Complex parallel and distributed systems must be tuned to achieve good performance. By monitoring the behavior of resources, a designer may identify bottlenecks and non-optimal design solutions and then modify some aspect of the system's design to improve performance. Typically, online resource monitoring and analysis are necessary for making decisions (automatically or with human assistance) such as resource reallocation for load-balancing purposes. If a human is in charge of decision-making, he/she faces the problem of diversity of resources and large volumes of performance information.

Visualization has been widely accepted as a means to deal with large-scale data sets, and to monitor dynamic and multivariate behavior of systems. Many visualization techniques have been developed and applied to complex systems. However, most of them have focused on domain-specific resources or levels in the system and have been developed in isolation. In contrast, parallel/distributed applications are executing in environments with greater heterogeneity and adaptivity, for example, a computational grid.

URV represents a new paradigm for constructing visualizations and for monitoring and analyzing complex grid systems. URV addresses the following needs in visualization of computational grid: (1) the need for applying standard visualization services; (2) the need for composing system-level views; and (3) the need for sharing visualization design knowledge. URV uses metadata technology such as XML to describe a resource and its visualization formally, and rigorous software-component technology to bundle resources with custom visualizations.

URV supports uniformity by describing resource and visualization components with XML-based formal descriptors. A descriptor provides a representation of the grid resource at a level of detail necessary for its visualization, hiding other details. This facility helps developers to design uniform interfaces for accessing and manipulating heterogeneous resources. Composition refers to support for multi-resource visualization. With this composition facility, the designer can construct a computational system that comprises multiple resources and view system-level performance across resources. The simplest approach to visualization composition is to aggregate the stock visualizations from each constituent resource, placing them side-by-side. However, we have to exercise caution with this simple method of composition since the separate visualizations may not be synchronized, which may cause the user to draw incorrect correlative inferences about the performance of the system. Composition requires exploiting spatial/temporal synchronization among separate visualizations and abstraction and layering in visualization. Reusability is achieved with URV by enabling descriptor registry and discovery. Using the directory service of grid middleware (e.g., Globus toolkit), users can retrieve existing visualizations and by composing them, can construct a new visualization. Distinctive services of URV include: (1) metadata service to define visualizations for any resource in the system, ranging from hardware devices to application software modules; (2) resource-visualization connection service to bind resource(s) to a visualization; and (3) visualization composition service to construct a new visualization based on multi-level, composable, reusable, and distributed components. The uniformity and composability of URV have the potential to lead to dramatic improvements and advances in grid resource visualization.