Using Negotiation for Dynamic Composition of Services in Multi-Organizational Environmental Management

Costin Bădică[1], Sorin Ilie[1], Michiel Kamermans[2],
Gregor Pavlin[2], and Mihnea Scafeş[1]

[1] University of Craiova, Software Engineering Department, Romania,
{costin.badica,sorin.ilie,mihnea.scafes}@software.ucv.ro
[2] Thales Nederland B.V., D-CIS Lab Delft, The Netherlands
{gregor.pavlin,michiel.kamermans}@d-cis.nl

Abstract. This paper discusses an efficient solution to contemporary situation assessment problems found in environmental management applications. The targeted problems are inherently complex and require processing of large quantities of heterogeneous information by using rich domain knowledge dispersed through multiple organizations. We assume a collaborative solution based on the Dynamic Process Integration Framework, which supports systematic encapsulation of heterogeneous processing services, including human experts. The encapsulation allows dynamic composition of heterogeneous processing services using advanced self-configuration mechanisms for optimal selection of service providers. Self-configuration is based on a framework for development of cooperative multi-issue one-to-many service negotiations. The framework allows the definition of: negotiation protocols, negotiation subjects composed of multiple negotiation issues, properties of negotiation issues, deal spaces, and utility functions of participant stakeholders. We show how this framework can be used for dynamic composition of workflows spanning multiple organizations in a disaster management information system.

Key words: workflow, software agent, service negotiation

1 Introduction

The increased complexity of socio-economic systems in combination with natural hazards might create conditions that, if not properly assessed, can evolve into dangerous incidents, for example chemical disasters. These are extremely complex and heterogenous phenomena that can threaten the population as well as the natural environment. For example, places such as harbors as well as highly industrialized areas (e.g. the port of Rotterdam, the Netherlands) are constantly exposed to such risks because many companies are using, transporting and/or processing dangerous chemical substances in their neighborhood. Agencies specialized in environmental protection and emergency management are continuously monitoring the situation with the goal of taking corrective actions when things tend to go wrong. For example, DCMR[3] in the Netherlands and

---

[3] DCMR Milieudienst Rijnmond, http://www.dcmr.nl/

DEMA[4] in Denmark are two environment management organizations that protect the people and the environment against chemical hazards. They are specialized in solving complex environmental management problems by applying systematic procedures that are aimed, for example, at monitoring the situation and, when needed, act in order to discover the source of an incident, handle the chemical substance(s), secure the area and eventually decide on whether to evacuate (part of) the population or take other protective or corrective measures.

Actually, many contemporary applications require reasoning about complex processes and phenomena in real world domains including crisis management, environmental management, and project management. For example, in crisis management advanced information processing is required for various tasks, including (i) identification of critical situations, (ii) impact assessment which takes into account possible evolution of physical processes, (iii) planning and evaluation of countermeasures and (iv) decision making. These tasks are characterized as very complex situations that require a fast and efficient response through flexible and dynamic collaboration between various actors including emergency units, local and/or regional authorities, human experts, a.o., as well as systematic processing of large quantities of very heterogenous information, based on rich expertise about different aspects of the physical world. Usually underlying business processes involve interactions that cross organizational and geographical boundaries and that are highly dynamic and unpredictable in nature, thus demanding flexible composition of problem solving and reasoning patterns that cannot be captured using traditional workflow technologies.

Unfortunately, the processing requirements usually exceed the cognitive capabilities of a single human expert; an expert typically does not have knowledge of all the relevant domain methods and can only process limited amounts of available information. On the other hand, full automation of decision making processes in such settings is not feasible, since the creation of the required domain models as well as the inference are intractable problems. Therefore, complex assessment is often carried out in systems which can be characterized as *professional bureaucracies* [1], a class of organizations in which the skills are standardized, the control is decentralized to a great extent and the experts do not have to share domain knowledge.

We introduce a service oriented architecture called Dynamic Process Integration Framework (DPIF) that supports inter-organizational collaborative information processing in dynamic and distributed workflows. DPIF allows seamless integration of heterogeneous domain knowledge and processing capabilities into coherent collaborative processes. Processes are encapsulated by software agents, each using identical interaction protocols and collaboration interfaces for service provision and consumption. The DPIF combines Multi Agent Systems (MAS) [9] and a service oriented paradigm in new ways which facilitate implementation of hybrid collaborative reasoning systems with emergent problem solving capabilities. In contrast to traditional MAS approaches [9], the DPIF facilitates integration of human cognitive capabilities right into problem solving processes in workflows; humans are not mere users of an automated system, but contribute as processing resources. From the problem solving perspective, the humans experts can be viewed as a specific type of processing modules, integrated into the

---

[4] Danish Emergency Management Agency, `http://www.brs.dk/`

overall processing system via assistant agents. Each expert is associated with a DPIF assistant agent, which collects all the relevant information and disseminates the conclusions/estimates to the DPIF assistants of other interested service providers. In other words, DPIF assistant agents dynamically connect service providers into information processing workflows and facilitate automated information dissemination. In addition, some of the reasoning processes might be automated, which would speed up the assessment and increase quality. However, full automation of complex assessment processes is likely to be unacceptable or even impossible.

The DPIF is a service-oriented approach which supports dynamic and decentralized creation of workflows that facilitate collaborative problem solving. The DPIF supports efficient composition of heterogenous processing services provided by different human experts and computational processes. Each process provides a well-defined reasoning service in the form of an estimate, prediction, cost, etc. The inputs of a process are provided by other processes or by direct observations, like sensor measurements or human reports. Processes are wrapped as DPIF agents that support standardized interaction protocols. Each agent registers in the DPIF system (i) the services that the expert/automated process can provide and (ii) the types of services supplying the inputs that are required to provide the services of the expert.

By using the registered services, agents distributed throughout different networked devices can autonomously form workflows enabling collaborative processing of human experts and automated processing tools. The DPIF supports service composition which explicitly takes into account the characteristics of *professional bureaucracies*. Service composition in such settings can be achieved through service matching and discovery based on local domain knowledge supplemented with a finer level of control based on one-to-many service negotiation. Each expert or automated process "knows" which types of information (i.e. other service types) are required for providing his or her specific services. Service negotiation allows us to filter potential links found through service discovery, using additional dynamically-adjustable service parameters, rather than relying on a perfect matching at the service discovery level.

The paper is structured as follows. We start in Section 2 with the introduction of a utilization scenario in the domain of environment management. The scenario is formalized using the DPIF model of agents and services. We follow in Section 3 with a discussion of the dynamic formation of multi-organizational workflows using the DPIF model of service composition. Then in Section 4 we show using an example extracted from the utilization scenario how service negotiation can help tuning the DPIF service compositions. In the last section of the paper we briefly summarize our conclusions.

2   Environment Management Scenario

We illustrate our approach by using an example derived from a real world use case investigated in the FP7 DIADEM project[5]. For the sake of clarity but without the loss of generality we assume a significantly simplified scenario. In a chemical incident at a refinery a leaking chemical starts burning which results in harmful fumes. The impact

---

[5] `http://www.ist-diadem.eu`

of the resulting fumes is assessed through a service composition involving collaboration of human experts, as depicted in Figure 1 and explained below:

- *The* Control Room *operator is triggered by the* Gas Detection *system about the possible presence of a chemical incident caused by the leak of a dangerous gas.*
- *The* Control Room *uses the information provided by the* Gas Detection *system and applies local knowledge about the industrial environment to determine the source of the incident. Consequently, it requests a report of the situation from the factory via the* Factory Representative. *The* Factory Representative *replies with report that confirms the incident and provides information about the type of escaping gas.*
- *The* Control Room *directs a field inspector denoted by* Chemical Adviser 1 *to the location of the incident.* Chemical Adviser 1 *has appropriate expertise to estimate the quantity of the escaping gas and to propose mitigation measures at the refinery.*
- *Health complaints are reported. We assume that they also arrive to the* Control Room *via the* Gas Detection *system. Consequently, the* Control Room *dispatches a chemical expert that holds expertise in estimating the gas concentration in the affected area. This expert is denoted as* Chemical Adviser 2.
- *The* Chemical Adviser 2 *requires information about the meteorological conditions, the source of the pollution, and the quantity and type of escaping fumes in order to estimate the zones in which the concentration of toxic gases has exceeded critical levels and to identify areas which are likely to be critical after a certain period of time. We assume that* Chemical Adviser 2 *gets the weather information from the* Control Room *and the information about the source, quantity, and type of the escaping gas from* Chemical Adviser 1. *The* Chemical Adviser 2 *makes use of domain knowledge about the physical properties of gases and their propagation mechanisms.*
- *In addition, the* Chemical Adviser 2 *guides fire fighter* Measurement Teams *which can measure gas concentrations at specific locations in order to provide feedback for a more accurate estimation of the critical area. This interaction between* Chemical Adviser 2 *and* Measurement Teams *involves negotiation to determine the optimal providers of appropriate measurements.*
- *A map showing the critical area is supplied by the* Chemical Adviser 2 *to a* Health Expert. *He uses additional information information on populated areas obtained from the municipality to estimate the impact of the toxic fumes on the human population in case of exposure.*

Analyzing the utilization scenario, we were able to identify an initial list of stakeholders that are involved in the collaborative incident resolving process. Each stakeholder is mapped onto a DPIF software agent that keeps track of services provided by the stakeholder, as well as services that are required by each provided service – see Table 1. For example, *Chemical Adviser 2* provides output service *Map of High Concentration Zones* producing a map of the critical area. In order to provide service *Map of High Concentration Zones* she consumes three input services called *Information about Source*, *Weather Report* and *Concentration Measurements* providing her with i) information about the source, ii) weather and iii) concentration measurements in the interest area, respectively.

The workflow shown in Figure 1 is not hard-coded in a centralized place. Rather, this workflow is dynamically composed using the DPIF service composition model. Composition of services is achieved through service matching and discovery based on local domain knowledge. Each expert or process has complete knowledge of which
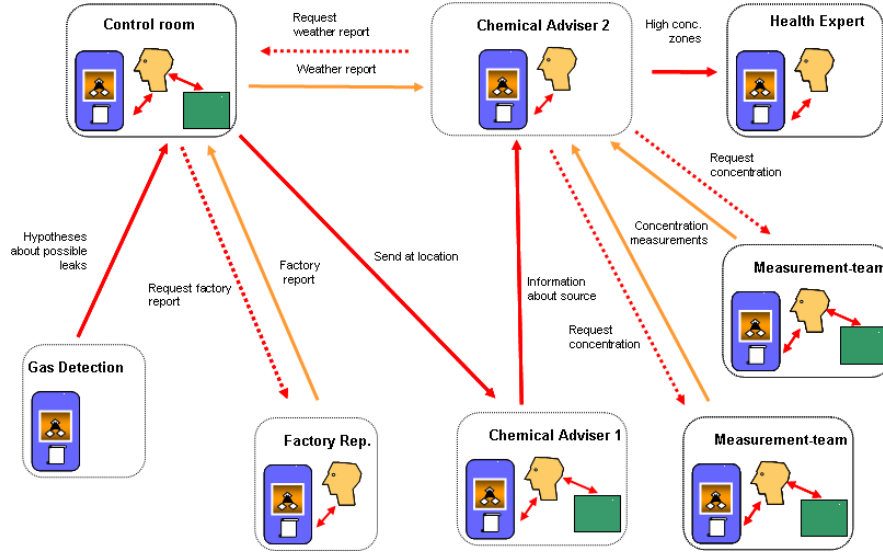
Fig. 1. Workflow for sample scenario.

types of services he/she can provide to the community – the set of *output services*, as well as the types of services that are needed from the community to provide the output services,i.e. a set of *input services*. Consequently, the DPIF does not require centralized service ontologies describing relations between services and centralized service composition methods. However, a global lightweight ontology of services is necessary to align the syntax and semantics of services at the configuration of an assistant agent, prior to joining the overall system. Such configuration based on aligned services descriptions is key to runtime service composition resulting in dynamic workflows [6].

3 Dynamic Workflows

A basic workflow element in the DPIF is abstracted as a local process representing either the reasoning process of a human expert or an automated system implemented by a computational procedure. Each local process corresponds to a function that takes values of n input variables $x_1, x_2, \ldots, x_n$ and produces the value of an output variable $y = f_y(x_1, x_2, \ldots, x_n)$. For example, the local process of the *Chemical Adviser 2* described in Table 1 implements the following function:

$$x_{Map\ of\ High\ Concentration\ Zones} =$$
$$f_{Map}(x_{Information\ about\ Source}, x_{Weather\ Report}, x_{Concentration\ Measurements})$$

In the DPIF, a local process is wrapped as a DPIF software agent. The values of the input variables are provided by the input services of the DPIF agent, while the value of the process output is produced by an output service of the DPIF agent. More details about the architecture and design of the DPIF framework can be found in [5]. In

Table 1. This table shows for each DPIF agent from the utilization scenario (i) the provided services (see column labeled *output service*) and (ii) the services required for each provided service (see column labeled *input service).*

Agent	Output Service	Input Service
Gas Detection	Hypothesis	n/a
Control Room	Send at Location	Hypothesis Factory Report
	Weather Report	n/a
Factory Representative	Factory Report	n/a
Chemical Adviser 1	Information about Source	Send at Location
Chemical Adviser 2	Map of High Concentration Zones	Information about Source Weather Report Concentration Measurements
Health Expert	Health Report	Map of High Concentration Zones
Measurement Team 1	Concentration Measurements	n/a
Measurement Team 2	Concentration Measurements	n/a

what follows we present details of how the DPIF is applied to the utilization scenario introduced in Section 2.

It is interesting to note that the expertise of a human expert that is represented by a DPIF agent of type *Chemical Adviser 2* is local in the sense that the processing associated to its function $f_{Map\ of\ High\ Concentration\ Zones}$ encapsulates it, while only the external interfaces of this function are visible in the DPIF framework to allow dynamic formation of workflows through service composition based on service matching and discovery. For example, the output *Map of High Concentration Zones* of the agent *Chemical Adviser 2* can be matched using the service name with the input *Map of High Concentration Zones* of the agent *Health Adviser*.

Similarly, the local process of the *Health Adviser* described in Table 1 realizes the following function:

$$x_{Health\ Report} = f_{Health\ Report}(x_{Map\ of\ High\ Concentration\ Zones})$$

A function that is reduced to direct observation of its output variable can be noticed for the DPIF agents of type *Measurement Team* as follows:

$$x_{Concentration\ Measurements} = f_{Concentration\ Measurements}()$$

Inputs to a certain function ($f_{Health\ Report}$ for example) can be supplied by other DPIF agents (*Chemical Adviser 2* for example), thus dynamically forming a collaborative workflow. From a global perspective this workflow formation can be seen as a function composition that is realized in the DPIF using service composition. The dynamic composition process continues until a function is obtained in which all variables have been assigned a value. The resulting function obtained through service composition represents a mapping between directly observable variables (i.e. $x_{Concentration\ Measurements}$ for example, which is provided by *Measurement Team* agents) and hidden variables of interest ($x_{Health\ Report}$ for example). An example of composite function that yields the

value of interest $x_{Health\ Report}$ is:

$$x_{Health\ Report} = f_{Health\ Report}($$
$$f_{Map\ of\ High\ Concentration\ Zones}($$
$$x_{Information\ about\ Source},$$
$$x_{Weather\ Report},$$
$$x_{Concentration\ Measurements}$$
$$)$$
$$)$$

4   Self-configuration Using Service Negotiation

In the DPIF, communication links between local processes in agents are facilitated firstly using service discovery: whenever an agent supplying some service (we will call this service the parent service, and the agent implementing it the manager) in a workflow requires data relating to some other service (we will call this required service the child service, and the agent implementing it the contractor), a communication link needs to be established between the manager agent and the contractor agent. However, there are two important aspects that affect whether and why links are established: i) we might have several agents in the system that provide the same service, i.e. that are able to realize the same task, and ii) we cannot always assume that an agent providing a service will automatically agree to supply the service asked for by a requesting agent. For example, the provider might be overloaded, or it might even consider that establishing a link is inappropriate, given the current context.

In addition, service discovery alone can only offer links between agents based on a broad level of service matching, while for solving a particular problem, a finer level of control is required to match services on additional parameters. Establishing links is based on one-to-many service negotiation. Rather than performing perfect matching with service discovery, negotiation allows us to filter potential links found through service discovery based on additional service parameters.

Negotiation is a process that describes the interaction between one or more participants that must agree on a subject by exchanging deals or proposals about this subject [3]. Negotiation about a service that one or more participants agree to provide to other participants is called *service negotiation*. We have developed a conceptual framework for service negotiation that is used in the DPIF. The framework is generic and it addresses negotiation protocols, negotiation subjects and decision components. Due to space restrictions we only briefly review the framework here. Details of the conceptual framework can be found in [2], while a brief description of the design and implementation is given in [7].

Our protocol supports one-to-many negotiations and it defines two roles: manager and contractor [8, 4]. The manager is the agent that requests a service and thus initiates the negotiation. The contractor is the agent that is able to provide the service requested by the manager. A set of generic negotiation steps are defined: (i) negotiation subject identification and negotiation announcement (initiation of negotiation), (ii) bidding, i.e.

making proposals and counter-proposals, (iii) deciding whether an agreement or a conflict was reached, and (iv) termination.

Negotiation subject comprises the service description and a subset of the service parameters that are important decision factors during negotiation (i.e. their current values are taken into consideration when selecting the appropriate service providers). During negotiation, these parameters are considered *negotiation issues*. Thus, when the negotiation designer configures the service, he also defines the negotiable parameters of the service (i.e. negotiation issues).

Negotiation issues are described by properties including their name, data type, and monotonicity. The name of the issue uniquely identifies the issue in a negotiation subject. The data type of the issue describes the type of the value the issue is allowed to take. It can be as simple as a number or string or more complex, as required for example to describe a geographical location or a date/time value. The monotonicity specifies whether the manager prefers higher values to lower values of this issue. Possible values are: (i) INCREASING if the agent prefers high utility values of the issue and (ii) DECREASING if the agent prefers low utility values of this issue.

An important characteristic of service negotiations in DIADEM is that they are *cooperative*. Cooperativity stems from the fact that the overall goal of negotiation participants is the optimization of the response for situation assessment in a chemical incident. Negotiations for a certain service provision are carried out only with agents that are able to provide the required service (i.e. that possess the domain knowledge or physical capabilities that are needed to provide the service). Provider agents will usually accept to offer their services if they are currently able to do so (for example if they posses all the necessary resources). During a negotiation: (i) the manager is the leading decision factor that looks to optimize the assignment of the negotiation task to the best available contractor(s); (ii) the contractor(s) make their best proposals for serving the manager, taking into account their current duties and availability, thus preserving their autonomy according to the principles of professional bureaucracy.

Service parameters can be classified into 4 classes: (i) DYNAMIC that specifies that the issue value is not fixed by the manager, i.e. the contractor can propose different values for the issue; (ii) FIXED that specifies that the issue value is fixed by the manager, i.e. if the contractor proposes a different value for the issue than the corresponding local utility of the issue is zero; (iii) CONDITION that specifies that the issue value is fixed by the manager, but if the contractor proposes a different value for the issue than the total utility of the proposal is zero; normally a contractor that cannot meet the issue value requested by the manager must decide to not bid because the utility of her bid will be zero; (iv) TRIVIAL that means that the issue is not taken into account in the computation of the bid utility, although it can be set in the request of the manager and consequently, it can be taken into account in the negotiation by the contractor and help her to make a more informed decision if and what to bid.

Negotiation participants in either manager or contractor role use utility functions to measure their preferences over deals. In our framework the manager uses a weighted additive utility function to evaluate proposals and to select the service provider. Each negotiation issue i has a weight $w_i \in [0, 1]$ and a partial utility function f_i. Note that weights are normalized i.e. $\sum_i w_i = 1$. Intuitively, the weight of an issue represents the

relative importance for the manager of that issue in the set of all issues associated to a negotiation subject. The partial utility of an issue i maps the issue domain D_i to a value in the interval $[0, 1]$, i.e. $f_i : D_i \rightarrow [0, 1]$. The definition of function f_i depends on the domain of the issue. For example, a possibility to define the partial utility function of a real valued issue with $D_i = [x_{min}, x_{max}]$ is as follows:

$$f_i(x) = \frac{|x - x^*|}{|x_{max} - x_{min}|}$$

where x^* is the reference value assigned by the manager to the issue i and $|x - y|$ is the distance between x and y (note that the distance actually depends on the data type of the negotiation issue). Note that a negotiation issue for which the partial utility is defined as a distance from the reference value (that represents the optimal value from the manager point of view) has always a DECREASING monotonicity.

Let I be the set of negotiation issues partitioned into sets $I^{\uparrow}$ and $I^{\downarrow}$ of issues with INCREASING and DECREASING monotonicity. The utility function of a proposal $x = (x_i)_{i \in I}$ has the following form:

$$u_m(x) = \sum_{i \in I^{\uparrow}} w_i * f_i(x_i) + \sum_{i \in I^{\downarrow}} w_i * (1 - f_i(x_i))$$

The utility function of the contractor agent can be defined as $u_c(x) = 1 - effort(x)$. Here $effort(x)$ represents the total effort that must be deployed by contractor agent c to be able to provide the terms and conditions required by call for proposals proposal x. The contractor will obviously choose to propose a deal that maximizes her utility, i.e. that minimizes the required effort for achieving the task required by the manager. In practice, function $effort(x)$ can take into account several factors, including for example the cost of providing service x, the existing commitments of agent c that were previously contracted and are not yet finalized, and/or the amount of resources that are required to provide the service. Obviously, a higher number of commitments not finalized yet of agent c or a higher cost of providing x will result in a higher value of $effort(x)$ and consequently to a lower value of the utility $u_c(x)$.

Let us suppose that *Chemical Adviser 2* agent plays the role of manager looking for a provider for the service *Measure Gas Concentration*. The optimal selection of the service provider takes into account: the location where the measurement must be performed, the quality of the measurement, and the duration for performing the measurement. Additionally we assume that the measurement quality is given as a percentage and that the maximum time frame for performing the measurement is 100 minutes. The description of the negotiation issue, together with the manager proposal are given in Table 2.

Weights of negotiation issues must be normalized as follows:

$$w_{Location} = \frac{1}{6}$$
$$w_{Quality} = \frac{2}{6}$$
$$w_{Deadline} = \frac{3}{6}$$

Table 2. Negotiable issues and manager request.

Issue	Location	Quality	Deadline
Reference value	*loc*	100	11:47 AM
Weight	1	2	3
Data type	REGION	NUMBER	DATE
Boundary	n/a	100	100
Negotiable	FIXED	DYNAMIC	DYNAMIC

Table 3. Contractors' bids.

Issue	Location	Quality	Deadline
Bid Value (1)	*loc*	70	11:58 AM
Bid Value (2)	*loc*	100	00:12 PM

Let us assume that there are two *Measurement Teams* in the system and each of them decides to bid with an offer for providing the service *Measure Gas Concentration*. Their bids are shown in Table 3.

The utility of the bid of the first *Measurement Team* is computed as follows:

$$u_{Location_1} = \frac{1}{6} \times (1 - \frac{0}{1}) = 0.166$$
$$u_{Quality_1} = \frac{2}{6} \times (1 - \frac{30}{100}) = 0.233$$
$$u_{Deadline_1} = \frac{3}{6} \times (1 - \frac{11}{100}) = 0.445$$

$$u_{MT_1} = u_{Location_1} + u_{Quality_1} + u_{Deadline_1} = 0.844$$

The utility of the bid of the second *Measurement Team* is computed as follows:

$$u_{Location_2} = \frac{1}{6} \times (1 - \frac{0}{1}) = 0.166$$
$$u_{Quality_2} = \frac{2}{6} \times (1 - \frac{0}{100}) = 0.333$$
$$u_{Deadline_2} = \frac{3}{6} \times (1 - \frac{25}{100}) = 0.375$$

$$u_{MT_2} = u_{Location_2} + u_{Quality_2} + u_{Deadline_2} = 0.874$$

Chemical Adviser 2 agent uses these equations to compute the utilities of each bid received from *Measurement Team* agents. Then *Chemical Adviser 2* applies a strategy that allows it to either immediately select the winning bid or to decide if to continue the negotiation using a new iteration. Let us assume that *Chemical Adviser 2* applies a strategy that considers acceptable only those bids valuating above a given threshold value. If none is above the threshold then *Chemical Adviser 2* can perform a second iteration either by relaxing the conditions of the call for proposals (for example by decreasing

the required quality of the measurements or by extending the deadline for performing the measurements) or by decrementing the threshold, thus giving a new chance to the *Measurement Team* agents to update their bids. If at least one bid is considered acceptable then *Chemical Adviser 2* can decide to accept one or more *Measurement Team* agents to contract the *Measure Gas Concentration* service. Assuming for example a threshold of 0.85, according to this algorithm *Chemical Adviser 2* will select the second *Measurement Team* after the first iteration.

5   Conclusions

This paper presents a solution which supports automated organization of experts which collaboratively solve complex problems found in contemporary environmental and crisis management applications. In particular, the approach combines the Dynamic Process Integration Framework (DPIF) and automated negotiation. DPIF is a wrapper technology which supports encapsulation and combination of heterogeneous processing capabilities in collaborative problem solving processes found in complex real world domains, such as environmental management. DPIF provides a technically sound infrastructure in which advanced negotiation processes can be systematically embedded. Experts and automated processes are wrapped such that their services become composable and negotiable.

Negotiation is achieved through an additional software layer that enhances DPIF agent communication with flexible service negotiation protocols. This negotiation layer allows definition and configuration of service negotiation protocols in agent-based collaborative processes. In particular our framework supports flexible configuration of multi-issue negotiation subjects, properties of negotiation issues, deal spaces, and utility functions of participant agents. An example covering use of utility functions in a sample negotiation scenario was discussed in detail, emphasizing how service negotiation can improve the selection of optimal service providers in environment management.

Acknowledgement

References

1. Estefania Argente, Vicente Julian, and Vicente Botti, Multi-Agent System Development Based on Organizations, *Electronic Notes in Theoretical Computer Science 150* (2006), Elsevier, 55–71.
2. Costin Bădică, Mihnea Scafeş: Conceptual Framework for Design of Service Negotiation in Disaster Management Applications, In: Bai, Quan and Fukuta, Naoki (eds.), Advances in Practical Multi-Agent Systems, *Studies in Computational Intelligence 325* (2010), Springer Verlag, 359–375.

3. Nicholas R. Jennings, Peyman Faratin, Alessio R. Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge: Automated Negotiation: Prospects, Methods and Challenges, *Group Decision and Negotiation 10(2)* (2001), Springer, 199-215.
4. Shamimabi Paurobally, Valentina Tamma, and Michael Wooldridge. A Framework for Web service negotiation, *ACM Transactions on Autonomous and Adaptive Systems 2(4)* (2007), ACM Press.
5. Gregor Pavlin, Michiel Kamermans, and Mihnea Scafeş: Dynamic Process Integration Framework: Toward Efficient Information Processing in Complex Distributed Systems, In: *Informatica 34* (2010), 477–490.
6. Ate Penders, Gregor Pavlin and Michiel Kamermans: A Collaborative Approach to Construction of Complex Service Oriented Systems, In: Essaaidi, Mohammad Malgeri, Michele and Bǎdicǎ, Costin (eds.), Intelligent Distributed Computing IV, *Studies in Computational Intelligence 315* (2010), Springer Verlag, 55–66.
7. Mihnea Scafeş, Costin Bǎdicǎ, Gregor Pavlin, and Michiel Kamermans: Design and Implementation of a Service Negotiation Framework for Collaborative Disaster Management Applications, In: *Proc.International Conference on Intelligent Networking and Collaborative Systems INCOS'2010* (2010), 519–524.
8. Reid G. Smith, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers*, *29(12)* (1980), IEEE Computer Society, 1104–1113.
9. Michael Wooldridge, *An Introduction to MultiAgent Systems* (2nd ed.), John Wiley & Sons, (2009).