ELECTRONIC PUBLISHING, VOL. 8(2 & 3), 259–272 (JUNE & SEPTEMBER 1995)

The stick-e document: a framework for creating context-aware applications

P. J. BROWN

Computing Laboratory
The University
Canterbury, Kent CT2 7NF, UK

e-mail: pjb@ukc.ac.uk

SUMMARY

There is increasing interest in computer applications that are aware of the user's context. Currently these applications are normally hand-crafted. A lot of them consist of presenting information to users as they enter a given context, for example a tourist nearing a site within a city or a visitor moving round a building. The paper presents a new form of document, and the supporting software, which allows such applications to be created simply by building a new document. The motivation is to make the creation and use of these applications as easy as creating and using web pages.

KEY WORDS Document Context-aware PDA HTML Stick-e

1 INTRODUCTION

This paper presents a new form of document, called a *stick-e document*, which is aimed at context-aware applications. It encompasses a metaphor whose purpose is to make such applications easier to create and understand. It covers a wide range of context-aware applications, but certainly not all, and removes the requirement that the creator of such applications needs to have computing skills; instead authorship just involves creating a stick-e document.

Although different in purpose, stick-e documents share a number of similarities with WWW documents, and with hyperdocuments in general. In particular, a body of WWW information, which we will call a WWW document (though some authors call this a WWW *presentation* and use the word 'document' to mean a page) is built out of a number of smaller components, the WWW *pages*. Similarly a stick-e document is built out of smaller components, which are called *stick-e notes*. Each stick-e note consists of two parts:

- the content, as is normal for any document.
- the context.

We begin by explaining the motivation for stick-e notes, and the metaphor that underlies them.

A good place to start is an application where the user carries a computer that knows where it is located. We shall assume that the computer is a PDA (Personal Digital Assistant),

CCC 0894–3982/95/020259–14

Received 22 March 1996
Revised 26 June 1996

like the current products of Psion, Hewlett-Packard (HP100, HP200, OmniGo), Apple (Newton MessagePad), etc. There are several current technologies for sensing position:

- the Olivetti active badge system [9], and the Xerox PARCTab [1], which are suitable for use inside buildings that have been appropriately wired.
- Global Positioning System (GPS) receivers, which sense their position anywhere on earth, provided they are outdoors.
- other schemes based on beacons, cellphone cells or even relatively humble bar-codes that are distributed round a building and are read by scanners carried by users.

If any of these devices is attached to a PDA, the PDA knows where it is. The metaphor that is the subject of this paper is that the user can then employ the electronic equivalent of a Post-it note: the user can, on the PDA, place a message in their current position (or indeed in any other specified position); whenever the user returns to the same position the message is *triggered* and the PDA brings the message to the user's attention. Thus the user has a personal equivalent of a Post-it note, which can be posted in any position in the space covered by the location-sensing technology, and which the user sees every time she visits that position. This electronic Post-it note is what we call a stick-e note: the content part of the stick-e note is the message, and the context is the current position. In practice the note will be attached to a rectangular or circular area, depending on the nature and granularity of the location-sensing system, rather than a single point. Stick-e notes can be used for exactly the same purposes as Post-it notes. They are personal in the sense that the only people who see them are the bearers of the computers that contain them. They can, however, be exchanged, published or broadcast, say over a wireless network, if it is desired to make them available to a wide audience.

2 GENERALISATION

The use of location is, in fact, only one example of a context of a stick-e note. In general the context part of the note can be a combination of elements of the environment that the user's computer knows about. This gives a stick-e document the capability of covering a variety of context-aware applications: indeed it represents the essential tenet of context-aware applications. Examples of elements of the context are:

- *location*, as we have already explained.
- *adjacency of other objects*: in an environment where people wear active badges, it is possible to detect which active badge wearers are in the same room. Given this technology, an element of a context of a stick-e note on John's computer could be 'in the presence of Bill'. In this case the stick-e note has notionally been attached to Bill, and whenever John meets him this note is triggered. Active badges can also be attached to equipment, and thus users can effectively attach stick-e notes to equipment, too. Extending the example, farm animals sometimes carry transponders that identify each individual animal. In this case stick-e notes can be attached to a farm animal. In addition to technology such as active badges or transponders that need to be attached to the bearers, my colleague John Bovey, has suggested detecting people, animals or equipment by characteristic sounds. As a specific example, species of bird can be recognised by their songs (though not the mute swan), and if your computer had a bird song recogniser, you could notionally attach a stick-e note to,

say, every chiff-chaff in the world. Whenever a chiff-chaff came into your presence (and sang) the appropriate stick-e note would be displayed.

- *critical states*: as an example of a critical state, assume that the computer has a temperature sensor attached. A stick-e note can then be given the context ‘temperature $< 1°$C’. The note is triggered when the condition is satisfied. The same metaphor covers share-price pagers, which may be triggered by ‘price of BP $<$ 400’; this is an example we shall explain in detail later.
- *computer states*: clearly the computer is aware of its own internal states, and stick-e notes can be attached to these states. For example a stick-e note could be attached to a certain directory, and this note would be triggered every time the user entered the directory.
- *imaginary companions*: all the contexts above have been detected entirely by the computer. An interesting, and indeed amusing, extension is to have contexts specified by the user, and set by the user by computer commands such as ‘I am now with ...’ or ‘I am not now with ...’. Clearly it must be extremely simple for the user to issue such commands or they will never be used; they might be as simple as making a single gesture with a pen. Such commands allow the user to specify imaginary companions, which we call *spirits*. (They can also be used for real companions, e.g. to register the presence of someone who is not carrying a badge, in order to see if there any notes attached to them.) Users can define spirits and then attach notes to them. For instance if the user has a hobby of map collecting, he might define Mercator as a spirit. When the user has occasional thoughts about maps, he attaches a note to Mercator. In other words he creates a stick-e note with ‘in the presence of Mercator’ as the context. When he settles down to his hobby, he says he is with the spirit of Mercator, and the notes are duly triggered. Here the stick-e note metaphor covers applications that might normally be done by e-mail (to an imaginary user) or by placing files in certain directories.
- *time*: a stick-e note can have a time or a range of times in its context: as an example a note with a future time as its context is the same as a diary entry.

Contexts can be made more general by allowing elements to be combined, though inevitably this adds complication and might be disallowed if simplicity of authorship was at an absolute premium. If a context contains two different elements, e.g. Fred and Bill, then the note is only triggered if both conditions are satisfied, i.e. if Fred and Bill are both present. Similarly the context that specifies Bill and Room 115 is only triggered if Bill is with the user and the user is in Room 115. In our bird song example a context could be Blue Tit and Great Tit (which would be triggered if both a Blue Tit and a Great Tit were in the presence of the user), and the note associated with this context might cover how to recognise the differences between the two. Elements can also be ORed together, e.g. Bill or John, or made *generic*, e.g. every member of the XXX group or even ‘any’ (see below).

Techniques exploiting many of the above aspects of a user’s context have already been pioneered by researchers at Xerox PARC [7], who are working with the *PARCTab*. The PARCTab acts as a combined active badge and graphics terminal. The PARC researchers describe a form of electronic Post-it note based on users sharing access to a directory on a central server for their PARCTabs. They also describe ‘Contextual reminders’, which cause messages to be triggered on a researcher’s PARCTab according to their location, their environment (people and equipment in the same room) and the time.

3 NOTES AND DOCUMENTS

Stick-e notes can be exchanged and published. The unit of exchange is not normally a single stick-e note, but a stick-e document representing a collection of notes. (Another parallel with hyperdocuments applies here: a component, the note, can be encompassed in several different documents.) Examples of documents are:

- a set of notes providing a tourist guide to a city.
- a set of notes relating to an activity, such as the Mercator notes for our map-collecting example.
- a set of notes representing a route and delivery instructions for a courier.

In most applications that we have considered the end-user will have a mobile device such as a PDA, though, as we shall see, there are also applications on static computers. If we assume the PDA case, dissemination can be achieved in any of the following ways:

- a document may be distributed by means of the Internet, and down-loaded into the PDA.
- a document may be exchanged between two PDA users by means of beaming.
- a document may be loaded into the PDA over a wireless link. This allows applications where the underlying information, and hence the document, is changing rapidly. An example would be a stick-e document covering current traffic problems. Individual notes within the document that are of relevance can be sent regularly over the wireless link according to the user's current context, specifically their position. (This application might involve an agent acting on behalf of the user to find the relevant notes [2]. It is not, in fact, an area we have explored in much detail, partly because there is a huge investment in it already, and we are more attracted to virgin territories.)

In general, some notes will be personal to the user and will never be published; an example would be some notes made on the outward leg of a walk, to help the user find their way back. In these cases the author and user are one and the same. In other applications an author might prepare a stick-e document with the specific aim of disseminating it to a large number of users.

3.1 SGML representation

In order to facilitate publishing and generally to make stick-e documents portable we have represented documents in SGML form. An example of a note, as encoded in SGML, is:

```
<note some attributes>
<required>
<with> J.D. Bovey <or> X. Chen
<at> co-ordinates of location
<content>
This is the content.
</note>
```

As can be seen, SGML tags cover the various elements of the context, such as companions (`<with>`) and location (`<at>`). The context is introduced by the `<required>` tag. There can, in fact, be both 'required' and 'casual' contexts, but we will not go into details here. For a deeper discussion, which also covers aspects of scaling up to large stick-e documents, plus issues of distributed systems, see [2].

Contextual elements can, if desired, be given *generic* settings, such as `any` or `none`. In particular `<at> any` matches any location and is suitable for a background note. In the above SGML example, the content part is a trivial one, being a line of text. In general the nature of the content is tailored to the module – called SESHOW, see below – that is used to display the note. As examples the content might be:

- in-line text, as in our example, which is simply displayed on the screen.
- an HTML page, which is displayed by a WWW viewer acting as SESHOW.
- a program that is executed by a shell acting as SESHOW.
- a set of instructions to control a mobile machine; here the user is the machine, not a person, and SESHOW controls the machine by obeying the instructions that are triggered by the machine's current context, thus causing the machine to act according to its context.

A document is an ordered set of SGML-encoded notes. We have explored two approaches to representing the document. In the first approach the notes are stored end-to-end within a single file – the whole being enclosed within the tags `<notes>` and `</notes>`. In the second approach each note is stored in a separate file, and the document simply consists of a sequence of filenames, representing the notes that encompass it; thus a single copy of a note can easily be made to belong to several different documents. The relative advantages of the two approaches are essentially the trade-off between extra flexibility and extra complexity.

3.2 Management and display of notes

At any one time, the user will have a number of stick-e documents stored in their computer: the user will make some of these documents *primed*, i.e. ready to trigger when the appropriate context is reached, and some will remain passive, i.e. they will just be in store ready for possible future use. For example the user may have several documents representing city tours, but only one of them may be primed.

Each note is triggered when its context is entered. It is simplest to think of triggering as an event like mail arriving: the user is notified that something is waiting. If the user enters a context, and then leaves it and re-enters it, the note will normally be triggered again. However each note can have a *Time-span* associated with it, which controls the frequency of such re-triggering (see later).

3.3 Previewing

We have introduced one element of pretence into the metaphor with the imaginary companions. It is useful to introduce another element of pretence: pretending you are in a given context. For example if the user wants to see what notes are attached to a given location a convenient way of doing this is to pretend that they are in that location and look at the notes that are triggered. This provides a good way of previewing notes. Indeed a user may want to preview a whole document by pretending they are in each of its contexts in turn. This is equally true for authors, who may want go through all of the notes within a document that they have created. As an extra facility, stick-e documents also support the concept of *tours*, which allow the author to present sequences of notes to the user.

4 SUPPORTING SOFTWARE

We hope that at this stage it is evident that the metaphor is both simple and wide-ranging. It brings together diverse applications ranging from diaries to the emerging route-marking applications found on GPS receivers. Most of the applications apply best to mobile users, because it is here that the context changes most often.

The metaphor can be realised in software either by:

- *generic software* that has the potential to cover all the kinds of context that we have outlined above.
- *specialised applications* such as an application designed for tourists to a city, where all the notes have a context that contains a position.

Obviously the specialised applications can have a more convenient user interface: in the tourist application the whole interface might be based on a map of the area, and a further simplification is that the users might be assumed to be passive readers rather than authors of notes.

The software to support stick-e documents contains the following components:

- *SEPREPARE*, which allows authors to prepare notes and documents.
- *SEMANAGE*, which deals with management of the documents that are primed. This component will usually be relatively trivial, but will be more complex in cases where notes are being continually broadcast, and an agent is needed to decide which of the broadcast notes to take as primed notes.
- *SETRIGGER*, which runs in the background and causes any of the currently primed notes to be triggered if its context becomes satisfied.
- *SESHOW*, which stores the notes that have been triggered and presents them to the user. There might be many alternative SESHOW components running at the same time, e.g. a tourist application like the one outlined above, plus a pager version of SESHOW that deals with paging notes, plus perhaps even a web browser.

If these components are related to a mailer, SETRIGGER corresponds to the component for receiving mail and SESHOW to the component(s) for presenting it to the user.

In many document preparation systems, especially in hypertext, the boundaries between author's software and reader's software have been largely removed: for example the reader might be able to edit the document that they are reading and, indeed, invoke any other authorship facility. Thus an SEPREPARE component may simply be an extended version of an SESHOW component.

The above components can, in general, run on different computers, and be connected by an arbitrary distributed system [2]. In the model we have implemented, an SESHOW module is 'in control' and periodically asks SETRIGGER to give it all the newly-triggered notes; the fall-back position, if there are no newly-triggered notes, is to supply those that have already been triggered and which remain in context (this is useful for background notes, which come up when there is nothing better). Several notes may be newly triggered at a time, and, when this happens, the order in which they are presented is the same as the order of the notes within the document that encompasses them. This gives authors some, albeit crude, control over the order of presentation. We have normally presented the triggered notes to the user end-to-end, one after the other, though when experimenting with a WWW viewer on a large screen we have used a presentation based on separate frames (i.e. each note is a separately scrollable area of the screen).

An additional facility, which has turned out to be surprisingly useful, is an option to display, before the content of each triggered note, an explanation of *why* it has been triggered. This was originally put in as an aid to authors trying to debug documents that contained notes with complex contexts (‘Why on earth was that triggered?’). However it proved useful to end-users too, who might be unaware that they had entered a given context. Consider as an example a note that has the context ‘on next meeting Jim’. If the user enters a room with several people in it, they might not be aware that they were with Jim, and hence an explanation

```
COMPANION MATCH: Jim
```

on the front of the content is a great help. An addition benefit is that it saves the author the tedium of repeating the context in the content of a note: on our sample note the content, instead of saying ‘When you next meet Jim, discuss ...’ can simply say ‘Discuss ...’.

5 APPLICATIONS

We will now consider some applications of stick-e notes in more detail.

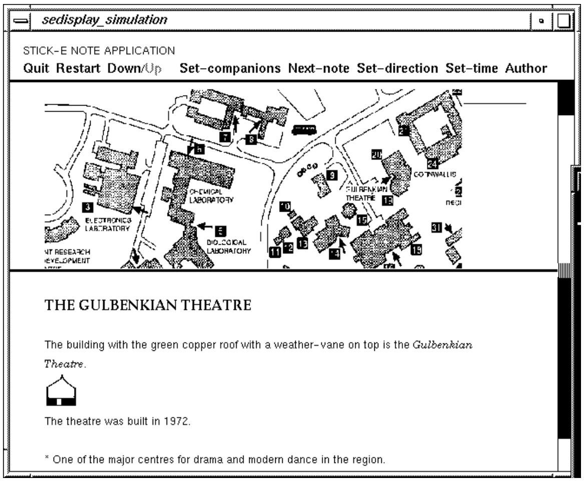
5.1 Tours

We will start with an application covering tours. These could relate to museums (indoors or outdoors), tours of cities, walks, car tours, etc. We assume the user has a hand-held PDA with a location sensor attached (GPS, active badge or bar-code, depending on the nature of the application). The equipment might be rented out from the tourist office, and would come pre-loaded with the appropriate stick-e document, which would already be primed. This removes the need for the user to worry about the SEMANAGE aspects.

This application has indeed been implemented, albeit in a prototype form that runs on a static workstation with a simulation of the user’s movements.

As can be seen from Figure 1, the user interface consists of a map, taking about half the screen, together with the current stick-e note. The user’s current location is shown on the map – ideally the map should always be centred on the user’s current location, but this is not done in the current prototype implementation – indeed the map facility is a desperately crude one in the prototype. Stick-e notes cover the interesting points on the tour; there is also a ‘background’ note, whose context is any location. This gives background information about the tour.

We said earlier that users might want to preview notes for contexts other than the current one, and a map interface provides a particularly good way of doing this – it would, of course, be foolish to expect the tourist to use some arcane notation to specify the pretended context. (In fact the underlying arcane notation is our SGML mark-up: pretended contexts are represented in the same way as contexts attached to notes.) In this case the user sets a pretended location by simply pointing at a location on the map: the stick-e note(s) for that location then appear, and the tourist can judge from this whether it is worth visiting the



THE GULBENKIAN THEATRE

The building with the green copper roof with a weather-vane on top is the *Gulbenkian Theatre*.

The theatre was built in 1972.

* One of the major centres for drama and modern dance in the region.

Figure 1. The tourist prototype

spot. At an extreme, the whole stick-e document can be used by a surrogate traveller who was in reality thousands of miles from the area and therefore never caused any of the notes to trigger by being at their location.

The application, as described so far, may have some attractions, but one could reasonably take a sceptical view: if the application was indoors, you could avoid all this technology simply by having paper notices displayed; even if the application was outside and electronic technology was appropriate, one can imagine equally effective versions of the application which used a different metaphor – indeed they probably exist already.

In general the strength of computer-based information presentation over the paper-based equivalent is that the former can adapt to the user's needs. The strength of the stick-e note metaphor – or so we have argued – is that it brings diverse aspects together. If the application we have described is to be convincing it needs to exploit these aspects, and this we have done. In particular it supports a rich range of contexts – not just location, in order to meet a wider range of user needs.

One way in which contexts are made richer is that they can incorporate the time – all PDAs know the time. Thus there might be one note for location X for the time 9-5, and another note for other times. On a larger scale there can be one note for a garden location for times in the spring, another for times in the summer, etc. This time element, then, is an aspect of the context that the user does not control, and may not even be aware that it is being used, but gives the author more flexibility to present pertinent information. (Such features have been available on surrogate travel systems for many years, for example in the *Aspen Movie Map* videodisks developed by Lippman and his colleagues at MIT in 1978.)

An extra element in the pretence provided by surrogate travel systems is that the user can choose to view information as it was at some time in the past (see, e.g., the Palenque system [11]). This is covered by the stick-e metaphor: it is possible for a context to include a time in the past, e.g. a note might apply to a point in Canterbury as it was in the seventeenth century; a user can cause such notes to be triggered by selecting an option to pretend the time is the seventeenth century rather than the present (the default). Unlike the examples above, this requires user interaction.

A second way that contexts can be made richer is to allow the user to change the information displayed according to their preferences. The natural way to do this using the stick-e metaphor is for the user to specify imaginary companions: thus the author of a stick-e document might specify some extra notes where 'in the presence of an architect' was part of the context; these notes would cover architectural features of interest to specialists. The user can be presented with a menu that allows them to select an architect as an imaginary companion, and users who pick this companion will see the extra notes. Thus the stick-e note metaphor here relates to the imaginary companion metaphor used by Apple with the guided tours of HyperCard stacks about American history [6]. In the stick-e case, the user can have any number of simultaneous imaginary companions, and can, at any time, add a new one or remove an existing one that has become a bore. This mechanism can be used either to enhance or filter the standard information: the standard information may have an imaginary companion Joe Bloggs, the typical visitor, and you could perhaps add an extra companion to extend your interest, or replace Joe Bloggs with Joe Bratt, who was only interested in fast-food outlets. We have successfully used this technique in our prototype (using the Set-... commands that can be seen in the menu in Figure 1), and have used stick-e notes to cover both the tourist sites and the routing information from one site to the next.

The imaginary companion metaphor can also be used to cover versions of the notes in different languages: if you had an imaginary Frenchman with you, you would see the French version of the notes. Indeed this might even be the reality: you, an English speaker, might have a real Frenchman with you, and you might want to bring up the French version for him to see.

So far we have discussed this application from the user's point of view. We will finish by considering how a stick-e document for tourism can be created by its author. In our prototype this was done by writing raw SGML, but a better way would be to create an authorship package that created the SGML. We envisage that the author would create the content of a note by using a word-processing package, and would create the context just by pointing at elements on the screen: a location on the map, a time on a clock, a companion from a list of people, etc. Indeed, if the author has a PDA plus GPS etc., the information can be prepared *in situ*, e.g. the author writes the bit about the view of the cathedral when she is actually viewing the cathedral, and the current location is automatically set as the location of each note the author prepares. Such an interface would help ensure that authors only needed a minimum of computer skills; certainly, even in our prototype, no more skills are needed than the equivalent of the skills to prepare, say, a web page.

5.2 Personal pagers

The stick-e metaphor covers personal pagers. In practice paging requests are normally broadcast, so we will assume a user with a PDA that has a wireless network connection that can field paging requests. A paging request for person X is simply a stick-e note with person X as the context. When X receives the note, since, by definition, X is always in the presence of X, the note triggers immediately. The metaphor is slightly more general, however, than traditional paging. A separate person, person Y, can accept a paging request for person X and prime this note. (Assuming the paging request is broadcast, Y's SEMANAGE agent might accept paging requests for certain other people such as X; alternatively Y might consciously take the paging request for X, knowing he might meet X.) Then if Y subsequently meets X the note will trigger; this could be useful if X had no connection to the paging network. To generalise this further, X need not be a person: X could be, say, a cow, or a library book (with an active badge in it); then the paging request is effectively a request to third parties to take a certain action if they encounter X.

5.3 Mobile workers

The metaphor can be applied in designing applications for mobile workers who visit homes, offices, electricity substations, etc. Couriers can, for example, attach stick-e notes to the buildings they visit, giving, say, delivery instructions (‘Go round to the second door at the back; mind the dog.’). These could be personal to one courier, or could be passed from one courier to his successor. Each day the courier company might have a set of deliveries to be made. This can be issued as a stick-e document: one note with the document might, for example, have the context of a certain building and its content might be ‘deliver parcels X123 and Y456 (recorded delivery) here’. As the courier travelled round, these notes, together with the personal notes, would be displayed as their context was entered. (Thus the courier could have two documents that were primed at the same time: the document issued by the company, giving the day’s delivery instructions, and a document containing their personal notes about various locations.) For some mobile workers, salesmen for example, notes might be delivered over a mobile phone link to cover new prospects that had arisen since a salesman left home. (Scientists often have little control over whether their ideas end up being used for good or ill; it would, however, be sad if the only successful use of stick-e notes turned out to be as an aid for double-glazing salesmen.)

A current problem with these applications is matching the granularity required to identify buildings, etc., with the granularity provided by the location-sensing device. Differential GPS (DGPS) can, however, provide accuracy to an arbitrary precision, down to less than a metre.

5.4 Rolling surveys

There are many applications where field workers gather data about geographical areas: the height and state of crops in various rectangles, the geological structure of the ground, etc. In its simplest form these applications require the field worker to divide their area into rectangles (or perhaps, other shapes), and to visit each rectangle and post a stick-e note to record the data. As with our suggested authorship package for the tourist application, the current location can automatically be attached in the note the author writes. In addition, for this application, it is necessary that the date of authorship be recorded too. When they next visit a rectangle to re-survey it, the previous note will be displayed and they will prepare a new note to record the latest data. Perhaps all the previous notes will be kept in the PDA to constitute a historical record, but the priority rules should be set so that, by default, the most recent note is displayed to the user.

Documents representing a collection of notes for an area may be exchanged between colleagues and published, and might be supplemented, as with the courier application, by the personal notes of the field worker, e.g. one position might have the note ‘Mind the barbed wire’.

In the case where the data may be of wide interest, such as a geological survey, the results might be published to the general public. For a user with the right equipment (PDA + GPS receiver) such data would be easier to use than a traditional paper map, because the triggering mechanism would cause the data for the current rectangle always to be displayed. Note, however, that the stick-e technology is only suitable for data covering discrete areas, rather than for continually changing data.

The SESHOW and SEPREPARE modules for this application should have a knowledge of the rectangles (ideally of equal size and butting) so that co-ordinates of the location-sensing system can be translated to the appropriate rectangle.

5.5 Control panels

A frequently-quoted context-aware application is the personal control panel: when the user is close to a device such as a copier or FAX machine, she can bring up a control panel on her PDA and, assuming the appropriate communication is supported (e.g. an IR link), control the device. As my colleague Jason Pascoe has observed, a stick-e document can be used to trigger such control panels: in this case SESHOW would be a special module for implementing these control panels, and the content of the stick-e document would be specifications, understandable by SESHOW, of the control panels to be used (one in the context of a certain copier, another in the context of a FAX machine, etc.). The application depends, of course, on there be some way of detecting the proximity of such machines.

5.6 Share price paging

Our final example, and another one that has been implemented in prototype form, is paging for stock exchange prices. Currently there are various commercial products that provide this service: the user specifies a set of thresholds on prices of shares, e.g. BP < 400 or IBM > 100, and the user is alerted if any threshold is passed. The service may either be based on a mobile paging devices or on a static workstation – our prototype took the latter approach.

The key to implementing this is a contextual element, called a *relation*, that can be applied to stick-e notes. A relation consists of an identifier, a relational operator (e.g. '<') and a value. The current environment contains the values of the identifiers, and a note is triggered when its relation holds. Thus a possible context is:

```
<relation> BP < 400
```

and this would be triggered if the current value of BP was less than 400. In our implementation, which is UNIX based, the identifiers are UNIX environment variables. A feed from a stock exchange price information service periodically updates some of these environment variables. For example BP might be set to 397, and in this case our sample note would trigger.

An interesting facet of this application is that the content of all the notes was null: they just consisted of a relation. The facility for explaining why a note was triggered was switched on, and this gave all the necessary information, e.g.:

```
RELATION MATCH:
  BP < 400 with BP = 397.
```

As we have said, the power of stick-e notes is likely to come from generalising existing applications by introducing new contextual elements. The share price application, however, appears to offer less scope for this than most, though one could, perhaps, introduce time-of-day and location, e.g. ‘BP < 400 when I am at location ... [the office] and it is between 9 and 5’.

6 FURTHER FEATURES OF NOTES

As a result of considering these and other applications, the need for notes to have a number of additional properties has been identified. The extra properties are not of concern to users, however, and only to some authors. Hence the burden of extra complexity that each new property inevitably brings is probably a burden worth bearing.

Firstly, notes need to have a number of optional *Time-span* fields to control how often the notes are to be repeated. Thus a paging note, once triggered, does not need to be triggered again; moreover if it is not triggered within a certain time, a day say, it should be discarded. Other types of notes might be capable of being re-triggered arbitrarily often, but there might be a minimum delay between triggerings (five minutes, a day, a week). If there is no minimum delay there is a danger of ‘edge-effects’, such as hovering on the boundary between two areas, causing huge numbers of triggerings.

Secondly, there needs to be a priority system to control the order and urgency with which the notes are brought to the user’s attention.

Thirdly, there are questions of adding programmability to provide more generality. As we have described them, stick-e documents cover those context-aware applications where both the contexts and the content are fixed in advance, though if the content is a program that is subsequently to be executed by SESHOW, there can be a dynamic aspect. More generality can be achieved by allowing the content to be an *active document*. This need has also been identified in projects to show WWW pages on mobile computers [8], [3]. As a further elaboration there are applications where the context might be calculated dynamically: an example would be a share price pager that triggered a note when a share price had followed a certain pattern, or behaved better than a certain benchmark index. We are, however, straying away from our original tenets, which are that we are creating a system for authors who are not computer-literate; instead we are going into the areas ably covered by the researchers at Xerox PARC [10], with their community of computers scientists.

Finally, our experiments have identified strong needs for quality-control tools for authors: if the author is creating a large stick-e document then almost certainly some of the contexts will accidentally be wrongly specified. One such tool – though certainly not a perfect one – is a declaration, attached to a document, to say which contexts the author uses. This can specify the area covered, a list of the possible companions, etc. Given this declaration, it is possible for SEMANAGE, when it loads a document, to alert the author to anomalies, e.g. a mis-spelled name of a companion or way-out location attached to a note within the document. These declarations have been implemented, and have proved useful, but are only one step towards a good quality-control environment.

7 RELATION TO *AIR*

The stick-e metaphor is inspired by the idea of Activity-based Information Retrieval, pioneered by Lamming and Newman at Rank Xerox Cambridge Research Centre [4], and encapsulated in the work on aiding human memory [5]. Indeed the stick-e note metaphor is a specialised example of AIR, but one that concentrates on the future rather than the retrieval of past events. Stick-e documents have a strong connection with the memory project in that they can deal with distributed personal reminders: a reminder to do something when you enter a given room (real or imaginary), meet a given person, etc. This in turn relates to the previously-cited Xerox PARC work.

8 CONCLUSIONS

The accent in mobile computing, and context-aware applications in particular, is moving from the technology to its applications. Context-aware applications are fundamentally different from static ones, and need new metaphors. This paper has presented a candidate for such a metaphor, which seems to possess the necessary attributes of generality combined with simplicity; the metaphor has been realised in a framework based on documents.

Stick-e documents can be realised in software in a generic form, or in specialised application-dependent forms such as the examples described above. Given that ease of use is a prime requirement, the latter are likely to be more attractive.

ACKNOWLEDGEMENTS

I would like to thank several colleagues who helped with this paper, particularly Duncan Langford, who coined the term *stick-e*, Jason Pascoe and John Bovey. I would also like to thank the referees for valuable comments.

REFERENCES

1. N. I. Adams, R. Gold, W. N. Schilit, M. M. Tso and R. Want, An infrared network for mobile computers, *Proceedings of USENIX Symposium on Mobile Location-independent Computing*, Cambridge, Mass., pp. 41–52 (1993).
2. P. J. Brown, *Facilitating the creation of context-aware applications*, Computing Lab., Univ. of Kent at Canterbury (1996).
3. M. F. Kaashoek, T. Pinckney and J. A. Tauber, Dynamic documents: mobile wireless access to the WWW, *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pp. 179–184, Santa Cruz, California, IEEE Computer Society Press, Los Alamitos, Ca. (December 1994).
4. M. G. Lamming and W. M. Newman, Activity-based information retrieval: technology in support of personal memory, in F. H. Vogt (Ed.) *Personal Computers and Intelligent Systems*, (Proceedings of Information Processing 92, Vol. III), Elsevier North-Holland, Amsterdam (1992).
5. M. G. Lamming, P. J. Brown, K. Carter, M. Eldridge, M. Flynn, P. Robinson and A. Sellen, The design of a human memory prosthesis, *Computer Journal*, **37**(3), 153–163 (1994).
6. T. Oren, G. Salomon, K. Kreitman and A. Don, Guides: characterizing the interface, in B. Laurel (Ed.), *The art of human computer interface design*, Addison-Wesley, Reading, Mass., pp. 367–381, (1990).
7. W. N. Schilit, N. I. Adams and R. Want, Context-aware computing applications, *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pp. 85–90, Santa Cruz, California, IEEE Computer Society Press, Los Alamitos, Ca. (December 1994).
8. G. M. Voelker and B. N. Bershad, Mobisaic: an information system for a mobile wireless computing environment, *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pp. 185–190, Santa Cruz, California, IEEE Computer Society Press, Los Alamitos, Ca. (December 1994).

9. R. Want, A. Hopper, V. Falcao and J. J. Gibbons, The active badge location system, *ACM Transactions on Information Systems*, **10**(1), 91–102 (1992).
10. R. Want, W. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis and M. Weiser, An overview of the PARCTab ubiquitous computing experiment, *IEEE Personal Communications*, **2**(6) (Dec.), 28–43 (1995).
11. K. S. Wilson, Palenque: an interactive multimedia digital video interactive prototype for children, *Proc. ACM CHI'88*, Washington, DC, pp. 275–279 (1988).