UBIWARE Deliverable D2.2:

Industrial Cases: 2nd Status Report

March, 2009

Date	March 4, 2009
Document type	Report
Dissemination Level	UBIWARE project consortium
Contact Author	Vagan Terziyan
Co-Authors	Artem Katasonov, Oleksiy Khriyenko, Sergiy Nikitin, Michal Nagy, Mikko Vapa
Work component	WP7
Deliverable Code	D2.2
Deliverable Owner	IOG, JYU
Deliverable Status	Mandatory, Internal
Intellectual Property Rights	Unaffected

Table of Contents

UBIWARE Deliverable D2.2:
Workpackage WP7:

Introduction

The objective of this workpackage is to trial UBIWARE on real industrial cases. This has two major goals for such case studies. The first goal is to evaluate the scientific concepts behind UBIWARE and to find problems and issues in UBIWARE that would otherwise be overlooked. The second goal is to facilitate the further utilization of UBIWARE in the industry. Several specific cases, proposed by the industrial partners, are analyzed, designed and prototyped based on the UBIWARE platform. The reasons for prototyping are the same: to identify issues in UBIWARE that would get overlooked if the work was only theoretical and thus abstract, and to demonstrate the benefits of UBIWARE in a tangible way so to facilitate future industrial adoption.

There are four industrial cases, those of Fingrid, Inno-W, Metso Automation and Nokia.

During the Year 2, with respect to all four cases the task is the following:

Task T2.1_w7: Developing a full prototype application: connecting to additional relevant resources and extending the interactions between them towards a sufficiently elaborated application.

UBIWARE Deliverable D2.2:
Workpackage WP7:

1 Fingrid case

1.1 Background

With respect to the UBIWARE approach and platform, Fingrid's main area of interest is in organizing smart data management related to the events/alarms which company gets from their control systems. Existing systems do not provide many possibilities for managing this data beyond storing it to a time-series log, and browsing it with some filtering possibilities. A wish is to that the data should get flexibly accessible, integrated with other related data, and possibilities should be provided for producing generalizing reports to the power system operation and asset management persons.

Fingrid has the following two databases that were so far the objects of interest in the UBIWARE's Fingrid case:

- **Event History database**: **Eventlog** (Oracle) in the office environment, to which data is automatically replicated from SCADA's event history database. A record in this database contains such information as the time of the event, event class, access area, substation ID, device ID, the state of the object, and some other.
- **Elnet database** (Oracle) that stores information about all the equipment, including circuit-breakers, disconnectors, transformers, capacitors, and other. A record in this database contains such information as device group, device ID, ownership (Fingrid or external), and other.

The unique device ID present in both databases enables join queries.

1.2 Special issues

Data security is a central concern in the case. It is major reason, along with *safety*, for putting the focus on historic analysis of events rather on the real-time operation. In result, the UBIWARE project team did not have a direct access to Fingrid databases. Rather, a surrogate Oracle database had to be created and populated with data provided by Fingrid. This database has two tables imitating the schemes of Eventlog and Elnet, and was sufficient for the application development purposes.

Later, the prototype was connected to the real databases in Fingrid office environment. Some inconsistencies between the real and the surrogate databases appeared, but were successfully resolved.

1.3 Fingrid industrial prototype

The architecture of the Fingrid prototype is sketched in Figure 1.1. There are three agents in the prototype.

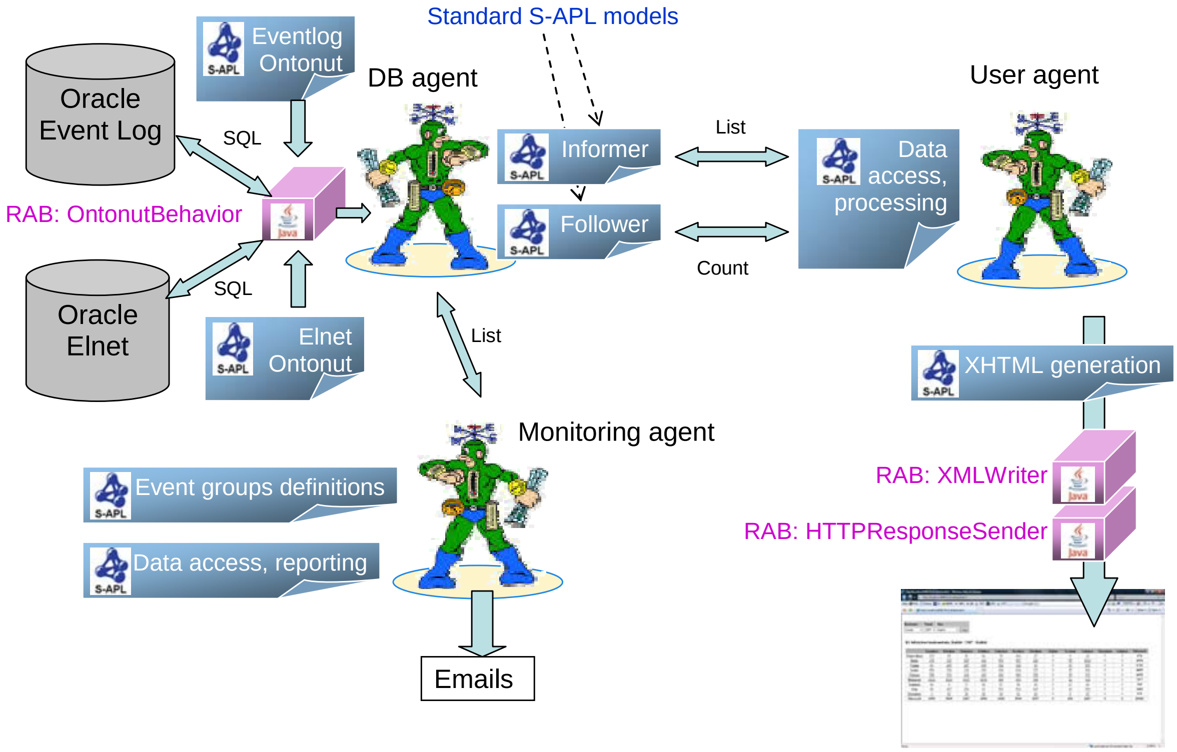


Figure 1.1 – Architecture of Fingrid prototype.

DB agent is responsible for interfacing with the databases. Implementation of this agent is based on UBIWARE's **ontonuts** approach. With this approach:

- DB agent receives from other agents queries that are formulated semantically and encoded using S-APL. The data is sent back to the requestors also in a semantic S-APL form.
- The databases (relational, non-semantic) are provided each with an *ontonut*, which is a description of the database schema that is sufficient for translating between S-APL

semantic queries and SQL as well as between database responses and a needed semantic form.

- Reusable java component OntonutBehavior takes care of generating SQL and translating responses. If S-APL query concerns both databases, OntonutBehavior two generates SQL sub-queries and cross-joins the results.

It is notable that given that the interface of DB agent towards other agents is provided by standard S-APL models *Follower* and *Informer*, DB agent does not have any single line of code (either Java or S-APL) that would be written just for it and is non-reusable. The only tailored elements are ontonuts, which are declarations, not behavioral code.

User agent is responsible for providing XHTML interface to a human user. User agent receives user queries, interacts with DB agent, and presents the data to the user.

Monitoring agent is an autonomously operating agent which is responsible for continuously checking the new events appearing in the Eventlog database and sending email notifications (see function 3 below).

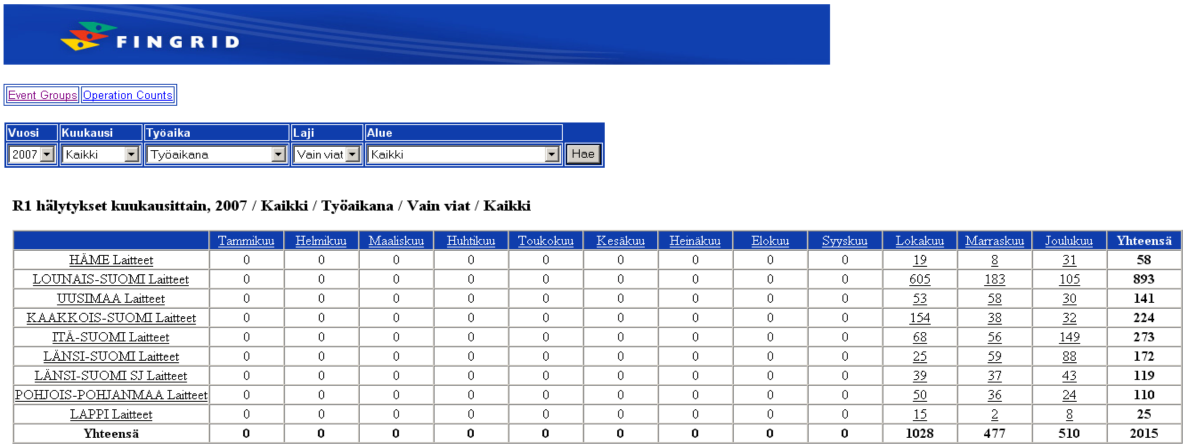
From the operational point of view, the present prototype implement the three functions that are described below.

Function 1. *Equipment alarms*

This function allows a user to either count the number of R1 category events or to retrieve all such events from the Eventlog. R1 category events are equipment alarms and identified by the Access Area attribute, which is one of the following: 00000008, 00000010, 00000020, 00000080, 00000100, 00001000, 00002000, 00020000 and 00040000. The 9 values above represent 9 defined geographic regions of Finland. The application counts only the alarms themselves and filters out the events related to the device returning to the normal state or operator’s acknowledgement of the alarm. The application also provides other restriction / filtering possibilities, it enables the user to count or retrieve the alarms:

- For a specific year or month.
- For a specific access area, i.e. geographic region.
- All alarms, or only those falling inside the official office hours (Monday-Friday, 8-16), or only falling outside office hours.
- All alarms, or only those corresponding to some physical faults which are identified by the device state being one of the following: ‘Vika’, ‘Hälyttää’, ‘Laukaisi’ or ‘Lauennut’.

Screenshots of XHTML reports of this function are shown in the two figures below. Figure 1.2 shows the table view, while Figure 1.3 shows the list view.

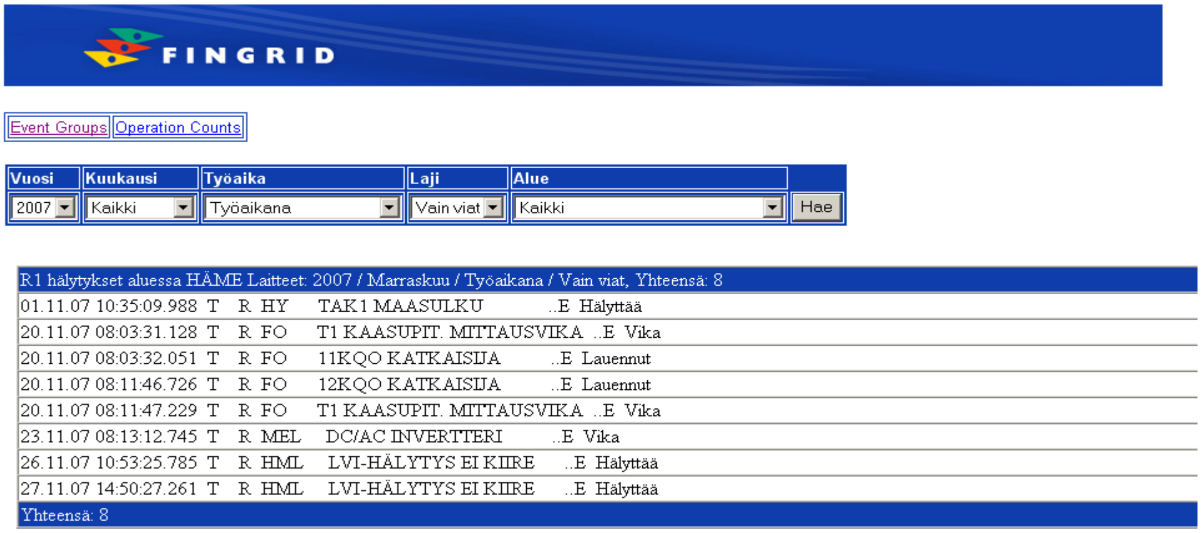


Vuosi	Kuukausi	Työaika	Laji	Alue
2007	Kaikki	Työaikana	Vain viat	Kaikki

R1 hälytykset kuukausittain, 2007 / Kaikki / Työaikana / Vain viat / Kaikki

	Tammikuu	Helmikuu	Maaliskuu	Huhtikuu	Toukokuu	Kesäkuu	Heinäkuu	Elokuu	Syyskuu	Lokakuu	Marraskuu	Joulukuu	Yhteensä
HÄME Laitteet	0	0	0	0	0	0	0	0	0	19	8	31	**58**
LOUNAIS-SUOMI Laitteet	0	0	0	0	0	0	0	0	0	605	183	105	**893**
UUSIMAA Laitteet	0	0	0	0	0	0	0	0	0	53	58	30	**141**
KAAKKOIS-SUOMI Laitteet	0	0	0	0	0	0	0	0	0	154	38	32	**224**
ITÄ-SUOMI Laitteet	0	0	0	0	0	0	0	0	0	68	56	149	**273**
LÄNSI-SUOMI Laitteet	0	0	0	0	0	0	0	0	0	25	59	88	**172**
LÄNSI-SUOMI SJ Laitteet	0	0	0	0	0	0	0	0	0	39	37	43	**119**
POHJOIS-POHJANMAA Laitteet	0	0	0	0	0	0	0	0	0	50	36	24	**110**
LAPPI Laitteet	0	0	0	0	0	0	0	0	0	15	2	8	**25**
Yhteensä	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1028**	**477**	**510**	**2015**

Figure 1.2 – Equipment alarms table view



Vuosi	Kuukausi	Työaika	Laji	Alue
2007	Kaikki	Työaikana	Vain viat	Kaikki

```
R1 hälytykset aluessa HÄME Laitteet: 2007 / Marraskuu / Työaikana / Vain viat, Yhteensä: 8
01.11.07 10:35:09.988  T   R  HY     TAK1 MAASULKU          ..E  Hälyttää
20.11.07 08:03:31.128  T   R  FO     T1 KAASUPIT. MITTAUSVIKA ..E  Vika
20.11.07 08:03:32.051  T   R  FO     11KQO KATKAISIJA        ..E  Lauennut
20.11.07 08:11:46.726  T   R  FO     12KQO KATKAISIJA        ..E  Lauennut
20.11.07 08:11:47.229  T   R  FO     T1 KAASUPIT. MITTAUSVIKA ..E  Vika
23.11.07 08:13:12.745  T   R  MEL    DC/AC INVERTTERI        ..E  Vika
26.11.07 10:53:25.785  T   R  HML    LVI-HÄLYTYS EI KIIRE     ..E  Hälyttää
27.11.07 14:50:27.261  T   R  HML    LVI-HÄLYTYS EI KIIRE     ..E  Hälyttää
Yhteensä: 8
```

Figure 1.3 – Equipment alarms list view

Function 2. *Operation counts*

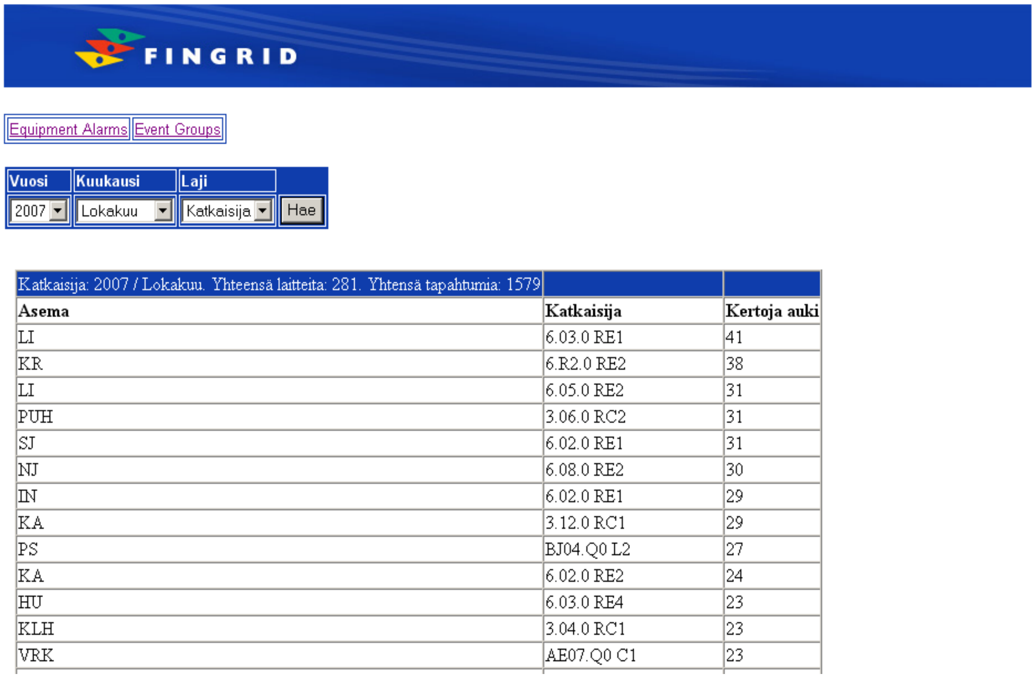
The number of operations is counted for all circuit-breakers and disconnectors owned by Fingrid. One operation is one open - close cycle, so an operation is counted from by “Auki” (open) events.

The ownership information is present only in the Elnet database, therefore the integration is required. The type of the device (circuit-breaker, disconnector) is also identified from Elnet, which has a field DEV_GROUP, which holds “CBR” for circuit-breakers and “DCN” for disconnectors. An additional source of this information (used to reduce the number of Eventlog responses) is the event priority field form the Eventlog database. This field has “9” for the states of disconnectors, “11” for states of the compensation equipment, which includes circuit breakers in reactor and capacitor bays, and “14” for the states of circuit-breakers except those in reactor and capacitor bays.

The results of the counting are represented as a list of devices ordered by decreasing number of operations. The filtering is enabled for:

- Device group.
- Event times (year, month).

Screenshots of XHTML report of this function is shown in Figure 1.4.



Vuosi	Kuukausi	Laji
2007	Lokakuu	Katkaisija

Katkaisija: 2007 / Lokakuu. Yhteensä laitteita: 281. Yhtensä tapahtumia: 1579		
Asema	**Katkaisija**	**Kertoja auki**
LI	6.03.0 RE1	41
KR	6.R2.0 RE2	38
LI	6.05.0 RE2	31
PUH	3.06.0 RC2	31
SJ	6.02.0 RE1	31
NJ	6.08.0 RE2	30
IN	6.02.0 RE1	29
KA	3.12.0 RC1	29
PS	BJ04.Q0 L2	27
KA	6.02.0 RE2	24
HU	6.03.0 RE4	23
KLH	3.04.0 RC1	23
VRK	AE07.Q0 C1	23

Figure 1.4 – Operation counts for circuit-breakers

Function 3. *Event groups*

This function adds pro-activity to the application in the sense that the application includes an agent that is autonomously:

- Checks for new events in the Event History database.
- Identifies if any of new events fall under the scope of some defined human "job responsibilities".
- Notify by e-mail the persons in charge about the event(s).

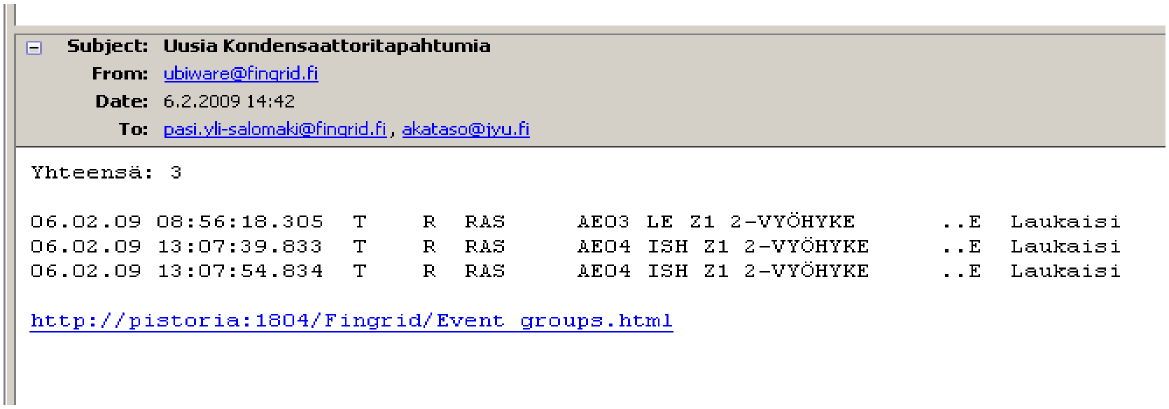
The implemented responsibilities are with respect to the events of three groups: related to circuit-breakers, transformers, and capacitors, respectively. Alarms of interest are faults only, i.e. have as the device state one of the following: 'Vika', 'Hälyttää', 'Laukaisi' or 'Lauennut'. An event group is identified by analyzing the device ID.

Notification of an alarm is done by email including STATIONNAME, POINTNAME, and ALARMTEXT. The interval for checking new events is configurable. All new events after the previous transmission are sent in the same e-mail message.

A responsible person is also able to use the Web browser to access the list of all the events from his “job responsibility”. For that, the existing interface is extended with corresponding queries. The filtering is enabled for:

- Group (Circuit-breaker, Transformer, Capacitor)
- Event time (year, month)

An example email is shown in Figure 1.5. Screenshots of XHTML report of this function is presented in Figure 1.6.



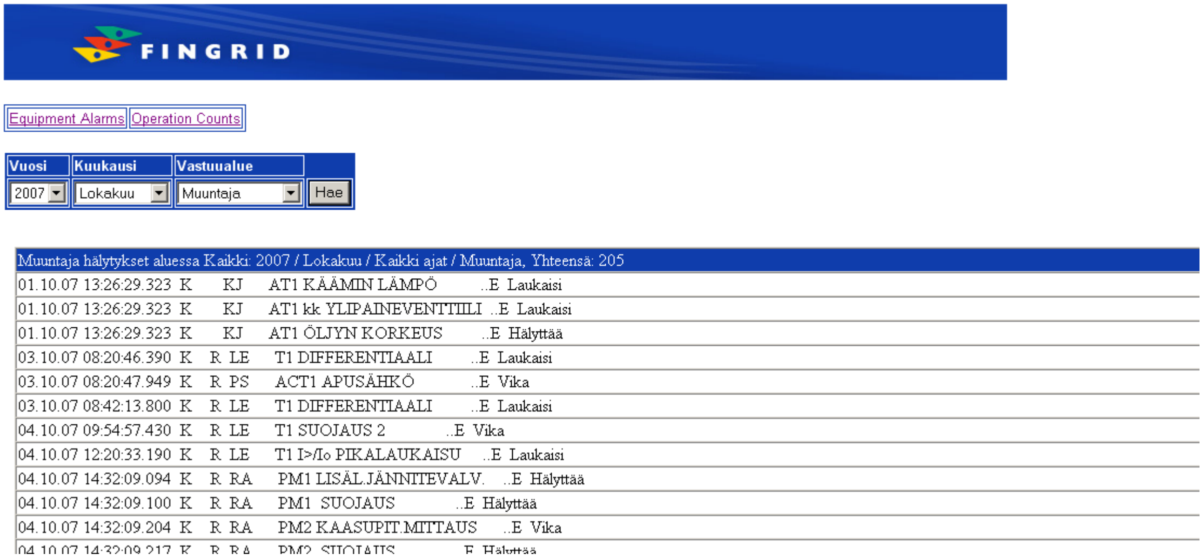
Subject: Uusia Kondensaattoritapahtumia
From: ubiware@fingrid.fi
Date: 6.2.2009 14:42
To: pasi.yli-salomaki@fingrid.fi, akataso@jyu.fi

```
Yhteensä: 3

06.02.09 08:56:18.305  T    R  RAS     AE03 LE Z1 2-VYÖHYKE      ..E  Laukaisi
06.02.09 13:07:39.833  T    R  RAS     AE04 ISH Z1 2-VYÖHYKE     ..E  Laukaisi
06.02.09 13:07:54.834  T    R  RAS     AE04 ISH Z1 2-VYÖHYKE     ..E  Laukaisi

http://pistoria:1804/Fingrid/Event_groups.html
```

Figure 1.5 – Notification email



Vuosi	Kuukausi	Vastuualue
2007	Lokakuu	Muuntaja

```
Muuntaja hälytykset aluessa Kaikki: 2007 / Lokakuu / Kaikki ajat / Muuntaja, Yhteensä: 205
01.10.07 13:26:29.323  K     KJ    AT1 KÄÄMIN LÄMPÖ          ..E  Laukaisi
01.10.07 13:26:29.323  K     KJ    AT1 kk YLIPAINEVENTTIILI  ..E  Laukaisi
01.10.07 13:26:29.323  K     KJ    AT1 ÖLJYN KORKEUS         ..E  Hälyttää
03.10.07 08:20:46.390  K   R LE    T1 DIFFERENTIAALI        ..E  Laukaisi
03.10.07 08:20:47.949  K   R PS    ACT1 APUSÄHKÖ            ..E  Vika
03.10.07 08:42:13.800  K   R LE    T1 DIFFERENTIAALI        ..E  Laukaisi
04.10.07 09:54:57.430  K   R LE    T1 SUOJAUS 2           ..E  Vika
04.10.07 12:20:33.190  K   R LE    T1 I>/Io PIKALAUKAISU     ..E  Laukaisi
04.10.07 14:32:09.094  K   R RA    PM1 LISÄL.JÄNNITEVALV.     ..E  Hälyttää
04.10.07 14:32:09.100  K   R RA    PM1  SUOJAUS             ..E  Hälyttää
04.10.07 14:32:09.204  K   R RA    PM2 KAASUPIT.MITTAUS      ..E  Vika
04.10.07 14:32:09.217  K   R RA    PM2  SUOJAUS              .E  Hälyttää
```

Figure 1.6 – Transformer events

1.4 Future opportunities

The future development directions for the Fingrid industrial prototype are currently under discussion. The opportunities that were considered before included:

- Extension of the statistic analysis of the Event History data
  - Analysis of efficiency of maintenance service providers. In case of an R1 alarm in their working area, the provider is notified automatically. One question is how much time it takes the provider to reach the substation to perform maintenance.
  - Filtering out (as an option) R1 events while "kuluvalvonta" is off, i.e. a maintenance work is underway.

- Further integration of data from Event History and Elnet
  - Analysis of the relationship between alarms (Event history) and the types of equipment (Elnet).

- Integration with Tosu database that is used by the maintenance service providers to report to Fingrid the costs for the work performed.
  - Understanding what alarm has led to what maintenance actions at what cost.

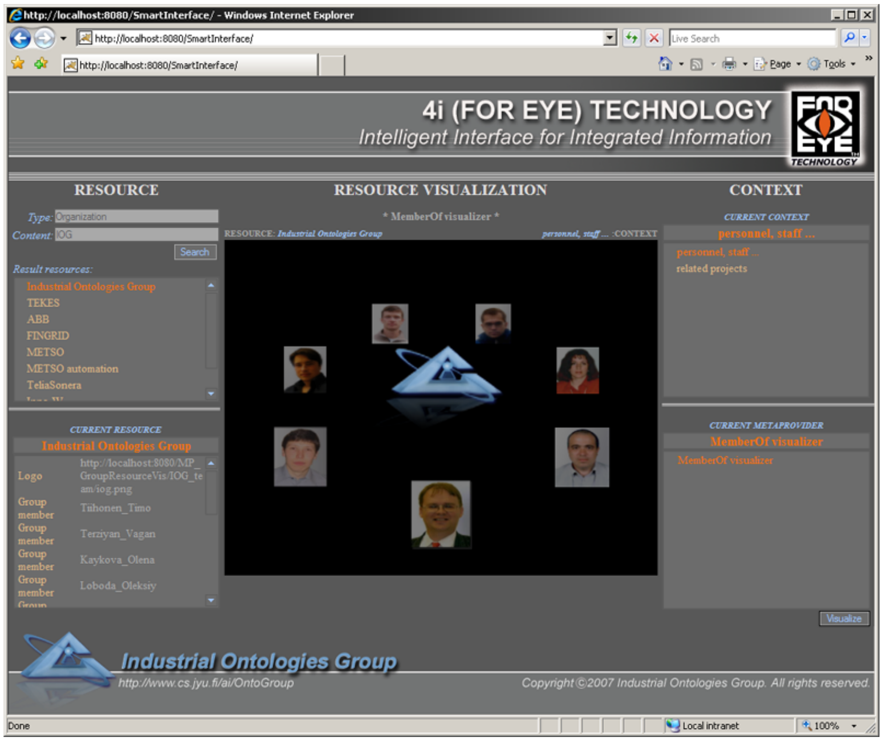
- Integrating data from Event History and data from Lightening notification service.
  - Matching the locations and times of R3,4,5 alarms with the locations and times of lightning strikes to automatically filter out lightning-caused disturbances (normally require no action to be performed).

UBIWARE Deliverable D2.2:
Workpackage WP7:

2 Inno-W case

2.1 Background

According to the plan, this part of the Deliverable 2.2 is aimed to present prototype development of Inno-W industrial case. The goal is to build an Idea Browser that provides functionality to discover similar ideas/proposals/projects in user-defined context, to calculate and visualize similarity/closeness of ideas with GUI tool. Such a case requirement has the best match with workpackage WP-5 and realized based on 4I (FOR EYE) Browser (see Figure 2.1).



http://localhost:8080/SmartInterface/ - Windows Internet Explorer

Type: Organization
Content: IOG

Result resources:
- Industrial Ontologies Group
- TEKES
- ABB
- FINGRID
- METSO
- METSO automation
- TeliaSonera

CURRENT RESOURCE: Industrial Ontologies Group

Logo	http://localhost:8080/MP_GroupResourceVis/IOG_team/iog.png
Group member	Tiihonen_Timo
Group member	Terziyan_Vagan
Group member	Kaykova_Olena
Group member	Loboda_Oleksiy

RESOURCE: *Industrial Ontologies Group*   *personnel, staff ...* :CONTEXT

CURRENT CONTEXT: personnel, staff ...
- personnel, staff ...
- related projects

CURRENT METAPROVIDER: MemberOf visualizer
- MemberOf visualizer

Copyright©2007 Industrial Ontologies Group. All rights reserved.

Figure 2.1 - current version of 4i(FOR EYE) browser.

2.2 Special requirements

Resource (ideas/proposals/projects) descriptions from Inno-W's Databases should be adapted/transformed to the RDF format.

2.3 Inno-W industrial prototype

2.3.1 Architecture of the system:

Idea Browser is based on general 4I Browser architecture. The main common Interface part – 4I GUI Shell, performs communication with resource repository and repository of visualization contests. Shell provides resource search functionality, presents resources properties to the user, provides selection of resource visualization contexts and visualization modules (MetaProviders). At the same time, Shell provides all necessary data for MetaProviders.

MetaProvider performs visualization function: depending on realization collects all necessary data and visualizes resource/resources in context dependent way. Considering the MetaProvider that presents resources in a context of their closeness to the selected one, certain distance measuring calculation is performed before visualization phase.

4I GUI Shell uses xml-based resource storage. Such architecture requires converting the date from original format to xml representation. In following example you can see Idea description with five main property types that are used during a distance calculation:

```
<resource>
    <resId>…</resId>
    <resClass>…</resClass>
    <name>…</name>
    <resTypeDes>
       <rtdItem>…</rtdItem>
       …
    </resTypeDes>
    <resContDes>
       <rcdItem>…</rcdItem>
       …
    </resContDes>
    <properties>
       <property>
          <prop_id>…</prop_id>
          <prop_name>…</prop_name>
          <prop_type>textField</prop_type>
          <prop_value>…</prop_value>
       </property>
       <property>
          <prop_id>…</prop_id>
          <prop_name>…</prop_name>
          <prop_type>keyWordsField</prop_type>
          <prop_values>
             <subValue>…</subValue>
             … (amount of sub values is not limited)
          </prop_values>
       </property>
       <property>
          <prop_id>…</prop_id>
          <prop_name>…</prop_name>
          <prop_type>complexTextField</prop_type>
          <prop_values>
             <subValue>(value from defined list of values in field
                       context)</subValue>
             … (amount of sub values is defined in field context)
          </prop_values>
       </property>
       <property>
          <prop_id>…</prop_id>
          <prop_name>…</prop_name>
          <prop_type>numberField</prop_type>
          <prop_value>…</prop_value>
       </property>
       <property>
          <prop_id>…</prop_id>
          <prop_name>…</prop_name>
          <prop_type>intervalField</prop_type>
          <prop_values>
             <subValue>…</subValue>
             <subValue>…</subValue>
          </prop_values>
       </property>
       …
    </properties>
  </resource>
```

For the moment, general adapter that enable to convert data from any format to the required one is not exists. That is why we consider such convertor as an external part of the system. Further, general adaptation module can be elaborated and imbedded to the Shell.

In this prototype we are concentrated on resource closeness visualization. Such visualization context implies user specification of the resource properties significance and existence of additional contextual information for the resources properties (depending on their types). In current prototype such contextual information is stored in separate xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<closenessContexts>
 <closenessContext>
  <closenessContext_id>…</closenessContext_id>
  <closenessContext_name>…</closenessContext_name>
  <calculation_method>…</calculation_method>
  <fieldContext>
   …
  </fieldContext>
  <fieldContext>
   …
  </fieldContext>
  …
 </closenessContext>
```

```
</closenessContexts>
```

Current implementation does not support visual creation and modification of visualization context yet, but it is planed to be added to the functionality of the 4I GUI Shell in the future.

Comparison between the resources is performed based on common properties. Current implementation supports just five types of the parameters (properties):

Text field types:

Type 1: Just a pure word/sentence. Additional contextual information for this field is its significance.

```
<fieldContext>
  <field_type>textField</field_type>
  <field_significance>…</field_significance>
  <field_calculation_method>…</field_calculation_method>
</fieldContext>
```

Type 2: Text field is presented by list of key words/sentences. Additional contextual information for this field is its significance.

```
<fieldContext>
 <field_type>keyWordsField</field_type>
 <field_significance>…</field_significance>
 <field_calculation_method>…</field_calculation_method>
</fieldContext>
```

Type 3: Text field is divided to the set of attributes and presented by correspondent list of values (words/sentences) of the attributes. In this case, the number of the attributes for certain text field should be defined and lists of possible (defined) values of the attributes should be defined and presented. In another words, it is defined amount of keywords, where each keyword is selected from a correspondent defined set of values. Additional contextual information for this field is the sets of values for each attribute (keyword) and the significance of the attributes, and as for all fields, significance of the field itself.

```
<fieldContext>
 <field_type>complexTextField</field_type>
 <field_significance>…</field_significance>
 <field_calculation_method>…</field_calculation_method>
 <corClasses>
  <corClass>
   <class_significance>…</class_significance>
   <value>…</value>
   …
  </corClass>
  <corClass>
   <class_significance>…</class_significance>
   <value>…</value>
   …
  </corClass>
  …
 </corClasses>
```

```
</fieldContext>
```

Number field: Just number that further will be normalized and compared. Additional contextual information for this field is its significance.

```
<fieldContext>
 <field_type>numberField</field_type>
 <field_significance>…</field_significance>
 <field_calculation_method>…</field_calculation_method>
</fieldContext>
```

Interval field: Field presented by start and end point on a numerical axis. Distance measuring function for such interval field is based on a distance between the centers of the intervals and the lengths of them. Additional contextual information for this field is the significance of these two main parameters, and as for all fields, significance of the field itself.

```
<fieldContext>
 <field_type>intervalField</field_type>
 <field_significance>…</field_significance>
 <field_calculation_method>…</field_calculation_method>
 <subField_significances>
  <value>…</value>
  <value>…</value>
 </subField_significances>
</fieldContext>
```

2.3.2 Distance Measuring

Resources (in our particular case - Ideas) compared based on selected set of properties that belongs to specified five field types. General distance (closeness) between to resources based on a list of attributes (properties) is a value from 0 to 1, calculated as a weighed value of all the distances of separate attributes.

$$D(X,Y) = \sqrt{\sum_{\forall i, x_i \in X, y_i \in Y} \omega_i \cdot d(x_i, y_i)^2}, \qquad (2.1)$$

where $d(x_i, y_i)$ is a distance by certain attribute and ω_i is weight for attributes. The requirement for the weights is:

$$\sum_{i=1}^{n} \omega_i = 1, \qquad (2.2)$$

where n is a number of attributes.

Now we have a problem, we have distances between the resources based on certain attributes separately. The centers of masses those sets are not balanced and differently influence on result. We have to modify the function with a correction part that balances centers of masses.

$$D(X,Y)=\sqrt{\sum_{\forall i,x_i\in X,y_i\in Y}\omega_i\cdot\left(0.5+\frac{0.5\cdot\left(d(x_i,y_i)-d_i'\right)}{\max\limits_j\left|d(x_i,y_i)_j-d_i'\right|}\right)^2}, \qquad (2.3)$$

where d_i' is *median* or *arithmetic average* of corresponding samples of distances by i -th attribute.

In case of a ***Text field (type 1)*** we calculates distance based on full matching of string values, and distance takes the values 0 or 1.

In case of a ***Text field (type 2)*** the distance between two objects based on such a field can be calculated based on following formula:

$$D(X,Y)=\frac{N'}{N}, \qquad (2.4)$$

where N' is a number of matched/equal instances and N is a general number of all instances in the lists of two comparable objects.

In case of a ***Text field (type 3)*** we have decided to use well-known PEBLS distance evaluation for nominal values. Let’s consider a text field that extended/enriched by set of sub-fields/attributes with values defined on a finite set of possible values. The distance d_j^l between two values v_1 and v_2 for j attribute in a context of attribute l is:

$$d_j^{l\,2}(v_1,v_2)=\sum_{i=1}^{k}\left(\frac{C_{1i}}{C_1}-\frac{C_{2i}}{C_2}\right)^2, \qquad (2.5)$$

where C_1 and C_2 are the numbers of instances in the training set with selected values v_1 and v_2, C_{1i} and C_{2i} are the numbers of instances from the i-th class, where the values v_1 and v_2 were selected, and k is the number of the classes of instances (from values of attribute l).

Thus, general distance d_j between two values v_1 and v_2 for j attribute is:

$$d_j(v_1,v_2)=\frac{\sum\limits_{l=1,\,l\neq j}^{n}d_j^l(v_1,v_2)}{n-1}, \qquad (2.6)$$

where n is a number of attributes (sub-fields).

Finally, distance between two objects based on such complex field can be calculated based on following formula:

$$D(V_1,V_2)=\sqrt{\sum_{\forall i,v_1^i\in V_1,v_2^i\in V_2}\omega_i\cdot d_i(v_1^i,v_2^i)^2}, \qquad (2.7)$$

where $d_i(v_1^i, v_2^i)$ is a distance by certain attribute (sub-field) and ω_i is weight for attributes. The requirement for the weights is:

$$\sum_{i=1}^{n} \omega_i = 1, \tag{2.8}$$

where n is a number of attributes.

If there is no differences in significance of the attributes, then $\omega_i = 1/n$ and

$$D(V_1, V_2) = \sqrt{\frac{\sum\limits_{\forall i, v_1^i \in V_1, v_2^i \in V_2} d_i(v_1^i, v_2^i)^2}{n}}. \tag{2.9}$$

If some attributes (sub-fields) are not specified, it means that there is no information in the current text field that concerns those attributes. The distances between values of such attributes are - “0”.

In case of a ***Number field***, distance measuring is bases on normalization all of the values from whole sample of them. The formula of a distance between two values v_i and v_k is:

$$d(v_i, v_k) = \left| \frac{v_i - v_k}{v_{\max} - v_{\min}} \right|, \tag{2.10}$$

where $v_{\max}$ and $v_{\min}$ are the maximum and minimum values from the sample.

In case of an ***Interval field***, we have decided to focus on main aspects of time periods comparison: durations of the periods and distance between the intervals. And the simplest formula that implicitly take into account these parameters (see Figure 2.2) is:

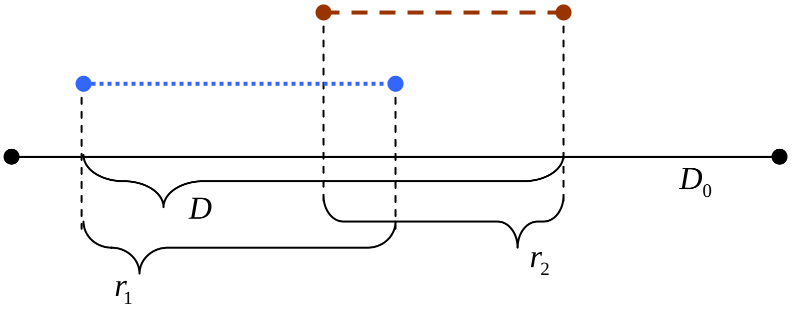


Figure 2.2 – The intervals comparison (Type1).

$$d([a_i, b_i], [a_j, b_j]) = \frac{D - r}{D_0}, \tag{2.11}$$

where

$$r = \frac{r_1 + r_2}{2} = \frac{(b_1 - a_1) + (b_2 - a_2)}{2},$$

$$D = \max(b_i, b_j) - \min(a_i, a_j),$$

$$D_0 = \max_{\forall p}(b_p) - \min_{\forall q}(a_q).$$

If we are going to explicitly control the influence of interval attributes and to tune their significances, then we can use another more complex formula. For such intervals distance measurement we are going to use formula that takes into account the distance between the intervals' centers and difference between the lengths of the intervals (see Figure 2.3):

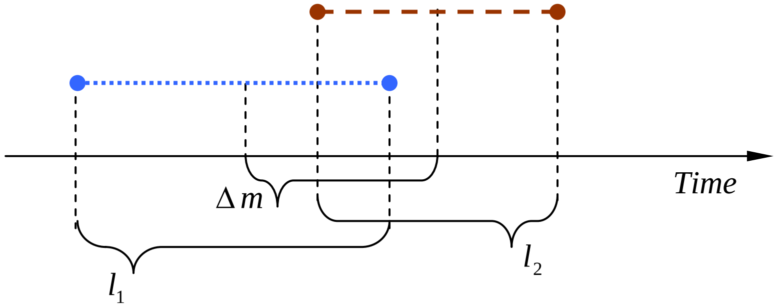


Figure 2.3 – The intervals comparison (Type2).

$$D(t_i, t_k) = \sqrt{k_m \cdot \left(\frac{|m_i - m_k|}{M} \right)^2 + k_l \cdot \left(\frac{|l_i - l_k|}{l_{\max}} \right)^2}, \qquad (2.12)$$

$$D(t_i, t_k) = \sqrt{k_m \cdot \left(\frac{\Delta m}{M} \right)^2 + k_l \cdot \left(\frac{\Delta l}{l_{\max}} \right)^2}, \qquad (2.13)$$

where m_i and m_k are the values of the intervals' centers on a time line, l_i and l_k are the durations/lengths of the intervals, M and $l_{\max}$ are maximum distance between the centers of two intervals and maximum duration/length of interval from a sample set. The coefficients k_m and k_l regulate significance of the distance between the intervals and difference between the lengths of the intervals. The condition for those coefficients is:

$$k_m + k_l = 1, \qquad (2.14)$$

But still, depending on sample, $\frac{\Delta m}{M}$ and $\frac{\Delta l}{l_{\max}}$ can differently influence on a result, even if the coefficients will be equal. We have to modify the function with a correction part that balances centers of masses.

$$D(t_i, t_k) = \sqrt{k_m \cdot \left(0.5 + \frac{0.5 \cdot \left(\frac{\Delta m}{M} - m'\right)}{\max\limits_j \left|\frac{\Delta m}{M}_j - m'\right|}\right)^2 + k_l \cdot \left(0.5 + \frac{0.5 \cdot \left(\frac{\Delta l}{l_{\max}} - l'\right)}{\max\limits_r \left|\frac{\Delta l}{l_{\max}}_r - l'\right|}\right)^2}\,, \qquad (2.15)$$

where m' and l' are medians or *arithmetic averages* of corresponding samples.

Other possible distance measuring methods:

To find the distance between two terms, it is also possible to utilize a dissimilarity measure, called Normalized Google Distance (NGD), introduced in (Cilibrasi, 2007). NGD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. The concepts are represented with their labels which are fed to the Google search engine as search terms. Given two search terms x and y, the normalized Google distance between x and y, $NGD(x, y)$, can be obtained as follows

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}}\,, \qquad (2.16)$$

where $f(x)$ is the number of Google hits for the search term x, $f(y)$ is the number of Google hits for the search term y, $f(x, y)$ is the number of Google hits for the topple of search terms x y and M is the number of web pages indexed by Google[1].

Intuitively, $NGD(x, y)$ is a measure for the symmetric conditional probability of co-occurrence of the terms x and y: given a web-page containing one of the terms x or y, $NGD(x, y)$ measures the probability of that web-page also containing the other term.

If we are dealing with nominal attributes that have values defined on an infinite set of values, it is also reasonable to utilize existing remote services that measure Semantic Relatedness of the strings. One of those services is Measures of Semantic Relatedness2 (MSRs). MSRs are computational means for calculating the association strength between terms. More specifically, MSRs take the form of computer programs that can extract relatedness between any two terms based on large text corpora. MSRs have been used to produce models of human web-browsing behavior (Pirolli, 2005), augmented search engine technology (Dumais, 2003), semantic relevancy maps (Veksler and Gray, 2007), essay-grading

[1] Currently, the Google search engine indexes approximately ten billion pages (M~10^{10}).

[2] Measures of Semantic Relatedness (MSRs) - http://cwl-projects.cogsci.rpi.edu/msr/msr-about.html

algorithms for ETS (Landauer et al., 1998), and could be useful for any cognitive models or AI agents that have to deal with text.

Lack of Standardization in MSR Services: Although there are multiple MSR services that are readily available to the research community, these services are (1) scattered and (2) inconsistently formatted. All of the available MSR web servers use different input/output standards, making it less than ideal for researchers that may want to compare, contrast, alter, and average these measures. Some MSRs are available to download, but these technologies are even more diverse in protocol, and are much harder to use. To make matters worse, many MSRs are not publicly accessible, and of the available MSRs, very few parameter sets (e.g. different corpora, different sensitivity parameters) are offered. For example, ICAN (Lemaire and Denhiére, 2004) is a well-founded MSR that one may implement, but no public ICAN service exists. PMI is a popular and easy-to- implement measure, but you would be hard-pressed to find a PMI service based on a news corpus, or an email corpus, etc. The MSR Web Server is an ongoing effort to gather various MSRs and corpora, to make these publicly available, and to give researchers easy standardized access to semantic relatedness scores from all MSR-corpus pairs.

We have not concentrated our efforts on development of measuring functions depending on remote services. It always brings unreliability to the system. But still, these methods can be utilized as well.

2.3.3 Visualization component

Talking about visualization techniques, we have to consider usability issues that bring user-friendly information representation. What is the closeness of the resources? It is a value from 0 to 1. The easiest way to show the distances between resources is to present them on a line. But in case, when we visualize resources by their visual representatives (images / resource logos), we have to take into account the sizes of the images and such line-based representation becomes not so convenient any more. It is also possible to show the compared resources on a circle or sphere with different radiuses (distances to the correspondent resource). Again, in this case, it is quite difficult to see the difference between the distances of two resources to the initial one, especially if they are located on the opposite sites from the center (initial resource).

Taking into account all this nuances, we decided to put the resources on a spiral that lies on a surface of the cone (see Figure 2.6). The minimal distance between the resources has been taken as a step on an axis/height of the cone. Just that parameter (distance on the axis/height) shows the closeness of the resources (see Figure 5.4). To avoid an overlap (in case of a viewpoint from the top of the cone) of the images that belong to resources located next to each other, we have calculated the location angle (α) on each (step-based) cone cut (see Figure 5.4). Additionally, we provided a possibility to rotate the cone to find the best view point (see Figure 5.5).

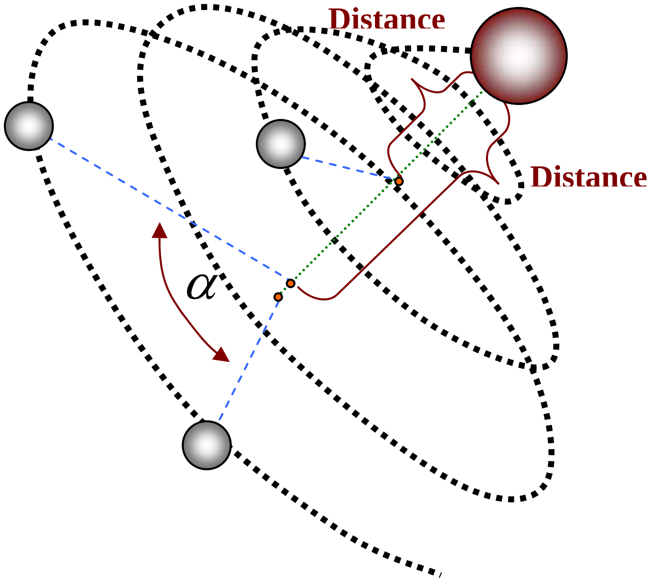


Figure 2.4 – spiral-based distance representation.

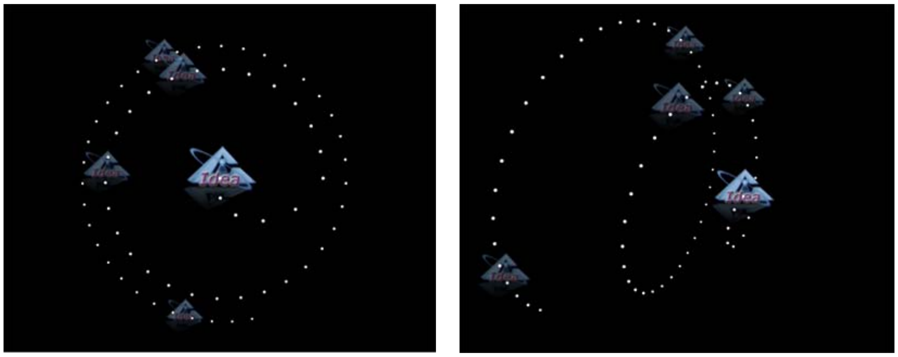
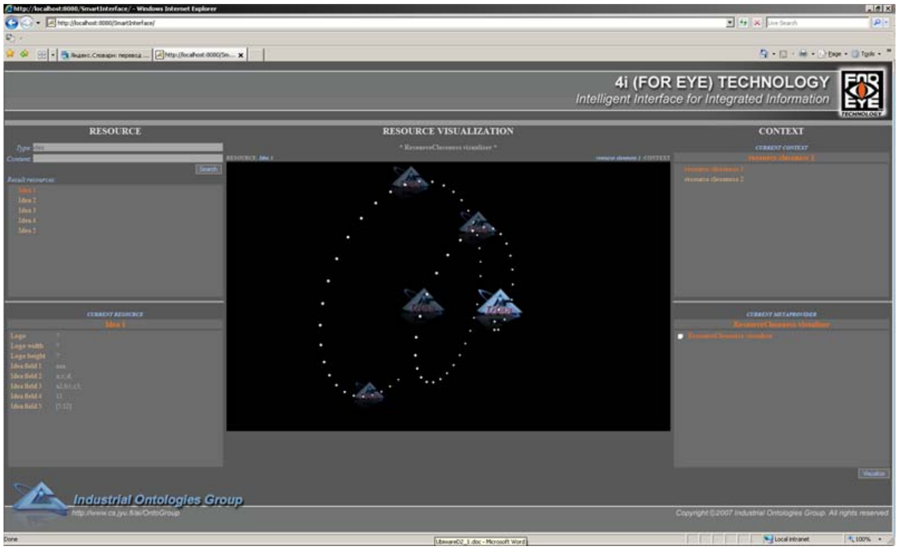


Figure 2.5 – spiral-based resource closeness visualization.


http://localhost:8080/SmartInterface/ - Windows Internet Explorer
http://localhost:8080/SmartInterface/
Live Search
Page
Tools
4i (FOR EYE) TECHNOLOGY
Intelligent Interface for Integrated Information
FOR EYE TECHNOLOGY
RESOURCE
RESOURCE VISUALIZATION
CONTEXT
Search
Visualize
Industrial Ontologies Group
http://www.cs.jyu.fi/ai/OntoGroup
Copyright ©2007 Industrial Ontologies Group. All rights reserved
Done
Local intranet
100%

Type: Idea
Content:
Result resources:
Idea 1
Idea 2
Idea 3
Idea 4
Idea 5

CURRENT RESOURCE
Idea 1

Logo	?
Logo width	?
Logo height	?
Idea field 1	aaa
Idea field 2	a, c, d,
Idea field 3	a2, b1, c3,
Idea field 4	11
Idea field 5	[5.12]

* ResourceCloseness visualizer *
RESOURCE: Idea 1
resource closeness 1 : CONTEXT

CURRENT CONTEXT
resource closeness 1
resource closeness 1
resource closeness 2

CURRENT METAPROVIDER
ResourceCloseness visualizer
ResourceCloseness visualizer

Figure 2.6 – resource closeness visualization in 4i (FOR EYE) Browser.

2.4 Future opportunities

As was mentioned before, general adapter that enable to convert data from any format to the required one is not exists. Further, we consider elaboration of a general adaptation module that can be imbedded to the Shell and will transform data from different formats to the internal resource representation format.

We consider resource closeness visualization as a one of the main functions of 4I Browser and are going to supply Interface GUI Shell with the interface for closeness context creation and modification.

We have not concentrated our efforts on development of measuring functions that depends on remote services. It always makes system more unreliable. But still, these methods can be utilized as well, if the goal can not be achieved in another way. We still are going to increase a number of distance measuring methods and types of compared resource description fields.

The same technique that we use for resource closeness visualization can be utilized for resource ranking. The only requirement for this is to describe “virtual/abstract” (or chose from existing) etalon resource and calculate the distances of all other resources to that one. Such approach can be utilized for not complex ranking methods. For complex methods for sure we have to elaborate appropriate modifications. We consider a work in this direction as a future one.

UBIWARE Deliverable D2.2:
Workpackage WP7:

3 Metso Automation case

3.1 Background

Metso industrial case was selected to be a test bed for a research and development within the WP2 (Distributed Querying and Integration) because of its “distributed nature” and big amounts of data in storages, that can not be collected in one place.

We integrate event flow data (events from monitoring and diagnostic systems) together with the structural and design data to provide a convenient assistant tool for an expert in diagnostics. In other words, ease the access to the relevant information needed for decision making.

We have specified a set of sources for integration:

- Alarm messages collected in RDF format (messages have been collected during two years)
- A sample of the DPM database (performance of each node of the paper machine)
- A sample of the Diary database (events handled and documented by factory workers)
- An excel sheet with the data about causticizing part of the plant from DNAExplorer.

The sources mentioned above are not fully integrated, but the most significant parts (derived from the use case) are semantically adapted. “Semantically adapted” means the description within the domain ontology and development of components, that represent the actual data sources as a virtual memory. I.e. the data is not fully transformed into S-APL, but it is annotated to answer the semantic queries instead.

3.2 Special requirements

The software requirements include a web browser with Adobe Flash player version 8.0 or higher. En example could be Internet Explorer 6, Internet Explorer 7, Mozilla Firefox 2, Mozilla Firefox 3, Safari, Opera. Supporting operating systems include MS Windows XP, MS Windows Vista, MacOS X and GNU/Linux.

3.3 Metso Automation Prototype

Architectural overview

The application consists of three main parts: Flash-based web application, Dispatcher servlet and Ontonuts agent. The whole process of communication is displayed in Figure 3.1.

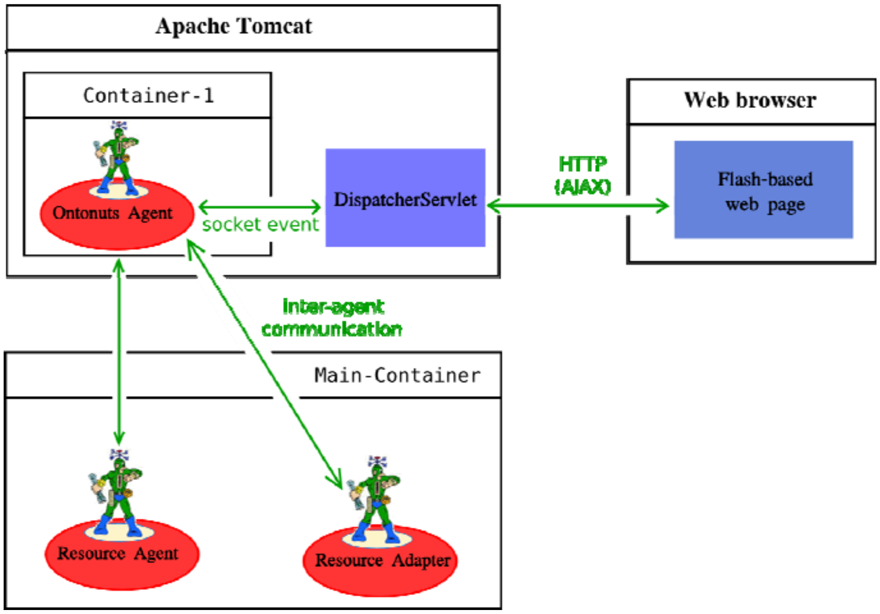


Figure 3.1 – The architecture of the prototype.

The flash-based application is running in the user's browser and communicates with the Dispatcher servlet via HTTP protocol. The whole communication is done asynchronously using the AJAX technology. The flash-based application is querying the servlet and the servlet dispatches the query to the Ontonuts Agent. When the query result is produced, it is sent back to the flash application. This is done every time the user initiates the query.
The communication between the Dispatcher servlet and the Ontonuts agent is done trough socket events. Ontonuts agent is running in a separate agent container (Container-1). The agent performs planning and executes sub-queries. For this purpose it accesses other resources (relational database, XLS sheets, sesame repository, etc.). For this reason it may further be connected to other agents, which act as resource adapters.

Flash application

The graphical part of the application was developed using an open-source framework called OpenLaszlo 4.2.0 (http://www.openlaszlo.org/). The framework is licensed under Common Public License (CPL) and therefore suitable for commercial use (OpenLaszlo, 2006). The resulting end-user application is a Flash-based or DHTML-based (JavaScript with HTML) web page. In both cases, the asynchronous way of communication is used. The application communicates with the server using the AJAX technology. Since DHTML exporting option was introduced only lately, we do not consider it mature enough to be used in a production environment. Therefore we decided to use Flash only.

Dispatcher servlet

Dispatcher servlet is a mediator between the graphical part of the application and the agent part of the application. It takes requests from the graphical user interface and sends them to Ontonuts agent. The other function is that on the first run it starts Ontonuts agent in a separate container.

Ontonuts agent

Ontonuts agent is responding to mediated user requests. The user can request information about a single object or provide a complex query, which result could be a group of objects. The agent uses the Ontonuts engine to perform complex queries. When the query is performed and the results are available, they are transformed into an XML and sent back to the user through the mediator (Dispatcher servlet).

Starting the application

The application starts with a splash screen (see Figure 3.2). While the splash screen is being displayed, the data is being loaded on the background. After the data is loaded, the starting button will appear. By clicking on the starting button, the main user interface window will appear and the application can be used.

Figure 3.2 – The starting splash screen.

The application is divided into two main areas: left pane and right pane (Figure 3.3). The left pane contains fields and buttons used for writing the input of the search. The right pane is used to display the search results. The left pane contains two sub-panes: immediate search pane (top) and quick search pane (bottom).

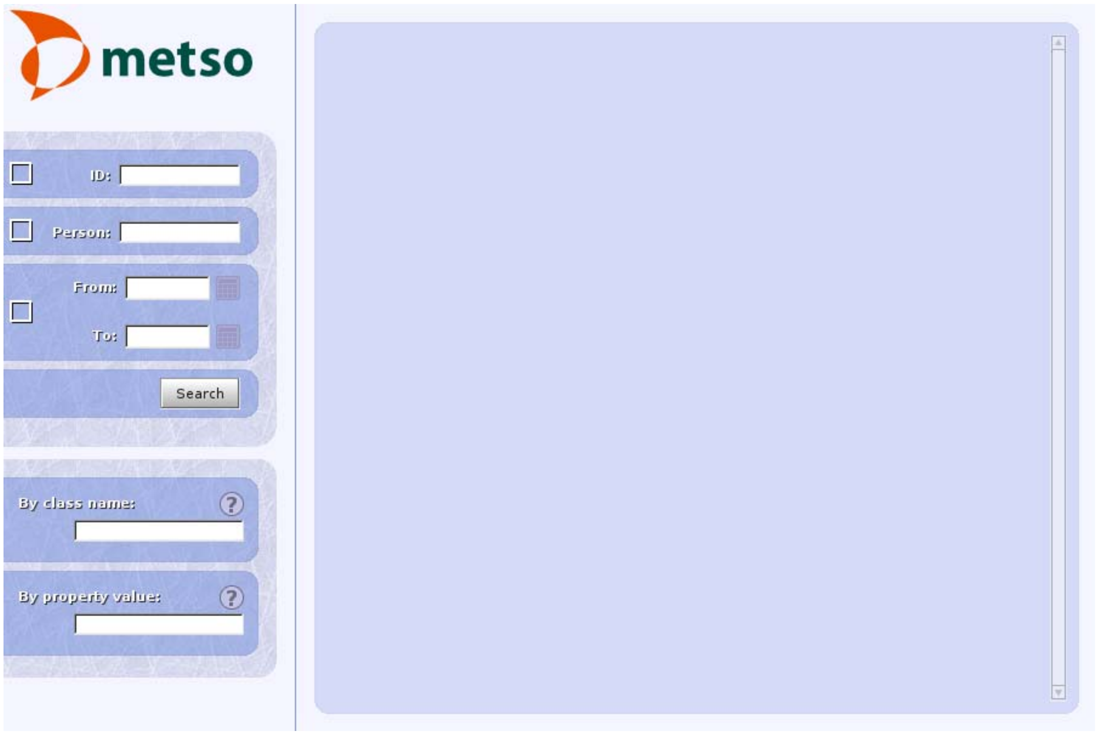


Figure 3.3 – The main screen.

Immediate search

The immediate search pane is located in the upper left part of the application. It consists of three input areas and one search button. Each input area can be activated or deactivated by clicking on the left checkbox (Figure 3.4). By clicking on the checkbox, the user says that this input criterion should be contained in the search. The logical connection between the criteria is logical AND.

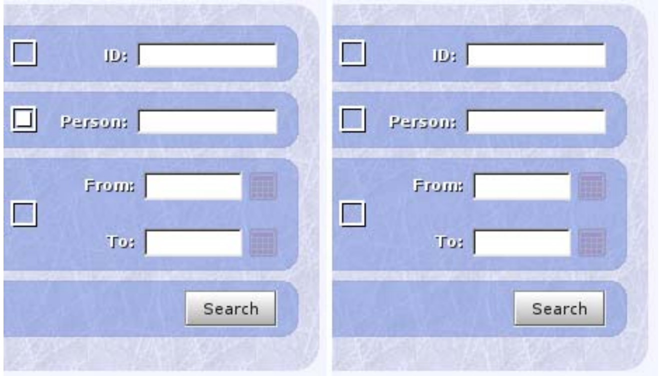


Figure 3.4 – The checkboxes.

The first field called *ID* is used for searching according to the resource ID of an object in the database. When searching for an object, it is important to enter the whole ID, not just a part of it. If the user wishes to enter just a part of it, he can use wildcards. The meaning of different wildcards is explained in Table 1 and an example of a wildcard search is displayed in Table 2.

Wildcard	Meaning
?	A substitute for zero or one character (number or letter)
*	A substitute for zero or more characters (numbers or letters)

Table1 - Wildcards

Searched string	Interpretation
person123	Search for all objects that have ID exactly *person123*
person1*	Search for all objects that have ID starting with *person1*
person	Search for all objects that have ID, which has *person* in the middle (This includes also ID "person", because an asterisk can stand for zero characters)
machine0?	Search for all objects that have ID, which starts with *machine0* and after it there is zero or one character (This includes also ID "machine0", because a question mark can stand for zero characters)

Table 2 – Examples of the wildcard search

The second field *Person* is used for person search. The user can enter any part of the person's first name, surname or login.
The third and fourth field is used for search for an event within a time frame. The user can set the starting date and ending date and click on the *Search* button. The application will then show all events that happened during this period. Note that the starting date and the ending date are also included in the search. It means that if you use 1.1.2009 as the starting date and 3.1.2009 as the ending date, the search will search also events that happened during these two days.

Quick search
The quick search pane is located in the lower left part of the application (see Figure 3.5). It contains two independent fields. One of them is used to search data by class name and the other one is used to search by the property value.

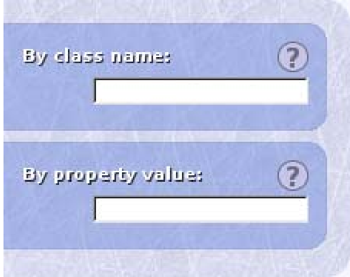


Figure 3.5 – A quick search pane.

In the *Search by class name* field, the data is loaded in the beginning of the application. While the user is typing the input string, possible options are being displayed in the dropdown menu beneath the field (Figure 3.6). The options will start to appear when at least two characters are written. After the menu is show, the user can use either mouse or keyboard to navigate through it. Up and down keys are used to move and the enter key is used to search for the selected item in the menu. After selecting an item from the list (by mouse or keyboard), the item will be displayed in the result pane.

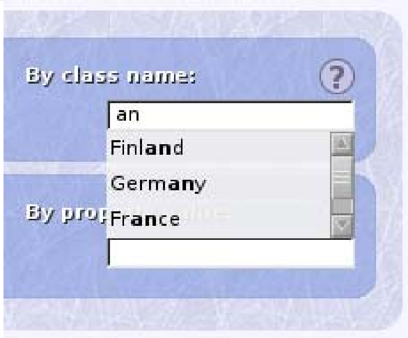
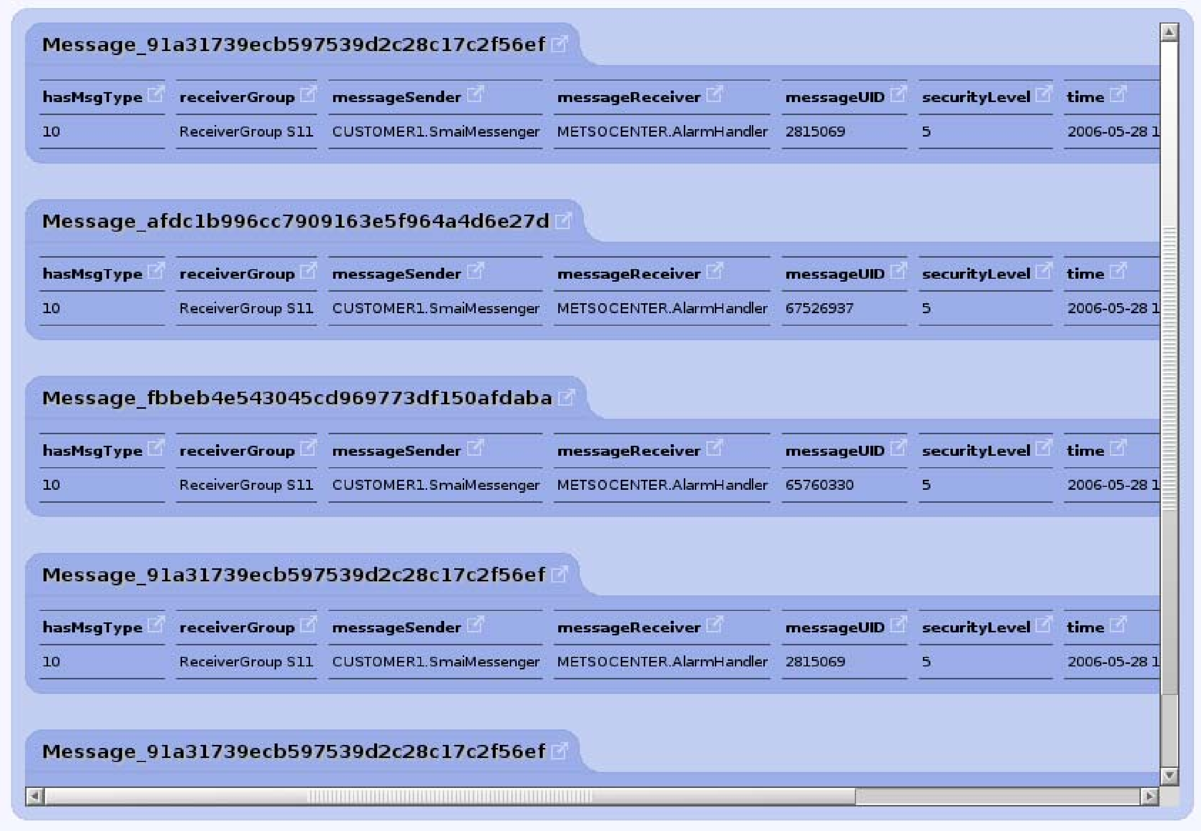


Figure 3.6 – A suggestions list menu.

In the *Search by property name* field, the user can start typing the desired string, but the result will show only when he stops writing for one second. This behavior is implemented due to the fact that the search can be very time consuming (especially with short strings). Therefore, if the user is not writing anything, he signalizes to the application that a search should be performed. The way of choosing an item from the menu is the same as in the case of *Search by class name* field.



Message_91a31739ecb597539d2c28c17c2f56ef

hasMsgType	receiverGroup	messageSender	messageReceiver	messageUID	securityLevel	time
10	ReceiverGroup S11	CUSTOMER1.SmaiMessenger	METSOCENTER.AlarmHandler	2815069	5	2006-05-28 1

Message_afdc1b996cc7909163e5f964a4d6e27d

hasMsgType	receiverGroup	messageSender	messageReceiver	messageUID	securityLevel	time
10	ReceiverGroup S11	CUSTOMER1.SmaiMessenger	METSOCENTER.AlarmHandler	67526937	5	2006-05-28 1

Message_fbbeb4e543045cd969773df150afdaba

hasMsgType	receiverGroup	messageSender	messageReceiver	messageUID	securityLevel	time
10	ReceiverGroup S11	CUSTOMER1.SmaiMessenger	METSOCENTER.AlarmHandler	65760330	5	2006-05-28 1

Message_91a31739ecb597539d2c28c17c2f56ef

hasMsgType	receiverGroup	messageSender	messageReceiver	messageUID	securityLevel	time
10	ReceiverGroup S11	CUSTOMER1.SmaiMessenger	METSOCENTER.AlarmHandler	2815069	5	2006-05-28 1

Message_91a31739ecb597539d2c28c17c2f56ef

Figure 3.7 – A multiple item result.

Result pane

The result pane is used for displaying three kinds of results: class result, single item result and multiple items result. In every result, there is a link which leads to another object as in the hypertext. A class result contains all the information about the class including superclass name and all property names with their domains. A single item result displays the information about a single instance of a class. This includes the name and value of all its properties. A multiple item result consists of a series of single item views ordered in a tabular form (Figure 3.6).

3.4 Future opportunities

Metso Industrial case has been developed as an example usage scenario that demonstrates the applicability of the UBIWARE platform to corporate distributed search needs. The software presented here is easy to adapt to other problem domains, therefore future research directions will include automated configuration of the interface in accordance with the ontology provided. The agent interaction within the application is domain independent and the dependent parts are covered with the Ontonut definitions, which bind domain-dependent resources to the platform. The Ontonuts engine is used within the application to create virtual data storage for an agent that answers user queries; however the applicability of this platform feature is not limited to querying. In the future platform releases we will improve the functionality of the engine to demonstrate its full potential.

4 Nokia Research Center case

4.1 Background

Mobile devices can now run web servers. However, the content inside the mobile devices is currently not accessible from outside (and cannot be indexed by Google). The content needs to be made public and flexibly searchable from outside depending on the access rights of the searcher.

Lots of research has been done how to index content and make it searchable on servers. But research how to search real-time from multiple servers organized into a social network is still very young. Nokia Research Center case advances social network search area and possibly generates a global scale search service.

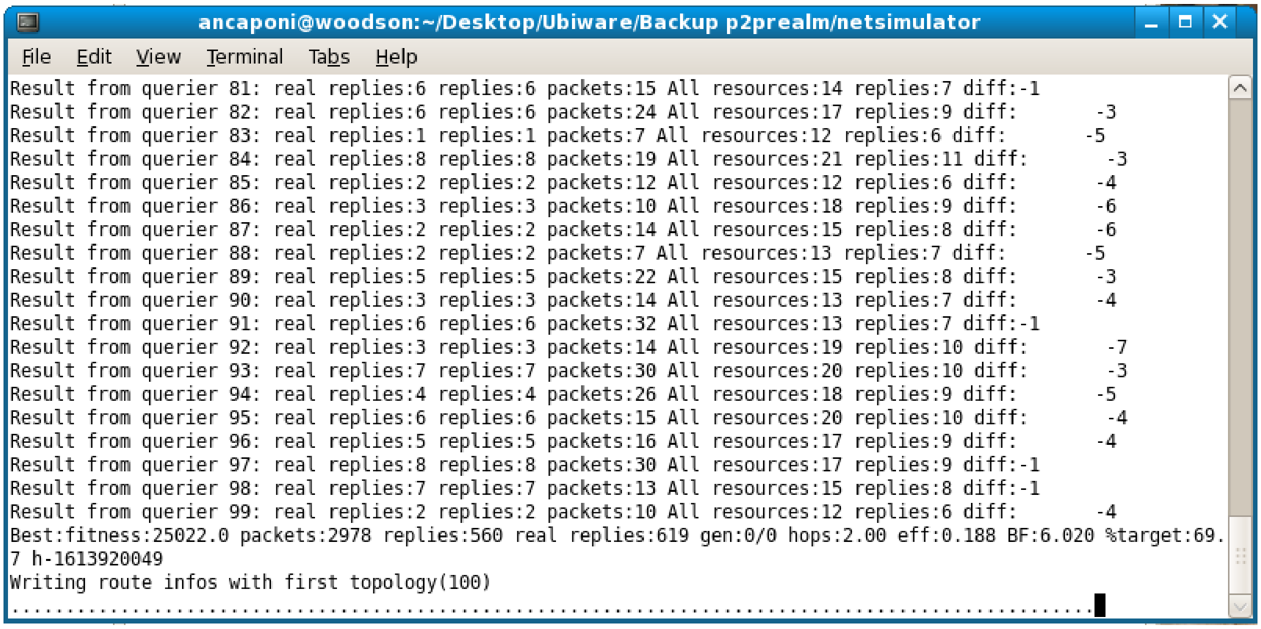
4.2 Nokia Prototype

4.2.1 Results of 2008

1. UBIWARE project selected Last.fm[3] social network site as a source for social network simulation data to get information how music community is organized. Last.fm social network site was crawled with data of 177000 social network users including the top songs the users are listening and the associated albums.
2. Last.fm data is used as a data for Peer-to-Peer Realm simulator[4]. Parts of Peer-to-Peer Realm network simulator were reimplemented to include features for social network simulations and importing the whole Last.fm crawl to P2PRealm is under construction:
    a. P2PRealm was chosen since we are not interested in simulating the network behavior (i.e. connection delays, packet loss, data corruption...), but we just need to emulate data connection between the peers of the network. Before finally choosing the P2PRealm simulator, the OverSim network simulator[5] was studied, but since its purpose is to emulate in detail different kind of networks, taking into account all the different phenomena that could affect the data transmission, we decided that the depth of the OverSim infrastructure exceeded our needs. Low-level simulation is also a significant performance bottleneck when simulating large graphs.

[3] Last.fm – Discover new music with free internet radio and the largest music catalogue online, www.last.fm
[4] P2PRealm – Peer-to-Peer Network Simulator, kotilainen.eu/papers/P2PRealm.pdf
[5] OverSim: The Overlay Simulation Framework, www.oversim.org

b. In order to simulate the social network and the peer network at the same time, few modifications were done to the original P2PRealm simulator (user interface of the simulator is shown in Figure 4.1). Specifically a new network layer was added: the social network. Each node in a P2P network is seen as part of two different networks: one is the data network, the one representing all the data connections available; i.e. if node A is connected with node B through a data connection, then node A can send and receive data to and from node B, and vice versa. The other network is the social network, which represents all the social connections available in the network; i.e. if node A is connected to node B through a social connection, then node A and node B might have common resources and know each other, but this does not mean that they are neighbors in the data connection network with each other: to do so they need to be connected by a data connection. According to this new need, in the original P2PRealm simulator, a representation of the social network was added.


ancaponi@woodson:~/Desktop/Ubiware/Backup p2prealm/netsimulator
File
Edit
View
Terminal
Tabs
Help

```
Result from querier 81: real replies:6 replies:6 packets:15 All resources:14 replies:7 diff:-1
Result from querier 82: real replies:6 replies:6 packets:24 All resources:17 replies:9 diff:       -3
Result from querier 83: real replies:1 replies:1 packets:7 All resources:12 replies:6 diff:       -5
Result from querier 84: real replies:8 replies:8 packets:19 All resources:21 replies:11 diff:       -3
Result from querier 85: real replies:2 replies:2 packets:12 All resources:12 replies:6 diff:       -4
Result from querier 86: real replies:3 replies:3 packets:10 All resources:18 replies:9 diff:       -6
Result from querier 87: real replies:2 replies:2 packets:14 All resources:15 replies:8 diff:       -6
Result from querier 88: real replies:2 replies:2 packets:7 All resources:13 replies:7 diff:       -5
Result from querier 89: real replies:5 replies:5 packets:22 All resources:15 replies:8 diff:       -3
Result from querier 90: real replies:3 replies:3 packets:14 All resources:13 replies:7 diff:       -4
Result from querier 91: real replies:6 replies:6 packets:32 All resources:13 replies:7 diff:-1
Result from querier 92: real replies:3 replies:3 packets:14 All resources:19 replies:10 diff:       -7
Result from querier 93: real replies:7 replies:7 packets:30 All resources:20 replies:10 diff:       -3
Result from querier 94: real replies:4 replies:4 packets:26 All resources:18 replies:9 diff:       -5
Result from querier 95: real replies:6 replies:6 packets:15 All resources:20 replies:10 diff:       -4
Result from querier 96: real replies:5 replies:5 packets:16 All resources:17 replies:9 diff:       -4
Result from querier 97: real replies:8 replies:8 packets:30 All resources:17 replies:9 diff:-1
Result from querier 98: real replies:7 replies:7 packets:13 All resources:15 replies:8 diff:-1
Result from querier 99: real replies:2 replies:2 packets:10 All resources:12 replies:6 diff:       -4
Best:fitness:25022.0 packets:2978 replies:560 real replies:619 gen:0/0 hops:2.00 eff:0.188 BF:6.020 %target:69.
7 h-1613920049
Writing route infos with first topology(100)
...................................................................................................
```

Figure 4.1 – User interface of P2PRealm.

3. To study how different search algorithms work on a community of music listeners we need distributed algorithms for discovering local social network topology of each user. Topology awareness algorithm solving this problem has now been specified and implemented to P2PRealm. Next step is to implement social network search algorithm, which provides the first simulation results on Last.fm data.

   a. Topology awareness algorithm is in charge of building the social topology of the network and to order the queries of a search so that the chances to receive more answers are maximized. According to the algorithm each peer node keeps information about the other peers it knows (because they are social neighbors or because they answered to previous queries). In the beginning each node will know only its 1st level social neighbors. When a query is started, the topology algorithm creates an empty list of N elements (where N is an integer number). Then the algorithm adds to this list the social neighbors of the sending node; if the number of these nodes is equal or more than N, then the list is complete and it must be sorted according to the criteria given below. If the list is not filled by the social neighbors of the node, then the algorithm starts adding to the list the social neighbors of the sending node social neighbors, i.e. the 2nd level social neighbors, the 3rd level social neighbors, and so on until the list is filled. Each new neighbor met in this process is then added to the social neighbors of the sending node, so that the filling process will be faster next time.

   b. Once the list of the querying node is full, it must be sorted: the criteria with which this list is ordered is under study and should be optimized in order to maximize the number of the answers received by the sending node.

4. Music listener should be able to get new and interesting music to his mobile device automatically and also be able to search music depending on his interests. This kind of functionality is planned to be provided via mobile social P2P prototype. User interface for such prototype is under design. With the prototype it will be possible to demonstrate real-time searching of relevant new content from other mobile phones (including also music content).
5. Social network search scenario can be further extended into a future information network. In addition to social aspect, this information network also supports searching based on location information and different interest areas of the users. Searching is a feature that should be actively done only by certain users and providing relevant information automatically to other inactive searchers would increase the speed of information flow significantly. Such functionality has been sketched as a part of the future information network.

4.2.2 Future Information Network

A vision of a future information network is presented in Figure 4.2. The information network combines different distance measures between the entities of the information network particularly emphasizing on three different types of relationship measures (solid arrows): social distance (SD), geographical distance (GD) and interest distance (ID). Nodes A, B, C, D and E represent the containers of information and can be workstations, servers or mobile devices. The information can be e.g., pictures (P), blogs (B), files (F), URLs (U), music (M), video (V), web pages (W) etc. The information is linked via social, geographical and interest distance links between the containers of information and also with URL pointers denoted with dashed arrows.

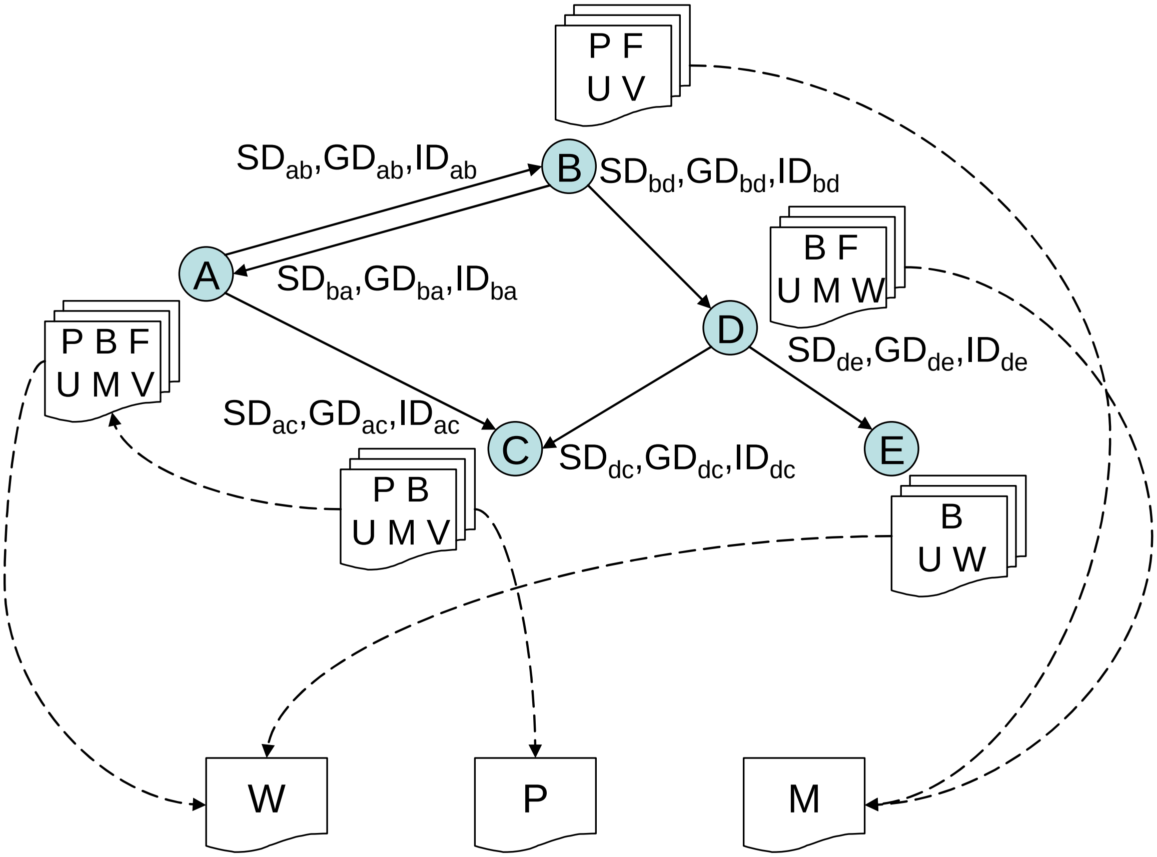


Figure 4.2 - Future information network.

The idea of the future information network is to provide means for users of the system to execute searches based on different distance measures and thus obtain information via traditional keyword search or even without any keywords depending on the density of URL pointers within the close neighborhood of the searcher. The user can learn of information that he/she does not know even exists (and therefore the user cannot search for it) by finding out the URLs that his/her social neighborhood has found interesting. Also the user can make his/her browsing information available to others and fully customize what parts of his/her personal device will be made accessible to other. This creates virtual representations of the users of the information network leading to a virtual reality experiences while using the information network. It is also expected that such kind of a network cannot be implemented in a centralized manner because of the large amount of information leading to bandwidth bottlenecks.

Consider these example use cases of the future information network:

1. If the user would like to locate a certain information containing a set of keywords, he/she can select an information container where to start searching and define whether he wants to search nearby friends (social distance is short), nearby people (geographical distance is short), people with similar interests (interest distance is short) or any other weighted combination of these distance measures. The search algorithm would then use the [SD,GD,ID] vectors of the selected information container to select which neighbors to query. The neighbors would return the matching results and at the same time a list of their

most promising neighbors filtered by the query vector's [SD,GD,ID] weights. If the results would not satisfy the searcher the lists of neighbors could be used to derive a new set of [SD,GD,ID] neighbor vectors for searching until a satisfying set of documents are located.

2. If the user would like to know what his nearby friends have found interesting, he/she can select his own information container (for example node A) and query his social neighborhood (social distance is short and geographical distance and interest distance can be any value). The query would return a set of URL pointers to documents which nodes B, C and D have accessed earlier (E might be considered to have too long social distance value and would not be queried). The result set would therefore contain document M. Now the user obtained information M without knowing any keywords matching M or even that such kind of information could be searched in the information network.

This kind of information network goes beyond keyword searching because keywords are not needed for obtaining interesting information. The information diffusion is semi-automated which results in an increase of the speed of information flow. Different algorithms need to be developed as a future work to point out which documents will be presented to the user if multiple URLs point to documents. This suggests studying collaborative filtering techniques and ranking algorithms based on multiple different criteria (social, geographical and interest distance values need to be taken into account). An interesting problem is how these information containers' distances are weighted in the document ranking algorithm.

There are many ways to measure the distances of social, geographical and interest relationships. It is also likely that these three dimensions might not be sufficient for all kinds of search cases. For example mobility and context information aspects need to be considered. To calculate the distance measures two information containers need to contact each other and store the distance values making it also an open problem how to update these values and for which information containers to keep these values up-to-date. This suggests a need for future work on topology awareness algorithms. In overall, the future information network can be implemented in many different ways: centralized, partially centralized or completely decentralized. Which architecture becomes the best alternative remains to be seen.

4.3 Future opportunities

During Spring 2009 we are planning to simulate resource discovery algorithms in mobile social P2P network scenario based on Last.fm crawl. NeuroSearch framework[6] provides variety of test cases and the purpose is to find a set of input variables useful for searching in a social network. Later result ranking algorithms can be tested and several publications will be written about these algorithms. A fully functional social P2P network prototype will be implemented if further funding of the project can be obtained after Spring 2009.

[6] An Adaptive Global-Local Memetic Algorithm to Discover Resources in P2P Networks, research.jyu.fi/p2pgroup/documents/P2PLNCS.pdf

References

Cilibrasi, R., Vitanyi, P.(2007) The google similarity distance. IEEE Transactions on knowledge and data engineering 19(3), 370–383

Dumais, S. (2003) Data-driven approaches to information access. Cognitive Science, 27(3), 491-524.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998) Introduction to latent semantic analysis. Discourse Processes, 25, 259-284.

Lemaire, B., Denhiére, G. (2004) Incremental construction of an associative network from a corpus. In K. D. Forbus, D. Gentner & T. Regier (Eds.), 26th Annual Meeting of the Cognitive Science Society, CogSci2004. Hillsdale, NJ: Lawrence Erlbaum Publisher.

OpenLaszlo - An Open Architecture Framework for Advanced Ajax Applications (White Paper), Nov 2006, 19 pp., http://www.openlaszlo.org/whitepaper/LaszloWhitePaper.pdf

Pirolli, P. (2005). Rational analyses of information foraging on the Web. Cognitive Science, 29(3), 343-373.

Veksler, V. D., Gray, W. D. (2007) Mapping semantic relevancy of information displays. Paper presented at the CHI 2007, San Jose, CA.