Design Approach Review

GUARDIAN

(General User Alert Display pANel)

AWIPS Build OB4

MDL

Tom Filiaggi
July, 2003
GUARDIAN

AGENDA

- Overview
- Data Flow
- Software Design
- (Data Handling)
- Graphical Interface
- Installation
- Performance
- Testing
- Hardware Resources
- Responsible
- Documentation
- Schedule
Background

- Currently, the AWIPS users can receive messages from various software in various forms with little control.
- Anecdotal evidence suggests that users are rather unhappy about how AWIPS software developers have chosen to notify them of various real-time issues, including:
  - Overwhelming audio
  - Unnecessary pop-ups
  - Lack of “pertinent” information (which is, of course, subjective)
- The growing number of meteorological monitors is quickly running out of space on the D2D tool bar.
Overview

Implementation Strategy

- Develop a new communicator to allow the user to filter what software messages they receive.
- Develop a new communicator to allow the user to define how they wish to be notified of messages with varying sources and priorities.
- Have a persistent process running on each workstation that utilizes general AWIPS IPC methods.
The General User Alert Display Panel will:

- Provide a vehicle to communicate messages from software to the user. These messages can be:
  - From meteorological monitoring software
  - System health alerts
  - Radar ingest messages
  - Based on Informix Triggers
  - Messages from any other client application . . .

- Provide methods to allow users to configure how these messages get communicated: blink, beep, pop-up, or even run an action script or play a sound file.
GUARDIAN

Customization

Set Up

Used Customization

Default Customization

Furnished Customization #1

Furnished Customization #2

Furnished Customization #3

GUARDIAN GUI
GUARDIAN

Message Receipt

- Process
- Application
- Local Application
- Script

Message Receipt

Work Station
GUARDIAN

Data Flow

Guardian GUI

Configuration Data

Guardian Processor

Guardian Log Files

NEW, persistent

Process

Application

Data Controller/COMMS Router

Guardian pinger

NEW, not persistent

Local Application

Script
GUARDIAN

Software Design

P - Programming Languages used
  – C++
  – TCL/TK
  – Perhaps shell script

P - COTS usage
  – TCL

P - Machine Specific Dependencies
  – LINUX (no plans for HP compliance)

P - Service APIs Required
  – none
P Use C++ foundation, with Tcl Interpreter.
P Use AWIPS IPC: registration and message handling.
P Use comprehensive configuration data, to provide user-friendly flexibility.
P Attempt to use one executable which will handle all of the above items.
P Run on each workstation to allow workstation-specific configuration.
**GUARDIAN**

Graphical User Interface

- **Must** remain “on top”, above all other windows!
- **Must** occupy minimal screen-estate, due to ‘on top’ status!
- Provide flexible methods to configure how messages get conveyed to the user.
- Provide customizable GUI appearance.
- Provide methods to store and retrieve configurations.
- Provide a small number of default settings, to ensure ‘out of the box’ use. Examples of such modes could be:
  - “Severe Weather”
  - “Forecast”
  - “System Monitor”
  - “Text WorkStation”
GUARDIAN

Graphical User Interface

- SOURCE
  - FFMP
  - SCAN
  - WWA_1
  - WWA_2
- ACTION
  - high
  - text:
  - blink:
  - pop-up:
  - priority:
  - audio:
    - scream.wav
    - help.au
  - action:
    - gun.sh
- background/img:
  - doom.jpg
  - help.gif
  - mush.png
  - #9954ac
  - green
- foreground:
  - red
  - yellow
  - #9954ac
  - #af7d5e
- COLOR
  - All
- LAYOUT
  - quad
  - log_length:
    - SCAN
    - MISC
    - FFMP
- Bitmap: clock.bmp
- Interval: 5
- Revert Time: 45
GUARDIAN

Installation

- Need to start GUARDIAN upon log-in! Log-in files will need to be edited.
- No changes to National Metadata files
- No changes to site-modified metadata files
- No changes to runtime setup files
- No expected cron usage
- No new runtime disk partitioning / directories anticipated being created
- New runtime metadata files created: Configuration files
- No core runtime system services changes anticipated
- No new COTS / freeware runtime packages anticipated
Assessment of performance
  - Minimal new CPU load
Assessment of shared services with new design
  - Expected increase in IPC Comms traffic.
No anticipated issues with algorithmic performance
Disk I/O usage to be determined
No anticipated use of remote shell, rcp, or other such system calls
Internal Testing: NHDW, NHDA

Alpha test: Yet to be determined, but have had several volunteer WFOs.
GUARDIAN

Hardware/Resource Usage Design

- No new Hardware or mods to existing hardware required by this item
- Additional Disk Space
  - Log space will depend on degree of use. May be significant.
- No anticipated use of Omniback/ tapedrive
- No anticipated use of the WAN
- No anticipated use of the SBN
- No potential problematic use of special hardware resources
## GUARDIAN

### Assignment of Responsible Individuals

<table>
<thead>
<tr>
<th>P Main Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom Filiaggi - MDL: Lead</td>
</tr>
</tbody>
</table>
GUARDIAN

Schedule

P Prototype Preparation
  • Fall, 2003

P User Interface Review & Alpha Testing
  • November, 2003

P End of Development
  • January, 2004