

Projekt i implementacja rozproszonego rejestru
menedżerów zarządczych dla systemów SOA ze
wsparciem dla samo-konfiguracji
promotor: doc. dr inż. Michał Sajkowski

Tomasz Wsuł

16.03.2011

SOA (ang. *Services Oriented Architecture*) — architektura zorientowana na usługi. Podstawowe cechy:

- architektura rozproszona,
- duża modularność kodu (aplikację tworzy się pisząc pojedyncze usługi),
- duża reużywalność kodu (brak silnych powiązań między usługami),
- możliwość łączenia wielu technologii (sprzętowych, programowych).

Znaczenie monitorowania aplikacji SOA:

- dostarczanie informacji o ogólnej kondycji systemu,
- możliwość poprawy parametrów pracy aplikacji,
- testowanie i debugowanie aplikacji (monitorowanie funkcjonalne),
- wykrywanie awarii,
- dostarczanie danych do zarządzania (np. równoważenie obciążenia przez inteligentny przydział pracy).

Schemat ideowy zastanego systemu

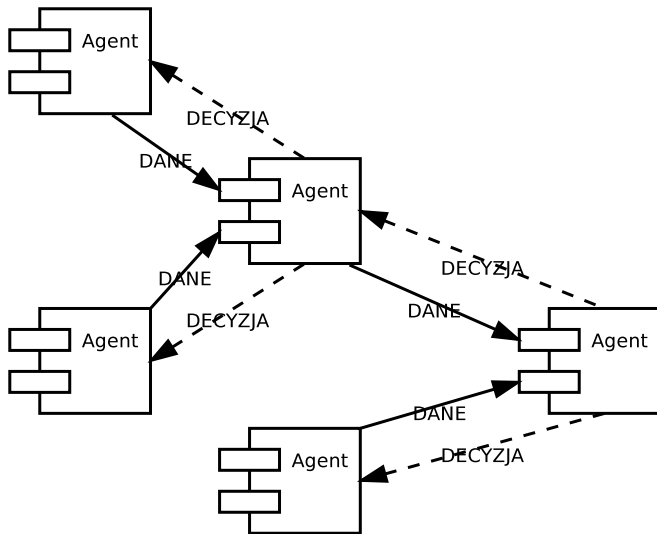


Figure: Schemat ideowy systemu monitorowania

System:

- usługi REST-owe
- identyczne agenty,
- połączone w graf acykliczny,
- duża autonomiczność agentów.

Agent:

- autodetekcja zasobów
- budowa modułowa z malutkim rdzeniem,
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

Istotne technologie użyte w systemie:

- platforma: Java, OSGi,
- format dokumentacji: lyx,
- wspierane IDE: Eclipse,
- repozytorium kodu: SVN.

System:

- ~~identyczne agenty~~, trudne aktualizacje i zmiany
- połączone w graf acykliczny,
- duża autonomiczność agentów.

Agent:

- autodetekcja zasobów
- budowa modułowa z małym rdzeniem,
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- identyczne ~~agenty~~, trudne aktualizacje i zmiany
- połączone, ręczna konfiguracja każdego węzła
- duża autonomiczność agentów.

Agent:

- autodetekcja zasobów
- budowa modułowa z malutkim rdzeniem,
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne~~ agenty, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~ problemy z podejmowaniem i przekazywanie decyzji

Agent:

- autodetekcja zasobów
- budowa modułowa z malutkim rdzeniem,
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne~~ agenty, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~ problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, wymagany każdy plugin w agencie
- budowa modułowa z małym rdzeniem,
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne agenty~~, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność....~~ problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, wymagany każdy plugin w agencie
- ~~budowa modułowa z małym rdzeniem~~, problematyczne dostarczanie pluginów
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne agenty~~, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~. problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, wymagany każdy plugin w agencie
- ~~budowa modułowa z małym rdzeniem~~, problematyczne dostarczanie pluginów
- automatyczne ładowanie pluginów,
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne agenty~~, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~. problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, wymagany każdy plugin w agencie
- ~~budowa modułowa z małym rdzeniem~~, problematyczne dostarczanie pluginów
- ~~automatyczne ładowanie pluginów~~, muszą być dostarczone do agenta
- rekonfiguracja w trakcie pracy (niemal dowolna).

System:

- ~~identyczne agenty~~, trudne aktualizacje i zmiany
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~, problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, wymagany każdy plugin w agencie
- ~~budowa modułowa z małym rdzeniem~~, problematyczne dostarczanie pluginów
- ~~automatyczne ładowanie pluginów~~, muszą być dostarczone do agenta
- ~~rekonfiguracja w trakcie pracy~~, brak ogólnego obrazu systemu monitorowania i monitorowanego.

System:

- ~~identyczne agenty~~, ogólnodostępne repozytorium pluginów
- ~~połączone...~~, ręczna konfiguracja każdego węzła
- ~~...autonomiczność...~~, problemy z podejmowaniem i przekazywanie decyzji

Agent:

- ~~autodetekcja zasobów~~, ogólnodostępne repozytorium pluginów,
- ~~budowa modułowa z małym rdzeniem~~, ogólnodostępne repozytorium pluginów
- ~~automatyczne ładowanie pluginów~~, ogólnodostępne repozytorium pluginów
- ~~rekonfiguracja w trakcie pracy~~, brak ogólnego obrazu systemu monitorowania i monitorowanego.

System:

- ~~identyczne agenty~~, ogólnodostępne repozytorium pluginów
- ~~połączone...~~, rejestr agentów w systemie monitorowania
- ~~...autonomiczność...~~, rejestr agentów w systemie monitorowania,

Agent:

- ~~autodetekcja zasobów~~, ogólnodostępne repozytorium pluginów,
- ~~budowa modułowa z małym rdzeniem~~, ogólnodostępne repozytorium pluginów
- ~~automatyczne ładowanie pluginów~~, ogólnodostępne repozytorium pluginów
- ~~rekonfiguracja w trakcie pracy~~, brak ogólnego obrazu systemu monitorowania i monitorowanego.

System:

- ~~identyczne agenty~~, ogólnodostępne repozytorium pluginów
- ~~połączone...~~, rejestr agentów w systemie monitorowania
- ~~...autonomiczność...~~, rejestr agentów w systemie monitorowania,

Agent:

- ~~autodetekcja zasobów~~, ogólnodostępne repozytorium pluginów,
- ~~budowa modułowa z małym rdzeniem~~, ogólnodostępne repozytorium pluginów
- ~~automatyczne ładowanie pluginów~~, ogólnodostępne repozytorium pluginów
- ~~rekonfiguracja w trakcie pracy~~, graf reprezentujący system ((zasoby+usługi)+zależności).

Repozytorium pluginów:

- dystrybucja pluginów w systemie,
- wpieranie rozwiązywania zależności między pluginami,
- powiązanie monitorowanych elementów z pluginami,
- ułatwianie aktualizacji pluginów.

Rejestr agentów:

- rejestrowanie/wyrejestrowywanie agentów,
- wspomaganie śledzenia stanu agentów (uruchomione pluginy),
- udostępniane akcje (np. migracja maszyny wirtualnej),
- heartbeat.

Graf systemu:

- węzły — wskazanie usługi/zasobu + metadane,
- łuki — zależności między usługami,
- historia — umożliwia analizę *post-mortem*,
- brak gwarancji spójności — system też nie musi być spójny (wiszące łuki).

- platforma: Java, OSGi,
- format dokumentacji: lyx,
- wspierane IDE: Eclipse,
- repozytorium kodu: SVN+git (plugin git-svn),
- baza danych: MongoDB — dokumentowa baza danych,
- api: REST + XML.

- koncepcja rejestru menedżerów zarządczych ze wsparciem dla samo-konfiguracji,
- implementacja rejestru,
- proste GUI dla użytkownika pozwalające na przeglądanie rejestru (przeglądarka WWW+XSLT+JavaScript),
- pełna dokumentacja JavaDoc,
- dokumentacja techniczna w formie pracy magisterskiej.

Pytania?