We propose the Component Performance Assurance Solutions (COMPAS) framework that can help developers of distributed component-oriented applications find and correct performance issues during as well as after development. The motivation for our work on this framework was derived from the following considerations:
- Performance can be critical for large-scale component oriented applications.
- A poor architecture, a bad choice of Commercial-Off-The-Shelf (COTS) components or a combination of both can prevent achieving the application performance goals.
- Performance problems are more often caused by bad design rather than bad implementations.
- Current performance modelling and prediction techniques, mainly require developers to build performance models based on assumptions about the run-time characteristics of the software application, which often is difficult to do, especially for complex middleware-based applications.
- Often, performance is “a function of the frequency and nature of inter-component communication, in addition to the performance characteristics of the components themselves” [P.C. Clements].

To address these issues, the presented framework is structured into three main functional parts or modules, that are interrelated:
- Monitoring: obtains real-time performance information from a running application without interfering with the application code or the application run-time infrastructure (i.e. the application server implementation). We are currently concerned with monitoring of EJB systems only.
- Modelling: creates UML models of the target application using information from the monitoring module. The models are augmented with performance indicators and can be presented at different abstraction levels to improve the understanding of the application from a performance perspective.
- Performance Prediction: the generated models of the application are simulated with different workloads (e.g. corresponding to different business scenarios); simulation results can be used to identify design problems or poor performing COTS components.

There is a logical feedback loop connecting the monitoring and modelling modules. It refines the monitoring process by focusing the instrumentation on those parts of the system where the performance problems originate.

The intent of the framework presented is not to suggest a development process that prevents the occurrence of performance issues in the design, but rather to enable early discovery of such issues and suggest corrections.

Models are represented in UML, which many enterprise-scale application developers are familiar with. The use of Model Driven Architecture and Enterprise Distributed Object Computing (EDOC) concepts facilitates navigation between different abstraction layers. The top-level models are represented using a technology independent profile, the Enterprise Collaboration Architecture from EDOC, in order to benefit from a standardized form of representation for business modelling concepts. Lower level models are represented using UML.
profiles such as the UML Profile for EJB which provide means to illustrate technology specific details. Regardless of the level of abstraction, each model is augmented with performance information presented using the UML Profile for Schedulability, Performance, and Time Specification. The Performance Prediction Module uses runnable versions of the generated modules and simulates them with different workloads as inputs displaying performance information in the same manner as in the modelling phase.

ADDITIONAL INFORMATION: http://www.eeng.dcu.ie/~mosa/projects/compas.html