Awareness Gauges

University of California, Irvine

Motivation

Whether one is a pilot or a software engineer, awareness gauges enable users to monitor key aspects of their environment without distracting their attention away from the primary task. Whereas a pilot can glance at an altitude gauge and in that swift glance be made aware of the altitude so that they can return attention to flying, a software engineer will benefit from gauges that can also be swiftly glanced at to provide feedback on the state of a system being modified dynamically while in use. The software engineer can then use this information to refocus their efforts on compensating for any problems encountered when dynamically changing the system, or when engaging in a variety of other software engineering tasks.

To accomplish this goal, we have developed an architecture for providing notification services optimized for powering gauges and awareness technologies. This approach, the Cross Application Subscription Service (CASS) architecture has been implemented using our prototype, the CASS Information Update Server (CASSIUS). While past notification services have been optimized for software interoperability, CASS is instead designed to allow rapid design of multi-modal gauges that use diverse techniques for representing information to users.

Key Benefits

- Monitors and integrates information from distributed sources (physical, digital, and virtual).
- Rapid development of diverse awareness gauges.
- Each gauge can focus on a different aspect of information.
- Each gauge can represent information in different styles.
- Each recipient of awareness information can select gauges whose style, focus, and intrusiveness best fit within their work environment.
- Designers can select a degree of tradeoff between the detail and variety of information (the depth and breadth of awareness).

AWACS Gauges

Gauges displayed below were implemented to represent activity within the UCI AWACS simulator. Each gauge represents different information about messages passed between AWACS processes. For example, the architecture diagram in Figure 1-A shows which processes and components are having activity and how much. The traffic light in Figure 1-B indicates when a dangerous sequence of events has occurred which could result in system deadlock or other problematic states. The line graph in Figure 1-D focuses on overall system activity. Each is useful for allowing developers to identify different types of system information at a glance.

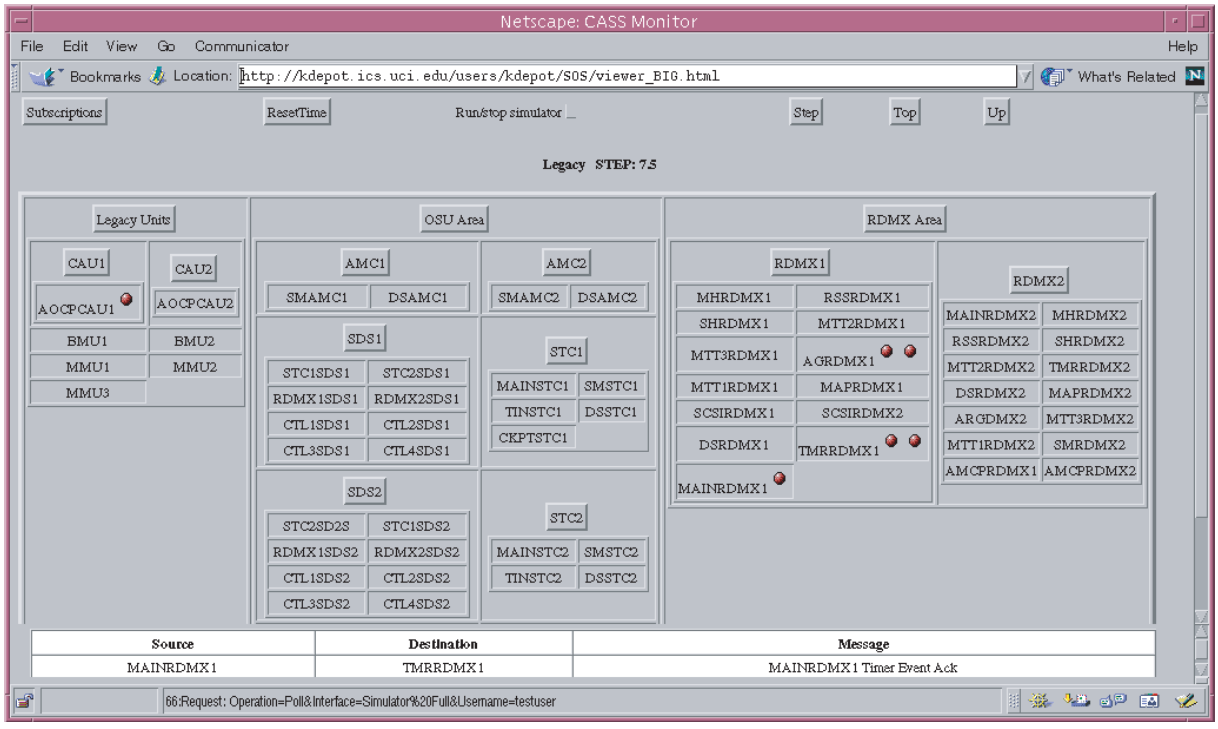

Figure 1-A: A Map of areas, components and processes, with dots representing messages passed/received by objects.

Figure 1-B: Traffic light indicates that system has entered dangerous state.

Figure 1-C: Clock shows how long simulation has successfully run.

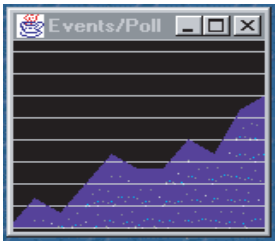

Figure 1-D: Graph shows trace of activity over time, emphasizing times of heavy load.

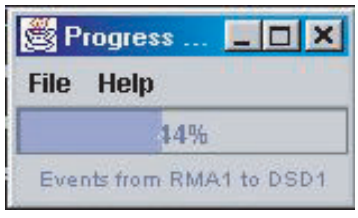

Figure 1-E: Monitors messages passed between two components, notifying user when the desired amount of traffic has occurred.

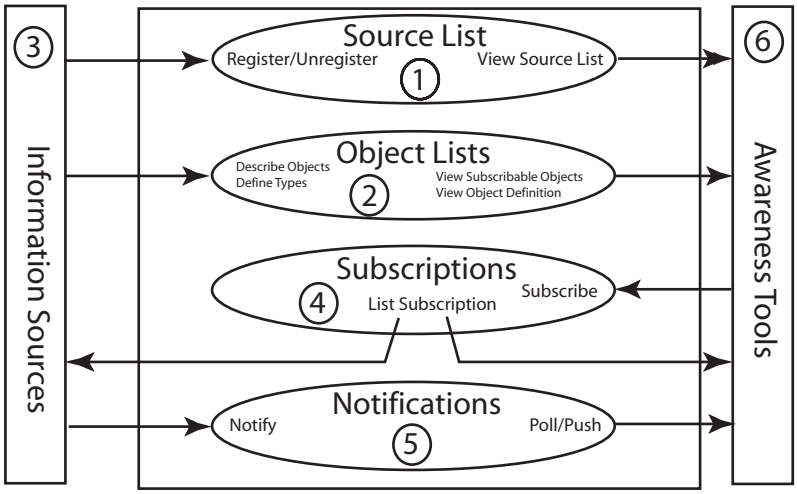

Figure 2: CASS Service Architecture

CASS Architecture

The architecture needs to provide a set of services to enable users to subscribe to any type of awareness information and to have that information represented in the style that best suits the users work environment. Numbers below correspond to numbers in Figure 2.

1) Provide users with a mechanism to locate information sources that can affect their work.
2) Isolate relevant subsets of information from that source.
3) Detect events within the information sources and subsets.
4) Allow people to subscribe to notification of changes to the information source or the subset of the source.
5) Send notifications of changes to a notification server, to be sent on to the awareness tools.
6) Provide effective user interface tools for representing awareness information to users.

Interchangeable Awareness Gauges

A critical goal of the CASS architecture is to enable awareness gauges to be interchangeable. We not only want users to be able to choose an awareness gauge that presents information in a manner that suits their work practices and work environment, but to be able to change gauges as the nature of their work or environment changes. In the case of a pilot, a numerical display or dial that shows the altitude may be just fine for normal conditions. Under other conditions (where the pilot does not want to take their gaze off of something else), some pilots may prefer an audio or haptic representation of height, others may prefer to use gauges that utilize different senses or different channels of visual information, which they are better able to monitor without distraction.

The same is true for software engineers. A gauge that displays the message passing in the AWACS simulator can be used to monitor the flow of information, a different gauge might represent load on the architecture, and a third might monitor for problematic states. These might all be used together, or one at a time as the developer's situation changes.

Furthermore, the manner in which these are represented might vary from user to user and from situation to situation. For example, a developer who wants to know when a problematic state has been reached may want a dialog box to pop up, or another form of alert from the desktop that is appropriate only if the user is currently using that computer. If they are away from their computer, they may also want loud beeping sounds or a message sent to a pager to draw their attention to that computer and the fact that a problem has occurred. For users not responsible for fixing this type of problem, they simply may want to be sent a weekly log of the problematic states are reached — a radically different style of awareness gauge for a different type of user.

Diverse Notification Servers

To enable any awareness gauge to understand the notifications from any system and be able to meaningfully represent it to the user, the CASS architecture assumes a fixed notification format. This format can be provided by a variety of notification servers. We have in fact used the CASS architecture with the Elvin and Siena servers as well as CASSIUS.

The specific format used by CASSIUS consists of a textual summary of the event, a numerical value quantifying and characterizing the event, an object that the event affected, and other information with predefined interpretations. The result is that anyone who builds an awareness gauge will know the format and interpretation for data from any source and can design the gauge to present that information regardless of the source. Different implementations of the CASS strategy may specify a different notification format, different fields, and different interpretations, but as long as all tools in an environment agree on a notification format, CASS is a powerful technology for creating awareness gauges.

References

Kantor, M., Redmiles, D. *Creating an Infrastructure for Ubiquitous Awareness*, Proceedings of INTERACT 2001, Tokyo, Japan, July 2001.

de Souza, C.R.B., Basaveswara, S.D., Redmiles, D. *Supporting Global Software Development with Event Notification Servers,* Proceedings of the International Workshop on Global Software Development, ICSE 2000, Orlando, FL, May 21, 2002.

Contact Information

Professor David F. Redmiles
Dr. Michael L. Kantor
Cleidson R. B. Souza
Information and Computer Science
University of California
Irvine, California 92697-3425

For more information about this work please contact David Redmiles at:

redmiles@ics.uci.edu
949-824-3823 (phone)
949-824-1715 (fax)

This material is based upon work sponsored by the Defense Advanced Research Projects Agency, and Rome Laboratory, Air Force Materiel Command, USAF, under contract number F30602-00-2-0607. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.