MONTSERRAT VOLCANO OBSERVATORY

GOVERNMENT OF MONTSERRAT

The role of the Software Engineer within the seismic monitoring programme

Glenn Thompson

MVO Open File Report 03/03

MONTSERRAT VOLCANO OBSERVATORY

GOVERNMENT OF MONTSERRAT

The role of the Software Engineer within the seismic monitoring programme

Glenn Thompson

MVO Open File Report 03/03

Montserrat Volcano Observatory
PO Box 292
Montserrat
West Indies

Tel: 1 664 491 5647
Fax: 1 664 491 2423
Email: mvomail@mvo.ms

* Affiliation inside cover

Affiliation of author

Glenn Thompson
Montserrat Volcano Observatory
Flemings
Montserrat
West Indies

Table of Contents

Executive Summary

The purpose of this report is to help define the role of the Software Engineer within the seismic monitoring programme, at the request of the MVO Director. This follows problems since my return from vacation in February 2003 where the Software Engineer usurped authority for much of the Seismologists' role and modified and removed crucial software and hardware without consent, undermining MVO's ability to monitor the volcano effectively.

What I am recommending here is a return to the working relationship and practices that were in place from October 2002 until January 2003. The Appendices illustrate the original terms of reference of the Software Engineer's position, as well as evidence of the practices that were in place.

1 INTRODUCTION

Software maintenance and development is crucial to the successful operation of the Montserrat Volcano Observatory, and the seismic monitoring in particular. While each Seismologist who has worked at observatory has brought some level of programming experience and a familiarity with common seismic software applications, and needed this, the main reason for hiring a Seismologist is to have an expert interpretion the seismic activity for the MVO Director and the authorities. It is therefore hugely beneficial to have an experienced Software Engineer at MVO to assist Team Seismic and be the main person to maintain and develop code.

Between October 2002 and January 2003 the Seismologist and Software Engineer had an excellent working relationship and much was accomplished, culminating in the successful move of the seismic monitoring to MVO Flemings. However on my return from vacation in February I found there had been a significant hardening of the Software Engineer's attitude, and within a day or two of my return he demanded that he take over all aspects of the seismic monitoring computer hardware and software systems, and that I “no longer touch the software or have ideas about it”. He also told me that things I felt were a priority, were not a priority. It is clear something odd happened during the few weeks I was away as we were good friends before, and that he resents working under any supervision. Since then he has been unwilling to discuss this issue with myself, and has meanwhile removed and modified critical hardware and software without my consent, compromising seismic data acquisition, alarm and archival systems. I have endured repeated, unprovoked outbursts of verbal abuse. This sort of behaviour is, unfortunately, hardly new at MVO and has been flagged up before and recognised by independent reviewers and BGS. However, it is my belief that the current situation is now worse than ever before in terms of its potential impact on our mission.

The MVO Director has asked for input from me on this and so I have tried to outline the role of the Software Engineer within the context of the seismic monitoring programme, to assist the Director in shaping terms of reference. I provide some background information on the seismic monitoring programme, and conclude with urgent work for the Software Engineer which has been in limbo for several weeks.

The Software Engineer should have no doubt of his crucial role with the seismic monitoring programme, and haven't selected his CV myself, and worked alongside him successfully for more than 3 months, I can attest to his excellent programming skills and knowledge of modern software technologies, and look forward to a resumption of our earlier successful working relationship. I take annual leave in the next few days and hope that the MVO Director will have the opportunity to read this report and restore my authority for the seismic monitoring before my departure.

2 DEVELOPMENT OF THE SEISMIC MONITORING PROGRAMME AFTER Y2K

Since January 2000, enormous strides forward have been made in terms of the seismic monitoring programme, particularly the seismic software. These are too many to cover in detail. Broadly speaking the main objectives were:

(1) **Stabilise data acquisition.** We were losing 16% data at the data acquisition stage. This was eventually achieved with the upgrade to new acquisition systems in March 2001. During the 14 months prior to this the VME often failed multiple times each day, and required up to 25 reboots to restart.

(2) **Develop a volcano alarm system.** The RSAM software was seizing up a lot, and didn't give any indication of event size. An event alarm system was developed (in 2001), and used a web server to send alarms by pager (faster and generally more reliable than modem-based paging) but could no longer be supported after Patch left (Aug 2002). An event and tremor alarm system was then developed, and has been considerably further developed by Joel. The preferred method of communication is now via SMS email to cellphones. The next development will be to use a magnitude scale consistent with that used for events.

(3) **Develop a remote monitoring capability.** Largely to reduce time in getting out warnings to Washington VAAC, the goal was to develop either dial-up or web-based remote seismic monitoring software. A dial-up capability first became possible in April 2001 with the QNX. The Remote monitoring website capability finally became a reality in mid-2002 (although there had been an internal one since November 2001) and has been much improved since. The content is provided through Matlab programs which run as scheduled tasks on EarthWorm3. This seismic monitoring webpage is crucial to rapid alarm response, making driving to the observatory obsolete.

(4) **Develop better analysis software.** At this stage we were basically only counting events. We had no decent software for looking at hypocentral patterns, or patterns based on event size, nor did we have an equivalent of RSAM for the digital network. The first tools were GMT-based, running on the Sun workstation, or Matlab-based, running on my laptop. When we eventually got a modern Windows PC (mid-2001?) and Matlab licence, a vast array of Matlab software began to be developed, including capability for looking at RSAM, tilt, Seisan waveform files, spectrograms, filtered digital helicorder, event count and event magnitude data. In recent months

there has been a move to better organize this software, before handing it off fully to the Software Engineer for future development/maintenance work.

(5) **Improve robustness of mission critical systems.** This involved developing auto-start capability for acquisition software, and diagnostic alarms for monitoring possible points of failure. Prior to these developments, it had been a common experience to arrive at MVO in the morning only to find the acquisition systems or the alarm system had fallen over at 6pm the previous evening, and had a major collapse occurred since, we'd have known nothing.

(6) **Develop better data archival software.** In early 2000 we were losing 50% of data by the data archival stage due to a combination of unreliable data acquisition, and poor data archival software. An interactive program called "archive2dat" was developed in 2001, and was made increasingly robust as possible pilot errors were identified and accounted for. Important features were it verified tarfiles, would only allow fixed periods of data on each tape, and tracked what data was on what tape. This program was then replaced in January 2003 by a program called "backup" and Team Seismic were instructed to use it without my consent. This new program is a backward step since it does not provide the same safety features as archive2dat such as (i) checking there is a tape in the drive, (ii) forwarding to the end of the tape to avoid overwriting data, (iii) checking for an end of tape condition and (iv) verifying tarfiles as they are written. I recommend that MVO go back to using archive2dat, or add the error checking this program incorporated into the backup program.

(7) **Merge the seismic networks.** This was first achieved with at the data analysis stage, but the goal is to move this upstream to the data acquisition stage (EarthWorm, see ADC card below) and ultimately the telemetry stage (seismic network upgrade proposal).

(8) **Assessment and acquisition of mission critical Ops Room spares.** Once the data acquisition systems had been replaced, there was a need to get the necessary spares to support these systems. This took several months, beginning on April 2001.

(9) **Assessment and acquisition of spares for the outstation networks.** Getting information about network spares was absurdly difficult as the Electronics Team procrastinated and tried various blocking tactics that the MVO Director found impenetrable. Eventually it took a visit by Dai

Stewart in March 2001 and his subsequent report in July 2001 to bring an end to what had been a 15 month long process to get a list of spares out of the Electronics Team.

3 CONTRIBUTIONS TO THE SEISMIC SOFTWARE

Numerous individuals have made valuable contributions to the seismic monitoring programme at MVO directly or indirectly. Every MVO Seismologist has spent some, even the bulk, of his/her time developing software. Dozens or even hundreds of programmers have contributed to packages such as Earthworm, Glowworm, Scream, Seislog, Seisan, GMT and Matlab that are used. Staff within BGS Seismology have generously provided programming support well beyond the call of duty, and without their help, the current acquisition systems would never have been installed.

David Silcott who was Software Engineer from 2000 to 2002 made several key contributions. He wrote modules to send alarms to pagers via the Cable and Wireless webserver, without which the MVO volcano and diagnostic alarm systems could never have been implemented. He wrote tools for analysing RSAM data. He also provided vital Linux administration, and was ever helpful with solving network problems.

Joel Maranhao brought helpful ideas to the table, such as having separate development and live systems. In the first few months we emailed back and forth a task list which was discussed frequently and modified as we learned more about a problem, solutions were found, or new needs identified [Appendix B]. He provided guidance on writing Perl modules according to industry best practice, and feeling this was not work he could complete alone, we set about working alongside each other as a team to make the necessary modifications. (Unfortunately, this work was only half complete by February 2003, and has effectively been abandoned since, leaving the seismic software in a vulnerable limbo state where half the software is written according to a different model than the other half). When new needs have arisen for a software tool, we have agreed that I write a software specification for Joel to implement [Appendix C], and given him the authority to code the solution in whatever way he felt fit, so long as it did not compromise existing seismic systems.

Most of the actual glue that holds the seismic monitoring systems together, or moves, archives and retrieves data are programs that I wrote. So are all the Matlab tools on which we rely. In all this amounts to around 200 programs. The seismic systems are complicated, but no more complicated than necessary. For example, it would not be appropriate to use a poorly known, highly specialised, or very expensive commercial programming language, because in the long run it might be difficult to maintain such software, especially due to staff turnover. Another consequence of staff turnover is the need for documentation and appropriate periods of overlap. At this time, I am the only person with a thorough understanding of our present seismic systems, and it is my hope that the Software Engineer will be willing to undergo further training. The observatory needs more than one person with a detailed understanding of the critical seismic monitoring systems.

4 SEISMIC SOFTWARE DEVELOPMENT NOW

Seismic software development is almost complete. That is to say, we have tools to do most anything we'd want, and its now largely a process of "cleaning up" the code, making sure it adheres to an overall design plan for the seismic software, and documenting the code. Unfortunately, this work has not moved forward in recent months, perhaps because the Software Engineer has committed himself to the MVO database project. It would still be desirable to tie off this work as soon as possible, so the seismic software is no longer in a state of limbo.

This database project is MVO's next big focus, now that the individual monitoring programmes are all on a firm footing. The aim is to move strongly towards integrated monitoring, and Joel has already made impressive progress. Integrating the seismic software with this database is likely to be the main requirement for further development of the seismic software.

Division of labour: While there will always be a need for the Seismologist to write programs to analyse a particular dataset in a particular manner, the long-term goal is very much that the Seismologist will be able to leave the bulk of the seismic software development and maintenance work to the Software Engineer. The plan has always been to work towards a situation whereby the Software Engineer feels comfortable to take on this responsibility for more and more of the seismic software, with the Seismologist taking more of a backseat role when it comes to programming. This is an important step towards giving the Seismologist a better opportunity to analyse data and perform monitoring-specific research (there was very little opportunity for this in the 2000-2002 period due to the overwhelming need for building the seismic monitoring infrastructure).

As the person responsible for the seismic monitoring programme, the Seismologist has to continue to guide and supervise the work of the Software Engineer, but nevertheless my approach has been to give the Software Engineer complete freedom to choose which programming languages and other technologies to use the software requirements as I see them, providing that seismic monitoring is not compromised. Creativity is encouraged as are new ideas. Indeed, emails show I have been highly supportive of new ideas by the Software Engineer in recent months.

5 DATA RECOVERY

There is an ongoing project to get continuous data from the digital seismic network (stored in 20-minute "rbuffer" files) online. This began in February 2003, but the 500 GB firewire hard drive onto which these data were being extracted, crashed in April 2003. Dave has a copy of Norton Utilities at his office in the Computer Unit, and plans to make this available, so repair of this hard drive can be attempted. Event data were also on this hard drive, including event data from the analog seismic network recorded in SUDS format and later converted to Seisan format.

Dave has taken delivery of a new centralized data server, which will have the capability to store up to 2 terabytes of data online. Clearly the objective should be to get all the seismic (and all other datasets presumably) onto this system, and backed up onto DLT media. The next step would be to load BGS Edinburgh's copy of the seismic data onto the same data server, thereby establishing a master dataset for the first time (there are significant gaps in MVO's seismic data that may exist in Edinburgh).

Having all these data online will (i) improve MVO's access to the data, (ii) allow derivative datasets (e.g. BSAM) to be constructed (iii) allow techniques like "Tanya trigger" to be applied to looking at episodes of cyclic hybrid swarms, and (iv) allow MVO to provide data to researchers much more efficiently than previously possible.

6 MERGING THE SEISMIC NETWORKS

In March I proposed the purchase of an analog-digital card which will enable data from the analog network to be acquired through EarthWorm. The total cost is less than US$2000, and will allow us to bring both networks in through EarthWorm, so effectively we would have one 10 station seismic network, instead of two completely independent low-station-number networks. This ADC was recommended by Andy Lockhart and others at VDAP/CVO and will give us most of the advantages of the seismic network upgrade (which could still be 18 months away) at about 1% of the cost:

(1) Lower event detection threshold
(2) Better classification
(3) Better hypocentral determination
(4) One fewer acquisition systems / computers to support (QNX/Seislog would be obsolete)
(5) Considerable simplification of the MVO seismic software (much of which is currently needed to try to combine data from both independent networks)

Another advantage is that it would provide us the software modifications required to make this work are identical to those required for the telemetry upgrade. So we could take care of this step now, rather than a riskier approach of modifying software and installing a new network simultaneously.

I see this purchase as top priority for seismic monitoring since in recent times we have been down to as few as 5 digital network stations, and 3 analog network stations. A minimum of 6 is needed for really effective triggering, and for hypocentral determination. By combining networks, we'd never had dropped below 6 stations.

7 THE SEISMIC UPGRADE PROPOSAL

In January 2003 DFID finally rejected the Reftek proposal which had resulted from the absurd tendering route they had pushed MVO down. Lars Ottemoeller (at that time, acting as cover Seismologist) was then asked to come up with a much cheaper proposal (this time hardware only, not a "turn-key" system including a maintenance contract). This proposal went to the MVO Board in March 2003, was modified following consultations between myself, Lars Ottemoeller, Brian Baptie and Jurgen Neuberg, and finally accepted by the MVO Board in late April 2003. We are now awaiting approval of funding (from GoM or DFID?), before ordering can begin.

The proposal is based on Guralp digitisers and FreeWave radios. However, following conversations between Jurgen Neuberg (our representative on the MVO Board) and Guralp, alternative radio equipment may be considered. There appears to be a consensus that Ethernet radios would be more suitable, and these should allow us to come in considerably under budget. Guralp is also interested in coming out himself to assist with the installation (good use for the "extra Seismologist" budget line), and has offered to sort out any problems with the Multimo station at Lee's Yard, and install a borehole broadband seismometer at Olveston free of charge. While this is of particular interest to Multimo, this would also greatly be to MVO's benefit.

8 THE ROLE OF THE SOFTWARE ENGINEER WITHIN THE SEISMIC MONITORING PROGRAMME

8.1 Seismic Monitoring Programme

The Seismic Monitoring Programme (SMP) is a collaborate effort between the Seismologist (who has overall responsibility and authority for the whole programme), and five specialist staff including Software Engineer, Network Administrator/Senior Electronics Engineer, Electronics Technician, and two Seismic Technicians. The current Software Engineer is an extremely valuable member of this team, and brings a wide range of skills to the table. If these are applied effectively, there is no doubt that MVO will benefit enormously. The purpose of what follows is to describe in detail the role that should be played by the Software Engineer.

8.2 Software Development

The following steps are involved in developing and maintaining software:

1. The User informs the Software Engineer of a specific need (the User in the case of the SMP is the Seismologist). The Software Engineer may also of course suggest needs to the User, but final decision rests with the User.

2. Software Engineer analyses the needs of the User. At the Software Engineer's request, the User could provide a concept note, or the User may offer it independently. Past practice has been for the Seismologist to present a software specification to the Software Engineer.

3. Software Engineer designs the software and gets the approval of the User before proceeding.

4. Software Engineer writes the code, with considerably freedom to decide the best choice of computer languages.

5. Software Engineer tests the code.

6. Software Engineer documents the code.

7. Software Engineer installs the code and maintains it.

8.3 Definition of Seismic Software

The *seismic software* consists of specialized seismic data acquisition, analysis, archival, restoration and backup software:

1. Acquisition systems include third-party products such as *Scream* (Multimo), *SA24*, *Seislog* and *EarthWorm*. EarthWorm is open-source and the Software Engineer may develop additional EarthWorm modules. Seismic parameters (e.g. triggering parameters, station lists etc.) are set by the Seismologist. Which seismic acquisition systems to use are also decided by the Seismologist.

2. Analysis systems include the third-party product *Seisan*, a number of MVO authored programs (written in the programming languages Matlab and Perl), and the seismic

monitoring webpage. The Seismologist sets the needs, but the method of implementation is at the Software Engineer's discretion (see "Software Development" section above).

3. Archival software includes MVO authored software for robustly recording seismic data onto backup media or centralized online data storage areas.

4. Restoration software includes MVO authored software for restoring/extracting seismic data from backup media onto centralized online data storage areas.

5. Backup software includes MVO authored software for systematic backups of seismic software and configuration settings.

The SMP is contingent on other software (e.g. diagnostic alarm system, integrated monitoring software). Other software is contingent on the SMP (e.g. the volcano alarm system, and, again, integrated monitoring software). However, these are not regarded as "seismic software" since the users include the whole Scientific Team (not just the Seismologist), and therefore all members of the Scientific Team will have needs to be met, and ideas that should be considered.

8.4 Suggested way forward

In this section the desire of the Software Engineer to officially take responsibility for the maintenance and development of the seismic monitoring systems is considered strongly. However, there needs to be regular dialogue, and the Seismologist's consent must be given before any major changes are made. Moreover, the Software Engineer must be responsive to what the Seismologist considers to be the needs of the SMP.

- Responsibility for keeping all existing seismic software running will be delegated to the Software Engineer. This has always been the plan, but in February 2003 the Software Engineer did not know enough about the complex seismic software to maintain & develop it effectively without the assistance of the Seismologist. This still needs to be remedied, but in principle to objective is to transfer responsibility immediately, but to allow close surpervision by the Seismologist initially.

- Seismic software development should follow the guidelines in section 2 above.

- The Seismologist will continue to write small-scale ad-hoc programs for processing seismic data, but these will be stored in private areas, and will be of a temporary nature. If a need for a more general application arises, this will be raised with the Software Engineer as a software development need.

- Windows administration (e.g. disk clean up, defragmenting, service packs, installation of software and *external* PC hardware) for seismic PCs will be delegated to the Software Engineer.

- Linux administration for seismic PCs will be delegated to the Software Engineer. This includes installation of *external* hardware and software. Nobody (including the Seismologist) will be allowed to change settings on the Linux PCs without the permission of the Software Engineer, except in an emergency when the Software Engineer cannot be reached (as occurred in April). The needs of the SMP (expressed by the Seismologist) must however be met.

- Installation of *internal* PC hardware is delegated to the Network Administrator (who insists on this) although he may in turn delegate it to the Software Engineer if he wishes.

8.5 Current needs of the SMP for the Software Engineer's attention

1. **RSAM alarms:** Currently there seems to be a problem with sending RSAM alarms (the modem).

2. **Lacie BigDisk:** The majority of the continuous seismic data, and all the triggered seismic data, had been recovered from tape and put onto this hard drive. It then failed. If the Software Engineer could attempt to recover the data from this hard drive (e.g. by using Norton Utilities from Dave), that could save us a lot of additional work.

3. **Seismic data online:** I would like all the seismic data stored on tapes to be extracted, converted to Seisan format and made available on the centralized data server that I understand Dave has now acquired. I'd like the Software Engineer to implement a simple method for doing this (e.g. extending the "backup" program). I can then recruit Team Seismic and possibly Night Duty Staff to help with the tedious tape work. Several outstanding requests for data by researchers can then be met.

4. **EarthWorm3 autostart:** Last time I checked, EarthWorm3 did not have autostart capability. I think Dave plans to address this situation, but I would be grateful if the Software Engineer would follow this up.

5. **SA242 reinstallation:** Last time I checked the SA24 backup PC, it needed Windows 98 reinstalling, and also the SA24 software. Since then the Software Engineer took this PC home and has been using it for his own purposes. If the Software Engineer could take of the reinstallation and return this PC to MVO, I can take care of setting the seismic parameters for this spare SA24 PC. Its important to have this ready to go in case the primary SA24 PC fails.

6. **Seisan PC duplicate:** I would like Seisan4 to be setup as a mirror for Seisan (it was a mirror in the past), so that one can take over the role of the other in the event of failure of either one. I'd like the Software Engineer to draft a plan for achieving this.

7. **Matlab software:** As mentioned in the previous section, I'd like the Software Engineer to make recommendations about the best strategy for getting the Matlab programs to a point where he would feel happy to maintain/develop these programs in future.

8. **Seismic monitoring webpage:** I'd like the Software Engineer to draft a plan for integrating the seismic monitoring webpage with his other web developments (and database).

The following items will have to wait until I get back:

9. **Data backups:** I'd like the Software Engineer to explain the new seismic data backup procedure, and test its performance so far, and incorporate the safety features from archive2dat. Finally, I'd like the Software Engineer to make recommendations concerning the centralized data server and DLT drive.

10. **Diagnostic alarms:** I'd like to the Software Engineer to make recommendations about possible "failure points" within the seismic software, and make sure these are covered by the diagnostic alarm system.

8.6 Meetings

Since there is rather a lot to discuss and get on top of, and there is a need to rebuild trust, I suggest that when I return in June 2003, we start having a regular weekly "seismic software" meeting. It would be helpful if the MVO Director attended these meetings.

APPENDIX A – ADVERTISEMENT FOR AN IT SPECIALIST

Below is the job advertisement for the position that Joel Maranhao filled (in the capacity of a UN volunteer). Note it stresses the need to *support* the monitoring effort, and work *under the supervision of the Seismologist* in relation to the seismic systems. These were aspects of the position that were seemingly forgotten during the February – May 2003 period, when Joel demanded he take over them.

MONTSERRAT VOLCANO OBSERVATORY

VACANCY FOR A SCIENTIFIC SOFTWARE ENGINEER

The Montserrat Volcano Observatory (MVO) is responsible for real-time monitoring of the active volcano on the Caribbean island of Montserrat, using state of the art monitoring techniques. The main methods used by MVO are seismic, ground deformation and gas monitoring, each of which is computer intensive. Continuous operation of seismic data acquisition and alarm systems is critical for public safety. A vacancy has arisen for an IT Professional to <u>support</u> the monitoring effort.

Main duties will include:

1. *administration of a Windows/Linux computer network*

2. *maintenance and continued development of seismic acquisition and alarm systems, under the supervision of the Seismologist*

3. *development of integrated data analysis and archival software*

4. *other computer hardware and software support required by the scientific staff*

Candidates, who are likely to be graduates, must have relevant experience in the administration of Windows NT/2000 and Unix, including Samba networking. Previous programming experience in Perl is essential. Programming experience in any of Java, C, Matlab, FORTRAN and SQL would be advantageous. Experience with modem communications and SMS messaging is desirable. Suitable candidates who are not graduates but who have experience in these areas will be considered.

Personal qualities that are essential include the ability to work within a small international team and to be highly adaptable. The nature of working with an active volcano requires flexible working practices, including night and weekend duties at times of crisis, or whenever real-time monitoring is compromised.

The post is for one year in the first instance, with the possibility of extension. Terms and conditions depend on qualifications, experience and current location.

Interested candidates should send a full CV that shows clearly the extent of their technical qualifications and experience, together with a covering letter that describes why they think their personal qualities make them suitable for the post, to Dr Peter Dunkley, Director, MVO, [address; fax; e-mail] to arrive no later than 1200GMT on [date].

APPENDIX B – TASK LIST

Below is an example of the task list as it stood on 28th October 2002:

MVO	**Task list**
	Joel Maranhao (JOE) / Glenn Thompson (GLE)
Version	v1.0
Date of release	20021020
Date of last modification	20021028

SAM GUI

+ Create a Matlab interface (with “guide”) to facilitate the use of the script P:\MFILES\RSAM\sam.m. This should include functionality to plot BSAM, CENG and DR data below.

NOTE: startup.m sets up work directories.

Description:

This data have a sampling rate of 1 minute. However, its often nice to smooth these data with a different sampling rate, e.g. 10 minutes, 30 minutes, 60 minutes. Those 4 options would cover it nicely - 1,10,30,60.

Here is an example call for sam.m. It plots data from 30 days ago (snum=now-30) to now (enum=now-4/24, the correction is for UT to local time) It reads data for 3 stations, mlgt, mwhz and mlyt. The type of data is 'rsam'. It doesn't do any special processing (processing = 'none')
It plots it as figure number 1 (fignum=1). It plots each trace as a separate panel - this is signified by using dummy values subfignum=0 and numsubfigs=0. It resamples the data to 60 minutes (resampmethod = 'resample', binsize=60). The plot type is normal (plottype='normal'). Spikes above 2000 counts are removed from the data (despiking = 'despike', threshold=2000). The title is Test Plot (titlestr='Test plot').

```
sam(now-30, now-4/24, {'mlgt';'mwhz';'mlyt'}, 'rsam', 'none', 1,0,0, 'resample', 60,
'normal', 'despike', 2000, 'Test plot');
```

Note, if subfignum and numsubfigs are not set to dummy values, multiple stations will plot in one panel. To see what I mean try:

```
sam(now-30, now-4/24, {'mlgt';'mwhz';'mlyt'}, 'rsam', 'none', 4,2,3,'resample', 60,
'normal', 'despike', 2000, 'Test plot');
```

This divides figure 4 (fignum=4) into 3 panels (numsubfigs=3), and plots the data for all stations in panel 2 (subfignum=2). This is useful when I produce figures for papers - I might want sam data in one panel, event data in another etc. But sam takes care of all this.

Environment: Windows 2000 Pro, Matlab

rbuffer2bsam.c

+ Debug the code in order to create proper BSAM, CENG and DR data in order process it with the Matlab script bob. Combine the functionality of related programs ampengfft.c, read_bob.c and write_bob.c so there is only one program to maintain. Create makefile to make recompilation easier? These programs are stored in /home/software/C on "Seisan".

Environment: C, Linux

The www.mvo.ms/seismology webpage

Structure the website into a hierarchy. Create templates, work on a simple design to present the plotted data with differente resolutions that will fit different screen settings (1024x768, 800x600, 640x480).

Password protect the webpage. Experience from other volcano observatories shows that this is necessary.

Questions:

(i) What do we want to show on the web?
- The starting point should be to look at the Alaska Volcano Observatory internal web page, in particular the IceWeb system that I developed there. Reduced displacement plots and spectrograms can be viewed on various timescales and there is an extensive on-line archive of these images.

(ii) What are we showing at the present moment?

- **EarthWorm helicorder plots** (1 hour long plots for the 5 last hours - and 24 hours plots for the 5 last days). The helicorder plots are available for different stations (Long Ground, Windy Hill, Roaches Yd, St Georges Hil, Garibaldi Hill, South Soufriere, Mango Hill, Lee's Yard).

Suggestions:

+ Create a clickable map to go directly to the different acquisition stations plots (very low priority).

+ Create a simple user interface to access the different plots. ex: Create a dropdown list where you can choose the hour or the day you are interested in.

+ Create a popup help window where the user can get information to understand the plots. Samples of signal representing VT swarms, hybrid swarms, banded tremor, major dome collaspe, regional, small dome collaspe, LP/RF... Warning about the interpretation of the signals (?Disclainmer or something?). These data shall be presented as facts only. The interpretation shall only be made by specialists.

- **EarthWorm 24-hour spectograms** for the 5 last days... give a short definition of a spectrogram and how to read it (popup help window).

- The **counts** and **energy plots**... definition and how to use this data in a popup window?

- The **pictures** of the 2 cameras (Windy Hill and Whites Yard)

Environment: Windows 2000 professional, javascript, perl, matlab, html, ASP, VBScript.

(iii) What kind of information could be useful to have online in the future?
(iv) Why do we want to show this information?

Drive installation on Linux box and set internet access and email facilities

This machine is called "Seisan" and its ip address is 192.123.0.83. It is the hub of the seismic monitoring system.

+ install second DDS-3 DAT drive
+ install second ATA 100GB HDD
+ debug the internet connection – this used to work fine
+ setup the mail services

Environment: Linux

Backup issues

+ Backup RSAM hard drive to DAT tape
+ Create file for all Product Keys for Matlab, Windows etc
+ Make a set of CDs for Windows 3.1, 98, NT4, 2000 and put in data safe
+ Make daily incremental backups of seismic drives (which directories?). SFILES (and WAVFILES?) could be particularly important.
+ Make monthly full system backups?
+ Document the backup and installation procedures for seismic software

General troubleshooting

Machine's name	Description of the problem	What to do	Status
Analyst PC	troubleshoot the message which appears every second – install service pack / reinstall Windows?	Install service pack 3 + stop start the Windows Management Instrumentation service.	OK
Marie's	Cookies, temp files	Run Washandgo	OK

Ricky's	Cookies, temp files	Run Washandgo	OK
…			

Alarm system - Callphone and Sendpage Java applications vs Perl

Description:

This application has been developed initially in Java. Callphone (.class) is called from Perl scripts (send_alarm.pl and test_alarm.pl) in order to issue alarm through telephone and pagers. These Perl scripts are scheduled jobs listed a crontab on seisan machine.
Environment: Linux, perl, java, TAP protocol.

Name of the machine: seisan

+ Rewrite this application in Perl or another language.

What has been done so far:

8. Created the MVOLib Perl libraries.
9. Wrote a new lib for the alarm system: MVOLib::Alarm.pm need to be tested
10. Wrote soufriere_alarm.pl (a copy of volcano_alarm.pl) for the testing
11. Installed the right packages to run the alarm system developed in Perl (ex: Device::Modem).
12. Added MVOLib to the autobackup.csh file

To do:

13. Improve the Callphone routine: if no one answers then ring later
14. Store alarms in a log file
15. Store alarms in a DB for further processing – shows them in a wage page (mySQL/Perl/ASP)
16. Type the right list of phone and pager numbers
17. Testing

Things to discuss later

+ Rewrite Java program that send pages (Perl?)
+ Rewrite script to collect data from QNX
+ Rewrite script to update SFILES and create Matlab database
+ Relational database to store information in Sfiles plus magnitudes and locations
+ Other Matlab GUIs
+ many other things…

APPENDIX C – TEMPLATE SOFTWARE SPECIFICATION

Joel was keen to bring his commercial experience of software engineering to bear on MVO, and I encouraged this. In October 2002, we decided that for any new software I felt needed to be developed, I would draft the requirements by filling in the following template. This was used successfully until February 2003 until Joel began working on his MVO database project and left much of the seismic software in limbo. This procedure should probably be reinstated to provide an audit trail, and be supplemented with additional documentation such as a man page on completion of the software.

MVO	**Specifications**
	Joel Maranhao (JOE) / Glenn Thompson (GLE)
Version	v1.0
Date of release	YYYYMMDD
Date of last modification	YYYYMMDD

<u>Introduction (Abstract)</u>

- **What kind of application or script?**

- Perl script, Visual C++, Visual Basic, Web application

- **What does the application or the script?**

- General introduction of what it does

Description (Detailed Specifications)

- **How does it work?**

Figure: drawn a schema of the application and its components.

Environment: languages, databases, operating systems, paths, environment variables.

Algorithm: description of the different steps of processing.

- **How to make it work?**

Syntax: how to run and use the application or script.

Input: parameters given to the application.

Output: what the application produces.

Files and folders:

Files dependencies: libraries, set of files needed to un properly the application (ex: startup.m for matlab scripts)

Installation: how to install the application.

Results: how to use the results.

Machines: on which machines is installed the application so far.

To be Improved

- **What needs to be done in the future?**

Bug fixes…

- **What can be improved?**

New features…