

2005

Realtime display of landslide monitoring data

Russ Pennell

University of Wollongong, russ@uow.edu.au

D. Ruberu

University of Wollongong, dhammika@uow.edu.au

P. Flentje

University of Wollongong, pflentje@uow.edu.au

Publication Details

This paper was originally published as Pennell, R, Ruberu, D & Flentje, P, Realtime display of landslide monitoring data, [AusWeb05 - The 11th Australasian World Wide Web Conference](#), 5 July 2005. In A. Treloar & A. Ellis (Eds.), *Australian World Wide Web Conference* (pp. 1-9). Australia: AusWeb 05 Online.

Realtime display of landslide monitoring data

Abstract

In areas of high landslide risk, dangerous situations can develop rapidly. The system described here provides near real-time landslide information via the web to researchers, emergency personnel and others assisting them to assess developing risks. Remote field stations collect data continuously and download this to a central site at varying intervals via mobile phone. Processing and display software written using the ASP.NET framework stores the data in directly-graphable form and displays graphs in response to web requests. Design challenges included the changing nature of the instruments in the field, resolved by the use of user-editable configuration files that allowed for instrument changes at short notice.

Keywords

landslide, ASP.NET, remote, monitoring, configuration, REALbasic

Disciplines

Arts and Humanities | Social and Behavioral Sciences

Publication Details

This paper was originally published as Pennell, R, Ruberu, D & Flentje, P, Realtime display of landslide monitoring data, [AusWeb05](#) - The 11th Australasian World Wide Web Conference, 5 July 2005. In A. Treloar & A. Ellis (Eds.), Australian World Wide Web Conference (pp. 1-9). Australia: AusWeb 05 Online.

Realttime display of landslide monitoring data

Russ Pennell^[1], Coordinator, Learning Design Unit, CEDIR^[2], University of Wollongong^[3], NSW, 2522. russ@uow.edu.au
Dhammika Ruberu^[4], Technical Production Manager, Flexible Learning Services, CEDIR^[2], University of Wollongong^[3], NSW, 2522. dhammika@uow.edu.au
Dr Phil Flentje, Research Fellow^[5], Faculty of Engineering^[6], University of Wollongong^[3], NSW, 2522. pflentje@uow.edu.au

Abstract

In areas of high landslide risk, dangerous situations can develop rapidly. The system described here provides near real-time landslide information via the web to researchers, emergency personnel and others assisting them to assess developing risks. Remote field stations collect data continuously and download this to a central site at varying intervals via mobile phone. Processing and display software written using the ASP.NET framework stores the data in directly-graphable form and displays graphs in response to web requests. Design challenges included the changing nature of the instruments in the field, resolved by the use of user-editable configuration files that allowed for instrument changes at short notice.

Introduction

This paper concerns the design and development of a system to provide landslide information via the web to researchers, emergency personnel and others assisting them to assess risk from landslides. Design challenges included the changing nature of the instruments in the field, resolved by the use of user-editable configuration files that allowed for instrument changes at short notice.

Context

The monitoring of landslides is a particular need in the Wollongong/Illawarra area of New South Wales in Australia. The city and suburbs are located on a narrow coastal plain and foothills, between the sea and a steep escarpment rising between 300 and 500 metres (Figure 1). 570 landslide sites have been identified in the area, many of them with likely impact on residences, railway lines or major roads.

Movement in these landslides is usually triggered by prolonged heavy rainfall rather than earthquake vibrations. Wollongong experiences frequent heavy rainfalls, suffering flooding and landslides with loss of life most recently in 1998. Annual rainfall at these sites varies from 1200 to 1800 mm.

Consequently a landslide research project at the University of Wollongong has been supported over the last 12 years by the Australian Research Council and several industry partners including Wollongong City Council, the Rail Corporation, Geoscience Australia and the Roads and Traffic Authority.

The research purpose of such monitoring is to determine whether landslide events can be related to measurable precursors, including cumulative rainfall at the specific site. While Pedrozzi^[1] has recently suggested that the regional prediction of triggering of landslides is not possible using rainfall intensity/frequency methods in an area such as canton Ticino in Switzerland, a regional landslide triggering rainfall threshold (intensity/frequency) curve may be relevant for the Wollongong/Illawarra area. In fact, a preliminary threshold has already been proposed for this area^{[2],[3]}.

Monitoring stations

Several monitoring stations have been established. These consist of 70 mm boreholes in which instruments are located. Readings from the instruments are stored onsite and periodically transmitted by digital cellular mobile phones to a personal computer located (before this project) in the researcher's office.

The instruments installed in the boreholes include In-Place-Inclinometers (IPI, producing voltage levels representing displacement) and vibrating wire piezometers (vwp, producing numeric frequency values representing water pore pressure). Most generally, three IPI's and two vwp's are installed in each borehole at depths spanning the location of the known slip plane. Rainfall Pluviometers have also been installed at all the field stations to record rainfall as it occurs (0.2mm or 0.5mm bucket tips).

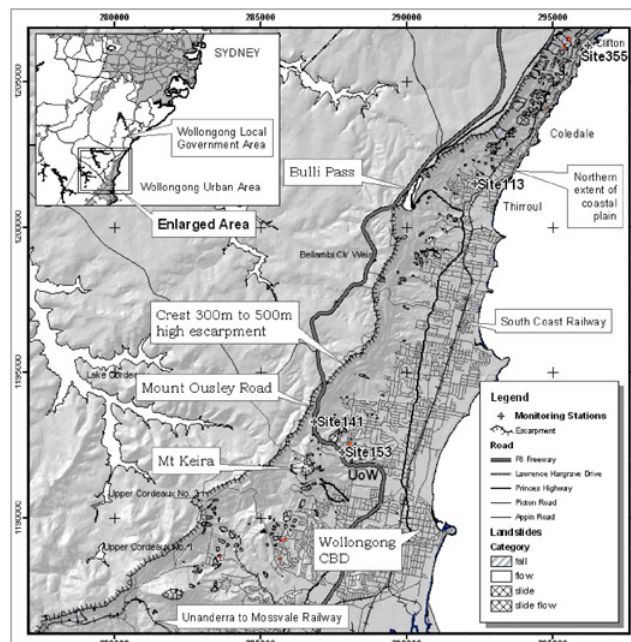


Figure 1 Location plan of Wollongong showing monitoring stations

Data acquisition and management

The data loggers at the boreholes record data hourly, and in low rainfall/dry times download data to the office weekly. When rainfall intensity increases the frequency of data download is increased to daily and even up to 4 hourly (at which time the data logger also starts recording data at 5 minute intervals). These varied data logger responses are triggered by rainfall intensity thresholds, for trigger intervals spanning 6 hours up to 120 days. The data collection and transmission to the researcher's office was thus completely automated.

In addition to this automated data collection, an operator could contact the field stations at any time from the office PC and download data. Software on the PC can perform appropriate calculations and display current or historical data onscreen. Hence the monitoring stations provided real-time information regarding the onset of landslide movement. However this data and its graphical representation was not available in a timely fashion to those who would be concerned with using the information in an emergency, nor to the researcher's geotechnical colleagues around Australia and overseas.

The web project

Staff of the Centre for Educational Development and Interactive Resources (CEDIR) were approached by the researcher to set up a website for this remote sensing data. Completely misunderstanding what was required, we arrived at the first project meeting with a graphic artist and a web programmer. In fact what were asked to do is shown in figure 2.

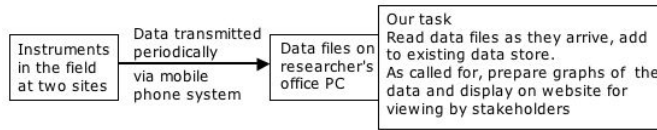


Figure 2 The task

The instruments in the field send their data as a string of comma-delimited values showing voltage levels, oscillator frequencies or rainfall increments, etc as shown for the simplest site below.

103,2003,176,1300,196.27,-643.05,19.63,3125.1,22.36,14.34,13.11
 translated as
 StringID,year,dayofyear,hhhh,IPIR(Downslope),IPIB(Transverse),IPITemp,vwpFreq,vwpTemp,FieldStationTemp,BatteryVoltage

Figure 3 Data string from the field

The conversion and display system existing before we became involved is shown in Figure 4.

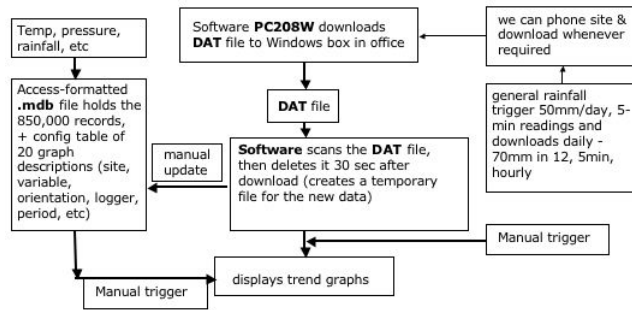


Figure 4 Existing system 2004

It can be seen from this diagram that an immense amount of data already existed in the system. What is not apparent is that considerable calculation is required to convert the data arriving from the field into measurements of the relevant variables to be graphed. In the extreme case (In Place Inclinometers used to measure earth movement) there are 26 constants involved in the conversion for each instrument. In the system shown here, the graphs produced could only be viewed on the PC in the researcher's office. In addition, the graphs had to be selected by a manual process from a set that had been entered earlier by the researcher.

Development

One of our programmers constructed a prototype for a single IPI using ASP.NET and verified that, starting from a data file on the PC as it arrived from the field, the prototype was able to produce the same calculated output as the proprietary software. A simple graph display was also developed. Having proven that the task was achievable, we began to specify the project in more detail, and complications began to multiply.

- Seventeen graphs were requested, mostly line graphs but one histogram and one of polar data
- Four data sites now exist, and we need to allow for future sites to be added to the system
- At three of the existing sites, there were three instruments at varying depths measuring earth movement and two instruments at different depth measuring pore pressure. Each instrument had its own set of conversion constants.

Two modes of graphing were required:

- 1) selectable from the set of 17 graphs, for a specific site and historical time period
- 2) an emergency graph produced by single click, showing graphs of five important variables for the two weeks previous to the current time.

Initially the plan was to substitute entirely for the proprietary software by converting all incoming readings to real values and storing them in MS-Access database format (as they were currently being stored) in the existing database structure for later manipulation or graphing as required.

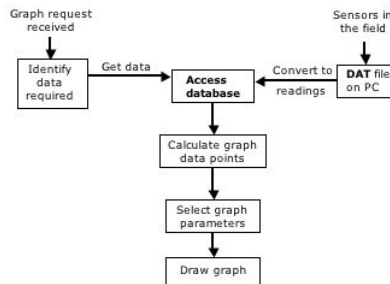


Figure 5 Structure of early design

In periods of heavy rain, records arrive at 5-minute intervals, sometimes leading to hundreds of records being stored in a day for each site. Some of the graphs (eg 120 day rolling cumulative rainfall) require manipulation of large amounts of data. Working with the prototype software, it soon became apparent that selecting and processing the data whenever graphs were requested would lead to unacceptable delays in graph delivery to the viewer. To reduce this delay, it was decided to store the data in its final graph-value form, leading to 4 data tables holding only hourly or daily data values for the 17 graphs. Then any graph requested could be delivered very quickly using just the sequential values in one table.

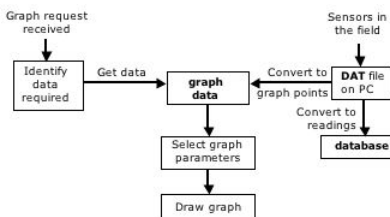


Figure 6 Structure of later design

From this perspective, the project then seemed naturally to be divisible into two parts:

- routines converting incoming data to real graph values
- graphing routines which would take data from the tables and display it as web pages.

As our only capable ASP.NET programmer was not available further for the project, we needed to have another programmer train in the system as he was developing the software. Programming of the web-based data display of the 17 graphs with variable data periods and scales proved more difficult than expected, so a temporary solution was required that would enable the client to partially satisfy the demands of external stakeholders. We had already separated the graphing system construction from the data-processing task, and now further decomposed the data processing task. We chose to continue to pursue the task of processing live data as it came from the field in ASP.net but simultaneously to provide a solution which would make the existing dataset more quickly available for graphing.

We wrote routines in REALbasic to carry out the conversion of data into hour-based graphable form and verified that the calculated values matched earlier records, as shown in figure 7 for IPI12295A, 12294A and 12297A.

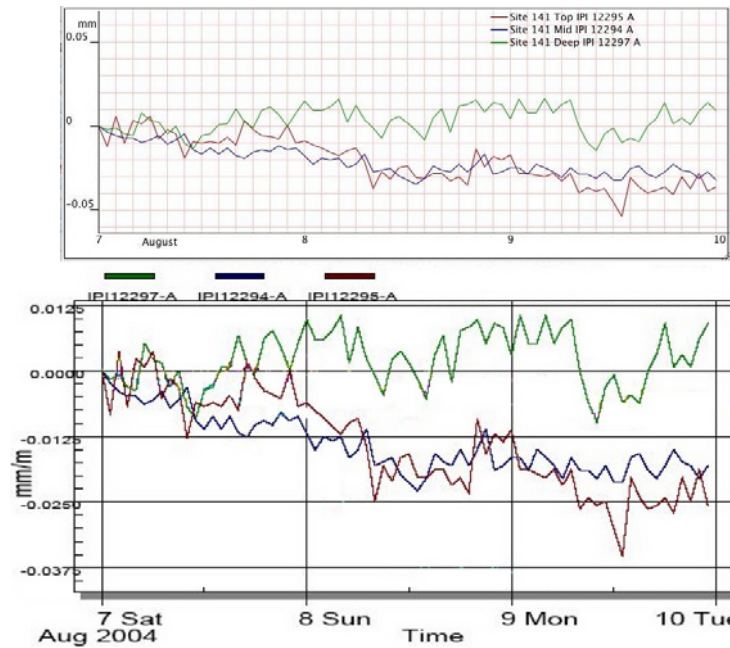


Figure 7 Existing graphs (bottom) vs REALbasic calculations and display

Display routines were written and historical data for 5 variables became accessible on the web. This historical data was updated manually every month while more robust display routines were written and the live data interface developed.

Processing the live data

The live-data processing module had to automatically process incoming data on a continuous basis, with data downloads arriving at unpredictable times from four or more sites. The files arrive in a designated directory and are named with the site number. Calculations need to be performed on the data carried, the graph database updated, the file moved to another directory for later access by the researcher, and copied as a backup to another directory.

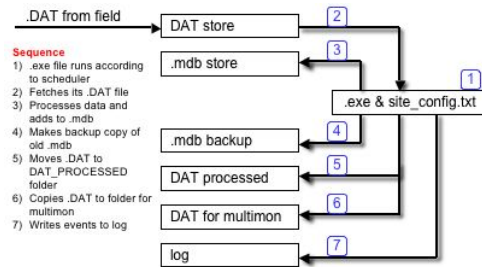


Figure 8 Live file processing sequence

We had two ways of achieving this: namely creating the data processing application with the FileSystemWatcher class and letting it run continuously or building the application without using the FileSystemWatcher class and initiating it as a Windows scheduled task.

The FileSystemWatcher class enables applications to receive notification when a change occurs to a specified directory or a file. Detailed information on the file system watcher class is available from the MSDN Library[[HREF7](#)]. This is an elegant solution when we need to process a file based asynchronous data stream, but this approach has the disadvantage that the application needs to be set up and run as a service on the server if it is to process data continuously. The need to be continually logged into the server is poor security practice.

For simplicity and flexibility we chose to implement the data processing application without using a FileSystemWatcher. At the initial testing stage we had a unique data processing application per site. Later we were able to develop a common application by successfully separating the data processing logic and site specific data, thus eliminating the need to maintain separate different applications for each site.

The application was developed directly in the .NET web environment and later compiled to be a standalone application. This approach allowed faster development; the programming of the web component could be done at any workstation and external clients and partners to see progress in real time by accessing a URL. The limitations of the web component approach are the inability to schedule running and the tendency to time out on long data runs.

Data separation was achieved by creating a unique configuration file per site. This is a plain text file consisting of data lines and comments. Each data line consists of a name value pair separated by a space. This file has all path names, database name, database connection string, and all site specific instrument information such as instrument ids, active and inactive instruments, instrument constants and information for error checking. When the application starts it parses this file and builds a StringDictionary. All the internal functions then reference this StringDictionary to obtain relevant data. Detailed information on the StringDictionary class is available from the MSDN Library[[HREF8](#)].

When it comes to updating the database it was possible to build the necessary SQL statements from the information in the configuration file, but we have used the OleDbDataAdapter to auto generate the commands using an OleDbCommandBuilder object thus utilising the power of ADO.NET [4]. It is not possible to fully explain this technique here but it is worth noting that for this approach, all the data tables must have a primary key and the select statement that was used to build the OleDbDataAdapter object must reference all the data columns that need updating. This second constraint can be easily met using a SQL query like "SELECT * FROM tableName WHERE nonExistentDataCondition". If we don't need to retrieve specific data, then by specifying a non existing data condition we still get all the relevant information without the overhead of actual data retrieval; this can be significant on a large table.

So we have a series of applications, one per site, each with a unique configuration file, with no interface, that are run periodically by the Windows scheduling system. The application checks for the existence of a DAT file specified in its own configuration file. If none is found then it terminates after creating a log entry. If there is a DAT file, then the file is

processed as shown in Figure 8. This structure enable us to add extra sites to the system without modifying existing code, reducing the chance of introducing errors into the existing system.

Log files are written for each initiation of the task, and for any actions taken or errors encountered. The log file gives us an indication of time taken to process data in each instance; one week of data from a multi-sensor site takes about 5 seconds to process to graphable form on a Pentium 3 system.

This project may have a long life and numerous structural extensions are likely. The modular design we have followed in building the data processing application caters for the future needs of the project without having to re-engineer the system components. With this structure it is possible to add a new field station to the system by creating a new database file, duplicating the application and the configuration file and changing the settings in the configuration file.

Web outputs

The website developed to display the landslide monitoring data is driven by ASP.Net code and is at [HREF9] (now password protected) which opens as shown in Figure 9. At present four monitoring sites are available and these can be selected from the menu on the left by clicking on the site locations on the index map.

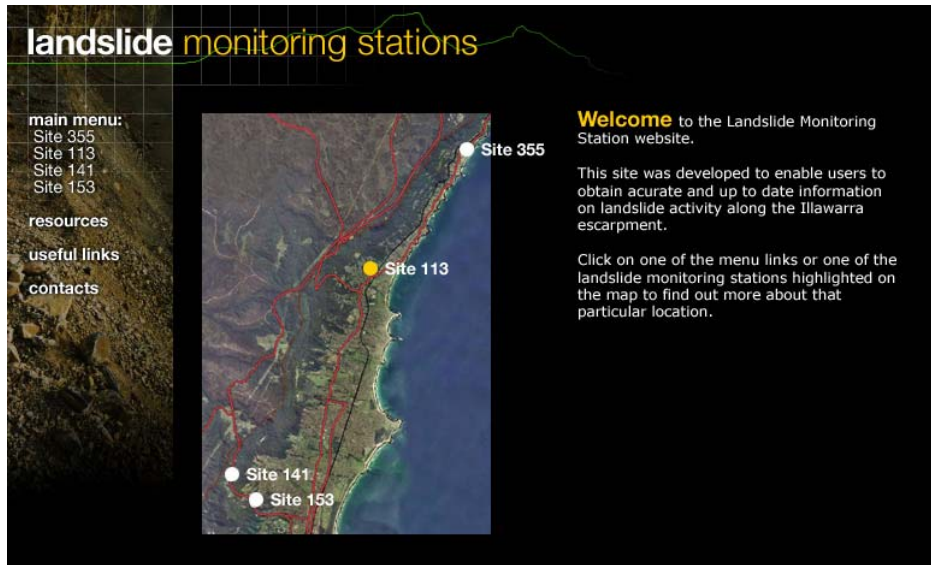


Fig 9 University of Wollongong real-time landslide monitoring website

The site specific pages open as shown in the upper part of Figure 10. The most recent 2 weeks of data is always available by selecting the 2 week overview button. This shows on one screen graphs of hourly IPI Total displacement, IPI rate, IPI azimuth, hourly rainfall and pore water pressure. The database of existing landslide measurements is also available for review for any 14 day period by selecting an end-date from the calendar and a data type from the range of seventeen available.

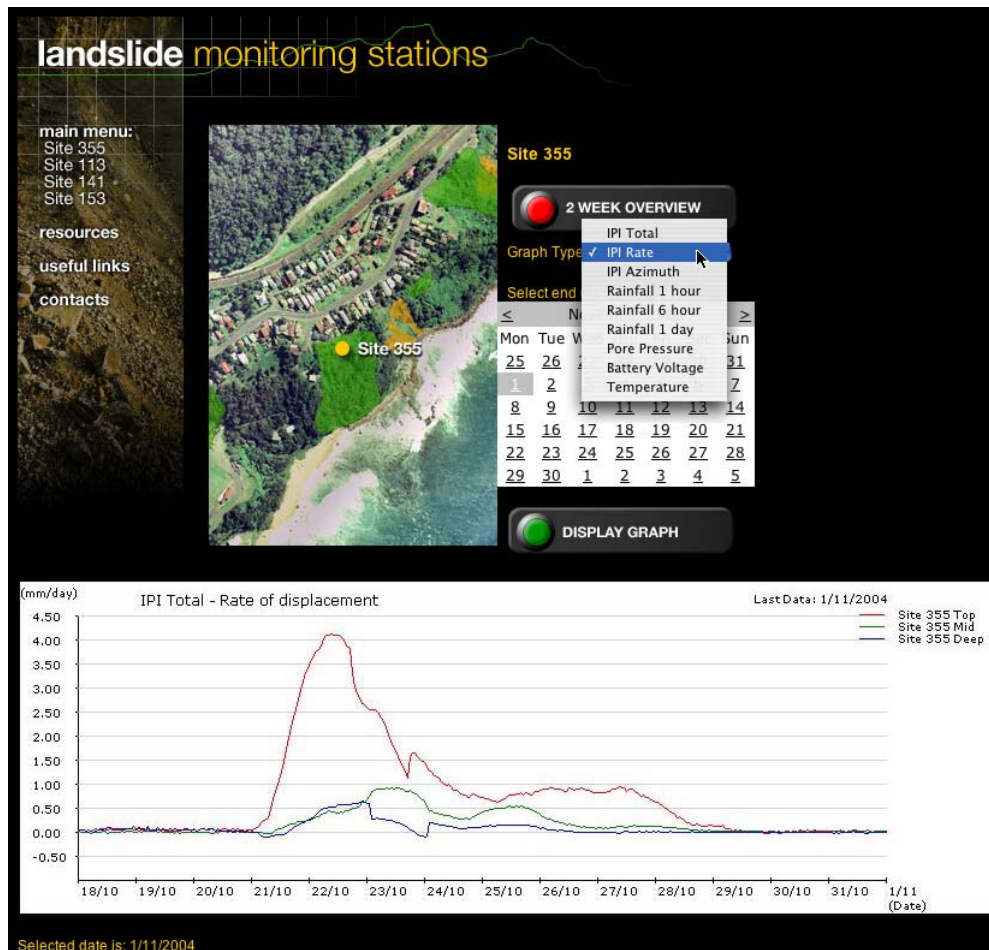


Figure 10 Landslide movement at three depths in October 2004

The display period will later be user-selectable up to 180 days. Additional sites are already in preparation, including unstable locations in the NSW Snowy Mountains such as at Thredbo, and an eventual network extending across Australia is likely. While password protection of the display system has been introduced as a precaution against misinterpretation of the data and consequent public misinformation, it is hoped that at least a limited dataset will remain available for public viewing.

Difficulties

The need for specific skills in the chosen programming environment slowed the development radically after its initial success. Later on in the process, events occurred which further slowed development.

- The existing data set proved to be partially corrupted and incomplete; after considerable efforts to locate and correct all the corruptions, we eventually had to abandon more than 12 months of records as unreliable and select a start date for recording after which we could be sure of data integrity.
- After a landslide at one site, two instruments were damaged and had to be removed. Random data from the field was still arriving, but had to be rejected by the conversion system.
- At one site a third pore pressure instrument was to be connected at some time in the near future, so provision for its data had to be made in the system, despite the lack of any data in transmissions from the field.
- As mentioned and shown above, the data arrives as comma-delimited strings of characters. At one stage the Canadian programmers for the firm supplying one of the instruments remotely changed the number of fields in the record without informing us, with predictable consequences as we applied conversion calculations to the wrong data fields.

Solutions

The solution that evolved from these and other challenges was to locate much of the detail of the system in plain-text configuration files, with a generic copy of the processing application for each site.

These configuration files can be easily edited by the operator to respond to a variety of system changes either directly or, at a later stage, through an interface application. Typical issues for the data handling system operator include

- dealing with erroneous data arising from eg sensor damage or removal
- excluding values for a chosen time period and a specific instrument
- selecting what no-show value will be substituted in the database for these periods
- recording the times when a datum shift has occurred (a move in the zero-reference point for displacement measurements) , and entering new reference values
- sensor changes for live data where the instrument parameters change.

Changes made in the field must be compensated for by changes in the configuration files, but these events are not concurrent, so the configuration editing must be cognisant of the time from which the processing changes are to operate.

Future development

One weakness of the current display is the use of a line plot for azimuth data. This is difficult to interpret, especially when small changes near 0 degrees flip over to 360 degrees. A solution being developed is the use of a Flash program to represent the data on a polar plot, with lengths of vectors also showing total movement in the chosen time period.

Work is also continuing on a Configuration File editor which will provide a contextual interface allowing the researcher to modify field station settings.

The graph-drawing routines developed for this case will be encapsulated into a class allowing further reuse.

Outcomes

The system developed has provided access for other researchers worldwide to original field station data in near-real time, rather than months or years later at a conference. By opening such an avenue for data sharing, it has encouraged further sharing in the field. For the organisations who provided funding to support the installations and the development of the web interface, the system has provided near-real time data to allow emergency response to developing landslide situations.

References

- [1] Pedrozzi, G. (2004). "Triggering of landslides in canton Ticino (Switzerland) and prediction by the rainfall intensity and duration method" in *Bulletin of Engineering Geology and the Environment*, Volume 63, Number 4, pp. 281–291.
- [2] Flentje, P. (1998). *Computer Based Landslide Hazard and Risk Assessment*. PhD Thesis, University of Wollongong, Australia.
- [3] Flentje, P. and Chowdhury, R. N. (2001). "Aspects of Risk Management for Rainfall - Triggered Landsliding" in *Proceedings of the Engineering and Development in Hazardous Terrain Symposium*. New Zealand Geotechnical Society Inc, University of Canterbury, Christchurch, New Zealand, The Institution of Professional Engineers New Zealand, August 24-25, pp 143-150.
- [4] Payne, C. (2003). *Teach yourself ASP.NET in 21 days* (2ed). SAMS Publishing, Indianapolis, pp 358-9.

Hypertext References

- HREF1 <http://www.uow.edu.au/cedir/about/staff/rpennell.html>
- HREF2 <http://www.uow.edu.au/cedir/>
- HREF3 <http://www.uow.edu.au/>
- HREF4 <http://www.uow.edu.au/cedir/about/staff/druberu.html>
- HREF5 <http://www.uow.edu.au/eng/staff/flentje.html>
- HREF6 <http://www.uow.edu.au/eng/>
- HREF7 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfSystemIOFileSystemWatcherClassTopic.asp>
- HREF8 <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfSystemCollectionsspecializedstringdictionaryclasstopic.asp>
- HREF9 <http://landres.uow.edu.au/ls/index.html>

Copyright

Russ Pennell, Dhammika Ruberu, Dr Phil Flentje, © 2005. The authors assign to Southern Cross University and other educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to Southern Cross University to publish this document in full on the World Wide Web and on CD-ROM and in printed form with the conference papers and for the document to be published on mirrors on the World Wide Web.