A grid application uses various resources across multiple levels and platforms during execution. To observe the performance problem of such an application, the integration of performance data is necessary. And system-level performance visualizations that represent data correlations are very useful for presenting this integration visually.

Performance visualization tools, however, often are specific to a particular resource at a certain level of the system, possibly with fixed views. Thus, they limit a user's ability to observe a performance problem associated with dynamic resources across platforms. To address this limitation, we are developing a new monitoring framework, Uniform Resource Visualization (URV), focusing on dynamic integration of performance data into system-level performance visualizations. The goal of this project is to provide users the tools to construct system-level views dynamically in grid environments with sharable visualization knowledge.

The novel approaches of this research project reside in the performance visualization knowledge (PVK), its query service, and composition service.

PVK describes which visual design works best with which monitoring goals, which combinations of visual design provide the most effective performance view, and how visual designs can be composed to create a system-level view. To represent PVK, we use a RDF-based ontology representation, such as DAML+OIL.

A query service enables knowledge acquisition. For instance, the query service is able to process and answer a query like ‘What visual designs are appropriate for visualizing metric X on resource Y?’. It is a web service, which allows users or software to make a query via a designated web interface or a SOAP protocol.

A composition service systemizes a construction process of a system-level performance view. We have identified different types of composition, ranging from simple to complex: placing two views in the same window and providing view synchronization; merging two or more separate data streams of the same type into a single view; merging two or more data streams of different types into a view connected with one of the streams; and deriving a new view that connects to two or more data streams. Our approach is to design PVK in a way that supports the validation of these different types of composition. For instance, PVK should be able to validate the composition of event types (e.g., whether a CPU load is appropriate to be presented with network hops in a single system-level view). Based on the validated composition, a composition service creates a final system-level view.
This research is the first effort to describe performance visualization knowledge with metadata that enables reuse and processing of the performance visualizations.

**Poster Presentation**
During the poster session, we plan to present two items: a regular poster and demonstration. The poster will summarize the research, including a detailed diagram showing a visualization composition process. For demonstration, we plan to present a front-end URV system installed on a laptop. The demo will show PVK, a query service with sample queries, a composition service, and several system-level performance views generated by the system. We will need a 4’x 4’ poster area with an electrical outlet and Internet connection.