Significant Event System Tutorial

Geoff Savage
January 2003
Tutorial Outline

- Goals
- Architecture
- Configuration
- User Interfaces
- SE Messages
- Operation
- Front ends
- Client Applications
- Documentation
SES Goals

• Monitor the health of the online system
• Produce, distribute, display, and log events which are significant to the experiment
  – Alarms
  – DAQ state transitions
  – Information
• Archive events for later review
  – Help in data analysis
  – Look for trends
Architecture 1

Key:
SE = Significant event
FIL = Filter request
DBR = Database request
DB = Database info
RC = Run Control
● = Filter
Architecture 2

- Client/Server architecture
- Built on the InterTask Communication package (ITC)
  - Network communications library developed by the computing division for DZero
  - ITC uses the Adaptive Communication Environment
  - Also used to move event data
- Server
  - Holds the current alarms (state) from the detector
  - DAQ state transitions and information are not stored
  - Listens for connection requests from clients on a well known port
Architecture 3

• Client
  – When clients start they connect to the server
  – All clients automatically reconnect to the server if the server is restarted
• Sender clients
  – Generate significant event messages
• Receiver clients
  – Receives significant event messages
  – Filters can be set in the server to restrict the messages received
• Applications can be both a sender and receiver client
Configuration 2

- The SES has two servers running
- We are worried about one server handling the large number of Level 3 SE messages that are generated along with the other SE messages
- The main server keeps the state of the experiment
- The L3 server routes L3 messages
  - Logger writes these messages to a file
  - L3 summary coalesces the messages and sends alarms as necessary (I don’t think this is written yet.)
- Notice that the L3 server is not connected to an Alarm Display or Watcher
Server

- The server holds the current state of the experiment
- The state is really just all the alarms stored in the server at any instance
- Alarms that transition from bad to good then back are stored by sender client for quicker reference
- Alarms that identify a bad condition have the transient alarm type
  - There is no way to detect a bad to good transition
  - After 15 minutes the server sets the good transition
- The server also holds the filters set for each receiver client
Disabled Alarms

- To have the system ignore an alarm that alarm can be disabled via the alarm display
- When disabled, an alarm in the server is marked as disabled and the alarm is stored in a file
- This provides persistence of disabled alarms
- When the server starts it reads in the file containing the list of disabled alarms
  - SES_DISABLE_FILE="/online/config/ses/ses.disable"
- When a new alarm arrives that matches the name of the disabled alarm the server automatically marks that alarm as disabled
Filters 1

- Receiver clients can set a filter in the server to restrict the messages forwarded from the server to the receiver client.
- A filter consists of one or more filter expressions.
- Each filter expression is mapped to a user defined identifier.
- The identifiers of all the filter expressions an SE message passes are added to the SE message by the server before the message is forwarded to the receiver client.
- The alarm display uses this feature to route an incoming SE message to the correct row(s).
Filters 2

• Each filter expression is a string consisting of keywords and python expressions
  – ‘priority > 100’ – messages whose priority field is greater than 100 pass this filter
  – “contains(name, ‘HV’)” – messages whose name field contains the string ‘HV’ pass this filter
• When a message is filtered the keywords are replaced by the value in the keyword field in the message
Filters 3

• An example would be a message whose priority field is set to 200. If the filter expression is ‘priority > 100’ then the server would replace the priority by 200 to get the expression to evaluate; ‘200 > 100’. This evaluates to true and the message is forwarded.

• Keywords
  – mtype, name (det_devtype_loc/attr), priority, host, transition, severity, atype
  – Contains(keyword, “substring”) – evaluates true if the substring is found in the keyword string
Filters 4

• More complex filter expressions can be constructed using the python operators ‘and’, ‘or’, and ‘not’
• “contains(name, ‘HV’) or contains(name, ‘LV’)” – messages whose name field contains the string ‘HV’ or ‘LV’
• “contains(name, ‘HV’) and contains(name, ‘SMT’)” – messages whose name field contains the string ‘HV’ and ‘SMT’.
Logger

- All SE messages received by the server are forwarded to the logger for insertion into a log file.
- The logger occupies a port so that in the default configuration only one logger can run on a host.
- There are three instances when a new log file is created.
  - When the logger starts
  - At midnight
  - When the server is restarted
SE Log Files

• When a new SE log file is created the logger requests the current state from the server
  – This make searching the log files easier
• Messaging sequence
  – The server replies with a “begin state transfer” message
  – The server then sends the current state (all alarm messages currently in the server)
  – The server ends with a “end state transfer” message
• This provides a clear boundary in the log file between old and new messages
• SE log file browsing is a different talk
Log File Locations

- Official SE log files are in /online/log/ses
- L3 SE log files are in /online/log/ses/l3
- Both servers write debugging information to files in /online/log/ses/server
- Both loggers write debugging information to files in /online/log/ses/logger
- The alarm watcher writes debugging information to /online/log/ses/aw
- There is a cron job that zips files in these directories more than four (?) days old
Alarm Watcher 1

- The alarm watcher pauses a run and speaks when three conditions are met
  - The run is pausable (autopause=’yes’ in the run configuration file)
  - The alarm severity is major
  - The alarms experiment priority is greater than 100
- The alarm watcher is a receiver client
- The alarm watcher has a configuration file that contains the filter for restricting the messages the watcher processes
  - SES_AW_CONFIG_FILE="/online/config/ses/aw.config"
Alarm Watcher 2

- The alarm watcher occupies a port so that in the default configuration only one alarm watcher can run on a host
- The alarm watcher monitor is available to adjust parameters in the alarm watcher
- Every two minutes the alarm watcher checks to see if an alarm that has paused a run is cleared, if not then a reminder message is spoken but the run is not paused again
The Alarms tab shows the name of the alarm(s) that paused the current run in the right hand column.

The left hand column shows the time the alarm was detected.

This always shows alarms that would pause a run even if speaking and run pausing is disabled.

Details on the alarms are available from the alarm watcher row of the alarm display in the major severity button.
Alarm Watcher Monitor 2

- The Report tab shows the configuration of the alarm watcher.
- The filter gets alarms with priority greater than 100 and run messages to determine the speak and pause values. Speak and pause are dependent on the running conditions.
- Every 120 seconds (limit) a reminder is squawked if a run pausing alarm remains in the alarm watcher.
- Meter represents the count up for reminder squawks.
- Coor is the status of the connection to COOR.
• Report updates the report data.
• The speak, pause, and meter limit values can be adjusted using the items under the server menu.
• Stop needs to be removed when used online. (Handy for testing.)
• Coor must be connected at startup.
Alarm Display 1
Alarm Display 2

- The number on each button reflects the number of alarms of a severity level that pass the filter for that row
- Each row has a different filter expression
- Alarms that pass multiple filter expressions will appear in multiple rows
- The default configuration file is read at startup
  - `SES_AD_CONFIG_FILE="/online/config/ses/ad.config"
  - Different configuration files are specified with the -c command line argument
Alarm display default configuration file.

```python
addRow(‘CAL’, "contains(name, ’CAL’)")
addRow(‘CFT’, "contains(name, ’CFT’)")
addRow(‘MUO’, "contains(name, ’MUO’)")
addRow(‘SMT’, "contains(name, ’SMT’)")
addRow(‘LUM’, "contains(name, ’LUM’)")
addRow(‘Control’, "contains(name, ’CTL’)")
addRow(‘Online’, "contains(name, ’ONL’) and not contains(name, ’SDAQ’)")
addRow(‘SDAQ’, "contains(name, ’SDAQ’)")
addRow(‘Magnet’, "contains(name, ’MAG’)")
addRow(‘Level 3 DAQ’, "contains(name, ’L3DAQ’)")
addRow(‘Level 2’, "contains(name, ’L2’)")
addRow(‘FPD’, "contains(name, ’FPD’)")
addRow(‘Alarm Watcher’, "contains(mtype, ’alarm’) and (priority > 100)")
```
Alarm Display 4

- Bad alarms are classified by severity
  - Minor – monitor the situation
  - Major – fix the problem
  - Invalid – read or write did not complete correctly

- Disabled alarms
  - Stay in the disabled column even if the state changes
  - Can enable (although there seems to be some problems with this that I haven't tracked down yet)
  - Will not pause runs

- Good
  - Remains in the good column for five minutes
  - An oscillating alarm can be seen in this column
Alarm Watcher Row

- The alarm watcher and the alarm watcher row in the alarm display have the same filter so they receive the same alarm messages.
- The major alarm button in this row will show the same alarms seen in the alarm watcher monitor.
- Remember that the alarm watcher only pauses runs for major alarms.
- If the filter was not loose enough to get alarms of all severities then a transition from a major to minor, invalid, or disabled alarm would not clear the alarm from the alarm watcher.
### Alarm Display 5

#### Left click a button to see the names of all the alarms in that category

#### Left click a name then click the show button to see the contents of the message displayed

<table>
<thead>
<tr>
<th>Group Name</th>
<th>MAJOR</th>
<th>MINOR</th>
<th>INVALID</th>
<th>DISABLED</th>
<th>GOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAL</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>CFT</td>
<td>0</td>
<td>32</td>
<td>22</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MUX</td>
<td>0</td>
<td>522</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>SMT</td>
<td>393</td>
<td>442</td>
<td>15</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>LUN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Control</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Online</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>SDAQ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Magnet</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Level 2 DAQ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Level 3 DAQ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FPD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Alarm Watcher</td>
<td>0</td>
<td>8</td>
<td>4</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>Status</td>
<td>Connection to server started</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Filename:** SES Tutorial, January 2003
Message Display 1
Message Display 2

- Show details about the alarm
- The contents of all the message fields are listed (this display needs some more work, volunteers?)
- For an alarm that oscillates within the five minute window that the good alarms remain, there will be multiple alarms listed in this view
Message Display 3

- Close – close the message display window
- Disable – disable the alarm show in the display
- Control
  - Bring up a parameter page for EPICS alarms to see current values in the front ends
  - For non-EPICS alarms the button is not present
- Guidance – display helpful information inserted into the database or in a file
- Command
  - Execute a command
  - Maybe an expert application for debugging
SE Messages 1

Each message is sent as a string with thirteen (13) space separated fields

- Version
- Timestamp
- Message type
- Name
- Priority
- Host name
- Database locator
- Parent
- Children
- Transition
- Severity
- Alarm type
- Parameters
SE Messages 2

- Version – currently v4
- Timestamp – seconds since the epoch (1970)
- Message type - command, alarm, info, l3, run, reply, server
- Name – must follow the dzero naming convention
- Priority
  - 0-255 importance to the experiment
  - Major alarms with priority greater than 100 pause runs
- Host name – where the message originated
SE Message 3

• Database locator
  - Currently unused
  - Speed up database lookups

• Transition
  - Bad (good -> bad)
  - Good (bad -> good)
  - Transient – notification of a problem not following the good/bad paradigm
  - Transient alarms are cleared by the server after 15 minutes. If the alarm condition is cleared the server still holds the alarm.
SE Message 4

- Severity – major, minor, invalid
- Alarm type
  - Binary – no data
  - Comment – data is a text string
  - Analog – values or text
- Parameters
  - Based on the alarm type
  - Binary – no extra data
  - Comment – text string
  - Analog – values or text
  - The difference is in the alarm message constructor
Operation 1

• The two servers, two loggers, and alarm watcher run as daemons
  - start_daq ses – start all daemons
  - stop_daq ses – stops all daemons

• User interfaces
  - start_daq alarm_display
  - start_daq alarmWatcher_mon
  - start_daq message_board
  - start_daq l3_message_board

• The message board is a debugging tool that displays the last 1,000 SE messages received in a window
Operation 2

• Each daemon can be started and stopped individually
  – start_daq ses_server
  – start_daq ses_l3_server
  – start_daq ses_logger
  – start_daq ses_l3_logger
  – start_daq alarmWatcher
  – Stop_daq ses_server
  – stop_daq ses_l3_server
  – stop_daq ses_logger
  – stop_daq ses_l3_logger
  – stop_daq alarmWatcher
Operation 3

- Configuring the system requires the identification of a host and port number for each application
- User interfaces run on the local host and connect to the server
- The configuration is stored in
  /online/data/d0online/d0online_names.py
    - SES_SERVER_ADDR="d0ol39.fnal.gov:52150"
    - SES_LOGGER_ADDR="d0olc.fnal.gov:52151"
    - SES_L3_SERVER_ADDR="d0ol39.fnal.gov:52245"
    - SES_L3_LOGGER_ADDR="d0olc.fnal.gov:52246"
    - SES_ALARM_WATCHER_ADDR="d0olc.fnal.gov:52153"
Operation 4

- Looking to see if the daemons are running
- Log into the computer specified in d0online_names.py
  - ps auxwww | grep seserver
  - ps auxwww | grep selogger
  - ps auxwww | grep watcher
  - www = wide, wide, wide
- On Linux all the threads are shown (hundreds for the servers)
- On OSF1 only the main thread is shown
- When starting each daemon all the data is passed via command line arguments
Front Ends 1

• Send messages to the server when alarms are determined by EPICS
  – Requires users to set alarm limits in the EPICS records
  – It’s also handy to set hysteresis limits to minimize oscillations
• Limit alarms
  – Ai, ao, longin, and longout
  – Return value, limits, and status
• State alarms
  – Bi, bo, mbbi, mbbo
  – Return state and status
• All other alarms return the status
Front Ends 2

- All the volatile information about the alarm is placed in a message in a posix queue
  - PV name, value, severity, status, time
- An independent task (seTask) removes the message, creates an SE message, and sends the SE message to the server
- The SES task on the front end (seTask) is responsible for maintaining the connection to the server
  - At startup the connection is established
  - Broken connections are reconnected
Documentation

- In the SigEvtSys CVS package
- Still being upgraded
- For users
  - This tutorial
  - ses_user.pdf
- Application developers
  - ses_api.pdf
- Online usage
  - ses_online.pdf
- SES developers
  - ses.pdf