

RAPID SYSTEM VISUAL PROTOTYPING (RSVP)

Recent advances in computing and multimedia technologies are making practical the integration of powerful high-level tools into domain-specific development environments. These open-architecture shells enable the systems engineer to conceptualize, visualize, and *rapidly prototype* interdisciplinary systems for a wide range of military and nondefense applications.

THE NEED

Systems problems increasingly require complex, multiple disciplines for their optimal solutions. Designers must therefore build on and integrate the work of others into the solution of their systems problems. Furthermore, sponsors, systems developers, and systems users need more effective ways than static paper specifications to communicate and collectively contribute to dynamic problem solutions.

A major inhibitor has been that the lowest level of abstraction on the systems side has generally been higher than the highest level on the technology side. For example, someone interested in medical, missile, or manufacturing applications may not be well versed in the intricacies of microprogramming, multichip modules, or multimedia. Thus systems designers often find it difficult to take full advantage of enabling technologies.

WHAT IS RSVP?

RSVP is a dynamic visual design environment to rapidly model and prototype real-world applications. It uses a family of visualization tools to increase productivity and dramatically accelerate prototype development.

A rapid system prototyping environment involves a family of integrated computing and multimedia-based tools that can quickly translate user requirements into a dynamic visual representation and accelerate concept formulation, architectural trade-offs, and systems implementation.

Human intelligence has been defined in terms of our capacity to develop tools—especially enabling, higher-level tools. Design tools allowing the designer to work at a high level of abstraction are now commonplace. These can help provide a friendly and user-familiar interface. In addition, the designer can work at a convenient module level rather than with cumbersome and error-prone detailed code. Such tools include graphical, computational, control, and communications modules. These, in turn, include a wide variety of functional packages such as multimedia databases, virtual instrumentation, and automatic code generation routines.

However, until recently, the systems designer was forced to master and attempt to piece together an array of disparate modeling and analysis tools. No single shell or executive had been developed to integrate the

necessary tools into a coherent, unified environment. What is required is an interactive open architecture formatted in a familiar configuration that can seamlessly import, interface, and integrate a broad range of external software packages. Users must be able to transparently invoke the appropriate functions and features as needed for their specific system application. The Laboratory's RSVP initiative is designed to do just that, by providing a family of domain-specific, open-architecture shells focused on particular areas of APL interest.

WHAT CAN RSVP DO?

RSVP's multimedia capabilities, open architecture, system interoperability, and adaptability to many user applications make it a powerful environment for dynamically assessing user needs, system functions, and prototype implementation options. When coupled with an open-architecture integrating shell or executive, RSVP technology can import programs, data, and multimedia inputs from many sources.

The designer can thus take advantage of past programming efforts, using existing validated software, and integrate diverse programs and information. Computer programs, data, and multimedia elements (e.g., video, pictures, and sound) may be loaded locally or downloaded from remote sites via networks. All this can be done at a high level of abstraction (i.e., at the module rather than at the detailed programming level). Much of the systems prototyping can be performed at the interface or boundary level, although software hooks are available to probe down to the code where needed. By transparently bridging the gap between systems concepts and dynamic prototypes, RSVP can significantly reduce the time required for optimized design.

Since these implementations can be rapidly modified or zoomed to varying levels of abstraction, sponsors, users, and prototype developers can establish a valuable dialogue. "Hot links" enable changes at one level of integration to be reflected dynamically at other appropriate levels. Figure 1 illustrates the RSVP-based systems approach and life-cycle role, from concept to system upgrades. All stakeholders can interactively and most effectively contribute to every phase of the program.

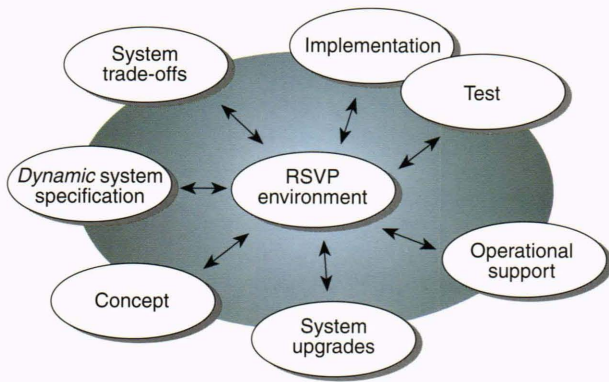


Figure 1. RSVP-based systems approach.

External hardware can be gradually phased in, replacing virtual and simulated modules, thereby facilitating dynamic test and integration. RSVP tools can provide valuable operational support when interfaced to the real-world environment. The virtual instrumentation and real-time modeling capabilities can help in assessing and formulating potential system enhancements and upgrades.

RSVP facilitates

- Concept visualization
- Speed of prototyping through boundary-level simulation
- Development of user-familiar interfaces and metaphors
- “Dynamic” system specification
- Requirements to real-world validation
- End-user participation in system concept formulation
- Fly-before-buy concept exploration
- Design by interdisciplinary teams
- Dynamic high-level modeling
- Phased hardware integration and test

The enablers that make these capabilities possible include the advent of low-cost, high-speed computation and networking, multimedia technologies, high-level design tools, and open-architecture integrating shells, which can be readily adapted to specific application environments and user interfaces. The key integrated features of RSVP tools are

- Multimedia capabilities
- Multienvironment, open architecture
- Multiple package integration
- Ability to import data from different sources

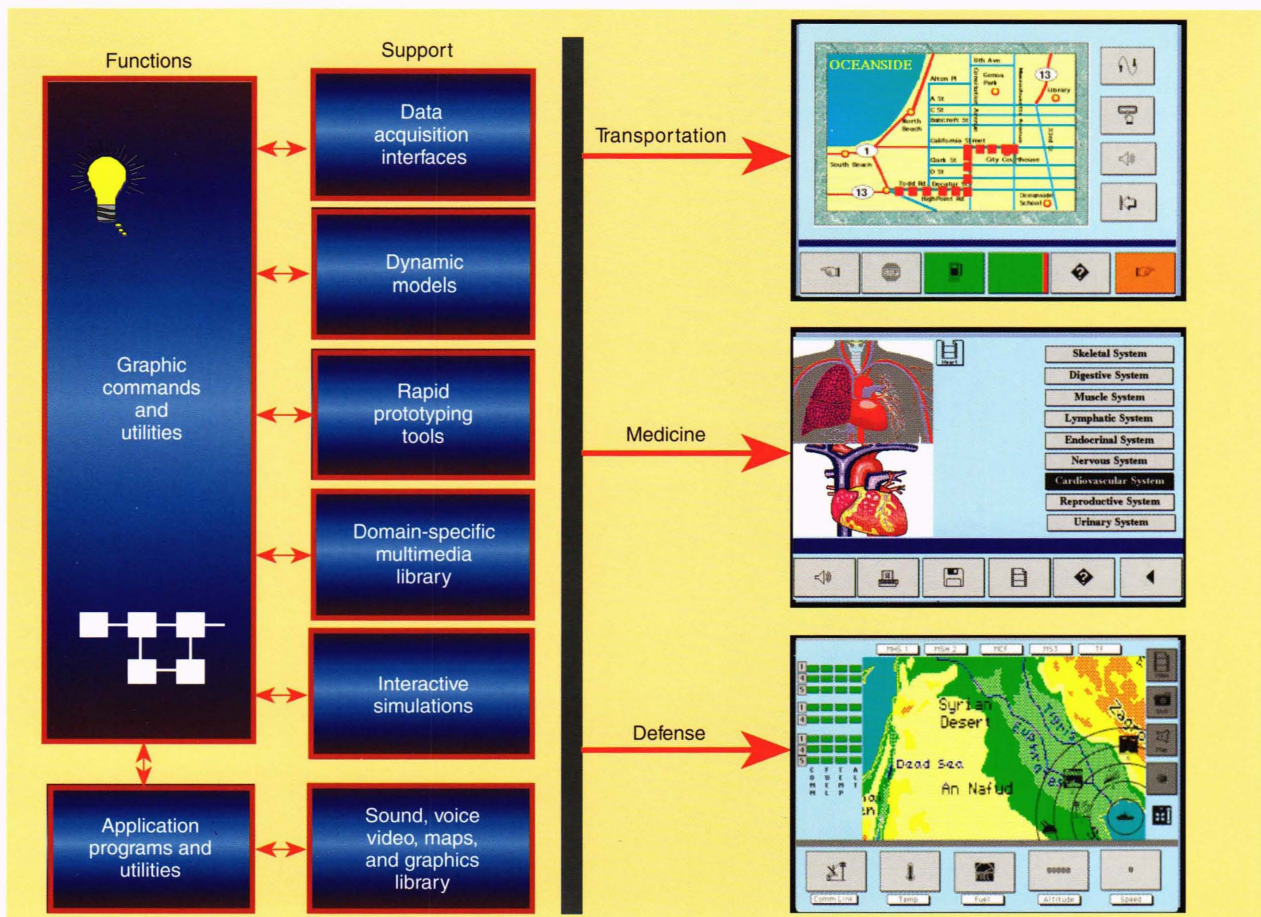


Figure 2. RSVP functions and typical applications.

- User-defined interfaces
- Object-oriented implementation
- Multiple, high-level design formats (i.e., from equations, flowcharts, block diagrams, etc.)
- Access to system functions and controls without coding
- Multilanguage capability (Fortran, Basic, C, Pascal, etc.)
- System interoperability

Rapid system prototyping facilitates quick reaction and responsiveness to user and sponsor needs. It permits the cost-effective exploration of innovative concepts and system trade-offs. RSVP tools may be zoomed out to provide a system perspective or zoomed in to focus on any specific implementation detail.

RSVP environments are particularly useful in the following contexts:

- Virtual and actual prototypes
- "What if" strategies
- Military tactics and tracking
- Networked simulation
- System evaluations
- Manufacturing data acquisition and control
- Transportation (commercial vehicle operations, smart sensors, navigation, emergency response, networks modeling, etc.)
- Health care (hospital and patient information systems, education, remote care, etc.)
- Artificial intelligence/heuristic simulations

Specific examples may include traffic simulation, unmanned autonomous vehicles, strike missions, missile tracking, joint power projection, and environmental cleanup. Figure 2 shows typical rapid prototyping examples ranging from transportation to medicine to defense environments. Each is driven by a high-level, domain-specific open-architecture shell or executive. The executive can import, invoke, and display system and support functions in a framework familiar to the designer and the end user. Although the executive can be configured for specific environments, it uses standard commercial modules and is therefore easily kept current.

FUTURE DIRECTIONS

Because rapid system prototyping allows quick response as well as enhanced realism and user interactivity, it can provide a powerful competitive advantage. Since RSVP can dynamically adjust the level of abstraction to bridge concept formulation, system development, and detailed implementation, it promises to significantly impact the field of interdisciplinary systems design.

Initiatives at APL in 1994 included seven hands-on workshops and an RSVP symposium cosponsored with the IEEE, which was telecast to 100 institutions worldwide on Internet video. Applications ranged from virtual instrumentation in underwater environments to missiles and from transportation to elementary through secondary school science education. The Laboratory will make a major commitment in 1995—including research and development, dissemination, and training—to ensure the effective utilization of this enabling technology throughout APL.

THE AUTHOR



PAUL L. HAZAN is Assistant to the Director for Advanced Computer Technology at APL. He received his B.Sc. (honors) and A.R.C.S.T. in electrical engineering from The Royal College of Science and Technology in the United Kingdom. Before joining APL in 1975, he was Technical Director of the Singer Company, Link Division. His current technical interests include design automation, very large scale integration, computer visualization, computer/video multimedia technologies, and rapid system prototyping. Mr. Hazan has served on the Board of Governors

of the IEEE Computer Society as Director for Microcomputers, and was founding chairman of the Technical Committee on Personal Computing.