

# A Reflective Runtime Environment for Dynamic Adaptation of Streaming Media

Muhammad A. Khan                      Stefan Fischer

*Institute of Operating Systems and Computer Networks*

*Technical University of Braunschweig*

*Muehlenpfordtstr. 23*

*38106 Braunschweig*

*Germany*

*{khan | fischer@ibr.cs.tu-bs.de}*

## Abstract

In order to provide QoS to applications in wireless networks, there exist a number of application adaptation schemes implemented at higher layers of the protocol stack which rely on stubs, skeletons or interfaces, implemented in some quality description language. A commonality found in the existing schemes is the need to specify QoS requirements. It is, however, very difficult for a human being to have exact knowledge of, and to precisely interpret the QoS requirements for an application, because, not only the resource availability changes with time, resource requirements of an application are also usually dynamic. Especially, in the cases where applications are designed by different people, and everyone has a different view of the system, adaptation specification may lead to conflicts. Also, the size of application code is relatively larger when an application is explicitly written to take advantage of different resource levels by switching across different execution modes (modal adaptations).

We aimed at developing a reflective runtime environment (RRE), which is independent of any QoS specification API, needs no stubs, skeletons or interfaces and does not require the application designer even to be aware of the notion of QoS. In this poster, we briefly describe our efforts to realize the concept of 'self-adaptive multimedia' by making use of reflection and aspect oriented programming to enable a multimedia application to change itself and adjust to the changing system state to achieve certain level of QoS, without any user involvement regarding the QoS specification. Reflection hooks are selectively inserted in the application code during the pre-processing phase and aspects are added dynamically at runtime, using an implicit weaving mechanism, depending upon the inbuilt adaptation policy. The core concept in RREs design is that at any specific time, the amount of resources available in the system are sufficient to execute a certain kind of application with a specific state. For example, in case of a streaming media client, at any particular time the resources may be sufficient to receive jpeg compressed data in CIF size with 29 fps, while at another instant, this may not be possible, but reception of h263 compressed stream with the same size and frame rate may be possible. Our runtime environment keeps monitoring the events of interest generated by an executing application and upon reception of such an event, reconstructs (alters) the executing application by combining different components (which are mostly media processing elements, like codecs, appearing as plug-ins with different processing characteristics), thus making it an application suited to execute with that much amount of resources.

Since, at any given time only those components of an application are present which are absolutely necessary, the application is spatially optimized for memory constrained devices. As QoS specification is not needed, adaptation becomes purely a property of the reflective runtime environment (thus making any a/v application adaptive at runtime, which was non-adaptive at compile time). We tested our approach on an RTP client and server applications available from Sun Microsystems, written in the Java Media Framework. The results thus obtained showed sufficient improvement in user perceived QoS. The architecture proposed by us can be implemented in any language, that provides support for reflection and component based software development. Since the implementation is at application layer, it is independent of any underlying protocol and is scalable.