

## **XCHANGE: High Performance Data Morphing in Distributed Applications**

Jay Lofstead and Karsten Schwan

College of Computing, Georgia Institute of Technology

{lofstead, schwan}@cc.gatech.edu

### **Abstract**

In both high performance and enterprise applications, it is common for components to generate, exchange, process, and store or display large volumes of data. A key problem for such exchanges is the mismatches of data being generated with the data required by communicating software components. Such mismatches arise from natural differences in the data representations used by different components and for acquisition or display. They are due to the need to customize or personalize data for certain devices or end users. This paper describes automated methods and associated generative tools for correcting such mismatches using overlay networks to connect data producers with consumers. These data morphing methods automatically generate data transformation codes from declarative specifications ‘just in time’ and ‘in network’, i.e., when and where needed. Runtime code generation takes into account the current nature of the data being generated, the current needs of data sinks, and the current resources available in the overlay connecting sources to sinks. In addition, it can consider the shared requirements of multiple consumers to reduce redundant data transmissions and transformations. Dynamic, ‘in network’ data morphing is realized with the XCHANGE toolset, which can generate transformation codes both from high-level declarative specifications like MathML and with methods that automatically correct for type mismatches. XCHANGE is integrated with ECho, the high performance event-based middleware and it is applied to both high performance and enterprise applications. Results show that for remote visualization of the data created by a high performance simulation, dynamic data morphing can better match the large data volumes being exchanged to available network resources. For an enterprise application in the healthcare domain, dynamic data morphing demonstrates the importance of generating code to reduce server loads.

**Keywords:** Data Transformation, Dynamic Data Morphing, Dynamic Code Generation, Overlay Networks, XML, Declarative Specification, Distributed and Enterprise Applications, High Performance Data Exchange.