

Edeyson

Self-Managed systems: an architectural challenge

Resumo

O artigo descreve sistemas auto-gerenciados (auto-configurável, auto-adaptável, auto-*) e como os mesmos evoluem diante de mudanças através de reconfiguração automática.

Mudanças em requisitos funcionais, não funcionais (como tempo de resposta e desempenho) ou de ambiente levam sistemas auto-gerenciados a se adaptarem via reconfiguração para atendê-las e, caso não consiga fazê-lo, degradar ou reportar uma exceção.

Embora o modelo tradicional de evolução de software consista num processo *off-line* no qual ocorre a evolução numa seqüência de *releases*, a evolução de sistemas auto-gerenciados ocorre com o sistema em execução, o que aumenta sua complexidade.

Diante disso, torna-se imprescindível uma arquitetura de software auto-gerenciado que suporte a reconfiguração dinâmica e apresente os seguintes aspectos:

- Generalidade - abrangente quanto aos domínios de aplicação;
- Nível de abstração – descrição de alto nível de mudanças.
- Potencial para escalabilidade – suporte a composição e técnicas de encapsulamento variando o nível de abstração.
- Construída em trabalhos existentes
- Potencial para uma abordagem integrada.

A incipiência na definição de tal arquitetura levou a uma interseção com a robótica e os trabalhos de sistemas autônomos. Destaca-se uma arquitetura em 3 camadas (Controle, Seqüenciamento e Deliberação) definida como modelo de referência para um contexto de discussão.

Críticas

O texto tem o foco em sistemas auto-gerenciados e numa arquitetura em 3 camadas que dê suporte a sua evolução. Esta ocorre com a mudança de requisitos ou do ambiente e, para que aja a reconfiguração dinâmica deve haver uma coleção prévia de componentes capazes de atender à mudança – reconfiguração de metas.

Não há menção a como a evolução do software pode ser registrada para posterior estudo. Sendo dinâmica, não fica claro no que consiste uma release do software.

Um ponto que ficou nebuloso no texto é que caso o software não consiga se reajustar dinamicamente deve degradar ou reportar uma exceção. O que significa tal degradação e como mapear as exceções possíveis é pouco claro.

A arquitetura em três camadas ficou clara, bem como sua colaboração para a reconfiguração através do atendimento e (re) planejamento de metas.

Questões para discussão

1. Como a evolução de software auto-gerenciado pode ser registrada?
2. O que consiste uma release do software auto-gerenciado?
3. O que significa a degradação do software auto-gerenciado?
4. Como mapear as exceções possíveis à auto-reconfiguração?
5. Como as camadas da arquitetura devem tratar tais exceções?

Referências

Kramer, Jeff and Magee, Jeff. Self-Managed Systems: an Architectural Challenge. OSE '07: 2007 Future of Software Engineering.