User Assistant Agent: Towards Collaboration and Knowledge Sharing in Grid Workflow Applications

Michal Laclavík, Martin Šeleng, Ladislav Hluchý

Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9
845 07 Bratislava, Slovakia
laclavik.ui@savba.sk
http://ikt.ui.sav.sk

Abstract. This paper focuses on a User Assistant Agent for collaboration and knowledge sharing in Grid Workflow applications. The theory, implementation and use of such system - EMBET are described. The key idea is that a user enters notes in a particular situation/context, which can be detected by the computer. Such notes are later displayed to other or the same users in a similar situation/context. The context of a user is detected from computerized tasks performed by the user. Also intelligent components in the grid middleware such as monitoring, workflow analysis or workflow composition can provide context sensitive notes to be displayed for the user. The User Assistant Agent was created in scope of the K-Wf Grid project, in which grid services are semi-automatically composed to workflows dealing with a user problem. It was identified that even when services and input and output data are well semantically described, there is often no possibility to compose an appropriate workflow e.g. due to missing specific input data or fulfillment of a user and application specific requirements. To help a user in workflow construction, problem specification or knowledge reuse from past runs, it is appropriate to display notes and suggestions entered by users or intelligent middleware components. Thus experts can collaborate and fill up application specific knowledge base with useful knowledge, which is shown to users in the right time.

1 Introduction

The experience management [1] solutions are focused on knowledge sharing and collaboration among users in organizations. A lot of companies have recognized knowledge and experience as the most valuable assets in their organization [2]. Experience is something that is owned by people only, not obtainable by computer systems. Anyhow, according to the state of the art in the area we can a create experience management system, which will manage (not create) experience and will be able to capture, share and evaluate experience among users. We can understand experience through text notes entered by a user. Such form of experience is the most understandable for humans, but it can be grasped by a computer system, though only partially. A computer system needs to return experience in a relevant context. Thus we need to model the context of the environment and capture and store the context of each entered note. In this paper we describe such solution for the **e**xperience **m**anagement **b**as**e**d on **t**ext notes – EMBET entered by users.

The key idea is that a user enters notes in a particular situation/context, which can be detected by the computer. Such notes are later displayed to other or the same users in a similar situation/context. The context of a user can be detected from many sources – actions provided in grid environment (portal, submitted jobs, past runs), a step in a business process or a workflow management system, used files or detection of other events performed in the computer. The solution was used and evaluated in the Pellucid IST project[1] and it is further developed in the K-Wf Grid IST project[2].

The main objective of the User Assistant solution based on EMBET system[3] is to provide a simple and powerful collaboration and knowledge sharing platform based on experience management infrastructure, which can be applicable in many areas especially in grid virtual organizations. The idea is to return information, knowledge or experience to users when they need it. Therefore it is crucially important to model and capture a user context and the described solution can be used only in applications where actions/tasks performed by a user are computerized and can be captured and reported to the system in the form of events.

The EMBET system can be applied in a different application. In scope of this article we focus on its use to support grid applications particularly knowledge support for construction, execution and reuse of grid workflows and results. When constructing a workflow from grid services, a user needs to have knowledge about problems, services, models or input and output data. Such knowledge can be formalized only partially and workflows solving a user problem can be composed only semi-automatically with user help. Experience/notes entered by experts can help users to create the best workflow for their needs.

The article first discusses a theoretical approach of the EMBET solution and its architecture, followed by examples given for the Flood Forecasting grid application.

2 Theory of the Approach

The EMBET system is built on several theories and dealing with several theoretical challenges:

- Experience Management,
- Context Modeling and Context Detection,
- and Context Matching

2.1 Experience Management

According to Bergman [1], the experience management is simply the capability to collect lections from the past associated to cases. We have a person who has a prob-

[1] http://www.sadiel.es/Europa/pellucid/
[2] http://www.kwfgrid.eu/
[3] http://ikt.ui.sav.sk/?page=software_doc/embet.php

lem p, which is described in a certain space, called the Problem Space (P). In the experience Management system we have Case-Lesson pairs (c, l), where we have a Case Space (C) and a lesson space (L). To provide appropriate lesson learned we need to fulfill the following steps:

1. User context detection from the environment which describes problem P
2. Our Model is described by ontology and Notes are stored with associated context, which describes space C
3. Notes represent learned lesson L which is associated with space C (note context). The note context is matched with a user problem described by the detected user context. Context matching techniques are applied to find match between knowledge and user context. As a result all applicable notes are matched and returned.
4. The lesson is left to be applied by the user by reading appropriate notes.

More on applying this theory in EMBET can be found in [3] [4].

2.2 Context Modeling and Context Detection

For context and also knowledge modeling we use semantic web approach – ontologies and the CommonKADS [5] methodology. We are able to model and detect context when the application domain is well modeled using ontologies. Our model is an ontology model based on five main elements: Resources, Actions, Actors, Context and Events. Other ontology models can be attached easily to the model where concepts from foreign ontology are specified as context if not specified otherwise. The main idea is to model environment events which provide context of the user. For more details see [6][7].

In EMBET we need to detect user context from events transformed to the ontology model. For each application we need to specify an appropriate algorithm for user context updating based on user related events [4]

We also need to detect context of information, knowledge or experience entered by a user. In chapter 4 we describe a concrete example of such detection. Detected context is only suggested to the user in form of checklist and a user confirms final knowledge /note context. A checklist is created of user current context and context detected from text of notes using automatic semantic annotation techniques of a knowledge note text. This semantic annotation is described in detail in [8], [9].

2.3 Context Matching

The role of EMBET is to assist users in relevant knowledge/suggestions, which are applicable to their current situation. EMBET needs to return experience in context where experience is relevant. Thus we need to match context of knowledge and context of a user to return appropriate knowledge to the user. In K-Wf Grid we use a simple matching technique where ontology elements present in knowledge context have to be found in user context; otherwise knowledge notes are not displayed. However, this simple matching algorithm is not sufficient in some applications and we need to use technologies based on similarity matching [10][11]. Another problem concerning voting on knowledge notes occurs, if similarity mechanisms are used since a vote weight depends on a similarity value and need to be determined.

Currently we are developing and testing a new context matching algorithm based on intersection of user and knowledge context.

We define 3 sets of concepts in (**1**):

$$U = \{u_1, u_2, ..., u_n\}\ ,\ K = \{K_1, K_2, ..., K_m\}\ ,\ W_i = \{k_{i1}, k_{i2}, ..., k_{il}\} \quad (1)$$

Where U is a set of user context, K is a set of knowledge notes and W_i is a set of knowledge context for knowledge K_i.

A problem occurs in context matching, because there are four different ways to match user context and knowledge context (**2**).

$$U = W_i\ ,\ U \subset W_i\ ,\ U \supset W_i\ ,\ U \not\subset W_i \wedge W_i \not\subset U \wedge U \cap W_i \neq \emptyset \quad (2)$$

It is clear that the best option is the first one because there is exact match in contexts. The question is what about the other equations? We decide to define the relevance r_i (**3**) (**4**), which sorts the detected knowledge from best to worst.

$$r_i = \frac{|U \cap W_i|}{|U \cup W_i|} \quad (3)$$

In (**4**) we show different approach which can compute different relevance in some cases.

$$r_i = \frac{2|U \cap W_i|}{|U| + |W_i|} \quad (4)$$

For all knowledge where $r_i > 0$ we create sorted list where the most relevant knowledge is on the top.

Both approaches (**3**) (**4**) give equal relevance in the two cases described in (4):

$$\begin{aligned} &1.\ U = \{u_1, u_2\}\ W_i = \{k_{i1}, k_{i2}, k_{i3}\},\ u_1 = k_{i2} \\ &2.\ U = \{u_1, u_2, u_3\}\ W_i = \{k_{i1}, k_{i2}\},\ u_2 = k_{i1} \end{aligned} \quad (5)$$

This equality is our future work. In addition, we would like to deal with voting on displayed knowledge notes. We need to calculate different voting weights based on relevance of user and knowledge context.

In our future work we will also evaluate usage of both relevance approaches (**3**) (**4**).

3 Architecture and Technology

Architecture of EMBET (Figure 1 left side) consists of 3 main elements:

- Core of the system
- GUI
- System Memory

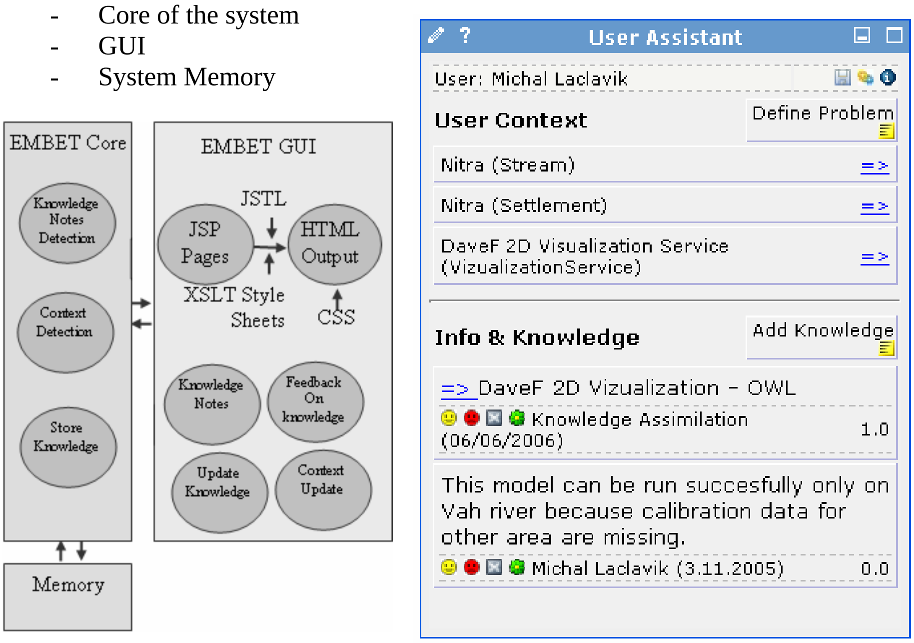

Fig. 1. EMBET Architecture and Graphical User Interface

EMBET Core provides the main functionality of EMBET. It determines a User context and searches for the best knowledge (in a form of text notes) in its Memory. The knowledge is subsequently sent through XML-RPC or SOAP to EMBET GUI.

EMBET GUI Graphical User Interface (Figure 1 right side) visualizes the knowledge and the user's context information to the user. Furthermore it informs the EMBET core about user context changes. The context can be reported also directly to the EMBET core from external systems (e.g. from workflow systems, received emails, or file system monitors). EMBET GUI visualizes knowledge based on XML[4] transformation to HTML through XSL[5] Templates processing. Moreover EMBET GUI has an infrastructure for a note submission and context visualization. It further provides a user with feedback (voting) on knowledge relevance. In addition it contains a user interface for knowledge management by experts where an expert can change a note and its context.

---

[4] http://www.w3.org/XML/

[5] http://www.w3.org/TR/xslt

EMBET Core - EMBET GUI interface is used for an XML data exchange between EMBET Core and EMBET GUI. The Interface is based on the XML-RPC[6] protocol for an XML message exchange.

Interface to Memory is used for information and knowledge extraction and storage. It is based on RDF/OWL data manipulation using Jena API, which EMBET Core uses to extract and store knowledge.

Experience is represented by text notes, an unstructured text, entered by a user. Ontology is stored and managed in the Web Ontology Language (OWL)[7]. The Jena Semantic Web Library[8] is used for knowledge manipulation and knowledge storing. The Java technology is used for developing the system and user Interface is based on the JSP technology. The XSL templates are used to transform XML generated from OWL to displayed HTML. Since the Java technology is chosen as background for the EMBET, a choice of the web server – Jakarta Tomcat[9] and implementation middleware is reasonable.

4 Example of Use

To better illustrate the use of EMBET in the process of user assistance, we present the following example from the K-Wf Grid project's flood forecasting application, which extends the flood prediction application of the CROSSGRID (IST-2001-32243) [12] project. The application's main core consists of simulation models series for meteorological, hydrological and hydraulic predictions. The models are organized in a cascade, with each step of the cascade being able to employ more than one model. For example, the first step - the meteorological weather prediction - can be computed using the ALADIN model, or the MM5 model.

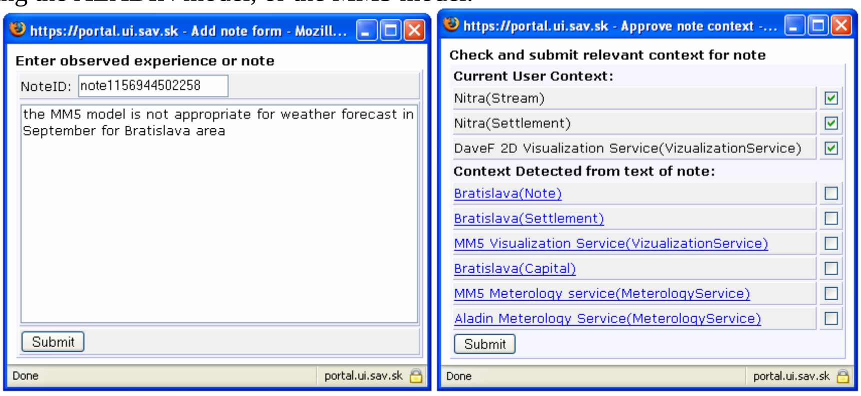

Fig. 2. Left: Entering new Note; Left: Note Context Detection, checked items are current user context, unchecked items are elements detected from text of the note. User selects only those items which are relevant.

---

[6] http://www.xmlrpc.com/
[7] http://www.w3.org/TR/owlfeatures/
[8] http://www.sf.net/
[9] http://jakarta.apache.org/tomcat/

Consider that the user has used the MM5 meteorology model and he/she wants to describe its properties (gathered knowledge), which may be relevant for other users. The proposed model of interaction is as follows:

- A user enters a note through UAA, stating that "the MM5 model is not appropriate for weather forecast in September for the Bratislava area" (Figure 2).
- From the workflow in which the user states this note, we know directly the current user context (checked items on Figure 2)
- Some of current context can be relevant to note and some does not have to be. The note is processed and its text related to the context, as well as the relevant context items are found in the ontology memory (GOM) (Figure 2). In this case, by finding the text MM5 we can assume that "MM5 Meteorology Service" is the relevant part of the context. There is other context relevant information which can be detected like "September", the time in which this note is valid.
- After the context detection, the user is provided with a checklist (Figure 2) where the user may select only the relevant parts of the note context.
- A user selects parts of the context, which were detected by the system as really relevant. He/she can modify the contents of the list and finally submit the note.
- Each time anyone works with the MM5 service for the Bratislava area in September, the note is displayed or it can be also displayed in similar contexts
- Each note can be evaluated by a user as being "good" or "bad" and the current results are always displayed along with the vote.

This model gives a good basis for experience management and knowledge sharing in a virtual organization as well as for application-related collaboration among users.

5 Conclusion and Future Work

Our solution was evaluated on the K-Wf Grid IST project, focused on building grid-based workflows, where users need to collaborate and share knowledge about different applications, computational models, grid resources or data. Previously it was evaluated on a selected administration application in the Pellucid IST project, where the context or the problem of a user was detected in the Workflow Management Application. Such solution may be applied in many further areas where the user problem can be detected from computerized tasks. Usually this occurs in any business process where actions are performed via a computer, e.g. workflow processes, document management, supply chain management or dynamic business processes where email communication is in use.

People like to enter notes or memos. This is a natural way of notifications for themselves or others to remind them of problems, activities or solutions. Therefore we think that such solution can be successfully applied and commercialized.

In K-Wf Grid we help users in workflow construction and execution by displaying knowledge notes and suggestions entered by the same or different users. Thus experts can collaborate and fill up application specific knowledge base with useful knowledge which is shown to users in the right time. In addition, intelligent components create and provide knowledge notes with attached context to the EMBET systems concerning past workflows runs or computed results. Thus users can reuse old workflows or results of execution, which are displayed in proper user context by EMBET system.

Acknowledgments

This work is partially supported by projects K-Wf Grid EU RTD IST FP6-511385, NAZOU SPVV 1025/2004, RAPORT APVT-51-024604, SEMCO-WS APVV-0391-06, VEGA No. 2/6103/6, VEGA 2/7098/27.

References

1. Ralph Bergmann: Experience Management: Foundations, Development Methodology, and Internet-Based Applications (Lecture Notes in Artificial Intelligence), 2002
2. Thomas H. Davenport, Laurence Prusak, Working Knowledge, ISBN:1578513014, 2000
3. Laclavik, M., Gatial, E., Balogh, Z., Habala, O., Nguyen, G.,Hluchy, L.: Experience Management Based on Text Notes (EMBET); Innovation and the Knowledge Economy, Volume 2, Part 1: Issues, Applications, Case Studies; Edited by Paul Cunnigham and Miriam Cunnigham; IOS Press, pp.261-268. ISSN 1574-1230, ISBN 1-58603-563-0.
4. Michal Laclavik: Experience Management based on Ontology and Text Notes (The EMBET System); Work submitted for the RNDr. degree; Ustav Informatiky, Prirodovedecka fakulta, Univerzita Pavla Jozefa Safarika v Kosiciach, 2005
5. August Schreiber et al.: Knowledge Engineering and Management: the Common-KADS methodology, ISBN: 0262193000, 2000
6. Laclavik, M., Babik, M., Balogh, Z., Hluchy, L.: AgentOWL: Semantic Knowledge Model and Agent Architecture; In Computing and Informatics. Vol. 25, no. 5 (2006), p. 419-437. ISSN 1335-9150
7. Michal Laclavik, Marian Babik, Zoltan Balogh, Emil Gatial, Ladislav Hluchy: Semantic Knowledge Model and Architecture for Agents in Discrete Environments; In: Frontiers in Artificial Intelligence and Applications, Vol 141, 2006. Proc. of ECAI 2006 Conference, G.Brewka et al.(Eds.), IOS Press, pp.727-728. ISBN 1-58603-642-4 ISSN 0922-6389 2006
8. Laclavik, M., Seleng, M., Gatial, E., Balogh, Z., Hluchy, L.: Ontology based Text Annotation – OnTeA; Information Modelling and Knowledge Bases XVIII. IOS Press, Amsterdam, Marie Duzi, Hannu Jaakkola, Hannu Kangassalo, Yasushi Kiyoki (Eds.), 2007
9. Michal Laclavik, Martin Seleng, Marian Babik: OnTeA: Semi-automatic Ontology based Text Annotation Method; In: Tools for Acquisition, Organisation and Presenting of Information and Knowledge. P.Navrat et al. (Eds.), Vydavatelstvo STU, Bratislava, 2006, pp.49-63, ISBN 80-227-2468-8.
10. Michal Laclavik: Ontology and Agent based Approach for Knowledge Management; PhD Thesis submitted for the degree philosophiae doctor; Institute of Informatics, Slovak Academy of Sciences, field: Applied Informatics, 2005
11. Balogh Z., Budinska I.: OntoSim - Ontology-based Similarity Determination of Concepts and Instances. In: Tools for Acquisition, Organisation and Presenting of Information and Knowledge. P.Navrat et al. (Eds.), Bratislava, 2006, pp.64-70, ISBN 80-227-2468-8.
12. Hluchy L., et al.: Flood Forecasting in CrossGrid project. In: Grid Computing, 2nd European Across Grids Conference, Nicosia, Cyprus, January 28-30, 2004, LNCS 3165, Springer-Verlag, 2004, pp. 51-60, ISSN 0302-9743, ISBN 3-540-22888-8