

ACTIVE

Deliverable D6.2.1

Demonstrator version of ACTIVated tools from WP1-WP3

Editor:	Igor Dolinsek, HSL
Deliverable nature:	Prototype (P)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31.8.2009
Actual delivery date:	11.9.2009
Suggested readers:	ACTIVE project Case study participants in particular and all Users of the ACTIVE knowledge workspace and ACTIVated applications in general.
Version:	0.3
Total number of pages:	3
Keywords:	ACTIVE, ACTIVE knowledge workspace, ACTIVated applications

Abstract

Deliverable D6.2.1 is a prototype which consists of the ACTIVE extensions of the SemanticMediaWiki, Windows File Explorer and LiveNetLife. This document provides a high level description of the ACTIVE-specific functionality of those applications, explains how they fit into the overall ACTIVE knowledge workspace architecture, and gives the location of the software deliverables and the related documentation in the project software repository. The demonstrator version covers the functionality which is needed to demonstrate and validate the key research concepts of the project. Based on the outcome of the case study validations this functionality will be expanded and/or modified for the fully functional prototype in the next development iteration.

[End of abstract]

Disclaimer

This document contains material, which is the copyright of certain ACTIVE consortium parties, and may not be reproduced or copied without permission.

All ACTIVE consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ACTIVE consortium as a whole, nor a certain party of the ACTIVE consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

[Full project title] ACTIVE – Enabling the Knowledge Powered Enterprise

[Short project title] ACTIVE

[Number and title of work-package] WP6: Integration

[Document title] Demonstrator version of ACTIVated tools from WP1-WP3

[Editor: Name, company] Igor Dolinsek, HSL

[Work-package leader: Name, company] Igor Dolinsek, HSL

[Estimation of PM spent on the Deliverable] 21

Copyright notice

© 2008 Participants in project ACTIVE

Executive summary

Deliverable D6.2.1 is a prototype which consists of the **ACTIVE extensions** of three **software applications**, contributed by the technical partners from WP1-WP3:

- SemanticMediaWiki from AIFB
- LiveNetLife from Jure Ferlez, hosted at JSI
- Windows File Explorer from EMIC

This document provides

- explanation how this application extensions fit into the overall ACTIVE knowledge workspace architecture
- high level description of the ACTIVE-related functionality of the applications
- instructions where in the project software repository these software deliverables and the related documentation are located.

Demonstrator version covers the functionality which is needed to demonstrate and validate the key research concepts of the project. Based on the outcome of the case study validations this functionality will be expanded and/or modified for the fully functional prototype in the next development iteration.

Note that the three applications are not part of the deliverable, only their extensions which provide the ACTIVE-related functionality are included here.

Note that the actual D6.2.1 deliverables are software deliverables (the installable software packages and the related documentation with installation and setup instructions and acceptance scenarios). These deliverables were reviewed according to review process described in the project's Software Development Guidelines.

List of authors

Company	Author
HSL	Igor Dolinsek
JSI	Jure Ferlez
AIFB	Denny Vrandečić
AIFB	Frank Dengler
AIFB	Stephan Bloehdorn
AIFB	Markus Krötzsch
EMIC	Marcel Tilly

Table of Contents

Executive summary	3
List of authors.....	3
Table of Contents	3
List of figures and/or list of tables.....	3
Abbreviations	3
Definitions.....	3
1 Introduction.....	3
2 Position in the ACTIVE knowledge workspace architecture.....	3
3 High level description of the extensions	3
3.1 Semantic MediaWiki extensions.....	3
3.1.1 Logging SMW	3
3.1.2 Managing user metadata	3
3.1.3 Context aware queries.....	3
3.1.4 Contagging resources.....	3
3.1.5 Installation and Configuration of the ACTIVE SMW Extensions	3
3.2 LiveNetLife extensions	3
3.3 Windows File Explorer extensions	3
4 Location of software deliverables	3
References	3

List of figures and/or list of tables

Figure 1 Position of D6.2.1 extensions in the ACTIVE workspace architecture 3

Abbreviations

AKWS – ACTIVE knowledge workspace

SMW – Semantic MediaWiki

ActiveMQ – Message Queuing System by Apache

Definitions

Not applicable

1 Introduction

Project partners are developing a number of applications which could be enhanced with ACTIVE-related functionality and integrated into the ACTIVE knowledge workspace as ACTIVated applications. This deliverable includes three such application extensions:

- SemanticMediaWiki from AIFB
- LiveNetLife from Jure Ferlez, hosted by JSI
- Windows File Explorer from EMIC

This document provides only the

- explanation how these applications fit into the overall ACTIVE knowledge workspace architecture
- high level description of the ACTIVE-related functionality of the extensions
- instructions where in the project software repository these software deliverables and the related documentation are located.

The reader of this document should be familiar with the ACTIVE project in general and in particular with the deliverables D5.1.1[2] and D5.1.2 [3]. The basics of SMW are given in deliverable D1.3.1 [6]. The overview of LiveNetLife is given in D2.2.1 [9].

In order to make use of some extension provided by the D6.2.1 deliverable, it is necessary to install the ACTIVE server (D5.3.1) and the basic application of the respective extension.

2 Position in the ACTIVE knowledge workspace architecture

SemanticMediaWiki, LiveNetLife were incorporated into the ACTIVE knowledge workspace as ACTIVated applications. The application extensions were developed according to the architecture guidelines, provided by D5.1.2 and by using the ACTIVE SDK (D5.4.1 [4]). Following figure (Figure 1) outlines their position in the overall ACTIVE knowledge workspace. Components related to D6.2.1 are marked as such.

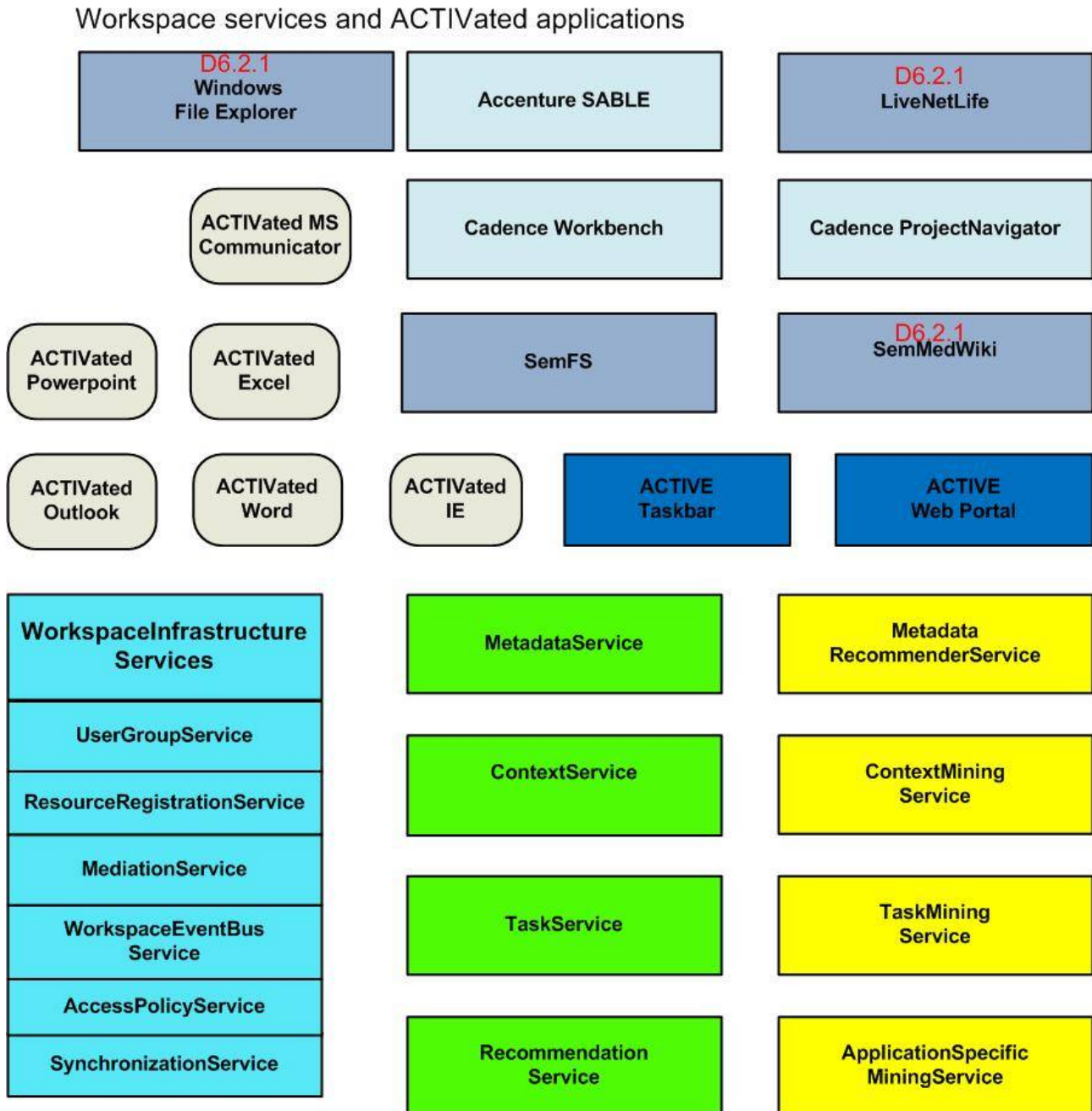


Figure 1 Position of D6.2.1 extensions in the ACTIVE workspace architecture

3 High level description of the extensions

3.1 Semantic MediaWiki extensions

Semantic MediaWiki is a server-side software component that is accessed via a web browser. This requires a slightly different approach towards activating the software, compared to standard desktop software like Microsoft Office.

This section describes the extended functionalities – Logging SMW, managing user data, context aware queries, and contagging resources. Furthermore, this section describes how to access, install, and configure the Semantic MediaWiki extensions for ACTIVE.

3.1.1 Logging SMW

Since all interaction with SMW is performed via the browser, and the browser will be logged anyway, all interactions with SMW are already available to AKWS on a low level. But beyond this low level interaction logging, we provide an extension to SMW that creates a message every time an article is saved. This message is extending the given schema.

This way tools can also examine what the actual changes to the SMW articles are. With the simple browser logging the actual changed text would not be visible due to the fact that the changes are given by POST requests.

Here is an example for an UpdateSMWArticleEvent:

```
<UpdateSMWArticleEvent>
  <EventDomain>PrimitiveEvents</EventDomain>
  <DateTime>8/10/2009 1:31:57 PM</DateTime>
  <Task>None for now</Task>
  <User>WikiUserName</User>
  <EventType>UpdateSMWArticleEvent</EventType>
  <Operation>UpdateArticle</Operation>
  <InputParameter>
    <URI>http://wiki.active-project.eu/wiki/Denny_Vrandecic</URI>
    <Text>Denny's office phone is 6050</Text>
  </InputParameter>
  <OutputParameter>
    <Text>Denny's office phone is 6098</Text>
  </OutputParameter>
</UpdateSMWArticleEvent>
```

There are two InputParameters: the URI of the page that is being changed, and the previous wikitext of the article in the “Text” parameter. The OutputParameter only contains one parameter, the new text. This allows the mining service (and any other listener) to see the changes of text in the message, and to be able to follow the actions of the users.

The messages are published to the PrimitiveEvent.SMWEvent.UpdateSMWArticleEvent topic and can be accessed there.

3.1.2 Managing user metadata

In order for SMW to integrate data from the AKWS, the wiki needs to be able to access the data about the user. For that it requires the users AKWS login and password. For the AKWS login a property “Property:AKWS login” is used, with the datatype Type:String (i.e. the page “Property:AKWS login” has the annotation “[[has type::Type:String]]”). On the user’s page we then can set a property with the AKWS login, i.e. on the page “User:Example User” there would be an annotation “[[AKWS login::example]]”. This way the relation between the user and their AKWS login will also be exported in the user’s RDF description and can thus be used by external tools.

The password on the other hand is sensitive data and should not be displayed to all users of the wiki or exported to the outside world. Therefore we use the SMW preferences system to save the password. In order to setup the password, a user has to go to their `Special:Preferences` page, and set up the AKWS password in the “Basic Information” section of the “User profile” preference panel.

Once both the login and the password are set up, the wiki can access AKWS and connect to the context data of the user.

Preferences

User profile
Appearance
Date and time
Editing
Recent changes
Watchlist
Search options
Misc
Gadgets

— Basic information —

Username: Denny

User ID: 1

Member of groups: Administrators, Autoconfirmed users, Bureaucrats, Users, Widget editors

Number of edits: 259

Registration time: 15:52, 30 July 2009

Real name:

Real name is optional. If you choose to provide it, this will be used for giving you attribution for your work.

Gender: Male

Optional: used for gender-correct addressing by the software. This information will be public.

Password: [Change password](#)

Remember my login on this computer

AKWS password:

— Internationalisation —

Language: en - English

3.1.3 Context aware queries

One of the major features of SMW is the ability to query the knowledge inside it. The AKWS holds the context of the user. In order to make the queries more specific to the user needs, SMW provides a way to add the context to the query, and thus provide results that are all relevant to the context.

In order to do so, the ACTIVE extensions add a new Special page, `Special:AskContext`, where SMW adds the context of the user to the query. In order for this to work, SMW and the AKWS have to be connected as described in the previous section.

The special page queries the context from the AKWS and sets that as the context for the query. The user can still choose to select another context, or to disregard the context altogether. For this, a new drop down list with a list of all contexts known to the wiki has been added to the ask page. Note: currently, in order for a context to be visible in the drop down list, there needs to be a wiki page that is in the category Context, i.e. in order to add a context to the Wiki, add a page (like "Paris bid 2009") and then put it into the category context by adding `[[Category:Context]]`. In a future release, all contexts from the AKWS will be visible.

3.1.4 Contagging resources

Whenever the user edits a page while he is in a specific context, the context is added to the page. Together with the context aware search, this will allow for pages to be found easier that have been edited while the user was in that given context.

Within Semantic MediaWiki, the context is simply modelled by articles of their own that are classified in the `Category:Context`. Furthermore a `Property:Context` is used to connect an arbitrary page to a given context, e.g. `[[context::ACTIVE]]`. The `Property:Context` has the type `Type:Page`, since contexts are described by their own page.

3.1.5 Installation and Configuration of the ACTIVE SMW Extensions

The ACTIVE extensions are accessible from the ACTIVE SVN. Since they are proper extensions of the already existing SMW system, they require a working SMW application. They also require a working AKWS in order to consume the events being created, and to provide the wiki with the context.

Besides that, the system has the following requirements:

- MediaWiki Version 1.16 or higher (which is considerably higher than SMW requires). The unified user management requires the `GetPreference Hook` that was only added for the most current version of MediaWiki. Since Version 1.16 has not been released yet, the ACTIVE extensions require actually the current SVN version of MediaWiki. The MediaWiki SVN version can be obtained as described here: http://mediawiki.org/wiki/Download_from_SVN
- For the ActiveMQ message queuing the system requires the STOMP library for PHP. The ACTIVE extensions were developed using Version 1.0 for PHP5. The Stomp library is available here: <http://stomp.codehaus.org/PHP>
- For SOAP calls, we are using the NuSOAP library. We developed the ACTIVE extensions using Version 0.7.3 of NuSOAP. NuSOAP is available here <http://sourceforge.net/projects/nusoap>

Once, these additional requirements are fulfilled, the ACTIVE extensions need to be copied to the MediaWiki extensions directory.

In order to add the ACTIVE extensions, the following line has to be added to the file `LocalSettings.php` of the Semantic MediaWiki instance to be extended.

```
require_once( "$IP/extensions/ActiveExtensions/ActiveExtensions.php" );
```

Before that line, we need to set up a number of configuration variables being set up. Currently, there are four such variables.

```
$agActiveMQHost = 'http://active-project.org:6663/activeMQ';  
$agAKWSServer = 'http://active.ijs.si:3000';  
$agStompLibrary = '/Users/username/Sites/stompcli-php5-1.0-20080916';  
$agNuSoapLibrary = '/Users/username/Sites/nusoap-0.7.3';
```

`$agActiveMQHost` gives the URL of the ActiveMQ host the update messages are sent to.

`$agAKWSServer` gives the URL of the AKWS server. Using that, the system queries for the UUID of the user and the context of the user.

`$agStompLibrary` gives the path to the Stomp library.

`$agNuSoapLibrary` gives the path to the NuSoapLibrary.

3.2 LiveNetLife extensions

We have developed a LiveNetLife contextualization plug-in called AKWS2LNL, which is an integration point between the LiveNetLife and the AKWS platform. The purpose of this integration is to extend the original LiveNetLife functionality by not only enabling people to meet when they are working on similar information per se, but by also considering their current working context to better understand the information they are currently using.. In particular, AKWS2LNL provides text contextualization web method, which takes as input a string of text and "contextualizes" it by removing all the text which is related to the user's current working context. The current working context description is provided by the AKWS Context Mining Service described in [5]. The AKWS2LNL contextualization method is consumed by the LiveNetLife server so that it can correctly find people who are working on similar topics even when they are using different contexts.

The achieved extended functionality of LiveNetLife is best demonstrated with an example. Originally, when two users visited web pages [Active Homepage](#) and [ACTIVE Summer School Homepage](#) then LiveNetLife would report that the web pages are 67% similar. This is because both pages give similar information on ACTIVE technology, however the former also includes basic project information, whereas the latter also includes summer school specific information. Now, by integrating with AKWS, the current working context of the user is also taken into account, when text in these pages is analyzed. For instance we expect, that if the two users are at the moment working in the "*ACTIVE Project*" context, then the two pages are less similar (one is about the project in general, the other about the summer school). Indeed, when the two pages are visited by users who are currently both working in "*Active Project*" context according to the AKWS, then LiveNetLife reports that the pages are only 50% similar. This is because all the information about the "*Active Project*" was removed from the text, leaving only summer school and more specific technology description in the two pages.

The developed AKWS2LNL plug-in is integrated with LiveNetLife by utilizing possible parameterization of LiveNetLife installation into user's web browser [7]. Most importantly, the parameters *LiveNetLife_UserId* and *LiveNetLife_ContextualizedServiceAddress* are used to enable simple connection with the AKWS2LNL decontextualization service. The former is used to signal the AKWS username of the user, the latter is used to signal the URL address of the AKWS2LNL contextualization service. For example these parameters signal that the user is *Dunja* and that the AKWS2LNL contextualization service is running at *hope.ijs.si*.

```
var LiveNetLife_UserId="Dunja";  
var LiveNetLife_ContextualizedServiceAddress="http://hope.ijs.si:9731/AKWS2LNL/";
```

3.3 Windows File Explorer extensions

The Windows File Explorer is a central desktop application to manage files on local or shared folders. An activation of this tool provides information of the user side usage of files, e.g. creation of new files, renaming of an existing file, moving files from one folder to another or simply the deletion of files.

The first of the ACTIVated Explorer is only an integration on a basic level. It fires the above mentioned events. Even this simple integration is useful to help the AKWS to maintain references to files on the local system within the Meta data store.

The current integration of the Windows Explorer exists on the file system itself. The implementation observes a given path (directory) and reports changes within this directory. Table 1 provides an overview of current events.

Table 1: Events of ACTIVated File Explorer

Topic	Description	Message Sample
PrimitiveEvents .FileBrowserEvent.FileCreated	New creation of a file	<pre><FileBrowserPrimitiveEventParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <ID>e9a873b8-a37a-40b6-868c-24cb4a6446e0</ID> <OldFilename>d:\tmp\test.txt</OldFilename> <NewFilename>d:\tmp\test.txt</NewFilename> <UserId>Max007</UserId> <Host>xyz</Host> <LastModified="">2009-08- 21T11:43:28.7204334+02:00</LastModified> </FileBrowserPrimitiveEventParameter></pre>
PrimitiveEvents .FileBrowserEvent.FileRenamed	Renaming of a file	<pre><FileBrowserPrimitiveEventParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <ID>a1e13011-10d8-424e-92e7-d9ea719c7dbb</ID> <OldFilename>New_test.txt</OldFilename> <NewFilename>test.txt</NewFilename> <UserId>Max007</UserId> <Host>xyz</Host> <LastModified>2009-08- 18T10:27:56.8195311+02:00</LastModified> </FileBrowserPrimitiveEventParameter></pre>
PrimitiveEvents .FileBrowserEvent.FileDeleted	Deletion of a file	<pre><FileBrowserPrimitiveEventParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <ID>03d94ea9-b101-423f-bff7-132cc6b897a2</ID> <OldFilename>test.txt</OldFilename> <NewFilename>test.txt</NewFilename> <UserId>Max007</UserId> <Host>xyz</Host> <LastModified>2009-08- 18T10:27:57.0375436+02:00</LastModified> </FileBrowserPrimitiveEventParameter></pre>
PrimitiveEvents .FileBrowserEvent.FileMoved	Moving a file	<pre><FileBrowserPrimitiveEventParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <ID>e80bfc3d-e1bd-46ce-9b2e-9e711ef40b2a</ID> <OldFilename>new_location\test.txt</OldFilename> <NewFilename>old_location\test.txt</NewFilename> <UserId>Max007</UserId> <Host>xyz</Host> <LastModified>2009-08- 18T10:27:57.0345434+02:00</LastModified> </FileBrowserPrimitiveEventParameter></pre>

In a next version the Explorer could be integrated in terms of tagging files, tag files for a specific context or highlight files within the current context (see Internet Explorer activation).

4 Location of software deliverables

Actual deliverables are located in the ACTIVE project subversion repository

<http://svn.active-project.eu/active/repos>

in

akws/bin/demonstrator

directory.

Software deliverables are provided in form of a zipped package and the related “readme” files:

- SemMedWiki_ext.zip:
- SemMedWiki_ext_readme.docx
- AKWS2LNLSetup.msi
- AKWS2LNLreadme.html
- AKWSFileObserverSetup.msi
- AKWSFileObserver_readme.docx

In order to make use of some extension provided by the D6.2.1 deliverable, it is necessary to install the ACTIVE server (D5.3.1) and the application of the respective extension.

It is possible to deploy each extension separately. Note that the applications are not part of this deliverable. They have to be obtained and installed as described in the respective readme files for the extensions.

References

- [1] <http://www.active-project.eu> : Project external website
- [2] ACTIVE deliverable D5.1.1: ACTIVE Knowledge workspace Requirements Analysis
- [3] ACTIVE deliverable D5.1.2: ACTIVE Knowledge Workspace Architecture and Design
- [4] ACTIVE deliverable D5.4.1: ACTIVE SDK
- [5] ACTIVE deliverable D5.3.1: Demonstrator version of ACTIVE Knowledge Workspace
- [6] ACTIVE deliverable D1.3.1: Collaborative articulation of enterprise knowledge – Early prototypes
- [7] ACTIVE deliverable D3.1.1: Knowledge Filters and Metaphors – Early Prototype
- [8] http://www.livenetlife.com/help_customize.aspx: LiveNetLife Customization Guide
- [9] ACTIVE deliverable D2.2.1: Module with algorithms for processing & integration of multiple modalities - Demonstrator

[end of document]