When WordHoard met Pliny: Annotation, context and application.

John Bradley, Senior Lecturer, Department of the Digital Humanities,
King's College London
john.bradley@kcl.ac.uk

One characteristic of new technology is that it takes time to understand all the new affordances the technology provides. The earliest printers tried first to produce books that looked as much like manuscripts as possible but later discovered that print had both possibilities and requirements that were not conceived of in the pre-print era. The digital revolution and particularly the internet has brought us the potential for transformation in communication, and we are perhaps now beginning to see some of this clearly. Goodness knows, we in the digital humanities (DH) are well aware that the new digital technologies in which we are engaged bring new things to the Humanities! However, it is possible – even likely – that we are still not seeing all the new kinds of potential that digital technology has opened to us.

My work has taken up issues around digital annotation – a topic that is of interest to a number of people in the digital humanities. In my view, almost all of the interest in digital annotation with our community has been from the perspective of the WWW, in particular in the context of Web 2.0: its public and social context. See, for example, Jane Hunter's excellent encyclopaedic overview of work on digital annotation in the web context (Hunter 2009). Indeed, Hunter directly acknowledges this focus in the "Scope and Definitions" part of her paper where she states that she has placed this work directly at the centre of where much of the recent thinking on annotation has been: the WWW, and she therefore focuses on the potential of the internet to enable annotation as a collaborative and social activity.

Much of our understanding of annotation within the WWW has grown out of work in the context of web-accessible digital libraries. For this, the highly influential work of Maristella Agosti and colleagues and, in particular, her seminal work on a formal definition of annotation as presented in Agosti and Ferro 2007 has been important. This work, in turn, has influenced the Open Annotation Collaboration project, an initiative which intends to "facilitate to emergence of a Web and resource-centric interoperable annotation environment" (OAC 2011, front web page). Here again, the thinking about annotation has been driven by the concerns of the World Wide Web, and therefore assumes that all objects that it supports for annotation will be web-accessible and web-based objects.

This way of viewing annotation – in the light of the WWW – is seductive not only because of the pervasive nature of the WWW in our thinking about digital things, but also because the continuing document-oriented nature of much of the web. As this paper will hopefully reveal, this document-orientation happens to fit well with characteristics of pre-digital technologies such as print, and means that we don't see other aspects of digital objects that are *not* shared by pre-digital ones, and which, as a consequence are barely explored through the lens of the WWW. Furthermore, I believe that, even within the web-centric perspective of software developers in the DH, certain assumptions about the nature of digital things on the WWW are changing: in particular the shift in thinking of the WWW as the deliverer of resources to the deliverer of applications. However, our focus, so far, on the document-centred WWW

and annotation in this context limits our understanding of the potential of, and the issues that arise from, annotation, and, perhaps even of digital objects more generally.

I intend in this paper, then, to encourage a somewhat broader perspective, derived from my work on the *Pliny* project, and to work on the significance of digital annotation that is at least a bit outside conventional WWW digital world view.[1] To the extent that the web-oriented DH development community is thinking about the still-emerging more interactive- and application-oriented WWW environments such as those enabled by HTML5 and AJAX, perhaps it will have useful things to say to them as well.

Pliny as an environment for personal annotation

Pliny (2009) was software written to explore some of the new potential for annotation in the digital world and was created to focus attention on the potential role for computing in supporting not social scholarly interaction, but personal research. It is, thus, necessary first to understand that Pliny is based on a different set of assumptions about the role of annotation in scholarship from pretty well all of the annotation-oriented WWW-based work. Indeed, my original intention with Pliny was to remind the DH development community that personal, rather than collaborative/shared, annotation taps into some fundamental elements of humanities scholarship. It too was worthy of study by the DH community, rather than being simply ignored as a result of the focus on the significance of collaboration that online-scholarship makes possible.

What is meant, within Pliny, about annotation for personal research? The primary starting point for understanding annotation there is to think about traditional pre-digital annotation: writing by a reader put into to a printed text for the purpose of enriching the reader's experience of reading that text. Pliny is, in fact, derived from thinking about what writing in a book is for, and to explore how doing this kind of annotation in a digital instead of print context affects or enhances this goal or purpose. At first glance one might think that, after all, "annotation is annotation" – that all forms of annotation share the same base principles and that there is no need for something different – at least at the technical level – for personal and public/shared annotation. However, there has been research done in computer science that suggests differently. See Marshall 1998 for some early, but still insightful, observations about different kinds of annotation, and some of the significance of the differences (described as "dimensions of annotation") – in particular the dimensions of "published vs. private" and "Global vs. institutional vs. workgroup vs. personal" (p. 41), and further discussion on the distinction between private and public annotation, and what happens when going from private notes to public ones in Marshall and Brush 2004. Indeed, I believe that much of the Pliny-related work, as described in the original papers about Pliny (Bradley 2008 and 2008a), and extended in a particular direction in Bradley 2008b and further still in this paper, shows that there a rather fundamental differences between personal and web-oriented annotation that can transform much of how we think about how might best apply digital technology to support the activity.

Since much of the thinking about annotation, even in the Web 2.0 context, is derived from the long standing practice of annotation on paper, let us start there (see figure 1).

[1] Some of what is reported here grew out of work funded by the Mellon Foundation's MATC award for Pliny. Parts of it were first reported in a poster displayed at the DH2011 conference by Timothy Hill and myself. (Bradley and Hill 2011).

Most of the time annotation on paper is a personal activity – what Marshall would consider at the private end of her published *versus* private dimension. This kind of annotation acts as a central activity for many scholars (Brockman *et al*. 2001). But, what is this kind of private annotation for? Of course, at the moment in which readers writes annotations, they do it to enhance their immediate understanding and retention of the material that they are reading. Does it have any longer-lasting purpose or use? My conjecture (supported by, among others, Brockman *et al* 2001), and expressed in how Pliny works, is that in fact this kind of annotation, indeed notetaking more generally, provides one of the bases for much scholarly research in the humanities: that notetaking fits into the activity of developing a personal interpretation of the materials the reader is interested in. (see discussion of this with regard to existing Pliny work in Bradley 2008 pp 265-6, and Bradley 2008a, section "So, what is humanities research, really?").

Thus, when the book reader writes a note on the paper s/he creates a situation where two rather different applications must co-exist on the page: the print media represented by the printed word and his/her annotation shown by the handwritten note. The owner, the technology and purpose of these two co-existing texts – the annotation and the print material – are quite different. Furthermore, there is a temporal side to this: whereas the printed text represents an endpoint in the "publishing application" that put it there, the hand-written annotation represents the beginning of an act of interpretation that is likely to continue into the future. When the reader writes something in a book, she or he intends to use this note in the process of developing his/her own ideas about the material that will continue after the writing of the note is over.

In some senses, then, a printed page with an annotation on it represents a *nexus* between these two quite different applications: (i) the presentation of the print, and (ii) the support for the annotation made by the individual reader. Although the annotation is on the same page as the print, it is quite a separate kind of thing from the print. Indeed, if handwritten annotation on a printed page worked in the way that many annotation services on websites operate – provided as a service of the book's publisher – it would in fact seem very peculiar, and, indeed perhaps strikingly inappropriate.

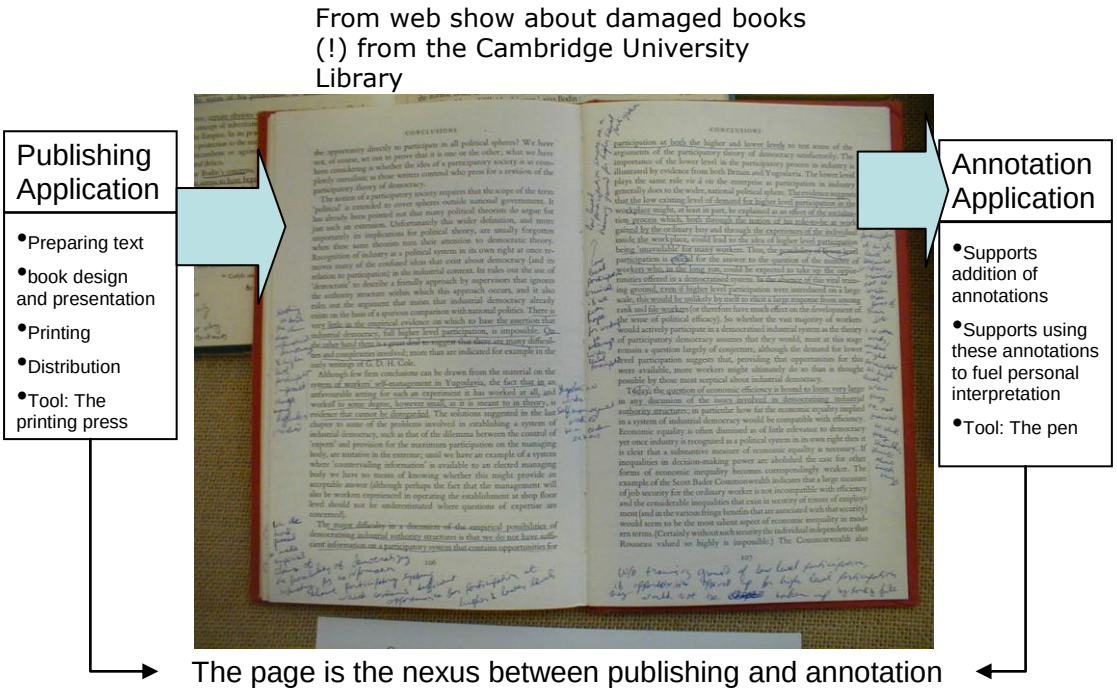

Figure 1: a printed page as the nexus between applications

Pliny, as initially installed, supports annotation for web pages, images and PDF documents. For each of these media types separate software components have been written which support, simultaneously, mechanisms to display the object (web page, image or PDF) and to support annotation of these objects. The annotation items, although initially appearing with the web or PDF page or image became also objects that work in the larger Pliny context as independent objects in their own right. Thus, in some ways like the printed book, the Pliny screen becomes the nexus between the *display application* of the image, web or PDF page and the separate-but-linked *annotation/notetaking application* (see Figure 2). Furthermore, the Eclipse platform (Eclipse 2011) in which Pliny operates already supports the dynamic addition of new components into an existing installation. Pliny could thus be relatively straightforwardly extended to add support for annotation for other media such as video or audio. The integration between these media and Pliny notes would be similar to that provided in base Pliny – annotations made on these media could also automatically fit into the separate interpretation development environment that Pliny supported.

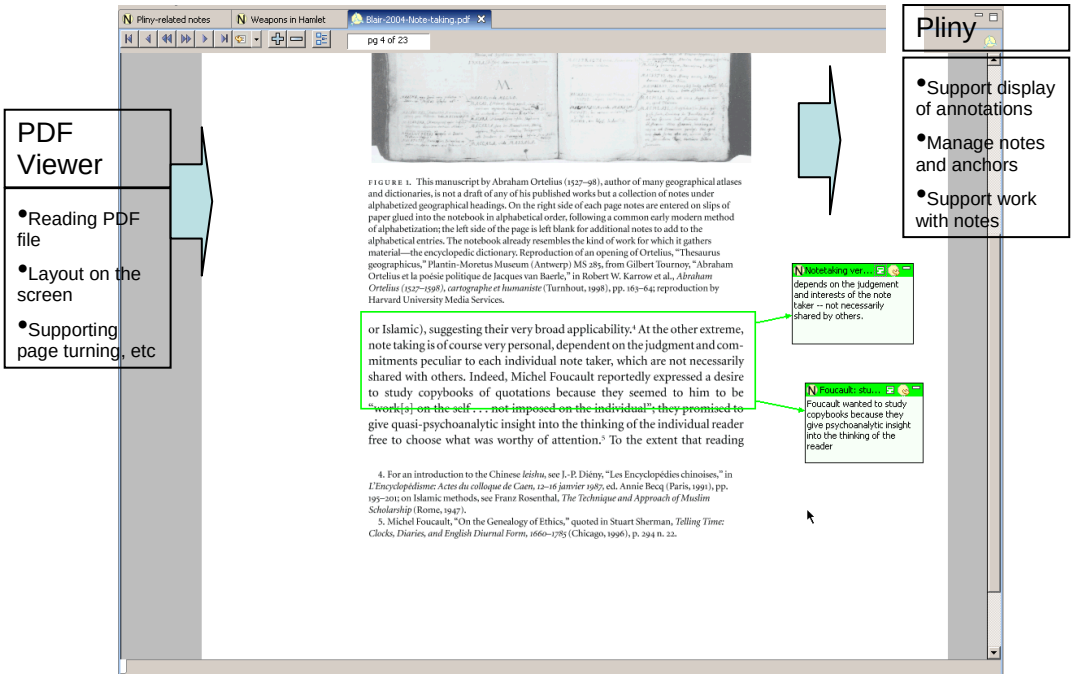

Figure 2: Pliny as the nexus between applications

Figures 1 and 2, then, emphasise the nexus nature of annotation on the printed and digital page, but don't adequately illustrate how these objects work within Pliny in the notetaking context (the application identified in the box to the top right in both figures 1 and 2). Figure 3 presents schematically a representation of the role of annotations in Pliny's more-general notetaking application: what I have described elsewhere as interpretation building. The material in figure 3 is organised into three areas. The annotations (shown in the left-most area) sit as transition points between the digital objects they annotate, and the digital model of their personal interpretation that the user builds in Pliny. This is where the "nexus" nature of annotation is represented.

The remaining two areas focus on the role of these annotations in notetaking and interpretation building. In the middle area we see someone using Pliny to discover and record concepts of interest to him/her. Although any real use of Pliny would likely result in many hundreds of concepts being identified and organised there, for the purposes of simplifying this diagram we only show two of them. Within each of these concept-objects, however, we see notes describing the concept, and links (through previously created annotations) to resources that relate to them. Finally, the third area to the right shows the user assembling the concepts and references to the original sources that have been stored in Pliny as s/he plans for two papers.

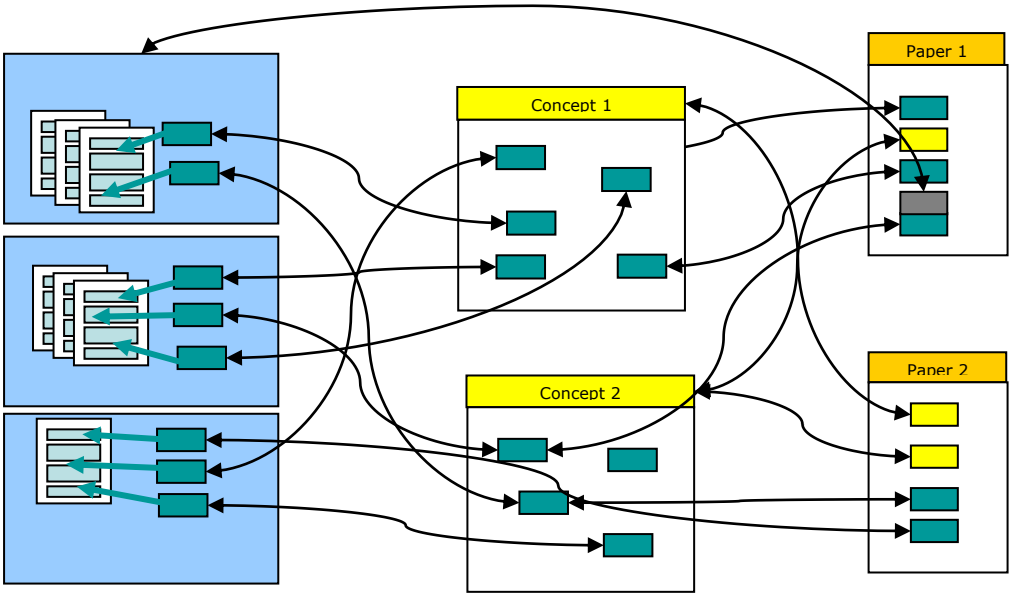

Figure 3: Pliny objects in its "notetaking" application

In the central area of figure 3 the reader makes connections that bind materials from diverse sources together in a way that reflects the reader's personal and particular interests. Note that the structure of these connections, although often bound to specific annotations in the material the researcher has worked with, take on their own structure that is quite different from the structure implied by a collection of notes in books. Although the task of interpretation started with the writing of annotations about what one is reading, its focus must shift in time towards the construction of objects that represent one's own interpretation, with its own, independent, structure and connections between its parts. The annotations still have a role in this, because they ground the interpretation in the sources that have been read – however, they operate now in the context of the reader's interpretation rather than the source's context. I have taken the liberty of calling this shift in significance of the annotations from their original target to having a role in the user's own emerging interpretation building as *a re-contextualising* of the notes. By showing the integration between annotation and interpretation development, Pliny draws our attention away from a focus on the building of annotation components added to, say, websites that support shared annotation, and towards the purpose that drives most acts of annotation in the first place: to support personal scholarship by (a) recording original thoughts (as original annotations) that arise in the mind of the reader as these objects are studied, and then (b) by supporting a way of incorporating these thoughts into a structure of interpretation that will almost always incorporate personal insights with references to ideas that arose from a range of separate documents.

Note, as well, that the various objects shown in figure 3 form a web of connections that to some extent tracks the web of connections in the Pliny user's mind as she creates the various objects represented there. Pliny, then, provides a kind of glue that connects references to documents of various media to the user's own set of ideas that are also stored as a network.

Annotation in the context of Applications

As is perhaps clear by now, Pliny is not a website, but an *application* that runs on its user's machine. This allows it to be more flexible about the kinds and range of resources it can work with, and (by not being itself served from the web) allows these materials from different resources and scattered across different places on the Internet to be brought together, including even personal objects not served over the internet at all. Furthermore, being an application that someone runs on their own machine emphasises its personal nature, and clearly reflects the personal ownership of any personal annotations its user creates.

Although Pliny is a software application, it is built on top of the Eclipse framework which provides a conceptual model for application development that is particularly well suited to the development of collaborating peer related components such as what is implied in the "nexus" understanding that I have just described. This is because the Eclipse framework has a richer understanding of software modularity than one finds in other conventional Java frameworks such as Swing, or, indeed in other non-Java environments too. With conventional Java applications a developer can indeed include components that come from other developers – a central idea of software modularity. Database engines like MySQL or XSLT transformation tools like Saxon are examples of software developed by one team of people, but often used by other projects as building blocks for their own application, even though they are then components that disappear inside this larger packaging. The developers in my department, for example, use MySQL in our Prosopography of Anglo-Saxon England project, but MySQL's use inside PASE is virtually invisible to the PASE user. Thus, the main application like PASE's becomes a "Borg application", reusing software development work from others as a way to efficiently implement aspects of the software that they need. Like the Borg on *Startrek* the enveloping software projects take over these applications to serve their needs, but then hide them inside their own packaging. Although the master project becomes a big tent containing many different components that help support it, from the user's point of view these components have been swallowed up, and users will only see the enveloping application as the thing they are using.

Not all modular software development operates in a way that hides the modules. The need for different applications to share a workspace so that they can all interact on their shared data is common in data- and text-mining toolkits, and the approach used there is often characterised as a kind of modularity called the *data-flow* model. One uses the data flow approach by connecting separate tools together – the data being processing is first passed into one tool which transforms it in some way and generates output that is passed (flowed) as input into the second, and so forth. Although data-flow does, indeed, enable a framework where different pieces of software can co-exist and remain evident to the user, this paradigm is insufficient for annotation, since annotations have not so much the need to share data that they "process" (what data-flow enables), as to share the screen with the materials that they decorate. The sharing of the screen as well as the data makes the nature of their co-operation of necessity more intimate than what the data-flow model imagines.

Having drawn our attention to the intimate nature of the interaction between components in Pliny's annotation framework, look at figure 4, which redrafts the

ideas in figure 3 into an application-oriented perspective. Here, the different applications (browser, PDF viewer, WordHoard and an application called "A") operate as peers – each visible to the user and clearly providing different and complementary functions for him/her. Furthermore, the yellow boxes – which represent the annotations, sit at the boundary (by sharing the screen) and are hence shown here as sitting *between* the application in which they are displayed, and the Pliny framework in which they are stored. This ability to combine data from two different applications on a resource as intimate as a computer screen window is uniquely made possible through Eclipse's software environment in which Pliny is built.

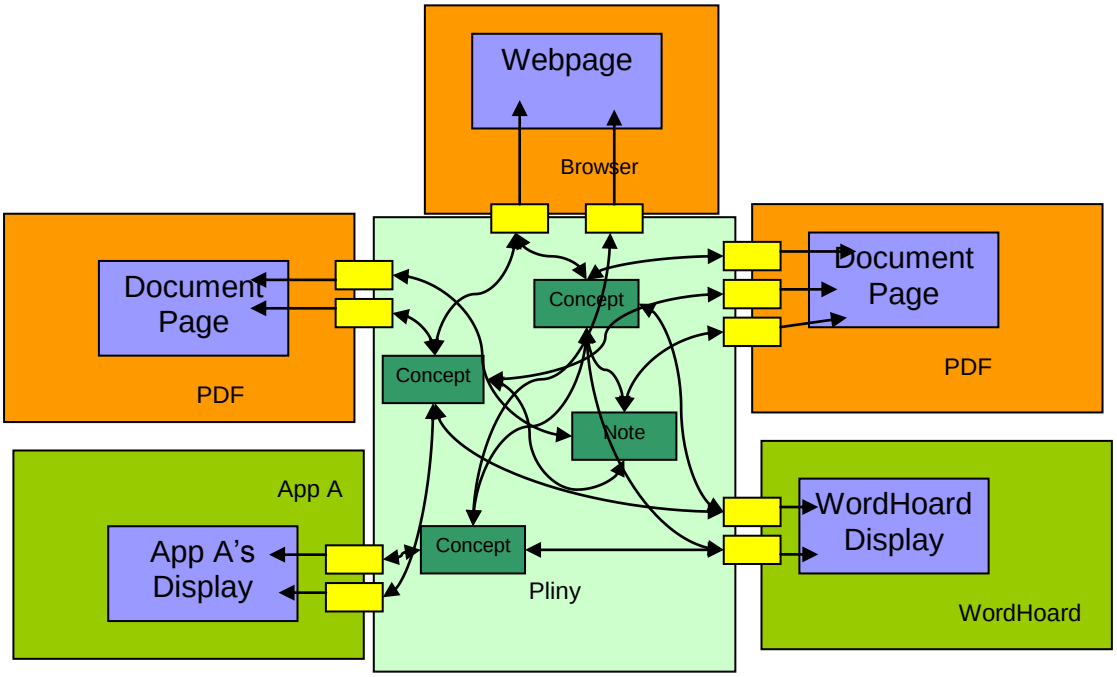

Figure 4: Pliny notes glue together separate applications

The top three applications in figure 4 (shown here in orange and already incorporated into basic Pliny) simply present PDF and web pages. They are really media players, but, by being "Pliny aware", are also able to support personal annotation (the small yellow boxes) of the different media they present. Once we notice the *application nature* of these components, however, we are in a position to take the application idea still further. We have already pointed out that Pliny is extendable to support annotation to different media by the addition of new Pliny-aware applications that displayed these other media. However, applications are not always simply media-players. Thus, Pliny's support for annotation did not need to be limited to relatively static *media* objects such as web pages or PDF files or digital video, but could extend to the displays generated by potentially more complex, interactive, and independently developed applications, as long as the developer of each of these applications wrote it in such a way that it was Pliny-aware.

This is implied in figure 4, where the two Pliny-aware software applications (an imaginary *App A*, and the real *WordHoard*) support Pliny annotation too. They are not simply pieces of software to display media files. Instead, they represent dynamic applications that the user uses to explore dynamically other kinds of data. In this way

of thinking, any display that these applications generated from their data could also be annotated, and these annotations that are attached to these displays could also be integrated into the user's set of ideas that are represented within Pliny. We have, then, Pliny acting not only as a tool to model a network of interconnections between media data (as suggested in figure 3), but also a tool to interconnect pretty well any kind of software applications as well – as long as it is written to accommodate Pliny annotation.

Exploring annotation beyond Media

Although application-thinking recognises that not all applications that might support annotation need to be merely presenters of media, digital annotation has been almost universally thought of as an activity connecting things to parts of media files. The reason for this orientation, of course, might well be that thinking about annotation has come from thinking about annotation on paper, and paper is a kind of media. Furthermore, much of the thinking about annotation has grown out of the digital library and WWW research community, where the objects of interest have been almost exclusively media-oriented "documents", rather than as a more diverse set of digital objects that can actually be represented in software. Indeed, much of WWW terminology, centered as it is still primarily on the conception of the web being made up of a large collection of documents, encourages one to recognise only media-like digital objects as the kind of objects involved in things like annotation. One sees this assumption everywhere. Note the definition, for example, in OntoText's widely quoted glossary of definitions of terms related to ontologies as "a form of meta-data attached to a particular section of document content" (OntoText 2011), where "document" is evidently thought of as a kind of media object – and this from a company that is working with ontology technologies that themselves are emphatically *not* document-like in nature. We see it again in the largely unconscious use of the word "media" as the things that might be annotated in the Open Annotations Collaboration's data model's *Guiding Principles* (Sanderson and Van de Sompel 2011, section 2). Even Agosti's formal model of annotation mentioned earlier seems to suffer from this kind of orientation, since her formal model builds towards its definition of annotation through a definition of a data stream (Agosti and Ferro 2007, section 6.2) and a segment in the stream (section 6.3) to the point where the anchoring point is defined as a segment of a stream (section 9). This "stream" view of digital data seems to me to be clearly one that is derived from a media-oriented orientation.

Viewing annotation as an activity that connects material from separate *applications* rather than *media* together is a more general one, and a better fit to the fuller potential of digital technology than the more static *media* perspective. It has the potential of liberating us from confining our thinking to things that are conventionally rendered over the WWW: largely static objects such as textual documents, images, 3D objects and even video and audio, and opens our thinking to deal with annotation in the context of the application-oriented perspective of the WWW that is, I think, still emerging. Indeed, this shift in thinking is in line with what is clearly a current trend in the digital humanities: towards thinking of the web as a place where applications (things like, say, text analysis, textual data mining or network analysis) can work on materials rather than merely presenting them. These tools when delivered over the web also do not exhibit a kind of "media orientation", and assumptions such as those mentioned above about annotation do not serve them well either.

The Mellon MATC prize allowed the idea of annotation in a broader application-oriented context to be explored within Pliny. Was the integration of a complex application, with Pliny to handle notetaking within that application, really practical? How did the act of supporting annotation in the Pliny context affect how the application had to be written? What, if any, were the technical constraints under which such an application would have to be written if it was to support personal annotation, and how onerous was it for developers to meet them? Before we started planning to try out Pliny integration with a large application we had already explored the development of small applications that co-operated with Pliny as test cases. We built a small application, for example, to allow someone to annotate a GoogleMap, and we did another to work with images from the image archive provided through the Victoria and Albert's public API (http://www.vam.ac.uk/api). Both these applications implemented parts of the Pliny approach to annotation handling, and allowed Google Maps, or collections of V&A materials, to integrate in the "Pliny way" with other materials; exactly as suggested by figure 4. However, as experiments, these applications were really "toy" applications: pieces of software that were rudimentary in nature, and hence both relatively small and based on only a subset of the full potential of the mechanisms which they might have exploited. Could this idea really work when the application was more complex?

WordHoard with Pliny

I was aware of Martin Mueller and Northwestern University's *WordHoard* project (WordHoard 2004-11) before the MATC aware had been granted, and had wanted even then to try out integrating WordHoard with Pliny. Here was software that, instead of running as a web application in a browser, ran as a Java application. Its orientation towards allowing the user to browse and search documents, and to perform various kinds of word-oriented analyses on them, plus its host of different kinds of presentations that could arise from this word-oriented work made it an excellent trial application for the views on annotation that had then emerged in my mind. Although WordHoard worked with text, it could not be thought of as a kind of media-presentation application in the way that a PDF viewer would be. Furthermore, it already supported annotation to some extent, albeit in a way that was, at least from a personal annotation perspective, more modest than what I wanted to explore with Pliny. As a result, I proposed to Mellon that the money that they had awarded for Pliny would fund a developer half-time for about two years to take the WordHoard code and gradually adapt it so that it could run in the context in which Pliny ran, and that could support the Pliny-supported annotation of its displays. Martin Mueller, and indeed the whole Northwestern development team, were happy with the idea and provided some guidance here and there although they were, naturally enough, unable to take part in the daily development work. I am thankful, however, for their generous support of the experiment.

The Mellon MATC funding has allowed us to explore this approach more substantially by applying the strategy used by Pliny as a real example of substantial integration between two independently developed substantial tools. The questions were:

1. How difficult was it to re-express WordHoard's user interface in this new Eclipse/plugin framework?

2. How difficult was it to integrate Pliny annotation into the user interface for WordHoard?

3. Pliny provides a broader context in which annotation operates than WordHoard does. Whereas annotation in standard WordHoard was thought of as a way to add commentary to WordHoard's texts that stays within WordHoard itself, Pliny annotation is thought of as a note-taking application that creeps into potentially all aspects of all the applications that integrate with it. Furthermore, the Pliny environment, with its potential for the re-contextualisation of its data (as described above) allows WordHoard objects to be referenced in contexts outside of WordHoard itself. How did this connection of WordHoard with Pliny change the way the WordHoard software had to operate?

4. Work is being done in by the larger Digital Humanities and other scholarly technologies community to think about individual annotation of webpages – see the OAC initiative, for example. By being based on Semantic Web technologies such as RDF (Lassila and Swick 1999) and URIs (Jacobs and Walsh 2004, Section 2), it is extendable into a range of media-oriented web-delivered objects, but it is not so clear how it fits with data which is not available as media through a browser/web frontend. What happens when the digital resource (like WordHoard) is not a web application?

This list of concerns, and of things learned from them is, of course, of interest to several different communities. In this article I focus on topics related to issues 3 and 4, but the other issues need to be discussed in the forums appropriate to them.

What was built

As it turned out, the task of building a complete version of WordHoard that co-existed with Pliny and allowed for the kind of intimacy of interaction implied in this article proved out to be a task that was too large for the funding provided for it. This was in part due to the challenge of getting our excellent Java programmers familiar enough with the Eclipse plugin way of doing things to allow them to be efficient in their development work – and this was compounded by the fact that our original programmer had to leave the project partway through, and as a result we had to change programmer midstream – requiring the training process to be carried out twice.

The initial aim was to, as much as possible, mirror the original WordHoard interface and integrate into it annotation components from Pliny, and we tried to do this by leaving the "business logic" part of WordHoard entirely alone – simply using the code that the Northwestern developers had written for it exactly as it stood, while re-expressing WordHoard's original interface in the new User Interface frameworks of Eclipse and Pliny so that we could add facilities for Pliny annotation to them.

Figure 5 shows part of the original WordHoard interface as created by Northwestern University, and Figure 6 shows our implementation of WordHoard working in the Pliny/Eclipse environment.

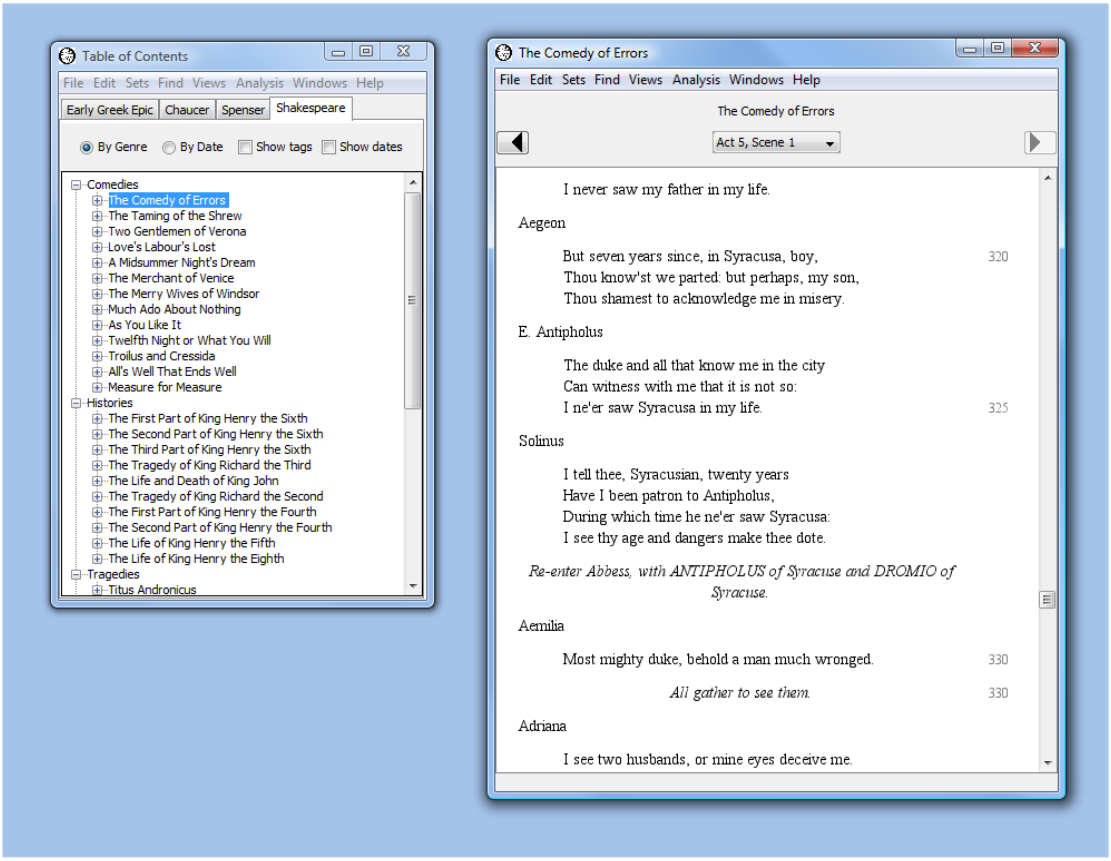

Figure 5: Original WordHoard Displays

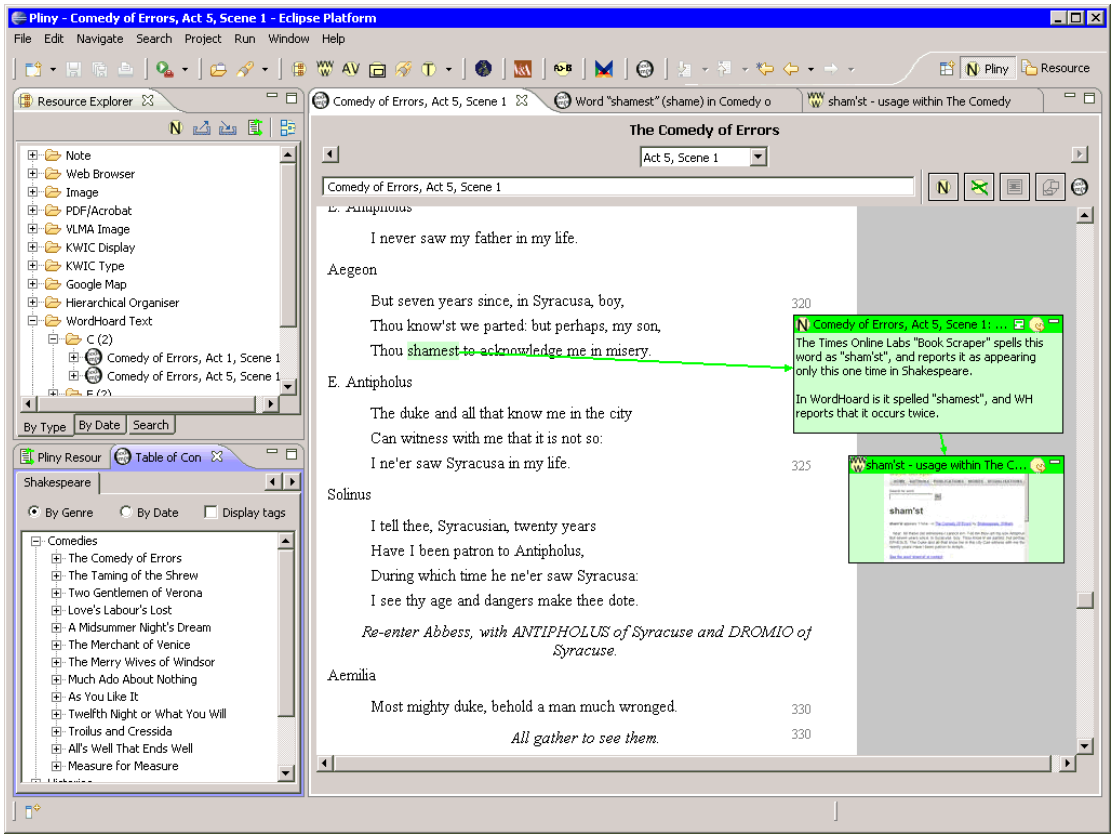

Figure 6: WordHoard in the Pliny environment.

Two of WordHoard's displays are visible in the screenshot of our Pliny-compatible interface for WordHoard shown in figure 6. In the lower left corner we can see the WordHoard "Table of Contents" display which, as in the original WordHoard application (shown in figure 5 in the left window), allows you to select a text from the various ones available through WordHoard. As one might expect from the Eclipse framework in which Pliny and our version of WordHoard was created, this display, along with the other displays we implemented, can co-exist with non-WordHoard materials. The top left area in Figure 6 shows, for example, Pliny's standard Resource Explorer. Annotations from different applications can share the screen too: here we see a reference to a web page placed as an annotation into the text display. Furthermore, in this screen shot the user has installed into their version of Pliny not only WordHoard, but also tools for annotating other objects: in this case, our prototype Google and V&A annotation tools. By having loaded the WordHoard, Google annotator and V&A annotation tools into Pliny, the user is now able to annotate not only Pliny's standard media objects (Web pages, PDF files and images), but can also annotate Google maps, material from WordHoard and images fetched through the Victoria and Albert Museum's public API (V&A undated), and use Pliny's re-contextualisation tools to bring them together.

Perhaps the most obvious place to start thinking about integration between WordHoard and Pliny was WordHoard's text display, where the texts within the WordHoard corpus can be viewed. Indeed, WordHoard itself supported an annotation component there already. We thought of Pliny's annotation paradigm as one centered on the provision of a 2-D space were notes can be laid out (there is a discussion of in what way this is central to the conception of Pliny in Bradley 2008 p. 271). Thus, for the text display – shown in figure 6 – we built the 2D space to tightly integrate with the display of the rather linear display of text itself which comes from WordHoard. Pliny annotation objects float in the 2 dimension space of the text area above WordHoard's linear text presentation so that the user could integrate Pliny objects in the same space that the text inhabits. This close integration between the linear presentation of the text and the 2D nature of the annotation makes it work in ways very similar to annotation on paper. Neither on paper, nor in our version of Pliny, is the user constrained by the way the text is displayed when deciding how to place their annotations. We think that the combining of WordHoard's text display with Pliny's 2D way of doing annotation, and the ability to anchor a Pliny object to a fragment of text, works quite well, and parallels to some significant extent the way in which annotation in books are actually done – closer, in fact, to what had been provided in the original WordHoard implementation.

An important part of our task with integrating Pliny into WordHoard centred around our recognition that a software user might want to annotation anything the application showed him/her: an idea that we have started to call the "annotation anything" principle. Not only the text display, but indeed all the displays that WordHoard could generate might create new ideas in the mind of the WordHoard user that he or she might want to annotate and record. Hence, we wanted to support annotation not only in WordHoard's text display, but in the other displays that it could generate too.

In the interest of software development expediency, we added Pliny annotation to the other WordHoard displays by means of a technically simpler approach than what we used in the text display, and provided an annotation *area* to the right of the main WordHoard display where notes and reference to other Pliny materials could be put as

a separate 2D space. The degree of integration between the annotations and the display that triggered them in the mind of the user was, then, substantially less than it had been for the text display, but it still at least allowed users to add comments about what they are seeing in the displays whenever they wished. You can see Pliny's annotation area for WordHoard's word information display on the right in figure 7.

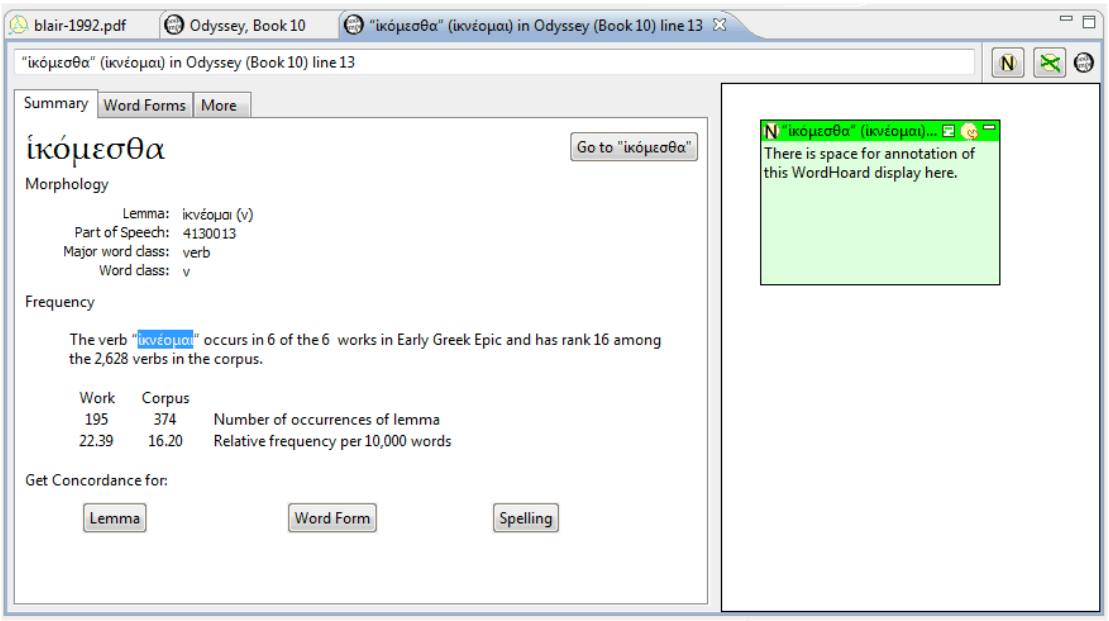

Figure 7: WordHoard word information display with Pliny annotation

We also did some work to render the WordHoard concordance tool with Pliny annotation support. Figure 8 shows a screenshot of the WordHoard concordance display centred on Shakespeare's use of the lemma "house".

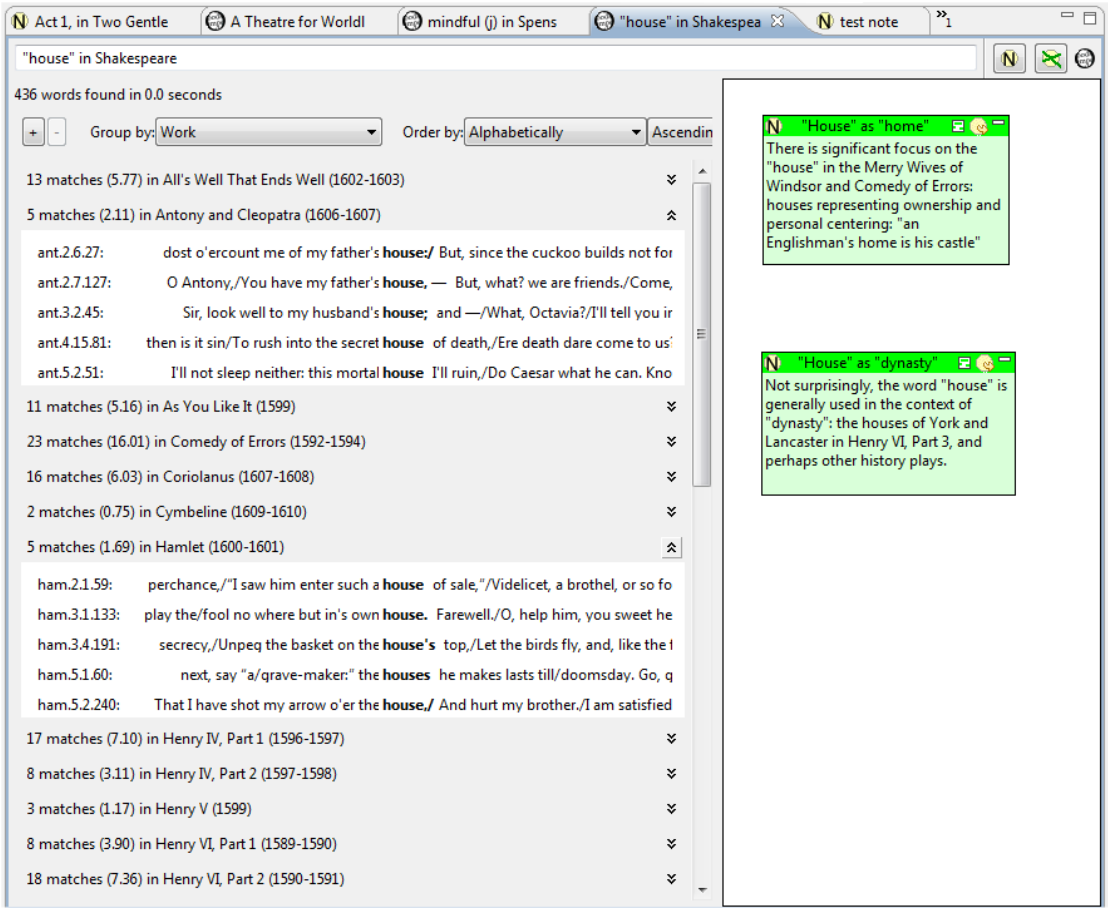

Figure 8: WordHoard Concordance display with Pliny annotations

In this case, perhaps because a KWIC concordance is both highly structured and can contain many entries, Northwestern's original developers chose to display it hierarchically. Within this display the user could choose among several different ways to group and order the KWIC entries, and could use the hierarchy to selectively open only certain of the categories thus created. In Figure 8, the user has chosen to group the lemmas by the work in which they appear, and we can see various Shakespeare plays with the number of occurrences of the word "house" displayed (gray background). Two notes have been added to the display already, but here we see the display at the moment that the user had chosen to open two of the work entries (*Anthony and Cleopatra* and *Hamlet*) so they can see the actual occurrences of *house* as items in a KWIC concordance (white background).

Annotation's standard model and its semantic deficit

Between Agnosti et al's 2007 formal definition of annotation, and the more recent and ongoing development of the Open Annotation Collaboration (OAC's) initiative we have an emerging understanding of annotation that I call here the "standard model". One thing that is perhaps surprising in this standard model is that it naturally incorporates what I am calling here a "semantic deficit" that is inherent in annotation on paper, and in media-oriented annotation as well.

Consider a situation when I am reading, say, Kierkegaard's *Fear and Trembling* in a printed copy, and I come across Kierkegaard's observation, and subsequent challenge,

to what he claims to be the commonly accepted statement "Only one who works gets bread". This passage triggers a personal reaction that I want to record. In my particular print edition this happens to be on page 57. So, I circle the part of the text on the page that talks about this and add a note beside it with my reaction.

There is, of course, a "semantic deficit" that arises here. I have written on page 57 of my copy to record a personal response to the issue that Kierkegaard is discussing, but I think of the link as being between my idea and Kierkegaard's point, not to the spot on this page in my book. There is a distinction between what I have done (made marks on page 57) and what I mean (comment on this part of Kierkegaard's discussion).

For personal annotations written into a printed book, this distinction does not really matter very much – page 57 in my copy always has been, and always will be about this proverb, and the fact is that when I see this annotation any time in the future I will think about its connection to the proverb rather than to where it is on the page. However, it is useful to note the incompleteness in the formal semantics from the reader's point of view. Someone talking about my note to a third party is more likely to say that it was linked to the place in the text where Kierkegaard is writing about this proverb, rather than "John linked his note to the text 7 centimeters from the top of page 57". Indeed, she might well say, "in my edition, this passage is on page 63", making it clear that she thinks of the link to the *idea* represented on page 57 in my copy, rather than to a spot on the page.

Indeed, in Pliny the method used to attach an annotation to an area in a PDF file works in a way that is very similarly to what happens on the printed page. The anchoring spot is recorded as an area on the printed page rather than linked to the objects the area on the page is showing. Indeed, Pliny is not aware of the link-to-the-idea at all. However, since PDF processors work hard to always place the text on the same place on the electronic page, this works fine. A PDF of *Fear and Trembling* that reproduced my edition of the text would always show this proverb on the same spot on page 57 too. Nonetheless, there is a kind of semantic gap between what is needed to allow Pliny to display the annotation, and what the user thinks of when he or she sees the anchor. Furthermore, almost no user viewing my annotations in this PDF file will notice the semantic gap because the context for display of the annotation (page 57, with all the text always displayed in the place) is always the same.

The issue is the same with media that operate in real time such as video or audio. Although we may not think of video or audio as "static", they both have a kind of fixeded-ness about them too. Take as an example the period of time from 19 minutes 45 seconds to 20 minutes in Stanley Kubrick's *2001*. This segment always represents the same sequence of image frames and therefore presents the same spot in the script no matter which digital version of the video one is looking at. From the viewer's perspective the same thing is always "going on", and if someone attaches a comment to *2001* for this time period it will always appear at the right semantic moment. However, the semantic deficit still exists: until we know what is going on in this time period we don't know what the comment might be about. As it turns out, 19.45 to 20.00 is the time when the prehistoric ape figure throws a bone into the air, and the scene abruptly shifts to the 21st century in near-Earth space. Although there is a semantic deficit between the specification of the annotation's anchor as an interval of time in the movie, the user seeing the comment in the context of the movie will not need to notice it.

This deficit between what is digitally recorded and what is "meant" is often implicit in thinking about digital annotation and, indeed, like the annotation to paper, often it does not matter given the fixed context in which the annotation will be displayed. However, anchoring the annotation not to a spot in a media file, but to something that is semantically meaningful reduces the deficit and improves the semantics of the anchor. Take an example outside of Pliny: from the OAC's *Hubble* example, where an image taken by the Hubble space telescope of deep space shows what appears to be a tightly packed area in space where there are many galaxies (http://www.openannotation.org/spec/beta/examples/hubble.html – section 2.7). All the annotation examples shown there identify this area in terms of an area on the computer file that contains the Hubble image. The semantic deficit here in these examples is that even though a different photograph of the same area of space might show the same apparent cluster of galaxies, the annotation's target for the Hubble image is specified only in terms of a particular image file and cannot automatically be transferred to these other images.

How could this semantic deficit be reduced, so that this comment about this area of space as shown from earth could be connected to all of them? An obvious answer would be to define the target of the comment to something that more satisfactorily links semantically to this region of space – perhaps the actual astronomical co-ordinates relative to the Celestial Sphere, for example. By attaching the annotation to something that connects to the "real world" rather than to an area in a piece of media that happened to capture it, we reduce the semantic deficit that separates the annotation from the thing being annotated, and improve the computer's ability to use the link in a more generalisable way.

The approach of locating a segment of a file as the anchor for an annotation works well enough for media-playing applications. even with the semantic deficit it entails. However, not all computer applications are media players. What might happen for them?

One of the potentially important differences is connected with the way that data that an application works with is represented. Most software developers call the set of digital objects that represent their program's data its *model*. Usually best development practice has the model objects kept separate from the surrogate objects that display the model's content on the screen. If one added annotation to an environment that had a backing data model, what would one formally attach the annotations to? One would expect annotations to link to objects in the model rather than the corresponding display surrogates. Since these model/anchor objects would represent the things the software is actually working with one might believe that this would end up reducing the "semantic deficit" that applies to the standard model of annotation. The model object the annotation attached to would indeed be much more likely to closely represent the thing a user thought the annotation was actually about.

This perspective of connecting annotation to an actual model representation of the object being discussed in the annotation does not appear to be present in either Agosti and Ferro or in the OAC data model. Surprisingly, however, Jane Hunter touches briefly on it in Hunter 2009. Even though Hunter's article focuses on "document annotation practices" (revealing from the start a strong document/resource/media orientation), she does note (page 4-1) that annotation could also mean something that is not attached to a *document about something*, but to the *something itself* when she recognises the meaning of "annotation" in computational biology – where annotation

is understood to actually attach information to particular "genes or proteins" themselves rather than merely to the pages describing them.

Separating Anchor from Target

In exactly the way that Hunter makes this distinction, Pliny annotation in WordHoard comes close to exploring *annotation of the thing itself* because WordHoard exhibits the classic approach to managing its data, with its data model representing the words in its text as linguistic objects. In WordHoard's data model, then, there exists instances of an entity called “word” that represent words in the text from a linguistic perspective. The word word *shamest* in line 322 from Act 5 Scene 1 in *Comedy of Errors* (shown as highlighted in Figure 6), for example, is an instance of the entity "Word" in WordHoard's model. Because we are working with model objects in WordHoard, we are in the position to actually attach comments to the things – the actual words – that the comments are about.

In a practical sense, digital annotation of model objects in a piece of software can only be done when the digital model object it is being attached to is accessible to the user – usually though some sort of display that the application can create. Here, however, we run up against the issue we mentioned briefly earlier that arises out of one of the basic principles of modern software applications – the separation of the “model” from the “display”. WordHoard’s words are a part of its model. However, any display of the model data that WordHoard can generate is built out of display surrogates for the model objects – graphical elements that can display on the screen and act as intermediaries between the data itself in the model and the window on the screen. This recognition of the separation between the model objects and its representation to the user is widely acknowledged among developers, and is a key element in the widely used Model-View-Controller (MVC) paradigm of software development.

This aspect of contemporary software design is relevant to our discussion here because it brings to the foreground the idea of “context” for the display of model data. In fact, displays for any particular piece of the model are generally created by displaying data from several closely-related pieces of the model, and thus the display is not the same thing as the entity it is displaying. Because the display of model data is separate from the model data itself it is possible to show the same piece of model data in different displays. Indeed, the three displays we have seen for WordHoard in the figures above all say something about its *word* entities such as shamest in Act 5 Scene 1 of *Errors*, but none of them are actually the same as the word itself.

- A WordHoard user can select a word in the text display. WordHoard recognises that the selection is a word and shows, at the bottom of the screen, a brief summary of the PoS for that word. Furthermore, the user can attach an annotation to that word, as we can see in figure 6, where a comment is attached to the word “shamest”.
- The WordHoard user can then request the display of the “Word Information Panel” which displays various kinds of linguistic information about the word. Figure 7 shows the Word Information display (but for a different word).
- The WordHoard user can select words for display in a KWIC concordance. Perhaps the KWIC display, similar to that shown in figure 8) would display the KWIC item for the word “shamest”. Although our current implementation of annotation in WordHoard’s KWIC concordance does not, in fact, allow an annotation to be attached to a particular KWIC instance, this could be added.

In these three cases the WordHoard word is presented in different contexts with, therefore, different kinds of information pulled from the data model and provided to the user. For the text display the user sees the word in the extended context of the text that surrounds it. For the Word Information Panel s/he sees it with its linguistic information, and with the concordance the word is displayed in the context of nearby words and other instances of the same word. Any annotation attached to the particular word such as "shamest", in any of these three display contexts is actually attached to the same semantic object. However the different context in which the word appears may cause the user to make quite different notes about it. Does it make sense, then, to say that the annotations are all attached to this same semantic thing? It wouldn't appear to be so since the things an annotator might want to say about a word in the context of the text in which it appears might well be different from what he or she would want to say when it appears (as it does in WordHoard's Word Information display) with its linguistic and morphological information presented about it.

Thinking about this a little more, then, the different contexts in which the WordHoard word appears can change the situation sufficiently to begin to affect the meaning of the annotation as well – for annotation purposes the meaningful anchor is not only dependent on the anchor object, but also the context in which it is displayed:

- In context of text display, an annotation could discuss word in textual context
- In context of Info display, it could discuss word and its morphology
- In context of Concordance: it could discuss the word in context of other usage of the same word in the text.
- On other hand, some of the annotations might be purely about the word in its own right (how it is spelled, say), and apply comfortably to all three contexts.

By planning to attach annotations to the WordHoard word itself we thought we had been reducing the "semantic deficit" that occurs when some text is used as an anchor for an annotation in a printed book. However, it turns out that for an annotation attached to a WordHoard word to have its full significance, the "word" as anchor for an annotation only a part of the story. The word's context in its display is seemingly also a significant element.

This clear distinction between the semantic anchor for an annotation and the context in which it appears is not explicitly made in the standard annotation model. Agosti and Ferro come close when they distinguish between a stream (as context) and a segment (as anchor) in that stream, but a stream segment is always a part of only one particular stream which contains it, and is thus always assumed to appear in the same context. The OAC data model does not recognise this situation well either, blurring the distinction between the display in which the annotated object appears and the portion of it to which it applies, and suggesting using of the W3C's Media Fragment specification (Sanderson and Van de Sompel 2011, section 3.7.2) to identify a portion of the target document where possible and what they call a "constrained target" if not (section 3.7.3). Furthermore, the developments in the Semantic Web have muddied the water here too: between a document *about* an object, a real-world object with its digital surrogate, and the context – a document – in which the surrogate appears. See, for example, Jeni Tennison's attempt to clarify URIs for documents *about* things versus URIs *for* real-world things in Tennison 2011.

Annotation in a dynamic display environment

And there is more to say yet. The discussion of the role of context in annotation should draw us back to the addition of annotation tools to WordHoard's concordance display (figure 8) because the visual display (the context for notes) can vary so much within that particular display. Recall that in this figure we see a screenshot of the WordHoard concordance display centred on Shakespeare's use of the lemma "house", with two KWIC displays opened by the user, for *Anthony and Cleopatra* and *Hamlet*). We can see immediate display context for the annotation playing a role when we consider the relationship between the two comments shown there, but perhaps added previously by the user, and how they read when looked at on this occasion when perhaps different works containing the "house" lemma concordance display are opened. Note that the two notes comment on the different meanings for "house" that Shakespeare has exploiting in the different plays. As the user opened and reviewed the different plays, these notes were added, and, although they were clearly revealed as a result of the user working with the KWIC entries, their appearance here, when KWIC entries for only two of the plays are visible, seems a little odd, since we can no longer directly see what motivated them.

The problem becomes even more evident if the user changes the ordering of the usages of "house". Figure 9 shows the beginning of the display now grouped and ordered by date of publication.

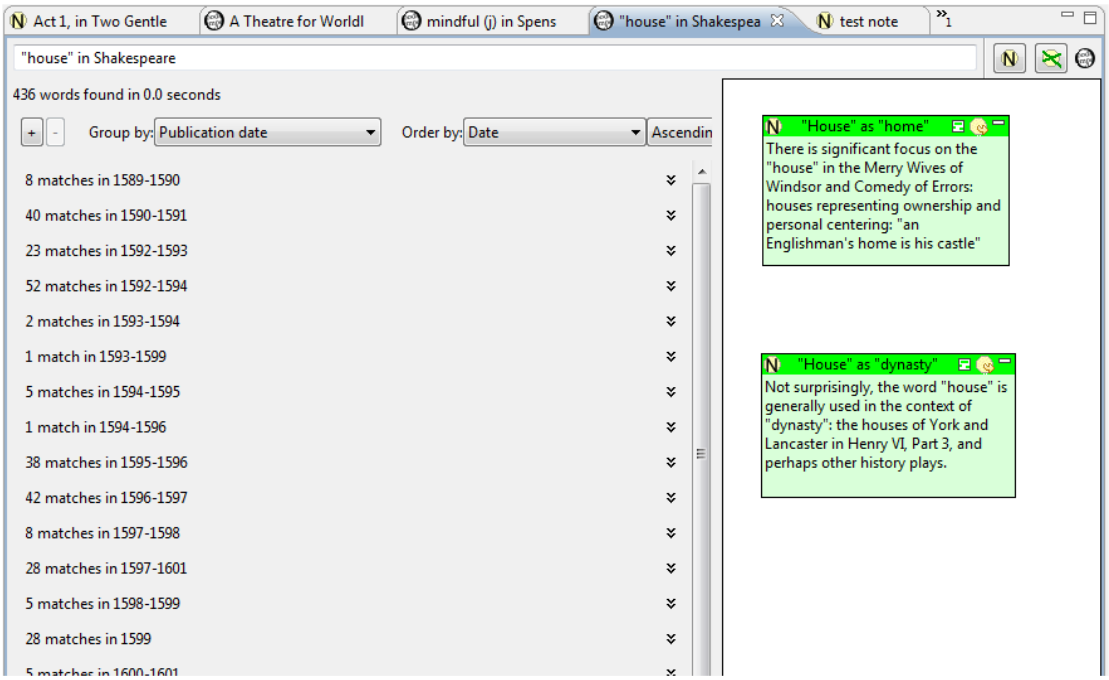

Figure 9: WordHoard concordance ordered by date

Here one can see the frequency of use of the word "house" varying substantially over the years, and indeed, perhaps this observation might be the cause for another Pliny note to be added to the two that are already there. Note, however, what has happened to the perceived relevance of the two pre-existing annotations made earlier when the concordance was ordered by the work. They are less evidently relevant when the concordance is ordered by Publication date. Similarly, any note made about Shakespeare's varying use of the word "house" over time – although relevant when

occurrences are displayed in this way – will seem out of place if the concordance is displayed again by Work.

The problem arose, of course, out of our assumption when building Pliny/WordHoard that the targets for annotation could simply be based on the particular display the user was looking at, and this error grew out of our experience of supporting annotation for Pliny's supported digital media; where this problem did not exist. There, thinking of an annotation as simply linked to a part of a display seemed to present no problems because the media files, such as image files, web pages, PDF documents, that drove the display were relatively static. The *context* in which the annotation was anchored (what comes before, what comes after, and what is actually visible) was always going to be the same. An annotation attached to a media file of this kind, then, gets some part of its meaning from the surrounding visual context in which it is displayed, and this surrounding context is always the same in media-like data. For this reason, the context for the annotation can be simply ignored in thinking about its formal representation. This was simply not true in the dynamic KWIC display that WordHoard provided.

Summary and Conclusions

The focus on digital libraries and on the World Wide Web has encouraged a view of digital objects primarily as documents, or perhaps more generally as media objects. Although much work has been done exploring environments for presenting these media objects and exploring various strategies for putting them together into larger objects such as collections, the focus on only media files as digital objects has constrained the thinking about the possible range of interactions and their semantics that a human being might have with them.

Digital annotation, as one of these kinds of interactions, has been thought of by many solely in this document/media context: primarily as an act carried out on static digital media formats such as those seen in a PDF or video file. This has been perhaps understandable, not only because of its origins in the media-oriented WWW, but also because the non-digital inspiration for digital annotation has mainly been the practice of annotation in printed (and therefore static) books. The publically available version of Pliny, in its basic distributed form, supported annotation for media objects, and did not seem to disturb the model of a "static" object being annotated either, even though, of course, a digital object needn't be simply a holder for static media content.

In spite of the strong presence of the WWW and media in our thinking about annotation, digital objects are not always pieces of media, and some of the strategies we have for exploring the potential of digital methods for the humanities do not suit them. Northwestern's *WordHoard* is a tool for supporting certain kinds of traditional scholarly activities – but its operation is not captured by thinking of it as a kind of media viewer. Instead, the user is given access to a set of mechanisms to explore the textual objects for themselves. Annotation can be useful in a tool like WordHoard since while using it s/he may well notice something which provides some new insight for which an annotation would be useful. Thus, annotation is a useful adjunct to what WordHoard does, even though annotation there does not fit comfortably with the digital media driven "standard model" of annotation. Although WordHoard is a particular application, it is certainly not the only application that cannot be well categorised as a piece of media display software. Tool kits such as text mining environments or network analysis environments (to take two examples of strategies in

vogue within the digital humanities at present) also do not work with a media orientation, and annotation frameworks that are media-oriented do not sit well with them either. Rethinking of annotation to encompass annotation outside of a media context is necessary to fit annotation properly into these new tools.

Perhaps if you are not an digital annotation enthusiast you are wondering why this issue might apply to you. Why should you care? Perhaps because annotation might have a place to play in the broader evangelical nature of the digital humanities – the desire by many in the DH community to promote the new tools as a new way to do the humanities. These new tools such as text mining often have seemed to be a hard sell in the humanities, and proponents of them have often found that traditionally-oriented researchers seem uninterested. Of course, trying to squeeze a scholarly interest "A" through a tool "B" which is manifestly not related to it provides, by itself, a good reason perhaps why sometimes these new tools have not penetrated the consciousness of mainstream humanities scholars. However, one can sometimes find situations where even if a particular new tool *does* seem to do something relevant to a scholar's interest, it still isn't being taken up. Perhaps this is because these more conservative colleagues don't see how to incorporate the results from these tools into the rest of their research.

Perhaps annotation helps in this situation, particular when done in the way that Pliny does it. If these new tools had Pliny annotation incorporated into them, it would be possible for humanists to use annotation to note things in the results from these new tools that struck them as interesting and to "bind" these results with what they are finding from their more conventional research work (arising from the reading of, say, a scholarly article presented in a PDF file format). If "App A" in figure 4 was one of these new tools, the ability of run it within the Pliny context would make it possible to integrate its results with material from other traditional digital and even non-digital sources. In this way one might move, through annotation, a step towards binding these tools within the framework of traditional humanities research practice.

Furthermore, this need to integrate new tool results with material from older scholarly practices does not only affect our more cautious colleagues who may be uncomfortable with these new research paradigms. Even a researcher who enthusiastically uses these new tools still needs to take the materials s/he finds there and to integrate them with references to traditional scholarship in order to present results to the public. Franco Moretti, for instance, developed his ideas about "distant reading" through using tools that treated his materials of study in highly original ways. However, he chose to present these ideas in the form of a narrative argument in the traditional way: through a printed book (e.g., Moretti 2005), and in that narrative he needed to combine results from his new way of doing research with material that came out of traditional methods, exhibited by references to mainstream scholarship. Annotation inside new digital tools such as those that support text mining, or the tools Moretti used, could provide a mechanism that allows these new research tools to better integrate with traditional scholarly practice – something even leading edge digital scholars still need to do as well.

In the work reported here, *WordHoard* was taken up as an representative of the new kind of tools. By supporting Pliny-style notetaking within *WordHoard* we allowed its user to both record something s/he has noticed in a *WordHoard* display, and then to integrate this with observations that arose from the conventional reading of other materials. By allows insights that arise in *WordHoard* to mix with insights developed

from traditional scholarship, Pliny allows *WordHoard* to integrate more readily into the traditional activities of scholarship. The scholar when writing an article could draw on notes that arose from insights that happened when s/he was using *WordHoard*, as well as when s/he was reading print or digital documents. In this way, annotation and notetaking become central acts both of traditional, print oriented or web, scholarship, but also as acts that can be associated comfortably with the newer, more dynamic, digital applications.

Pliny's working environment provides a powerful model for integration of not just media-presenting tools such as its (already existing) image and PDF annotation tools and potentially other media such as 3D, video or audio objects, but also as an environment which encourages newer, much more broadly conceived, applications to co-exist and even potentially interoperate in complex ways. Our work with the integration of *WordHoard* with Pliny both demonstrated the plausible, practical, nature of this kind of integration, but also revealed the need for some new thinking about the relationship between annotation and these more general, digital but non media-oriented, objects with which applications like *WordHoard* operate.

Acknowledgements

The work reported on here came from research that was funded in large part by **The Mellon Foundation** through their MATC award program. The author is particularly grateful for the recognition that made this work possible. I am also grateful to **Prof. Martin Mueller** and the team who developed WordHoard at Northwestern University for their interest and support for this experiment. Finally, I am very grateful for the work contributed by my two DDH colleagues who did a significant amount of the development work for the WordHoard/Pliny software, and in spite of the technical challenges it represented worked at it very professionally and with good insights and good spirit: Payman Labbaff first, and subsequently Timothy Hill.

Pliny's development and the thinking about annotation and notetaking that it represents was made possible by the provision of research leave for me at **King's College London**, and the continued provision of some research time after it was over. I am deeply grateful to KCL, and in particular to the head of (then) CCH (now DDH), **Harold Short**, and my colleague **Willard McCarty** for their support for this work over a number of years.

References

[Agnosti et al 2007] Agnosti, Maristella and Nicola Ferro. "A Formal Model of Annotations of Digital Content". *ACM Transactions on Information Systems*. Vol 26. No. 1 2007 Article 3, 57 pages. DOI=10.1145/1292591.1292594. Online at http://doi.acm/org/10.1145/1292591.1292594.

[Bradley 2008] Bradley, John. "Thinking about Interpretation: Pliny and Scholarship in the Humanities". In *Literary and Linguistic Computing* Vol. 23 No, 3, 2008, pp. 263-79. doi: 10.1093/llc/fqn021.

[Bradley 2008a] Bradley, John. "Pliny: A model for digital support of scholarship". In *Journal of Digital Information (JoDI).* Vol 9 No 1 2008 (formally No. 26). Online at http://journals.tdl.org/jodi/article/view/209/198.

[Bradley 2008b] Bradley, John. "Playing together: modular tools and Pliny". draft of paper, given at *DH 2008* (University of Oulu, Finland), June 2008. Online at http://pliny.cch.kcl.ac.uk/docs/oulu-paper.html

[Bradley and Hill 2011] Bradley, John and Timothy Hill. *When WordHoard met Pliny: breaking down of interaction silos between applications*. Poster presented at DH2011 conference, Stanford University, June 19-22, 2011. Draft available online at http://pliny.cch.kcl.ac.uk/docs/Stanford-Poster.pdf

[Brockman et al 2001] Brockman, William S., Laura Neumann, Carole L. Palmer, Tonyia J. Tidline. *Scholarly Work in the Humanities and the Evolving Information Environment, a report from the Council on Library and Information Resources* (Washington DC: Digital Library Federation, Council on Library and Information Resources). (2001_. Online version at http://www.diglib.org/pubs/dlf095/

[Eclipse 2011] *Eclipse* home website. At http://www.eclipse.org/

[Marshall 1998] Marshall, Catherine C. "Toward an ecology of hypertext annotation". In *Proceedings of HyperText 98.* Pittsburgh PA. ACM Press. pp. 40-49.

[Marshall and Bush 2004] Marshall, Catherine C and A.J. Bernheim Brush. "Exploring the relationship between personal and public annotations". In *Proceedings of JCDL '04 conference*. Tucson AZ. ACM Press. pp. 349-357

[Victoria and Albert (no date)]. *Victoria & Albert Museum API Documentation*. Available online at http://www.vam.ac.uk/api

[Hunter 2009] Hunter, Jane. "Collaborative semantic tagging and annotation systems". In *Annual Review of Information Science and Technology*. Vol 43, Issue 1 (2009). pp.1-84. DOI: 10.1002/aris.2009.1440430111. Online (account needed) at http://onlinelibrary.wiley.com/doi/10.1002/aris.2009.1440430111/full

[Lassila and Swick 1999] Lassila, Ora and Ralph R. Swick (eds). *Resource Description Framework (RDF) Model and Syntax Specification*. W3C (1999). Online at http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[Moretti 2005] Moretti, Franco. *Graphs, Maps, Trees: abstract models for a literary theory*. Verso Books. London. (2005).

[OAC 2011]. *Open Annotation Collaboration.* Website at http://www.openannotation.org/

[OntoText 2011]. *Glossary*. Webpage for OntoText AD corporation. At http://www.ontotext.com/kim/getting-started/glossary

[Pliny 2001]. *Pliny: A Note Manager*. At http://pliny.cch.kcl.ac.uk/

[Sanderson and Van de Sompel 2011] Sanderson, Robert and Herbert Van de Sompel (eds). *Open Annotation: Beta Data Model Guide* . Online at http://www.openannotation.org/spec/beta/.

[Tennison 2011] Tennison, Jeni. "What Do URIs Mean Anyway?". In blog *Jeni's Musings*. Online at http://www.jenitennison.com/blog/node/159

[WordHoard 2004-11] *WordHoard: An Application for the close reading and scholarly analysis of deeply tagged text*. Online at http://wordhoard.northwestern.edu/userman/index.html