A new Approach to Collaborative Information Processing in Complex Environmental Management Problems

Gregor Pavlin, Michiel Kamermans, and Kees Nieuwenhuis

Thales Nederland B.V., D-CIS Lab
Postbox 90, 2600AB, Delft, the Netherlands
gregor.pavlin@d-cis.nl, michiel.kamermans@d-cis.nl,
kees.nieuwenhuis@d-cis.nl

Abstract. Contemporary environmental management applications often require situation assessment based on the processing of large quantities of heterogeneous information and rich domain knowledge. This paper discusses an efficient solution to such processing challenges, which is based on the recently introduced Dynamic Process Integration Framework (DPIF), a service oriented approach to collaborative reasoning. The DPIF supports (i) a systematic encapsulation of heterogeneous processes via software agents and (ii) self configuration mechanisms that automate creation of meaningful workflows implementing complex collaborative reasoning processes. In addition, the DPIF is supported by novel tools and methods which facilitate construction of service oriented systems using rich domain knowledge while, at the same time, the collaboration between heterogeneous services requires minimal ontological commitments.

1 Introduction

Contemporary environmental management applications frequently require situation assessment based on reasoning about complex processes and phenomena. Often this can be achieved only through adequate processing of large quantities of very heterogeneous information, based on rich expertise about different aspects of the physical world. However, the processing requirements usually exceed cognitive capabilities of a single human expert; an expert typically does not have knowledge of all the relevant mechanisms in the domain and cannot process huge amounts of available information. Therefore, complex assessment is often carried out in systems which can be characterized as professional bureaucracies [9], 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 illustrate such processing by using an example derived from a real world use case investigated in the FP7 DIADEM project. For the sake of clarity but without the loss of generality, we assume a significantly simplified scenario: in a chemical incident a leaking chemical starts burning which results in harmful fumes. The impact of the resulting fumes is assessed through a collaboration of experts captured by figure 1. The factory staff (FS) at the incident can estimate the quantity of the escaping chemical and its type. This information is passed on to a chemical expert at the incident location (CE1) who estimates the type and quantity of toxic fumes. By knowing the location of the fire, the meteorological conditions, and the quantity and type of the produced fumes, chemical expert (CE2) can estimate the zones in which the concentration of toxic gases exceeded critical levels and identify areas which are likely to be critical after a certain period of time. CE2 uses domain knowledge about the physical properties of the gases and their propagation mechanisms. In addition, CE2 improves the estimate of the critical area by using (i) information about the distribution of health complaints obtained from a dispatch center (CEC) and (ii) gas concentration measurements obtained at specific locations by measurement teams (MT). A map showing the critical area is supplied to the health expert (HE) who estimates the impact of the toxic fumes on the population in case of exposure.

Unfortunately, such assessment is often jeopardized through inability to quickly establish adequate communication between the relevant experts; the use of traditional communication means, such as phones, typically results in a significant communication overload and does not facilitate dissemination of rich information, such as for example pictures, movies and annotated maps.

We tackle these challenges with the help of the Dynamic Process Integration Framework (DPIF). It combines Multi Agent Systems (MAS) and a service oriented paradigm in new ways which facilitate implementation of hybrid collaborative reasoning systems with emergent problem solving capabilities. 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 put service providers into workflows and facilitate information dissemination. In addition, some of the reasoning processes might be automated, which would speed up the assessment and improve its quality. However, full automation of complex assessment processes is likely to be unacceptable or even impossible. Therefore, the DPIF supports collaborative processing based on a combination of automated reasoning and cognitive capabilities of multiple human experts, each contributing specific expertise and processing resources. Note that, in contrast to traditional MAS approaches [3], the DPIF facilitates integration of human cognitive capabilities right into the problem solving processes in workflows; humans are not mere users of an automated system, but contribute the processing resources.

The DPIF supports service composition which explicitly takes into account the characteristics of Professional Bureaucracies [9]. As we show later, composition of services in such settings can be achieved through service discovery based on local domain knowledge; each expert or process knows which types of information (i.e. other service types) are required for providing a specific service.

Consequently, the DPIF does not require centralized service ontologies describing relations between the services and centralized service composition methods. Instead, we use simple service ontologies which serve primarily for the alignment of the semantics and syntax of messages exchanged between the processes in workflows.

In this way we obtain systems which support processing based on rich domain knowledge while, at the same time, the collaboration between heterogeneous services requires minimal ontological commitments [2].

This paper provides a rationale for using DPIF in combination with novel approaches to collaborative construction of service ontologies to solve a relevant class of environmental management challenges. Theoretical and technical details of the mentioned components, tools and methods can be found in [5, 6].

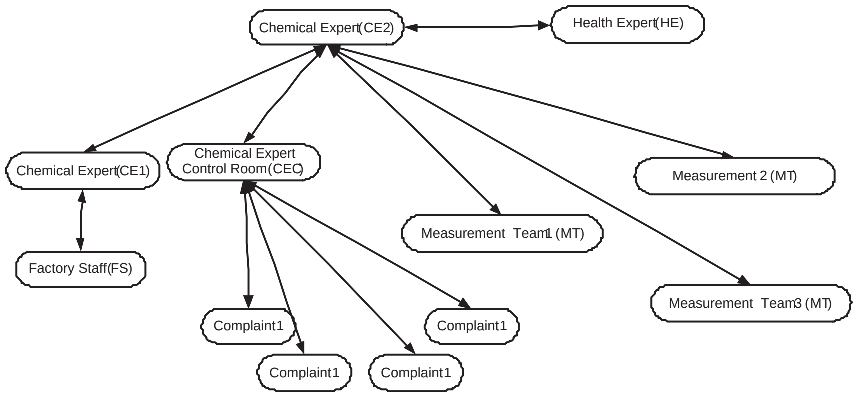

Fig. 1. A collaborative situation assessment in an environmental management process. Arrows denote information flow between different experts, each processing relevant information of different types.

2 Collaborative Processing

The presented environmental management example shows that reasoning in a situation assessment process is based on data-driven workflows established between heterogeneous processes. In such workflows difficult problems can be solved through collaboration of heterogeneous processes, each focusing on a relatively small subset of relevant aspects in the targeted domain.

Moreover, we reason about a situation which can be viewed as a specific combination of known types of events and processes, each understood by a human expert or modeled by an artificial agent. For example, the way chemicals burn and react, the effects of exposure to toxic fumes, etc. are independent of the location and time. Therefore, we can obtain general knowledge about such processes which can be used for the analysis in any situation involving such phenomena. In other words, we can assign roles to different experts and artificial agents based on their domain knowledge and models prior to the operation. However, since each situation (e.g. chemical incident) is a unique combination of known types of events, a specific workflow consisting of a particular combination of processing nodes is required for adequate situation assessment. In addition, due to unpredictable sequences of events it is impossible to specify an adequate workflow a priori. For example, given the wind direction, experts for the evacuation of hospitals and schools might be needed. However, if the gas is blown to the open sea instead, no evacuation experts are needed in the situation assessment process.

Clearly, a major challenge is creation of adequate workflows which correctly integrate the relevant processes and support globally coherent processing in decentralized collaborative systems. This can be achieved with the help of the Dynamic Process Integration Framework described in the next section.

2.1 Dynamic Process Integration Framework

The Dynamic Process Integration Framework (DPIF) supports decentralized creation of workflows that facilitate collaborative problem solving. The DPIF is a service-oriented approach (SOA) which supports efficient composition of very heterogeneous processing services provided by different experts and automated reasoning processes. In the context of the DPIF, information processing is abstracted from human or machine instances; a reasoning process is either provided by a human expert or an automated system implemented by a software agent. Each process provides a well defined reasoning service in the form of an estimate, prediction, cost estimate, etc. The inputs for each of such processes are provided by other processes or by direct observations (i.e. sensor measurements and reports from humans).

A human expert or an automated inference process is represented in the system by a software agent, a functional (i.e. processing) module which (i) supports standardized collaboration protocols and (ii) allows incorporation of arbitrary reasoning approaches. In other words, the agents provide a uniform communication/collaboration infrastructure allowing seamless combination of heterogeneous processes provided by human experts or implemented through AI techniques. Each agent registers in the DPIF-based system (i) the services supported by its local processing capabilities and (ii) the required inputs, i.e. types of services that should be provided by other agents in the system.

By using the registered services, agents distributed throughout different networked devices can autonomously form workflows in which heterogeneous processes introduce collaborative reasoning. Figure 2 shows a simplified example of such a workflow. The configuration of workflows is based on the relations between services captured by local models; each agent *knows* what service it can provide and what it needs to do this. This local knowledge is captured by the relations between the variables in partial domain models. Thus, *no centralized ontology describing relations* between different services of various agents is required, the creation of which is likely to be intractable.

In other words, globally coherent collaborative processing is possible by combining local processes, without any global description of relations between inputs and outputs.

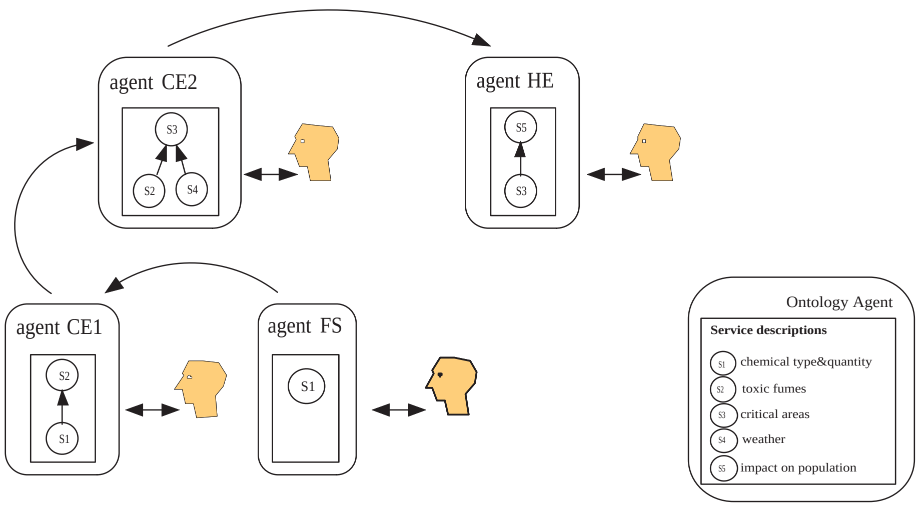

Fig. 2. An example of a DPIF-based workflow involving only human experts. The represented DPIF agents are assisting human experts; they establish communication between experts, each processing specific information. Directed links connecting agents correspond to the information flow between different experts.

3 Processing Workflows

A basic workflow element in the DPIF is a local process. Moreover, in the following discussion the term local process refers to a reasoning process provided either by a human expert or an automated system implemented by a software agent. Each local process corresponds to a function $F : \{X_1, ..., X_n\} \rightarrow Y$, mapping values in a domain $\{X_1, ..., X_n\}$ to values of some variable of interest Y. The value of Y for particular values of arguments is given by $y = f_y(x_1, ..., x_n)$.

Such functions can be either explicit, based on some rigorous theory, or implicit, when they are provided by humans or sub-symbolic processes, such as for example neural networks. An example of a mathematically rigorous mapping is the function $x_{CE1} = f_{x_{CE1}}(x_{FS})$, an explicit formula describing the relations between the fume volume per time unit represented by X_{CE1} and the escape rate of chemicals denoted by X_{FS}. This function is used by the Chemical Expert CE1 in figure 1. An implicit mapping, on the other hand, is performed by the health expert (HE) who estimates the critical regions with respect to the impact on the residents. HE interprets information about critical concentration X_{CE2} in combination with information on population distribution X_{POP} by using an implicit function $x_{HE} = f_{x_{HE}}(x_{CE2}, x_{POP})$.

3.1 From Local to Global Processes

An expert or an artificial agent often cannot observe values of certain variables; i.e. variables cannot be instantiated. Instead, the inputs to the local function are supplied by other processes forming a collaborative workflow (see section 2). Thus, the inputs to one function are outputs of other functions used by the information suppliers. From a global perspective this can be seen as a function composition; in a function, each variable which cannot be instantiated is replaced by a function. This process continues until a function is obtained in which all variables are instantiated, i.e. all free variables in the resulting nested function have been reduced to direct observations. In this way, a global function emerges as different processes are connected in a workflow. The resulting function is a composite mapping between directly observable variable states and hidden variables of interest.

In other words, a workflow in a DPIF system corresponds to a full composition of functions, in which each variable replaced by a function corresponds to a required service. This yields the value of the variable of interest. Let's assume an example with six service suppliers shown in figure 3(a), using the following functions:

$$x_a = f_a(x_b, x_c), x_b = f_b(x_d), x_c = f_c(x_e, x_f),$$
$$x_d = f_d(x_g), x_e = f_e(x_h), x_f = f_f(x_i).$$

then the workflow supporting collaborative computation of the value for x_a corresponds to the composite function

$$f_a(f_b(f_d(x_g)), f_c(f_e(x_h), f_f(x_i))) \tag{1}$$

It is important to bear in mind that in DPIF no explicit function composition takes place in any of the agents. Instead, the sharing of function outputs in a workflow corresponds to such a composite function; i.e. a workflow models a (globally emergent) function, mapping all observations of the phenomena of interest (i.e. evidence) to a description of some unknown state of interest.

Each workflow corresponds to a system of systems, in which exclusively local processing leads to a globally emergent behavior that is equivalent to processing the fully composed mapping from direct observations to the state of the variable of interest.

4 Dynamic Service Ontologies

In order to be able to automatically compose heterogeneous services provided by different developers or experts, the definitions of service interfaces have to be standardized, which is achieved with the help of explicit service ontologies.

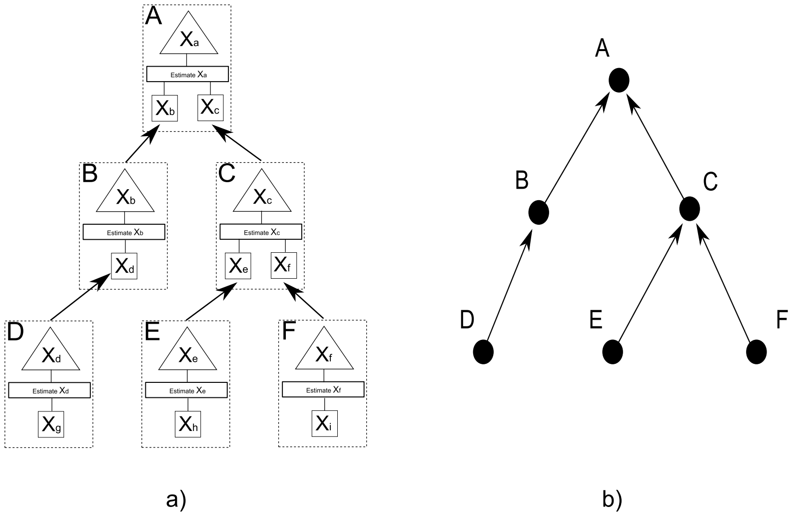

Fig. 3. a) A self-organized system of agents, each supplying information concerning a particular variable. These outputs are based on other inferred or directly observed variables. b) A directed graph capturing the workflow between the agents from a).

Services declared in the DPIF are typically provided by many stakeholders from different organizations whose capabilities evolve with time. Large systems of service descriptions have to be maintained and it is very difficult to specify a complete set of services prior to the operation. In other words, traditional approaches based on rigorous centralized ontologies, such as for example [1, 7], which capture service descriptions and relationships between information provided by different types of services are not practical; we simply do not know which relevant services will be available in the future and maintenance of large ontologies is likely to be very expensive or even intractable.

Fortunately, the locality of domain knowledge in the DPIF approach supports efficient creation of service ontologies. Because self organization and processing are based on domain knowledge encoded in local functions, we can avoid traditional approaches to constructing centralized ontologies, which describe domains in terms of complex relations between the concepts corresponding to the processing services. Instead, the services and relations between them are described by using two types of light weight ontologies:

- *The global service ontology* merely captures service descriptions, the semantics and syntax of messages used for (i) service invocation and (ii) dissemination of service results. This ontology is used for the alignment of the semantics and syntax of service descriptions at design time.
- *Local task ontologies* coarsely describe relations between different types of services supplying different types of information. In principle, they describe which types of services provide inputs to the function used by a specific

service. These relations reflect the local knowledge of each processing module. Moreover, the local ontology supports runtime creation of workflows based on service discovery.

The global ontology is a central element of the service description procedure. In order to make sure that all agents speak the same language, the global ontology captures three types of elements, namely (i) a verbal description of the service to be provided, (ii) conditions under which this service can be invoked, and (iii) a collection of representational elements resulting from the information gathered by this service. While the vocabulary with which these descriptions can be specified is rigidly formalized, it is rich enough to allow the description of arbitrarily complex services. The global ontology is used by a matching process in which service suppliers are provided with a list of existing service descriptions, based on keywords and free text. The descriptions retrieved from the global ontology are displayed in a form that facilitates inspection of the relevant subset of existing services. If an existing service description corresponds to the new service, it is adopted. Otherwise a service definition editor allows the experts to provide a new service description, which is then added to the global ontology. By making experts responsible for deciding whether they perform a role similar to another domain participant or a genuinely new role, we overcome the problem of an a priori defined ontology that is likely to be unable to account for all aspects of the domain and expert capabilities.

The *local task ontologies*, on the other hand, are created with the help of a task definition tool which supports specification of the required inputs (provided by other services) for each provided service. In this way different services are related locally, based on the local domain knowledge. The task ontologies are stored with agents of participating experts. The boxed directed graphs inside each agent in Figure 2 represent local task ontologies associated with the assistant agents. Arrows in the graphs indicate relations between the different service types: services corresponding to the leaf nodes provide inputs the service provided by the respective agent. These relations captured by local task ontologies are central to the service discovery, which is typically initiated from within the local services. Consequently, if each expert is made responsible for the description of relations between the provided and the needed services, systems using complex relations between services can be built in a collaborative way, without any centralized configuration/administration authority.

In other words, the introduced division into a global service ontology and local task ontologies dispersed throughout the system of agents allows collaborative definition of services by experts.

Construction of such ontologies is facilitated by the OntoWizzard, a tool that (i) helps the users discover the services in the global ontology by using keywords and written language, (ii) provides an interface facilitating inspection of the human readable descriptions and (iii) has editors for defining local task ontologies. By using this tool, the experts define elements of the global service ontology and the local task ontologies without using any formal language. At the same time, the tool automatically translates expert inputs to rigorous local and global ontologies captured in the OWL format. Normally, human operators do not have a direct access to the global service ontology. Instead, a special ontology agent (see Figure 2) provides search services and maintains the OWL files defining the global service ontology. In other words, by deploying the two types of ontologies in combination with simple construction procedures, rigorous, machine understandable service descriptions can be created without any formal knowledge of the underlying ontology techniques. Detailed discussion on the local and global ontology as well as the OntoWizzard tool can be found in [6].

Similarly to the DPIF approach, the OpenKnowledge framework [8] avoids creation of centralized heavy weight ontologies describing all aspects of the domain. However, while the DPIF requires a mere specification of the provided and supplied services, the OpenKnowledge framework also requires specification of *interaction models* shared by the collaborating peers. Such interaction models define workflows for each processing task a priory; the OpenKnowledge approach assumes that collaborating peers understand interaction protocols and the processing sequences of collaborating peers. This can introduce additional complexity to the system configuration in which services and processes are specified. Since the DPIF is targeting Professional Bureaucracy systems [9], it is assumed that experts do not share knowledge about local processes.

5 Conclusions and Future Work

The DPIF supports uniform encapsulation and combination of heterogeneous processing capabilities which facilitate collaborative reasoning in complex situation assessment problems in environmental management applications. In the DPIF context, human expertise and automated processes are abstracted to functions with well defined outputs and inputs; each function provides a particular reasoning service given certain inputs.

The DPIF provides *function wrappers*, software agents which standardize function interfacing. The interfaces are based on standardized service descriptions as well as uniform self-configuration, negotiation and logical routing protocols. With the help of the DPIF encapsulation methods very heterogeneous services can be made composable and negotiable.

The DPIF agents support automatic formation of workflows in which heterogeneous functions correspond to suppliers and consumers; outputs of some functions are inputs to other functions and so on. In other words, a workflow corresponds to a set of nested functions that captures dependencies between very heterogeneous variables. Creation of workflows and routing of information is based on the relations between different types of information. These relations are captured by *local functions* wrapped by different modules. The DPIF approach assumes that each expert or an automated process can declare the inputs and outputs of the contributed local functions, which is sufficient for automated creation of globally meaningful workflows by using service discovery. Thus, in contrast to traditional approaches to processing in workflows, *neither centralized configuration of workflows nor centralized knowledge of the combina-*

tion or routing rules are needed. The resulting systems support processing based on rich domain knowledge while, at the same time, collaboration between heterogeneous services requires minimal ontological commitments. Combined with the proper communication services, a DPIF-based systems facilitate distribution of processes over multiple platforms and networks. Decentralized creation of emergent processing workflows is useful in large scale environmental management problems, where it is difficult to maintain a centralized overview of the sensing and processing resources.

A basic version of the DPIF and the service configuration tool OntoWizzard have been implemented and are being evaluated in the context of the FP7 DIADEM project. Moreover, a fully automated DPIF variant using Bayesian networks is used for robust distributed gas detection and leak localization [4]. Currently the DPIF is being enhanced with advanced negotiation mechanisms and interfaces to Multi Criteria Decision Analysis tools and Scenario Based Reasoning methods streamlining human-based processing in workflows.

6 Acknowledgments

The presented work is partially funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D, ref. no: 224318.

References

1. David Chiu and Gagan Agrawal. Enabling ad hoc queries over low-level scientific data sets. In *SSDBM*, pages 218–236, 2009.
2. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, 1995.
3. Nicholas R. Jennings, Katia P. Sycara, and Michael P. Georgeff. Editorial. *Autonomous Agents and Multi-Agent Systems*, 1(1):5, 1998.
4. Gregor Pavlin, Frans C. Groen, Patrick de Oude, and Michiel Kamermans. *A Distributed Approach to Gas Detection and Source Localization Using Heterogeneous Information*. Springer Verlag, 2010. ISBN 978-3-642-11687-2, in print.
5. Gregor Pavlin, Michiel Kamermans, and Mihnea Scafes. Dynamic process integration framework: Toward efficient information processing in complex distributed systems. In *IDC*, pages 161–174, 2009.
6. Ate Penders, Gregor Pavlin, and Michiel Kamermans. A collaborative approach to construction of complex srvice oriented systems. In *IDC*, 2010.
7. Quan Z. Sheng, Boualem Benatallah, Marlon Dumas, and Eileen Oi-Yan Mak. Self-serv: A platform for rapid composition of web services in a peer-to-peer environment. In *VLDB*, pages 1051–1054, 2002.
8. Ronny Siebes, David Dupplaw, Spyros Kotoulas, Adrian Perreau de Pinninck Bas, Frank van Harmelen, and David Robertson. The openknowledge system: an interaction-centered approach to knowledge sharing. In *Proceedings of the 15th Intl. Conference on Cooperative information systems (CoopIS) in OTM 2007*, 2007.
9. Chris J. van Aart, Bob Wielinga, and Guus Schreiber. Organizational building blocks for design of distributed intelligent system. *International Journal of Human-Computer Studies*, 61(5):567 – 599, 2004.