to other items of context, including:

- What other tasks are ongoing on the device at the time;

- With whom is the task being undertaken (if anyone) – see more below;
- When (date, time) they are being undertaken.

Indeed, most aspects of the device are contextual, as with UbiComp, or should be seen as such, with some more influential than others. For instance, the date the action is undertaken may be indicative (paying bills every 1st of the month, for instance). In others, the day of the week may be contextually important (accessing work files outside of Monday-Friday, for instance). It is difficult to imagine that any device would implicitly know how the context is important in these cases, but there are two important things to note here. The first is that it can be told, and simple rules can be established to ensure contextuality in time and how sensitive the device is to these (and these may not necessarily even need to be user-created). The second is that the device can learn (see rule 4 above).

The context of other people (With whom...) is also interesting. That's because there are things you may not want to do with some people that it's fine to do with others. It may be fine to share pictures of your house with some people, but others should not be trusted because they may share it outside of the sensible limits (with results of, for instance, potential burglary when criminals know you're out). In a work situation, some information may be sharable via email with some colleagues but not others (and some not at all). Once again, a series of rules are necessary in some cases to ensure their correct influence on device comfort (see below). Also, again, there are learned behaviours which can have some effect. However, it's worth bearing in mind here that, the fact that you do something with someone else a lot, does not mean it's sensible (or should make the device any more comfortable) – that said, it's difficult for any device to decipher social conundrums, while sharing work information with the same person outside of work may be indicative of a problem.

6.2 This Task is Proscribed (Flagged)

To paraphrase, just because it's possible to do something, it doesn't mean you should. Consider for example the sexting phenomenon. It's not particularly novel, but the transport medium is, and the potential for abuse is horrible. On a more prosaic level, considering someone else's feelings before sending a less than sensitive text or email in response to a perceived slight is probably sensible (we acknowledge an approach to natural language is more than just useful here, but don't go further with the idea).

It's possible to think of many more things that you may do with mobile devices that are less than sensible, in fact. Some of them should probably indeed be proscribed. Some in fact are – driving and texting, for instance – but it's difficult for the device to know if you are indeed the driver even if you're in a moving car (we can certainly tell you're moving) and are texting (likewise).

It is not our intention, however, to introduce a system for mobile devices that stops people doing things with them. That said, it's worth allowing these 'proscribed' actions to negatively influence device comfort to the extent that the device lets the user (owner) know about it, and provide the impetus for *second thoughts*, which as we all know are often very important.

As with the other interesting contextual situations, many, or most, of these actions are in fact more promisingly codified and put into a rule-base than reasoned about on the fly by the device. Thus, the actual act of sending pictures (excepting certain specified closely monitored situations or contexts) could be proscribed so as to affect device comfort negatively. It cannot in truth be otherwise because the potential for abuse far outweighs the benefits for some users. Thus, a rule such as 'No pictures except to these people or this protected store' for a preteen is probably a sensible enough rule.

We are in the process of determining (and testing the efficacy and acceptability of) sensible rules for proscription. Once more, bear in mind that proscription may be a rather strong word for what we have in mind.

6.3 Calculating Task Comfort: $Task_D$

As discussed above, the situation specific task comfort level is used in calculating Situational Device Comfort. The previous sections have introduced how the level may be affected by tasks, including task proscription. Here, we show this more formally.

The $Task_D$ task comfort level is calculated from an initial level, $Task_D(i)$. This can be predefined or, for instance, based on previous levels of comfort. The higher this level, the more comfortable the device is likely to be with regard to task in general. Every different contextual piece of evidence for the device affects the final $Task_D$ level based on this initial level.

Put simply:

if TaskNotProscribed **then**
  $Task_D = Task_D(i)+$
  $f(IsNew, InContext, IsRegular, Date, AppList)$
else
  $Task_D = 0$
end if

Note that, if the task is proscribed, we can return a zero value, but also refuse to execute the task regardless of overall Device Comfort, depending on the security posture of the device (and the predefined action by the person or organization that proscribed the task). The parameters in the function are used to determine the extent of the effect on the final $Task_D$ level. Note that we are reducing this to apps running because that's rather easier at present than determining the real task. Note also that a full list of statements (for example, app in a list of business or personal apps, document being accessed, if any, is rated personal, and so on) is beyond the scope of this paper, so a shorter algorithm is shown here to illustrate the procedure:

$T = 0$;
if IsNew(app) **then**
  $T = T - Const(New)$
else
  if UsedInContext(app) **then**
    $T = T + Const(AppInContext)$
  else
    $T = T - Const(AppInContext)$
  end if
  if UsedRegularly(app) **then**
    $T = T + Const(Regular)$
  else
    $T = T - Const(Regular)$
  end if
  if 09 : 00 <= Time <= 17 : 00 and IsBusinessApp(app) **then**
    $T = T + Const(AppInContext)$
  else
    $T = T - Const(AppInContext)$
  end if
end if
...
return T
...

And so forth. Where **Const** represents some predefined value for increasing or decreasing the T value, which is ultimately used to increase or decrease $Task_D(i)$. So, for example, if the app is labelled a business app (say, a financial accounting or time reporting app) and used between 9 and 5, for argument's sake, the Task Comfort is increased, or at least remains the same, and if not, it is decreased.

7. Location Awareness and Comfort Zones

Consider the average mobile device of the past few months. Quite apart from the specifications of camera, GSM radio and wifi, it holds a great deal of data. In fact, it could likely quite happily hold all of the personal and work related information from the past ten to fifteen or so years of our lives. This is not an altogether comfortable thought, even if it is extremely liberating at the same time. It's not enough to be innocent; privacy is important, and others seeing this information out of context and unauthorized is rather a problem. Note that it is not necessary for the device to be stolen for this to happen. Crossing the border into a different country opens up the possibility that the device may be copied, and there's not an awful lot to be done about it.

There are, in other words, some places where my device should be less comfortable in, for example, sharing its data with others, than other places. It's often just a case of enhancing comfort in a familiar place, and reducing it in an unfamiliar one. This is about ensuring that your device is comfortable with where it is. There are sensibly some things which the device should not show, be capable of, or do, in some places. Of course, there are the obvious ones, like flight mode (if one considers an aircraft to be a 'place').

We have developed the concept of 'Comfort Zones' to further refine the concept of location-based Comfort.

7.1 Comfort Zones

A Comfort Zone (and its opposite, the Discomfort Zone) is a specified area or location in which the device is able to determine an effect on device comfort. An example of a Comfort Zone is 'Home' or 'Work', while a Discomfort Zone may be 'Any Border Crossing' or 'Downtown Baghdad'. The purpose of a Comfort Zone is to allow the device to more accurately determine its Comfort. Thus, in a 'Home' Comfort Zone its possible for the device to enhance Comfort, while in a Discomfort Zone Comfort is compromised.

We have developed several Comfort/Discomfort Zones in Ref. 22), and it would be simple for the user to add more. The inbuilt GPS capabilities of today's mobile devices makes this particularly straightforward. As well, there are times when you achieve (dis)comfort as a result of people around you, no matter where you are.

Thus, there are two distinct types of Comfort Zone: physical and social. Physical Comfort Zones are places that are identifiable via, for example, GPS or cell tower signals. They can be categorized as zones that don't move, and include Home, Work, Airports and Borders. They're relatively straightforward to set up and don't change often, if at all. An extension of the physical zone is a Mobile Zone, such as the route you take to commute, and this is itself a series of connected physical zones.

Social zones are more flexible affairs. One kind of social zone is the 'Social Area Zone,' which is a small area defined by some sensable properties. Below we describe such an approach using local SSID capture in defined areas. Another is the 'Social Contact Zone,' which is defined by the existence of some number of known others (or their devices) around the sensing device. For instance, if three or more 'social friends' with their devices are present, then the device may feel it is in such a zone. Note that 'social friends' are not the same, necessarily, as business colleagues, and the existence of three or more of the latter may still create a Social Comfort Zone, but with distinctly different characteristics, for example in what kind of information can be accessed.

7.2 Discomfort Zones

As we mentioned above, there are places you just don't want to be. The same should apply to your devices. Thus, we introduce Discomfort Zones. These are specific, prescribed areas (and they can be defined by the user) where the device ceases to operate or allow access, and the existence of a device in a Discomfort Zone will, *in most cases*, discount any further Comfort reasoning. We are also experimenting with the ability of the *location* to prescribe itself as a Discomfort Zone (locations can't make themselves Comfort Zones arbitrarily, for obvious reasons), which is analogous to the ubiquitous computing notion of context in place, and could be expressed as simply as 'don't allow phones to ring in a library,' although there is more to Discomfort Zones than this.

Comfort Zones and Device Comfort are complementary. At times, Comfort Zones may supersede anything else within the Comfort lexicon. Most usually, this is where proscriptions exist (for instance, 'do not access banking records outside of a Home Zone'), or rules for access exist (for instance 'allow access to company information when inside Work Zone') – in other words, Comfort Zones trump all other considerations in extremis. However, in general, as with all the other Comfort considerations expressed in this paper, the existence of (and placement of a device in) a Comfort Zone is used as input to the Device Comfort considerations

7.3 The Tahrir Zone

Recent events in North Africa and the Middle East have shown the power of communication amongst individuals, often using mobile technologies. Of course, this power is not the only thing to have made a difference, and perhaps the least, but it did indeed help provide a disruptive effect. This has led us to think about the concept of what we now call *Tahrir Zones* in recognition of this effect. The translation of the word as 'liberation' in English is not lost to us.

The key aspect of such zones is the transcendence of comfort – the fact that, regardless of what may be happening to cause discomfort, communication must persist. Thus, in a Tahrir Zone, a mobile device, *regardless of its comfort level*, must make all efforts possible, through any channel available, to communicate in an asynchronous fashion. Any information that is newly available *and created by the device* within a very short timeframe (and therefore, we may hope, related

to the current situation), should be permitted to be transmitted with whatever means are available, to whomever is reachable. On the other hand, incoming information should be compartmentalized for user interaction as soon as the user is able and willing. This may achieve the following: attempts to block access are in some way limited, while information can be shared in an ad hoc fashion without extra user effort, regardless of the affect on the overall comfort of the device (or perhaps, without affecting the comfort at all).

There are issues with such Zones: how to create them, how long they last, and additional security for device information whilst acknowledging the value of what the device creates. That said, there are far-reaching implications for Zones such as these where ordinarily Device Comfort may be extremely low, but where the context is urgent and important.

7.4 Formalizing How Location Effects Comfort

Overall Device Comfort is increased or decreased based on location in the following way:

```
accuracy = get(DeviceLocation) with available data
if DeviceInSpecifiedComfortZone then
  return Const(ComfortZone) * accuracy
else
  if DeviceInSpecifiedDiscomfortZone then
    return Const(DiscomfortZone) * accuracy
  end if
  if DeviceInSocialZone then
    return SocialZone(Location)
  end if
  return Const(NoZone)
end if
```

The returned value is, of course, Loc_D in the equations in Sections 5.7 and 11. The algorithm also shows that, regardless of where the device is, if it is outside of a comfort zone, the effect on overall Device Comfort is dependent on a predefined security posture – for instance, if the device is 'pessimistic,' (see e.g., Ref. 21)) being outside of any zone is likely to decrease overall Device Comfort, whilst if it is 'optimistic,' the opposite may be true.

The *SocialZone* function ascertains the number of trusted devices within range of this device, and if they belong to a pre-specified Social Zone as defined above. The more devices available, the greater the effect on Device Comfort.

Note the use of the *accuracy* value. Any device has a number of different ways to determine its location (as discussed above and in Sections 10.2 and 10.3). The more of these that are available and support each other, the less likely it is that the device is being spoofed in some way (GPS spoofing as well as SSID spoofing is real and possible). The more evidence in support, the more 'accuracy' the device can associate with a location, and thus, ultimately, the greater assurance it has that the final effect on Comfort is valid. For more on this, see Section 10.3.

7.5 Comfort Zones and Comfort

It's worth mentioning that the effect on overall Device Comfort of being in a (Dis)Comfort Zone is additional to the fact that by dint of the device being within such a predefined zone, certain capabilities are enabled by default (e.g., encrypted volumes are unlocked.) Imagine, for example, that a specific volume is listed as being opened when the device is within listed Comfort Zone(s). Being in one of these Comfort Zones may well make the overall Comfort level of the device higher, and thus some actions enabled, but this is *in addition to* the fact that a volume is unlocked. Likewise, being in a Discomfort Zone may mean the Device Comfort level is lowered, and thus some applications, for instance, may not be opened, but regardless of the Device Comfort level, in the discomfort zone, the volume *will not* be unlocked. Indeed, out of the Comfort Zone, this is the case. This is what makes the Tahrir Zone a possibility, because the overall Device Comfort can be ignored in specific locations for specific predefined actions. It's also what makes the potential for Policies that circumvent Device Comfort.

8. Policies: Following the Rules

There are times when comfort is not enough. Quite often, security has to rely on rather stark decisions: something is allowed, or it is not. In the case of a mobile device this is no different. Thus, the 'comfort' device specified here also needs a set of rules to prescribe or positively proscribe certain actions. It is possible to imagine a device that the owner's employer has supplied, which will

not allow access to work information or assets whilst out of work (location) or in specific places (allied with Comfort Zones) or contexts. It may be possible to imagine a device that does not allow preteens anything other than rudimentary access to social networking except when they are within a rather more transient comfort zone (for instance, within 100 metres of their parents). An organization may insist that no internal documents or information be accessible outside of the perimeter of the organization (easy enough to achieve with the capabilities of comfort described above).

Indeed, it is possible to think of many instances where rules would be applied regardless of device comfort, thereby overriding the comfort consideration altogether. This is the reason for the incorporation of an override rule base, which accomplishes traditional security alongside the 'soft' security of device comfort. Languages such as Ponder [⋆1] would be used to describe the particular rules. The rules here are similar to, but not the same as, those in the sections discussed above - similar to because they can be specified in the same way, but different because their strength is greater than the comfort consideration of the device and cannot be overruled even if the device considers itself to be 'comfortable.'

9. 'I'm not Comfortable with This'

When the device is uncomfortable, it should say so. A red dot in the status bar is almost certainly not enough. And if it says so, it should justify its discomfort (something we have stated before is particularly powerful with trust [23]) and we feel is equally suitable with Comfort).

This statement can take the form of what we have called *Pearls of Wisdom*, which are simple statements to the effect of 'Are you sure you should do this' or 'I am uncomfortable doing this' or 'This location (or situation) makes me uncomfortable.' Its actual wording is likely important, and we are planning to investigate this further. However, the message is clear that the device has crossed a threshold from ambivalence to discomfort, and the user should know about it.

9.1 Annoying Technology

We (hopefully) learn from our mistakes. Again hopefully, we have the opportunity to do so. As we have stated above, the device comfort concept is not intended to produce a device that the user cannot use to do what they want to do. While it's possible to imagine some situations where the device should absolutely not permit an action [⋆2], it's not a sensible option for several reasons, the least of which is that the user can just borrow someone else's device (without a Comfort Daemon!) and do it anyway.

Our purpose is simply this: to allow the user to have second thoughts about what they are (about to be) doing. We call this aspect of the model *Annoying Technology*. Quite often, we believe, pointing out that something is rather questionable or silly is enough to make the person realise that it is silly. While we acknowledge that this may in fact have the opposite effect on some people (or at least those of a certain ages group), consider: if they do it anyway, after the device said it was uncomfortable, and the result was negative, there is a learning process involved. 'I told you so' is not a necessary statement for the device to make, but it could certainly be in the mind of the user... Perhaps they will think twice next time the device says it's uncomfortable, and perhaps that instance would be a much more important one. This concept is akin to the need expressed in explanation systems and explanation-aware computing (see for example Ref. 29)). It's also realistically a subset of Persuasive Technology (cf Ref. 7)) because through being annoying it hopes to change behaviour, or *at least* a mind, when the power to proscribe behaviour is both present and not used.

There are exceptions to this rule, and of course we believe they should be user controlled (but the device may express its discomfort if they are controlled!). Most notably, as described above, there are situations where location trumps everything. If the device is in a prescribed Discomfort Zone, there are some things it just will not do, and some information it just will not give. We introduce the concept of Password Escrow allow the device to protect its data whilst maintaining deniability on the part of the user.

9.2 Password Escrow

In many cases, plausible deniability is a key to safety – if you don't have the keys to the safe, why bother trying to force them from you? The same can be true

⋆1 http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml

⋆2 We leave it an exercise to determine what these may be...

of information access – not knowing the password removes your ability to access the information even under duress (so, why bother with the duress?) Password Escrow attempts to achieve some measure of this by holding passwords for access to on-device information in a place both off the device and unknown to the user. Such an escrow service would hold passwords for the device to obtain when it felt comfortable enough to do so, and no action of the user could get them or force the device to do so. The structure of such a service is simple: The device encrypts with a random password, and sends this password to the escrow service, which confirms receipt. On confirmation, the device itself erases the password from memory. When it is comfortable, it can access the escrow service to receive the password back.

Extensions to the model are straightforward: the device can compartmentalize information and encrypt compartments, accessing different ones in different locations (see below for details of an implementation of this). More, the device can specify conditions to be met when it stores the password – for instance 'IF location $\neq$ Work Zone AND current user $\neq$ Steve THEN do not hand password back.' Such simple conditions can help to ensure that the passwords are more secure, and are similar to the concept of using different passphrases when under duress and unable to say so.

9.3 Emergencies Have Consequences

Many mobile phones let you make emergency calls, even if they're locked. Even if they have no credit left for calling.

This is a good thing.

However, when emergency calls have been made, what does the device do then? Right now, it slips back into whatever mode it was in before. However, consider the effect an emergency might have on a user – distraction, whilst seemingly not a big issue, is in fact a challenge[⋆1]. Regular behaviour in information protection can be lost to distraction, and, for example, phishing attacks can succeed where they may not have in normal circumstance (as has been stated more than once, it only takes one successful attack). The comfortable device, of course, takes this into account as part of context and as a result prevents certain actions following emergency calls (or similar occurrences) in order to protect the distracted owner.

More formally:

if Distracted **then**

$DeviceComfort = DeviceComfort - Const(Distracted)$

end if

As usual, the constant is predefined. Of course, determining Distracted Mode is not overly difficult because we can specify certain applications or actions (such as '911 Dialed' or 'website ABC visited' or 'Bank Website Visited,' or perhaps 'Between 0300 and 0600,' for instance).

10. Current Status

The Device Comfort model is a wide-reaching conceptual vision. Of necessity, we have split the research and development work associated with Device Comfort into a series of sub-projects, many of which are still ongoing, some of which have yet to commence. The majority of development has taken place using Nokia N900 devices and Apple's iPhone (iPad development is beginning) as well as standard Linux-based laptops for development. For the most part, the developments here lay the groundwork for Device Comfort, and should be taken as the first steps along the road.

10.1 Digital Elastic and Zero Interaction Security

Device Comfort is not simply about a personal mobile device figuring out where it is and acting accordingly. Our goal is that the device should be able to help its user protect themselves and their physical, as well as virtual (information) assets. One aspect of this, in the physical domain, is the subject of the Digital Elastic project. With Digital Elastic, we introduce a 2-way relationship between devices and other physical 'assets' the user may have. A couple of examples will illustrate the point.

The mobile device (which in this case is both the N900 and Linux laptops running Python) is paired with a bluetooth device, for instance a headset. When the two are close together, the mobile device remains unlocked. When the bluetooth device moves away beyond a specified distance, the mobile device locks up and cannot be accessed without a password.

⋆1 Thanks to Eric Neufeld for pointing this one out.

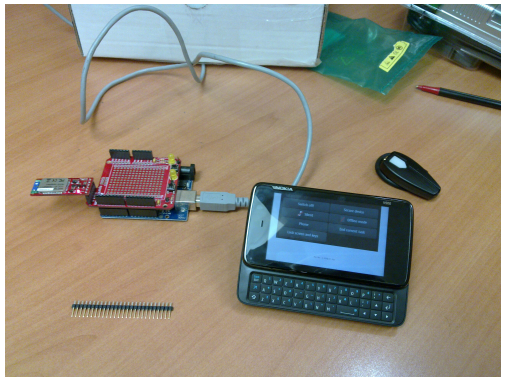

Fig. 1 Setup for digital elastic with N900 and arduino with bluetooth.

Items of value are 'tagged' with bluetooth tags, which are paired to the 'comfort' device. When these items (tags) move beyond a specific range away from the comfort device, the user is alerted by the comfort device.

The same principle applies with social location – with enough paired dongles around it, a device can open up more information – indeed, it can open up (as well as lock) information in a gradual fashion based on the number of paired devices around it.

We initially implemented bluetooth-based lock/unlock using off the shelf bluetooth devices, similar in fact to the concept of Zero Interaction Authentication [5]. However, more interesting is the concept of *active tokens* for what may be called *Zero Interaction Security*. In the active token model, a token or set of tokens contains information that the device can use in order to permit certain functionality (an initial setup is shown in **Fig. 1**).

Consider some examples: if there are two active tokens, each could store half of a cryptographic key for information encrypted on the device. When only one token is present, the information is not available. The presence of both active tokens enables the device to decrypt the information. Another example is where each of several active tokens may contain part of the information needed to access secure network facilities in specific locations. Unless the device is present, the information is not available, and unless the device is registered with all the tokens, it still can't allow user access to the network. What's more, when the device goes out of range (leaves the location) of the location protected by the active tokens, the needed access information is lost and the device cannot access the network. This is similar to the use of transient authentication [28] but works both ways because the presence of active tokens can serve to increase device comfort as well as network accessibility, allowing different actions based on the availability of tokens in the environment, rather than a token on the user (see Section 10.1.1).

In both of these examples, note that Zero Interaction on the part of the user is needed – the physical presence of the device and tokens is enough. However, additional security can be obtained because the tokens are active, and can require input from the device via the user (for instance, a password) in order to better ascertain identity.

We have currently developed active tokens using Arduino boards, and are moving to embedded linux boards because of their increased storage capacity, and thus the ability to store individualized information for specific devices, another step in the direction of user-oriented security.

10.1.1 Formal Security

In Section 5.7 we discussed General Device Comfort, one parameter of which was the Device's Sec_D level, the level of security it felt (or, to put it another way, how secure the device perceived itself to be).

It's a relatively straightforward value to determine if there is something to determine, which is to say that, the presence of the active or passive tokens can give the device information it needs to calculate the value. For instance, the presence of two active tokens that corroborate each other can result in a higher Sec_D level than if there was a passive security token. Likewise, the instant after a passphrase has been entered may be seen as a more secure moment than ten minutes later. We use these simple rules, and others, to determine the value of Sec_D, which is then used as shown in Section 11.

10.2 Location-Based Information Access

Similarly to the Zero Interaction Security model, we can use device location in a more general sense to perform security/comfort related actions. In the Location-Based Information Access model, which was developed in Python on a Nokia N900, the device has several encrypted volumes (using TrueCrypt), each

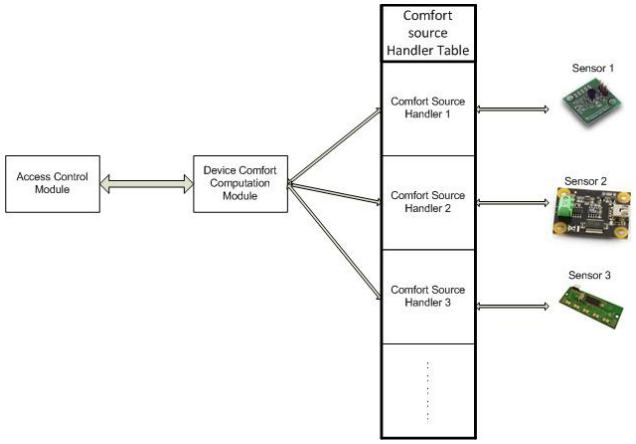

Fig. 2 Architecture of Location Based Information Access Modules.

of which has a separate passphrase for access not known to the user (but stored on device, for ease of implementation – further work will include the Password Escrow service described above).

When the device moves into a specified zone, the volume associated with that zone is opened and can be browsed normally. On leaving the specified zone, the volume is closed and locked once more. While a simple application this has shown us the elegance and power of location-based sense-act capabilities quite succinctly, and everyday use over the past few months have demonstrated its worth.

To implement and test the Location-Based Information Access, we have designed and implemented an open architecture (**Fig. 2**) that allows for simple and fast accommodation of different types of comfort zones. The main components of the architecture are:

- **Comfort Source Handler Table:** This table contains all the handlers (drivers or call back functions) for all comfort sources (sensors). These handlers are hardware specific, and are used for pulling raw data from the sensing module and converting it into a comfort value.
- **Device Comfort Computation Module:** The Device Comfort Computation Module is responsible for pulling the comfort data from the Comfort Source Handlers and combining the data into single device comfort value. It also allows the user to assign different weights for different sources of device comforts.
- **Access Control Module:** The Access Control Module in the user interface module which receives request from the user/application, and queries to Device Comfort module to compute the device comfort value. Based on the computed device comfort value and the threshold defined by the user, the Access Control Module decides whether to grant the request from the user or deny it.

A designer that needs to integrate a new type of comfort zone can simply add the Comfort Source Handler into the Comfort Source Handler Table, and Device Comfort Computation Module will take care of calling the handler and including its output in the final device comfort value.

10.3 Social Contact Zones

Give its ease of development and low complexity, Location-Based Information Access is valid for devices such as laptops with greater computing power. One question that remains, for most laptops, is 'where am I?' – services such as those used by Apple's iPod Touch can use Wi-Fi as location enablers, and the same can apply regardless of device. To address the rather coarse nature of this we have developed a means of using local Wi-Fi devices as location enablers.

The work here has two different stages. The first stage was concerned with building an active map of a local area. To do this, we used an iPhone and an identified area within Ottawa, Canada. We currently capture publicly available broadcast WiFi data; specifically, SSID, BSSID, data encryption type, signal strength, and broadcast channel. Data capture and/or directed packet inspection is not required for this project.

A visualization of the resultant map can be seen in **Fig. 3** – the height of the bars represents signal strength. Note that map itself is simply textual information and this representation is a very useful human-readable version. Different colour bars designate different networks. Visualization of collected data is for research purposes; we are investigating different ways of viewing the data, and the changes over time of the data, to allow us to understand the environment, and to allow benchmarking of derived algorithms.

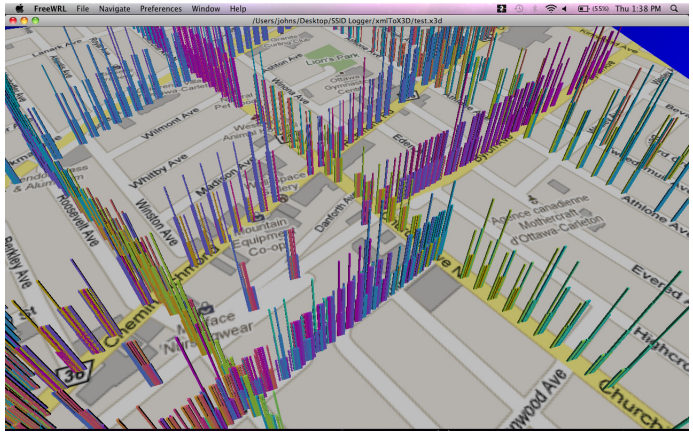

Fig. 3 A sample SSID map – Social Contact Zone.

Note that this is a 'living' map – the number and availability of SSIDs will change daily, if not more regularly, for more about which, see below. That given, we are continually re-running this stage and building a small library of maps.

The second stage is using the maps as input to a Comfort-reasoning device. Consider the following use case: The device has on board a map that it can use to compare against its current context (the networks it can 'see'). Given a specific set of networks available, it can accurately place itself within the Social Contact Zone with some degree of certainty. This degree of certainty is based on the similarity between its current situation and what the map says was there are some point, as well as the age of the map. It is a direct input to the Device Comfort reasoning algorithm, thus while we may be sure we're close to Starbucks at a particular location, the SSIDs of the surrounding networks may be sufficiently different to make us worry about our conclusion, and comfort is negatively affected accordingly (consider the potential for spoofing SSIDs to fool devices).

While continually updating the maps, we are working on stage two, as well as the ability of devices to share Social Contact Zone maps between themselves.

10.4 Device Comfort Primitives

We are actively working, as a result of the Digital Elastic/Zero Interaction Security project, on a list of primitives for the foundation of Device Comfort implementation. Primitives are the building blocks that a comfort-enabled device may want to use to determine its comfort level or act on it. Following the definition of primitives, we will develop a high-level policy language in order to program this kind of behaviour using the building blocks. This implementation could be done via a particular application provided by the employer, the parent, the device manufacturer, or as an add-on to another app, and so forth.

A list of primitives identified to date follows. They are divided into comfort-related 'events' and 'actions,' and one can mix and match them to come up with policies. Some of these primitives have already been implemented in the ongoing projects, and they are documented here. Some are extant in already deployed architectures but are included here to show their applicability. In all, there may be an effect on overall Device Comfort we don't specify here, but which is largely clear with some thought.

10.4.1 Events

- Owner is too far away (i.e., dongle associated to the owner is too far) (Digital Elastic).
- Tagged valuable is too far away (Digital Elastic).
- x familiar devices nearby (can combine with a threshold to determine comfort level) (Social Comfort Zones)
- User's face is familiar/unfamiliar (face recognition using phone camera matches owner's picture, or an owner's friend's picture, for example in FaceBook profile)
- Entered/left comfort zone (home, work, etc.) (Comfort Zones)
- Place is familiar/unfamiliar (Social Comfort Zones. maps)
- Device use is familiar/unfamiliar (does or doesn't match pattern of application use, time of use, frequency of use, etc.)

10.4.2 Actions

- Lock/unlock device (Digital Elastic)
- encrypt/decrypt drive (Location-Based Information Access)
- Take and send photo of user (smile, you're on camera, and I'm reporting you to the owner!)
- Send location updates (so the owner who lost the device can trace it and

recover it)

10.5 Operating System Support

Much of the security-related work above is ultimately reliant on the ability of the device to protect itself if, for example, stolen. It may or may not be trivial, but it is possible, to crack passwords on the device, for instance. To that end, a low level security setup is a best practice for allowing the implementation of these and other security possibilities. Our own work to date has largely been through proofs of concept, but the Zero Interaction Security model has gone a step further, and uses a specifically modified DBus for message handling, and a modified kernel to protect encrypted information (as well as encrypted swap partitions). We are currently in the process of profiling this setup on the N900, as subjectively we are noticing a slowdown in operations when, for example, encrypted swap is used. However, as with all things, the more protection wanted or needed, the more the user may be prepared to accept a performance hit.

11. Determining Overall Device Comfort

Now that we have the rest of the picture, we can finalize the determination of General and Situational Device Comfort.

Recall from Section 5.7:

- General Device Comfort

$$ComfortG = f(Loc_D, Soc_D, Sec_D, ComfortB, TD^{(t-y)})$$

- Situational Device Comfort

$$ComfortS = f(ComfortG, Task_D)$$

We can then state that, for the purpose of this work:

$$ComfortG = ComfortB + \frac{(Loc_D + Soc_D + Sec_D)}{3}$$

And:

$$ComfortS = ComfortG + Task_D$$

We are examining through experimentation the efficacy of these simple equations.

12. Related Work

Much mobile device security relies on more traditional models of security – the assumption that the user who is authorized is capable, and if the device is stolen, ensuring it is unusable. This is the model followed by approaches such as F-Secure, for instance.

In many cases, the user is expected to take 'reasonable care' ⋆1. with information belonging to the institution. The device comfort model, potentially allied to the concept of regret outlined in Ref. 23) of course addresses reasonable care rather well.

Location-based security is naturally a rather hot topic. Reference 10) discusses a model for trust beacons (allied with Bluetooth) which allows for mobile device authentication based on location. This controls access to resources owned by the organization. Again, this follows the model of proscription which disallows actions out of context. While the device comfort model can enable this more strict approach, we have chosen to follow a less proscribed model for the reasons given above and through the paper. Additionally, device comfort enables different actions in different contexts, location being only one of them, and also extends the location concept rather more effectively, for less cost. In Ref. 27) a combination of user wearable tag and mobile device is used to ensure the device does not work if stolen (or lost), similarly in Ref. 36). Again, device comfort based on Biometric Daemon techniques described above addresses the same problem with less cost (and still allows re-authentication with as little effort), while it also manages capabilities and gives a learning approach.

Location-based trust and privacy tends to take the route that people want to preserve privacy about where they are (cf Refs. 8), 9)). The device comfort approach does not in fact take this into account at present, but there is no reason it should not or could not. However, it attacks a completely different problem, that of behaviour, task, or access to resources in context (something akin to location-based DRM or encryption as in Ref. 19), 26)).

It's been twenty years since Weiser's original vision of the invisible computer [39)]

⋆1 See for instance, http://my.gwu.edu/files/policies/LaptopSecurityFinal.pdf

but the vision remains powerful and difficult to achieve. The field of Ubiquitous Computing (UbiComp) has grown to encompass a vast array of concerns, from HCI, security, ad hoc networks and contextual reasoning and representation. Indeed, work in UbiComp that is relevant to Device Comfort is naturally largely related to Contextual Awareness [4),6),32)]. Work on trust in UbiComp has largely focused on the concept of managing ad hoc and long term trust reasoning [3),35)], often with arguments for formality [13)], while the concept of degrees of trust in context is directly relevant [34)]. The work on what may be called trust*worthy* devices [⋆1] is again a focus on the relationship between user and device (or to paraphrase, usability and acceptance). Finally, we have some sympathy with Langheinrich [14)] in allowing trust decisions to be made by humans, but argue that such a stance in fact *increases* the need for the non-human actors in the system to be able to reason about their human counterpart users, hence the Device Comfort model.

Whilst the contextual reasoning elements of Device Comfort are akin to UbiComp, and many of the considerations elsewhere are similar, the major differences between Device Comfort and what has come before lie in two directions – the first is the enablement of the device to reason irrespective of what the world can do or tell it (thus, for instance, we can allow or disallow actions based on our context, not based on what context the world tells us). The second, more subtle and powerful, is the extension of trust into device reasoning, and the notion of a trust-based *relationship* between device and user (not just user and device) that is unaccounted for in other models. Put succinctly, the Device Comfort model aims to provide a sensible mode of behaviour in context, whilst the UbiComp notion is more directed to providing services and availability regardless of where the user is, what they want to do, and how they want to do it, whilst ostensibly being acceptable to the user and sensitive to their security. The real difference comes when the device tries to help the user understand why they *shouldn't* in fact be doing what they want, how they want, where they want.

Finally, this work is related to the idea of context awareness in trust models [30),38)] in that the device comfort is related particularly closely to the context the device finds itself in. We believe that the considerations of comfort are an enhancement of trust because they take into account for the first time the device-owner relationship as well as the device-environment context.

⋆1 The object of the TwUC workshops, http://www.twuc.info/

13. Conclusions and Further Work

The device comfort model, based on enhanced device-owner trust, regret management, and concepts gathered from the Biometric Daemon, is being used for several security-related purposes on mobile devices, including:

- Allowing users to do what they please, whilst giving them pause if what they please is questionable in some way;
- Not allowing some things under certain circumstances;
- Allowing the learning process on the part of potentially immature users;
- (Not covered here) Allowing potential control over device other than the user (e.g., in the situation of corporate-owned devices), most effectively through the Comfort Rulebase.

Ongoing research includes in depth user studies, refinement of the bidirectional trust model for mobile devices and comfort, and interaction design and testing.

13.1 The Promise of Comfort

The potential for such a capability in mobile devices is very large. It can allow freedom with learning (through annoyance, at least!), the development of a deep user trust in the capabilities of the device when it matters, and the protection of users from not just their own actions, but also those of others. It can also allow the device to protect itself in the event of loss of theft, and provide a means for limiting access to other potentially trusted sources based on Comfort as well as simple rules and policies.

References

1) Ahmed, M., Quercia, D. and Hailes, S.: A Statistical Matching Approach to Detect Privacy Violation for Trust-Based Collaborations, *6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM*, pp.598–602 (2005).
2) Briggs, P. and Olivier, P.L.: Biometric daemons: Authentication via electronic pets, *Proc. CHI 2008*, New York, NY, USA, ACM, pp.2423–2432 (online), DOI:http://doi.acm.org/10.1145/1358628.1358699 (2008).

3) Camp, L.J.: Design for Trust in Ambient and Ubiquitous Computing, *Proc. 6th International Conference on Autonomic and Trusted Computing*, Berlin, Heidelberg, Springer-Verlag (2009).
4) Chen, G. and Kotz, D.: A survey of context-aware mobile computing research, Technical Report, Department of Computer Science, Dartmouth College (2000).
5) Corner, M.D. and Noble, B.D.: Zero-interaction authentication, *Proc. ACM International Conference on Mobile Computing and Communications, Atlanta, GA* (2002).
6) Dey, A.K. and Abowd, G.D.: Towards a better understanding of context and context-awareness, *In HUC'99: Proc. 1st International Symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, pp.304–307 (1999).
7) Fogg, B.J.: *Persuasive Technology*, Morgan Kaufmann (2002).
8) Gedik, B. and Liu, L.: Location Privacy in Mobile Systems: A Personalized ANonymization Model, *Proc. 25th IEEE International Conference on Distributed Computing Systems* (*ICSCS'05*) (2005).
9) Gruteser, M. and Liu, X.: Protecting Privacy in continuous location-tracking applications, *IEEE Security and Privacy Magazine*, Vol.2, pp.28–34 (2004).
10) Jansen, W. and Korolev, V.: A Location-Based Mechanism for Mobile Device Security, *World Congress on Computer Science and Information Engineering*, pp.99–104 (2009).
11) Johanson, B., Fox, O. and Winograd, T.: The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, *IEEE Pervasive Computing*, Vol.1, pp.67–74 (2002).
12) Junglas, I.A. and Watson, R.T.: Location-based services, *Comm. ACM*, Vol.51, pp.65–69 (online), DOI:http://doi.acm.org/10.1145/1325555.1325568 (2008).
13) Krukow, K., Nielsen, M. and Sassone, V.: Trust models in ubiquitous computing, *Phil. Trans. R. Soc. A*, Vol.366, pp.3781–3793 (online), DOI:10.1098/rsta.2008.0134 (2008).
14) Langheinrich, M.: When Trust Does Not Compute — The Role of Trust in Ubiquitous Computing, *Proc. Privacy Workshops of Ubicomp'03* (online), available from ⟨http://guir.berkeley.edu/pubs/ubicomp2003/privacyworkshop/papers/ubicom p-trust.pdf⟩ (2003).
15) Langton, C.G.: *Artificial Life*, Addison Wesley, Advanced Book Program (1989).
16) Langton, C.G.: *Artificial Life: An Overview*, MIT Press (1997).
17) Lederer, S., Dey, A.K. and Mankoff, J.: Everyday Privacy in Ubiquitous Computing Environments, Technical report, World Bank, Policy Research Department (2002).
18) Lee, A.J., Winslett, M. and Perano, K.J.: TrustBuilder2: A Reconfigurable Framework for Trust Negotiation, *Trust Management: Proc. IFIPTM 2009: 3rd IFIP International Conference*, Ferrari, E., Li, N., Bertino, E. and Karabalut, Y. (Eds.), Springer (IFIP AICT 300), pp.176–195 (2009).
19) Liao, H.-C. and Chao, Y.-H.: A New Data Encryption Algorith Based on the Location of Mobile Users, *Information Technology Journal*, Vol.7, No.1, pp.63–69 (2008).
20) Marsh, S.: Formalising Trust as a Computational Concept, PhD Thesis, Department of Computing Science, University of Stirling (online), available from ⟨http://www.stephenmarsh.ca/⟩ (1994).
21) Marsh, S.: Optimism and pessimism in trust, *Proc. Iberoamerican Conference on Artificial Intelligence/National Conference on Artificial Intelligence* (*IBERAMIA94/CNAISE94*), Ramirez, J. (Ed.), McGraw-Hill (1994).
22) Marsh, S.: Comfort Zones: Location Dependent Trust and Regret Management for Mobile Devices, *Proc. TruLoco 2010 Workshop at IFIPTM 2010* (2010).
23) Marsh, S. and Briggs, P.: Examining Trust, Forgiveness and Regret as Computational Concepts, *Computing with Social Trust*, Golbeck, J. (Ed.), Human Computer Interaction Series, chapter 2, pp.9–44, Springer (2009).
24) Mayrhofer, R.: Ubiquitous Computing Security: Authenticating Spontaneous Interactions, Habilitation thesis, University of Vienna (2008).
25) Miller, F., Vandome, A. and McBrewster, J.: *Artificial Life*, VDM Publishing House Ltd. (2009).
26) Mundt, T.: Location dependent digital rights management, *Proc. 10th IEEE Symposium on Computers and Communications*, pp.617–622 (2005).
27) Nicholson, A.J., Corner, M.D. and Noble, B.D.: Mobile Device Security Using Transient Authentication, *IEEE Trans. Mobile Computing*, Vol.5, No.11, pp.1489–1502 (2006).
28) Noble, B.D. and Corner, M.D.: The Case for Transient Authentication, *Proc. 10th ACM SIGOPS European Workshop* (2002).
29) Potter, A.: A discourse approach to explanation aware knowledge representation, *Explanation-aware computing*, Roth-Berghofer, T., Schulz, S., Leake, D.B. and Bahls, D. (Eds.), pp.56–63, Menlo Park, CA, AAAI Press (2007).
30) Quercia, D.: Trust Models for Mobile Content-Sharing Applications, PhD Thesis, University College London (2009).
31) Reeves, B. and Nass, C.: *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*, Centre for the Study of Language and Information (2003).
32) Schilit, B.N., Adams, N. and Want, R.: Context-Aware Computing Applications, *Proc. Workshop on Mobile Computing Systems and Applications*, pp.85–90, IEEE Computer Society (1994).
33) Schiller, J. and Voisard, A.: *Location-Based Services*, Morgan Kaufmann (2004).
34) Shankar, N. and Arbaugh, W.A.: On Trust for Ubiquitous Computing, *Workshop on Security in Ubiquitous Computing, UBICOMP 2002*, pp.44–54, IEEE Computer Society (2002).
35) Sillence, E. and Briggs, P.: Ubiquitous Computing: Trust Issues for a "Healthy"

Society, *Social Science Computer Review*, Vol.26, No.1, pp.6–12 (online), DOI:10.1177/0894439307307680 (2008).
36) Sun, D.-Z., Huai, J.-P., Sun, J.-Z., Zhang, J.-W. and Feng, Z.-Y.: A New Design of Wearable Token System for Mobile Device Security, *IEEE Trans. Consumer Electronics*, Vol.54, No.4, pp.1784–1789 (2008).
37) Tanviruzzaman, M., Ahamed, S.I., Hasana, C.S. and O'Brien, C.: ePet: When Cellular Phone Learns to Recognize Its Owner, *Proc. SafeConfig'09*, pp.13–17 (2009).
38) Wang, Y., Li, M., Dillon, E., guo Cui, L., jing Hu, J. and jian Liao, L.: A Context-aware Computational Trust Model for Multi-agent Systems, *Proc. IEEE International Conference on Networking, Sensing and Control, ICNSC 2008*, pp.1119–1124 (2008).
39) Weiser, M.: The Computer for the 21st Century, *Scientific American*, Vol.265, No.3, pp.66–75 (1991).
40) Ziebart, B., Maas, A., Dey, A. and Bagnell, J.D.: Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, *Proc. UBICOMP: Ubiquitious Computation* (2008).

(Received November 4, 2010)
(Accepted April 8, 2011)
(Original version of this article can be found in the Journal of Information Processing Vol.19, pp.231–252.)

Stephen Marsh is a Research Scientist at the Communications Research Centre, an Agency of Industry Canada. He holds a Ph.D. in Computer Science from the University of Stirling, in Scotland (1994) for his seminal work on Computational Trust. He is an Adjunct Professor at Carleton University, University of Ontario Institute of Technology, and the University of New Brunswick. His research interests include Computational Trust and Comfort, Critical Infrastructure Interdependencies and Information Security.

Pamela Briggs is Dean of the School of Life Sciences at Northumbria University, and Co-Director of the Psychology and Communications Technology (PACT) Lab part of the Centre for Cognition and Communication. She also holds a Chair in Applied Cognitive Psychology. Pam's main research interest is in computer-mediated communication and she has acted as principal investigator on a number of research council projects. She recently completed a project on trust in online information and advice and she has an ongoing project on trust, identity and privacy issues for Ambient Intelligence.

Khalil El-Khatib was an assistant professor at the University of Western Ontario prior to joining the Faculty of Business and Information Technology, University of Ontario Institute of Technology, in July 2006. He received a bachelor degree in computer science from the American University of Beirut (AUB) in 1992, a master degree in computer science from McGill University in 96, and a Ph.D. degree from the University of Ottawa in 2005. Between the years of 1992 and 1994, he worked as a research assistant in the Computer Science Department at AUB. In 1996, he joined the High Capacity Division at Nortel Networks as a software designer. From February 2002, he worked of Research Officer in the Network Computing Group (lately renamed the Information Security Group) at the National Research Council of Canada for two years, and continued to be affiliated with the group for another two years. His research interests include security and privacy issues in wireless sensor network and in mobile wireless ad hoc networks (MANET), cloud computing, biometrics, ubiquitous computing environments (smart spaces), E-health, IP telephony, feature interaction for VoIP, QoS for multimedia applications, and finally, personal and service mobility.

Babak Esfandiari is an Associate Professor in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. He holds a Ph.D. from The University of Montpellier, France. His research interests include Agent-based systems, network computing and management, and symbolic machine learning.

John A. Stewart is a Computer Network Researcher in the Network Systems Group at the Communications Research Centre in Canada. He obtained his B.Sc in 1982; involved in various organizations focusing on X.25 and TCP/IP based networking. Network Coordinator for Canada's Defence Research Establishment's DREnet, including migration from ARPAnet to Internet and initial configuration of ".ca" domain. Recently involved in ISO-standardization of graphics, and represented the Web3D Consortium on the W3C HTML5 working group. A related project, FreeWRL, was the largest and longest-running open-source project managed by Canadian Federal Government.