Leveraging the subtleties of location

Simon Dobson

School of Computer Science and Informatics, University College, Dublin IE
simon.dobson@ucd.ie

Abstract

Virtually all pervasive computing systems use location as a major parameter governing their behaviour. Simple models of location ignore the richness that arises from humans' perception of location which – if leveraged – can greatly improve a system's ability to reason with location information. We explore how this richness arises, how it can be used to improve reasoning, and the challenges that this gives rise to.

1. Introduction

A large number of pervasive computing systems use some form of location for affecting their behaviour. Location-based services are available commercially, albeit in a primitive form, from many mobile telecommunications providers, and a number of more sophisticated systems have been discussed in the literature.

Despite this, location is a remarkably subtle concept to reason with. There are a huge number of possible answers to a superficially simple question such as "where is *x*". Each kind of answer reveals something about the way in which we conceptualise location and any services based upon it. Moreover no one approach is truly canonical, in the sense that *any* particular representation will be sub-optimal for some applications even if it is optimal for others.

In this paper we explore the ways in which a richer location taxonomy can be used to improve a pervasive system's ability to reason with location information. We focus on how the various views of location support various possible application domains, and how one can map between them to support a range of views within the same system. We draw five conclusions:

1. that we need to adopt a **multi-modal, multi-model approach to modeling location** in general pervasive computing frameworks – where as many as eighteen distinct models may be possible;
2. that we will be able to address a wider range of applications if we can **exploit automatically the structural mappings** between the various models;
3. that **location is a concept that spills over into other parts of the knowledge space** in some unexpected ways;
4. that, far from being constrained by sensors and wireless communications, the **essence of pervasive computing lies in synthesising data from a range of sources** to extract and utilise the maximum amount of available information; and
5. that **diverse information can improve both the reliability and precision of location-based services**, as long as it is used with due sensitivity to the errors involved in sensing and inference.

These suggest a more holistic and fusion-driven approach to location-aware systems, context modeling and systems development. We first review the location taxonomy we will be using, and explore the ways in which different applications domains are best supported by different parts of the taxonomy, and how we can maximise the impact of location through judicious use of uncertain reasoning and the structural mappings between views.

2. A taxonomy of location

When we speak of location we typically mean determining where some person or artefact is located in the real world. This can be used for a range of applications including adapting behaviour, controlling appliances, surveillance and healthcare[9].

Location sensor systems typically work in one of four ways[5]: direct object tracking; transponder-based object tracking; object-based environmental tracking; and inference from actions. Each approach has its strengths and all suffer from noise, occlusion and missed events. Rather than engaging in a critique of particular sensor technologies, we can instead start from the other end of the development spectrum and ask two questions: what are the **conceptual models** that a developer might use to reason about location?, and to what extent can these models be **realised directly** within a development environment?

Suppose for a moment that we are trying to locate our colleague Waldo without the use of any significant information technology – for example by ringing his home or office and asking where he is. We can imagine a whole host of ways in which this question might be answered. Each is a *plausible* and *correct* answer to such a location question, and – assuming a certain degree of optimality in human languages – should identify a conceptual space in which humans reason about location.

Without any claim to exhaustiveness, we have identified eighteen recognisably different answers:

At $53°4'N, 1°17'W$ (absolute position) As typically given by GPS and related systems. GPS location has no immediate connection with the real world, implying that a detailed map needs to be constructed of the features of interest. Behaviour is unlikely to be conditioned by co-ordinates *per se*, but rather by what (else) is at these co-ordinates.

In A1.15 (named space) A space identified by name in some agreed namespace, sometimes referred to as "white pages" naming, possibly with hierarchical inclusion of spaces. Logical naming is particularly well-suited to systems that infer location from other clues such as use of a computer keyboard. The location of the clue can typically be expressed quite neatly using a logical space name. There are of course some unstructured spaces that do not have meaningful names, and it is probably better to resist the temptation to invent them.

In a conference room (named class) A space identified by function or membership of some set, sometimes referred to as "yellow pages" naming. Such names are generally functional (as in this case), although it is conceivable that some other naming scheme might be used ("in a red room"?). This is an example of an *uncertain* location, in the sense that the possible locations not only have physical extent (which is true for most cases) but typically have *disconnected* physical extent.

In his (Waldo's) office (subject static space) A functional space related directly to the individual. Such locations may be precise in space but unstable in time, which we expand on further later.

In his car (subject dynamic space) A functional space whose location and connectivity with other spaces are not fixed. . So is it a location? – few would argue for a definite "no", but it is clearly a location of a different order to others.

In Widget and Sons' offices (related space) In a space defined by association with some other entity rather than the individual.bibliography.bib

With Willard (related association) This is location as co-location, defined relative to the location of some second party. If Willard's location is known, then so is Waldo's; of Willard's location is not known, then neither is Waldo's – although we know that they are together, which is sufficient for some tasks regardless if exactly *where* they are together. There is an obvious recursion if one asks where Willard is and receives the answer "with Waldo". However, a sufficiently rich set of possible location approaches should reduce the possibility of this happening in practice.

At 1000 he will be ... (in the future) Location expressed as a future expectation. This is important, as many applications will ask for location in order to prepare for a future event, and so this answer may be completely adequate: if the application is trying to arrange things for Waldo's 1000 meeting, then it is probably not germane that he is currently on a particular street. Indeed, this points to a weakness in many services conceived as location-based: it is not the exact current location that counts, but the *next relevant location for the application*. We return to this point later.

At 0800 he was ... (in the past) Location expressed as a previous observation or assumption. We might make educated guesses about Waldo's range of possible locations based on how far he can have travelled since his last sighting. The further we get from 0800, of course, the less reliance can be placed on this method.

Near/Within ... metres of ... (in vicinity) As typically found in Wi-Fi or Bluetooth network access points used as location sensors, although technically true of GPS too

Between ... and ... (on path) Location as an expectation on a path or elongated region. The interest of this form of answer is that it locates someone on a *path* rather than in a *place*.

Either at ... or... or ... (discrete set) In some situations Waldo's location may be narrowed-down to a small number of discrete locations, without any higher-level connection such as a path. A good example of this is "flip ambiguity" when triangulating[7].

His badge/phone was last seen at ... (by proxy) An indirect observation of something intimately connected to the individual. That this assumption of flawed is obvious: Waldo may have leant his cellphone to a friend, or may have been robbed of his badge. This entwines the location problem with an identity problem – which can be just as subtle. In applications that use artefacts as a surrogate for a person the identity is often split between "something you carry" and "something you know" – ATM cards are the most familiar example.

Meeting Widget and Sons (task) Location as involvement in a task, which can then be located.

At this time he is usually ... (by default) A default answer in the absence of any contra-indications. Absent any other information we may use default logic (in the formal or informal sense) to locate Waldo. People are far more regular than is generally realised – experiments show a frightening regularity in some specific cases[4] – so the use of defaults can be very powerful.

Not ... (by negation) Perhaps the least expected form of answer – and the most confusing from a computer science perspective – would be to answer a question of where someone *is* with an answer about where they *aren't*: surely this doesn't constrain the possible locations enough to be of any use at all. This turns out not to be the case: we encountered it in designing a system for a user with a physical disability, where action could bbibliography.bibe triggered by knowing that the individual had left home (to go to work), without actually being able to locate them otherwise. This style of response can be much easier to generate than any of the others, as it is inherently limited to a small scale. Nevertheless, many systems that are are conceived as location-dependent may actually be "non-location"-dependent, in the sense that they behave according to someone's non-presence in a location regardless of their actual location elsewhere.

Out/on holiday (non-located task) Involvement in a task without a specific usable location, which may narrow-down where Waldo is *not* as above.

No idea (unknown) Most designers would expect this answer some (if not most) of the time. However, it should be clear from the foregoing that it can be made almost arbitrarily unlikely in practice by combining fragments of knowledge from other sources. Some of these may have little obvious relevance to location but – with enough information and (admittedly uncertain) reasoning – can be used to contribute to at least some form of answer.

The full taxonomy is shown diagrammatically in figure 1. The diversity of answers suggests that modeling location in a system that aspires to generality will be a major challenge. However, much of the information needed to use most of the areas in the taxonomy is either available now or could be made available to a rich pervasive computing system.

3. The richness of location

In using absolute and/or named-space location we are in effect adopting a very simple spatial logic in which location is represented either as a point on a plane or a node in a tree. In fact space is much richer than this: a truly pervasive logic of location would include the *uses* of spaces, the *selection* of spaces, the *reconfiguration* of their topology, and so forth. Some of this richness is critical even for simple pervasive computing systems, and is often accessible even at the current state of the art.

3.1 Sources of location

Location is often seen as a matter of sensing and hardware. GPS, RFID and other technologies are sometimes promoted as the complete solution to location in pervasive systems. Leaving aside the limitations on resolution and deployment that apply to any technological choice, it is clear from the above analysis that no single model of location can capture the richness implicit in how humans think about it.

It is interesting – and perhaps revealing – to compare these models against the currently-available location technologies. Almost all the mainstream systems provide either absolute position or in-vicinity location. GPS uses an absolute global co-ordinate system, while BATS[11], Crickets[8] and UBISENSE use co-ordinates local to the space containing the transponders. RFID[10]-based location is based on transient proximity to a sensor. PlaceLab[6] is partially hybrid in that it uses triangulation of Wi-Fi or cellular signals to locate a user within a local co-ordinate system with a large degree of vicinity uncertainty. In each case, location information is either used directly or used to drive a view based around a simple hierarchy of named spaces – and ignore all the other views. From a user's perspective, of course, most statements about location will fall into exactly these other views. In fact, if we are willing to accept impreci-

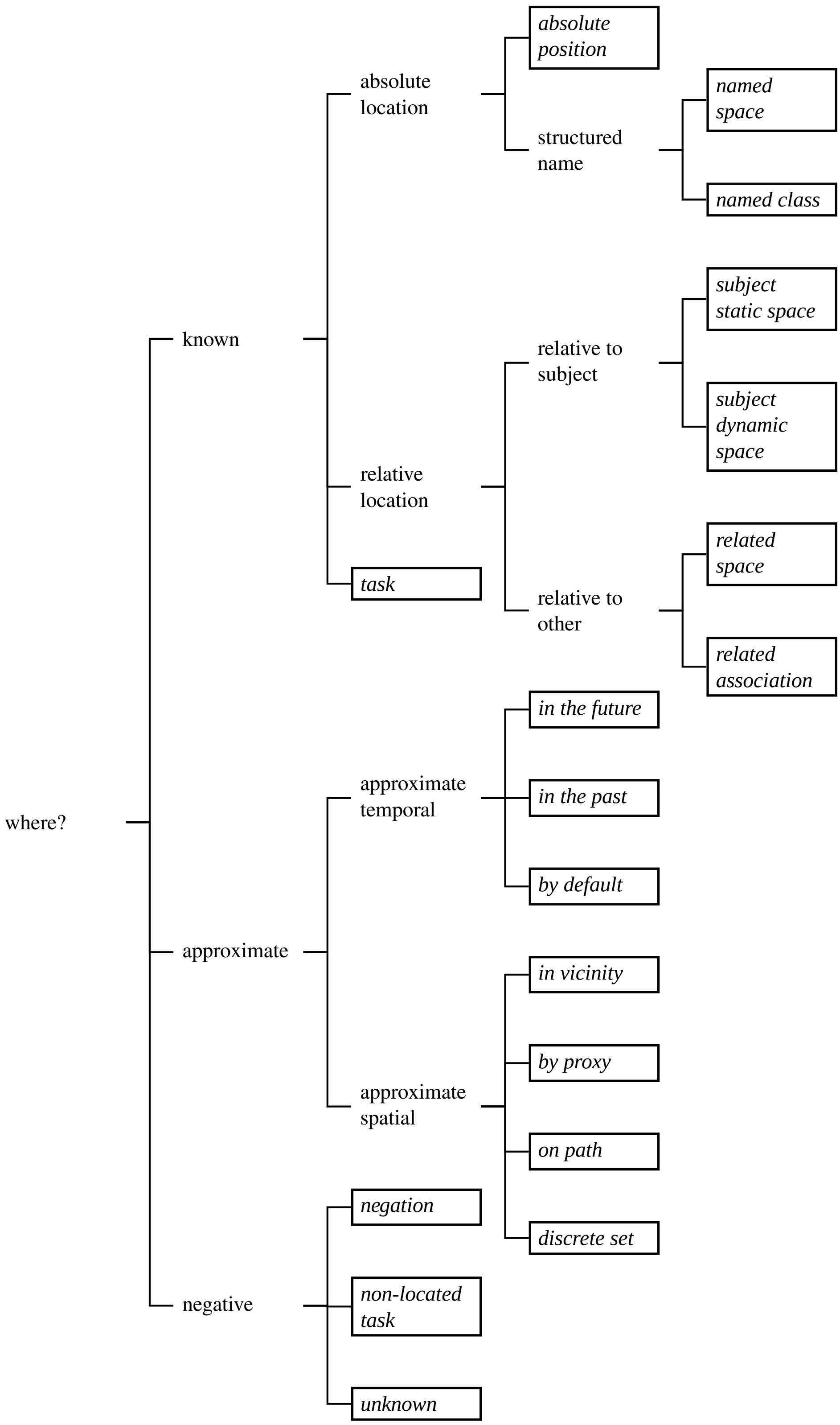

Figure 1: A taxonomy of location in pervasive computing

sion and uncertain reasoning, we can build location-sensitivity into a wider range of applications, and even consider location-based services with *no* explicit location hardware al all.

The key observation is that location information is implicit in many other information sources, albeit in an imprecise way. A meeting scheduled in a diary, for example, may have a location attached to it that can be used to infer the diary-owner's location at the time of the meeting, absent any information to the contrary. The point is that location information is implied by other pieces of information. We refer to this phenomenon as **contextual layering**: context is composed of layers, each describing an individual aspect of the world, and each potentially constraining the possible values of other layers. The resulting constraint system is what allows indirect inference of location (and other things).

The same raw information can be used to generate a number of views on location. Continuing the diary example we may present the subject's location as "with Willard" (the person he is meeting); "in the downstairs meeting room" (named space); "in a meeting room" (named class); and so forth. Similarly this inference may be supported by further information such as GPS co-ordinates related to a map. This sort of multi-view location system is potentially much more useful to a wider range of applications. The same information base can – if suitably linked and reasoned over – support a range of models of location.

3.2 The precision trap

An obvious counter to this argument is that one cannot trust information inferred from so tenuous a source as a user's diary: the basic information may not be up-to-date, does not account for real-world delays, *etc*. This is true as far as it goes: however, it assumes that all applications *require* up-to-date, accurate information to make their decisions – and this is far from being the case.

Although designers generally strive to obtain precise answers to questions about location, in actual fact there are many instances in which precision is unobtainable or unnecessary – and often both. If we adopt a view that the answers we give to location questions must be precise and known to be true (or, a little weaker, must have precision and confidence lying within tight intervals) we may outlaw a number of applications that do not need this level of precision.

Perhaps the best example of this is the completely negative case of answering a question of where a person *is* with information about where he *isn't*. This is actually enough information to build a range of applications, for example to lock the house when the owner is away regardless of where else they actually are.

There are further imprecisions. Location changes with time. When we ask someone's location we might aspire to receive an up-to-date answer, but will often have to cope with an answer that refers to a time in the past. Location statements are therefore time-bounded, in the sense that they "age" from being valid and relevant to being virtually useless.

We can even devise scenarios in which up-to-the-minute location is the *wrong* model. Suppose we have built a location-aware restaurant guide, and Waldo asks about for a lunch recommendation while on the train to his 1000 meeting. We do not care that the train is in (for example) Co Laoise: what we care about is that the meeting – the next time Waldo is in a position to use a restaurant – is in Co Cork, and this should be the location we use. In other words, we are actually asking the "wrong sort" of location question to be answered using (for example) GPS co-ordinates: the answer we want is intimately connected to the application we are building and the relationships with other contextual layers.

Location references are also less precise than they appear because they are unstable to perturbation. This is particularly noticeable for references that become unstable over time. We should not infer, for example for example that Waldo could see into Willard's office from his own *then* simply because he can *now* – the values of the referents may have changed before or since the observation was made.

The topology of spaces can also change. A car or train is a classic example – a "space that moves", and which has a dynamic connection with other spaces while at the same time having a distinct identity of its own. If we admit "mobile" spaces to a location model, we introduce dynamism into the map in terms of physical location and the accessibility of spaces from one another.

One could of course simply remove the notion of a car as bibliography.biba space and only allow "static" spaces, but this has two disadvantages. Firstly it is an unnatural and somewhat arbitrary decision to allow one kind of named space but not another. Secondly (and more importantly) there are useful behaviours a system might take when a person is in this space – switching a cellphone to hands-free, for example. These activities emphatically bind to the mobile space, not to the succession of static spaces the user may occupy.

3.3 Fully pervasive reasoning

Is there an ideal location system for pervasive computing? We would argue that the answer is "yes", but that it lies not in improving location hardware but in the ability to fuse uncertain information from the widest possible set of sources. We would contend that it is in this information synthesis, rather than in sensing *per se*, that the core contribution of pervasive computing resides.

What is needed for such a system? Given that information available in one layer can be used to infer (bounds on) information in other layers, we need firstly to be able to represent and reason with the information in various layers within a unified framework. A case can be made to regard each layer as controlled by a tool, but to combine the layers to allow reasoning independent of any tool, and indeed to control the selection of tools and the construction of applications by cross-layer reasoning[2].

Setting aside naïve examples, applications must assume that any query they make into such a model will deliver an uncertain result. Using path-based location, for example, does not give a subject's location but instead gives his *probable* location, *absent any information to the contrary*, on a particular path whose exact details may also be fuzzy. This does not sound like much information to work with, but is certainly better than none and may be completely appropriate for (for example) confirming reservations for hotels only when the subject actually begins a journey.

In general any query will also result in multiple answers, inferred by different reasoning paths. In order to arrive at a consensus answer suitable for decision-making, applications may need to be able to determine which answer is more certain (a standard problem for uncertain reasoning), and also which answers are wholly or partially the same but expressed in different terms. For example the answers "on the concourse of University College", "at 1230 he will be in the library", and "not in his office" might all support (with decreasing precision) a single contention about Waldo's location and should be taken *en masse*. This sort of reasoning can plug the gaps in location knowledge more reliably and cost-effectively than additional sensors.

3.4 Structuring

Rich models of context raise interesting challenges for programming environments. It is probably fair to say that traditional imperative and/or event-based approaches are insufficient when addressing complex reasoning across an extensible, multi-layered model.

However, pervasive computing systems do not vary their behaviour arbitrarily. There is generally a close link between the way a system reacts to context and the structured identified within that context. Location-based applications, for example, change their behaviours in response to users' moving from one place to another, where "place" may be defined using *any* view of location. If we allow these views to be used explicitly we can often capture *exactly* the "seams" in the context that cause behavioural changes[1]. This means that there is a close relationship between the model we use to represent location – and more generally any contextual layer – and the way we structure the behaviour that takes place there.

A good example is the behaviour of a wireless device with short-range communications. The device may be "pushed" information when in a particular place: but that place may involve a named space, a class of spaces, a co-location with some other user. and so forth. However, given a suitable definition of place, the *changes* in the behaviour of the system occur exactly when the place changes, and remain constant (or largely constant) within a place. This opens up the possibility of representing a complete adaptive behaviour as a closed form which may be analysed alongside the context that controls it. We have dealt with this topic more extensively in [3].

4. Conclusion

We have explored a taxonomy for location in pervasive computing systems and used it to derive opportunities and challenges for more advanced reasoning. We have argued for a more holistic, linked and inference-driven model of location – and by extension of *all* context – to support a wide range of precisions and models of context from a single richly-linked context base. We draw five general conclusions.

Firstly, most current models of location – and indeed *any* single view or small collection of similar views – do not leverage all the location information that is actually available to a pervasive system. This is true even for current systems, and is dramatically true when one considers systems engineered specifically to support shared context models and inferencing in the manner of [2]. This suggests that we can obtain better results by supporting a range of location models from a single knowledge base, allowing applications to choose the most appropriate model for them and using inference to drive this from the available context sources.

Secondly, the different views of location are structurally related in a number of interesting ways. This both facilitates reasoning and transitions between models, and provides constraints to remove noise. Many of these structural relationships form a strong correspondence between the context and the adaptive behaviour of applications, allowing behaviour to "emerge" from context.

Thirdly, location is not a discrete concept but is rather one that spills over into other parts of an ontology. It is possible to draw location deductions from other information, providing one has access to sufficient richness *and* one is careful in handling the uncertainties in the reasoning process.

Fourthly, the above discussion demonstrates that it is perfectly possible to build a location-adaptive system with *no* location hardware sensing capability at all. Given a rich link structure, one may compensate for the unavailability of some information by using inference, structural mapping and other more advanced techniques. This "network effect" is a key contribution of pervasive computing and is something that deserves significantly more exploration.

Finally, all context is to some extent unreliable. A context-aware system that relies on a single source of information to perform a critical function or inform a critical decision is fundamentally flawed. The potential of context aware systems comes, not from the use of sensors, actuators, wireless *et al*, but rather from the rich interconnection of diverse information sources and their use in synergy to build the most comprehensive view possible of a user's needs and intentions. Viewed from this perspective, a multi-modal, multi-model view of location coupled with similarly flexible programming structures is more likely to yield reliable ambient intelligence.

5. References

[1] Joëlle Coutaz, James Crowley, Simon Dobson, and David Garlan. Context is key. *Communications of the ACM*, 48(3), March 2005.

[2] Simon Dobson. Applications considered harmful for ambient systems. In *Proceedings of the ACM International Symposium on Information and Communications Technologies*, pages 171–176. ACM Press, 2003.

[3] Simon Dobson and Paddy Nixon. More principled design of pervasive computing systems. In Rémi Bastide and Jörg Roth, editors, *Human computer interaction and interactive systems*, volume 3425 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.

[4] Richard Greenane. Managing interactions in smart environments. Master's thesis, Department of Computer Science, Trinity College Dublin, 2002.

[5] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[6] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. Place Lab: Device positioning using radio beacons in the wild. In Hans Gellerson, Roy Want, and Albrecht Schmidt, editors, *Proceedings of PERVASIVE 2005*, volume 3468 of *LNCS*, 2005.

[7] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 04)*, 2004.

[8] Nissanka Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of MOBICOM 2000*. 2000.

[9] Debashis Saha and Amitava Mukherjee. Pervasive computing: a paradigm for the 21st century. *IEEE Computer*, 36(3):25–31, March 2003.

[10] Roy Want. The magic of RFID. *ACM Queue*, 2(7):41–48, October 2004.

[11] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Comunications*, 4:42–27, 1997.