

# Proactively Composing Web Services as Tasks by Semantic Web Agents

Vadim Ermolayev and Natalya Keberle, Intelligent Systems Research Group<sup>1</sup> (ISRG), Zaporozhye National University, Ukraine

Oleksandr Kononenko, Nokia Research Center, Helsinki, Finland

Vagan Terziyan, Industrial Ontologies Group<sup>2</sup> (IOG), University of Jyväskylä, Finland

## ABSTRACT

*The chapter presents the framework for agent-enabled dynamic composition of Semantic Web Services. The approach and the framework have been developed in several research and development projects by ISRG and IOG. The core of the methodology is the new understanding of a Semantic Web Service as a capability of an intelligent software agent supplied with the proper ontological description. It is demonstrated how diverse Web Services may be composed and mediated by dynamic coalitions of software agents collaboratively performing tasks for service requestors. Middle Agent Layer is introduced to conduct the transformation of a Web Service request to the corresponding task, agent-enabled cooperative task decomposition and performance. Discussed are the formal means to arrange agents' negotiation, to represent the semantic structure of task-activity-service hierarchy and to assess fellow-agents' capabilities and credibility factors. It is argued that the presented technique is applicable to various application domains. Presented is the ongoing work on designing and implementing agent-based layered architecture for intelligent rational information and document retrieval. Finally, the discussion of the OntoServ.Net framework for the development of P2P mobile service infrastructures for industrial asset management provides the extension of the Web Service composition approach.*

*Keywords:* asset management; composition; information retrieval; ontology; Semantic Web Agent; Semantic Web Service.

Semantic Web Services are the emerging technology promising to become one of the future 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 be the key elements in assembling intelligent software infrastructures in the near future.

There is the emerging consensus that the ultimate challenge is to make Semantic Web Services automatically tradable and usable by artificial agents in their rational, proactive interoperation on

the next generation of the Web. It may be solved by creating frameworks, standards, and software for automatic Web service discovery, execution, composition, interoperation and monitoring (McIlraith et al., 2002). Personal opinion of the authors is that the list should be extended by the means making services the subject of automated negotiation and trade. It is also important for future service enabled Semantic Web infrastructures to cope with business rules<sup>3</sup>, the notions and mechanisms of reputation and trust with respect to services and service providing agents, dynamic character, flexibility, reconfigurability of partial plans (Ermolayev and Plaksin, 2002), workflows, and modeled business processes.

Current industry landscape provides only initial and very partial solutions to the ultimate problem. Existing de-facto standards for Web Service description (WSDL), publication, registration and discovery (UDDI), binding, invocation, and communication (SOAP) provide merely syntactical capabilities and do not fully cope with service semantics. Known industrial implementations, such as HP E-speak (Karp, 2003), base on these standards and do not completely solve the challenge of semantic interoperability among Web Services. It should be mentioned that major industrial players realize the necessity of further targeted joint research and development in the field (Layman, 2003).

More recent research and standardization activities of DARPA DAML community resulted in offering semantic service markup language DAML-S (Ankolekar et al., 2002) based on RDF platform. This initiative has later resulted in the development of OWL-S – the successor of DAML-S based on the World Wide Web Consortium's Web ontology Language (OWL). The constellation of XML based languages/ontologies for business process description is also expanding: WSFL, ebXML, BPML, RuleML, BPEL4WS ...

This chapter is the reworked and extended version of (Ermolayev et al., 2004). It is intended to highlight what should be still done on the top of recent research findings in order to make Web Services automatically tradable and usable by Semantic Web Agents in their rational, proactive interoperation on the next generation of the Web. Conceptual frames for this development are under intensive discussion and some proposals already appear (for example WSMF (Fensel & Bussler, 2002)). Recent research activities of the Semantic Web community in Europe and Americas in several projects clearly demonstrate that the problem of dynamic composition (or orchestration) of Semantic Web Services is one of the mainstream tasks. The approaches to solve this problem differ. However, the common trend is to use a goal-directed technique as it is done in distributed problem solving and dynamic distributed planning. For example DIP project<sup>4</sup> proposes mediation approach for goal-directed Semantic Web Service orchestration using goal mediators, DIP Interface Description Ontology based on WSMO (Roman et al., 2005). The

tendency which one can smell in the air is the search for a synergetic technology enhancing Semantic Web Services with the ability to demonstrate a kind of a goal-directed, i.e. proactive behavior. A natural locus of such a capability is known as an intelligent software agent.

The chapter offers a new understanding of a service as an intelligent agent capability implemented as a self-contained software component. From the other hand, provided that agents negotiate and trade exchanging services in the process of their cooperative activities in open organizations, a service may be considered (as, say, in E-speak) a kind of a generalized resource. This approach evidently implies the appearance of a rational service providing agent demanding certain incentives and aiming to increase its utility. If for example a service requested from a travel agency is ‘BookRoundtrip(‘Kiev’, ‘Erfurt’, 22/09/2003, 25/09/2003, ...)', the price paid by the requestor will comprise the prices of consumable (OWL-S, 2003) resources (air fare, hotel room, ...) plus the incentive paid to the service holder for ‘BookRoundtrip’ service component usage. This remark seems to be rational as far as we pay either the salary to the office manager or a fee to a travel agent, who make arrangements for us in a human-business environment. Moreover, it is not in the eye of the service requestor, but the agent performing ‘BookRoundtrip’ service will realize according to the service markup or the Partial Local Plan (PLP) in our terminology (Ermolayev et al., 2001; Ermolayev et al., 2005) that the requested process (OWL-S, 2003) or the task in our terminology (Ermolayev et al., 2001; Ermolayev et al., 2005) is composite and will require cooperation with at least air companies’ service providing agents and hotel booking service providing agents. These independent actors will evidently also intend to increase their own utilities by requesting fees for their services.

The chapter first provides the overview of the basic notions, approaches and architectural solutions with respect to agent paradigm, WWW and the Semantic Web, Semantic Web Services. Detailed discussion of the popular travel planning scenario helps to claim that full-scale Web Service exploitation needs solutions beyond the facilities of today’s semantic service markup. The chapter focuses on one of the major open problems – dynamic composition of a desired complex service by a coalition of rational cooperative freelance software agents.

Next it is argued that it is a reasonable architectural solution to introduce an Agent Middle Layer (for example (Sycara et al. 1999)) between services and service consumers. Negotiation on Web service allocation based on the authors’ approach (Ermolayev and Plaksin, 2002) is proposed as the mechanism for dynamic composite service formation. OWL-S (OWL-S, 2003), our *Task and Negotiation Ontologies* (Ermolayev et al., 2001; Ermolayev et al., 2005; Ermolayev and Keberle, 2006) are used for service dynamic composition and to facilitate inter-agent-operability.

Further on it is described how the approach to dynamic agent-based service composition is applied to intelligent rational information retrieval from distributed autonomous resources. Finally, the OntoServ.Net (Kaykova et al., 2004; Terziyan, 2005; Terziyan and Kononenko, 2003) framework and the aspects of service mobility and service adaptation are discussed. The architectural principles for service composition in a peer-to-peer service network are also outlined.

## WHAT AN AGENT IS?

**Agent paradigm** in software engineering is one of the powerful means to narrow the semantic gap between the conceptualizations we use to analyze and to model the phenomena of the real world and the resulting distributed software system. If compared to the objects in OOSE, which may be interpreted as the analogy of inanimate entities in the real world, agents generally represent sentient animate objects, able to cognize, typically human beings. Intelligent software agents are therefore used when the software needs to possess some “human” features like the ability to perceive the environment and reactivity, apparent proactive behavior in succeeding at a goal on behalf of the human owner, ability to learn from their experience, and social behavior. One of the inherent intelligent features of agents is the ability to form social structures – teams, communities, coalitions, and organizations. A **rational agent** as the member of a social structure needs to balance its **individual rationality** and **benevolence** in facilitating to the growth of the group utility. Agents often use **negotiation** mechanisms adopted from human encounters for that. An agent also needs to obey its social commitments and the conventions that regulate group behavior within the social structure. A team or an organization of agents that cooperate in a physically and, possibly, geographically distributed network form a software system called a **Multi-Agent System (MAS)**. An agent and a MAS are the main conceptual patterns of Agent Oriented Software Engineering (**AOSE**). From the engineering perspective at the lower level of abstraction the essential features of agents in a MAS are their ability to communicate with each other and to coordinate their activities. Coordination means achieving coherence in the group activities and thus providing that the solution of a problem or the accomplishment of a task is obtained with less effort, fewer resources consumed, and better quality. Communication stands for the ability to exchange the pieces of information within an encounter in a uniform way and using shared terminology. Communication among agents in open systems, which are typical in the majority of real world cases in e-business, enterprise application integration, and so forth, is a challenging interoperability task. The solutions are approached by standardizing the languages for communication (for example FIPA ACL) and developing formal machine-processable representations of the common terminology in the form of ontologies. Ontologies, formalized in

ontology description languages (for example OWL) provide: a conceptualization – a formal model of real world phenomena in a domain; a vocabulary – a set of terms or symbols identifying concepts; and an axiomatization – the rules and the constraints on concepts and their properties that capture the characteristic aspects of a domain.

Agent paradigm and AOSE gain more and more popularity as one of the key enablers of the emerging Semantic Web – the new generation of the Web. The abstract architecture of the Semantic Web is outlined in W3C<sup>5</sup> WWW Technical Architecture Group (TAG) Architecture Specification.

More details may be borrowed from, for example, (Jennings, 2000), (Ermolayev and Plaksin, 2002).

## W3C WWW ARCHITECTURE

**WWW Architecture** provides the abstract specification of the architecture of the Web. It figures out the conceptual model, the properties and the semantics of WWW constituents and defines the underlying principles and the basic constraints of Web-based system development. WWW architecture specification fixes the design choices approved by W3C and approves the good practices of using Web technology that guide future growth, consistent and successful evolution of the Web.

The primary task of **W3C TAG** is to develop and maintain the consensual specification of the basic principles of the Web technology in order to facilitate and coordinate cross-technology architecture developments inside and outside W3C. TAG claims **identification**, **interaction**, and **representation** as the key aspects of Web architecture and derives its abstract specification from these concepts.

**Identification** on the Web is based on the semantics and the use of the Uniform Resource Identifiers (URIs) which are global identifiers and are central to the architecture of the Web.

**Interaction** is defined by TAG as the communication of resources that involves URIs, messages, and data among agents over WWW. TAG provides the basic concepts for messages, Web agents, interaction styles, and the use of metadata and the protocols for agents. TAG also defines the architectural constraints and the assumptions for agent interaction and the patterns for human user interaction on the WWW.

**Representation** of data on the Web is grounded on the defined concepts of media types, data formats, encoding, namespaces, general hypertext infrastructure and the use of XML as the core

language. It is worth mentioning in the context of the representation aspect that the **representation of metadata** on the Web is not explicitly defined by the Web architecture specification yet and is likely to be based on the Semantic Web principles for the next generation of the Web.

The **Separation** of Content, Presentation, and Interaction is yet one more important principle of the WWW architecture. It concerns the development of the standards for highly interoperable distributed systems in open and dynamic environments, where information is created, accessed and processed at the high level of autonomy with respect to the capabilities and the heterogeneity of the Web agents involved.

## **W3C WEB SERVICE ARCHITECTURE & THE SEMANTIC WEB**

**Web Service Architecture** specifies generic concepts and defines the framework for the creation of Web Services. Web Services are modular software components accessible over a WWW. A Web Service is supplied with the description specifying its interface in a machine-processable way to provide for the interoperability in open distributed software systems. These descriptions contain the specifications of the message formats, data types, transport and serialization protocols.

The following de-facto industrial standards outline today's technological frames for Web Service development and publication: WSDL – Web Service Description Language, UDDI for Universal Description, Discovery and Integration, SOAP (Simple Object Access Protocol) for Web service binding and invocation, and XML and HTTP for serialization. However, ongoing research activities push forward the state-of-the-art by developing an extensible ontology-based framework for Semantic Web Services.

**W3C Semantic Web Initiative** aims primarily to provide a comprehensible framework for identifying, representing and processing the semantics of WWW resources. The ultimate vision of the Semantic Web is the worldwide distributed device for computation, inhabited with artificial service providing agents. It is therefore extremely important to have Web Service semantics formally and explicitly represented in a machine-processable way. Such semantic representations in the form of ontologies are essential for automated service discovery, invocation, orchestration, and trade and evidently extend today's technological frames. Semantic Web resources and services will have semantic annotations – small ontologies providing both a meta-description of the resource and the vocabulary of the relevant concepts. Semantic Web initiative spends substantial effort on ontology language (RDFS, DAML, OWL) development

and standardization.

## **OWL-S & SEMANTIC WEB SERVICES**

The concept of Semantic Web Services (SWS) is the synergy of Web Service technologies with the Semantic Web framework. It assumes that the Semantic Web infrastructure is the top layer of the conventional WWW. This semantic layer contains Web Service ontologies, notations and standards for service description, facilities for service discovery, orchestration and integration. SWS will be widely used in the future Web, where intelligent agents will discover Web service providers, reason about their capabilities by analyzing their semantic descriptions and dynamically compose services on demand in cooperation with the service providing agents having appropriate capabilities.

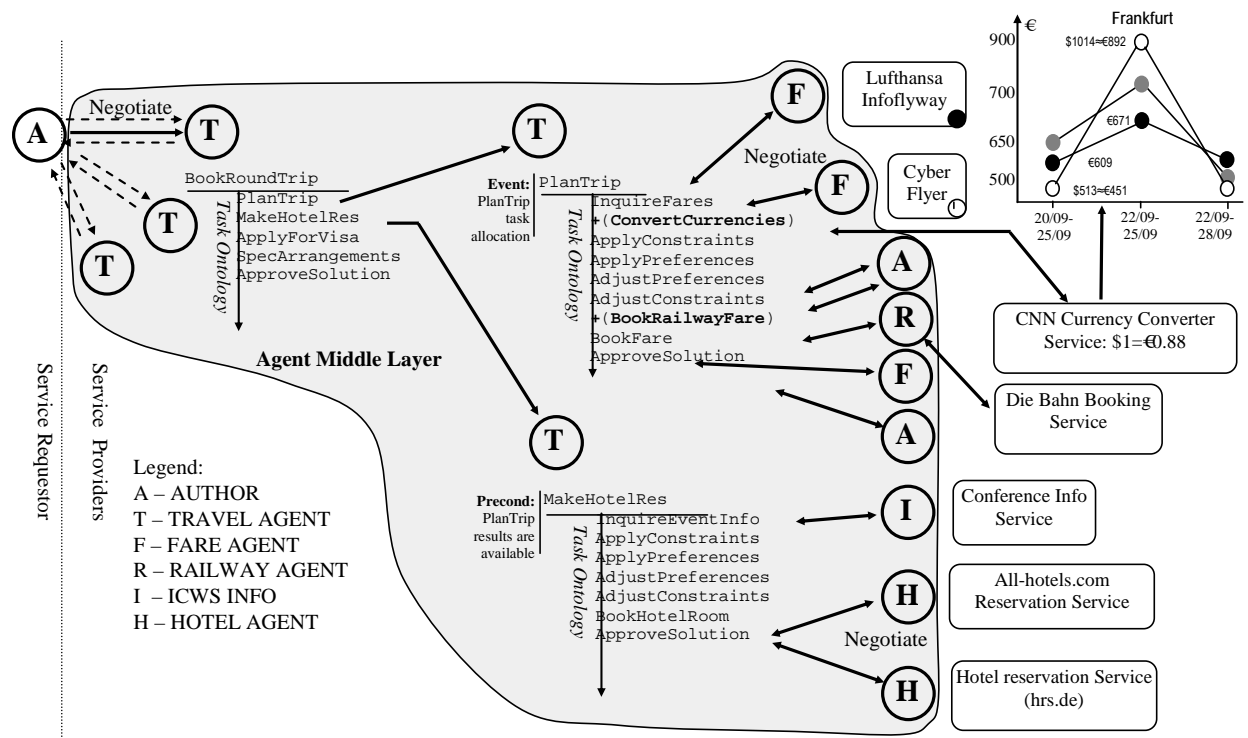
One of the pioneering targeted SWS initiatives is the development of **OWL-S** (OWL-based Web Service ontology). OWL-S is the extension of OWL ontology language. It specifies the core set of concepts for describing the granularity, the properties, the capabilities and the grounding of a Web Service. If compared to current industry standards, OWL-S provides a higher degree of flexibility and expressiveness in describing service semantics, allows to model extensible service hierarchies and type systems, and provides the means for specifying the constraints and the rules for Web Services.

## **TRAVEL PLANNING SCENARIO**

Let's consider the mentioned travel planning scenario having in mind that our intentions have become true and Web Services are available at the desired level of semantic interoperability. The authors have played the following exercise assuming themselves as "intelligent software agents" participating in cooperative execution of a conference trip planning task (Figure 1). Each agent possessed his or her beliefs about the environment, and capabilities in performing one or another activity related to the overall high-level goal achievement – 'BookRoundtrip("Kiev, Ukraine", "Erfurt, Germany", 22/09/2003, 25/09/2003, "ICWS'03-Europe", ...)'. Agents' capabilities were: their knowledge of relevant Web sites providing human-oriented services and their ability to operate these services via Web interfaces. Agent roles were:

- **AUTHOR (A)** – an agent representing one of the research paper authors intending to attend ICWS'03-Europe and requesting 'BookRoundtrip' service
- **TRAVEL AGENT (T)** – an agent actually providing 'BookRoundtrip' service by generating and conducting corresponding task execution

Figure 1: "BookRoundTrip" task execution and service composition



- FARE AGENT (F) – agents providing various airfare information and booking services
  - ICWS INFO (I) – an agent providing information services on ICWS'03-Europe local arrangements, infrastructure, accommodation, and so forth at Erfurt
  - HOTEL AGENT (H) – agents providing hotel room reservation services
  - BUSINESS PARTNER (P) – an agent representing A's business partner in Austria with whom A intends to meet in Germany at the time of the conference to discuss a joint proposal
- As usual in travel planning A is capable of just invoking a T with 'BookRoundtrip' task, to formulate his or her constraints, preferences and needs for special arrangements, and to approve the solutions proposed by the chosen T. According to 'BookRoundtrip' description in terms of *task ontology* (Ermolayev et al., 2001) known both to A and T (but with different granularity), service inputs are<sup>6</sup>:

Starting\_Point= "Kiev, Ukraine"

Destination="Erfurt, Germany"

Beg\_Date =22/09/2003

End\_Date=25/09/2003

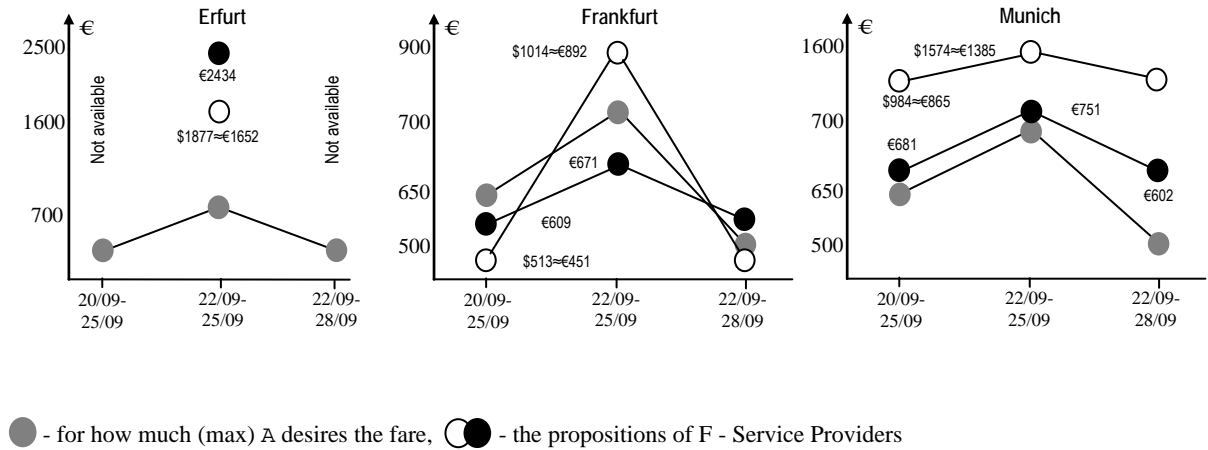
Event="ICWS'03-Europe"

Preferences=("low fare", "fast connections", "4-star hotel", "continental breakfast", "conference discounts")

Constraints =(Budget = €1500, Payment=(VISA, USD), Hotel >= 3-star, Room-per-night



Figure 2: Fare desirability function and service propositions



<=€110, Hotel\_Location="in Max 20 min walk from the Conference venue")  
 Special\_Arrangements=( ( Event="business dinner", Agent = ("Prof. Heinrich C. Mayr",  
<http://www.ifi.uni-klu.ac.at/IWAS/HM/Staff/Heinrich.Mayr/>), Date=(23/09/2003-  
 24/09/2003), Location=(Erfurt, Munich))

The process starts with the arrangement (Ermolayev and Plaksin, 2002). A undertakes to hire one of the Ts as the contractor for the job. This arrangement is performed in frame of the Extended Iterative Contract Net negotiation. The flow of round trip booking, which T performs for A, is presented in Figure 1. At first, T accepts the task from A using agents' communication interface. This interface is based upon ACL (FIPA, 2003) for FIPA<sup>7</sup>-compliant agents (e-Appendix A-1<sup>8</sup>). T then uses its beliefs on how to 'BookRoundtrip' (e-Appendix A-2), formalized according to the task ontology (e-Appendix A-6), to derive that the accepted task is complex and involves at least 'PlanTrip', 'MakeHotelRes', 'ApplyForVisa', 'SpecArrangements' and 'ApproveSolution' activities. 'PlanTrip' activity is chosen as the first one to be performed (PLP of task ontology (Ermolayev et al., 2001)) and appears to be also a complex task: 'InquireFares', 'ApplyConstraints', ..., 'BookFare', 'ApproveSolution'. Before outsourcing fare inquiry to F, T 'notices' that a slight change in the starting or ending date of the trip may result in a substantial decrease in the airfare expenses because of the Sunday Rule discounts<sup>9</sup> commonly offered by air companies. In our example this means to T that the dates 20/09-25/09 and 22/09-28/09 should be also rationally considered for the trip. T negotiates these input changes with A, asking A to provide desirability values for these dates (Figure 2 – gray dots) indicating max price A is ready to pay for the fare within the specified dates. Requirements, which T specifies for 'InquireFares' service, are thus slightly changed by introducing the list of date pairs for which the service should be performed. Contract Net negotiation is then initiated by T having Fs as participants.

F-s propositions<sup>10</sup> resulting from ‘InquireFares’ service execution are also outlined in Figure 2. These results imply the necessity to use one more service, which was not initially planned by T’s PLP for the task. As far as the offers are provided in different currencies, T needs to change the task and to request the service for currency conversion<sup>11</sup> (+(‘Convert Currencies’, e-Appendix A-3), Figure 1). Conversion results are presented in Figure 2. It is now easy for T to derive that the acceptable proposition is still for the dates 22/09-25/09, but with the destination at Frankfurt (not at Erfurt) which were not the initial ‘BookRoundtrip’ task inputs received from A. However, this result complies with A’s preferences as far as there are nonstop flights available from Kiev to Frankfurt (but not to Erfurt and Munich). This implies the necessity for T to ‘AdjustPreferences’ by inquiring A’s service. The mechanism may be similar to inputs negotiation discussed above and the outcomes may cause the invocation of some new activities. For example change to a train at Frankfurt-Main Airport inquires the ‘BookRailwayFare’ service from Die Bahn<sup>12</sup> Agent. The discussion of these emerging task branches is omitted, as far as it is conceptually similar to that already given before. It is however important to notice that the activities that were not initially planned often emerge and appear to be critical to the overall goal achievement not only in the discussed scenario.

It is not informative to discuss subsequent activities of T. Hotel booking and visa application services are performed merely in the same manner and the agents use similar mechanisms of task decomposition and negotiation for that. Special arrangements list is also considered as the list of trip planning tasks. However it should be mentioned that the execution of these activities should be properly coordinated: for instance, hotel reservation requires that the fare has been already booked as the precondition (check-in and check-out dates, money left) and German Consular Service may require that the fare and the hotel room have been booked before issuing the visa.

Other important aspects not mentioned before are the ones of credibility, trust and meaning negotiation among agents participating in cooperative task performance and service composition. Recall special arrangements input for the illustration. T will negotiate with P on various aspects while arranging the business dinner. The dilemma for P in this case is if to trust T (as the contractor of A, which is the trusted one because of the long record of partnership) and allow him or her to make the arrangements for P, or to reason that A may be not really experienced in arranging business dinners in Germany and to decide to better rely on his or her credible partners from Germany. In the latter case P will inform T that it will better arrange the event on its own. This in turn may affect the necessity of the approval from A.

## **COOPERATIVE DYNAMIC SERVICE COMPOSITION**

Let's enumerate the features needed to rationally provide composite flexible services for the automation of the scenarios like that of travel planning in e-business settings. Intelligent service provider needs to be capable of:

- Understanding the semantics of the activity it is supposed to perform, reasoning on if the activity is atomic or complex, decomposing tasks according to his/her/its knowledge and experience;
- Adjusting activity inputs, requestor preferences and constraints in order to proactively reach the higher-level goal
- Negotiating with the requestor and the other service providers in a rational way on optimal service execution or allocation in order to increase its own utility or to obtain the common meaning of the service inputs, outputs, preconditions and after-effects
- Monitoring and assessing the credibility and the trustworthiness of the other service providers to minimize risks
- Coordinating service performance flow according to the inputs and preconditions

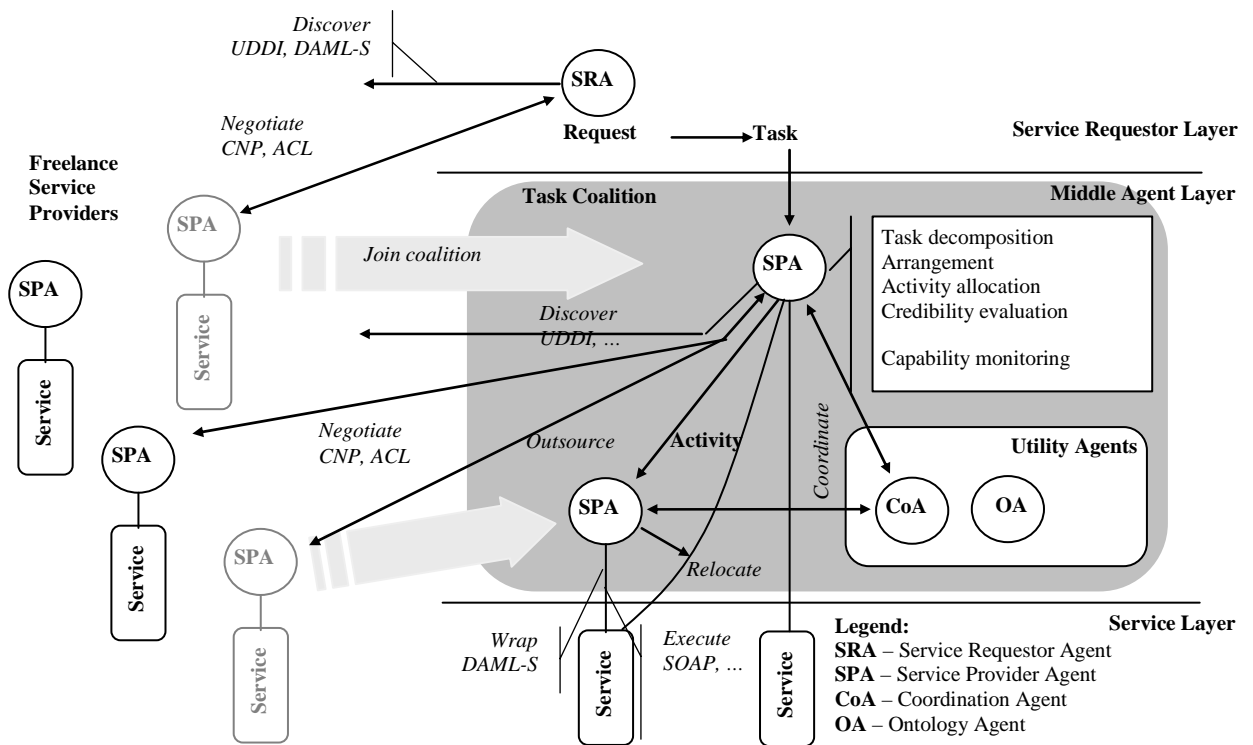
It seems obvious that service providing distributed open software systems possessing these capabilities may be most naturally designed and assembled of intelligent software agents. Software agent platforms and agent-based software systems are already used for service brokerage (McIlraith et al., 2002), matchmaking (Sycara et al., 1999), and coordination (Papadopoulos, 2001). The remainder of this section will shortly present the formal approach to dynamic task decomposition and performance by dynamically formed teams or coalitions of rational software agents (Ermolayev et al., 2001; Ermolayev and Plaksin, 2002).

### **Middle Agents for Service Composition**

The 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 of WSMF (Fensel and Bussler, 2002). However, the framework for intelligent dynamic service composition adaptable to the changes in the environment affected by the service execution flow has not been worked out before.

The proposal of the Mediation Framework for Agent-enabled Service Provision targeted to dynamic service composition is presented in Figure 3. Control flows are labeled with legends in italic; data flows are marked by bold legends. The principles on which the proposal is centered are:

Figure 3: Agent-Based Service Provision Mediation Framework



- Agent-based middle layer is required for scalable, intelligent, dynamic service composition
- Composite services are interpreted as tasks comprising activities of varying granularity by the agent middle layer
- Service mediator is formed dynamically as the coalition of service providing agents (SPAs) participating in the task execution
- SPAs join task coalitions only for the time their service is required for the respective task
- SPAs are economically rational (Nwana, 1996), autonomous and independent in making their decisions – the only fact one SPA believes about the behavior of another SPA is: it is at worst individual rational (Sandholm, 1996)
- SPAs are capable of: incoming task decomposition according to their local knowledge (task ontology, PLP); making arrangements for activity outsourcing to other SPAs based on extended iterative contract net negotiation; activity outsourcing to the chosen contractor SPA; adjusting their beliefs on other SPAs' capabilities and evaluating SPAs' credibility through monitoring cooperative activities
- Services are self-contained modular loosely coupled program components wrapped by SPAs; an SPA may allow another SPA to use its service by providing service context relocation

- Specialization of an SPA is defined by the set of services it wraps

If the framework is examined from the point of implementability with existing service markup solutions the state of affairs may look like given in Figure 3. Yet unsolved or partially unsolved problems of service mediation are:

- Lack of common semantic ground and commonly accepted mechanism for activity allocation, activity parameters adjustment and meaning negotiation – negotiation ontologies family

Research findings attacking this problem are published in (Ermolayev et al., 2005b; Ermolayev and Keberle, 2006).

- Insufficient representation of task/activity/service dynamic structure and granularity – task/process ontologies family

These ontologies has been further developed in PSI project (Ermolayev et al., 2006b).

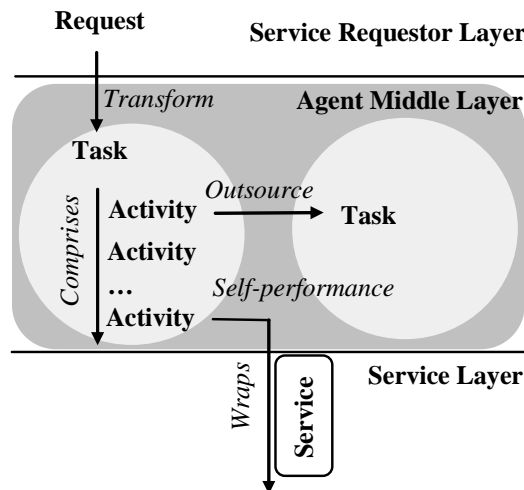
- Lack of common specifications/criteria for capability monitoring, credibility and trustworthiness assessment.

The proposed architectural layering is likely to remain valid for the request-task-activity-service ontology hierarchy: a service request is interpreted as the task at the requestor layer; these tasks are decomposed into activities at the middle layer; and activity descriptions wrap service markups. The remainder of the section provides some outlines to approach the solutions of the open issues.

### **Request-Task-Activity-Service Hierarchy**

The semantic hierarchy for a request-task-activity-service reflects the principles of the proposed architectural layering. A request belongs to the sphere of Service Requestor Layer and is specified in terms of *task ontology* (Ermolayev et al., 2001; Ermolayev et al., 2006b). 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 specifications. In case the task is complex and should be decomposed into atomic activities at the local level of granularity, the next round of activities allocation negotiations is initiated. Only the activities the given SPA is not capable to perform on its own are negotiated with other SPAs, while the ones corresponding to the initiator's capabilities are scheduled for self-performance. Only an activity for which it is true that: (a) it is atomic and (b) the SPA is able to perform it on its own, is in the relationship with the corresponding service or service loop. Atomic activity execution is performed by the SPA through invoking its capability model (Ermolayev and Plaksin, 2002): activity context is translated into OWL-S markup corresponding to the service profile; the service is then invoked

Figure 4: Semantic Layering



via the interface specified by its binding (or grounding in terms of OWL-S) description. Service invocation loop may actually result in one or several service runs depending on the wrapping activity inputs. For example ‘InquireFares’ service will be performed three times as far as three different date intervals are to be processed (Figure 2). Semantic facet of request-task-activity-service layering is presented in Figure 4. Specifications for ‘InquireFares’ activity and service are given in e-Appendix A-5.

### Capability & Credibility Assessment

An SRA and SPAs are to be able to determine which of the SPAs are capable to perform the task to be allocated. Possible mechanism to define the perspective contractors is capability matchmaking (for example based on LARKS (Sycara et al., 2002)), or service discovery technique based on UDDI, or another service matching facilities, for example semantic matching based on OWL-S profiles (Paolucci et al., 2002), or WSMO SWS descriptions (Roman et al., 2005). However, in case there is some capability beliefs record maintained autonomously by an SPA in the course of cooperative task execution, the use of this knowledge may substantially facilitate lowering computation costs by eliminating unnecessary directory/matching service usage. Evidently, if A believes that B, C and D are capable of performing desired activity because they did it before, it will rather proceed to contracting negotiation with B, C and D directly instead of trying to find some other SPAs<sup>13</sup> with matching capabilities.

A model and a mechanism of agents’ capability assessment based on SPA beliefs representation in the form of Fellows’ Capability Expectations Matrix (FCEM) has been elaborated in frame of the reported research (Ermolayev and Plaksin, 2002). SPAs accumulate and adjust their local

beliefs on the capabilities of their collaborators from the experience of cooperative performance. New portions of this knowledge appear each time an activity is being outsourced to an SPA. Subjective beliefs of the SRA on the probabilities of its fellows' capabilities to perform the given activity are thus updated. FCEM for capability beliefs representation is maintained in the following form:

$$\mathbf{C} = \begin{matrix} & a^1 & \dots & a^j & \dots & a^m \\ \text{SPA}_1 & \begin{bmatrix} c_1^1 & & c_1^j & & c_1^m \end{bmatrix} \\ \dots & \dots & & c_i^j = (q_i^j, p_i^j) & & \dots \\ \text{SPA}_n & \begin{bmatrix} c_n^1 & & c_n^j & & c_n^m \end{bmatrix} \end{matrix}, \quad (1)$$

where dimensions  $m$  and  $n$  change reflecting the appearance of new incoming activities and newly discovered or perishing SPAs.

Capability estimations  $c_i^j$  change each time an agent negotiates with its fellows on outsourcing an activity. Element  $q_i^j$  in tuple  $c_i^j$  stands for the quantity of recorded negotiations with fellow agent  $\text{SPA}_i$  concerning activity  $a^j$ . Element  $p_i^j$  stands for the capability expectation. The rule for  $c_i^j$  updates is as follows:

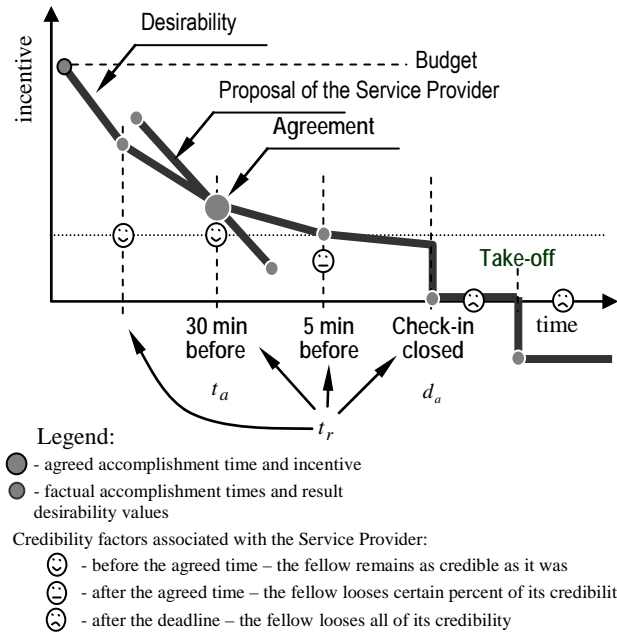
$$\begin{aligned} 1. \quad & p_i^j \leftarrow p_i^j + \frac{r}{q_i^j} \\ 2. \quad & q_i^j \leftarrow q_i^j + 1 \end{aligned} \quad (2)$$

where  $r$  is equal to: 0 – if the fellow rejected the activity, 0.5 – if the fellow replied that it can accept the activity and 1 – if the activity was finally allocated to the fellow.

One more aspect influencing a task requestor's decision to allocate an activity to one or another negotiation participant is its assessment of the participant's *credibility*. A self-interested SPA, due to the appearance of the new highly attractive activity offers in the competitive environment or due to the peculiarity of its behavior, may lower previously declared capacity (Ermolayev et al., 2001; Ermolayev and Plaksin, 2002) it is spending for the bulk of the activities under execution. This will lead to the increase of the performance duration, which may therefore seriously decrease the requestor's desirability of these results and, thus, lower the credibility value for the SPA selling its fellows short.

Let, for example a service outsourced to an SPA be 'DeliverAirTickets'. The result of the service is: the tickets are at the gate counter. The agreed delivery time is 30 minutes before the check-in, though the deadline advertised by the SRA before is the time when the check-in starts. The SRA will evidently consider the SPA that delivered the tickets before or right in the agreed time as credible. However, if the SPA delivers the tickets in five minutes before the check-in, the SRA

Figure 5: Activity Accomplishment Times and Corresponding Credibility Changes



may rightfully feel aggrieved, though it still has the chance to check in for the flight. The credibility of the SPA in the eye of the SRA will therefore be lowered. Further on, if the tickets appear at the counter after the check-in has been opened already, the SRA may rightfully consider that the contract terms were seriously violated by the SPA. Its credibility should be therefore drastically lowered. Finally imagine an SRA still waiting for its tickets at the counter when the plane is already taking off. In the latter case the SRA may even want to require a penalty in addition to lowering SPA's credibility to zero. To summarize, it is natural to measure the changes of an SRA's beliefs on the SPA's credibilities by the losses of the desirability of the service results based on the stricken contract deal (refer to Figure 5).

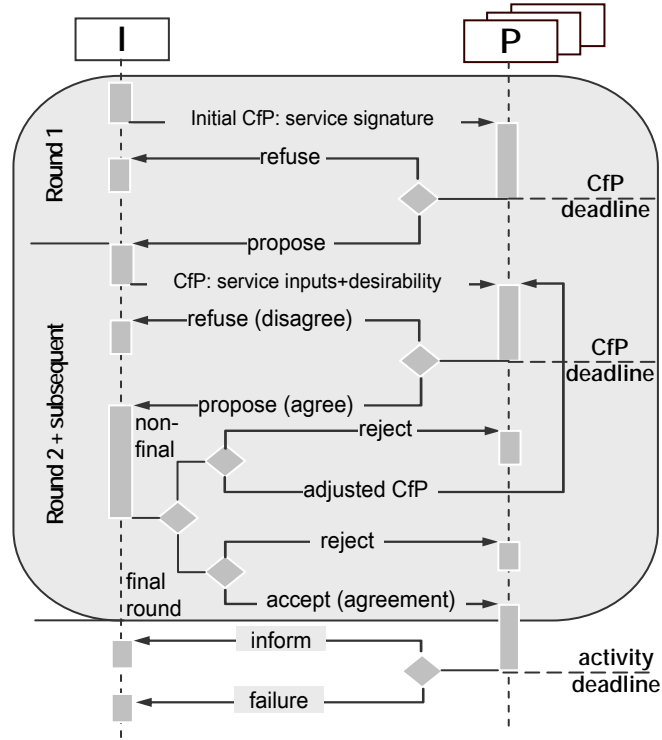
The mechanism of accounting fellows' credibility values is similar to that of adjusting the beliefs on changing fellows' capabilities (1-2). Credibility assessment values change over time as the requestor agent adjusts its subjective beliefs by comparing the desirability values (Figure 5) derived from:

- 1st - activity duration the executive committed to within the activity allocation arrangement negotiation and
- 2nd - actual results delivery time. Corresponding credibility matrix elements are then recomputed due to the following

$$C_{i,j} := C_{i,j} \times \begin{cases} 1, t_r \leq t_a \\ p_a(t_a/t_r), t_a < t_r \leq d_a \\ 0, t_r > d_a \end{cases} \quad (3)$$



Figure 6: Extended Iterated FIPA Contract Net Protocol



where:  $t_a$  is the time the parties have agreed to accomplish the activity  $a$ ,  $t_r$  is the actual time of  $a$  results delivery,  $d_a$  is the deadline and  $p_a$  is the weight coefficient characterizing the current priority of  $a$  for the activity requestor agent.

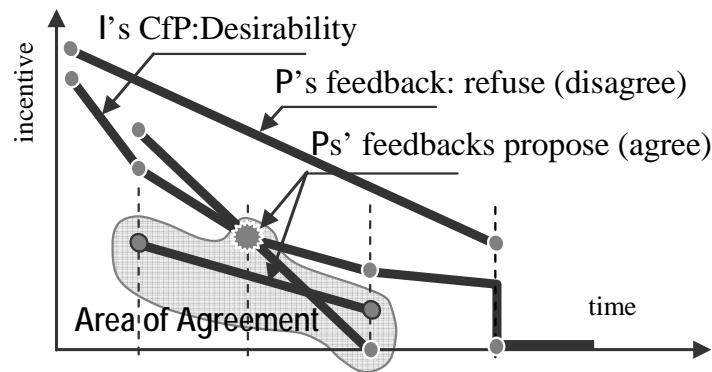
Credibility threshold values associated with respective activities and stored in agents' PLPs are used by task requesting agents to assess possible risks and alter their strategies.

### Negotiation on Activity Allocation

As it was mentioned above, negotiation on activity allocation takes place each time an agent realizes, according to its knowledge of the activity or because of the overload, that the activity should be outsourced to one of the fellow SPAs. An extension of the **FIPA Iterated Contract Net protocol** has been proposed as the interaction protocol for this kind of negotiation (see Figure 6). An SRA is considered an Initiator (**I**) in this encounter. The SPAs about which **I** believes that they are capable to perform the activity (FCEM) form the party of the invited Participants (**P**).

The first round of the interaction, which is actually the extension of the FIPA protocol, aims to find out if any of the known capable **P**s may agree to perform the activity. The negotiation set for this round contains activity signature only (for example 'DeliverAirTickets'). An **I** may start

Figure 7: Negotiation – Agreement and Disagreement



exploring other opportunities of outsourcing the activity if all **Ps** from the sphere of its awareness (Ermolayev and Plaksin, 2002) refuse in the first round. For example **I** may require the list of matching SPAs from the Matchmaker Agent (MA, see Figure 8).

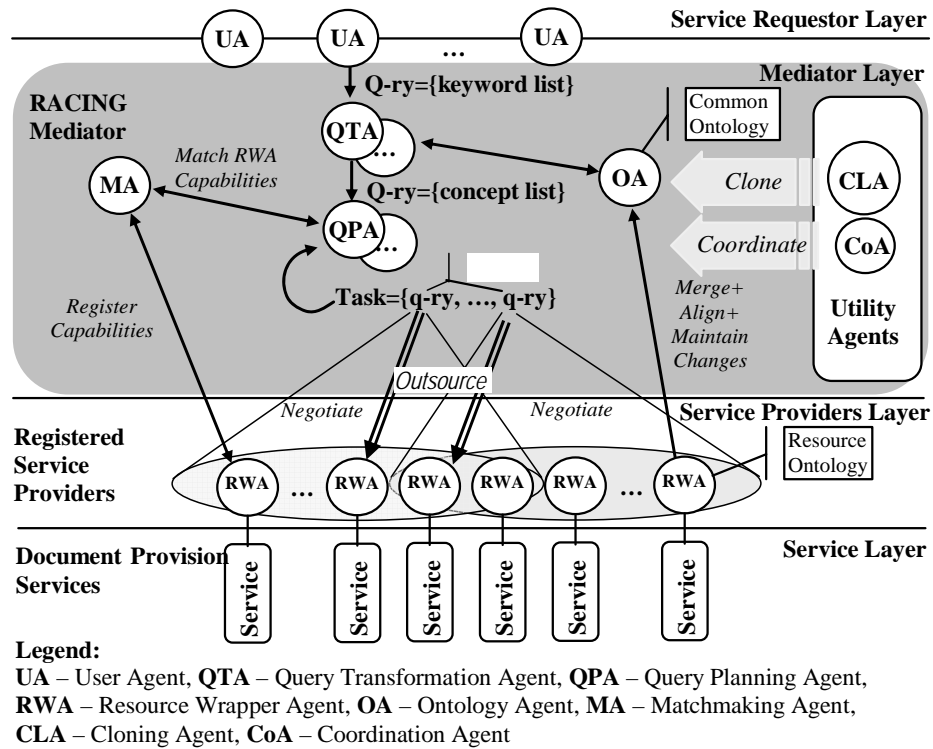
Negotiation on the second and the subsequent rounds is about the terms of the possible contract. An **I** advertises the activity inputs and the discrete results desirability function as the incentive over time. **I** then chooses the best **Ps** proposal weighted by the respective credibility values in case several **Ps** proposals result in the agreement. Subsequent rounds are used to adjust the activity inputs or the desirability function in the case if no one of the **Ps** has agreed on the previous round (for example dates or destination point in Figure 2).

**Ps'** refusals and propositions are shown in Figure 7. These feedbacks are formulated in a constructive way allowing **I** to adjust its CfP in the subsequent round. A feedback contains two incentive-time points defining the segment on which a possible agreement may be stricken. Evidently, the area of agreement for the current round could be formally defined as the union of all those parts of the feedback segments that are on and below the **I's** desirability function polyline. All other points of **Ps'** feedbacks indicate their disagreement with the offer in the current negotiation round.

An **I** considers the negotiation round as final if it can accept one of the **Ps'** agreement and strike the contract deal. The chosen **P** becomes the contractor and commits itself to the task coalition for the time necessary to perform the outsourced activity. Task coalitions are considered to be a kind of social structure. Coalition members are bounded with coalition commitments and convention regulating their ratios of self-interest and benevolence (Ermolayev et al., 2001).

*Negotiation ontology* (Ermolayev et al., 2001) is used as the namespace and the formal semantic frame for the contents of the messages agents communicate with while negotiating on activity

Figure 8: RACING Reference Architecture



allocation. Further development of the Negotiation ontology in PSI project<sup>14</sup> resulted in Generic Negotiation Ontology (Ermolayev and Keberle, 2006).

## RACING<sup>15</sup> FUNCTIONALITIES, AGENTS, & SERVICES

A reader might argue that, fairly, travel planning is not the task that really requires sophisticated agent-enabled automation technique: negotiations, coalitions, service wrapping and composition – at least from the customer’s side. Travel planning is not that time consuming to make its performance impossible without automation. Moreover, a human will sometimes be better in arranging loosely formalized things that require intuition and context-dependent understanding with the complexity beyond the capacity of, say, first order logic based languages. However, the presented technique is applicable not only in case you plan your conference trip (Ermolayev et al., 2001; Ermolayev and Tolok, 2002). The similar approach of dynamic cascading composition of activities and tasks has been further developed in PSI for the modeling an simulation of dynamic engineering design processes (Ermolayev et al., 2005; Gorodetski et al., 2005)

Let’s project the above discussion to distributed information and document retrieval domain. In the terms of document retrieval a service request is commonly formulated as a search phrase – a first order logic expression 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, are marked-up according to different standards and *often cost money*. A task for document retrieval may be presented as the set of interrelated activities distributed over the document providers. These activities wrap the (partial) queries derived from the initial user's request.

The goal of the RACING project is to provide mediation facilities for user query processing by the means of semantic decomposition of the initial query, rational distribution of the partial queries among independent, autonomous, rational document retrieval service providers wrapping respective document resources, and the fusion of the obtained results (Figure 8). User agents acting on behalf of the human users or real organizations (for example libraries) and service providing agents are considered as business representatives or business models in frame of the project. RACING mediation may thus be classified as B2B mediation. It is evident that such a kind of intelligent activities really needs sophisticated automation to be scalable and gracefully downgradable.

User query processing, resource wrappers registration by the capability matchmaker and common ontology maintenance are the basic functionalities of the RACING mediator (Figure 8). 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.

For example the outline for the user query processing scenario is as follows. The process starts at UA with the 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. User profiles (mappings of their most frequently used key words or phrases to the Mediator Common Ontology (MCO) concepts) are incrementally collected, stored at OA (Ermolayev et al., 2003) in the form of the reference ontology and used by QTAs. The UA actually generates and conducts the task of query processing and acts as the proxy between the user and the mediator. Query processing task generated by the UA contains 'CloneQTA', 'TransformQry', 'CloneQPA', 'ExecuteQry' activities. Cloning activities are outsourced to CLA, which clones QTA and QPA for query processing. 'TransformQry' activity is outsourced to QTA, which performs the transformation of the query in terms of keywords to semantically matching query in terms of the concepts of the MCO. The last activity is outsourced 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 that may require different capabilities from document service providers. This

extraction is guided by topic classification of the Common Ontology. The 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<sup>16</sup> the list of RWAs capable to perform document providing services relevant to the partial query. 'PerformQry' activity allocation is negotiated with pre-selected RWAs in terms of service 'overheads' over time and document price and the contractor is chosen for query performance. Contractor RWA receives the partial query in terms of MCO. It therefore needs to transform the query into the terms of its resource ontology. This transformation activity is outsourced to OA, which actually holds the necessary mappings. RWA then invokes document service that it wraps with the transformed query and provides documents relevant to the query to QPA.

RACING mediator has been partially implemented as a mock-up application. RACING framework for intelligent ontology-enabled distributed information retrieval has been further on elaborated in UnIT-Net<sup>17</sup> project (Ermolayev et al., 2004b). UnIT-Net research prototype has been implemented as the distributed system with the centralized mediator using semantically reinforced Web Services as intelligent Information Resource providers (Keberle et al., 2004).

## **SERVICE COMPOSITION IN P2P SERVICE NETWORKS**

One of the essential prerequisites for the implementation of a RACING-like service composition platform is the provision of the proper underlying infrastructure. It becomes even more important in the cases when the environment requires more sophisticated capabilities than those provided by the conventional WWW. This section presents the OntoServ.Net framework (Terziyan, 2005; Terziyan & Kononenko, 2003) for the intelligent composition of Web services on the Semantic Web enabled industrial environment. OntoServ.Net is the agent-enabled framework for the management of industrial devices in the peer-to-peer network of maintenance Web services. In OntoServ.Net the principles of the Semantic Web are used for the development of interoperable Web services and ontology-based information management. Peer-to-peer technology provides the means to organize the communication infrastructure, and agent technology enables the implementation of the problem-oriented behavior of network components (Terziyan, 2003).

OntoServ.Net is a fully decentralized environment that is a peer-to-peer network comprising service platforms located at maintenance sites and service providing centers. P2P structure of OntoServ. Net reflects existing approaches towards the creation of business-partnership environments where companies can share resources (in particular, Web services) that were previously used only internally. The enlargement of such resource sharing environments heads

towards a global P2P network with highly independent nodes. Though a semi-structured architecture will likely be used (with large service centers within newly created communities), peer-to-peer interactions reflect the reality of today's businesses.

The maintenance of complex industrial machinery, for example a paper mill, requires to control hundreds of factors and involves many services to monitor various sensor data, analyze general condition parameters, performance, etc. Hardware configurations vary from one machine to another, and thus, require an individual approach to the organization of the maintenance process and servicing.

The set of condition monitoring and maintaining services in OntoServ.Net is dynamically composed depending on the current needs of a machine. It changes when a fault state processing is required, or some service is substituted by the other one in order to provide more efficiency or to follow degradation processes along the machine's lifetime. OntoServ.Net service network improves performance and maintenance quality by providing the most appropriate services available on the network.

Recently the synergetic approaches to the design of service infrastructures combining the features adopted from the Semantic Web, Web services and P2P computing are under intensive research. Latest results prove good potential of such combinations for the cooperative use of distributed heterogeneous information sources and services (see for example Terziyan, 2003; Sivashanmugan et al., 2002). Service discovery and composition of Semantic Web Services in a decentralized network present new challenges for the research community and demand thorough study.

In addition to a P2P structure of the service network OntoServ.Net presents new aspects related to the service composition problem which were not thoroughly studied before: service mobility, individual rationality of SPAs and their intended readiness to cooperatively work in a P2P environment.

### **Service Mobility**

The specificity of the maintenance activities performed by services in OntoServ.Net requires that these services are mobile. The reasons are: a need for guaranteed service availability, a need for minimization of the communication traffic over the network during long-term servicing due to costs and/or technical restrictions, strict constraints for service response time, security and privacy issues, etc. Service mobility may naturally be implemented if the services are provided by mobile agents able to migrate between agent platforms. Mobile services persist on the local

service platforms on the site and terminate after servicing. Actually, service instances arrive to a local platform and are withdrawn later. However, some data may be returned to the original SPA to update its knowledge base regarding the performed diagnostics and efficiency of actions taken. This knowledge is used later on for the improvement of the service quality (Terziyan, 2005).

### **Rational Agent-Services**

OntoServ.Net services are wrapped by SPAs. SPAs, in addition to providing their services on SRAs' requests, reason about which activities to perform in a given case. OntoServ.Net has no division for service requestor and service provider layers, since both services and agents are conceptually the same. Resource Wrapping Agents (RWAs) represent industrial machines or their parts and provide Web Services to grant access to or operation on the respective devices. RWAs also act as SRAs. For example they acquire advanced diagnostic services from another SPA to monitor basic parameters of the machine.

Resource wrapping agent shell (OntoShell, a framework for resource and service adaptation to the Semantic Web-enabled environment (Terziyan, 2003)) can be applied to a wide range of resource types, including humans, knowledge bases, and industrial devices. OntoShell allows wrapping services implemented within the framework of W3C Web Service architecture or, in principle, any other software development technology that provides external application programming interfaces.

### **Service Composition Strategy in OntoServ.Net**

Service composition in OntoServ. Net is performed by platform-manager agents that act as mediators between service agents scattered over the network and local RWAs. A platform manager controls services' mobility and supports the P2P discovery mechanism of the OntoServ.Net environment, which is based on the matchmaking of a service request to dynamic service profiles (Kaykova et al., 2004; Khriyenko et al., 2004). A profile presents not only the service interface and the semantics, but also comprises the generalized description of SPA's successfulness in some states of the previously serviced SRAs. A dynamic profile is therefore required for credibility assessment. Since services are assumed to implement various learning techniques, their quality highly depends on the previous invocations, the samples for self-learning collected by SPAs, and initial training sets. If a service is complex and requires the invocation of other services, the performance is conducted by a local platform manager. The platform manager agent performs service discovery either locally or network-wide and provides

inter-platform communication facilities.

To round up, the OntoServ.Net framework provides the means for the development of agent-enabled P2P Web Service infrastructures in the networks of complex industrial machinery. The framework is applied to the development of the business models and the implementation of the secure service platforms that support new type of mobile services. It is based on the synergy of P2P and the Semantic Web, which ensures the successful deployment of industry-strong solutions based on agent technology.

### **Further Development of OntoServ.Net framework**

Since the previous publication in (Ermolayev et al., 2004) many elements of the conceptual framework described above have been implemented by the IOG in SmartResource Project<sup>18</sup> (Kaykova et al., 2005a; Kaykova et al., 2005b; Terziyan., 2005). The main focus of these development activities is to contribute to fast adoption of Semantic Web and related technologies in industry. It includes research and development aimed to design a Global Understanding Environment (GUN) as next generation of Web-based platforms by making heterogeneous industrial resources (files, documents, services, devices, business processes, systems, organizations, human experts, etc.) web-accessible, proactive and cooperative in a sense that they will be able to automatically plan their own behavior, monitor and correct their own state, communicate and negotiate among themselves depending on their role in a business process, utilize remote experts, Web-services, software agents and various Web applications. Three fundamentals of GUN platform are Interoperability, Automation, and Integration.

Interoperability in GUN requires utilization of Semantic Web standards, RDF-based metadata and ontologies, and semantic adapters of the resources. Automation in GUN requires proactivity of resources based on the use of agent technologies. Integration in GUN requires ontology-based business process modeling and integration, and multi-agent technologies for coordination of business processes over resources. GUN is a concept which is used to name a Web-based resource “welfare” environment which provides a global system for automated “care” of (industrial) Web-resources with the help of heterogeneous, proactive, intelligent, and interoperable Web services. The main players in GUN are the following resources: service consumers (or components of service consumers), service providers (or components of service providers), decision-makers (or components of decision makers). All these resources can be artificial (tangible or intangible) or natural (human or other). It is supposed that “service consumers” will be able: (a) to proactively monitor their own state over time and changing context; (b) to discover appropriate “decision makers” and order the remote diagnostics of their



own condition to these “decision makers”. The “decision makers” will then decide in the automatic manner which maintenance (“treatment”) services are applicable to that condition; (c) to discover appropriate “service providers” and order the required maintenance to them.

Industrial resources (for example devices, experts, software components, etc.) can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) the sensors with digital output, data structuring (for example XML) and semantic adapter components (XML to Semantic Web). Agents are assumed to be assigned to each resource and are able to monitor semantically reach data coming from the adapter about the states of the resource, decide if a deeper diagnostics of the state is needed, discover other agents in the environment which represent “decision makers”, and exchange information (agent-to-agent communication with semantically enriched content language) to get diagnoses and decide if a maintenance is needed. It is assumed that “decision making” Web-services will be implemented based on various machine learning algorithms and will be able to learn based on the samples of data taken from various “service consumers” and labeled by experts. The implementation of agent technologies in GUN framework allows for the mobility of service components between various platforms, decentralized service discovery, FIPA communication protocols utilization, and MAS-like integration/composition of services (Terziyan., 2006).

## **CONCLUDING REMARKS**

The paper presented the framework for agent-enabled dynamic Web Service composition. The core of the methodology is the new understanding of a Web Service as an agent capability having proper ontological description. It is demonstrated by the travel planning example how diverse Web Services may be composed and mediated by dynamic coalitions of software agents collaboratively performing tasks for service requestors. It is also claimed that such a mediation facility may substantially enhance today’s solutions available in Web Service provision. This vision is grounded on the results obtained in agent-enabled business process modeling and management. Some parts of the presented framework have been implemented in several research and development projects run by ISRG and IOG.

It is stated that though the concept of service mediation is not totally new there is still some work to be done before it becomes a real engineering technology. For example the framework for intelligent dynamic service composition and decomposition according to the changes in the environment affected by the service execution flow has not been explicitly worked out before. The framework introduces the agent middle layer to conduct the transformation of a service

request to the corresponding task and for further cooperative task decomposition and performance. Outlined are the formal means to arrange agents' negotiation on activity allocation, to represent the semantic structure of the request-task-activity-service hierarchy and to assess fellow-agents' capabilities and credibility factors. Further on, it is argued that the presented formal technique is applicable not only to the tasks like travel planning. Presented is the reference architecture of the rational multi-agent mediator for intelligent information and document retrieval of the RACING project. Presented aspects of service composition and mobile-agent service representation in a peer-to-peer network of service integration platforms extend RACING principles of service composition by the aspects of mobility. The experience of applying OntoServ.Net framework to the development of P2P service infrastructures provides also the evidence of the applicability of the agent-enabled Web service composition framework to real-world industrial applications

Though thorough standardization and harmonization work should be performed before the presented approach becomes an engine for Web service provision, the authors are certain that agent-enabled rational Web Service composition and mediation may provide a substantial contribution, bringing closer the day when the brave new world of machine-processable automated Web Services comes true.

## ENDNOTES

---

<sup>1</sup> Intelligent Systems Research Group, Department of IT, Zaporozhye National University, Ukraine, <http://ermolayev.com/ISRG/>

<sup>2</sup> Industrial Ontologies Group, Departments of IT and CS, University of Jyväskylä, Finland, <http://www.cs.jyu.fi/ai/OntoGroup/index.html>

<sup>3</sup> International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 14 June 2002, Sardinia (Italy) <http://tmitwww.tm.tue.nl/staff/gwagner/RuleML-BR-SW.html>. Diffuse: Guide to Web Services [http://www.diffuse.org/Web\\_Services.html](http://www.diffuse.org/Web_Services.html)

<sup>4</sup> DIP- Data, Information, and Process Integration with Semantic Web Services, <http://dip.semanticweb.org/>

<sup>5</sup> World Wide Web Consortium, <http://www.w3.org/>

<sup>6</sup> Service inputs are given semi-formally in order to avoid unnecessary details and save space.

<sup>7</sup> Foundation for Intelligent Physical Agents, <http://www.fipa.org/>,

<sup>8</sup> e-Appendixes A-1 – A-7 may be downloaded from <http://ermolayev.com/services/app.htm>

<sup>9</sup> “One of the most common low fare restrictions is the requirement for your stay to incorporate at least one Sunday. For example for a round-trip New York to Miami a passenger flying Tuesday to Thursday might pay £328, but a passenger whose stay includes a Sunday would pay much less - £188.” – <http://www.flightcatchers.com/helpmenu/Howtofindcheapestfare.htm>

---

<sup>10</sup> Lufthansa Infoflyway Booking Service <http://lufthansa.com/> and Cyber Flyer Booking Service <http://cyberflyer.galileo.com/> were used in the described exercise to obtain the offers from F-s.

<sup>11</sup> CNN Currency Converter: <http://qs.money.cnn.com/tq/currconv/>

<sup>12</sup> <http://www.bahn.de/>

<sup>13</sup> Applying to a capability registry may still appear to be necessary in case B, C and D fail to provide constructive proposals.

<sup>14</sup> Performance Simulation Initiative (PSI) is the research and development project of Cadence Design Systems, GmbH.

<sup>15</sup> RACING: Rational Agent Coalitions for Intelligent Mediation of Information Retrieval on the Net. <http://www.zsu.zp.ua/racing/> Project funded by the Ukrainian Ministry of Education and Science under the grant No 0102Y005339.

<sup>16</sup> As QPAs in RACING have limited life time, RWAs' credibility and capability assessment is performed by MA for registered resource wrappers. QPAs supply MA with necessary data obtained from cooperation with RWAs.

<sup>17</sup> UnIT-Net - IT in University Management Network, <http://www.unit-net.org.ua/>

<sup>18</sup> SmartResource project, [http://www.cs.jyu.fi/ai/OntoGroup/SmartResource\\_details.htm](http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm)

## REFERENCES

Ankolekar, A., Urstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., & Sycara, K. (2002, June 9-12). DAML-S: Web service description for the Semantic Web. *Proceedings of Int. Semantic Web Conference*, Sardinia, Italy.

OWL-S (2003). The DAML services coalition. DAML-S: Semantic markup for Web services. <http://www.daml.org/services/owl-s/1.0/>

Ermolayev, V., Jentzsch, E., Karsayev, O., Keberle, N., Matzke, W.-E., and Samoylov, V. (2005, October 24-28). Modeling Dynamic Engineering Design Processes in PSI. In: J. Akoka et al. (Eds.): ER Workshops 2005, Proc. 7<sup>th</sup> Int Bi-Conf Workshop on Agent-Oriented Information Systems ([AOIS-2005](#)), Klagenfurt, Austria, Springer LNCS 3770, pp. 119–130

Ermolayev, V., Jentzsch, E., Keberle, N., Samoylov, V., and Sohnius, R. (2006b, April). The Family of PSI Ontologies v.1.5. Reference Specification. Technical Report. Cadence Design Systems, GmbH, 47 p.

Ermolayev, V., Keberle, N. (2006, May 30-31). A Generic Ontology of Rational Negotiation. In: Karagiannis, D., Mayr, H.C. (Eds.): Information Systems Technology and its Applications. 5-th Int Conf ISTA'2006, Klagenfurt, Austria, pp. 51-66.

---

Ermolayev, V., Keberle, N., Kononenko, O., Plaksin, S., and Terziyan, V. (2004). Towards a framework for agent-enabled semantic web service composition. *Int. J. of Web Services Research*, 1(3), 63-87.

Ermolayev, V., Keberle, N., Matzke, W.-E., and Vladimirov, V. (2005b, November 6-10). A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval. In: Y. Gil et al. (Eds.): ISWC 2005 Proc. 4th Int. Semantic Web Conference ([ISWC'05](#)), Galway, Ireland, LNCS 3729, pp. 201 – 215.

Ermolayev, V., Keberle, N., Shapar, V., Vladimirov, V. (2004b, July 15-17). Ontology-Driven Sub-Query Extraction for Distributed Autonomous Information Resources in UnIT-Net IEDI. In: A. Doroshenko, T. Halpin, S. W. Liddle, H. C. Mayr (eds.) *Information Systems Technology and its Applications. Proc. 3-d Int. Conf ISTA'2004*, Salt Lake City, Utah, USA, pp. 137-150, GI LNI Vol P-48.

Ermolayev, V., Keberle, N., & Tolok, V. (2001, November 27-30). 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.

Ermolayev, V., Keberle, N., Plaksin, S., & Vladimirov, V. (2003). Capturing semantics from search phrases: Incremental user personification and ontology-driven query transformation. In M. Godlevsky, S. Liddle & H. Mayr (Eds.), *Information systems technology and its applications. Proc. of the 2nd Int. Conf. ISTA2003 Lecture Notes in Informatics (LNI)*, P-30, GI-Edition, Bonn, 9–20.

Ermolayev, V.A., & Plaksin, S.L. (2002). Cooperation layers in agent-enabled business process management. *Problems of Programming*, 1-2, 354-368.

Ermolayev, V.A., & Tolok, V.A. (2002). Modelling distant learning activities by agent task coalitions. In Q. Jin, J. Li, J. Cheng, C. Yu & S. Noguchi (Eds.), *Enabling society with information technology*. Tokyo: Springer-Verlag.

Fensel, D., & Bussler, C., (2002). *The Web Service Modeling Framework WSMF*. White paper. Retrieved June 3, 2002, from <http://www.cs.vu.nl/~dieter/wese/wsmf.paper.pdf>

FIPA. (2003). *FIPA Communicative Act Library Specification. Doc. No: XC00037H*. Retrieved June 14, 2003, from <http://www.fipa.org/specs/fipa00037/XC00037H.pdf>

Jennings, N.R. (2000). On agent-based software In K.P. Sycara, M. Klusch, S. Widoff & J. Lu.

- 
- (1999). Dynamic service matchmaking among agents in open information environments. *SIGMOD Record*, 28(1), 47-53.
- Karp, A. (2003). *E-speak explained*. HPL-2000-101 20000807 External Technical Report. Retrieved June 3, 2003, from <http://www.hpl.hp.com/techreports/2000/HPL-2000-101.pdf>
- Kaykova, O., Khriyenko, O., Kovtun, D., Naumenko, A., Terziyan, V., and Zharko, A. (2005a). General Adaption Framework: Enabling Interoperability for Industrial Web Resources, *International Journal on Semantic Web and Information*
- Kaykova et al. Kaykova, O., Khriyenko, O., Terziyan, V., and Zharko, A. (2005b, August 25-27). RGBDF: Resource Goal and Behaviour Description Framework, In: M. Bramer and V. Terziyan (Eds.): *Industrial Applications of Semantic Web*, Proceedings of the 1-st International IFIP/WG12.5 Working Conference IASW-2005, Jyväskylä, Finland, Springer, IFIP, 83-99.
- Kaykova O., Kononenko O., Terziyan V., and Zharko A. (2004, February 17-19) Community Formation Scenarios in Peer-to-Peer Web Service Environments. In: *Proc IASTED Int Conf on Databases and Applications (DBA 2004)*, Innsbruck, Austria, pp. 62-67.
- Keberle N., Ermolayev V., Shapar V., and Vladimirov V. (2004). Semantically Reinforced Web Services for Wrapping Autonomous Information Resources. *Bulletin of V. Karazin Kharkiv National University*, Series «Mathematical Modelling. Information Technology. Automated Control Systems», 629(3), 56 – 69.
- Khriyenko, O., Kononenko, O., & Terziyan, V. (2004). OntoEnvironment: An integration infrastructure for distributed heterogeneous resources. *IASTED International Conference on Parallel and Distributed Computing and Networks*, PDCN-2004 (to appear).
- Layman, A. (2001, April 11-12). Web services framework. *Proceedings of W3C Workshop on Web Services 2001*, San Jose, CA, USA. Retrieved June 3, 2003, from <http://www.w3.org/2001/03/wsws-program/>
- McIlraith, S.A., Son, T.C., & Zeng, H. (2002). Semantic Web services. *IEEE Intelligent Systems*, *Special Issue on The Semantic Web*, 16, 46-53.
- Nwana, H.S. (1996). Software agents: An overview. *Knowledge Engineering Review*, 3(11), 205-244.
- Paolucci, M., Kawamura, T., Payne, T., & Sycara, K. (2002, June 9-12). Semantic matching of Web services capabilities. *Proc. of Int. Semantic Web Conference (ISWC'2002)*, Sardinia, Italy.

---

Papadopoulos, G.A. (2001). Models and technologies for the coordination of Internet agents: A survey. In A. Omicini, F. Zambonelli, M. Klusch & R. Tolksdorf (Eds.), *Coordination for Internet agents - Models, technologies, and applications*. New York: Springer-Verlag.

Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D. (2005) Web Service Modeling Ontology, *Applied Ontology*, 1(1): 77 – 106

Sandholm, T. (1996). *Negotiation among self-interested computationally limited agents*. PhD thesis, University of Massachusetts, Amherst.

Siebes, R. (2002, October 30). *Peer-to-peer solutions in the Semantic Web context: An overview*. EU-IST Project IST-2001-34103 SWAP. Vrije Universiteit Amsterdam.

Simple Object Access Protocol (SOAP) 1.1. W3C Note (2000, May 8). Retrieved June 3, 2003, from <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

Sivashanmugan, K., Verma, K., Mulye, R., & Zhong, Z. (2002). *SpeedR: Semantic peer-to-peer environment for diverse Web service registries*. Final Project, CSCI 8350. Enterprise Integration.

Sycara, K., Widoff, S., Klusch, M., & Lu, J. (2002). LARKS: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5, 173–203.

Terziyan, V. (2003). Semantic Web services for smart devices in a “global understanding environment”. In R. Meersman & Z. Tari (eds.), *On the move to meaningful Internet systems. Lecture Notes in Computer Science*, 2889, 279-291.

Terziyan, V. (2005). Semantic Web Services for Smart Devices Based on Mobile Agents, *International Journal of Intelligent Information Technologies*, Idea Group, 1(2), 43-55.

Terziyan V. (2006). Challenges of the “Global Understanding Environment” based on Agent Mobility, In: V. Sugumaran (ed.), *Advanced Topics in Intelligent Information Technologies*, Vol. 1, Idea Group, (Book chapter in press, available in: [http://www.cs.jyu.fi/ai/ATIIT\\_Chapter-2006.doc](http://www.cs.jyu.fi/ai/ATIIT_Chapter-2006.doc)).

Terziyan, V., & Kononenko, O. (2003). Semantic Web enabled Web services: State-of-art and industrial challenges. In M. Jeckle & L.-J. Zhang (Eds.), *Web Services - ICWS-Europe 2003, Lecture Notes in Computer Science*, 2853, 183-197. Springer-Verlag.

---

UDDI Technical White Paper. <http://uddi.org/>

Thaden, U., Siberski, W., & Nejdl, W. (2003). *A Semantic Web based peer-to-peer service registry network*. Technical report.

Web Services Description Language (WSDL) 1.1. W3C Note 15. (2001, March).  
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315/>

*Vadim Ermolayev is an associate professor at the Department of Information Technologies of Zaporozhye National University and the lead for the Intelligent Systems Research Group. His research interests include intelligent information integration, agent-enabled e-business and e-learning, ontologies, Semantic Web Services, evolution and adaptability in intelligent software systems, and dynamic business process management and performance. He received his PhD in Mathematical Modeling and Computer Science from Zaporozhye State University. More information may be retrieved from <http://google.com/search?q=Ermolayev>*

*Natalya Keberle is an assistant professor at the Department of Information Technologies of Zaporozhye National University and the member of the Intelligent Systems Research Group. Her research interests include the ontological models and the methods for presenting evolving semantics in intelligent information systems. More information may be retrieved from <http://google.com/search?q=Keberle>*

*Oleksandr Kononenko is a research engineer at Helsinki Nokia Research Center and an associate member of Industrial Ontologies Group at the Department of Mathematical Information Technology of the University of Jyväskylä. His research is concentrated around the topics of software architectures for existing and evolving Web technologies. Contact him at [Oleksandr.Kononenko@nokia.com](mailto:Oleksandr.Kononenko@nokia.com)*

*Vagan Terziyan is an associate professor and a senior researcher at the Department of Mathematical Information Technology of the University of Jyväskylä. He is also the chair of the Department of Artificial Intelligence and Information Systems at the Kharkiv National University of Radioelectronics. His research interests and competences comprise distributed AI, multiagent systems, Semantic Web and Web services, peer-to-peer computing, knowledge management and machine learning. He received a Dr.Tech in Engineering degree from Kharkiv National University of Radioelectronics. More information may be retrieved from <http://google.com/search?q=Terziyan>*