An Intelligent Broker Approach to Semantics-based Service Composition

Yufeng Zhang
National Lab. for Parallel and Distributed Processing
Department of Computing Science
National Univ. of Defense Technology, Changsha, China
Email: yufengzhang@nudt.edu.cn

Hong Zhu
Department of Computing and Communication Technology
Oxford Brookes University
Oxford OX33 1HX, UK
Email: hzhu@brookes.ac.uk

Approaches to service composition

- Workflow:
  - Workflow definition + workflow engines:
    - e.g. orchestration and choreography; BPEL4WS; OWL-S
- Planning:
  - E.g. OWLS-Xplan, SHOP2, etc.
- Broker:
  - Sycara *et al*. [2003, 2004] :
    - high flexibility
    - wide applicability

Current state of service brokers

The functions of service brokers:

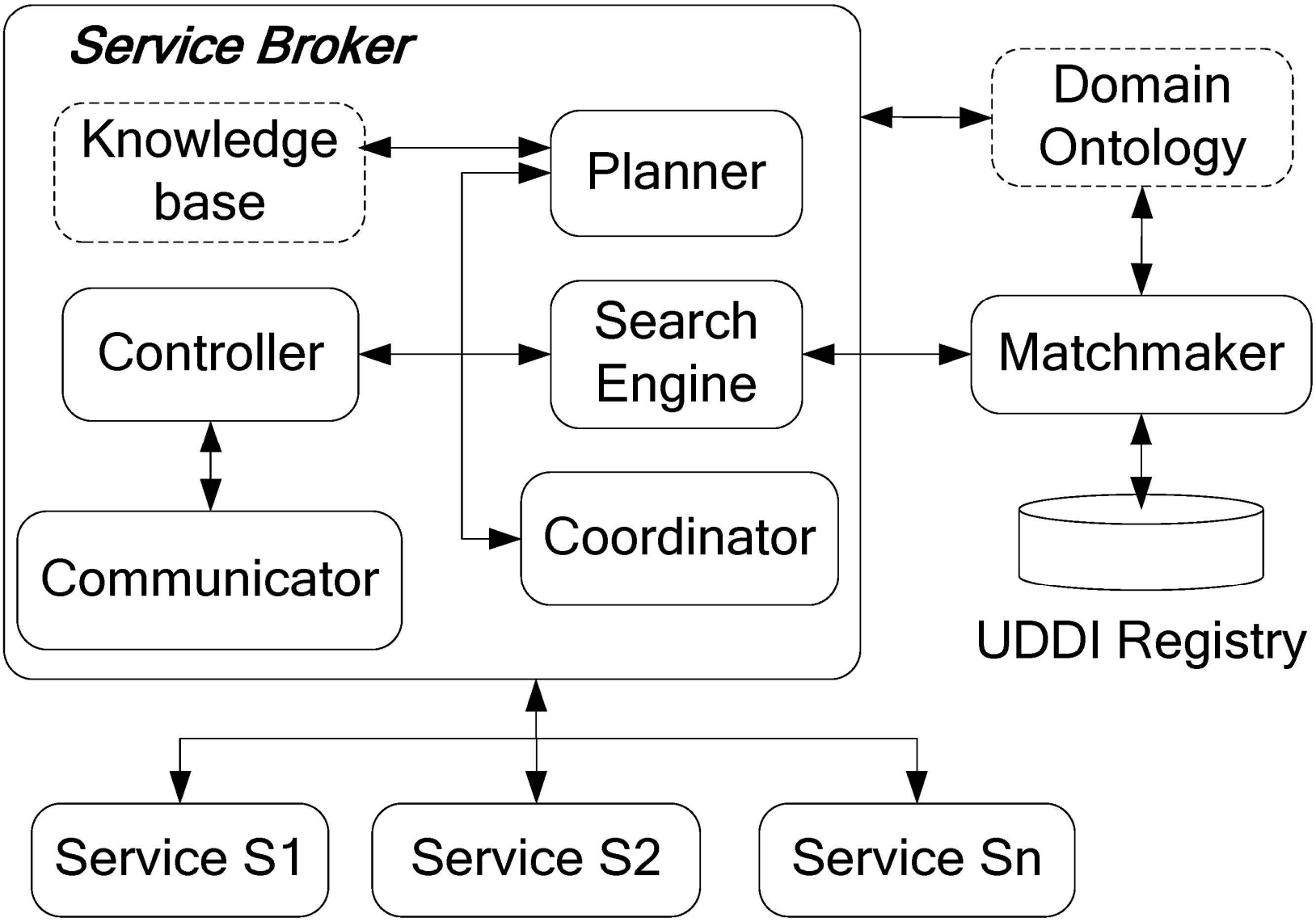
- **Interpreting** the semantics of service queries and the registered capabilities of service providers;
- **Searching** for the service providers that matches a requester's query and sometime selecting the one with best track record of quality of services;
- **Invoking** the selected service provider on the requester's behalf and interacting with the provider if necessary to fulfil the query;
- **Returning** query results to the requester.

The goal

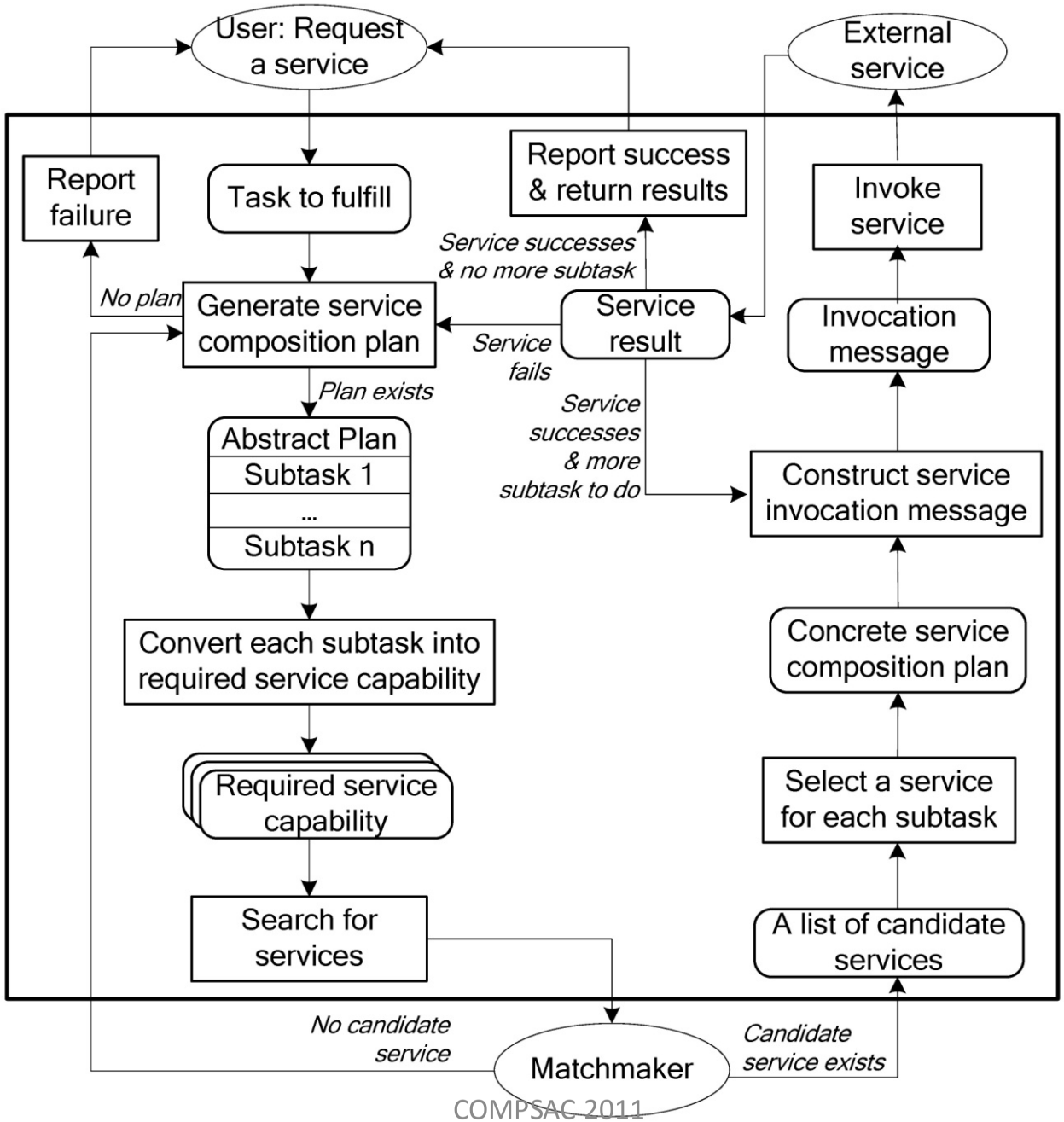
To enhance the power of service brokers with the capability of:

- **decomposing** requested services into a number of subtasks,
- **searching** for the best fit services for each subtask,
- **composing** and coordinating these services in execution.

I-Broker: Architecture



I-Broker: Control Process

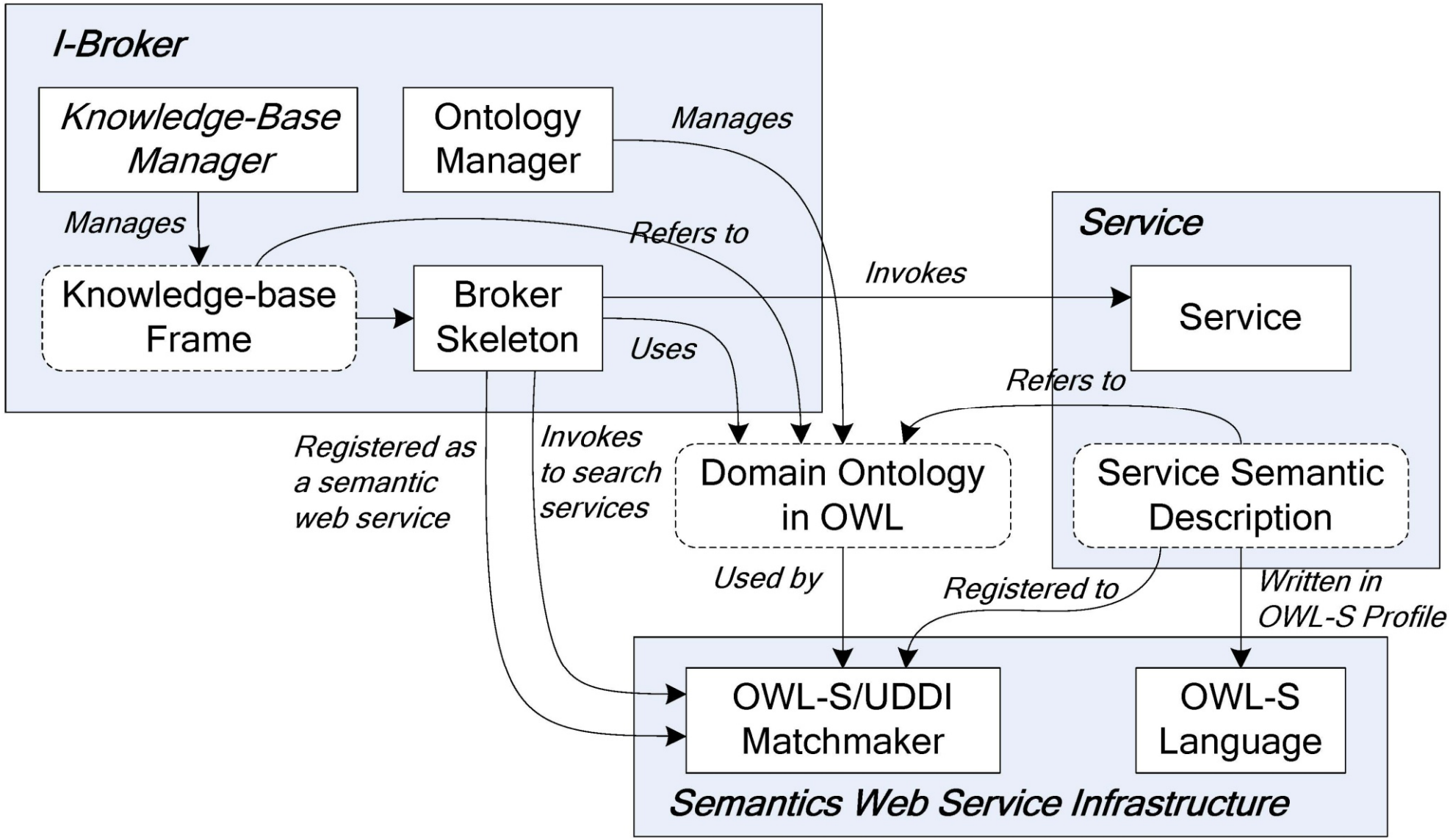


Knowledge-base

- It contains codified knowledge on how a task can be fulfilled by a number of subtasks.
- Each type of tasks is defined by a set of parameters
  - descriptive parameters
    - describes the functionality of the task, such as the activity of the task, the execution environment of the task, and so on.
  - functional parameters
    - gives the data to be transformed by the task, including input and output data.
- The values of parameters are concepts defined in the ontology of the application domain.
- The knowledge is represented in the form of rules:

$$T(p_1,\ldots,p_n) \Rightarrow T'_1(p_{1,1},\ldots,p_{1,n1});\ldots;\ T'_k(p_{k,1},\ldots,p_{k,nk})$$

Prototype Implementation



Experimental Evaluation

- Goal:
  - to evaluate the scalability of the proposed approach
- Design of the experiments
  - Three types of objects in the experiments:
    - Services:
      - registered to the semantic WS registry.
      - capabilities represented in the form of service profiles.
    - Rules in knowledge-base:
      - represented in the form of XML files
      - stored locally within the broker as the knowledge-bases.
    - Service requests:
      - represented in the format of XML using the ontology.

The services that test broker searches for

The service requests submitted to the brokers

About the workflow in the application domain and the usages of specific services

Experiment method

- Data mutation technique: [Shan&Zhu 2009]
  - Seeds of test data
  - Mutation operators

  } Test data (mutant test cases)

Seeds of Services

Name	Description
CASCAT	Generate test cases from algebraic specifications
Test Translator	Translate test cases from CASCAT format into Test Executor format
Test Executor	Execute tests for a numeric calculator WS
Klee	Symbolic execution of C code
Magic	Check component's conformance to specification
XML Comparator	Compare XML files
Java NCSS	Metrics for Java program
Findbugs	Find bugs in Java program by static analysis
PMD	Find potential bugs in Java by static analysis
WSDL Test Gen	Generate test cases from WSDL
WS Test Executor	Execute tests generated by WSDL Test Gen

Mutation operators

Given an ontology.

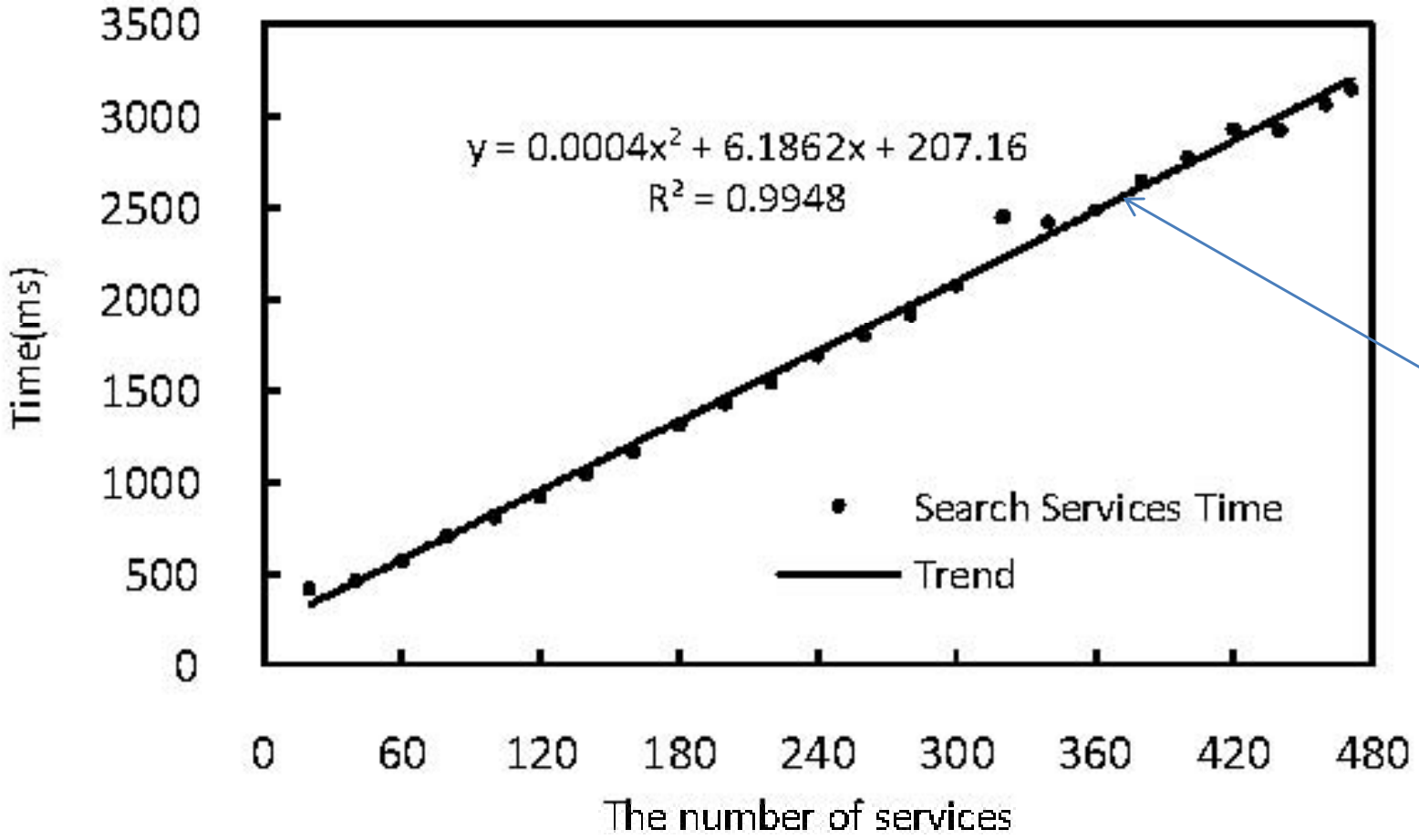
Let x be any of the parameters in service profiles.

- **RxF:**
  - Replace the x parameter in the profile, which is a class in the ontology, by its father class in the ontology;
- **RxS:**
  - Replace the x parameter in the profile by one of its subclasses in the ontology;
- **RxB:**
  - Replace the x parameter in the profile by one of its brother classes in the ontology;
- **RxN:**
  - Replace the x parameter in the profile by a class in the ontology that has no relation to the parameter.

	seeds	Mutants	Total
Service	11	460	471
Rule	40	2049	2089

Scalability w.r.t. the number of services

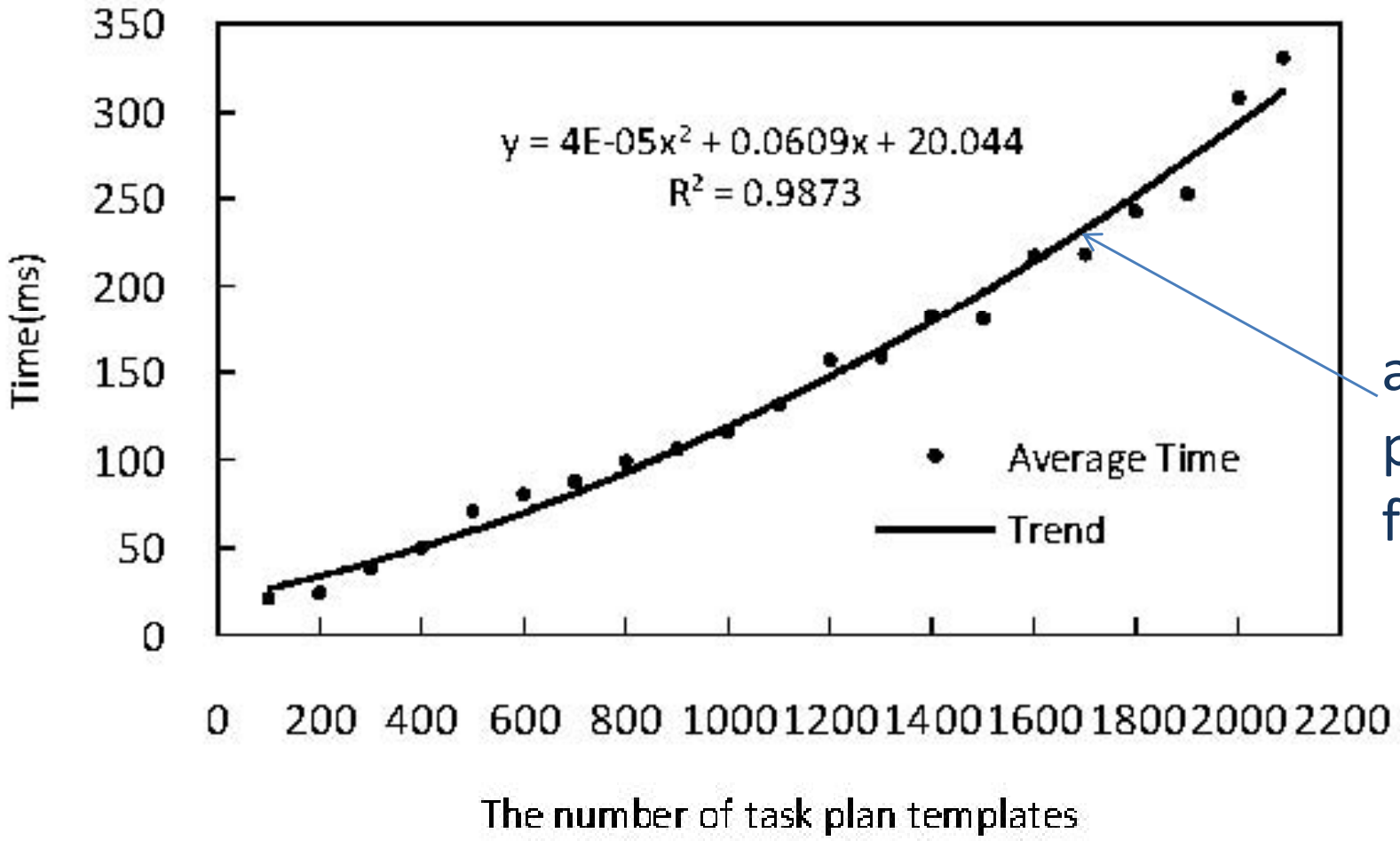
- Vary the registry size from 20 to 471
- Execute the broker repeatedly for 30 times



almost a linear function

Scalability w.r.t. the size of KB

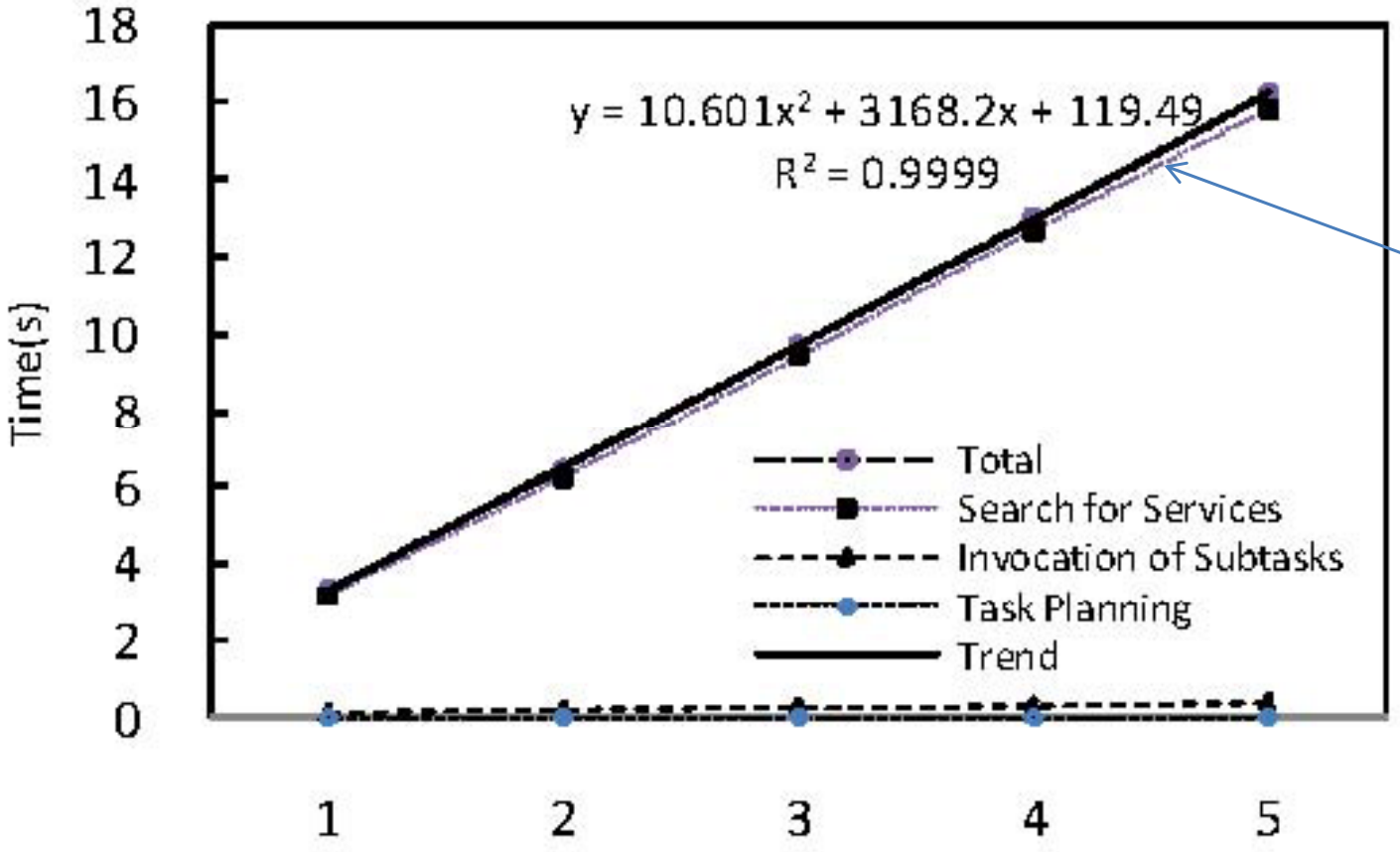
- Vary the knowledge size from 100 to 2089
- execute the broker on each service request for 20 times



a quadratic polynomial function

Scalability w.r.t. to task complexity

- vary the service requests (5 classes of requests)
- execute the broker on each service request for 30 times



A quadratic polynomial function

Comparison with Related Works

Workflow	Planning	I-Broker
Workflow encoded in executable form	Workflow knowledge is represented in the form of task decomposition rules	Workflow knowledge is represented in the form of task decomposition rules
	The knowledge of workflow is simply converted from OWL-S	Workflow knowledge is treated as domain knowledge
Workflow knowledge is used statically	Plans are mostly generated by a brutal force of inference • Use the whole service registry as the search space	Use workflow to generate service plan dynamically Use rules as a means of reduce search space: • scalable • efficient
		Knowledge-base and ontology are open, and composible

Future work

- Develop knowledge-based manager
  - Enable user to write rules in a language of high level of abstraction rather than directly in XML
  - Support populating, updating, and testing the knowledge-base
- Embed service monitoring functionality in the service broker
  - QoS directed service selection
- Replace Semantic Web Service with a better semantic inference mechanism