Integration of User-Developed Software with SIMDIS

W.A. Doughty, J. Binford, and D. Graves
Tactical Electronic Warfare Division
D. Emminizer
ITT Advanced Engineering and Sciences

Introduction: SIMDIS is a software toolkit that provides two- and three-dimensional interactive graphical and video display of live and postprocessed simulation, test, and operational data. SIMDIS has evolved from a display tool for the output of missile models to a premier government off-the-shelf (GOTS) product for advanced situational awareness and visual analysis. Figure 11 illustrates the various applications of SIMDIS to the Department of Defense (DoD) community. Since its inception as an Office of Naval Research-sponsored project, the toolkit has experienced significant userbase growth, with more than 1,000 current registrations within the DoD community. The toolkit has also undergone significant improvements in features and functionality. To better support the growth in capabilities, and to allow the community to customize the toolkit for their needs, a modularized plug-in applications programming interface (API) was designed and implemented. This plug-in API facilitates the integration of custom user-developed software with the SIMDIS toolkit. The plug-in approach improves the software development process by promoting modularization and reuse of software components. The approach also provides users with the ability to independently add new features to the SIMDIS toolkit based on the user's specific requirements. These new features can be added without the knowledge of, or access to, the SIMDIS source code.

Modularization of Software Components: Throughout the development cycle, enhancements and features specific to certain users have been continually integrated into the SIMDIS application's source code. While this approach has been very successful, the new product starts to overgrow its original design and, as such, becomes more complicated to use and maintain. This complexity problem is greatly improved through the use of modularization, and is one of the driving factors behind the development of the plus-in API.

User-Developed Software Modules: Specialized knowledge is necessary to develop and maintain many user-requested features, a problem that further complicates custom development. A good example relates to recent efforts in processing test and training range data from DoD ranges. Each range has a unique data format that requires development of interface software for display with SIMDIS. Typically, NRL developers work with range personnel to understand their specific data and implement the software required to integrate with the toolkit. The software is maintained by NRL and must be revised and upgraded with each change to the range data format. Providing users with the ability to develop and maintain their own software modules eliminates the need for SIMDIS developers to be directly involved with routine range software maintenance.

SIMDIS Plug-in API: Implementing a plug-in software architecture for SIMDIS solves the complex-

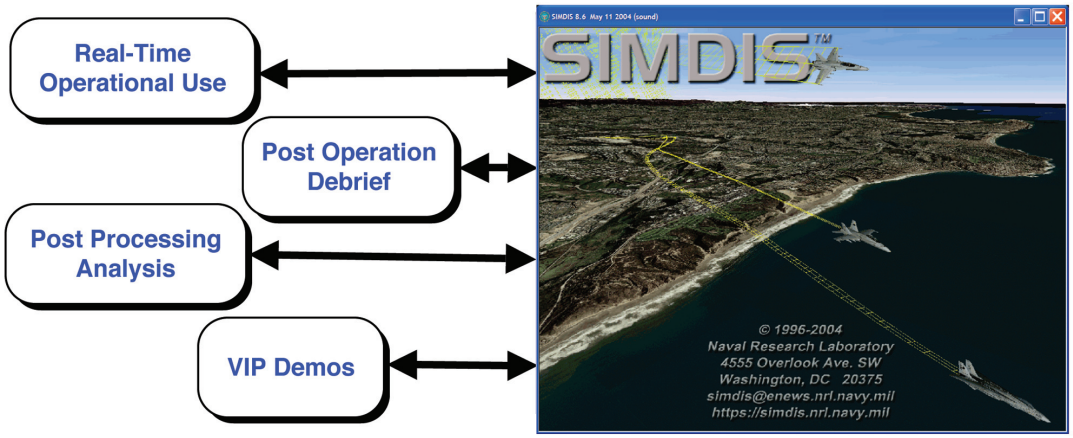

FIGURE 11
SIMDIS is a set of GOTS software tools used by DoD ranges and system centers to support 3D analysis and visualization of test and training missions for air, sea, and undersea warfare areas.

ity problem. A plug-in architecture allows third-party developers to read and write SIMDIS data, create custom pull-down menus, and generate specialized user defined graphics. Figure 12 illustrates the relationship between SIMDIS, the plug-in API, user applications, and external data sources.

A plug-in API provides modularity and structure to SIMDIS, while allowing users to create specialized components that can be shared with other users. SIMDIS provides the display engine and data maintenance functionality for all plug-ins. Plug-ins provide the unique or enhanced functionality to the base tool.

The plug-in API developed for SIMDIS is cross-platform capable. Just as SIMDIS supports Linux, SGI, Sun, and various Windows operating systems, the plug-in API is supported across all four architectures. Plug-in applications are dynamically loaded library (DLL) files for Windows platforms and shared object (SO) files for UNIX systems. Both formats allow users to add custom software modules to the SIMDIS runtime environment.

Several difficulties arise when developing a plug-in interface capability. An important issue is backward compatibility. Although new releases of SIMDIS support newer versions of the plug-in API, each release must utilize version management to allow plug-ins created with previous API versions to remain compatible. Another issue is that improperly developed plug-ins may not only fail to function properly, but may cause the entire SIMDIS application to fail. Special consideration and testing must be performed on all plug-ins prior to their inclusion in an operational fielded version of the software. Ultimately, the development of the plug-in API results in careful tradeoffs between power, flexibility, speed, fail-safe capabilities, and ease of use.

Applications: Over the past year, the Southern California Offshore Range (SCORE) has been a key sponsor of the plug-in API development for SIMDIS. As such, many of the initial plug-ins supported the SCORE users, providing realistic test cases to improve the overall plug-in API design.

One of the initial SCORE plug-ins was a special analysis tool that enabled real-time calculations and display of exercise platform data, such as relative slant and ground distances, true and relative bearing angles, and RF and optical horizon calculations. The plug-in adds a special GUI in the SIMDIS toolkit applications that provide the operator, during a Fleet exercise, the ability to select any number of object pairings for monitoring relative angle and range information.

Another plug-in developed for the SCORE range focuses on exercise events. This Event plug-in was developed to read XML-formatted events using the Simple Object Access Protocol (SOAP) on a standard TCP/IP network connection. The events are items such as marking buoy drops during an antisubmarine warfare exercise, and are entered into a separate non-NRL application by the exercise operators. This application acts as a server, providing the data over the network using the SOAP protocol. The Event plug-in then receives and displays the information on the SIMDIS 3D application for use by the local and remote exercise operators.

The SCORE Pan Tilt Zoom (PTZ) Control plug-in monitors and directs range surveillance cameras located on San Clemente Island. Using track data sources fused and visualized by the SIMDIS application, SCORE operators track platforms and view the digital video of remote cameras in real-time. Figure 13 shows the PTZ Control plug-in tracking a ship with SIMDIS. It pictures the method for directing a camera to view the ship, as well as the real-time streamed video display.

The Pacific Missile Range Facility (PMRF) has sponsored the development of a new two-dimensional GOTS open-source plotting application, called Plot-XY. Plot-XY is designed to be compatible with the SIMDIS plug-in API, so that plug-ins written for one application can be loaded by the other. The plot application relies solely on plug-ins to supply data for graphing. Plug-ins are included in the Plot-XY distribution for reading native SIMDIS file formats, the SIMDIS Data Client Server (DCS) network format, and the PMRF iNet network and file formats. Plot-XY is easily configured to import other data formats through plug-ins.

Summary: The NRL SIMDIS toolkit continues to gain acceptance within the DoD community. The addition of the powerful plug-in API capability is a much-needed enhancement. It allows users to modify and add capabilities to satisfy their own requirements. Users can add these capabilities without the need to understand or even possess the SIMDIS source code. The plug-in API also allows greater modularity in adding new features, resulting in a more maintainable baseline of the SIMDIS toolkit. Overall, the plug-in API will be of great benefit as common modules are shared, providing an impetus and mechanism to encourage greater software reuse and interoperability within the DoD community. For more information on SIMDIS, visit the website at https://simdis.nrl.navy.mil.

[Sponsored by SCORE and PMRF]

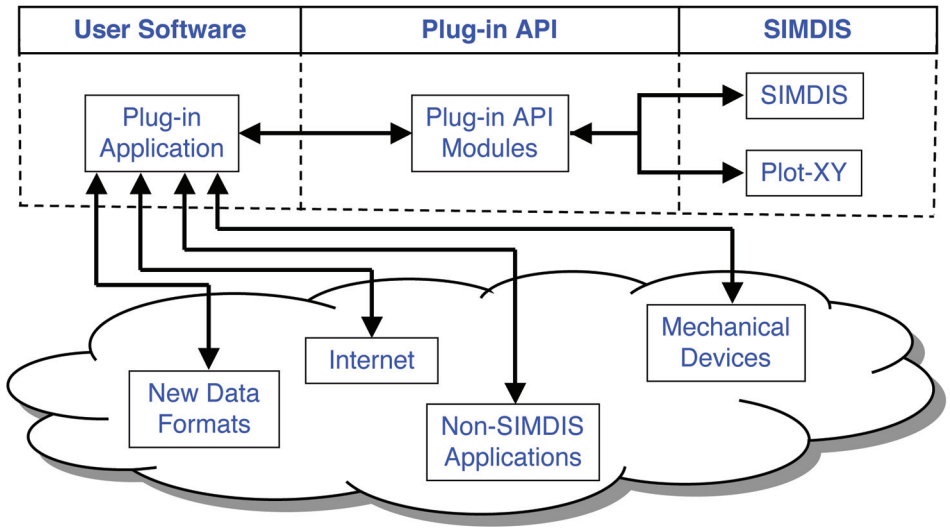

FIGURE 12
Integration of external data sources with SIMDIS through the plug-in API.

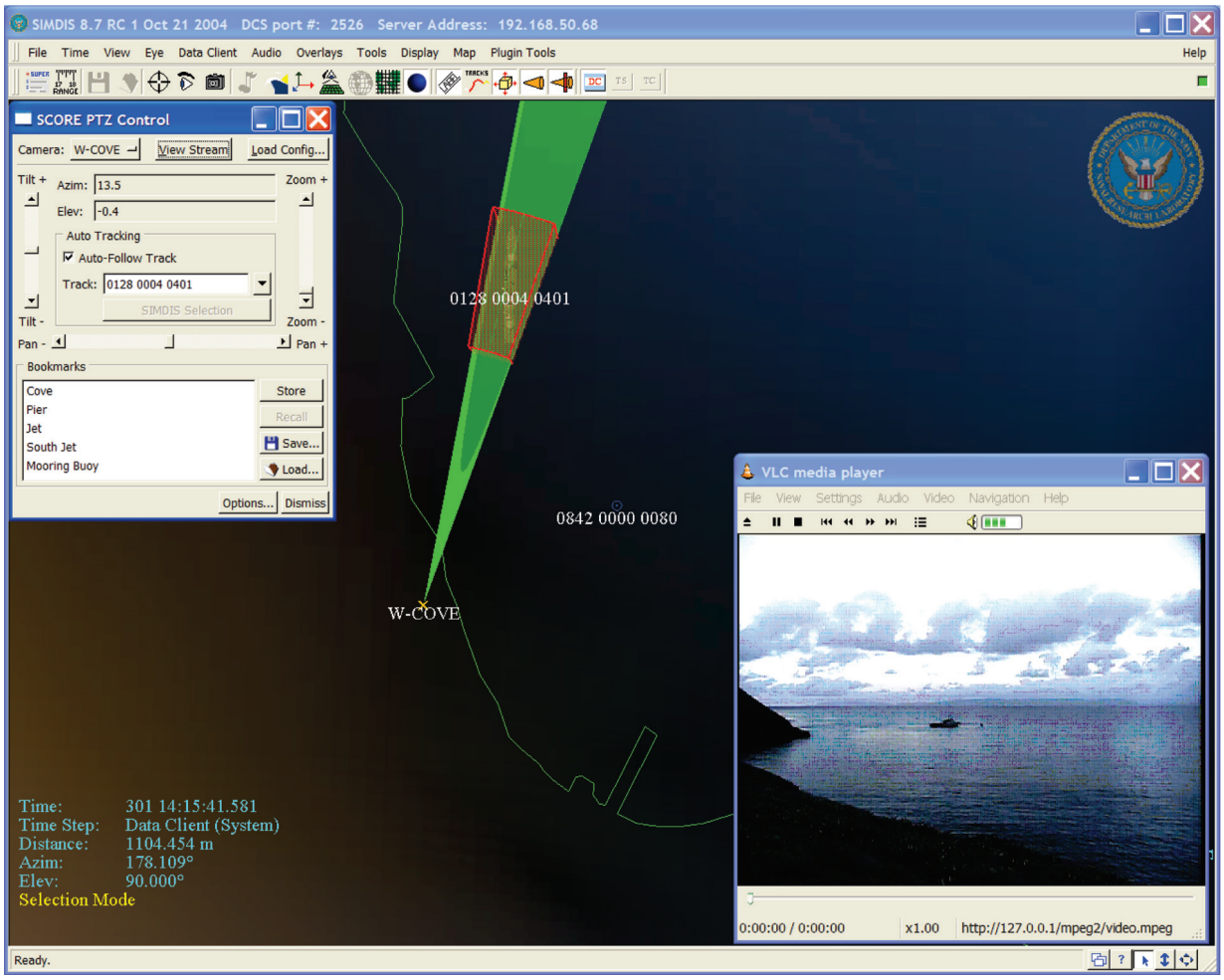

FIGURE 13
SCORE PTZ Control plug-in directs a camera to follow the radar track of a ship and simultaneously displays the live video.