RACING: Agent-Mediated Web Service Composition for Rational Information Retrieval

Vadim Ermolayev

Dept. of Mathematical Modeling and Information Technologies
Zaporozhye State University
69063 Zaporozhye
Ukraine
eva@zsu.zp.ua

Web services are the emerging technology promising to become one of the key enablers of the Semantic Web. There are strong prerequisites that, being self-described and self-contained modular active components, web services will appear to become the key elements in assembling intelligent service providing software infrastructures in the future.

The domain of Semantic Web Services' orchestration is becoming increasingly hot. Several ongoing initiatives define compositional notations for web services. These notations express the flow of control and data across a collection of web services **whose choreography performs a workflow.** For example, IBM, Microsoft and BEA have recently released BPEL4WS[1] as the specification for coordinating business processes over the Web. OASIS has formed the Technical Committee to continue the work on the Web Services Business Process Execution Language.

From the other hand, having a compositional specification it is not just enough for web services to be orchestrated in a dynamic business process, like having a recipe doesn't yet grant having a meal. A pro-active component capable to understand the "score" is required. Pro-active understanding of the process specification is not only the ability to ensure the right sequence and the proper combination of the components. It also means the capability to find the best supplier in the dynamic and open environment. This is why more and more attention is paid to the field of agent-enabled web service composition. An interesting fact is that agents may acquire new capabilities by assimilating the semantics of web services' orchestration. As Paul Buhler and José Vidal wrote[2]:

> "… the semantic web and the emergence of a Web Services component model can facilitate agent-based workflow management in open environments. If agents are used to wrap semantically described Web Services, then the **semantic service descriptions become the basis for determining the agent's first-order abilities**. Likewise, **a common semantic markup** for Web Services will **facilitate effective communication between agents**."

Agent-Mediated Cooperative Distributed Service Composition

Our contribution to the research in Semantic Web Services domain is the ongoing development of the framework for Agent-Mediated Cooperative Distributed Service Composition (CDSC). The main idea of the approach is to apply the results and the techniques from Cooperative Distributed Problem

[1] Business Process Execution Language for Web Services: http://dev2dev.bea.com/techtracks/BPEL4WS.jsp

[2] **Paul Buhler and José M. Vidal.** (2003) Semantic web services as agent behaviors. In B. Burg, J. Dale, T. Finin, H. Nakashima, L. Padgham, C. Sierra, and S. Willmott, editors, *Agentcities: Challenges in Open Agent Environments*, pages 25-31. Springer-Verlag.

Solving (CDPS) area and our framework for Agent-Enabled Business Process Management and Performance to web services composition. From the architectural point of view a distributed multi-agent system acts as the mediator between service requestors (both humans and software agents) and service providing agents. Web services are treated as the capabilities of service providers. Service providing agents wrap respective services. The major role of the service mediating system is, thus, to guide, to co-ordinate the pro-active team work of service providing agents. Given a group of service providing agents is collaborating for a compound service provision, such a service may be considered a dynamic business process assembled of a proper combination of the capabilities of the participating agents. The agents form their coalitions for service provision via the rounds of negotiations. They negotiate on the outsourcing and the provision of the services they wrap. These negotiations actually form the mechanism of contracting in the process of the mentioned dynamic assembly. Our current activities are focused on the implementation of these high-level ideas for intelligent agent-based mediation of distributed information retrieval in frame of the RACING project[3].

The goal of the RACING project is to provide mediation facilities for content-driven query processing: query transformation, query decomposition and distribution among independent, autonomous, rational document retrieval service providers wrapping respective document resources. The results of this collaborative query processing are likewise fused by the mediator. In a nutshell the **overall high-level goal** of the RACING mediator is:

- To deliver **semantically matching** (to the requestor's query) **result** (a resource or a set of resources, possibly from different providers)
- For a **rationally negotiated incentive**
- In the **agreed time**

The Performers: Agents for Service Composition

Conceptual idea of service mediation is not originally new and has been argued by many authors. Strong mediation has been for instance claimed as one of the basic principles for WSMF[4].

The agents of the RACING mediator and the agents that wrap their information resources collaboratively participate in performing business processes of information retrieval and information fusion by providing their web services in a proper composition. From a user point of view, who doesn't see the cooking, these business processes are simply the web services provided by the mediator. Going back to the kitchen, the following features are essential for an intelligent service provider:

- Have appropriate **formal representation** of the **semantics of the services** it is capable to perform (Task Ontology[5] in RACING)
- Be capable to pro-actively **adjust service inputs**, assess requestor's **preferences** and **constraints** (incremental user profiling and ontology-driven query transformation in RACING)
- Be capable to **negotiate** in a rational way **on optimal service provision** and **sub-service outsourcing** (Contract Net Protocol and Negotiation Ontology in RACING)
- Be capable to **monitor and assess** the **capabilities** and the **credibility factors** of other service providers (reinforcement learning technique in RACING)
- Be capable to dynamically **plan and coordinate** the **service execution flow** (Partial Local Plans and Coordination Agent in RACING mediator MAS)

[3] http://www.zsu.zp.ua/racing/ - the project is supported by Ukrainian Ministry of Education and Science, Grant No 0102У005339.

[4] Fensel, D., Bussler, C. The Web Service Modeling Framework WSMF. White Paper http://www.cs.vu.nl/~swws/download/wsmf.paper.pdf, last accessed May 3, 2003

[5] Please refer to: Ermolayev, V. Keberle, N., Tolok, V. (2002) OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors. In: H. Arisawa, Y. Kambayashi, V. Kumar, H.C. Mayr, I. Hunt (Eds.):Conceptual Modeling for New Information Systems Technologies ER 2001 Workshops, HUMACS, DASWIS, ECOMO, and DAMA, Yokohama Japan, November 27-30, 2001. Revised Papers - LNCS vol. 2465, p. 390-402

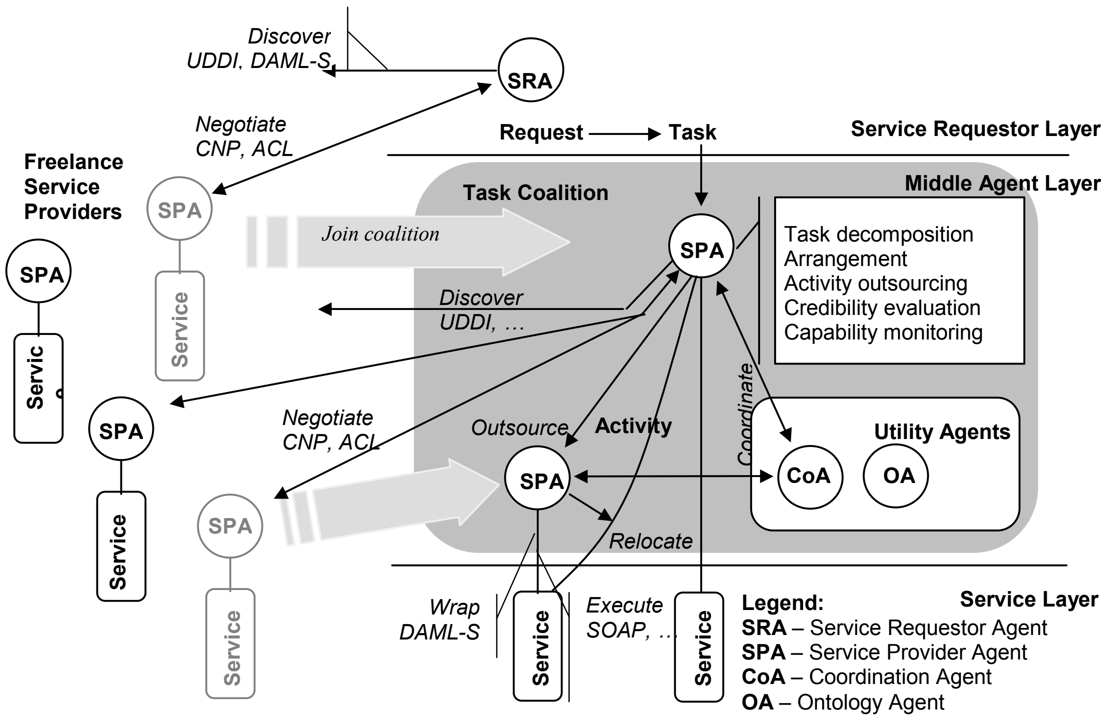


Fig. 1. Agent-enabled web service orchestration. A request for a web service is processed as the task to perform the business process of web service orchestration by the coalition of freelance Service Providing Agents (SPAs). The compositional notation is provided by the RACING Task Ontology (DAML+OIL). SPAs join the coalition through negotiation on service provision.

The architectural blueprint of the RACING agent-based web service orchestration framework is shown on Fig. 1. This orchestration is driven by the compositional notation of the RACING Task Ontology. Task Ontology formalizes the partial local knowledge of service providing agents about the Request-Task-Activity-Service hierarchy of the business process they participate in.

The “Score”: Request-Task-Activity-Service Hierarchy

Request-task-activity-service semantic hierarchy reflects the principles of the mentioned architectural layering (Fig. 1.). A request belongs to the sphere of Service Requestor Layer and is specified in terms of the Task Ontology.

The function of the SPA chosen as the contractor for the specified request is to determine if the incoming task is the atomic activity according to its local knowledge (Partial Local Plan). In case the task is complex it is decomposed into atomic activities at the local level of granularity of the given SPA. The next round of negotiations may be initiated for the part of the “summoned” activities. The negotiation set is the set of the activities to be outsourced. Negotiation participants are the SPAs about which the initiator believes that they are capable to perform the activities from the negotiation set.

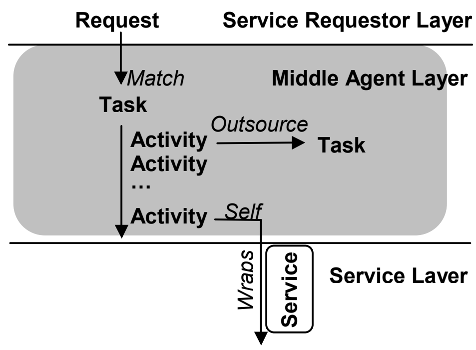
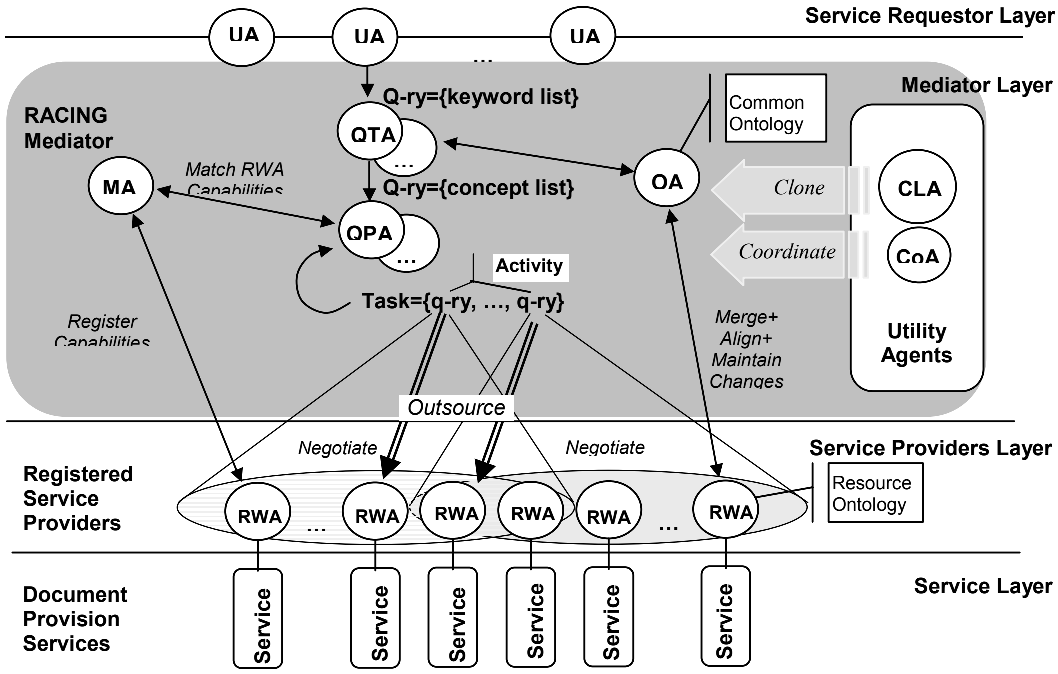


Fig. 2. Request-task-activity-service semantic hierarchy.

Only the activity for which it is true that: a) it is atomic and b) the SPA is capable to perform it on its own – is in relationship with the relevant service or service loop. Atomic activity execution is performed by SPA by invoking its capability method: activity description is translated into DAML+OIL markup corresponding to Service Profile; the wrapped service is than invoked via the interface specified by its binding (or grounding in terms of DAML-S) description. Service invocation loop may actually result in one or several service calls depending on the wrapping activity inputs.



Legend:
UA – User Agent, **QTA** – Query Translation Agent, **QPA** – Query Planning Agent,
RWA – Resource Wrapper Agent, **OA** – Ontology Agent, **MA** – Matchmaking Agent,
CLA – Cloning Agent, **CoA** – Coordination Agent

Fig. 3. RACING Architecture.

The RACING Orchestra: Mediating Document Retrieval

In the field of document retrieval a service request is traditionally presented in the form of a search phrase – a first order logic predicate over the list of keywords or phrases. Documents (web pages, scientific papers, magazines, books) are stored at disparately structured, distributed, autonomously maintained databases or text collections in a digital form. These document sources are marked-up according to different standards, belong to different legal entities and **often cost money**. A task for document retrieval may be presented as the composition of interrelated activities. These activities are derived from the initial user's request.

User request processing, resource wrappers registration by capability matchmaker and common ontology maintenance are the basic functionalities of RACING mediator (Fig. 3.). Though only query processing may be considered as a real business process involving third-party service providers for money, the other two ones are also performed as tasks and require various types of negotiation and semantic interoperation.

Let's discuss RACING query processing scenario to have more details from inside the process. Wrappers registration and Ontology maintenance scenarios are orchestrated with the same principles. Their discussion is omitted for the sake of saving paper space.

RACING: User Request Processing

The process starts at UA with formulation of the query in terms of the key phrases familiar to the given user. UAs are cloned by CLA utility agent each time a new user comes to the mediator and perish when the user leaves. Information about user preferences (mapping of his/her/its, if a software agent, most frequently used key words to Common Ontology concepts) is incrementally collected, stored at OA in the form of the User Profile reference ontology and is used by the QTA for the query transformation. UA conducts the task of query processing and acts as the proxy between the user and the mediator. Query processing task contains 'CloneQTA', 'TranslateQry', 'CloneQPA', 'ExecuteQry' activities. The cloning activities are allocated to CLA, which clones the QTA for the query transformation and the QPA for query processing. 'TransformQry' activity is allocated to QTA, which performs the transformation of the query predicate in terms of keywords to semantically equivalent query predicate in terms of the concepts of mediator's common ontology. The last activity is allocated to QPA, which generates the following set of activities for 'ExecuteQry' task: 'DecomposeQry', 'PerformQryset'. Query decomposition is performed by QPA in order to extract the parts of the incoming query, which may require different capabilities from document service providers. This extraction is guided by topic classification of the common mediator ontology. Resulting set of partial queries is performed by QPA as the following activity sequence: 'MatchRWA', 'PerformQry'. Matching activity is allocated to MA for a certain incentive over accomplishment time. MA returns the list of RWAs capable to perform document providing services relevant to the partial query. 'PerformQry' activity outsourcing is negotiated with pre-selected RWAs in the terms of service 'overheads' over time and document price. The contractor for the query performance is chosen by the results of this negotiation. Contractor RWA receives the partial query in terms of Common Mediator Ontology. It therefore needs to translate the query into the terms of its Resource Ontology. This translation activity is outsourced to OA. RWA than invokes document service it wraps with the translated query and returns the (list of) URIs of the documents relevant to the query to QPA.

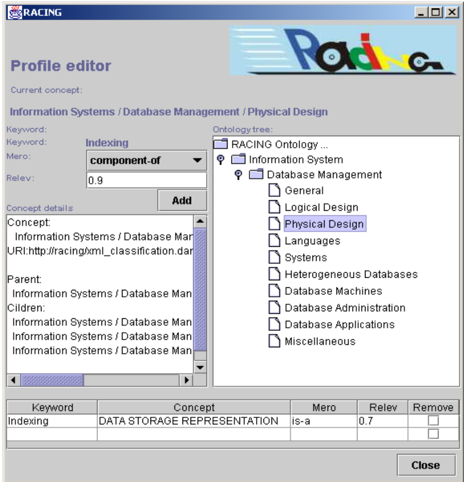


Fig. 4. RACING User Profile Editor – the interface for ST3 of the Query Transformation Service

More Details on the Semantic Aspects

User request processing in RACING is the ontology-driven process from the beginning to the very end. The paper space doesn't allow to discuss how the domain semantics guides the process at every stage. However, at least one step of the dynamic business process should be examined from this angle. One of the tricky phases of user's query processing is its transformation to the search predicate built of the concepts of the mediator's common ontology.

The transformation methodology[6] is based on incremental user profiling. The mapping of a user's keywords to the concepts of the domain ontology is built according to the transformation rules. These rules are based on the usage of the rich set of the semantic relationships comprising subsumption, synonymy, instantiation and meronymy provided as the DAML+OIL ontology[7].

The test multi-agent system comprises two FIPA-compliant[8] agents: User Query Transformation agent (QTA) and RACING mediator Ontology Agent (OA). QTA is the agent which has direct contact to the user and performs the query transformation. The user interfaces of QTA are shown on Fig. 4 and Fig. 5. One of the major functions of

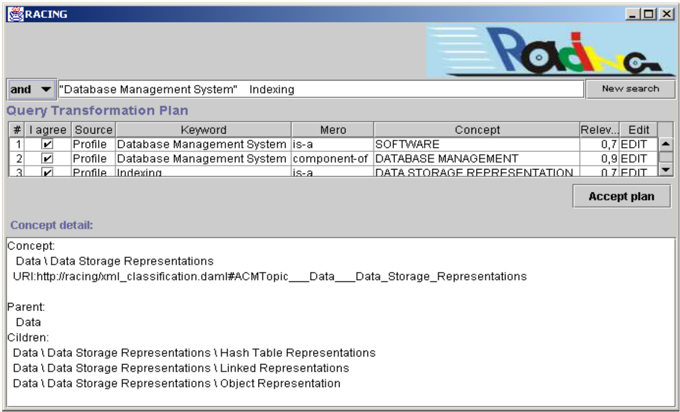


Fig. 4. The interface for the Query Plan refinement and approval service.

[6] For the details please refer to: **Ermolayev, V., Keberle, N., Plaksin, S., Vladimirov, V.** Capturing Semantics from Search Phrases: Incremental User Personification and Ontology-Driven Query Transformation. Submitted to: 2-nd Int. Conf. on Information Systems Technology and its Applications (ISTA'2003), Kharkiv, Ukraine, June 19-21, 2003, http://eva.zsu.zp.ua/eva_personal/PS/EVA-ISTA-03-draft.pdf

[7] http://eva.zsu.zp.ua/eva_personal/ontologies/racing-meronymy.daml

[8] Implementation platform is FIPA-OS: http://fipa-os.sourceforge.net/

the OA is mediator knowledge base management. OA supplies QTA with the contents of the user profile, RACING Meronymy ontology and the required portions of the domain mediator ontology (ACM Topic in our current implementation). Though the current test implementation uses the specific taxonomy as the domain ontology, it is evident that the methodology is ontology invariant. Any other widely recognized ontology[9] may be incorporated into the mediator knowledge base due to the import facility of the OA. Moreover, the incremental profiling technique may provide valuable feedbacks for the enrichment, revision or harmonization of the domain ontology. The refined ontology may be than exported by the OA and made publicly available.

Concluding remarks

This is the very short and high-level report on the RACING approach to web service orchestration on the next generation Semantic Web. Though the focus of the project is intelligent document retrieval, the approach and the methodology are applicable to the variety of application domains, for example Supply Chain Management, Workflow Automation, Enterprise Application Integration. From our point of view, the methodology provides both essential components for dynamic rational intelligent web service orchestration – the compositional notation (the recipe) and the Service Providing Agents as intelligent executives (the personnel at the kitchen).

Acknowledgements

The article is based on the working drafts, the technical reports of the RACING project[10] team[11]. The results of the discussions in the Agents research group of Zaporozhye State University have also been used. I'd like to express my special thanks to the people from the team, who substantially contributed to the project: Natalya Keberle, Sergey Plaksin, Vladimir Vladimirov.

[9] Coded in DAML+OIL
[10] http://www.zsu.zp.ua/racing/
[11] http://www.zsu.zp.ua/racing/list/e-people.htm