

# Open Corpus Adaptation++ in GALE: Friend or Foe?

David Smits and Paul De Bra

Faculty of Mathematics and Computer Science, Eindhoven University of Technology,  
Postbus 513, 5600 MB Eindhoven, The Netherlands  
d.smits@tue.nl, debra@win.tue.nl

**Abstract.** “Open” has quickly become the hottest topic in any field related to *information*, including open government data, open learning resources, open user models, ... Open Corpus Adaptation has been defined as the ability to perform adaptation to resources located anywhere on the Web. This leaves the definition of and control over the adaptation in a central place. GALE adds the ability to have the adaptation (definition) distributed over the Web. In this paper we describe how GALE achieves this functionality and we raise the question whether this is actually a desired feature or potentially a dangerous addition with unintended consequences.

## 1 Introduction and Motivation

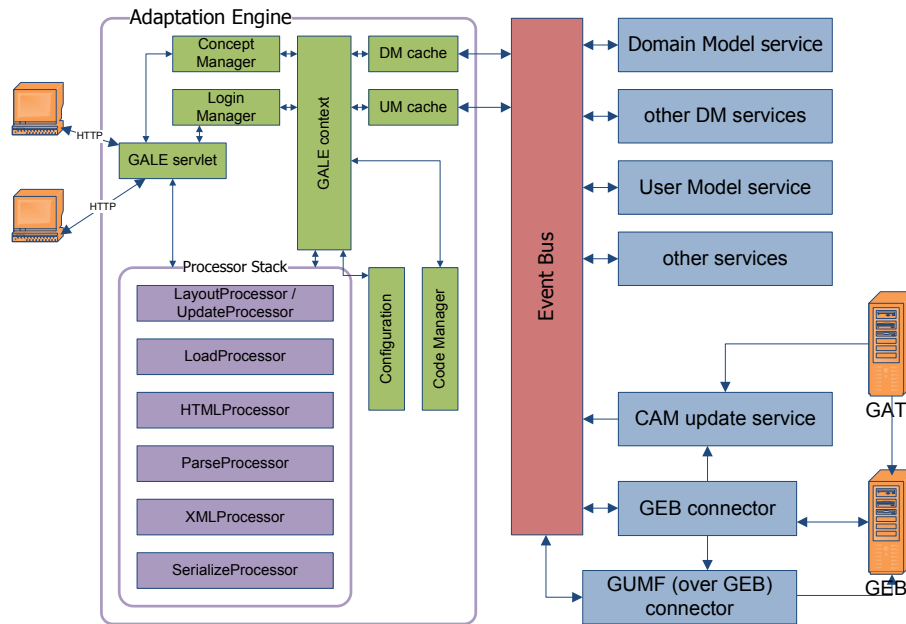
Using hypertext to open up and link all available information was first suggested by Ted Nelson when introducing Xanadu (see <http://www.xanadu.net/>) and became a reality soon after the introduction of the Web. The initial Web was a “safe” environment, where all information was static. The browser could download any web page and display it, and the user would be assured that this would not have any side effects. Since then the (on-line) world has become much more dynamic. Our data resides “in the cloud”, processing is done “in the cloud”, but even when just accessing websites they know (remember) who we are, and they may cause our browser to execute code we have no control over.

So far *adaptive* hypermedia applications have been “safe”: an adaptive application is served by a single *adaptive hypermedia system* (AHS), providing adaptation to local resources, and storing user-related information in a local database. These applications are also “closed”. Initiatives to open up AHS have so far approached two aspects: 1) the *user model* has become distributed [1], integrating information coming from many different adaptive and non-adaptive applications, including social networks, and 2) the *resources* have become distributed in *open corpus adaptive hypermedia* [2]. Distributing user model and resources has also been a goal in the GRAPPLE project (<http://www.grapple-project.org/>) in which GALE (the GRAPPLE Adaptive Learning Environment) was developed. Within GRAPPLE it was foreseen that the definition of the adaptation would be kept centralized, created through a graphical authoring toolset GAT (GRAPPLE Authoring Tool) [3]. GALE has been designed to be able to perform to resources that can be loaded from anywhere on the Web (retrieved through HTTP). This leads to a seemingly strange situation: an author defines adaptation for a resource in GAT and GALE performs adaptation to that re-

source that is created by someone else, located anywhere in the world, without the author of the resource having any influence on the adaptation that will be performed to that resource. Would it not be logical to enable authors of resources to also define the adaptation (and user model updates) associated with that resource? This is exactly what the *Open Corpus Service* in GALE allows. In Sect. 2 we describe this “open corpus adaptation++” and then (Sect. 3) we discuss the feasibility of actually uptake of this functionality and the potential dangers involved.

## 2 Open Corpus Adaptation++ in GALE

Figure 1 below shows the overall architecture of GALE. This architecture is “distributed” around an internal Event Bus.



**Fig.1.** Core GALE architecture

In “standard” use an application is defined by an author and added to GALE through the “CAM update service”, resulting in a *domain model* (DM) which in GALE contains the conceptual structure and the adaptation for the application. The GALE event bus can connect different DM services with the common adaptation engine. It can also connect different *user model* services, including an internal GALE user model service and an external GRAPPLE User Model Framework GUMF. In this paper we concentrate on the Open Corpus Service, which is a DM service.

A domain model in GALE is defined using the GAM language (GALE application model). The authoring process normally results in a set of *concepts* with for each

concept the associated GAM code that defines *properties*, *user model attributes* and *event code* for the concept and its attributes. All requests to GALE normally specify a concept (not a web page). When the concept specification refers to a concept on an external server (the concept is requested from another server through HTTP) the Open Corpus service retrieves that concept and scans the file for a <meta> element with a 'name' attribute with value 'gale.dm'. When no information for the current concept is found, the Open Corpus service searches for files called concept.gdom and concept.gam (where concept stands for the actual concept name). It does so from the current path in the URL up to the root of the server specified. The first description found on the current concept is used.

Below is an example <http://gale.win.tue.nl/elearning.xhtml> (taken from [4]) with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:gale="http://gale.tue.nl/adaptation">
  <head>
    <meta name="gale.dm" content="
      { #[visited]:Integer `0` {
        event `if ({#suitability} && {#read} < 100)
          #{#read, 100};
          else if (!{#suitability} && {#read} < 35)
            #{#read, 35};` }
        #knowledge:Integer !`GaleUtil.avg(new Object[]
        {${<=(parent)#knowledge},${#read}}).intValue()`
        #[read]:Integer `0`
        #suitability:Boolean `true`
        event `#{#visited, ${#visited}+1};` } " />
    </head>
    <body>
      <p>This page is a placeholder for the elearning
        concept.</p>
    </body>
  </html>
```

We don't describe the details of the GAM syntax and semantics here, but only briefly explain the example code:

- The code event `#{#visited, \${#visited}+1};` } means that when the concept is accessed the value of the "visited" attribute is increased by 1.
- The attribute "visited" is an integer, and when its value changes its event code is executed which updates the "read" attribute.
- The attribute "read" is also an integer.
- The attribute "knowledge" is an integer which is not stored but calculated from the "read" value and the "knowledge" value of the children of the "elearning" concept.
- The attribute "suitability" is a Boolean, which is "true" by default. This too is not stored but calculated when needed. If there were prerequisites for the "elearning" concept there would be an expression that defines the condition for the concept to become suitable.

Another “page” can “inherit” this adaptation (GAM) code as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:gale="http://gale.tue.nl/adaptation">
  <head>
    <meta name="gale.dm" content={->(extends)
      http://gale.win.tue.nl/elearning.xhtml} />
  </head>
  <body>
    <p>This page uses the elearning template.</p>
  </body>
</html>
```

When a whole application domain is stored in a single file the “meta” element for the concepts/pages would look like:

```
<meta name='gale.dm' content='redirect:course.gam' />
```

and the file “course.gam” might have contents like:

```
welcome.xhtml {
  ->(extends)http://gale.win.tue.nl/elearning.xhtml
  ->(extends)layout.xhtml
  <-(parent)gale.xhtml
  <-(parent)gat.xhtml
}
gale.xhtml {
  ->(extends)welcome.xhtml
  ->(parent)welcome.xhtml
}
gat.xhtml {
  -> (extends)welcome.xhtml
  ->(parent)welcome.xhtml
}
layout.xhtml {
  #layout:String `
  <struct cols="250px;*">
    <view name="static-tree-view" />
    <struct rows="60px;*;40px">
      <view name="file-view" file="gale:/header.xhtml" />
      <content />
      <p><hr />Next suggested concept to study:
        <view name="next-view" /></p>
    </struct>
  </struct> `
}
```

Again we do not explain this code but just illustrate that code can be shared between different concepts/pages, and can be placed in individual files or combined into a single GAM file.

When GALE retrieves “open corpus GAM definitions” it treats them just like a locally stored definition: the concepts are created, user model information is stored and updated, and the adaptation of other concepts (and the retrieved concepts themselves) can depend on user model values for both these external and internal concepts. The event code in GAM is essentially arbitrary Java code. This has potentially serious implications which we discuss in the next section.

### 3 The implications of Open Corpus Adaptation++

When “dynamic” content was first introduced on the Web it came with significant security concerns. To illustrate:

- Browser plug-ins consist of executable code that can potentially harm the end-user’s computer. It has full access to all resources to which the browser has access. A harmful plug-in can not only crash the browser but also wipe the user’s hard drive, send spam messages, search for critical personal data on the hard drive like credit card numbers and transfer that to a criminal organization, etc.
- Scripting code can be made somewhat less dangerous depending on what the scripting language allows.
- Java Applets are running within a *Sandbox* environment: they cannot read or write any information on the hard drive and they can only make network connections to the site from which they are downloaded. The end-user can make an exception (for signed applets) to allow access to the hard drive and network.

The Open Corpus Service in GALE allows arbitrary GAM code to be stored in the domain model, after which it is executed by the GALE Adaptation Engine (AE). This AE executes GAM event code which is arbitrary Java code that stores, retrieves and updates user model information, but that in principle can try to also do anything else. The security measures within GALE are:

- The AE runs in a *Sandbox* environment just like browser applets. The code has no direct access to the hard drive or the network. Its only “way out” of the *Sandbox* are the methods the *Sandbox* provides. These methods must allow the service to store and retrieve user model data.
- The only user model access that is allowed is to the user model of the user for whom the AE is executing code. This currently prevents GALE from providing “group adaptation” but it is at least “secure”.

Although the adaptation engine cannot do anything “truly harmful” it does perform user model updates. And with open corpus adaptation++ the AE performs user model updates defined by possibly unknown authors. When the end-user types the URL to access a remote concept on any server through the local AE that local AE will execute whatever GAM code the unknown author has written. This code may potentially retrieve “private” user model information, and it may also destroy valuable information in the user model. This is currently a concern that is specific to GALE as GALE is the only “open corpus adaptation++ engine” we know of. GALE provides basic safety of user model information by limiting user model updates to concepts with a URI relative to the URI of the concept where the code resides. But the issue as to what should be allowed (and what not) in open corpus adaptation++ is still open in general.

## 4 Discussion and Conclusions

This paper presented the concept of *Open Corpus Adaptation++* where not only the *corpus* is distributed over the Web but also the *adaptation model* is distributed. This is currently just a novel feature offered by GRAPPLE's Adaptive Learning Environment GALE, and not yet widely used because the current authoring tool set GAT still does not support specifying open corpus adaptation++. The code shown in Sect. 2 is clearly not intended to be hand-written by human authors, so authoring tools will be needed.

But most importantly the paper has raised concern that open corpus adaptation++ can be potentially harmful so we should discuss what is permissible and what should be blocked for arbitrary adaptation models loaded from the Web.

## Acknowledgement

We wish to thank the European Commission, project 215434 (GRAPPLE) for their financial support for this research.

## References

1. Abel, F., Henze, N., Herder, E., Krause, D., Interweaving Public User Profiles on the Web, In Proceedings of UMAP 2010, User Modeling Adaptation and Personalization, LNCS 6075, pp. 16-27, Springer, 2010.
2. Brusilovsky, P., Henze, N., Open Corpus Adaptive Educational Hypermedia, in: The Adaptive Web, pp. 671-696, Springer, 2007.
3. Hendrix, M., Cristea, A.I., Design of the CAM model and authoring tool. A3H: 7th International Workshop on Authoring of Adaptive and Adaptable Hypermedia Workshop, 4th European Conference on Technology-Enhanced Learning, 2009.
4. Smits, D., De Bra, P., GALE: A Highly Extensible Adaptive Hypermedia Engine, Proc. of the ACM Conference on Hypertext and Hypermedia, Eindhoven, 2011.