AUGMENTING SEMANTICS TO DISTRIBUTED AGENTS LOGS

Enabling Graphical After Action Analysis of Federated Agents Logs

Rani Pinchuk[1], Sorin Ilie[2], Thomas Neidhart[1], Tiphaine Dalmas[3], Costin Bădică[2] and Gregor Pavlin[4]

[1] *Space Applications Services, Leuvensesteenweg 325, Zaventem, Belgium*

[2] *University of Craiova, Software Engineering Department, Craiova, Romania*

[3] *Aethys, Edinburgh, U.K.*

[4] *Thales Nederland B.V., D-CIS Lab, Delft, The Netherlands*

Keywords: Software agents, Topic Maps, Ontology, Logging, After action analysis, Crisis management, Decision Support, Graphical querying.

Abstract: A distributed multi-agent system is used to support collaborative situation assessment and decision making for effective management of chemical hazards crisis in industrial areas. The system of agents supports creation of complex information flows between large numbers of stakeholders. The disseminated information and the system states are logged, which supports the analysis of the collaborative crisis management processes as well as the performance of the multi-agent systems framework. An ontology is designed to model the logged process. The fragments of logs that are meaningful to the users are converted to topic maps using the designed ontology. These topic maps are then merged to provide a federated picture of the data. A graphical query mechanism for querying the topic maps has been developed. This query mechanism creates graphical representations of relevant excerpts of the merged topic map, allowing conducting a thorough analysis of the logs.

1 INTRODUCTION

In modern industrial areas many factories deal with hazardous materials. Because of the combined growth of the industry and the residential areas, it is very common to find houses very close to factories.

Such is the case, for example, in Rijnmond, a heavily industrialized and densely populated area in Rotterdam, the Netherlands. In such locations, a harmful chemical leak must be dealt with immediately. The leaking chemical compound must be identified, the leaking facility must be located, the weather conditions must be assessed and, accordingly, the development of the leaked gas plume should be estimated. The danger that the leak poses to the residents should then be understood and, as a result, decisions should be taken in order to protect the population (i.e. immediate evacuation or shelter).

Managing such a crisis requires expertise in many domains. For example, experts in the chemical plants can assess the type and amount of the escaped gas. Experts familiar with gas distribution models can predict the gas plume basing on meteorological information and environment conditions (provided by other experts). Medical experts can then estimate the danger to the population. Other experts have to check the available transportation infrastructures.

As information and knowledge are spread across multiple experts, communication and workflows have to be coordinated.

The Dynamic Process Integration Framework (DPIF) is a service-oriented architecture that supports such collaborative crisis management by automating the management of communication between the experts, and facilitating workflows (Pavlin, 2010).

DPIF includes a multi-agent system *where each agent represents a human expert, or other modules* (e.g. a decision support module or a gas detector). Through the DPIF system, the experts can register themselves as providing a certain service (i.e. expert for predicting the evolution of the gas plume), and as dependent on other services (i.e. weather report). Later on, when coping with an actual situation, the experts will get requests to provide their expertise.

Pinchuk R., Ilie S., Neidhart T., Dalmas T., Bădică C. and Pavlin G..
AUGMENTING SEMANTICS TO DISTRIBUTED AGENTS LOGS - Enabling Graphical After Action Analysis of Federated Agents Logs.
DOI: 10.5220/0003658802320241
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2011), pages 232-241
ISBN: 978-989-8425-80-5

The system will automatically manage these communications, and build the necessary workflows (i.e. the weather report must be provided before the prediction of the plume can be done).

Because crisis situations tend to be very hectic, it is important to be able to conduct a thorough after action analysis following trainings and actual incidents.

Of course, such after action analysis can take advantage of the fact that most communication is done using the agents, and therefore logs are available to fully follow the chain of events (Ilie, 2010).

However, it is not trivial for end users to understand the full picture, and answer their questions, solely basing on technical log entries. Moreover, because the system is naturally distributed, different log files are created by the agents in the different locations. A federation of these data must be done before the full situation can be understood.

We cope with these challenges by converting the logs to semantic representations where the information meaningful for the users conducting the after action analysis is kept. These representations are then merged, and the resulted data is then queried and browsed graphically.

In order to further explain the approach we first introduce the details of the DCMR[1] use case (Badica, 2011). Later, we present the Topic Maps ontology designed for representing the information from the logs that is meaningful to the users. The conversion of the log files to topic maps and their merge is described next and finally a graphical query and browsing user interface over the data is shown.

2 THE DCMR USE CASE

We illustrate our approach by using an example derived from a real world use case investigated in the FP7 DIADEM project (http://www.ist-diadem.eu). For the sake of clarity we assume a significantly simplified scenario. In a chemical incident at a refinery, a chemical starts leaking and forms a toxic plume spreading over a populated area. The impact of the resulting fumes is assessed through a service composition involving collaboration of human experts, as explained below and shown in Figure 1:

1. 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.
2. 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, she requests a report of the situation from the factory via the Factory Representative. The Factory Representative replies with a report that confirms the incident and provides information about the type of escaping gas.
3. 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.
4. 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.
5. 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. Chemical Adviser 2 makes use of domain knowledge about the physical properties of gases and their propagation mechanisms.
6. In addition, 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.
7. A map showing the critical area is supplied by the Chemical Adviser 2 to a Health Expert. He uses additional 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

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

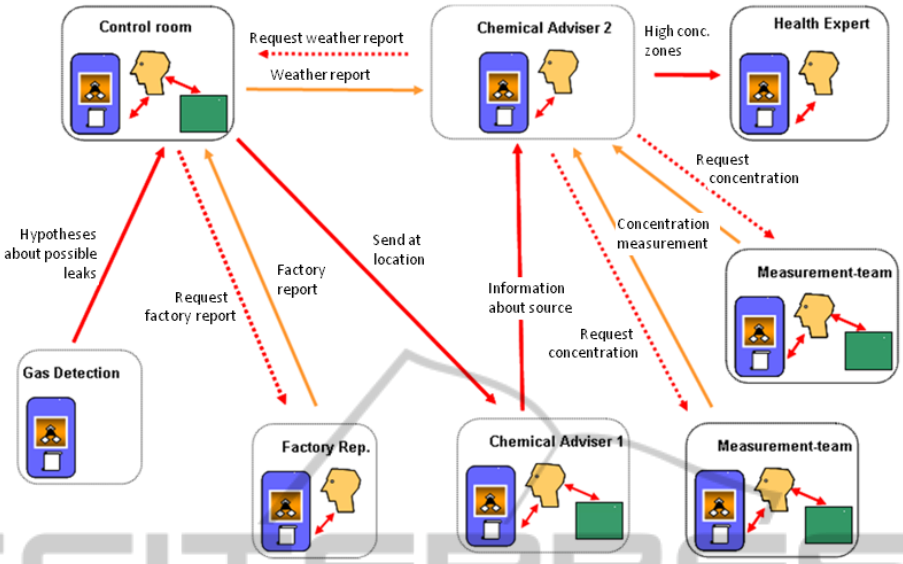

Figure 1: Workflow of the scenario.

Table 1: The provided and required services for each DPIF agent from the utilization scenario.

Agent	Provided Service	Required 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
Measurement Team 1	Concentration measurements	n/a
Measurement Team 2	Concentration measurements	n/a

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 four input services called *Information about source*, *Leak at location*, *Weather report* and *Concentration measurements* providing her with (i) information about the source, (ii) information about the presence gas leak at location, (iii) weather information, and (iv) concentration measurements in the interest area, respectively.

The interactions between DPIF agents during the collaboration workflow described in the utilization scenario can be represented using an AML sequence diagram presented in Figure 2. Note that on this diagram we can identify three types of messages:

- INFORM messages correspond to "bottom-up" information propagation. For example, the Gas Detection system can generate hypotheses about a possible chemical incident via the *Hypothesis* service by processing information received from sensors. Note that "bottom-up" information propagation can trigger other "bottom-up" propagations, as seen for example when the Control Room, based on analyzing the information got from Gas Detection and the information requested and received from Factory Representative agent decided to send Chemical Adviser 1 at the incident location.
- REQUEST messages correspond to ``top-down" requests for information. For example, whenever the Control Room is notified with a hypothesis about a possible incident, she will look for more information about it by explicitly requesting new information from a

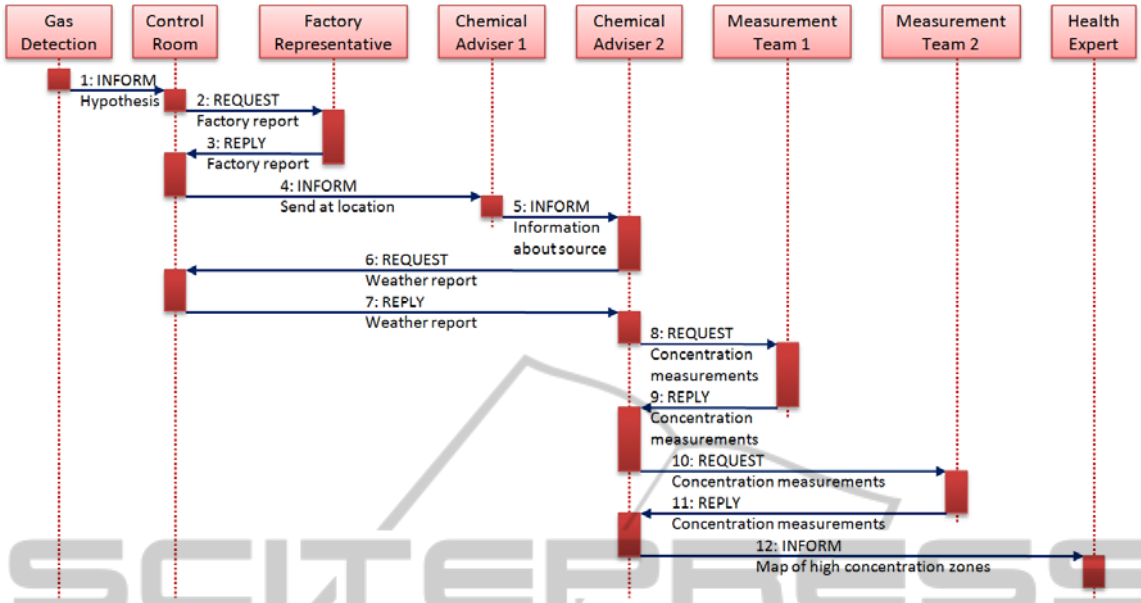

Figure 2: AML representation of the interactions from the scenario.

Factory Representative located at the factory that is suspected as being the cause for the incident. An example in this sense is the message REQUEST Factory Report shown on Figure 2.

- REPLY messages correspond to replies with information that was requested in ``top-down'' fashion. An example on Figure 2 is the message REPLY Factory Report. A REPLY message always follows a REQUEST message in a request-reply interaction.

Moreover, the workflow shown in Figure 2 is not hard-coded anywhere. 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 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.

3 FROM LOG FILES TO TOPIC MAP

3.1 Topic Maps Ontology

The users interested in after action analysis cannot read the agents logs directly. The DPIF logs are written as unique entries represented in JSON (Ilie, 2010).

```
{
 loggerName: 'StructuredLogging.HealthExpert',
 level: 'INFO',
 timeStamp: 1304582631229,
 humanReadableTimeStamp: '2011-05-05 11:03:51',
 message: {
  eventType: 'Message_Sent',
  source: 'HealthExpert',
  context: 'NONE',
  properties: {
   content: {
    from: 'HealthExpert',
    to: 'ChemicalAdviser2',
    sendertaskid: 4,
    receivertaskid: 6,
    data: {
     data: 'Has the information been verified?'
    },
    timestamp: {
     unit: 1000000000,
     value: 1305529442047351
    },
    sequencenumber: 1,
    publisher: 'DeinzillaNegotiationProcessingPlugin'
   },
   to: 'ChemicalAdviser2',
   messageId:'HealthExpert_nl.decis.combined.framework
              .core.components.processors.ChatMessage_
              1_1305529442'
  }
 }
}
```

The logs contain much technical information that is not relevant for the users. Many entries in the logs

are related to other entries, and cannot be understood by themselves. In addition, like the agents themselves, these logs are distributed - which makes the understanding of the overall picture more difficult by just looking at individual logs.

This is where we introduce Topic Maps to model the parts of the logs that are meaningful to the users. Topic Maps is a standard (ISO/IEC 13250:2003) for knowledge representation and information integration. In Topic Maps, each subject in the knowledge domain is represented by a topic. In Figure 3, *Chemical Adviser 2* is a topic representing the agent of Chemical Adviser 2. Each topic may have a type. Here, the topic *Chemical Adviser 2* is typed by the topic *Agent*, meaning that *Chemical Adviser 2* is an agent.

Topics can be associated with other topics. The topic *Chemical Adviser 2* is associated with the topic *Health Expert* and with the topic *Chat Message 9*. The type of this ternary association is *Sends*. The fact that topics are associated with each other provides a structure that corresponds to the way we conceptually grasp knowledge. It also allows asking questions about the knowledge within the topic map, e.g. "What message was sent to the Health Expert?" Association is built from players, role types and association type – all are topics. For example, *Health Expert* plays the role *Destination* in the association of type *Sends* and *Chat Message 9* plays the role *Message* in the same association.

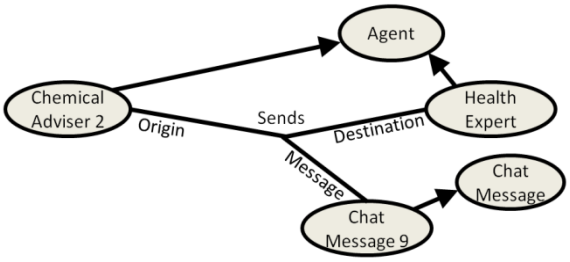

Figure 3: Excerpt of the DCMR logging topic map.

Each topic can hold extra information called occurrences. These are pieces of information about the topic, similar to paragraphs in a book that are referenced by the book index. Occurrences can be simple textual information or references to any other media, such as web pages, audio, video, etc. Like associations, occurrences are also typed. For example, the topic *Chat Message 9* may have an occurrence of type *Data* containing the value "I confirm that the information is indeed verified".

Topic names, occurrences and associations can all be scoped. The scope can then be used in order to filter those elements in or out when using the topic map.

Scopes can be used to provide different perspectives (e.g. describing a scene from different points of view) in the same topic map. They can also be used in order to describe dynamic situations. For example, the message described above was sent at 11:03:51.229 and was received at 11:03:51.316. This *Sends* association can be scoped by a topic that represents this period in which the message is actually sent. This way, a topic map can model the dynamics of a crisis management communication.

As was mentioned above, the topic map contains topics that represent the different types, association types, occurrence types and scopes. This group of topics can be seen as the structure or the skeleton, of the topic map and referred here as Topic Maps ontology.

The Topic Maps ontology used is described in the Figure 4, Figure 5 and Figure 6 below. In these figures, created by Onotoa (Niederhausen, 2009) the suggested GTM (Graphical Topic Maps notation - ISO 13250-7) is used.

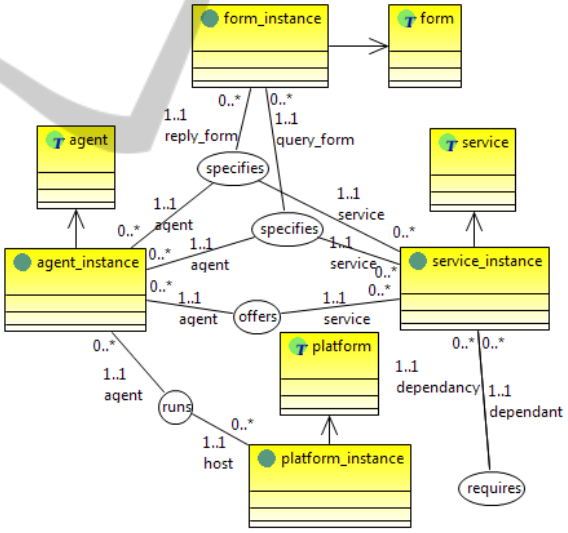

Figure 4: Agents and offered services.

In Figure 4 the agents and their offered services are described. When the agent starts to run, a log entry is added containing the reference to the running platform. It then specifies a service which usually has a query form and a reply form. Note that the actual fields in the forms are described as occurrences of the form instances. The dependencies between the services may be defined as well at that point (i.e. in order to provide a map of high concentration zones, weather report, concentration measurements and information about the source leak

must be available). After the agents specify the services, and their dependencies, they may start to offer these services (Penders, 2010).

Some agents will provide the services bottom up. For example, the gas detector service will send its payload when unusual gas is detected. However, much of the communication between the agents will be based on negotiation process. An agent that needs a certain service will send a call-for-proposal message to other agents offering that service (see Figure 5). The message itself is a filled-in form containing the details of the request. Agents that can provide the necessary service will send a proposal with certain issues (i.e. the service can be provided in certain time frame) to the requesting agent. The requesting agent can then accept the proposal of the proposer.

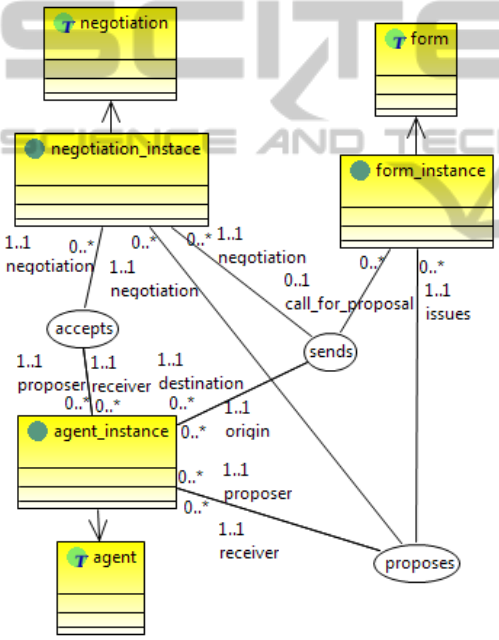

Figure 5: Negotiation.

Figure 6 describes the message ontology. When an agent provides a service to another agent, it sends a payload form and possibly a description form. These are filled-in forms containing the information required. In addition to this message, agents can send to each other chat messages. This chat messages are simple text messages similar to the common SMS.

Note that negotiation as well as bottom-up notification messages are just used to dynamically create links or acquaintances between the agents. Chat messages actually represent the information exchanged by the agents along these dynamically established communication channels.

In order to be able to merge the topic maps created from the different agent logs, each topic must have a unique identifier. In Topic Maps terminology these are the *subject identifiers*.

In and across the log files, agents have unique identifiers. Messages and negotiation sessions have unique identifiers as well.

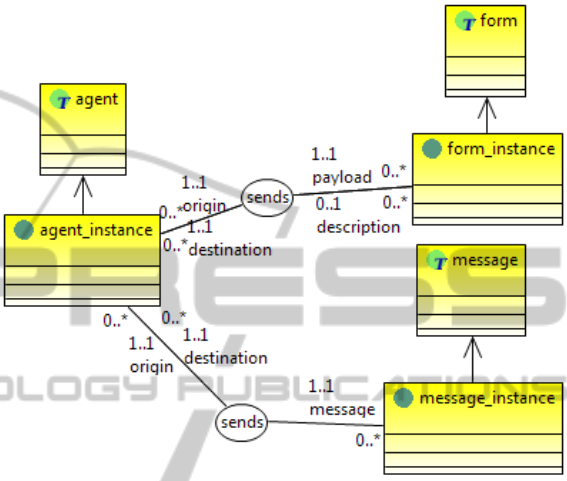

Figure 6: Messages.

The message identifiers are used to relate *messageSent* log entry with *messageReceived* log entry so the start and end time can be used in the scope of the association describing the sending of the message. These identifiers were used also to uniquely identify the forms sent. Note that we have to be cautious with time; if the time stamps are local times recorded automatically by the agents sending and receiving the message, then their clocks are not necessarily synchronized.

The negotiation identifiers were used in order to relate different log entries to the same negotiation session.

3.2 Populating the Topic Maps

As mentioned above, each entry in log files is written as a JSON object.

GSON (http://code.google.com/p/google-gson/) has been used to parse JSON entries (http://www.json.org/). Ontopia engine API has been used to define the topics and the associations between them (http://www.ontopia.net/). Ontopia engine also allowed the merging of the topic maps created for the different logs files. Finally, it was used to export the merged topic map as an XTM 2.0 file (ISO 13250-3: Topic Maps — XML Syntax).

4 VISUAL LINK ANALYSIS

The created topic map contains the relevant information from the distributed log files organized in a meaningful way. However, the end users must be provided with tools to access this information.

A Topic Maps structure can be considered as a graph where nodes represent the information items (topics) and edges between them put them in some defined relation (associations).

Research (Thomere, 2006), (Heim, 2010) for specialized analysis of linked data shows the possibilities for link analysis in different application domains, though the question remains how users can efficiently express the information elements and the relations they are interested in to facilitate their analysis process.

Instead of incrementally filtering out information, one can visually compose a query containing all information elements that are of interest. The resulting visualization links together the elements that match the query and consequently allows the user to identify new relationships. At that point, manual browsing is possible to further analyze the available data.

In Figure 7, the basic concept of visual querying is illustrated. The left part of the figure defines a query that shall include two specific nodes and any possible path between them. The result is a spanning tree of the sub-graph constrained by our query, which can be seen on the right.

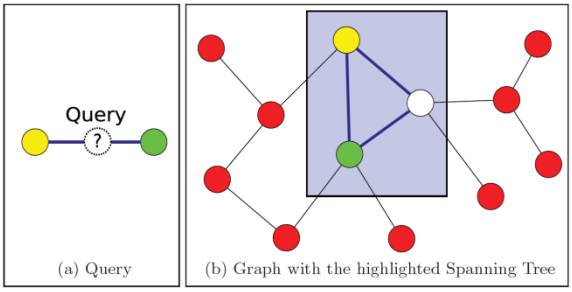

Figure 7: Visual query example.

It is possible to define different kinds of queries using this approach. Constrains over the topic types, and/or their names are possible. Also constraining the association types or role types is possible.

For example, Figure 8 shows a query to find topics of type *Service* who are associated with the association of type *Requires* to other topics of type *Service*.

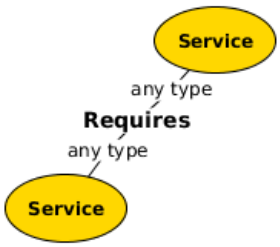

Figure 8: Query - dependencies of services.

The result of this query can be seen in Figure 9: the full dependency chain of the services is provided in a very visible and clear manner.

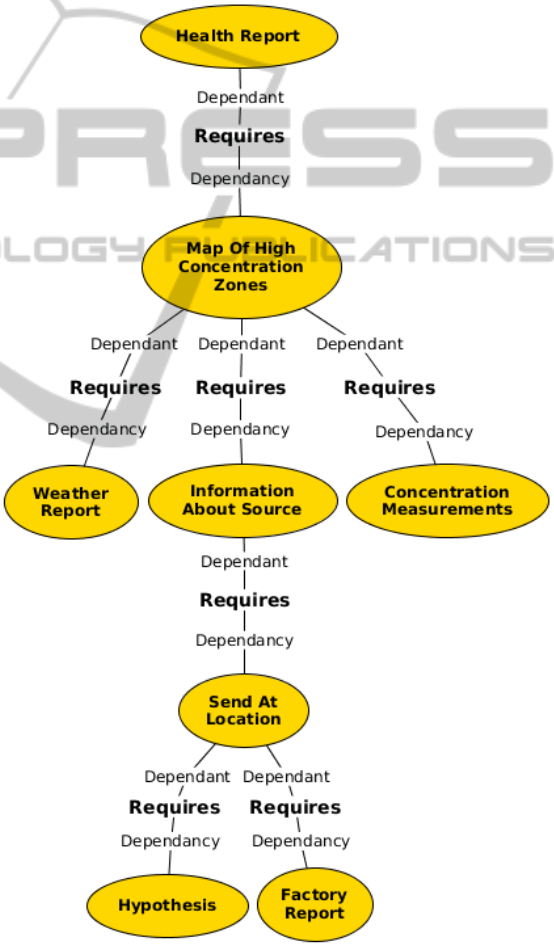

Figure 9: Result - dependency of services.

One can query how the service *Health Report* is related to the service *Information About Source*. In that case, constraints for the names of these two topics should be added, however the constraint over the association type is not necessary. This query can be seen in the left side of Figure 10. On the right side the result of that query is presented. It can be

seen that the relation between these two topics is found by following two associations. The query processor tries first to find links with one association, continues for links with two associations and so on. In order to avoid running exceedingly deep queries, the user may adjust the maximum search depth.

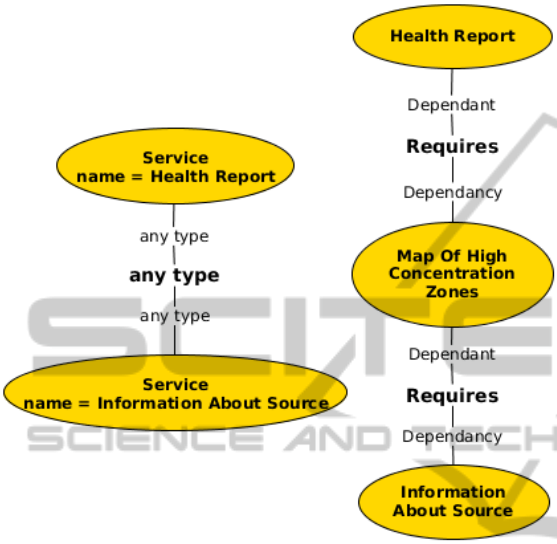

Figure 10: Finding relation between two specific services.

When a result graph is provided, the user may further browse through it. For example, in order to check which agent offers the health report, the user can right click on the *Health Report* service, and select to browse to the *Health Expert* through the association of type *Offers* as shown in Figure 11.

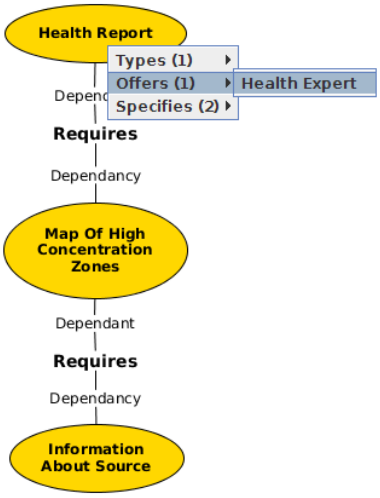

Figure 11: Browsing interface.

The resulting graph is shown in Figure 12.

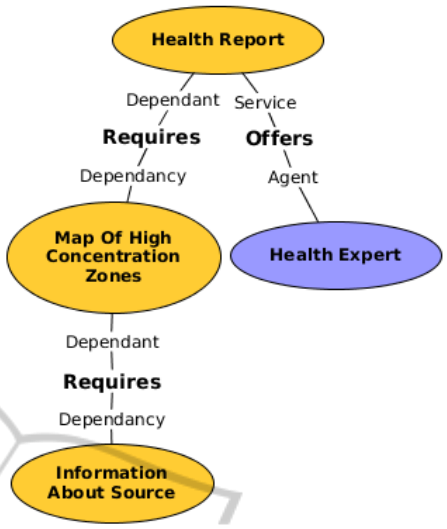

Figure 12: Browsing result.

Another interesting feature provided is the ability to filter according to time on the query results. In Figure 13 the relationship between two agents is queried.

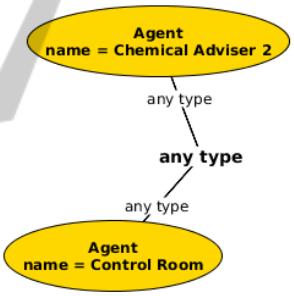

Figure 13: Querying relation between two agents.

The resulted graph, shown in Figure 14, is quite complex to understand as it includes the negotiation steps between the two agents, as well as the resulted message containing the requested report. Moreover, it is not clear when each message is sent. Especially during after action analysis it is very crucial to understand the timing of various events.

This is accommodated by introducing a time filter exposed to the user as a time slider. The user can slide two time range markers to define a time range and associations scoped with periods outside of the chosen range are filtered out. This way, the result shown in Figure 14 can be clarified as shown in Figure 15, Figure 16, Figure 17 and Figure 18.

In order to simplify managing the time slider, it includes ticks that represent the events available in the graph. This way, the user can easily adjust the range markers to the next event.

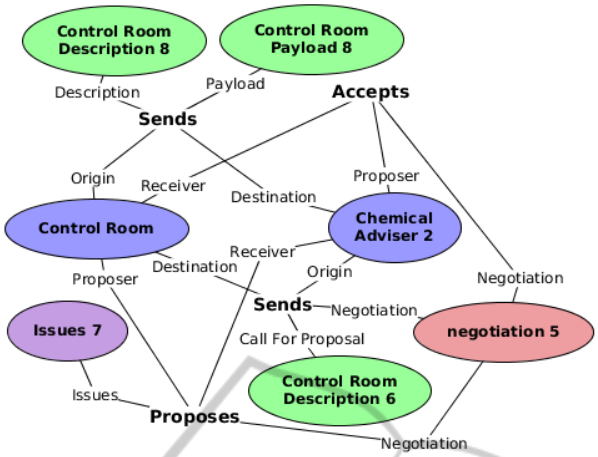

Figure 14: Communication between the Control Room and the Chemical Adviser 2 agents.

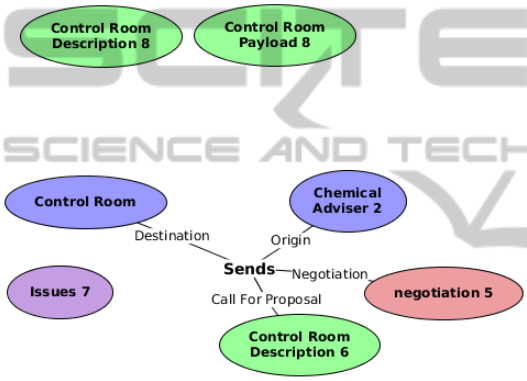

Figure 15: Call for proposal is sent.

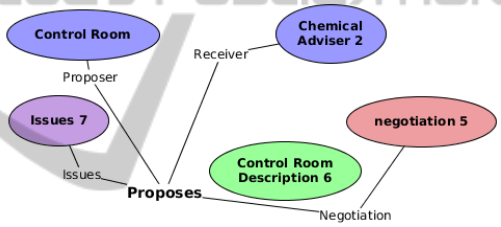

Figure 16: Proposal is sent back.

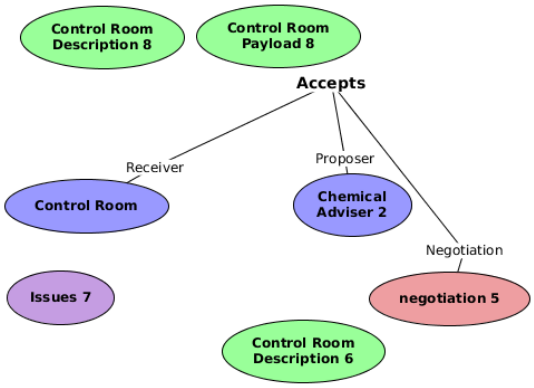

Figure 17: The proposal is accepted.

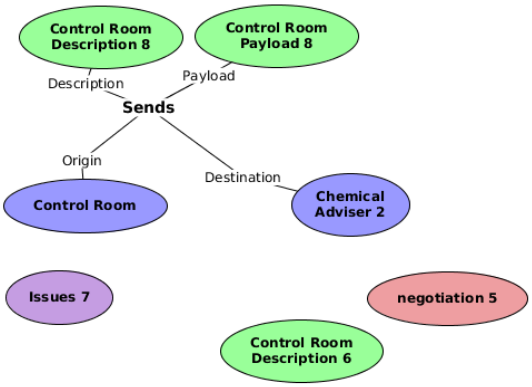

Figure 18: The service is provided.

5 CONCLUSIONS AND FUTURE WORK

Log files contain vital information for conducting useful after action analysis. However, especially in complex and distributed settings, the logged data is too large and complex to analyze.

It has been shown how to convert log files from a DPIF-based distributed multi agent system into one merged topic map. The transformation exploits the inherent structure of the messages as well as the

protocols used in the system. The created topic map contains the relevant data for the users.

Having the topic map enables the users to use different tools for analyzing the events logged. A novel visual link analysis technique has been presented, where the user can define queries in a graphical and intuitive manner. It was shown also how the user can further browse from the resulted graphs, and how the graphs can be filtered by time in order to analyze dynamic situations.

The concept is also portable to other domains: it can be applied to any log-producing application that is bound by an ontology. The only additional efforts would be the design of the Topic Map ontology and the population of the Topic Map from logs.

As future work, we are currently working on merging agent logs with additional data coming from procedure guide software. The procedure guide (see Figure 19) allows users to define and to follow procedures during crisis situation or in other complex settings. The different procedures and related resources are kept in a topic map. Moreover, the actual actions of the users, while following a procedure, are logged as well in a topic map.

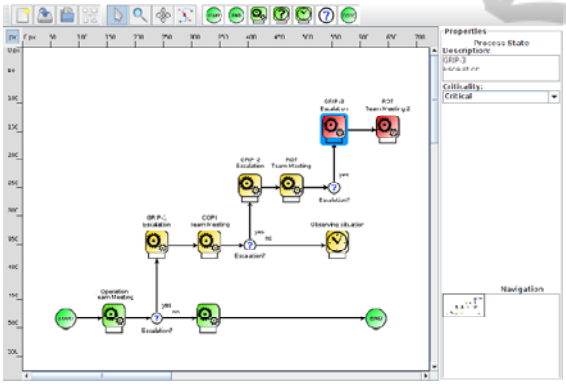

Figure 19: Procedure Guide.

Merging the topic maps of this procedure guide with the topic map representing the agents' logs will allow analyzing not only what the different professionals did but also to align that with what they were supposed to do.

ACKNOWLEDGEMENTS

The work reported in this paper received funding from the European Union Seventh Framework Programme (FP7) under grant agreement n°224318 within the DIADEM project.

REFERENCES

Thomere, J., Wolverton, M.: Presentation of Information for Link Analysis. *In: Capturing and Using Patterns for Evidence Detection - Papers from the AAAI Fall Symposium*, pp. 99-104, AAAI Press (2006).

Heim, P.; Lohmann, S.; Stegemann, T.: Interactive Relationship Discovery via the Semantic Web. *In: Proceedings of the 7th Extended Semantic Web Conference* (ESWC2010), Part I, LNCS 6088, pp. 303-317, Springer, Heidelberg (2010).

Ilie, Sorin; Scafes, Mihnea; Badica, Costin; Neidhart, Thomas and Pinchuk, Rani, Semantic logging in a distributed multi-agent system. *In: Computational Cybernetics and Technical Informatics* (ICCC-CONTI), 2010 International Joint Conference (2010).

Niederhausen, Hannes, Onotoa - a Visual Topic Map Schema Editor. In: Maicher, L.; Garshol, L. M. (Eds.): Linked Topic Maps. *Fifth International Conference on Topic Maps Research and Applications*, TMRA 2009 Leipzig, Germany, November 12-13, 2009, Revised Selected Papers. *Leipziger Beiträge zur Informatik*.

Gregor Pavlin, Michiel Kamermans, Mihnea Scafes: Dynamic Process Integration Framework: Toward Efficient Information Processing in Complex Distributed Systems. *Informatica* (Slovenia) 34(4): 477-490 (2010)

David Damen, Gregor Pavlin, Cor Van Der Kooij, Ray Desmidt, Vanessa Evers, Andi Winterboer, Costin Badica, Tina Comes, Achim Lilienthal, Sahar Asadi, Bernard Fontaine, Thomas Neidhart, Leo Schou-Jensen, and Jan Steen Jensen: *DIADEM Environmental Management Requirements Document*, Issue 1.14.0 (2010).

Costin Badica, Sorin Ilie, Gregor Pavlin, Michiel Kamermans, and Mihnea Scafes: Using Negotiation for Dynamic Composition of Services in Multi-Organizational Environmental Management. *In: Proc.of International Symposium on Environmental Software Systems*, ISESS-2011, Springer (2011) (in press)

Ate Penders, Gregor Pavlin, and Michiel Kamermans: A Collaborative Approach to Construction of Complex Service Oriented Systems. *In: Intelligent Distributed Computing IV, Studies in Computational Intelligence* 315, Springer, 55-66 (2010)