McMaster University

Multi-View and Context-based Analysis for Service Integration

Anis Yousefi and Kamran Sartipi
{yousea2, sartipi}@mcmaster.ca

Department of Computing and Software
McMaster University
Hamilton, ON, Canada

November 2009

Summary

Service-oriented architecture (SOA) enables reuse and sharing of system resources across organizations via the use of interoperable services to build complex service-based processes. However, SOA lacks the required infrastructure to thoroughly analyze services and uses producer-provided service description for service management purposes such as service selection and integration.

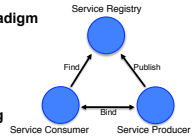
This research aims at devising an analysis-based framework for service integration which integrates static and dynamic service analysis techniques to make sophisticated decisions. The proposed framework benefits from techniques in decision support systems, reverse engineering and data mining to enhance the state of service integration.

We propose a decision flow based service selection mechanism that considers requester's situation and its domain properties when answering a query. The proposed mechanism improves the effectiveness of service integration through dynamic selection of the appropriate analysis methods based on the context of the requester.

Introduction

Service Oriented Architecture (SOA) *is regarded as key paradigm for implementing enterprise applications by integrating reusable packages of functionality, called services*

- **Publish-Discovery-Interaction paradigm**
  - Service description (e.g. WSDL)
- **Service selection and integration are based on description matching**



Service Integration: Problems and Applicable Techniques

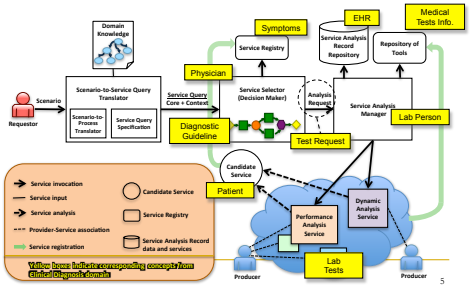
Problems in Service Integration

- **Limited knowledge about services**
  - Service descriptions, which are written by the producer!
- **Lack of thorough service analysis infrastructure**
  - Causing the selection of improper services
- **Fixed selection mechanism**
  - Less adaptability

Applicable Techniques

- **Static analysis**
  - Extracting software properties by analyzing source code and documentations
- **Dynamic analysis**
  - Extracting run-time properties using software instrumentation and execution traces analysis

The Proposed Framework



Scenario to Service Query Translation

Sample Scenario from healthcare

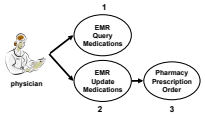
Mr. X needs to get a repeat of his usual medications. He visits his family practitioner, Dr. P. Dr. P pulls up Mr. X's chart in her EMR. She updates the EMR with Mr. X's new medication. When Dr. P closes Mr. X's chart on the EMR, it automatically sends a prescription request to a pharmacy to dispense.

Recognizing the required functionalities

- The required functionalities
  - Dr. P. needs to pull up Mr. X's chart from her EMR (Electronic Medical Record) system.
  - Dr. P. needs to update Mr. X's chart with new medications.
  - EMR needs to request pharmacy for dispensing a prescription.

Specifying the Integrated Service

The Process	The Functionalities



Functionality 1 - EMR Query Medications

Functionality: type = EMRQuery
Input: type = patient identifier
Output: type = medication history

Functionality 2 - EMR Update Medications

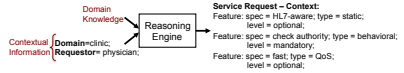
Functionality: type = EMRUpdate
Input: type = patient identifier
Input: type = new medication
Output: type = medication history

Functionality 3 - Pharmacy Prescription Order

Functionality: type = PharmacyPrescriptionOrder
Input: type = patient identifier
Input: type =new medication
Output: type = acknowledgement

7

Specifying Service Request

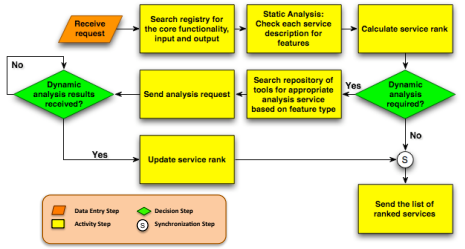


EMRQuery service request

Core:
Functionality: type = EMRQuery;
Input: type = patient identifier;
Output: type = medication history;
Context:
Feature: spec = HL7-aware; type = static; level = optional;
Feature: spec = check authority; type = behavioral; level = mandatory;
Feature: spec = fast; type = QoS; level = optional;

8

Service Selection Flow



9

Phases of Dynamic Analysis

Phase 1 – Scenario Generation

- Generating a set of functionality-oriented scenarios for the requested functionality
  - Scenario is a sequence of service operations with specific values associated with a functionality
  - Example: for functionality 'EMRQueryMedications', a scenario set for a candidate EMR service includes a call to operation 'query' on the candidate service with different sets of medications as input

Phase 2 – Extracting Execution Traces

- Executing the scenario set on an instrumented candidate service to extract the trace (sequence) of function calls related to the requested functionality
  - Includes function inputs and outputs

10

Phases of Dynamic Analysis

Phase 3 – Execution Pattern Mining

- Extracting pattern of function calls using *sequential pattern mining*
  - The extracted pattern represents core functions that realize the required functionality
  - Example: pattern associated with functionality 'EMRQueryMedications'

EMR.query.receive, EMR.login, EMR.query.queryDB, EMR.response.returnResults

Phase 4 – Feature Verification

- Verifying features on the extracted pattern
  - Example: behavioral feature *"service must require user to login before sending out medication history"* is validated on the extracted pattern

11

Contributions

- **Enhancing the state of SOA by providing analysis infrastructure and analysis-based service integration framework**
  - A mechanism for extracting service properties through dynamic analysis and execution pattern mining
  - An open SOA-based analysis framework for service management purposes
  - A service selection mechanism based on decision support systems strategy with the following characteristics:
    - Flexible
    - Context-based
    - Domain-aware
    - Analysis-based
  - A mechanism for semi-automatic scenario to query translation

12