Technical University of Cluj-Napoca, Romania
Department of Computer Science

A Hybrid Firefly-inspired Approach for Optimal Semantic Web Service Composition

Cristina Bianca Pop, Viorica Rozina Chifu, Ioan Salomie,
Ramona Bianca Baico, Mihaela Dinsoreanu, **Georgiana Copil**

WoSS, June 7, 2011
Timisoara, Romania

Agenda

- Context
- Existing Approaches
- Objectives
- Our Semantic Web Service Composition Model
- The Hybrid Firefly Selection Method
- Experimental Framework
- Testing and Evaluation
- Conclusions and Future Work

Context

- Optimal Web service composition selection
  - Combinatorial optimization problem
  - NP-hard problem
- How to approach the selection problem?
  - Exhaustive search algorithms – high computation time
  - Approximate algorithms (such as meta-heuristics)
    - Near-optimal and in some cases the optimal solutions
    - Finite time
    - Partially processes the search space

Existing Approaches

Approach	Composition Method	Solution Encoding	Features
Genetic-based [Jiang2011] [Wang2008]	Abstract workflow of tasks – each task has a set of concrete services	Integer array / binary string	Genetic algorithms – bind concrete services to the abstract tasks Genetic operators – generate new workflow instances (new candidate solutions) Fitness functions – evaluate the quality of a candidate solution based on *QoS* attributes
Particle Swarm Optimization-based [Wang2010] [Ming2007]	Abstract workflow of tasks – each task has a set of concrete services	Particle position – concrete services mapped on workflow tasks Particle velocity – indicates how the concrete services should be changed	Addition, subtraction and multiplication operators on positions and velocities to identify the optimal composition solution *QoS*-based selection Variation operator to avoid stagnation in local optimum
Immune-inspired [Xu2008] [Gao2006]	Abstract workflow of tasks – each task has a set of concrete services	Binary string	Applies immune-inspired principles (the clonal selection process) *QoS*-based selection Perform mutation processes to generate new solutions

Objectives

- Develop a **hybrid firefly selection method** for identifying the optimal composition solution featuring
  - A combination of the extended version of the firefly algorithm [Lukasik2009] with genetic operators
  - Selection criteria - the *QoS* attributes and the semantic quality of the services involved in composition
- Test and evaluate the proposed method

Our Semantic Web Service Composition Model (I)

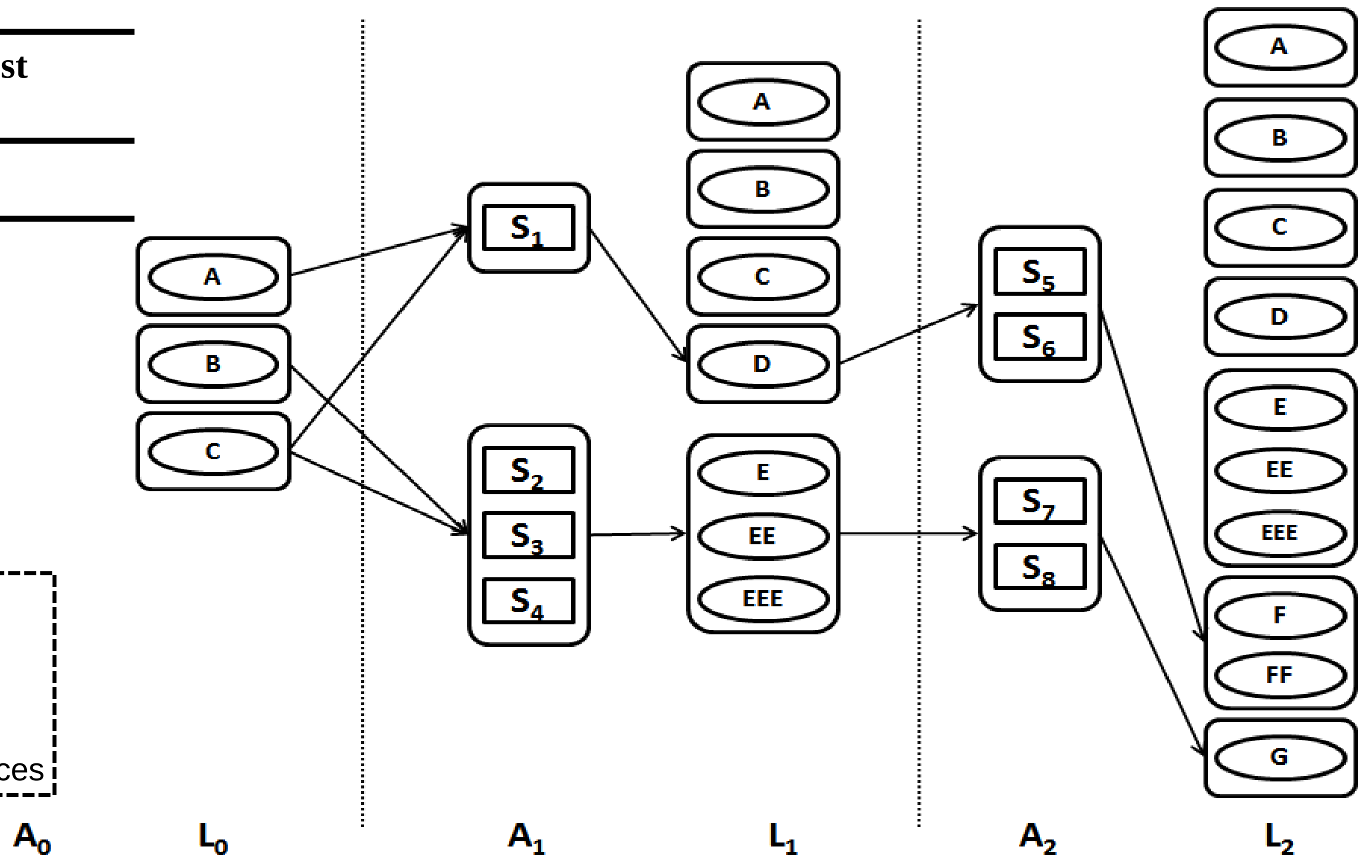
- Mapping AI planning graphs to semantic Web service composition

AI planning graph concepts	**Semantic Web service composition concepts**
Planning graph	Enhanced Planning Graph
Action	Ontology concept annotating a service operation
Precondition	Ontology concept annotating a service input parameter
Positive effect	Ontology concept annotating a service output parameter
Initial state (A_0,L_0)	(Ø, Ontology concepts describing the user provided input parameters)
Goal literals (L_g)	Ontology concepts describing the user requested output parameters
Action set (A_i)	Set of clusters of services for which the input parameters are L_{i-1} literals
Literal set (L_i)	Union of the L_{i-1} set and the sets of the outputs of the services in A_i

Our Semantic Web Service Composition Model (II)

- An example of an Enhanced Planning Graph (EPG)

User Request Inputs	User Request Outputs
A, B, C	F, G



The Hybrid Firefly Selection Method

Overview

- Finds the optimal or a near optimal composition solution according to *QoS* user preferences and semantic quality by using
  - The **Enhanced Planning Graph** (*EPG*)
  - A **fitness function** for quality evaluation

The Hybrid Firefly Selection Method
Biological Inspiration

- Fireflies produce short and rhythmic flashes for
  - Communication - attract mating partners
  - Hunting – attract potential prey
- Source of inspiration for the Firefly Algorithm [Yang2008]

From http://nicholasnghwaeedays.blogspot.com/2010/06/fireflies-omgfacts.html

The Hybrid Firefly Selection Method
Problem Formalization (I)

- Mapping the firefly mating behavior to the problem of selecting the optimal or a near-optimal Web service composition solution

Firefly mating behavior concepts	Web service composition concepts
Firefly	Artificial firefly
Firefly position	A service composition solution
Firefly brightness	The quality of a composition solution evaluated with a multi-criteria fitness function
Attractiveness between two fireflies	Similarity between two composition solutions
Firefly movement	Modification of the firefly's current composition solution
Environment	Enhanced Planning Graph

The Hybrid Firefly Selection Method
Problem Formalization (II)

- Artificial firefly
  - $firefly = (sol, score)$ where $sol = \{solElem_1, ... solElem_n\}$

$$solElem_i = \{s_{kl}^j \mid s_{kl}^j \text{ is the } l-th \text{ service from cluster } k \text{ on layer } i \text{ of the EPG}\}$$

- QF multi-criteria function

$$QF(sol) = \frac{w_{QoS} * QoS(sol) + w_{Sem} * Sem(sol)}{(w_{QoS} + w_{Sem}) * |sol|}$$

$$QoS(sol) = \frac{\sum_{i=1}^{n} w_i * qos_i(sol)}{\sum_{i=1}^{n} w_i}$$

$$Sem(sol) = \frac{\sum simS(s_{kl}^j.out, s_{qr}^p.in)}{nLinks}$$

The Hybrid Firefly Selection Method

The Selection Algorithm - Overview

- Objective
  - Identify the optimal or a near-optimal service composition solution in a short time and without processing the entire search space
- Inputs
  - EPG – Enhanced Planning Graph resulted from the Web service composition process
  - w_{Sem}, w_{QoS} – weights indicating the relevance of a solution's semantic quality compared to its *QoS* quality
  - noF – the number of fireflies

$$noF = Round(\sqrt[n]{noSol})$$

- Output
  - The optimal or a near-optimal composition solution

The Hybrid Firefly Selection Method
The Selection Algorithm – Stages

- Initialization stage
  - Associates a randomly generated service composition solution to each firefly
- Iterative stage
  - Checks the attractiveness between every two fireflies based on the score of their service composition solutions evaluated with QF
  - Repeated until the best identified solution has been the same for the last noS iterations

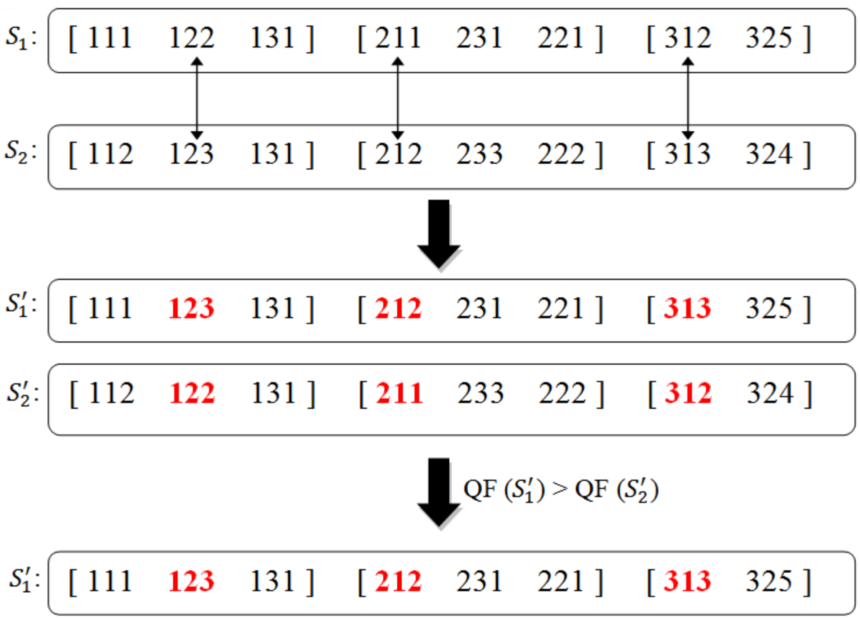
The Hybrid Firefly Selection Method
The Selection Algorithm – Iterative Stage (I)

- Check the attractiveness between every two fireflies, $firefly_i$ and $firefly_j$
  - If $firefly_i$ is attracted to $firefly_j$ ($QF(firefly_i.sol) < QF(firefly_j.sol)$) then
    - Compute distance $r = QF(firefly_i.sol) - QF(firefly_j.sol)$
    - Apply a crossover operation on $firefly_i.sol$ based on distance r
    - Generate a random mutation vector μ
    - Apply the random mutation vector on the new $firefly_i.sol$
- Identify and store the best service composition solution
- Generate a random mutation vector μ
- Apply the random mutation vector on the best service composition solution

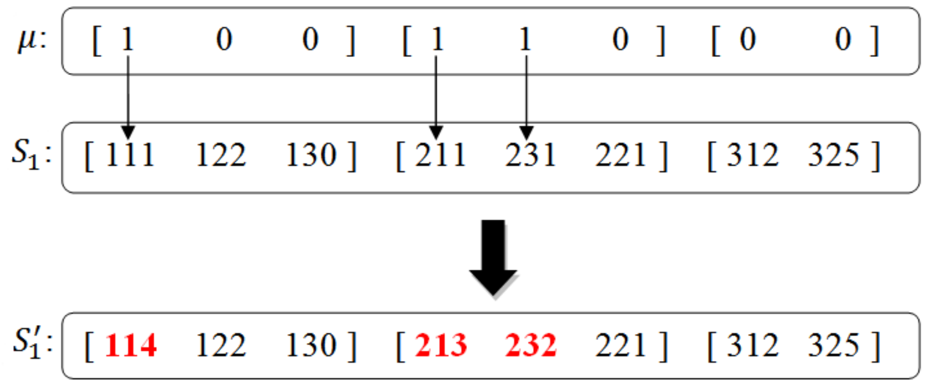
The Hybrid Firefly Selection Method

The Selection Algorithm – Iterative Stage (II)

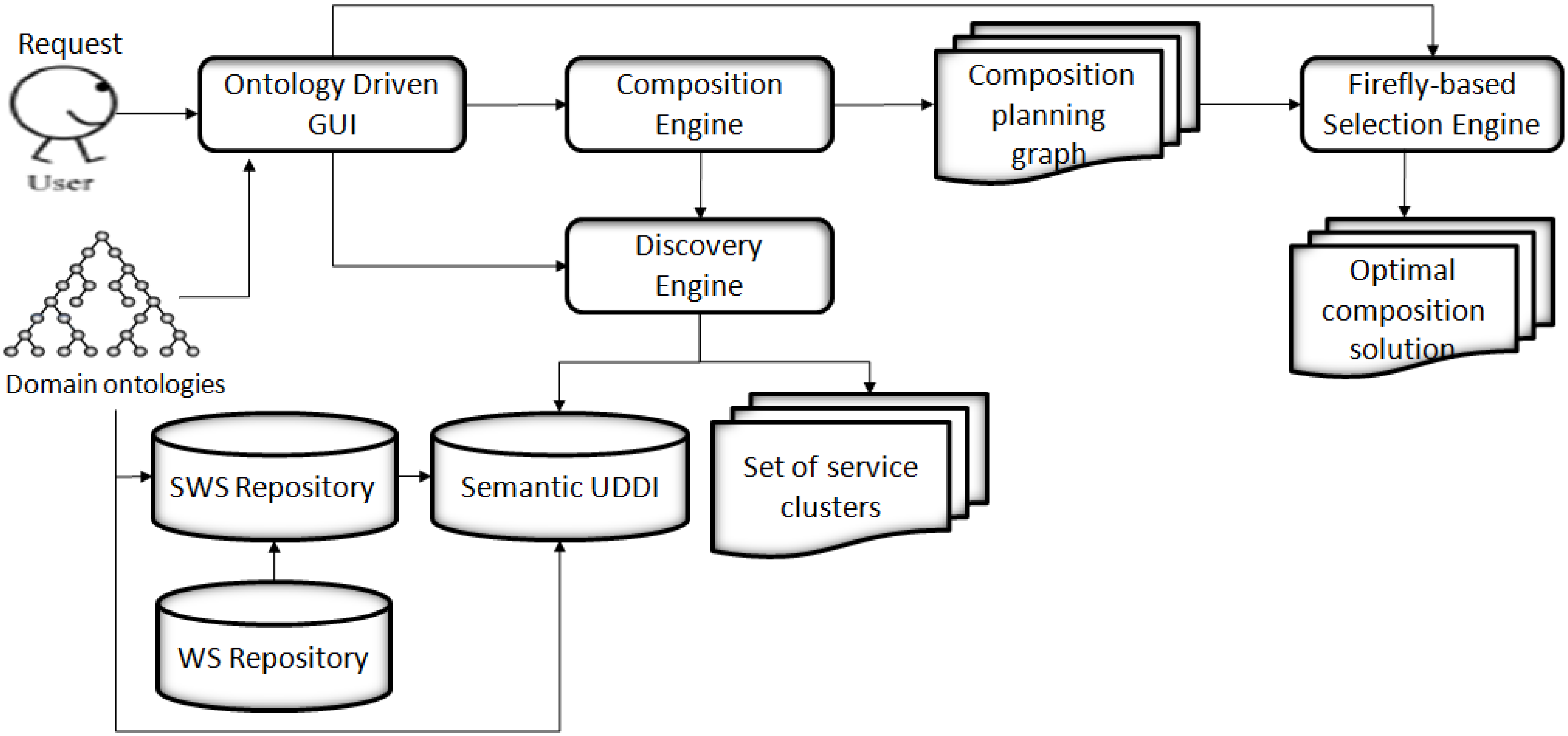
Example of applying the crossover operator



Example of applying the random mutation operator



Experimental Framework
Overview



Testing and Evaluation
Testing Scenario

- Example of user request

User inputs	Requested outputs	QoS weights	Semantic quality weight
SourceCity, DestinationCity, StartDate, EndDate HotelType, NumberOfPersons, NumberOfRooms, CarType, ActivityType	AccomodationInvoice, FlightInvoice, CarInvoice	Total QoS = 0.55 Av = 0.30, Rel = 0.30, Ct = 0.15, Rt = 0.25	0.35

- Resulted *EPG* configuration
  - 3 layers
  - 11 clusters
  - 51 semantic Web services

Testing and Evaluation
Testing Methodology (I)

- Identify the optimal composition solution by exhaustive search
  - All 13996800 *EPG*-encoded possible solutions were generated
  - For each possible solution the value of the QF function was computed $\rightarrow$ the optimal composition solution has the score of 0.6947
- Set the initial values for the adjustable parameters
  - Crossover thresholds: $r_1 = 0.001$, $r_2 = 0.005$, $r_3 = 0.01$
  - Number of stagnations: $noS = 3$
- Tune the values of the adjustable parameters
  - Run the selection algorithm on various configurations of the adjustable parameters, starting from the initial configuration
  - Analyze how the different configurations of the adjustable parameters influence the convergence of the algorithm towards the optimal composition solution

Testing and Evaluation
Testing Methodology (II)

noS	r_1	r_2	r_3	*NoIt*	*noSol*	*QF*	*STDEV*	T_{exec}	*Processed space (%)*
3	0.001	0.005	0.01	3,09	2103	0,6862	0,0084	8,4	0,015
3	0.003	0.006	0.008	3,14	2186	0,6857	0,0089	9,3	0,0156
3	0.03	0.06	0.09	3,21	2146	0,6821	0,0125	8,4	0,0153
3	0.01	0.05	0.09	3,15	2192	0,6836	0,0111	7,8	0,0156
6	**0.001**	**0.005**	**0.01**	**6,12**	**2145**	**0,687**	**0,0076**	**8,55**	**0,0152**
6	0.003	0.006	0.008	6,1	2120	0,6857	0,0089	8,8	0,0151
6	0.03	0.06	0.09	6,1	2120	0,6857	0,0089	8	0,0151
6	0.01	0.05	0.09	6,17	2098	0,6835	0,0112	7,4	0,0149
9	0.001	0.005	0.01	9,11	2079	0,6848	0,0098	7,65	0,0148
9	0.003	0.006	0.008	9,11	2079	0,6839	0,0107	8,1	0,0148
9	0.03	0.06	0.09	9,24	2070	0,6821	0,0126	7,9	0,0148
9	0.01	0.05	0.09	9,25	2140	0,6838	0,0108	8,2	0,0153

Conclusions

- Main contributions of our work
  - A hybrid firefly selection method for identifying the optimal service composition solution
    - Identifies a near-optimal solution ($STDEV = 0.076$) in 8.55 seconds by processing 0.015% of the search space
  - A proof-of-concept framework to validate the proposed approach
- Future Work
  - Comparatively analyze the hybrid firefly selection method with other selection methods using the same test scenarios