

# AI & SIMULATION

A. Martin Wildberger

*Electric Power Research Institute*

The subject of this special issue of *SIMULATION*, the DoD High Level Architecture (HLA), like all standards, will ultimately be both liberating and confining. Its overarching structure, described by federation rules, formal interface specifications, and object model templates, frees the simulation designer from the pressure to repeatedly invent and re-invent a way to think about simulation as a process each time that a new problem, situation or phenomenon is to be simulated. Even if that pressure arises mainly from a personal desire to “do it better this time,” the requirement to follow the HLA forces the designer to put emphasis where it belongs—on the model and its implementation, rather than on how to build simulations in general. This change of emphasis will work to make our models more useful and authentic, and to make their implementation more complete and accurate, just as the rules of a game force us to channel our innovations within that context, inventing new strategies for play and improving the skill of the players.

On the other hand, despite a laudable effort to make the HLA extremely abstract and to put in place a mechanism for its continued evolution, there will come a time when it will be found too confining and a radically new paradigm will be substituted. In fact, that very high degree of abstraction may well sow the seeds for its eventual overthrow. Thinking abstractly requires a rigorous discipline that few of us can maintain for very long. In order to understand the abstraction, we rapidly descend to examples, which are far easier to grasp. But what we gain in concreteness, we lose in generality. When we reason from the examples, it is easy to mistake accident for essence.

In order to explain and demonstrate the HLA, it was first necessary to implement it, and those early implementations have already achieved the status of dogma derived ineluctably from the HLA's Ten Rules. (I cannot avoid wondering if there was a conscious or unconscious patterning after the Decalogue received by Moses.) The supporting software, and especially the RTI (Run Time Infrastructure) implementation, available freely from the DMSO (Defense Modeling and Simulation Office), are a set of handy—possibly even brilliant—tools, but they are not as generic as the HLA itself. Nevertheless, those implementations are the means by which the HLA is being introduced to the simulation community, and all of us are learning the HLA in the context of those implementations. Despite DMSO's repeated insistence that HLA is a

framework and not an implementation, that distinction is easily lost in practice.

But if the existing implementations are spread worldwide and used repeatedly, doesn't that meet the chief goal of HLA: to enhance the *reusability* and *interoperability* of simulation components? Reusability means that component simulation models can be re-used in different simulation scenarios and applications. Interoperability means that the reusable component simulations can be combined with other components on distributed computing platforms of different types without the need for re-coding. Reusability has been the Holy Grail of Software Engineering from its beginning. Interoperability makes reusability far more practical by extending its possibilities beyond one, or a very few, CPUs. But interoperability became a necessity only when we began to produce distributed simulations running on heterogeneous processors. Even if components were never to be reused, they had to be able to migrate between multiple diverse processors networked together. The HLA achieves reusability and interoperability through its Object Model Template (OMT) that standardizes how to specify all objects and interactions that are managed by a federate and are visible outside the federate.

Some of the applications in this special issue illustrate the possibilities for reuse and cross-processor operation that the HLA opens up. In their development of the Combat Trauma Patient Simulation (CTPS), Petty and Windyga used the HLA framework and ownership transfer services to integrate and make interoperable a set of existing commercial and military simulations, each one specialized for a different part of the process. Shockley, Parsons, and Morgenthaler describe work on a High Level Architecture-compliant version of the Virtual Command Post (VCP). Their goal is not only to develop a VCP that is compliant with basic HLA requirements as a stand-alone federation, but also to develop a set of VCP functional elements that can interact as separate HLA federates in other federations. They hope to make this VCP implementation the federate of choice for all HLA federations that require Command, Control, Communications, Computers, and Intelligence (C4I) functions. They also expect their individual VCP components—e.g., audio, video, and commander federates—to be used as separate functional elements in other HLA federations. Zeigler, Hall and Sarjoughian tell how they exploited the HLA framework to overcome the

---

Please direct communications to Martin Wildberger, EPRI, 3412 Hillview Ave., P.O. Box 10412, Palo Alto, CA 94303.  
Phone: (650) 855-1043; Fax: (650) 855-2287; E-mail: mwildber@epri.com

“not invented here” objections to reuse that are prevalent even within a single corporation.

Measured, professional evolution of the HLA has been, and continues to be, the stated goal of the DoD-wide Architecture Management Group (AMG). But how does an abstract architecture “evolve”—i.e., unfold and develop from what was already implied in its original form? It’s obvious that many of the features of the HLA grew out of hard lessons learned in designing and running distributed simulations. The separation of data and architecture that is basic to the HLA started as something the best programmers did unconsciously and later became a recognized professional practice that was more often advocated than accomplished. At a more detailed level, HLA’s communication management by subscription and publication seems like a further generalization of the message broadcasting with receiver selection that I used in a distributed naval battle simulation circa 1983.

It is not as obvious how the HLA should evolve in the future. It is easy to criticize aspects of the current specifications. For instance, the strict object-class hierarchy in the FOM (Federation Object Model) may be too confining. It may need to evolve to include more complex relations, even all Boolean functions. Meanwhile, the AMG is supporting HLA standardization beyond DoD through the Object Management Group, NATO, and the IEEE. One paper in this special issue, by Schulze and Klein, addresses the problems that the HLA faces in becoming a simulator interoperability

standard in the civil world and specifically in the transportation sector. Outside of DoD, the HLA cannot become the standard by fiat. Instead, it must do so by showing real cost benefit. But outside of DoD, can the HLA resist adulteration and splintering? Consider the fate of UNIX.

If the HLA is to survive within DoD and prevail outside, it must at least evolve to meet an ever changing environment, new breakthroughs in hardware development, new understanding of the software development process, and new tools for system design in general. In the long run, even this may not be enough. What does it mean to use an abstract architecture that guarantees the independence of hardware and software when that distinction is made meaningless by quantum or biological computing? But long before that issue needs to be faced, new implementations will be held up as HLA exemplars and the AMG will have to expand the HLA to encompass them without particularizing it to the exclusion of yet newer implementations. Perhaps the evolution of biological species is not too far-fetched a model to follow. Federates and even federations die, but the HLA lives on.

#### To look further:

- HLA’s Home Page is: <http://hla.dmsomil/>.
- An excellent set of educational and training material on the HLA is available from the McLeod Institute site at California State University, Chico: [www.ecst.csuchico.edu/~mcleod](http://www.ecst.csuchico.edu/~mcleod). ■