Using The IVIS SDK In Vehicle Simulations

Roger K. Shultz, PhD.

NEXIQ Technologies, Inc. 2475 Coral Court Coralville, Iowa 52241 Phone: 319-545-5400 Fax: 319-545-5315 roger.shultz@nexiq.com

October 21, 2001

ABSTRACT

An In-Vehicle Information System Software Development Kit (IVIS SDK) enables rapid software product development. The IVIS SDK generates code into a production vehicle platform while providing user interface designers with drag and drop auto-generation support.

By using an In-Vehicle Information System (IVIS) software framework, applications move from experimentation to production-ready, while compressing the development time. This lowering of development time and cost frees the system designer to study the impact of IVIS applications on safe and effective vehicle operation.

During the simulation stages of a new vehicle system, experiments with a range of graphical user interfaces are now possible. By replacing a simulation cabin's standard instrumentation with LCD panels, the IVIS SDK provides wide experimentation with safety and usability.

In addition, the IVIS software framework architecture is designed for the integration of real-time data into trigger conditions. These trigger conditions comprise logical combinations of data sources (such as road speed, location, engine temperature, proximity to vehicles, etc). The application designer does not care if these data values are from a simulator or actual vehicle sensors. The designer uses this data source abstraction to ensure that the important data conditions are presented to the vehicle operator at the proper point in time.

Control of devices is also a fundamental part of the IVIS framework. Example devices include CD players, cellular phones, wireless modems, user feedback indicators, etc. These devices are either controlled autonomously or through user interaction.

Incorporating the IVIS SDK into a simulation environment adds flexibility and speed to experimentation. The underlying software framework abstracts the vehicle and its environment. The interface designer quickly moves new techniques for user information presentation or control into simulation experiments.

IVIS SDK OVERVIEW

An IVIS SDK provides automotive human-factors engineers with tools that allow easy and quick builds of invehicle applications that are production ready. IVIS SDK users need no software development expertise. These applications are portable to different hardware and software platforms, reducing the cost for the automotive manufacturers as they integrate new technologies in the vehicle¹.

The IVIS development environment provides mechanisms for plugging applications into the IVIS framework. A visual development environment, such as the Graphical User Interface Design Environment (GUIDE)¹, generates the user interface utilizing real-time graphics and corresponding user input devices such as buttons, knobs, and voice activated commands. In addition, GUIDE establishes each application's access to vehicle data such as speed, oil pressure, or engine RPM. Figure 1 shows an instrument panel under development in GUIDE. Drop down menus associate vehicle data with trigger conditions for indicator visibility, gauge needle position, and many other graphical component properties.



Figure 1. GUIDE View

An IVIS SDK development environment can provide code generation with rapid and error free application integration. The following IVIS SDK capabilities have been shown to provide fast and low-cost system development¹:

- Streamlined performance for processor- and memory-constrained embedded systems
- Simple modifications for changing communications, vehicle parametric data, devices, and graphical user interfaces
- Easy extension of application software modules using vehicle data and devices
- An interoperable framework with interfaces for communication, operating system, device drivers, graphics hardware, and system definition tools
- Reliability with error reporting, error analysis, and tools for development, testing, and debugging

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288). 2

- Anti-aliased, real-time graphics for multi-colored LCD panel vehicle interfaces
- Platform independence for applications with plug and play devices and vehicle buses

Simulation of IVIS applications can be supported on the PC and embedded platforms. Values for data sources drive an IVIS application through the IVIS simulator. With the simulator, an executable can be tested running on the same PC, running connected to a serial-port, or running connected via TCP/IP.

An example simulator interface is shown in Figure 2. Each data source is available for simulation. These include integer, string, list, and image data sources. The user can select different data transports (e.g. TCP/IP or Serial) for communication between the executable and the simulator.



Figure 2. The IVIS SDK GUIDE Simulator

Combined together, technologies such as GUIDE, the IVIS Software Framework, and the Simulator provide a powerful application generation kit. A user interface designer is free to move quickly from testing on a PC to testing on an actual vehicle platform. Once experiments validate the effectiveness of the interface, the same application code is rapidly moved to production.

SIMULATION EXPERIMENTS USING THE IVIS SDK

Display processing units and the IVIS SDK extend vehicle-driving simulations. The flexibility of an IVIS SDK allows an experimenter to rapidly define, try, and re-define user interfaces for such systems as safety, telematics, and entertainment².

A typical simulation system extended with a re-configurable interface is diagrammed in Figure 3. The human interface experimenter automatically generates an application with the IVIS SDK. The application is downloaded to one or more display processing units. When the vehicle is driven, the display units execute the human interface application. State changes to the human interface are caused by simulation session data (i.e. proximity to other vehicles, speed, heading, etc). By integrating the IVIS SDK with your simulator, the full power of computer graphics (and other computer-controlled devices) can be tested in controlled experiments.



Figure 3. Re-configurable Interface Extension to a Simulation System

Displays are blended into the simulation cockpit to provide a genuine look and feel. Results of vehicle operator actions are reflected in the IVIS interfaces on the displays, as well as the simulated driving environment. An example of vehicle display systems is shown in Figure 4.



Figure 4. In-Vehicle Information System Displays

APPLICATION EXAMPLES

As a simple example, consider a collision-warning interface that uses concentric circles of varying colors. Such an interface application built with GUIDE is shown in Figure 5.



Figure 5. Collision warning interface example

The concentric circle images each have trigger conditions for visibility. Assume that the simulator can send a warning value from 0 to 50. As the value increases, the criticality of the warning is increased. For example, a value of 0 may means no cars ahead on the road and a value of 35 may mean the operator is approaching a vehicle at a dangerous rate of speed. A value greater than 40 may signal an eminent collision.

As the simulator sends the warning value an IVIS I/O stream converts it to the WARN data source. The interface designer chooses WARN as the data source used in each concentric circle. The condition for (a) is WARN > 10, (b) is WARN > 20, (c) is WARN < 30) and (d) is WARN > 40. The WARN varies with the vehicle speed and traffic conditions. Is this way, the simulated environment provides instant feedback to the operator.

Experimental interfaces such as that shown in Figure 5 are produced in a matter of minutes. They are quickly downloaded to a display-processing unit in the simulator vehicle cab. Experimental data is collected and the interface changed for the next set of experiments.

A second example demonstrates the enhancing of driver awareness through the use of blind-spot sensor data sources and graphical indicators. In Figure 1, graphical indicators are used to warn the driver of vehicles hidden in their left or right blind spot.



(a) No vehicles in blind spots

(b) Vehicle in left blind spot

(c) Vehicle in right blind spot

Figure 6. Enhanced blind spot awareness

Both of the examples shown in Figure 5 and Figure 6 are depicted during PC-hosted interface design and test. The application code may be downloaded to embedded platforms and run on an LCD display in the simulator cab.

CONCLUSIONS

Adding a software framework to the simulation environment enlarges the experimenter's interface options. The IVIS SDK is easily ported to most simulation systems. GUIDE allows rapid design of a wide range of display-based interfaces. Non-graphical devices are easily controlled using the IVIS communications and data warehouse interfaces.

Real time data from the simulated experience triggers changes in the interface. Just as a real vehicle provides data to IVIS applications, the simulation platform provides such data as road speed, engine temperature, proximity to front vehicle, etc. The designer sets data limits and conditions to control warning displays or devices.

Experiments are platform independent. A minimal cost and effort is associated with moving experimental interfaces from the desktop to small-scale simulation systems. Similarly, moving the experiments to a larger scale dynamic simulation or a test vehicle is quite cost effective.

REFERENCES

- 1. Machlab, Hass et al. Hardware Independent Software Framework for In-Vehicle Information Systems. SAE 2001 World Congress. March 5, 2001.
- 2. "IVIS is NEXIQ Technologies' powerful software framework that makes adding in-vehicle technologies quick and easy". NEXIQ Technologies, Inc. http://www.nexiq.com, October 2001.