

RAIDE Research Demo Script

Shang-Wen Cheng
 Carnegie Mellon University
 zensoul@cs.cmu.edu

The research demo starts with a contextual overview of the Rainbow approach, introducing the components of the adaptation cycle and their customization points, as illustrated in Figure 1. Table 1 enumerates the artifacts to customize the framework.

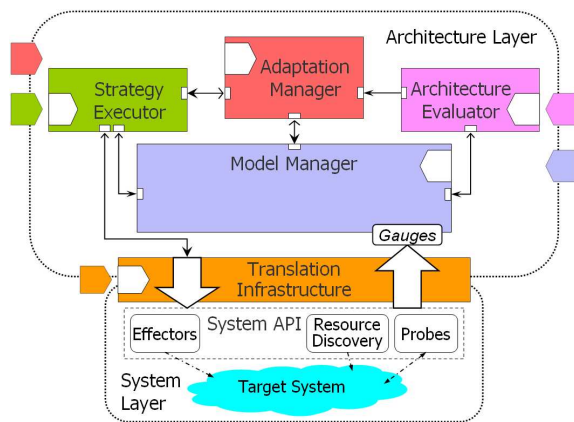


Figure 1. The Rainbow framework

Table 2. Framework customization points

Framework Component	Customization Artifacts
Translation Infrastructure	Probes, gauges, effectors Gauge-probe mapping Operator-effector mapping
Model Manager	Architecture model Environment model
Architecture Evaluator	Architectural constraints
Adaptation Manager	Adaptation strategies Utility preferences
Strategy Executor	Style operators

We show that the monitoring mechanisms—probes and gauges—observe the running target system. Observations are reported to update properties of the architecture model managed by the Model Manager. The *Architecture Evaluator* evaluates the model upon update to ensure that the system is operating within an

acceptable range, as determined by the architectural constraints. If the evaluation determines that the system is not operating within the accepted range, the Evaluator triggers the Adaptation Manager to initiate the adaptation process and choose the strategy. The Strategy Executor executes the strategy on the running system via system-level effectors. (7 min)

We then unveil the RAIDE workbench, introducing its parts by way of an example target system, Znn.com, which is a news service that serves multimedia news content to its customers. Based on real news sites like CNN.com, Znn.com is architecturally a web-based client-server system with an N-tier style. We showcase the following aspects of the workbench (8 min):

- The workbench explorer allows viewing and managing the various artifacts of the project for customizing Rainbow for Znn.com.
- The AcmeStudio plug-in allows visualizing and editing the architecture model of Znn.com, including the properties to monitor and the architectural constraints to evaluate.
- A YaML plug-in allows editing specification files for the probes, gauges, and effectors for Znn.com; we spend a little time on the gauges.
- The Stitch editor enables composing adaptation strategies to make Znn.com self-adaptive, with syntax highlighting, simple code completion, on-save parsing, and an outline view; we show the adaptation strategies defined for Znn.com.
- The Utility editor allows specifying preferences and profiles that capture the business objectives when self-adapting Znn.com; we show the utility profile for a few quality dimensions and a couple possibilities for utility preferences.
- The Rainbow SDK and the main property file to complete Rainbow customization.

We execute an adaptation run of the customized Rainbow for Znn.com in single-laptop mode to demonstrate Rainbow self-adaptation in action. We

briefly introduce the Rainbow Control Console. After the adaptation run terminates, we show the resulting architecture model and the adaptation log. (4 min)

To demonstrate the ease of modifying Rainbow customization, we tweak the utilities specification for

Znn.com with a different utility-preferences scenario. We execute Znn.com adaptation and show the run result. (5 min)

This concludes our RAIDE research demonstration.

Screenshot

Below is a screenshot of the Rainbow Adaptation Integrated Development Environment built in Eclipse.

