

## Automatic Generation of Workflow Execution Provenance

Roger Barga, Microsoft, [barga@microsoft.com](mailto:barga@microsoft.com)

This presentation describes our efforts to define an extensible model for execution provenance related to workflow enactment and the implementation of mechanisms that automatically generate this provenance data from a commercial workflow processing engine.

We had three primary research objectives in our project. The first objective was to explore the nature of execution provenance, ranging from specific steps taken during workflow enactment to produce a result, to a record of services invoked during execution and data and parameters used, to deviations from the prescribed workflow model. Since the intention is that this provenance is a machine readable artifact, we have designed a machine readable XML format that reflects our model. The second objective is to demonstrate the practicality of generating this provenance data automatically from a workflow enactment engine. Dynamically creating a workflow execution trace, in particular one that can be re-executed is a challenge that depends on the capabilities of the runtime in which the workflow was executed. We identify the capabilities we leveraged and features we would have preferred to have access to. The third objective is to define the management and reasoning that can be performed over a workflow provenance trace, and to construct algorithms to reason over provenance data.

We believe a Problem Solving Environment, built around a commercial workflow management system can serve as a productive platform for scientists to define and manage experiments, and provide access to high performance computing resources. Our implementation is based on Microsoft Windows Workflow Foundation, which is an extensible framework and is part of the upcoming Microsoft's next generation development Framework, WinFX [3]. The workflow in Windows Workflow Foundation is composed from a set of activities, compiled to a .NET assembly. It can be executed under the Common Language Runtime (CLR) in a variety of container processes. Thus a serialised workflow execution with all its parameter settings and values is a provenance record for the result arising from it, but also itself needs provenance information (which workflow specification was it instantiated from, who enacted it, was it interactively steered, and if so how). We identify future extensions we are making to our model and considerations for our current implementation.