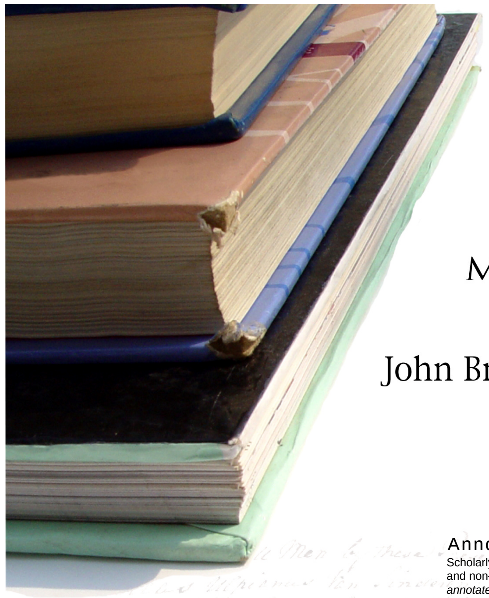


PLINY

Making a contribution: modularity, integration and collaboration between tools in Pliny

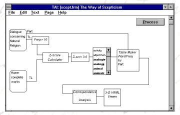
John Bradley, Centre for Computing in the Humanities, King's College London
http://pliny.cch.kcl.ac.uk, john.bradley@kcl.ac.uk

What is Pliny?

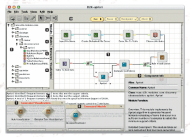
- Pliny is about two things. It illustrates
  1. some of the potential that arises out of developing software that supports annotation and notetaking for the Humanities, and
  2. some of the issues for Graphical User Interfaces (GUI) that should be considered when developing modular software toolkits.
- This poster is primarily about item 2.

Toolkits for Humanists: pipelining

- Much discussion about toolkits for humanists has focused on a modular approach that centers on data pipelining. – a technique much used in data visualisation and related fields.
- Pipelining has also proven to be a powerful model for many textual transformation (see Wilhelm Ott's *TuStep* for very fine example).
- Pipelining serves certain type of computing applications better than others, and is somewhat foreign to the GUI interface



From: Bradley and Rockwell (1995). "The Components of a System for Computer Assisted Text Analysis". Prepared for the *CETH Workshop on Future Text Analysis Tools*"



From: *D2K Toolkit User Manual.* Automated Learning Group, NCSA. 2003

Annotation and resource enrichment: direct manipulation

- An important element of scholarly work is *enrichment*: adding a new layer of materials on top of base materials:
  - TEI markup is often thought of in these terms.
  - Annotation/Notetaking is also this kind of activity.
- Annotation/Notetaking cannot be modelled effectively of in terms of dataflow modularity.
  - An annotation tool must be more like an editor than a transformation utility.
  - Like a text editor, annotation needs to feel to the user as if s/he has "direct manipulation" access to the objects: this is very much a GUI issue.

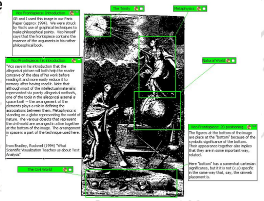
CCH Centre for Computing in the Humanities

KING'S College LONDON University of London

Annotating Everything:

Scholarly annotation might apply to all kinds of digital and non-digital materials. A scholar might want to *annotate anything*.

image



Vico Frontispiece: Introduction
GR and I used this image in our Paris Paper (approx 1994). We were struck by Vico's use of graphical techniques to make philosophical points. Vico himself says that the frontispiece contains the essence of the arguments in his rather philosophical book.

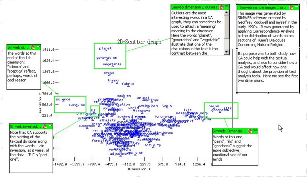
Vico Frontispiece: his introduction
"Vico says in his introduction that the allegorical picture will both help the reader conceive of the idea of his work before reading it and more easily reduce it to memory after having read it. Note that although most of the intellectual material is represented via purely allegorical methods, one of the tools in the allegorical arsenal is space itself -- the arrangement of the elements plays a role in defining the associations between them. Metaphysics is standing on a globe representing the world of nature. The various objects that represent the civil world are arranged in a line together at the bottom of the image. The arrangement in space is a part of the technique used here."

from Bradley, Rockwell (1994) "What Scientific Visualization Teaches us about Text Analysis"

Vico Frontispiece: 2D positioning
The figures at the bottom of the image are place at the "bottom" because of the symbolic significance of the bottom. Their appearance together also implies that they are in some important way, related.

Here "bottom" has a somewhat cartesian significance, but it is not (x,y) specific in the same way that, say, the simweb placement is.

application output



Simweb di...
The words at this end of the 1st dimension: "science" and "sceptics" reflect, perhaps, words of cool reason.

Simweb dimension 2 outliers
Outliers are the most interesting words in a CA graph, they can sometimes be used to attach a "meaning" meaning to the dimension. Here the words "planet", "generation" and "vegetable" illustrate that one of the discussions in the text is the contrast between the

Simweb sample image: Intro
This image was generated by SIMWEB software created by Geoffrey Rockwell and myself in the early 1990s. It was generated by applying Correspondence Analysis to the distribution of words across sections of Hume's Dialogues Concerning Natural Religion.

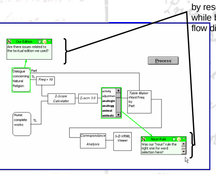
Its purpose was to both study how CA could help with the textual analysis, and also to consider how a CA tool would affect how one thought about the provision of text analysis tools. Here we see the first two dimensions.

Simweb inverted...
Note that CA supports the plotting of the textual divisions along with the words -- ain inversion, as it were, of the data. "P1" is "part one".

Simweb: Dimensio...
Words at this end, "pains", "ills" and "goodness" suggest the more subjective, emotional side of our minds.

Software output from: Bradley and Rockwell (1997). *Simweb Correspondance Analysis Visualizer*.
URL: http://tactweb.mcmaster.ca/cgi-dos/simweb/simweb.bat

process descriptions


Comments added by researcher while building the flow diagram
Process
Dialogue concerning Natural Religion
Part
TL
Freq > 10
Z-Score Calculator
Z-scr> 3.0
Table Maker Word Freq by Part
Hume: complete works
Correspondence Analysis
3-D VRML Viewer

Our Edition
Are there issues related to the textual edition we used?

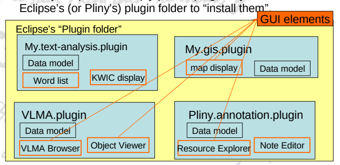
Noun Rule
Was our "noun" rule the right one for word selection here?

Tool Modularity: Pliny and Eclipse

- Pliny takes a modular approach to tool component design based on the Eclipse (http://www.eclipse.org) model.
- Eclipse (and Pliny) supports modularity in ways other than just file-sharing, pipe-lining (although, of course, it provides for these too).
- Much of Eclipse is designed to allow for a sense of integration at the GUI level – on the screen, between separately built components.

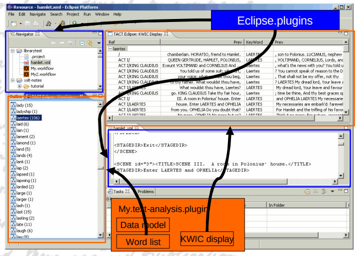
Plugins

- In Eclipse a plugin provides a package framework for a single tool.
- Plugins can contain GUI elements (called by Eclipse views or editors) that can display in panes on the screen.
- In Pliny/Eclipse one simply places plugin objects in Eclipse's (or Pliny's) plugin folder to "install them".



Workbench

- The eclipse workbench manages the windows layout objects to manage screen space: panes, menus, toolbars, etc.
- The user can choose to combine GUI elements from different plugins on the screen at the same time.
- On the following screen displays from a prototype text-analysis plugin co-exist with conventional Eclipse displays in the Workbench.
  - Behind-the scenes synchronization between screens from different plugins is possible – so that if a user clicks on a line in the KWIC display (from the text-analysis tools plugin) the Eclipse XML text editor can be made to jump to the line containing the selected word.



```
ACT I/                chamberlain. HORATIO, friend to Hamlet.   LAERTES   , son to Polonius. LUCIANUS, nephew
ACT I/                QUEEN GERTRUDE, HAMLET, POLONIUS,         LAERTES   , VOLTIMAND, CORNELIUS, Lords, and
ACT I/KING CLAUDIUS   Exeunt VOLTIMAND and CORNELIUS And ...    Laertes   , what's the news with you? You told u
ACT I/KING CLAUDIUS   You told us of some suit; what is't,      Laertes   ? You cannot speak of reason to the D
ACT I/KING CLAUDIUS   your voice: what wouldst thou beg,        Laertes   , That shall not be my offer, not thy
ACT I/KING CLAUDIUS   to thy father. What wouldst thou have,    Laertes   ? LAERTES My dread lord, Your leave a
</pre>
```

```
</STAGEDIR>Exit</STAGEDIR>
</SCENE>

<SCENE id="S1"><TITLE>SCENE III.  A room in Polonius' house.</TITLE>
<STAGEDIR>Enter LAERTES and OPHELIA</STAGEDIR>
```

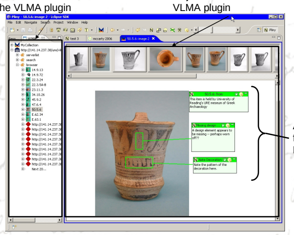
The Registry and managed Memory sharing

- Eclipse provides a registry which allows a plugin to offer services to other plugins.
- Memory sharing can be managed between plugins.
- An object in plugin A can declare (by implementing a Java Interface) that it has the necessary behaviour to allow it to be displayed in displays managed by plugin B.
- These mechanisms supports collaboration between different plugins.

Collaboration between Plugins

- The *Virtual Lightbox for Museums and Archives* (VLMA) is a framework developed by University of Reading, the Max Planck Institute for the History of Science and Oxford Archaeology which gives a user access to an RDF server managing metadata about images, and the images themselves.
- I took the code for the VLMA and created a prototype VLMA plugin from it that supported locating and displaying images from the VLMA system within the Eclipse/Pliny framework.
- Annotation components from the Pliny plugin could co-exist and co-operate with materials provided by my VLMA plugin.
- The following 2 screenshots show this in operation.

Pliny within VLMA

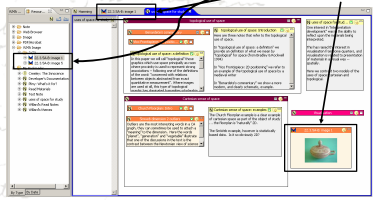

Resource Browser from the VLMA plugin
Object viewer from the VLMA plugin
Annotations from the Pliny plugin
Pliny - 50.5.6: image 2 - Eclipse SDK
File Edit Navigate Search Project Window Help
test 3
mccarty 2006
50.5.6: image 2
MyCollection
serverlist
search
browser
Next 20...

50.5.6: from
This item is held by University of Reading's URE meseum of Greek Archaeology

Missing design ...
A design element appears to be missing -- perhaps worn off??

Note Decoration
Note the pattern of the decoration here.

VLMA within Pliny



Contribution model

- It is easy to add new components (as plugins) into Pliny/Eclipse, and allow them to communicate with each other. This has lead to the language used in Eclipse of a plugin object "making a contribution" to the operation of another plugin.
- Examples for Pliny
  - contributing support for new data formats to Pliny:
    - An plugin could be developed for video or audio that stored its annotations in a Pliny format to allow them to appear on other Pliny screens.
    - A plugin could be developed to support Pliny-like annotation of XML/TEI documents directly.
    - A plugin could be developed to store bibliographic materials that integrated with Pliny
  - Pliny can contribute annotation support to other plugins (such as the VMLA example)

Conclusions

- Eclipse's plugin model allows for the development of tools by independent developers that inter-operate not only at the data level but also at the GUI level – on the screen.
  - This is important for computer users who think of computing in terms of the GUI.
- Pliny provides a set of plugins that support annotation and notetaking – two key activities within Humanities research
- The Eclipse framework allows both others to contribute new functions to Pliny (support annotation of other kinds of digital materials, for example), and allows Pliny to contribute its annotation/notetaking functions to other tools, such as GIS or Text Analysis plugins.
- Building tools that work together in these ways still requires coordination between tool builders, but it provides a framework in which such coordination is more effective.

Implications for Software Development

- The benefits of integration for toolkit development are available within the Eclipse framework, and I believe are obvious.
- The benefits come at a cost, however:
  - Eclipse creates applications, not web sites. Tools such HTML, CSS, XML and XSLT provide only peripheral assistance to application development.
  - The Eclipse framework operates within Java, but is not built on the more familiar Sun-Java AWT/Swing/Applets platforms, and will therefore need to be learned by most Java programmers.
  - Development of tools in this way requires a highly professional attitude to software development, that might go beyond the resources available to many in the humanities.

Acknowledgements

- *Pliny*'s development was made possible by the provision of research leave for myself at **King's College London**.
- I am deeply grateful to KCL, and in particular to the head of CCH, **Harold Short**, for making this possible.
- I am also grateful to **Willard McCarty** with whom I have spoken from time to time about scholarly research and practice, and who has given me many useful insights.