FieldNote: a Handheld Information System for the Field

Nick Ryan, Jason Pascoe

Computing Laboratory, University of Kent, Canterbury, Kent CT2 7NF, UK.

{N.S.Ryan, J.Pascoe}@ukc.ac.uk

David R. Morse

Computing Department, Faculty of Mathematics and Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK.

D.R.Morse@open.ac.uk

Abstract

Research fieldwork in disciplines such as archaeology and the environmental sciences often involves construction of a GIS for analysis and visualisation of field observations. At either end of this process, initial planning and the final production of a synthesised model and published results are well supported by conventional GIS. In between, much effort is often expended on data entry, format conversion and other tedious tasks before the analysis can begin. Multiple field visits and the need to collate the work of different specialists may significantly increase the complexity and tedium of these tasks.

The FieldNote system described here integrates handheld tools for data collection and re-use into the process in a way that is minimally intrusive on the fieldworkers normal tasks. The concept of context-awareness is used to maximise the automation of data collection and retrieval in the field, and to enable rapid sharing of information between co-workers.

1. Introduction

The practical nature of fieldwork often means doing something else at the same time as recording data. Usually, this activity is intimately connected with the observation or measurement itself, for example, watching a giraffe through a telescope, or measuring the acidity of a river with a pH probe. In all cases, the primary task is observation or measurement, and the act of recording should interfere as little as possible with this primary task. Fieldworkers need to spend as much time as possible in observation, and have only a limited attention capacity to deal with recording, be it on a paper form, a tape recorder, a camera or a handheld computer. Elsewhere, we describe a prototype 'Minimal Attention User Interface' for handheld computer use when the fieldworker can only afford the occasional glance at the device [11][13].

Fieldwork requirements may include high speed interaction where observations are time-dependent and have bursts of activity, for example in traffic censuses or animal and human behaviour studies. Here, the fieldworker needs to enter large amounts of data rapidly and accurately. This may conflict with the need to record contextual information, essentially metadata, associated with the observations, for example the recorder's identity, their location and the time of recording.

Fieldwork often involves repeated visits to a field site. One visit may not be enough to collect sufficient data, or the intention may be to carry out long term monitoring to detect change, or to make observations under a variety of environmental conditions such as weather, or seasonality. In the field, it is often useful to have access to observations from previous visits. In a conventional field notebook, previous visits can be reviewed by flicking back to previous pages in the notebook. A handheld computer can offer more flexible systems for searching and displaying data from previous visits, or by other fieldworkers.

Planning and interpretation of results may draw upon information from other disciplines. An ecologist studying the present day distribution and abundance of animals and plants might be interested in the history of a field site, particularly the human influences upon it. In other words, they might be interested in the site as interpreted by an archaeologist. An archaeologist might wish to know about the ecological and geological processes that have shaped the past and present day landscape. Brown [3][4], in his work on context awareness, has used the metaphor of an 'imaginary companion' to illustrate how an electronic implementation of this concept might work. In the absence of real physical people, their presence can be simulated using Brown's [3] 'stick-e notes' and context awareness, as illustrated in the next section.

2. Context awareness

The term 'Context Awareness' was introduced by Schilit *et al*. [16] to describe a new class of software applications that exploit the changing environment of a mobile computer user (see also, Fitzmaurice [7]). These applications sense the environment in which they are running (their context) and adapt their behaviour accordingly. For example, in an office environment, the contexts of interest may include where you are, who you are with and what resources such as printers are nearby [16], although the range of physical or logical environmental variables which could be sensed or acted upon is potentially infinite.

An alternative approach to those of Olivetti and Xerox, where some of the important work on context awareness has been carried out [9][20], was proposed by Brown [3]. This focused on issues of representing, manipulating and managing context to facilitate the development of context-aware applications, in much the same way that authoring tools have greatly eased the task of producing multimedia World Wide Web pages.

Brown's metaphor for context-aware applications is based on the electronic equivalent of the well-known 'Post-It' note, which he has termed the *stick-e* note [3]. These are 'attached' to an environmental context and become active when the user enters that context. The event of a stick-e note becoming active has been termed 'triggering' [4], thus stick-e notes are triggered when their context becomes relevant to the user. Triggering might involve displaying the stick-e note in a suitable browser. A user could enter the context of a stick-e note actively, perhaps by moving to a particular location, or passively, through time advancing. This metaphor defines a means of delivering context-aware information that is particularly suited to mobile applications.

3. Mobile Computing in a Fieldwork Environment

In the Mobile Computing in a Fieldwork Environment project (MCFE) we have been investigating how inexpensive handheld computers connected to a Global Positioning System (GPS) receiver can be used for data collection and the authoring and delivery of field exercises [11][12][14][15].

Our early prototypes used a single HTML document for each recorded note, with context represented as Dublin Core style metadata [14][6] in the header section. Whilst convenient, this led to problems where metadata hidden in the header often needed to be exposed in the body of the document. Subsequently, we have modified this model to employ dynamically generated documents for presentation use, and pre-defined HTML forms for data recording.

4. The FieldNote System

The FieldNote system has been designed to support the full range of fieldwork-related activities from initial planning through data collection to collation, display and analysis. Its primary aims are

1. To integrate and optimise the efforts of a team of fieldworkers by providing a decision-support role to help in planning each successive day's "campaign".

2. To reduce the effort required in field recording by providing tools that
   - automate the collection of contextual information,
   - eliminate the need to transcribe data from field notes into the GIS/database,
   - intrude minimally into the main tasks of the fieldworker, and
   - are perceived as offering significant benefits in recording efficiency and ease-of-use over conventional paper-based methods.

3. To provide the fieldworker with access to relevant contextual information collected by colleagues.

The first and, to a lesser extent, the third of these aims might be addressed by taking conventional GIS software into the field and using it in much the same way as one would in a desktop environment. This is quite feasible with modern laptop computers. For the smaller project, simple replication of all or part of the main GIS on a laptop may suffice for overall project control and decision-support. However, when different groups of specialists need to share information or contribute to the overall development of the GIS, this solution may lead to difficulties in combining and synchronising the work of each group, particularly if work on the GIS continues at the team's base whilst others are collecting new data in the field. To overcome this problem, some form of active replication or mirroring between the main and field databases is required.

Whilst they are portable, laptops achieve their functionality at the expense of size, weight and battery life and can hardly be considered as *mobile* devices. They are well suited to work at a base-camp or in a vehicle but, for our second aim, smaller, lighter and less intrusive handheld devices are required.

The FieldNote system has three main components: a database for storage and retrieval of notes, handheld systems for data collection and access to contextual information in the field, and a desktop suite for editing, analysis and visualisation (Figure 1). These can be combined in various ways to suit the needs of different field projects, from small scale data collection by an individual through to long term collaboration by groups of specialists.

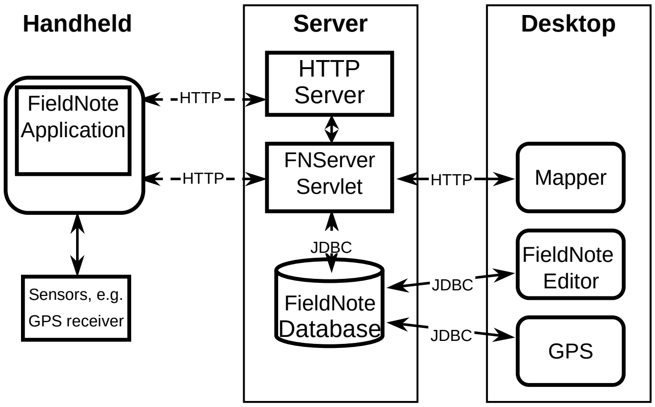


Figure 1: Basic architecture of the FieldNote system showing communication between handheld, database and 'desktop' applications.

4.1. The FieldNote Database

The database component stores recorded FieldNotes, GPS measurements and other spatial and attribute data. Applications may access the data directly using the facilities of the host DBMS, via JDBC, or indirectly via a Web server and Java servlets. The latter is the key to providing flexible communications between the fieldworkers' handheld systems and the shared data store, irrespective of the physical location of the database.

There are two implementations of the database. The first is largely experimental and employs an "Object-Relational" DBMS that has been extended to support a range of spatial and temporal objects. The use of this system as a background data store for a GIS has been described elsewhere [2]. It has a number of benefits including avoidance of an artificial distinction between spatial and non-spatial data, the implementation of spatial and temporal object behaviour as extensions to SQL, and an ability to support polymorphic queries. The latter provides a particularly useful facility that allows map layers consisting of different vector and raster data types to be composed dynamically and returned as a single result set by simple database queries.

The second database implementation uses an entirely conventional relational model. This was developed for two reasons; firstly it offers the possibility of straightforward integration with existing GIS software and, secondly, because it enables part or all of the data to be taken into the field on a laptop PC and manipulated by a simple DBMS such as Microsoft Access.

4.2. Communications

With both the main and field portable implementations of the database, an HTTP server is used for data exchange with the handheld machines. Various servers and database access methods were used in early prototypes before finally choosing Apache [1], a server that will run on both Unix and Windows NT or 95. Java servlets provide the data exchange mechanism, communicating with the handheld devices by HTTP and the database server by JDBC. The main tasks of these servlets are to parse incoming FieldNote data and perform necessary database updates, and to retrieve and serialise stored notes or spatial data for transfer to the handhelds.

Communications between handheld and server involve a small number of message types which are formatted as XML documents or document fragments [18]. The mobile device begins by sending an HTTP 'POST' request containing a `<context>` message to request a new session:

```
<context session=”new”></context>
```

and the server responds with a similar message containing a new session identifier:

```
<context session=”123”></context>
```

Once opened, a session lasts until a pre-set timeout occurs or until the client requests that it is closed. In practice, a session typically would last for a working day or the duration of a field visit.

At any time, including the request to initiate a new session, the mobile device may send contextual information as part of the `<context>` message (hence the name). This might include the user's identity, location, interests and the spatial extent of required information. For example:

```
<context session=123>
  <person role="user">
    <firstname>Nick</firstname>
    <lastname>Ryan</lastname>
  </person>
  <location type="point">
    <proj type="UTM">
      <zone>33</zone>
      <datum>Euro 1950 (mean)</datum>
    </proj>
    <point3d>
      <x>281993</x><y>4686790</y><z>205</z>
    </point3d>
  </location >
  <interests>
    fieldnote,contour,track
  </interests>
  <spatial type=”box”>
    <range>100</range>
  </spatial>
</context>
```

This data is cached by the server as session context and is used to select information to send to the mobile. In the above example, the user requests FieldNotes, contour lines and tracks within a bounding box centred on their current position and extending 100m to the north, south east and west. The server responds to such a message by extracting the specified information from the database and converting it to a serialised XML form before passing it to the mobile client.

Subsequently, the client may send further `<context>` messages with content that refines or updates the server's cached values. Thus, when the user moves to a new location, it is only necessary to send a message containing changed data, for example:

```
<context session=123>
  <location type="point">
    <point3d>
      <x>281905</x><y>4686722</y><z>202</z>
    </point3d>
  </location >
</context>
```

Other message types currently defined include

- `<spatialObject>` – encapsulates spatial data sent to the client, such as the contour and track vectors requested above.
- `<fieldnote>` – encapsulates both new or updated FieldNotes sent to the server, and those matching the user's required context sent to the client.
- `<query>` – the client sends either a predefined, or an SQL query to the database server.
- `<response>` – encapsulates the server's response to a query.

This protocol is in an almost continuous state of flux as experiments with the system continue. We wish to avoid proliferation of specialised messages, and recognise that the <query> and <response> pair is only an interim solution to general *ad hoc* queries. In this regard, we are following developments such as the XML-QL proposal with interest [19].

The combination of database, web server and Java servlets allows considerable flexibility in the scale of system employed in the field. In the simplest scenario without wireless communications, handheld devices may be primed with a collection of FieldNotes and background map data before leaving for the field. Any new or modified notes are returned to the database on return. This requires only a wired network connection between the server and the handheld. An individual fieldworker could achieve more regular updating via a dial-up Internet connection using either GSM or PSTN, but this would be an expensive option for a team.

A more robust and secure approach can be achieved by taking a full or partial copy of the database into the field on a laptop computer. Regular synchronisation between handhelds and database, typically during lunch breaks and at the end of a day's work, provides both security for recorded data and makes the data available to all members of the team. In the next stage of development, we intend to experiment with radio-modems to provide permanent, if necessarily unreliable, communications between handhelds and this 'field server'. The radio-modems will also be used to provide live differential correction data for GPS receivers from a fixed base station.

In the current system, data collected on the handhelds is buffered until a connection to the database is available, at which point all updates are transferred. This works satisfactorily for temporary wired connections, but will be less suitable for connections using radio links. For this, we anticipate some extensions to the communications protocol to ensure that data is not lost across breaks in communication.

A further development will enable the field server to communicate with the main server back at the team's base using the same protocol as that employed between handheld and field server. This will permit periodic mirroring of new field data on the main server back at base when a communication link, typically a dial-up Internet connection, is available.

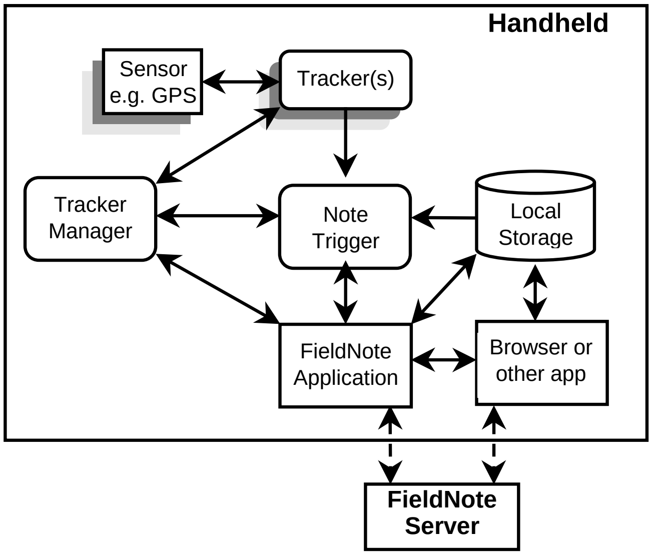


Figure 2: Architecture of the FieldNote handheld system.

4.3. FieldNote Handheld

We take the view that it is inappropriate to attempt to replicate desktop software designed for use with large displays on small handheld devices (see also [10]). Thus, rather than attempting to reproduce GIS functionality, we have tailored the software for these machines specifically to the needs of the fieldworker outlined in our second aim.

Several prototype handheld systems have been developed. Previously, these have been implemented on either the 3Com PalmPilot or the Apple Newton MessagePad. Here, we concentrate on the latter. However, now that this device is no longer in production, subsequent development now concentrates on the PalmPilot and the family of handheld devices based on WindowsCE.

The "Stick-e" suite runs on a PalmPilot and includes programs for defining note templates, recording notes and displaying a simple map of notes around the user's current location [12][13]. These programs are not fully integrated with the FieldNote system described here, but have been used extensively for developing prototype user interface mechanisms and has been tested in two seasons of animal behaviour studies in Kenya.

The FieldNote system developed on the Newton uses available commercial and shareware HTML editors and browsers for producing note templates and for writing and editing notes. HTML forms provide the main input mechanism for structured data. The submission mechanism stores the data locally on the Newton, and marks it for transfer to the database when a connection is available. Similarly, forms, notes and spatial data may be retrieved from the server as described above.

The first generation Newton prototype provided a basic map display with features such as coordinate conversion routines so that locations could be displayed using latitude and longitude, Universal Transverse Mercator (UTM) or the UK Ordnance Survey National Grid. Since then we have moved towards a more open system designed to support a wider variety of external sensors and context aware applications. The architecture of this second prototype is based around a set of communicating modules providing context aware services (Figure 2). The main components of this architecture are:

1. **Tracker modules:** these are used to monitor sensor devices. Typically, these sensors are connected to the handheld computer via a serial, infra-red or PC card interface, and a separate tracker module is required for each type of external device. More than one tracker module may be active at any time, monitoring different aspects of the user's environment. The tracker modules provide a common programming interface to contextual information across a wide range of sensors. Each module normally runs as a background task providing services to the NoteTrigger module and FieldNote applications.
   Several classes of tracker have been defined, including:

   - **LocationTracker**: used with a GPS receiver or other locating device.

   - **AttitudeTracker**: used with devices such as electronic compasses, inclinometers and range-finders to provide the user's or computer's azimuth and elevation angles and ranges to distant objects.

   - **EnvironmentTracker**: used to monitor a range of sensors providing information about current conditions in the immediate vicinity of the computer. Typical sensors might measure variables such as temperature, humidity, wind speed, etc.

- **PseudoTracker**: used to track logical contexts. Here, the user might register interest in particular subjects or notes authored by a particular person.

2. **TrackerManager:** used to organise a number of trackers on a single machine. The manager is used to select the default tracker module for each registered class of tracker (location, attitude, etc.).

3. **NoteTrigger:** this provides an optional context-aware triggering capability. It enables the user to define search criteria as a set of contexts of interest. It then runs in the background attempting to match notes with the current context. When a match is found, the note can be displayed, edited or executed using an appropriate application. The NoteTrigger module may monitor internal 'sensors', such as the system clock, external sensors via one or more tracker modules, or 'pretend' contexts using internally maintained lists.

Although tracker modules have been defined for a wide variety of sensor types, all of our trials to date have employed one of several GPS Location trackers to provide continuous awareness of location. This serves to locate each item of recorded data as well as to trigger the display of previously recorded data around the fieldworkers current location. The simplest of these GPS trackers decodes receiver output in the form of NMEA messages. However, a limitation of the standard NMEA messages is that they only support output of computed positions from the GPS receiver. If the receiver can be supplied with live differential corrections by a radio modem or other receiver, these computed positions may be sufficiently accurate for the more demanding applications. However, sub-metre accuracy can be achieved by post-processing the raw data from the GPS receiver in combination with data from a second fixed receiver. Simple handheld GPS receivers rarely provide the necessary raw data and for this we have turned to OEM receiver cards such as Trimble Navigation's 'Lassen SK8' [17] and the Canadian Marconi Company's 'Allstar' [5]. Each uses a different tracker module to decode the proprietary binary output formats of these receivers so that both an approximate position and the data required for more exact location can be recorded in the field.

We describe the programs that use tracker services as **FieldNote** applications. Typically these are purpose-written programs such as the mapping and note creation module, **FieldMap**, described below. However, the interpreted NewtonScript environment and semi-open nature of much of the built-in software of this machine have also enabled us to provide interesting extensions to these built-in programs.

In common with many PDAs, the Newton offers simple 'notepad' and 'diary' programs. In both cases it is possible to extend the user interface of these programs by adding buttons or menu items, and this has been widely used as a means of adding functionality. For example, several available web browsers add an 'Open in Browser' option to the Notepad, thus enabling users to edit their HTML source with this built-in editor and switch to the browser to view the results. In our case, we have added options to insert data from our Tracker modules into notes or diary entries. By clicking on a 'Location' button, the user can automatically insert their current coordinates at the position indicated by the normal text cursor.

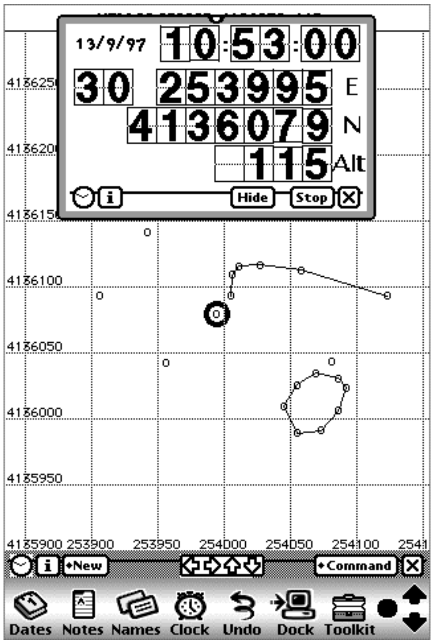
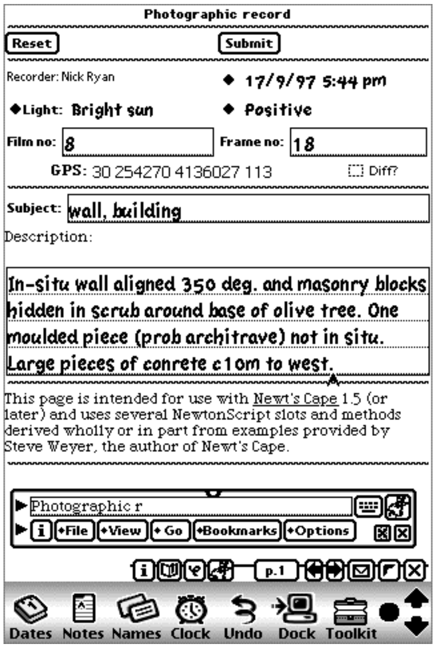

13/9/97 10:53:00
30 253995 E
4136079 N
115 Alt

Figure 3: The Newton FieldMap application with a GPS tracker display in a popup window. The map shows the current position (large dark circle) and that of mapped points, linear and polygonal features. Clicking on any of the small circular symbols gives access to the FieldNote form associated with the map object.


Photographic record

Recorder: Nick Ryan ◆ 17/9/97 5:44 pm

◆Light: Bright sun ◆ Positive

Film no: 8 Frame no: 18

GPS: 30 254270 4136027 113 □ Diff?

Subject: wall, building

Description:

In-situ wall aligned 350 deg. and masonry blocks hidden in scrub around base of olive tree. One moulded piece (prob architrave) not in situ. Large pieces of conrete c10m to west.

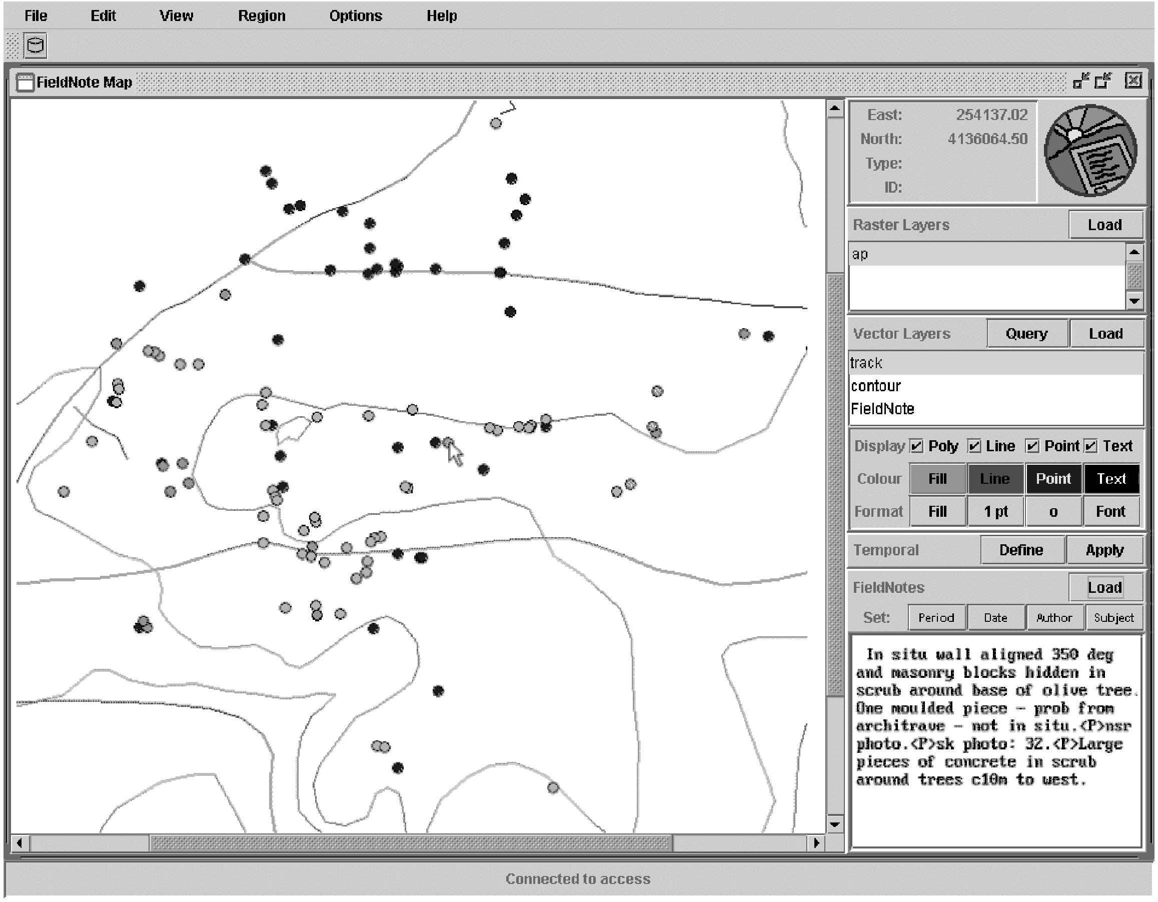
This page is intended for use with Newt's Cape 1.5 (or later) and uses several NewtonScript slots and methods derived wholly or in part from examples provided by Steve Weyer, the author of Newt's Cape.

Figure 4: An HTML form for recording photographs displayed using the Newt's Cape browser [21]. The recorder's name, date, time and GPS-derived location are automatically inserted into the form.

The **FieldMap** application (Figure 3) supports basic field mapping activities with text notes or form data attached to points, lines or polygons. As in a conventional GIS or CAD system, these objects can be associated with one or more display layers. Typically, these layers would correspond to a set of subject keywords in the user's selected logical context. FieldMap can be configured either to display all notes stored on the machine, or to use the NoteTrigger module to supply it with notes matching the user's context requirements.

To create a note, the user selects from a list of available HTML forms. The 'contextual' fields of the form, such as location, time and user identity, are automatically filled in with data from the trackers, leaving the user to enter additional information and press the submit button to register the note (Figure 4).

To retrieve a note, the user taps on the associated map symbol. A popup menu offers the choice of editing the note or viewing its content. Editing uses the same form as that used for creation. In the current implementation, users may edit any note but we intend to introduce some form of version control that will enable the history of field observations to be recorded. For viewing, the system generates a simple textual HTML page containing the data.



East: 254137.02
North: 4136064.50
Type:
ID:

ap

track
contour
FieldNote

In situ wall aligned 350 deg and masonry blocks hidden in scrub around base of olive tree. One moulded piece - prob from architrave - not in situ.<P>nsr photo.<P>sk photo: 32.<P>Large pieces of concrete in scrub around trees c10m to west.

Figure 5: . The Java prototype map display, a part of the FieldNote Desktop system, showing a set of notes collected at the archaeological site of *el Gandul*, Andalucia, Spain, in September 1997.

4.4. FieldNote Desktop

The desktop component comprises a suite of closely related modules that together provide

1. a map-based interface used for displaying and manipulating spatial representations of FieldNotes and related spatial data,

2. a simple forms-based interface for entering, examining and editing FieldNotes,

3. GPS receiver monitoring and recording to provide differential correction data from a fixed base station, and

4. a post-processing facility for differential correction of raw GPS observations collected in the field.

The first prototype desktop program was a relatively simple application written mostly in Tcl/Tk and developed from a similar system that had been written for an earlier project [2]. The second prototype is now nearing completion and has been written entirely in Java. It will, therefore, run on any suitable hardware platform and can be configured to work with a local database stored on the same machine, or a remote database via an Internet connection.

Only the map interface is described here (Figure 5). The remaining modules are all relatively straightforward and employ direct JDBC connections to the database (local or remote). The map interface communicates with the database via the web server and the same Java servlets as those used to service handheld applications. The servlets act as a middleware layer enabling the differences between object-relational and conventional database implementations to be hidden from the application.

The program functions in a similar way to many GIS display modules in that it allows different collections of geometric information to be retrieved and overlaid to form composite maps. A limited number of map manipulation and analytic functions are also provided. All data manipulation is performed by either the object-relational database server or by the Java servlets that access the conventional relational system. Because it only needs to deal with serialised data streams from the database, this ‘front-end’ program is little more than a user interface. The simplicity of the program lies in the fact that it only performs two basic tasks, query composition and map display. Queries are dispatched to the server and results are returned as text, images or vector data in a form that can be readily displayed by the program.

The controls on the right hand side of the window provide access to layers and allow the user to specify colours, line styles, temporal constraints, etc. If required, a composite result can be saved as a new map layer object in the database. Objects to be displayed are selected by composing and executing SQL queries that can be entered in a pop-up window. However, the interface also incorporates a number of aids to query composition so that all or part of a query can be generated using menu options or other interface elements.

The map display area shows FieldNote locations for a set of notes collected on the site of an Iberian and Roman period town at el Gandul, Andalucia, in September 1997. These are superimposed on contour and roads/tracks layers. Each FieldNote is represented by a small circle, the colour of which reflects a subject keyword in the note. In this case, the colours represent different types of building materials and so provide an indication of the varying types of structures across the site. The cursor, an arrow visible near the centre of the map, points at one of the note symbols, and the body text from this note is displayed in the text area at the lower right corner of the window.

5. Conclusions

We have used commercially available handheld computers, GPS receivers, databases and web servers to implement FieldNote, an information system for fieldworkers. The system has been developed over a period of two years including field trials in Kenya, Italy and Spain where it has been successfully used by environmental scientists and archaeologists. These trials have demonstrated that it is possible to collect more data more quickly than would have been possible using conventional paper-based methods.

The system aims to integrate many aspects of the production of a research GIS, and in this it has proved successful, with major reductions in the time spent on tedious data transcription and conversion tasks.

Several intended future developments discussed above are in hand. In particular, we intend to carry out further trials in the coming year that will concentrate on the use of *ad hoc* networking in the field in order to maximise the real-time decision support capabilities of the system. We also intend to further develop the Minimal Attention User Interface concept, which has proved particularly successful in animal behaviour studies.

Acknowledgements

We would like to thank Simon Keay, David Wheatley Sarah Poppy and Graeme Earl of the Department of Archaeology at Southampton University, Janet Bagg of the Department of Anthropology at the University of Kent, and Kathy Pinkney and Alan Birkett of the Department of Biology at Manchester Metropolitan University, all of whom have made very valuable contributions through their willingness to allow us to turn their field work into trials of the MCFE system.

Several of our colleagues at the University of Kent have contributed to the ideas described here. Particularly noteworthy is Peter Brown whose *sticke-note* concept provided one of the original inspirations for the project.

Mobile Computing in a Fieldwork Environment (MCFE) was a two-year project funded by the UK Joint Information Systems Committee (JISC) through the JISC Technology Application Program (JTAP), grant number JTAP-3/217.

References

[1] Apache (1999) *The Apache HTTP Server Project*, http://www.apache.org/

[2] Bagg, J. & Ryan, N.S. (1997). ‘Modelling historical change in southern Corsica: temporal GIS development using an extensible database system’, in Z. Kemp (ed.), *Advances in GIS 4*, Taylor & Francis, London, 42–55.

[3] Brown, P.J. (1996). ‘The Stick-e document: a framework for creating context-aware applications’. *Electronic Publishing: Origination, Dissemination and Design*, 8(2-3): 259-272.

[4] Brown, P.J. (1998). ‘Triggering information by context’. *Personal Technologies*, 2(1): 1-9.

[5] CMC (1998). *ALLSTAR 12-Channel GPS Receiver OEM Module*. http://www.marconi.ca/Docs/CMC/Components/cmt1200e.html

[6] Dublin Core (1999) *The Dublin Core: A Simple Content Description Model for Electronic Resources*, http://purl.oclc.org/dc/

[7] Fitzmaurice, G.W. (1993). ‘Situated information spaces and spatially aware palmtop computers’. *Communications of the ACM*, 36(7): 39-49.

[8] Gessler, S., & Kotulla, A. (1995). ‘PDAs as mobile WWW browsers’. *Computer Networks and ISDN Systems*, 28(1-2): 53-59.

[9] Harter, A., & Hopper, A. (1994). ‘A distributed location system for the active office’. IEEE Network, 8(1): 62-70.

[10] Kristoffersen, S., & Ljungberg, F. (1998), Your mobile computer computer is not a stationary computer, in Gellersen, H-W. (ed.) *Handheld CSCW, Workshop at the CSCW'98 on Handheld CSCW, Seattle, 14 November, 1998*, Technical Report, Dept of Computer Science, University of Karlsrühe, Germany.

[11] MCFE (1999). *Mobile and context-aware computing research at UKC.* http://www.cs.ukc.ac.uk/research/infosys/mobicomp/

[12] Pascoe, J., Morse, D., & Ryan, N. (1998a). 'Developing personal technology for the field'. *Personal Technologies*, 2(1), 28–36.

[13] Pascoe, J., Ryan, N. and Morse, D. (1998b). 'Human-Computer-Giraffe Interaction: HCI in the Field', in C. Johnson (ed.) *Proceedings of the First Workshop on Human Computer Interaction with Mobile Devices*, GIST Technical Report G98-1, Department of Computing Science, University of Glasgow.

[14] Ryan, N., Pascoe, J., & Morse, D.R. (1997). 'Enhanced reality fieldwork: the context-aware archaeologist assistant', in L. Dingwall, S. Exon, V. Gaffney, S. Laflin & M. van Leusen (eds.) *Archaeology in the age of the Internet: Computer Applications & Quantitative Methods in Archaeology 1997*, BAR International Series S750, Oxford, February 1999. Archaeopress, 269–274.

[15] Ryan, N., Pascoe, J., & Morse, D.R. (1999). 'FieldNote: extending a GIS into the field'. in J.A. Barceló, I. Briz & A. Vila (eds.) *New Techniques for Old Times: Computer Applications in Archaeology 1998*, (Ed. by), BAR International Series, Oxford, March 1999. Archaeopress.

[16] Schilit, B.N., Adams, N.I., and Want, R. (1994). 'Context-aware computing applications'. *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pp 85-90. IEEE Computer Society, Santa Cruz, CA.

[17] Trimble (1998), *Trimble products - OEM: Lassen-SK8*. http://www.trimble.com/cgi/omprod.cgi/pd_om006.htm

[18] W3C (1998a) *Extensible Markup Language (XML) 1.0*, http://www.w3.org/TR/1998/REC-xml-19980210.html

[19] W3C (1998b) *XML-QL: A Query Language for XML*, http://www.w3.org/TR/1998/NOTE-xml-ql-19980819.html

[20] Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R., & Weiser, M. (1995). 'An overview of the ParcTab ubiquitous computing experiment'. *IEEE Personal Communications*, 2(6): 28-43.

[21] Weyer, S. (1997), Newt's Cape web browser: http://www.netaxs.com/~weyer/newton/releases.html