Wednesday, April 2: Other Model-based Self-repair (Owen Cheng)

Title

Gross, P.N., Gupta, S., Kaiser, G.E., Kc, G.S., Parekh, J.J. An Active Events Model for Systems Monitoring. *Proceedings of the Working Conference on Complex and Dynamic System Architecture, Brisbane, Australia, Dec 2001.*

Summary

This paper provides the solution for component monitoring mechanisms in the distributed computing environment. The authors presented a framework for communication between data-source probes and action-based gauges. This framework is based on the intelligent event model called Active Events model. A *Probe* is an individual sensor attached, either statically or dynamically, to a running program. Gauges are software entities that gather, aggregate compute, analyze, disseminate and/or visualize measurement information about software systems. There are three aspects of the probe-gauge relationship that make the problem of connection difficult: the dynamic nature of individual probes, the dynamic topoloty of the various components, and the heterogeneous nature of the system involved. The authors propose the "Active Events" models to encapsulate the probe information. There are two proposed submodels: SmartEvents and Gaugents. SmartEvents is a lightweight submodel for frequent, similar events. Gaugents is a more sophisticated submodel for more significant events. The paper gave the example of the systems that each submodel is appropriate. It also provide the implementation information of each submodel.

Title

Combs, N., Vagel, J. Adaptive Mirroring of System of Systems Architectures. Proceedings of the First ACM SIGSOFT Workshop on Self-Healing Systems (WOSS '02).

Summary

This paper identified an agent-based work flow system to mirror critical elements in large system architectures. The author used an Adaptive Mirroring technique to insert the agent service within the target system. The Adaptive Mirroring is based on the use of probes to intercept a target system workflow and circumnavigate data and control through an alternative path when error occurs. The model is consisted of two layers: core infrastructure and a Service and Contract workflow protocol that can dynamically substitute services. The paper also described the mechanism in great detail. This paper doesn't really mention the high level view of the system. The adaptation is done directly in the system using the Adaptive Mirroring mechanism.

Title

Bond, A., Sud, J. Service Composition for Enterprise Programming. Proceedings of the Working Conference on Complex and Dynamic System Architecture, Brisbane, Australia, Dec 2001.

Summary

This paper describes the design of adaptive service integration environment designed to combine enterprise modeling techniques, lightweight workflow, and dynamic service integration technology. Although the middleware provided many benefits including the contribution to the interconnection of distributed software elements and increase the interoperability between legacy, existing, and future heterogeneous applications, it is not enough for enterprise system. The enterprise system often are complicated, tailored to the environment, and difficult to dynamically adapt to changes in the application set. We required an integration environment that is abstracted from the distributed programming environment. The authors proposed the Open Distributed Service Infrastructure(ODSI) to deal with the enterprise environment problems. The purpose of the authors is to create an environment where applications are described as service collaborations the use enterprise guidance to direct their behavior. The enterprise information is made available through an enterprise model. The ODSI architecture is consisted of seven layers: OS, Middleware, Service, Peers, Enterprise Bus, Business process, User, and Enterprise model. There are four main components to make this architecture works: peers, workflow engine, workflow manager and enterprise model.