Simulator design using open standards

Real Time Innovations looks at the challenges and benefits of adopting open standard middleware solutions in the development of high-performance COTS-based simulator systems, and shows how key issues of deterministic performance and connectivity can be addressed.

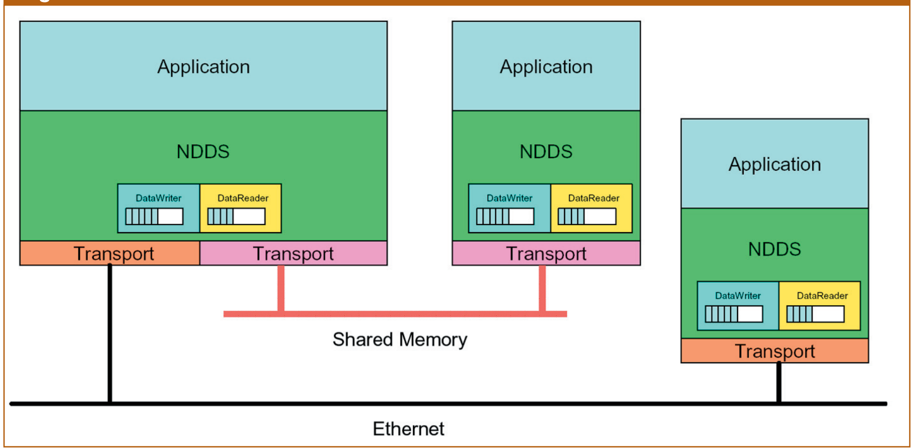
The aim of simulation is to produce and control animated images, sound reproduction, and device feedback in a manner as realistic and responsive as the real world, and chasing this ideal has constantly pushed the industry forward in many different ways. Individual simulators have adopted techniques such as multi-processor systems, high performance graphics cards and distributed sensors and actuators to approach the desired objective.

Increasingly such systems consist of many different high performance processing subsystem units that need to communicate in real-time. Although COTS-based open-standard hardware such as VME and Unix/PC systems has been commonly used, until recently the only way to provide software connectivity between applications has been with proprietary systems solutions. And as such systems get larger and more distributed, the performance issues of latency, determinism and system bottlenecks are becoming ever more important in maintaining the simulation experience. A further issue is the critical need to ensure investment in application software can be re-used effectively across multiple projects. The combination of these vital but difficult issues is driving the need for more formalised software structures and a growing move towards COTS middleware adoption.

These issues have to be tackled at two levels: both inside (between the system components of the simulator) and outside (for distributed simulator to simulator connectivity) such systems. High Level Architecture (HLA) has emerged as a widely adopted middleware standard for simulator-to-simulator connectivity and data sharing, especially in the defence system market. But until recently there was no viable open standard middleware solution able to address effectively the even more demanding requirements of real-time data distribution within the individual simulator.

This article will examine the problem in greater detail, and introduce an integrated design approach based on open standards middleware that addresses these issues at both levels.

Fig 1: The NDDS 4.0 architecture



One of the key issues in achieving the best performance in a simulator design is optimisation of the 'man-in-the-loop' function. This control loop has to respond in real time to the changing environment and operator input, possibly within a network of simulators. The demands on the network throughput will vary, from a dynamics model running possibly as fast as 1000Hz exchanging information with an I/O device at up to 100Hz, with a resultant potential network throughput requirement of 10ms per data packet; or from the dynamics model to an image display device that needs data input at anything up to 60-80Hz to match the screen refresh. If data appears outside of this time interval it will have to be ignored; a common response is to reduce the simulation fidelity to allow state recovery, a situation that simulator developers always seek to avoid.

As mentioned previously, the issue of simulator-to-simulator data sharing has already been addressed by the development of an open standard middleware for linking simulators promoted by the US Department of Defense and implemented by the Simulation Interoperability Standards Organisation (SISO). This is the High Level Architecture (HLA) and its application programming interface, which is already used by a number of simulator developers to connect systems together. HLA is a publish/subscribe architecture where elements publish data onto the bus to be picked up by other units that subscribe to that data, commonly referred to as 'federated data'. This allows a system to be distributed, avoiding the bottlenecks of a client-server architecture and allowing the system to be more easily scalable.

The HLA Run Time Infrastructure (HLA-RTI) is most commonly used to link simulator systems together in a wide area network,

The National Automotive Center (NAC) recently developed and demonstrated a distributed Vehicle Proving Ground (VPG) simulator. This project was a cooperative effort between the University of Iowa National Advanced Driving Simulator and Simulation Center (NADS-SC) and the US Army TACOM-TARDEC Ground Vehicle Simulation Laboratory (GVSL). Researchers connected several high-fidelity simulators to demonstrate how US Army vehicles and components can be tested and evaluated using VPG technology. Researchers chose the RTI's NDDS real-time middleware to tie together the VPG simulator sites. *"As we evaluated different networking middleware options, including CORBA, DCOM and HLA, it became clear that NDDS was not only the easiest to use, but also the most effective at controlling update rates, message ordering, and message latencies."*
– Dr. Yiannis Papelis, Chief Technical Officer of NADS-SC

and in some cases it is even being seen as a way to link the processing elements within individual simulators.

This approach has the advantage of avoiding having to translate the data from one format to another. Unfortunately, the HLA-RTI wasn't designed to provide the speed and detailed control of real-time performance required by systems that need consistently low latencies and deterministic responses.

As a result, simulator developers often end up creating proprietary protocols and methodologies to enable data distribution with adequate real-time performance within the simulator.

However, there is another open standard that also uses a publish/subscribe architecture that provides a much better fit for our real-time requirements. Data Distribution Service (DDS) is an open standard data oriented middleware optimised for hard real time systems with the low latency and quality of service capabilities to provide the required speed and level of control of real-time performance built in. This commercial off the shelf (COTS) middleware is already widely used in aerospace, military and industrial applications (see DDS box). Better still, its similar structure allows it to work very well alongside, and in co-operation with, the existing HLA standard.

Like HLA, DDS uses a publish/subscribe model where data dissemination between producers and consumers may be from one-to-one, one-to-many, many-to-one, or many-to-many. The communication model is decentralised, with publishers and subscribers loosely coupled and having no knowledge of each other. This means that publishers and subscribers can join and leave dynamically, providing an ideal platform for a flexible and scalable simulation system architecture. This data-centric development approach, enabled through a standard API, also enables modularisation of simulator development and thus the potential for significant application code re-use.

DDS builds on the definition of HLA to add support for object modelling, and ownership management. It addresses a number of performance-related issues not dealt with by HLA, such as a rich set Quality of Service (QoS) policies, a strongly typed data model, and support for state propagation including coherent and ordered data distribution. In particular, the QoS capability of DDS allows the designers to maintain levels of priority for data ensuring a fine level of control which ensures the minimum latency requirements are met across the distributed system.

Real Time Innovations (RTI) has implemented an enhanced version of DDS, designed to maximise benefits of the deterministic data environment provided by the standard. The latest version, NDDS4.0, provides pluggable transports that support any media, from wireless to switched fabrics with programmable parameters, programmable Quality of Service (QoS) and customisable data types.

The pluggable transports are unique to RTI's implementation of the DDS standard and when combined with the QoS mechanisms of DDS it provides performance "tuneability" of data paths to the underlying

What is DDS?

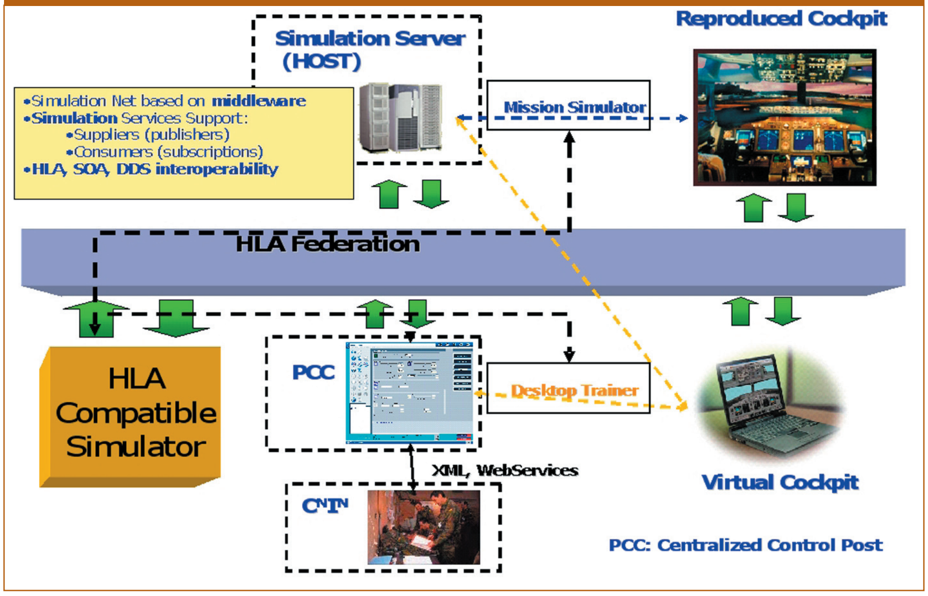
The Data Distribution Service (DDS) is a newly adopted open specification from the Object Management Group (OMG), a group of around 800 members, for data-centric publish/subscribe communications in real time systems. These include applications in aerospace and defence, distributed simulation, industrial automation, distributed control, robotics, telecom, and networked consumer electronics.

The publish-subscribe model connects anonymous information producers (publishers) with information consumers (subscribers). The overall distributed application is composed of processes, each running in a separate address space and even on different computers. The API and Quality of Service (QoS) are chosen to balance predictable real-time behaviour and implementation efficiency/performance.

The specification provides a platform independent model (PIM) that can then be mapped into a variety of platform specific models (PSMs) and programming languages.

DDS draws upon common practice in existing publish/subscribe architectures including HLA, OMG event notification service, Java Messaging Service (JMS), and experience with Real-Time Innovations (RTI's) NDDS product. Many enhancements have been specifically designed to provide the higher latency and determinism required by real time distributed systems.

Fig 2: Nextel's Simware uses DDS in the kernel to combine HLA-compatible simulators in a common environment



simulator connectivity fabric. This also means that any transport medium can be used, from the widely used TCP/IP protocols to a switched fabric for more high performance applications.

NDDS4.0 also uses direct end-to-end messaging, which eliminates context-switch delays. Combined with the support for message prioritization, this ensures a predictable and low level of data latency, while pre-allocated memory prevents allocation and fragmentation delays that would otherwise contribute to a lack of deterministic operation in the responsiveness of the subsystems.

Enhancements to the memory handling further enhance the deterministic performance of NDDS, as there are no shared threads or memory that requires locking. It also provides dedicated buffers to prevent the different processes having corrupted shared memory.

Another advantage of NDDS 4.0 for the simulator developer is that there is no server process to crash, and applications do not share address space through the middleware, so operating system processes can be isolated and this fully protects all applications. This open-standards approach is already in use by several simulator developers, including CAE for flight simulators. NDDS is used to link subsystems via high speed IEE1394 Firewire links in its SimXXI product line. It opted for NDDS as an open standard available off the shelf, rather than having to develop a proprietary technology.

Simulator developer Nextel Engineering Systems in Madrid, Spain, is also using NDDS to link together the different elements within its simulator architecture. It evaluated all the different technologies available, seeking an open standards solution which did not compromise performance or reduce scalability for its Simware kernel, to link together both new and legacy simulation elements – see box below.

The performance oriented approach of NDDS was key to the decision by companies such as Nextel and CAE to provide the robust, deterministic and controllable quality of service that is vital for high performance simulator system designs based on commercially available open standards middleware architecture.

Force Technology are leaders in multi ship simulators. Their latest marine simulator development provides a complete environment for training tug boat captains to maneuver large vessels such as oil and gas tankers into restricted spaces. This demanding application has to simulate in real-time the dynamic configuration of up to four tugs and the resulting variable forces on the tow lines when handling vessels. RTI NDDS real-time middleware was selected by Force Technology because it met the critical performance requirements for interconnecting distributed ship simulators.

Nextel chooses DDS for simulator kernel

Nextel Engineering Systems in Spain wanted to develop a new generation of simulation and information systems that used a common architecture and set of tools but also integrated several pre-existing technologies.

A new division, Simware, was set up to develop the new architecture using existing technologies and standards such as Web distributed architectures and XML interoperability, with Web system integration based on the Network Centric Warfare concept.

It also had to be services oriented rather than technology oriented to allow for development reusability and HLA compatible with a publish/subscribe approach.

The simulation modules are based on an engineering development environment to provide for modular design and development regardless of manual or automatic software coding, and a publish/subscribe Real Time Communication Protocol had to be used to provide maximum flexibility for distributed simulation. Nextel evaluated all the products on the market that could supply efficient performance of a publish/subscribe protocol for HLA interoperability. This was vital to provide interoperability for the common architecture, and this selection has a high impact on simulation performance especially in frequency refreshing.

As a result, the DDS software from Real Time Innovations - NDDS - was integrated into the Simware kernel to provide a common interface for all simulation components and information systems interconnection.

Nextel sees a key advantage in NDDS providing the capability to manage data instead of objects, and the publish/subscribe protocol that provides the capability of having systems and subsystems publishing and subscribing data in an uncoupled way, providing (potentially) a system with a high-level of redundancy.

The custom data types avoid having to employ a fixed set of data, which would otherwise increase the likelihood of network saturation and degrade the latency of the system.

Nextel also appreciated the multicasting transmission, UDP transport mode and the NDDS quality services for error and failure communications control and monitoring.

The common publish/subscribe model shared between DDS and HLA allows a more consistent interface for the interoperability of different types of simulator such as helicopters, planes, missiles and others, and from different, HLA-compatible suppliers.

As a result, Nextel's Simware has been selected by the Spanish Ministry of Defence to be included in the COINCIDENTE Program as the architecture and tools standard to be considered for future Defence projects. Simware is being used as the interface between other HLA simulators working as objects in a common simulation environment.

As of March 15, 2006 RTI has changed the name

***/NDDS/* to */RTI Data Distribution Service/*.**

The product itself remains the same.