

GRAPPLE

4.4 Version: 1.0

Device and network adaptation infrastructure

Document Type	Deliverable
Editor(s):	Daniele Gambarotto, Michele Dicerto, Lucia Oneto (GILABS)
Author(s):	Daniele Gambarotto, Michele Dicerto, Lucia Oneto, Fabrizio Anagi (GILABS)
Reviewer(s):	Fabian Abel, Frederic Minne
Work Package:	WP4
Due Date:	M18
Version:	1.0
Version Date:	2010-01-15
Total number of pages:	9

Abstract: This report describes the development work of the Network and Device Adaptation. The document contains a description of the integration in the GALE system. A brief tutorial of the tools is provided.

Keyword list: content adaptation, mobile devices, XSL engine

Summary

This report describes the development work of the Network and Device Adaptation and outlines how the GRAPPLE Device Adaptation (GDA) is integrated into the GALE system. Further, this document provides a brief tutorial on installing and using the GDA tools.

Authors

Person	Email	Partner code
Fabrizio Anagi	f.anagi@giuntilabs.com	GILABS
Michele Dicerto	m.dicerto@giuntilabs.com	GILABS
Daniele Gambarotto	d.gambarotto@giuntilabs.com	GILABS
Lucia Oneto	l.oneto@giuntilabs.com	GILABS

Table of Contents

SUMMARY	2
AUTHORS	2
TABLE OF CONTENTS	2
TABLES AND FIGURES.....	3
LIST OF ACRONYMS AND ABBREVIATIONS	3
1 INTRODUCTION	4
2 DESCRIPTION OF THE TEXT RECAP AND IMAGE RESIZE MODULES.....	5
3 SPECIFICATION OF THE XLS FILES	6
4 IMPLEMENTATION AND INTEGRATION OF THE GDA TOOL WITH GALE.....	6
4.1 How to install it in GALE	6
4.2 How it works.....	7
5 CONCLUSION	8
REFERENCES	8
6 APPENDIX A: DEVICEADAPTION.XSL.....	8

Tables and Figures

List of Figures

Figure 1: GRAPPLE adaptation process details..... 4

List of Acronyms and Abbreviations

GAL	GRAPPLE Adaptation Language
GAT	GRAPPLE Authoring Tool
GALE	GRAPPLE Adaptive Learning Environment
GDA	GRAPPLE Device Adaptation
GRAPPLE	Generic Responsive Adaptive Personalized Learning Environment
HTTP	Hyper Text Transfer Protocol

1 Introduction

The purpose of this deliverable is to describe the implementation work to build a software module able to adapt the content of the adaptive courses deployed by GALE for mobile users.

This deliverable is the accompanying documentation related to software prototype delivered by Task 4.4 – Implementation of an infrastructure for device and network adaptation. This task develops the technical infrastructure based on the models from task 4.3. The core of the infrastructure will be an adaptation engine that will get as input contextual information, contents descriptions and semantic information (about user and domain). Properly developed matching and transformation algorithms, based on the set of rules identified as part of the model, will process the contents according to the current context.

As described in [1], the course adaptation is performed in a central core engine, GALE, where the courses are adapted based on the user and device profiles. This requires the integration of the GDA Tool within the GALE framework as displayed in Figure 1.

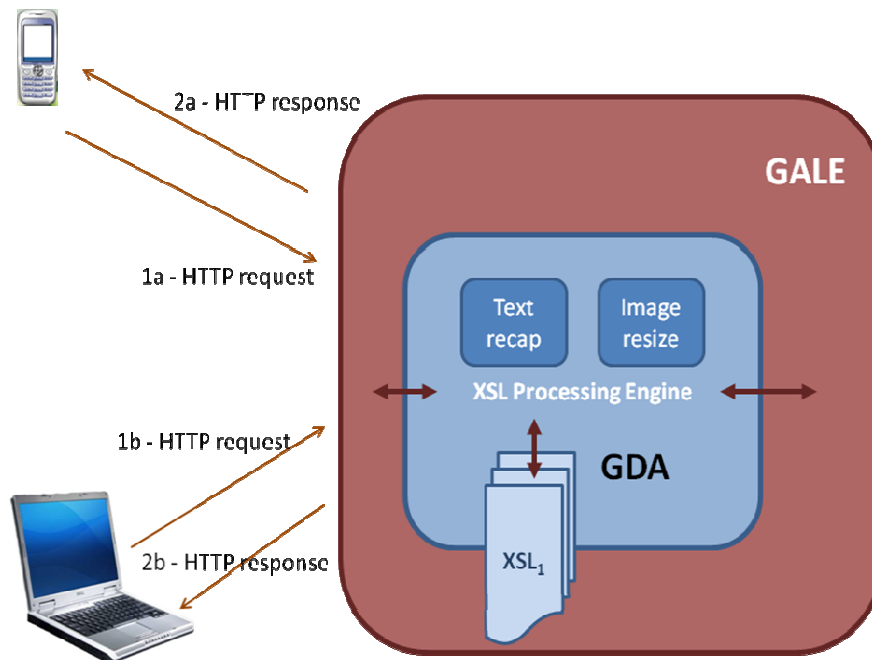


Figure 1: GRAPPLE adaptation process details.

GDA permits to process content published into GALE and deliver it to mobile device. It uses an XSL file to transform XHTML formatted content into a proper representation that will be sent to a mobile client. For image resizing, it replaces the original image URL with a URL of a resized image produced by the image resizing module. During XML transformation, text is summarized by making use of the text summarization module. Requests are sent via HTTP Post handshaking.

The implemented GRAPPLE adaptation process satisfies the following software requirements:

- GDA is the module in charge to adapt the content (course) in relation to the type of device
- The device can be a computer or a mobile.
- Content adaptation means: appropriate resize of the images and proper recap of the text (when necessary)
- GDA has to be able to analyze the HTTP request messages that carry device profile information
- GDA is composed by an XSL Processing Engine able to apply the XSL file appropriate to the learner's device (computer or a mobile), and a series of modules in charge of resizing image, repaging and text recap
- GDA is a module internal to GALE (adaptation engine of the GRAPPLE system).

2 Description of the Text Recap and Image Resize modules

The Text Recap (ResizeText) and Image Resize (ResizeImage) modules are implemented as Java servlet modules.

The Text recap module uses the Classifier4J¹ Java library (version 0.6, Classifier4J-0.6.jar) and receives original text and the maximum number of sentences that have to be present in the summarized text.

Then the following input parameters are mandatory:

- **text**: original text to elaborate
- **numLines**: number of sentences that compose the summarized text.

The module elaborates the original text only in case the original text is composed by a number of sentences greater than **numLines**.

These parameters can be passed by GET/POST http method.

Example of usage:

`http://<Servlet Server Address>/ResizeApp-0.2/ SummarizeText?text=<Text to summarize>&numLines=4`

ResizeImage is a servlet that uses the library *Image scale*² (imagescalePackage.jar) to reduce properly the image. The module is able to elaborate the following formats:

- jpg,
- gif,
- png.

The reduced image is generated in jpg format.

The module needs the following mandatory parameters of type text:

- **ResizeImageServletURL** = URL of resize image servlet module
- **ImageWidth** = width of resized image in pixels
- **ImageHeight** = height of resized image in pixels.

If one of two size parameters (ImageWidth and ImageHeight) is equal to 0 (zero), the other must be greater than 0. Images are resized without stretching.

An original image with height x width less of 100x100 pixels is not resized.

ResizeImage can have also the following optional input parameters:

- **Dir** = folder where the original images are stored
- **Expires** = time interval in seconds that indicates how long the images are stored in cache
- **Cache** = address of the cache where the images are stored.

Example of usage:

`http://<Servlet Server Address>/ResizeApp-0.2/ResizeImage?urlImage=<Gale Server Address>/gale/Milkyway/img/img_milkyway.jpg&width=200&height=0`

Text recap and Image resize modules can be also installed into a server different from the GALE one.

¹ <http://classifier4j.sourceforge.net/>

² <http://coldjava.hypermart.net/servlets/imagescale.htm>

3 Specification of the XLS files

In order to allow a proper content adaptation based on the type of device, it is necessary to have at least one XSLT file able to provide the adaptation rules as described in [1].

DeviceAdaption.xsl (see Section 1) is an XSLT file used to transform the original XHTML file and send the proper result to the user's device.

It is loaded at run-time by GALE framework from "WEB-INF" internal path: it copies all the XHTML elements as they are, except for images and text.

For elements image XSLT transformation replaces *src* attribute, so it is able to point to URL of resize module.

Text elements are replaced with result coming from Text Recap module.

XSLT transformation is performed only for request coming from mobile devices.

4 Implementation and Integration of the GDA Tool with GALE

This chapter describes how to install the GDA Tool into the GALE framework and identifies the issues still open.

4.1 How to install it in GALE

To install GDA into GALE the following steps have to be followed:

1. Copy deviceAdaption.jar into WEB-INF/lib path of GALE installation;
2. Copy DeviceAdaption.xsl into WEB-INF path of GALE installation;
3. Add this bold content to galeconfig.xml (red content is already present)

...

```
<bean id="processorList" class="org.springframework.beans.factory.config.ListFactoryBean"
scope="session">
<property name="sourceList">
<list>
```

...

```
<bean class="giuntilabs.grapple.gale.processor.DeviceAdaptionProcessor" scope="session">
  <property name="deviceAdaptionProcessorTable">
    <map>
      <entry>
        <key>
          <value>ResizeImageServletURL</value>
        </key>
        <value><URL of servlet server>/ResizeApp-
0.2/ResizeImage</value>
      </entry>
      <entry>
        <key>
          <value>SummarizeTextServletURL</value>
        </key>
        <value><URL of servlet server>/ResizeApp-
0.2/SummarizeText</value>
      </entry>
      <entry>
        <key>
          <value>ImageWidth</value>
        </key>
        <value>200</value>
      </entry>
```

```

        <entry>
            <key>
                <value>ImageHeight</value>
            </key>
            <value>0</value>
        </entry>
        <entry>
            <key>
                <value>minLenghtToSummarize</value>
            </key>
            <value>2</value>
        </entry>
        <entry>
            <key>
                <value>lenghtSummarizedText</value>
            </key>
            <value>2</value>
        </entry>
    </map>
</property>
</bean>
...
</list>
</property>
</bean>
...

```

4.2 How it works

The configuration file in GALE, galeconfig.xml, manages the usage of the Image Resize module mainly by playing with the mandatory parameters

- **ResizeImageServletURL** (URL of resize image servlet module),
- **ImageWidth** (width of resized image), and
- **ImageHeight** (height of resized image).

If one of two size parameters is equal to 0 (zero), the other must be greater than 0: then image is resized without stretching. As described above, original images with height/width less of 100px are not resized.

For the Text Recap module, the following parameters are used:

- **SummarizeTextServletURL** = URL of Recap Text servlet module
- **minLenghtToSummarize** = minimum length of the sentence to summarize
- **lenghtSummarizedText** = number of sentences summarized.

Open issues

There are a few issues still open:

- the words that contain punctuation could distort the result of summarized module (for instance the word AHA! causes this problem)

- it is not able to manage paragraph content in original file that contains tag like “”: this may corrupt the result of Text Recap module.

5 Conclusion

This report presented a description of the development work of the GDA Tool and the related Release Notes. The work could be carried out to introduce some important improvements. First at all, the text summarization algorithm could be improved to guarantee a greater correctness by fixing also the still open issues. Second, the page lay outing based on the device features needs a further research and development work. This could be possible adding more granularity in the device identification. Third, right now the GDA Tool does not manage video files, but only text and images.

All these considerations can be taken into account for a second phase of the project.

References

1. Cantini S., Lorenzon A., Oneto L., Device and Network Adaptation Model, v1.0, July 2009.

6 Appendix A: DeviceAdaption.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xobj="giuntilabs.grapple.gale.processor.DeviceAdaptionProcessor">
  <xsl:output method="html"/>
  <xsl:param name="resizeImageServletURL"
select="'http://localhost:8080/ResizeApp-0.2/ResizeImage'" />
  <xsl:param name="baseImageURL" select="'http://localhost:8080/gale/'" />
  <xsl:param name="ImageWidth" select="'800'" />
  <xsl:param name="ImageHeight" select="'600'" />

  <xsl:template match="*[not (name()='img')] ">

  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates/></xsl:apply-templates>
  </xsl:copy>
</xsl:template >
<!-- Summarize text by calling SummarizeText implemented in java package
-->
<xsl:template match="text()" >
  <xsl:value-of select="xobj:SummarizeText(.)" />
  <xsl:apply-templates/></xsl:apply-templates>
</xsl:template>

<!-- Change Img tag src attribute URL to resize applet URL-->
<xsl:template match="*[ (name()='img')] " >
  <xsl:if test="starts-with(@src, 'http') ">
    <xsl:copy>
      <xsl:copy-of select="@*[not (name()='src') and not
(name()='width') and not (name()='height')]" />
      <xsl:attribute name="src">
<xsl:value-of select="$resizeImageServletURL"/>
<xsl:text>?urlImage=</xsl:text>

```

```
<xsl:value-of select="@src"/>

<xsl:text>&width=</xsl:text>
<xsl:value-of select="$ImageWidth"/>
<xsl:text>&height=</xsl:text>
  <xsl:value-of select="$ImageHeight"/>
</xsl:attribute>
<xsl:apply-templates></xsl:apply-templates>
</xsl:copy>
</xsl:if>
<xsl:if test="not (starts-with(@src, 'http')) ">
  <xsl:copy>
    <xsl:copy-of select="@*[not (name()='src') and not
(name()='width') and not (name()='height')]" />
    <xsl:attribute name="src">
<xsl:value-of select="$resizeImageServletURL"/>
<xsl:text>?urlImage=</xsl:text>
      <xsl:value-of select="$baseImageURL"/>
<xsl:value-of select="@src"/>
<xsl:text>&width=</xsl:text>
        <xsl:value-of select="$ImageWidth"/>
        <xsl:text>&height=</xsl:text>
        <xsl:value-of select="$ImageHeight"/>
      </xsl:attribute>
    <xsl:apply-templates></xsl:apply-templates>
  </xsl:copy>
</xsl:if>
</xsl:template>
</xsl:transform>
```