



USER ASSISTANT AGENT USER MANUAL

WP5

Document Filename:	KWF-WP5-UAA-IISAS-v1.0-SoftwareUserManual
Work package:	WP5
Partner(s):	II SAS
Lead Partner:	II SAS
Document classification:	PUBLIC

Abstract: This document provides a user manual for User Assistant Agent (UAA) K-WfGrid component.

Delivery Slip

	Name	Partner	Date	Signature
From	Michal Laclavik	IISAS	28/09/2006	
Verified by	Piotr Nowakowski	CYFRONET	11/10/2006	
Approved by	Steffen Unger	FIRST	11/10/2006	

Document Log

Version	Date	Summary of changes	Author
0.1	29-Nov-05	Draft version	Michal Laclavik
0.2	22-Dec-05	Updated version including recommendation from internal review	Michal Laclavik
0.3	30-Aug-06	Final software user manual	Michal Laclavik
0.4	03-Oct-06	Final software user manual update	Michal Laclavik
1.0	11-Oct-06	QA check	Piotr Nowakowski

CONTENTS

COPYRIGHT NOTICE	4
1. INTRODUCTION.....	5
1.1. ABBREVIATIONS AND ACRONYMS	5
1.2. REFERENCES AND SOURCE CODE	5
2. SOFTWARE USAGE.....	6
2.1. RUNNING THE SOFTWARE	6
2.1.1. <i>Operating Requirements</i>	6
2.1.1.1. Local hardware requirements	6
2.1.1.2. Local software requirements	6
2.1.1.3. Grid infrastructure requirements	6
2.1.2. <i>Step-by-Step User Setup</i>	6
2.2. BASIC OPERATION	7
2.3. ADVANCED FEATURES.....	7
2.4. KNOWN PROBLEMS.....	7
3. INTERFACE REFERENCE GUIDE.....	8
3.1.1. <i>Note Adding Scenario</i>	8
3.1.2. <i>Problem/context Definition Scenario</i>	10
3.1.3. <i>Context Matching and Notes Display Scenario</i>	11
3.1.4. <i>Initializing Red Workflow Scenario</i>	12
3.1.5. <i>Initializing of Past Workflows</i>	12
3.1.6. <i>Displaying the Results of Workflow Execution</i>	12
4. TROUBLESHOOTING Q&A	13
5. CONTACT INFORMATION AND CREDITS.....	14
6. THE LICENSE AGREEMENT.....	15

COPYRIGHT NOTICE

Copyright (c) 2005-2006 by Institute of Informatics, Slovak Academy of Sciences & K-Wf Grid consortium. All rights reserved.

Use of this product is subject to the terms and licenses stated in the GPL license agreement. Please refer to Section 5 for details.

This research is partly funded by the European Commission IST-2002-511385 Project “K-WfGrid”.

1. INTRODUCTION

The main role of User Assistant Agent (UAA) is to assist user with relevant knowledge/suggestions, which are applicable on current user situation. In UAA, we understand experience through notes entered by user. Such notes are displayed to the users in same or similar context. Thus UAA helps users in collaboration and knowledge sharing. Notes can be generated also automatically (e.g. past workflow and result notes generated by KAA-WXA). Notes can also contain reference to other resources such as workflows, results or images.

The other UAA role is problem definition where user can describe his/her problem typing free text and relevant semantic description of problem is detected and processed by other K-Wf Grid components to build and execute workflows for given user problem.

UAA is built on EMBET architecture developed in Institute of Informatics, SAS. This architecture is being extended during K-Wf Grid project and sometimes when we refer to EMBET or UAA this can be identical.

1.1. ABBREVIATIONS AND ACRONYMS

EMBET – Platform for Collaborating and Knowledge Sharing via User Assistance. Abbreviations comes from “**E**xperience **M**anagement **b**ased on **T**ext **N**otes”

UAA – User Assistant Agent sometimes referenced as EMBET

GOM – Grid Organizational Memory sometimes referenced as Memory

GWES – Grid Workflow Execution Service

KAA – Knowledge Assimilation Agent; See KAA manuals for more details

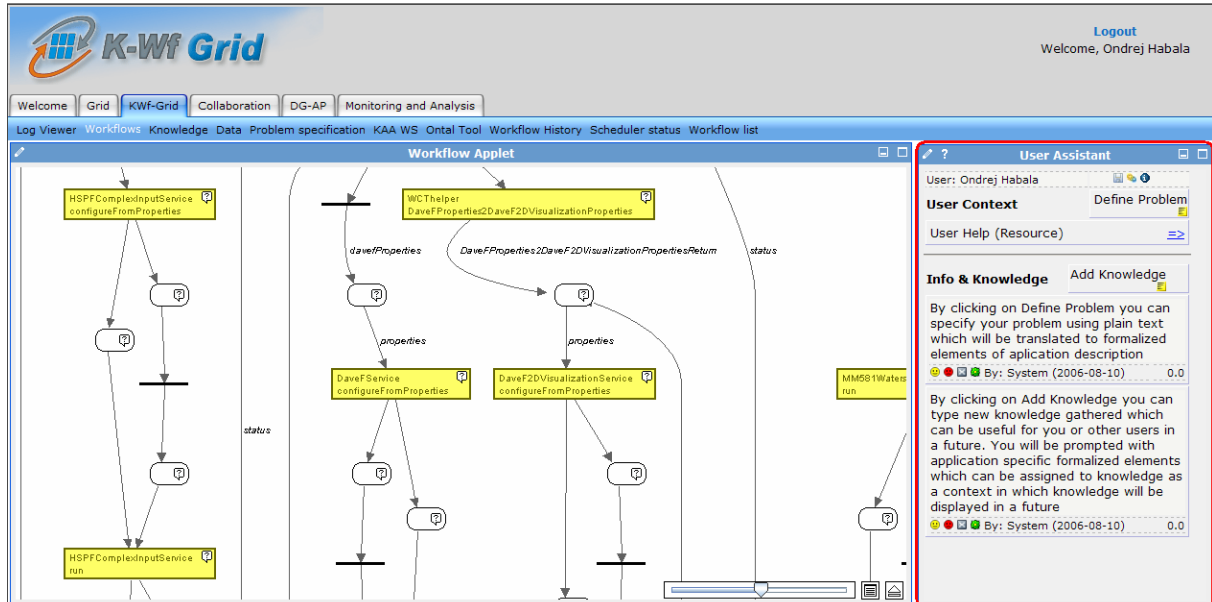
KAA-WXA – Workflow XML Analyzer is part of KAA which process past workflows to find knowledge resulting to Result and Workflow Notes.

1.2. REFERENCES AND SOURCE CODE

The full source of EMBET and EMBET GUI presented in this document are available on K-Wf Grid’s central CVS repository. The JavaDoc documentation can be build out of source code using ant build file.

2. SOFTWARE USAGE

Software usage is described in this chapter as well as in chapter 3 where GUI screenshots with examples are provided. For more information on technical details please see developer manual.



On the screenshot K-Wf Grid portal GUI with User Assistant Agent (UAA) portlet (marked in red square) is visible. Scenarios and screenshots in chapter 3 show only UAA part of K-Wf Grid portal GUI.

2.1. RUNNING THE SOFTWARE

2.1.1. Operating Requirements

UAA is connected with several K-Wf Grid components. To exploit all UAA features K-Wf Grid portal, Grid organization Memory (GOM) and Grid Workflow Execution Service (GWES) need to be in operation. It can be run on same machine as other components or on single computer. UAA can be set up also as independent EMBET component in other applications

2.1.1.1. Local hardware requirements

No special hardware requirements are defined. UAA is created as regular Java application.

2.1.1.2. Local software requirements

EMBET/UAA software needs several Java libraries such as log4j or Jena Semantic Web Toolkit libraries. UAA Core is build as standalone Java application. UAA GUI is build as Servlet application called from UAA portlet which is part of K-Wf Grid portal. UAA Core requires Java 1.4 or higher and Ant. UAA GUI requires Tomcat or other Servlet container. If UAA Core and UAA GUI are on different machines, port for its communication needs to be open. By default UAA Core use port 8000 for XML-RPC procedure calling.

2.1.1.3. Grid infrastructure requirements

UAA Core imports K-Wf Grid Ontologies from GOM.

2.1.2. Step-by-Step User Setup

UAA/EMBET core can be executed using ant build script “ant start” and UAA/EMBET GUI can be installed to tomcat using ant install in “webapps” directory of the distribution.

2.2. BASIC OPERATION

In this chapter we provide list of basic features. To understand this features in practice see chapter 3 for more details on use case scenarios.

- To display the basic help of the UAA/EMBET click on ⓘ icon. This help can be extended by users using Add note in help context
- To define your problem for workflow composition or to return relevant knowledge in form of notes, workflows or results you need to describe your problem by free text. Click on “Define problem 📝” to type your text description.
- To add new knowledge in form of notes click on “Add Knowledge 📝” to type your note.
- By clicking on close 🗑 icon under displayed note you can close the note to be not displayed for you in the future.
- By clicking on face icon you can vote good 😊 or bad 😡 on note relevance. This will have influence on note order
- By clicking on 🗨 icon you can post message for the note author to start discussion and collaboration.
- By clicking on ➡ you can do an action on an associated resource (context or note) like to start workflow composition, submit and reuse past workflows or view past results of workflow execution. See scenarios for more details.

2.3. ADVANCED FEATURES

In this chapter we provide list of advanced features. To understand this features in practice see chapter 3 for more details on use case scenarios.

- By clicking on save 💾 you can save current memory knowledge to Ontology Memory / GOM. In case of crash of the system or restart the knowledge would be lost if not saved. Saving is not done automatically due to use of Jena memory model because Jena MySQL backend it is too slow for real and fast interaction.
- By clicking on 🧠 icon you can execute KAA-WXA tool to start analyzing past workflows. The result of analysis is new automatically generated knowledge notes related to past workflows and results of workflow execution.

2.4. KNOWN PROBLEMS

- Reimplementation of Annotation algorithm is needed because now it is case sensitive which match only some results.
- Reimplementation of Context matching algorithm is needed since similarity of context is not supported. This will have influence on voting and note/knowledge ordering mechanism.

3. INTERFACE REFERENCE GUIDE

To better illustrate the use of UAA 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 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.

3.1.1. Note Adding Scenario

This scenario describes how Note - experience representation is born in UAA system.

On Figure 1 Note adding window can be seen. When such note is confirmed UAA annotate Note and detect ontology elements showed in Figure 2.



Figure1: Entering new Note UAA Window

On Figure 2 results of context detection can be seen. Checked items are current user context, unchecked items are elements detected from text of the note. A user selects only relevant items and submits the note context. Please note that ontology items detected from text are clickable. By clicking on any item, UAA/EMBET window will show relevant help Notes on the item. This will help user to understand what kind of objects were detected and what they represents.



Figure 2: Note Context Detection

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:

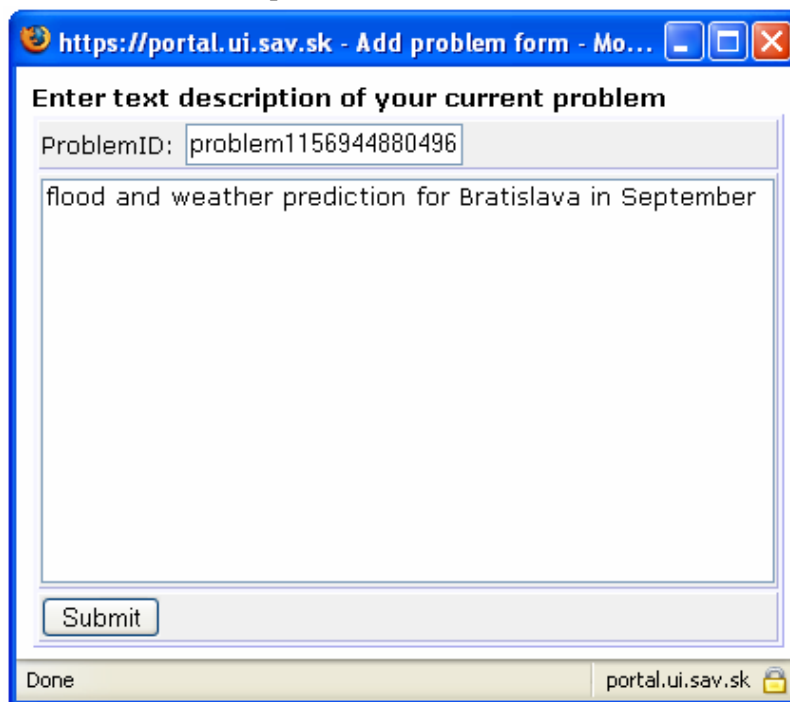
- A user enters a note through UAA, stating that “the MM5 model is not appropriate for weather forecast in September for Bratislava area” (Figure 1).
- From the workflow in which the user states this note, we know directly the current user context (checked items on Figure 2)
- Some of current context 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 presented with a checklist (Figure 2) where the user may select only the relevant parts of the context, which will trigger this note.
- A user selects parts of the context, which were detected by the system as really relevant. He/she can subsequently modify the contents of the list and finally submit the note.
- Each time anyone works with the MM5 service for Bratislava area in September, the note is displayed or it can be also displayed in similar contexts
- Each note can be voted by a user as being “good” or “bad” and the current results are always displayed along with the vote. This feature is not implemented yet.

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.

3.1.2. Problem/context Definition Scenario

The main idea of this scenario is that user context (environment/ problem definition) is changed and thus relevant experience can be showed after such change. In K-Wf grid application this can be archived by:

- reporting a change of user context from an external system. For example when a user selects a different workflow or web service in K-Wf Grid GUI event for updating user context is generated.
- Free text explanation of a user problem (Figure 3). Such free text is then annotated by Anotate class and detected elements are presented to the user to confirm detected user context.



The screenshot shows a web browser window with the address bar displaying 'https://portal.ui.sav.sk - Add problem form - Mo...'. The main content area has a title 'Enter text description of your current problem'. Below the title is a 'ProblemID:' label followed by a text input field containing 'problem1156944880496'. Underneath is a large text area containing the text 'flood and weather prediction for Bratislava in September'. At the bottom of the form is a 'Submit' button. The browser's status bar at the bottom shows 'Done' and 'portal.ui.sav.sk' with a lock icon.

Figure 3: Problem Definition UAA Window

Free text problem definition is important, when a user starts to work with the system and wants to define problem to be understandable for computer System. UAA then can return relevant experience for such problem.

Free text problem definition is important also because such problem definition discovers appropriate web services and data for construction of Workflow of Services.

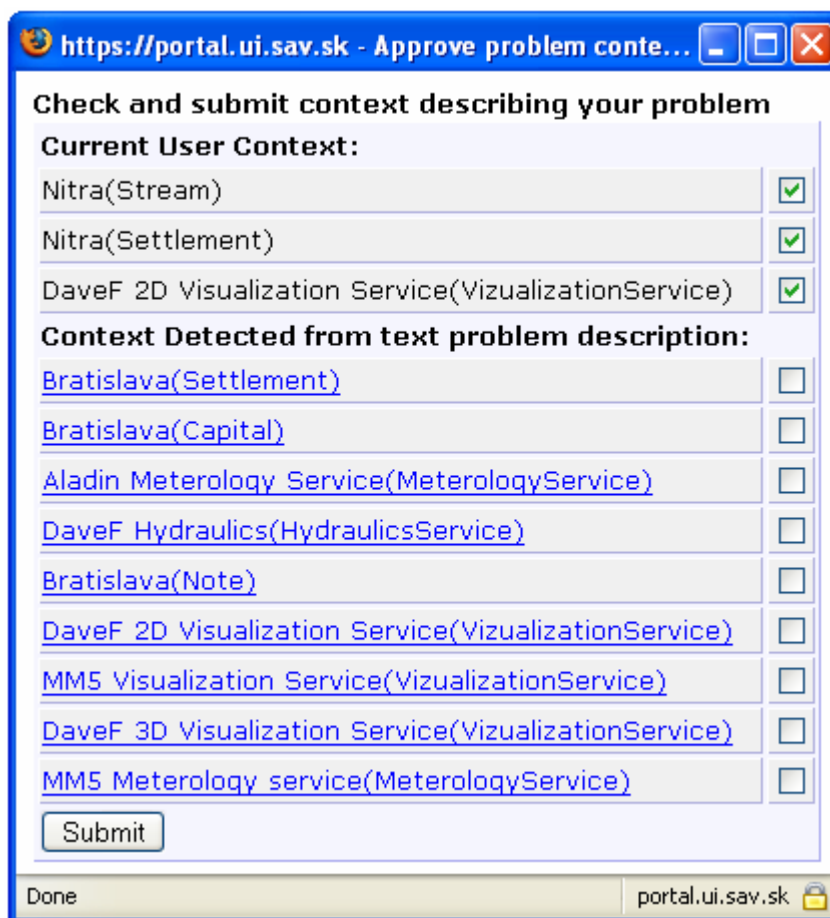


Figure 4: Problem Context Detection

A workflow is constructed and executed by other K-Wf Grid components but those components use detected ontology elements (web services: MM5 service, Aladin, DaveF and output data definitions: Bratislava Location, September) to compose an appropriate workflow.

When a user selects relevant problem context and submits it, even a user context change is generated and UAA executes context updating algorithm. Similar as in Note adding scenario user can click on ontology elements in context list to get help notes in UAA GUI.

3.1.3. Context Matching and Notes Display Scenario

When context of an actor/user is changed as showed in previous scenario, note updating algorithm is executed and relevant notes are assigned to the user model. The Notes are then displayed to the user as can be seen on Figure 5. Some of the notes have clickable “=>” sign. This means some resource is attached to the note. This can be document URI, a result file of a workflow execution or past workflow which can be initiated again by clicking on the icon.

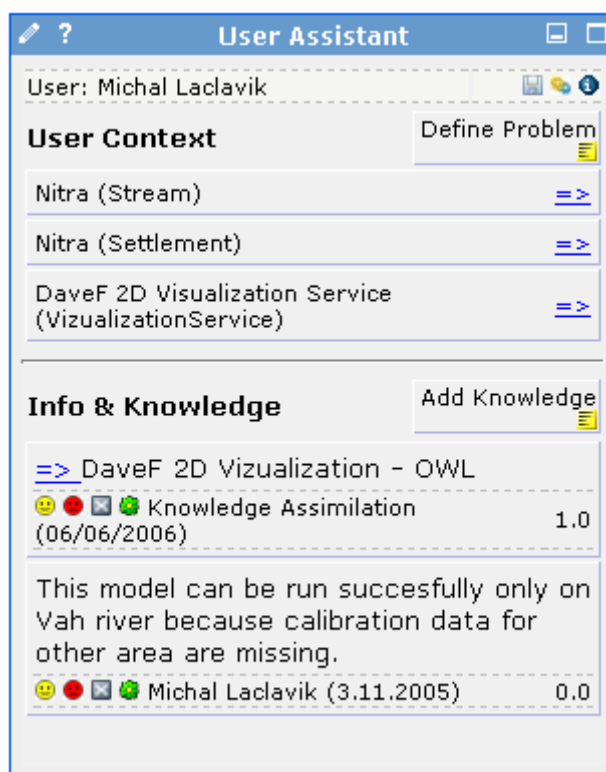


Figure 5: UAA/EMBET GUI

3.1.4. Initializing Red Workflow Scenario

UAA K-Wf Grid specific functionality is initializing so called “Red Workflow”. Red Workflow is simple workflow, which consist of element specifying desired user output or desired service effect. Such elements are detected using problem definition scenario. By clicking on “=>” selected user context is used to build red workflow which is sent to GWES.

3.1.5. Initializing of Past Workflows


UAA K-Wf Grid specific functionality is to reuse past composed or executed workflows. If for a user context some workflow notes are displayed, user can click on “=>” note icon to resubmit such workflow to GWES. As example please see first note “Dave 2D Visualization – OWL” on Figure 5. Workflows notes are automatically generated by KAA-WXA tool. See more details on KAA-WXA tool in KAA manuals.

3.1.6. Displaying the Results of Workflow Execution

UAA K-Wf Grid specific functionality is also to view result files of past executed workflows. If for a user context some result notes are displayed, user can click on “=>” note icon to open result file in new browser window. Result notes are automatically generated by KAA-WXA tool. See more details on KAA-WXA tool in KAA manuals.

4. TROUBLESHOOTING Q&A

Q: How do I show general Help?

A: By clicking on  icon you can display general help.

Q: How I install UAA/EMBET GUI?

A: It is part of K-Wf Grid portal – one of portlets. You need also to install Servlet application under /embet path in tomcat.

Q: When I log to portal UAA seems to be not working.

A: It can be that you are not in ontology as a user. Please send email to: laclavik.ui@savba.sk

5. CONTACT INFORMATION AND CREDITS

Michal Laclavik

- Author of EMBET architecture
- Core implementation
- Memory implementation & integration with GOM
- 2nd version of GUI implementation
- Ontea - Semantic based annotation

Email: laclavik.ui@savba.sk; Web: <http://laclavik.net/>

Emil Gatial

- Author of First GUI implementation
- Reloading of GUI using AJAX technologies
- First implementation of annotation algorithm
- consultations

Email: emil.gatial@savba.sk

Zoltan Balogh

- GUI related and other consultations

Martin Seleng

- Semantic Annotation (Ontea) success rate measuring and evaluation.

Martin Maliska, Branislav Simo

- Java programming and related consultations

Ondrej Habala, Ladislav Hluchy

- General consultations
- Thanks for pushing to work on UAA ☺

Bartek Kryza, Marian Babik, Jacek Kitowski

- Ontology, Knowledge description, semantic web consultations
- GOM integration

K- Wf Grid & Pellucid Project Consortium

- Ideas
- Evaluations
- Consultations

6. THE LICENSE AGREEMENT

GPL License

See <http://www.gnu.org/licenses/gpl.txt> for more details.