

On Not Writing a Review About Mirador

Mirador, IIF, and the Epistemological Gains of Distributed Digital Scholarly Resources

Drs. Joris J. van Zundert

Huygens Institute for the History of the Netherlands — Royal Netherlands Academy of Arts and Sciences

Department of Literary Studies

Amsterdam, The Netherlands

<http://www.huygens.knaw.nl>

<http://jorisvanzundert.net/>)

joris.van.zundert@huygens.knaw.nl

Abstract

This piece mushroomed from a simple enough looking suggestion to write a review about Mirador (a viewer component for web based image resources). While playing around and testing Mirador however, a lot of questions started to emerge. Questions that in a scholarly sense were more significant than just the functional requirements of textual scholars and researchers of medieval sources for an image viewer. Questions that are forced upon us because of the way Mirador is built, and by the assumptions it thereby makes—or that its developers make—about its role and about the larger infrastructure for scholarly resources that it is supposed to be a part of. This again led to a number of epistemological issues in the realm of digital textual scholarship. And so, what was intended as a simple review resulted in a long read about Mirador, about its technological context, and about digital scholarly editions as distributed resources. The first part of my story gives a straight forward review like overview of Mirador. I then delve into the reasons that I think exist for the architectural nature of the majority of current digital scholarly editions, which still are mostly monolithic data silos. This in turn leads to some epistemological questions about digital scholarly editions. Subsequently I return to Mirador to investigate whether its architectural assumptions do answer to these epistemological issues. To estimate whether the

epistemological “promise” that Mirador’s architecture holds may be easily attained, I gauge what (technical) effort is associated with building a digital edition that actually utilizes Mirador. Integrating Mirador also implies adopting the emerging standard IIIF (international image interoperability framework). A discussion of this “standard-to-be” is therefore in order and subsequently the article considers the prospects of aligning the IIIF and TEI “standards” to further the creation of distributed digital scholarly editions.

Keywords

Mirador, IIIF, distributed knowledge, digital scholarly resources, information architecture, epistemology

Mirador at First Glance

Mirador (<http://projectmirador.org/>) is an open source, web based general purpose image viewer written in JavaScript. Rashmi Singhal of Harvard Arts & Humanities Research Computing (<http://darthcrimson.org/people/>, <https://github.com/rsinghal> on Github) and Drew Winget of Stanford University Libraries (<https://medium.com/@aeschylus>, <https://github.com/aeschylus> on Github) are the chief authors of the code, although some forty five additional developers contributed to the codebase according to Github (<https://github.com/ProjectMirador/mirador/graphs/contributors>). The development of Mirador was possible because of a grant from the Andrew W. Mellon Foundation to Stanford University. Creating Mirador has been—and still is—very much an open source and community effort, that also involves, for instance, well known scholarly open access advocates such as Robert Sanderson of the Getty Foundation and Loyola University of Maryland based Jeffrey Witt, known amongst other things for the Scholastic Commentaries and Texts Archive (<http://scta.info/>).

So, apart from an impressive open source community driven project, what does Mirador deliver? In what follows I examine and use Mirador 2.1.2. It should be noted that at the time of reviewing there was already a version 2.4, but this version was not yet compatible with some of the additional tools that I wanted to experiment with in connection to Mirador. The reader should thus take care to compare statements about Mirador with current developments, especially as meanwhile Mirador is preparing for a new life cycle and version 3.0 is expected to be released end of 2018.

If Mirador has been rigged up for you it actually provides a lot of image viewing capabilities

right out of the box. I will discuss Mirador primarily from a perspective of a textual scholar interested in transcribing historical manuscripts, but obviously the viewer is not limited to showing facsimiles. Because Mirador is agnostic to the kind of images one uses, it basically will tailor to the needs of any scholar wanting to study images of objects—historians, art historians, book historians, musicologists, literary researchers, and so forth. That is, at least, when two dimensional images will do. Mirador lacks any 3D viewing and rendering capabilities, so caters best to people wanting to study and compare paintings, facsimiles of manuscripts, images of typeset pages, photographs, music score, anything basically that was intended to lead its life on a 2D medium.

Mirador is highly configurable, and the view that greets the user may therefore vary. A typical initial setup, however, might not be unlike what is depicted in figure 1. Out of the box Mirador offers convenient panning and zooming, and some controls to adjust basic image aspects, such as rotation, contrast, saturation (cf. Figure 2.).

For the purposes of showing Mirador's capabilities I have opted to use a few folios from a digitized medieval Dutch manuscript, the so called Comburg Manuscript (Comburger Handschrift, Württembergische Landesbibliothek Stuttgart, shelfmark Cod.poet.et phil.fol.22). It contains a collection of some well known Middle Dutch works that have survived time. The folios shown contain the first columns of the Middle Dutch fable *Van den Vos Reynaerde* (transl. *Of Reynaert the Fox*), of which a full translation is also open access available in English (Bouwman and Besamusca 2009).

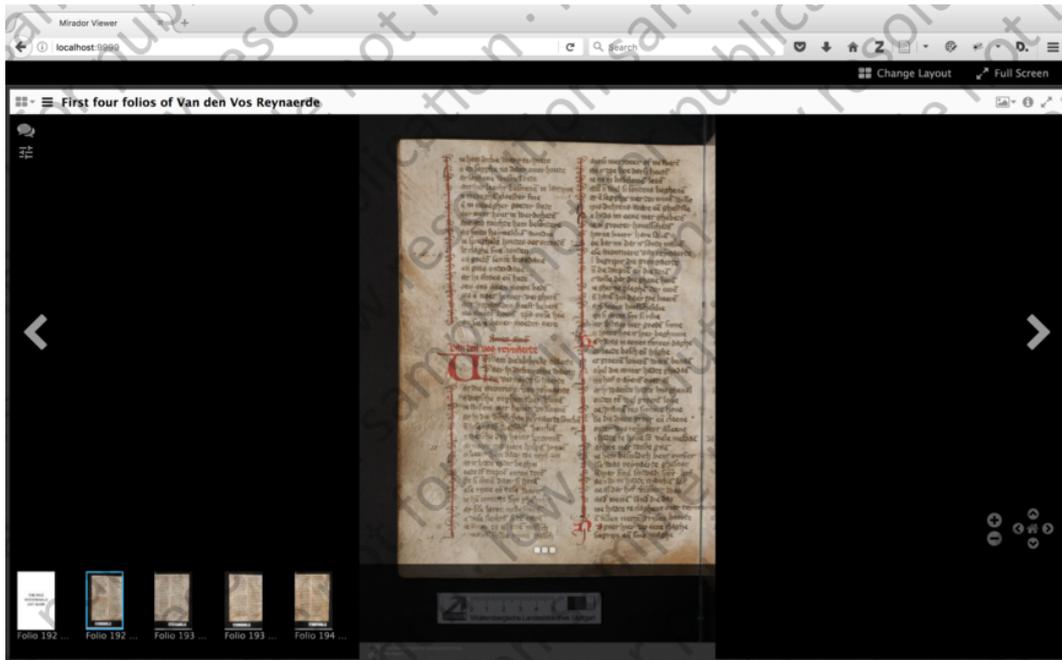


Figure 1: Mirador “Out of the Box”.

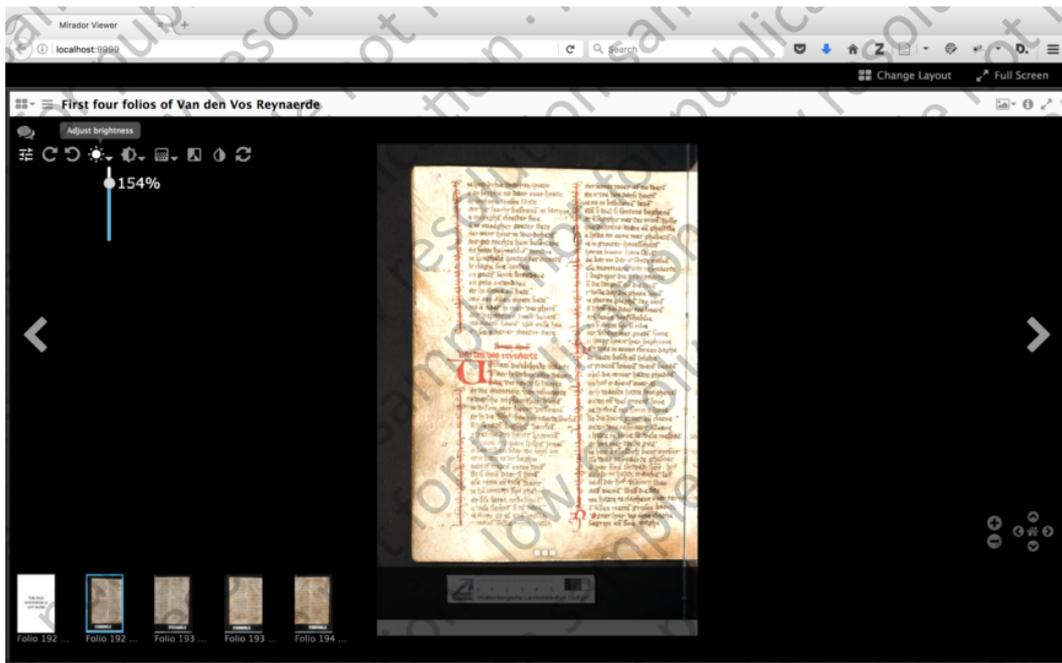


Figure 2: Item, with image controls expanded.

If a collection of images is being viewed also controls to leaf through the collection are provided: left and right pagers, and a thumbnail bar. The latter can be collapsed to safe screen estate for viewing the actual image (cf. Figure 3 and 4). In case a collection has properly added metadata Mirador will also instantly display an index/contents, if so

configured (Figure 5).

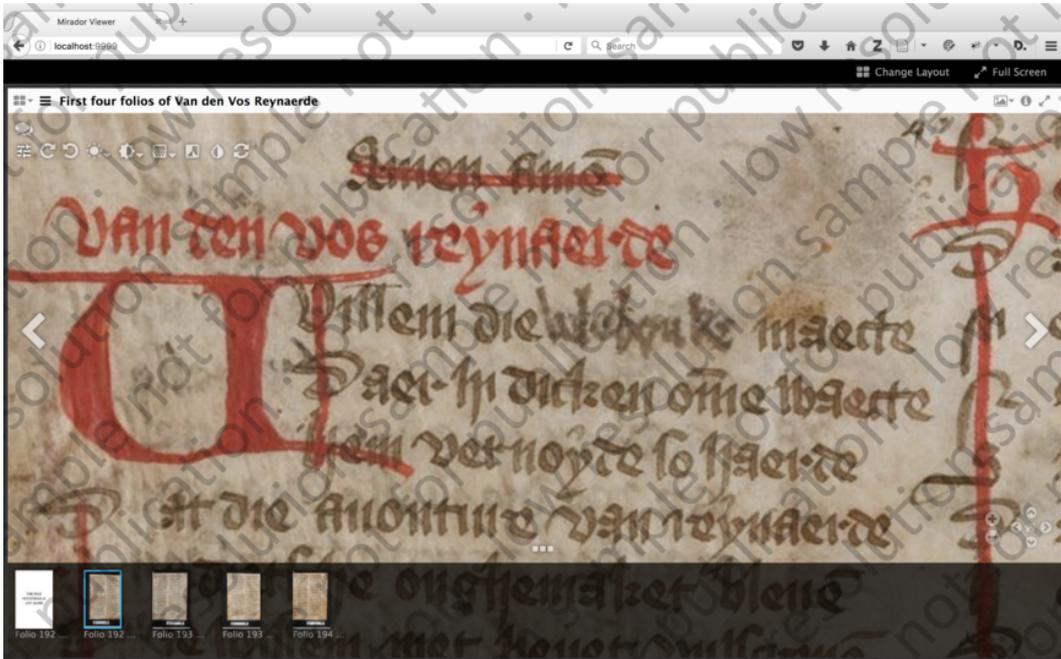


Figure 3: The same image, zoomed and panned. Image rendering controls expanded at the top left.

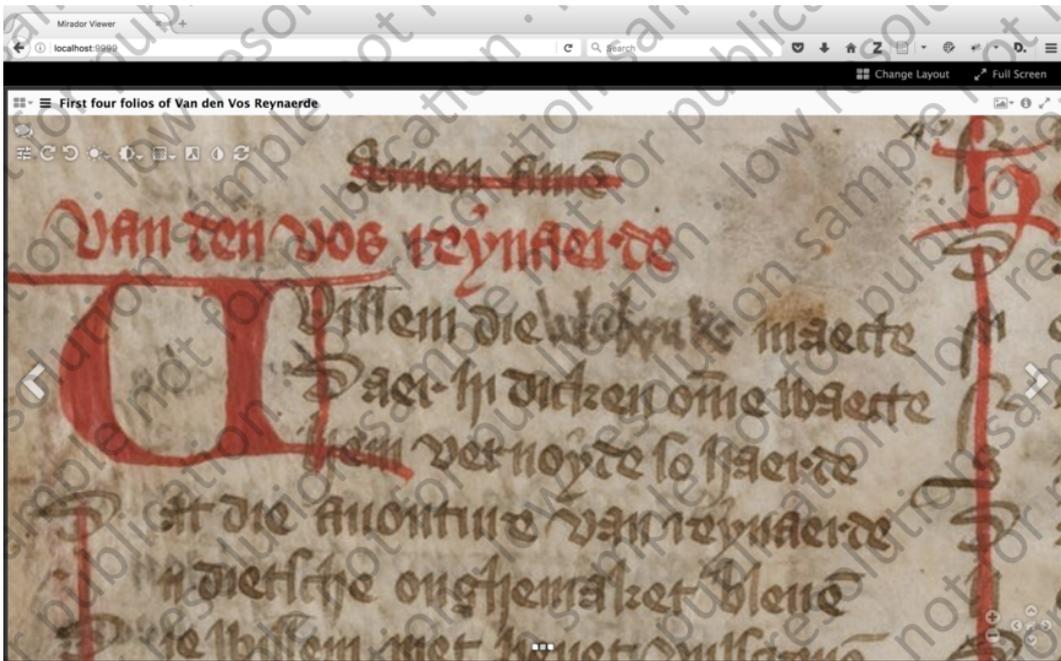


Figure 4: Same, with thumbnail strip collapsed.

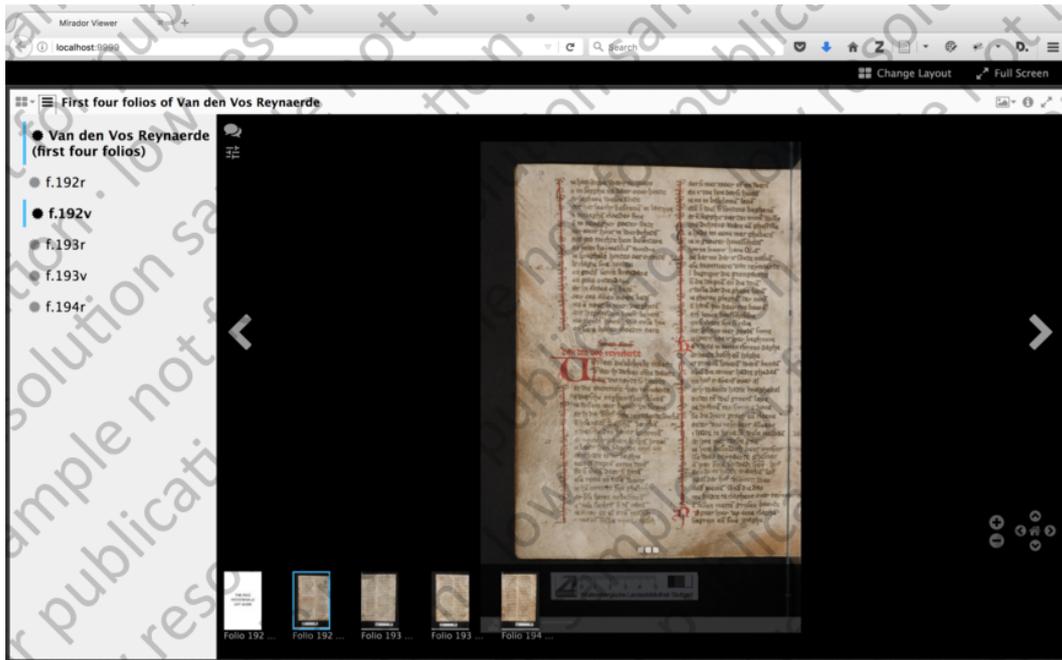


Figure 5: Mirador viewer with index visible.

Mirador's technical designers and developers ought to be commended for not suffering from the 'not invented here' syndrome. Instead of developing a completely new code base, they have maximally reused existing software components. This is a sensible development strategy as it lessens the development burden and prevents the reproduction of many bugs, technical pitfalls, and maintenance issues. It is also a development strategy that is in line with the nature of open source and community based software development.

Essentially then, Mirador wraps a number of pre-existing software libraries together and tries to turn that combination into a general purpose image viewer. I will defer judgement on whether that attempt was successful for now. First let us see what is in the box. Mirador encapsulates the OpenSeadragon (<https://openseadragon.github.io/>) viewer which delivers image viewing, zooming, and panning abilities. OpenSeadragon ensures a seamless and high quality viewing experience. Under the hood JQuery (providing GUI controls plus a score of general purpose code) and TinyMCE (a powerful 'rich text' editor) are code libraries that by now can be called part of the furniture in web development. A few other utility libraries are also encapsulated by Mirador. One reused component with a more central role is the Isfahan.js (<https://github.com/aeschylus/Isfahan>) window manager, which itself uses a tiny part of the well-known and currently very popular d3.js (<https://github.com/d3/d3>) data visualization library that powers many a flashy dendrogram, wobbly network graph, and shiny bar chart. This window manager is a quintessential cog of Mirador. A window manager

is the software component that allows you to open, resize, and move around windows on your computer's screen. Isfahan does the same for windows inside your browser. Mirador reuses Isfahan to allow the user to open and arrange an arbitrary amount of image viewers within the same browser window. This drives one of Mirador's self-proclaimed paramount features: the ability to compare images. Given that comparing codices and manuscripts is the bread and butter of textual scholarship there is utility in being able to put two, three, or more codices next to each other on one's screen. So, if your Mirador viewer instance is setup for a particular collection of manuscripts, additional "slots" can be added for viewing other folios in the collection (Figure 6).

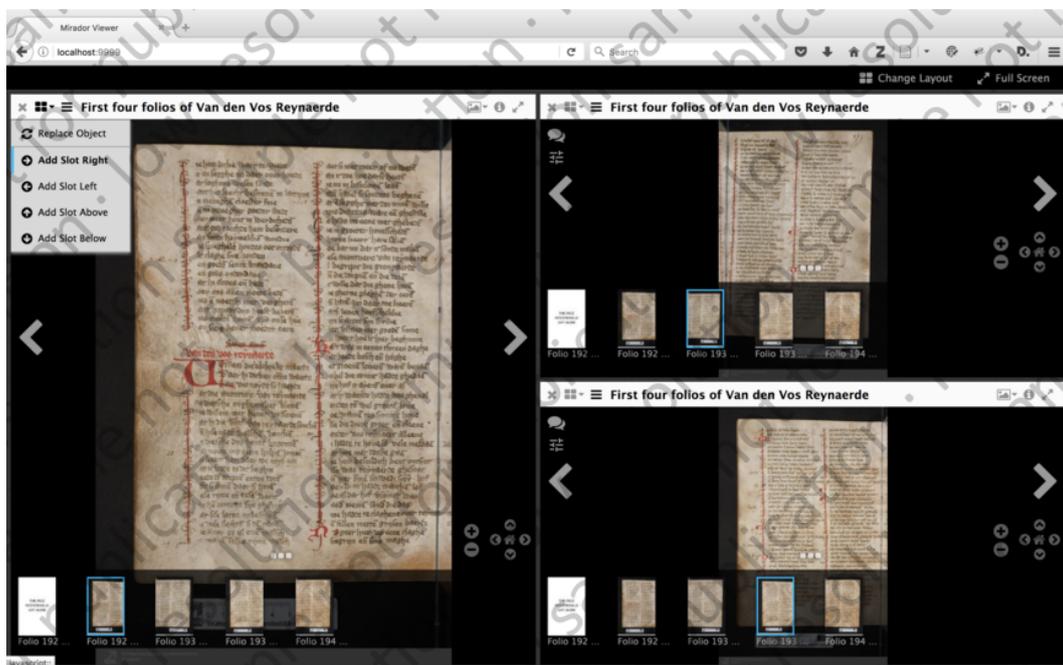


Figure 6: Multiple folios open for comparison.

Adding more and more views of manuscripts will of course pretty soon turn your workbench into an impressively cluttered graphical interface, even if you happen to have a 5120 by 2880 pixel monitor with 27" screen diagonal—which you probably do not as roughly half of that (1322 by 768 pixels on a 15" screen) is much more middle of the road currently. As Mirador is highly configurable the user may want to opt for another approach and have all that clutter of index buttons, paging buttons, thumbnails, and image properties controls taken out to attain, as the documentation (<http://projectmirador.org/docs/docs/getting-started.html>) has it, 'Zen mode' (Figure 7).

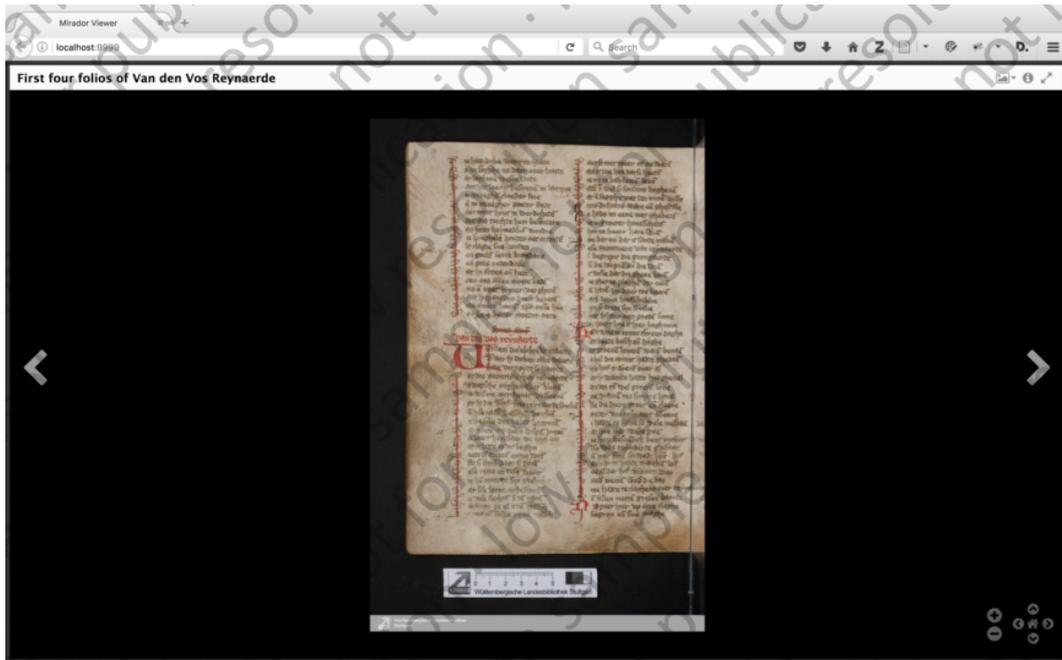


Figure 7: 'Zen mode'.

From a scholarly point of view all that viewing “power” at your finger tips is a dream of course. But I am a pretty spoiled scholar annex developer, and therefore I still miss a *locked scroll* or *parallel pan* feature which would scroll or pan both (or more) images that I am looking at simultaneously. This would be a feature that makes comparing lines a far less tedious task. (It should be noted that although this feature is not available ‘out of the box’ there is a plugin that exactly provides this behavior.)

Another indispensable, I would argue, feature of Mirador is the ability to annotate images, for which a convenient tool ships with Mirador by default. A scholar can draw a border around any arbitrary area and add annotation text for that area (Figure 8).

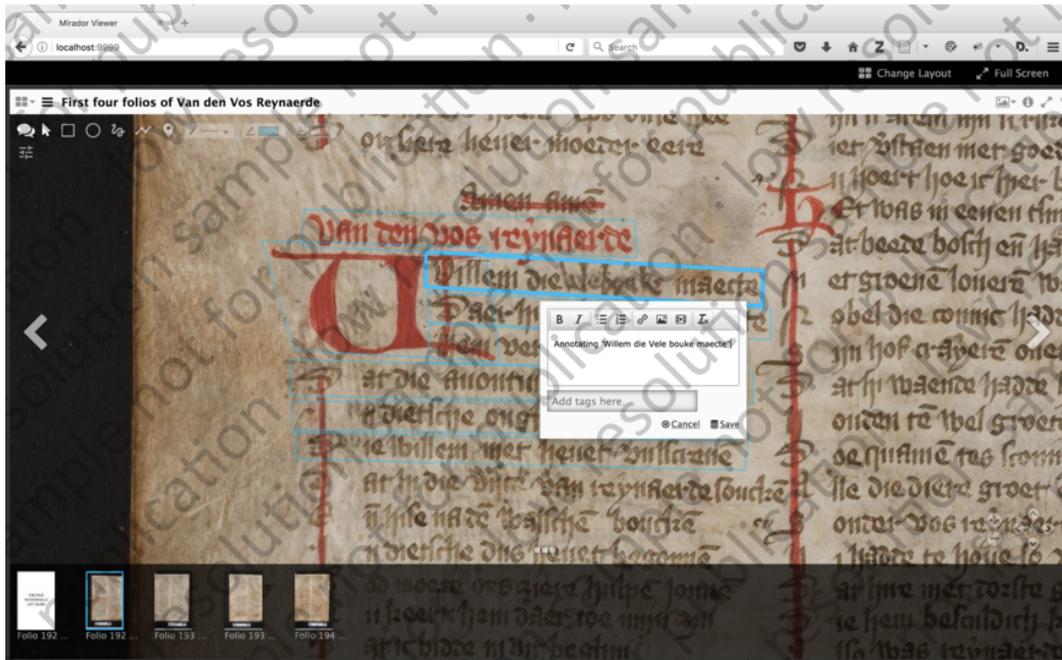


Figure 8: Annotation creation in Mirador.

In all then Mirador delivers out of the box an impressively well functioning and rich viewer and annotation tool. Panning and zooming is all rather nicely instant and seamlessly which makes for a comfortable viewing experience.

How not to write a review for Mirador

Of course I could have called it a day right there. I was to hand in this review by January 2017. I think I theoretically could have made that deadline. But the above still felt improperly incomplete as a review of Mirador. What I wrote there is not wrong—at least not as far as I can see—but it is also a far cry from the story of which I think Mirador is a chapter. I therefore decided to try to tell the grander story. And as it goes with thinking things like ‘How hard can it be?’ I never saw the end of it. Let me tell you about it...

Given the wrapping and encapsulating of ready made components that Mirador does one could critique it for being “just” a thin graphical user interface over a set of pre-baked code libraries, adding little of essence that could not have been achieved by several other means. However, first of all such a judgment would not be fair to the effort it must have taken to integrate all these libraries and functionalities. But more importantly: it would not acknowledge the pivotal role Mirador plays in what may be no less than a paradigmatic shift in how we understand, approach and interact with cultural heritage resources. Clarifying this

will take a little explaining.

The Persistence of Silos

The ultimate transcription environment has become somewhat of a holy grail within digital textual scholarship. Many attempts have been and are being made to create a transcription environment that surpasses any other and basically is even better than all other text editors, including MS Word (the surpassing of which arguably should not be too hard an achievement). Plenty of integrated transcription environment therefore exist. EPT (Kiernan, Dekhtyar, Jaromczyk, and Porter 2004), T-PEN (<http://www.t-pen.org>), TextGrid (<https://textgrid.de/>), eLaborate (<http://elaborate.huygens.knaw.nl/>), and CTE (<http://cte.oeaw.ac.at/>) are some of the ones that I know to exist or to have existed, and I know of several others being in a state of perpetual development ‘pre-beta’.

There is a graveyard somewhere too for scholarly transcription environments. But the failed attempts almost never get proper epitaphs or eulogies—the one for project Bamboo by Quinn Dombrowski (2014) being the notable exception—which is a pity because such eulogies would be highly informative. A defining trope of these eulogies would be that they would all state that the tool was built as an “integrated transcription environment”.

Integrated is a more formal term for “does it all”. These tools often want to be the start and end all of digital textual scholarship work. And most often this means that these tools want all resources to reside in one place (more specifically on the computer or server where the tool is deployed). The reasons for this are not technical, but institutional necessity and development convenience. The institutional make up of academia and its (grant) funding schemes favors local institution level digitization and development (Prescott 2016).

Collaborative development between institutions is often frustrated by funding limitations, and moreover requires significant additional coordination effort than local development. Lastly, from the point of view of the developer it is simply convenient to have all data in the same form and format and at an arm’s length. This is exactly the same as how convenient it is for a scholar to have all sources and secondary literature on her desk: it saves a lot of tedious logistics to gather and process information. The effect of convenience for institution and developer however is that these tools turn into what is called “data silos” (https://en.wikipedia.org/wiki/Information_silo): all images, texts, annotations, and so forth, need to reside on the same server to be used by the tool.

This has been a long standing problem in digital scholarship technology. As far back in 2007 when I and colleagues of mine started out on a project called Interedition

(<http://www.interedition.eu/>) the situation was almost perfectly the same: almost each institution, if not even each individual professor in textual scholarship was somehow involved in creating a large integrated all purpose research environment. This caused (and causes) a lot of reinventing wheels and duplication of effort, in a field that is notoriously understaffed as to digitally and computationally skilled scholars and developers. At the time it seemed to us—us being a mixed group of digital humanities developers, researchers, and any hybrid form in between—a good idea to rather reuse tools and resources instead of having local copies of text files and annotations, local tools, and the local burden of integration and graphical user interface development. This concern was not just on the developer side of things, to keep development loads small, it also seemed to us that keeping tools and data locked in one place behind a one-size-fits-all interface would be in stark contrast to the heterogeneous demands and requirements scholars had for document resources that were spread, quite literally, across the planet. What use would it be to have an alignment tool locally in Würzburg if one of the documents it needed to align was in the Bibliothèque Nationale in Paris? Our buzzword of the day became ‘interoperability’. The ideal was that no matter where you were you would have a local tool working just as easily on a digital manuscript facsimile in, for instance, Firenze as on a print edition in New York. We reasoned along the lines of service oriented architecture. That is: distributed resources would be reachable via the same technical protocol language, which would guarantee that any local interface speaking that access language would be able to approach and use them. In that way it would not matter if I would be using T-PEN and my colleague in Berlin would be using EVT (<https://visualizationtechnology.wordpress.com/>), we still would both be able to hook into the same resource in Stanford. In 2011 me and my colleague Peter Boot argued for such services based digital scholarly editions also in more academic fashion (Boot and Van Zundert 2011). It turned out however that between dream and reality stand institutional politics, and practical considerations.

Practical techniques and methods for decentralization resource reuse have been around for decades. Obviously the very protocol of the Web (i.e. HTTP, https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) is intended to make both local and remote resources reachable from one and the same location, that is in fact the whole point of the Web (Berners-Lee 1995) actually. Roy Fielding’s 2000 definition of a Service Oriented Architectural style (SOA) to interoperate digital resources on the Web called REST (https://en.wikipedia.org/wiki/Representational_state_transfer), did much to facilitate the creation of lightweight, small, and easy maintainable web services, based on Internet technology that had been around since 1994(!). REST made it easy to create Web APIs

(Application Programming Interfaces), which in turn made it easy for developers to create web clients that could talk to more than just one specific web server and vice versa. Suddenly, it seemed, there was a shared syntax and vocabulary for web applications to interact. A solution that was simple and obvious (at least to web application programmers). The right technology being around is a necessity, but in itself a suitable technology is not sufficient to change an institutionalized tradition of walled in local digital resources and specific local methods of working with those resources; blood is thicker than water. Thus, after the better part of two decades data silos and paywalls are still around, even if almost all scholars, scientists, and librarians agree that sharing data and documents in principle is a virtue (Fecher, Friesike, and Hebing 2015).

Why Are DSEs Still Information Silos?

Arguably the majority of digital scholarly digital editions (DSEs) are data silos still. Browsing through the issues of RIDE (<http://ride.i-d-e.de/>), one sees exclusively digital editions that are fully locally integrated server applications. Also Greta Franzini's (2016) excellent Catalogue of Digital Editions (<https://dig-ed-cat.acdh.oeaw.ac.at/>) for the majority lists API-less digital editions: 12 out of 258, or less than 5% do have an API—and whether those APIs are sufficient to 'un-silo' the editions is unclear. How come nothing has changed in this situation in over almost two decades? As has been argued above, this is not a problem of technology. The institutional landscape and development convenience may be part of the explanation, however: sharing digital resources does not require collaboration per se. A digital edition sharing its resources with the world only requires unilateral action by its editor. The simplest thing that could possibly work is allowing web directory access to a server location that hosts a dump of the digital materials—a technical no-brainer really. Such open data editions however, do not seem to thrive. How come?

I have at least two contentions about this. And they are truly contentious: I have only experience and hearsay to back them up, no real data, survey, or statistical analysis. But I offer these as the gambit of academic debate. My first contention is that textual scholars are still deeply entrenched in an intellectual hedonistic ideal of publishing the definite edition. Most editors think of an edition as a complete and finished product. Something that should not be tempered with, because it is argued and polished with arduous effort to academic perfection. The idea of reuse of that edition as “data”—especially as “primitive” or “raw” data in some computerized algorithmic process—in the eyes of scholarly editors is a category error. Because they do not regard that particular form of reuse as viable textual scholars also do not expect textual resources to be offered as such. In other words: there is

no innate wish within the textual scholarship community to push for a more distributed and interoperable model for text resource reuse. Even if it would be convenient for textual scholars themselves to be able to compare manuscript A in Rome with manuscript B in Zürich from their desk in London, that convenience is not a sufficient motivation to strive for some generalized method for decentralized access to textual resources because that simply is not part of the teleological worldview of the textual scholar. Plus, I think, most textual scholars that did produce a digital scholarly edition would argue that one actually can reuse it: look, it is a website, you can look at it, read it, use it. However, again that is a limited teleological conception of resource and reuse. It is the edition offered exclusively as a whole, as a solid philological fact, which is—I should be careful to point out—very much situated. A scholarly edition should be treated as a time, context, and editor dependent collection of interpretations, rather than a set of ‘philological facts’ of which I doubt they exist as Jerome McGann (2015) will have it. In contrast with a holistic teleological view of the (digital) edition, the need of colleague scholars is seldom for the edition as a whole, but more often for a part thereof. Colleague scholars want to compare a particular reading in one text with a different reading in another, or want to compare concepts and ideas over a range of documents, contrast different historical perspectives on particular events described differently in subsections of different codices. With the digital scholarly editions we have this requires scholars to navigate a plethora of different graphical user interfaces and differently implemented search tools, differently visualized annotation sets, etc.

My second contention is that the creators of digital editions—be they developers, scholars, hybrids, or any team like combination thereof—veritably never have the resources (time, funding, or skills wise) that enable them to produce the type of web serviced edition that Peter Boot, I, and many others (Robinson 2015; Siemens 2016; Thiruvathukal, Jones, and Shillingsburg 2010, and so forth) were thinking about. It is time and effort consuming enough to create meticulous transcriptions in TEI-XML of costly digitized manuscript folia, and to put these together in some concerted form to bring them to the Web. If those priorities are met, there is usually pretty little capacity left to tend to luxuries of machine access to the digital results.

Iron Manning the Digital Scholarly Edition

If the two contentions above hold water, they go a long way to explain why digital scholarly editions are all still based on a model of a unique, undivided, and complete product, and why the dream of “all readers may become editors too” (Robinson 2004) by reusing the various parts of digital editions, and creating their own transcriptions and annotations did not take

off. The second contention stands model for a great number of practical and pragmatic choices and limitations that influence the process of digital scholarly editing and the result of that process. Here however, I am not interested in these practical considerations. Nobody denies that it is highly convenient to have a text available anywhere, any time. Nobody denies that it is practical that you do not need to return to the national library of another country to check particular artefacts on a specific folio. No one has denied the facilitating nature of a full text search. And every editor is aware of the limitations that funding, institutional policies, and capacity put on any edition. But still the obvious practical, pragmatic, and (possibly) financial, benefits of digitally publishing scholarly editions have not convinced the majority of scholarly editors that digital editions are worthwhile.

I wonder in how far this may be due to the particular stance that we, digital scholarly editors and computational humanists, took in arguing the digital scholarly edition. Advocates of the digital scholarly edition, like me, have been hammering on the practical advantages and revolutionary paradigm shifting nature of electronic editions for decades. Julianne Nyhan and Andrew Flinn argue (2016) that there is a distinct and over used revolutionary motif in digital humanities publications. Arguably this motif has worked as a shibboleth for the community, but it also may have worked rather counter productive for the acceptance of digital methods in the humanities proper where the need for and advantages of a revolution were greeted with justifiable skepticism. Peter Robinson's "The Digital Revolution in Scholarly Editing" (2016) is an interesting publication in this respect. Clad in revolutionary terminology it is exactly highly critical of the usual as revolutionary depicted aspects of digital scholarly editing. Robinson argues that neither what we do as digital textual scholars, and neither what we make constitute any revolution in scholarly editing. We are still editors of texts after a critical fashion. Digital resources and environments may scale our work, but essentially do not warp us into a whole new undiscovered paradigm. But then Robinson continues to argue that there is still a truly revolutionary aspect to digital textual scholarship. It changes who we (textual scholars) are:

Every edition I have discussed so far has been made according to what we might call the Alexandrian consensus. The librarians gathered the many texts of Homer together; the scholars studied them and created a text; a grateful world took that text and read it. This model rests on two pillars. The first pillar is that only qualified scholars may examine the primary documents. The second pillar is that only qualified scholars have the authority to make a text the rest of us may read. Both pillars are now fallen. We are moving to a world where every manuscript and every book from the past is online, free for anyone to look at. You no longer

need to be tenured and well-connected to see a manuscript: increasingly, all you need is an internet connection. As for academic authority: peer-review and tenure committees are fine things but no-one is going to assert that only approved scholars can read manuscripts. (Robinson 2016:198)

Robinson takes my second contention above (the perpetual lack of capacity, funding, and skilled personnel) as a major argument in favor of shared open digital editions. The facsimile materials should be put on the Web publicly under the most free license possible, open to all to transcribe, annotate, interpret, copy, perform, and so forth. An admirable altruistic and democratic argument, further underpinned by the fact that what scholars do is usually financed through tax money.

The issue here is not whether I agree with Robinson or not, but that this “revolution” is simply asserted, that the future course of (digital) textual scholarship cannot be but this one, and that it is already happening. Much academic literature about digital textual scholarship seems to subscribe to a similar premise. Franz Fischer’s excellent contribution in *Speculum* (2017) postulates:

the (albeit slowly) growing number of digital critical editions increases the demand for assembling and providing critical texts that are in the form of a textual corpus, because only collections or corpora of texts that are otherwise dispersed on various websites allow for a systematic analysis and for efficient research across the works of a specific author, genre, subject, period, or language as a whole.

Again it is not whether Fischer is wrong or right here—for all practical purposes I actually agree with his contention. What I find interesting is that Fischer postulates a world of existing digital editions (or data) and then suggests several solutions for how to reconcile the heterogeneity and specificity of critical editing with the homogeneity of digital corpora. Solutions that are foremost conceptual methodological, but with elements of IT architecture mixed in.

I want to cast Robinson’s and Fischer’s premises here as “iron manning”, which is an inverse form of making a straw man argument (Casey 2012). This is unfair because their argument is well intended and not necessarily wrong, but for the sake of argument I use their premises as examples of what advocates of digital humanities often seem to do: assert some digital ideal and reason what methodologically ought to happen given this idealistic situation. Fischer and Robinson reason from asserted situations that are ideal from their perspectives, respectively that there are more and more digital editions and that editions are to be publicly open and

sharable. These are subjective practical and pragmatic ideals to start with. One can wonder however, if practical and pragmatic ideals resulting from a digital medium first and from scholarship only second are appealing to conventional textual scholars. Are textual scholars not primarily interested in what can be known about a text, and should we not therefore first of all demonstrate the epistemological added value of the technologies we propose?

What is the Epistemological Case for Digital Scholarly Editions?

The arguments cited above strike me as deterministic. That a digital technology exists and a futuristic ideal can be based on it does not necessarily mean utopia can and must be reached. Could it possibly be that most textual scholars simply judge that the model of the singular unified complete text—or codex in any case, be it a digital or print one—is simply sufficient, is even superior to any open and machine readable shared digital model yet presented? Fair is fair: we have little evidence that scholars and readers see a reason for an open, interoperable, distributed model for text resources, at best there is evidence of the contrary in many a failed tool, and the little use being made of digital editions (Porter 2013). The open and social edition that is passionately argued for by Ray Siemens and others (Siemens, Crompton, Powell, and Arbuckle 2016) is a prototype at best and does not yet attract a mainstream audience. Until now a paradigm shift in scholarly editing towards open and distributed digital editions seems not to have happened. Do readers really want to be editors? Moreover, most editors seem to be reluctant to embrace different, more open methods. Proponents stress the practical aspects of open editions all the time, and especially the pragmatic democratic character, but they never explain what the epistemological gain is in going through the extended trouble of creating open digital editions. Neither do they much consider what may be lost, as some encourage us to do (Sondheim et al. 2016). Putting tax financed editions digitally in the public domain is ethical, it does not explain how that changes our epistemological grasp of the subject matter or historical objects. Proponents point to the “wisdom of the crowd”, but also that is not revolutionary, either from pragmatic or epistemological point of view. It was always already perfectly possible to write a ‘letter to the editor’. Most editors would be happily surprised to receive one. Most of us do not edit the works of Darwin or Dante’s *Commedia*, but works more akin to some obscure and opaque 12th century book of prayers. The potential public reach of these works does not justify the development of heavy weight digital infrastructure or applications on the off chance that there may be an interested individual out there that actually knows more about your edition than you learned in the fifteen odd years studying it.

Calling it a revolution is not making a strong epistemological case. So what could be the

epistemological case? Why do we not discuss this in our field widely? Again as a possible opening for a debate, two contentions. The first is that distributed open digital editions further quality, and higher quality information progresses knowledge. What quality of information is, is a whole other conundrum that I will not detail here, suffice to say it is also a highly situated concept (cf. Borgman 2015; Gitelman 2013). But the epistemological argument for distributed information lets itself be construed rather easily. It is connected to skill. Suppose I am a scholar in need of high quality digital facsimiles of some folios of some manuscript. I could try to obtain high quality photographs and digitize them myself. But soon I would be facing questions like “How much DPI and what color depth should these images be?”, “How and where should they be stored?”, “What is a feasible standard for the technical description of these photographs?”. Chances are a textual scholar is less adequate at answering these questions than a library based digitization expert. The quality of the production of such digital facsimiles is related to the skill and knowledge of the producer. In contrast, if I want to be assured of the best possible transcription I am going to take my chances with the textual scholar specialized in 12th century European paleography, rather than with the librarian. This difference in assured quality of information does not evaporate post production. The curation and maintenance of digital information is yet another expertise, best left in the skilled hands of people upholding some digital repository. Thus digital scholarly editions may be sites of intersecting knowledge that affirm and support specific and highly skilled expertise. Of course, this does not mean they cannot also be at the same time altruistic and democratic, opening up editions to the public, but the primary epistemological scholarly gain seems to be in better, more specific support for quality knowledge and expertise.

My second contention in support of distributed information is related to distributed knowledge, also known as group knowledge or indeed as “wisdom of the crowd” (https://en.wikipedia.org/wiki/Wisdom_of_the_crowd). As pointed out above, arguing some epistemological advantage through “wisdom of the crowd” seems a dodgy fad at best, but that distributed knowledge adds up to more than local or individual knowledge is well known (Fagin, Halpern, Moses, and Vardi 1995) and can be explained relatively easily. Suppose a person X knows that fact A is related to fact B, but she does not know that fact B is related to some other fact C. Suppose also person Y knows that fact B is indeed related to C, but he in contrast does not know that B is related to A. The distributed knowledge that neither person has is that all three facts are related. They could gain this knowledge if the local information was somehow exposed and eventually shared (e.g. through word-of-mouth or publication). Actually this kind of turning distributed knowledge into added knowledge is

what textual scholars do all the time. What is, quite inexactly (Timpanaro 2005), called Lachmann's method is an excellent example of this. A scholar may find two copies of a text sharing the same copying error, in other words: information distributed over two sources. Combined this information adds the knowledge that these copies are genealogically closely related, closer in all likelihood than copies not having that error.

The salient point here is that connecting distributed information might be done computationally, whereas it currently must be done by hand, because the information that digital scholarly editions hold is represented almost exclusively through visual interfaces. This means that the epistemological benefit they can have as serving the constituting information of distributed knowledge is dependent on human agents connecting the dots. These human agents are part of a social epistemology (Goldman and Blanchard 2016), and it is not a given at all that distributed knowledge will not be uncovered in such a system of networked knowledge. However, if the information within these silos of data would be exposed in a way that non-human agents, such as web crawling software for instance, could navigate, possibly much more information could be related with epistemological gain much quicker than we can now. Distributed information systems proliferate this potential. If I need to create a digital edition that takes its images from one server and takes its transcriptions from another, and its annotations from yet another, I have to make the interface application on my computer talk to these other computers. And if my computer can, so can other computers.

Thus we have two epistemological arguments pro open distributed digital scholarly editions. Especially the second is an indistinct opaque artificial intelligence promise at best, colloquial known among developers as the 'semantic dream'. Even though the dream of an 'apparatus fonitum' is beautiful, its pursuit seems not be a very attractive proposition to textual scholars: most of the essential infrastructural pieces and veritably all the appropriately curated content are missing, and creating them requires great effort and technical skills that are far from abundantly available within the textual scholarship community. The promise of the first argument (furthering knowledge by leveraging quality of information) is only slightly less opaque, but at least the state of the art in digital infrastructure is such that this benefit could be attained quite easily and with feasible effort.

Silos and Epistemological Gains

The majority of current digital scholarly editions leverage neither of the two potential epistemological benefits described above. Most are based on a process of copying, creating or even re-creating all resources in one single digital location (i.e. on one server), forming

silos that gather many kinds of different information with different curation and maintenance needs. This situation will not change because of the latter epistemological argument I noted. That argument is based on the promise of non-human agents that still need to be designed and we have very little inkling of the value of the knowledge that would emerge in this way from distributed information.

The argument on quality of information, delegating tasks to the best available expertise, may actually be convincing for textual scholars. But the incentive for textual scholars to build distributed systems is at best altruistic: creating web services based distributed digital scholarly editions is harder and requires more technical expertise than creating complete and finite websites. The data silo is a cheaper, technically less complicated solution that is less dependent on many external stakeholders, quicker to realize, more adaptable to local needs, and usually has better predictability for deliverables and turnaround.

Does this mean that technically networked information is both epistemologically and pragmatically inevitably a non avenue for textual scholarship, that the idea serves no purpose? This remains to be seen. As with so many semantic technologies the value of distributed information for textual scholarship is in the stage of “promise”, and its value cannot be determined as there are simply hardly any real implementations to test drive. Given the complexities and unclear pay off it is also not a development that scholars can be expected to lead all by themselves. This is another conundrum: it is up to technologists and digital humanists that believe deeply in its promise to demonstrate the value of distributed information resources, but what epistemological affordances such networks might create is hardly up to the technologists to evaluate, as they are not textual scholarship experts.

Mirador as an Argument for Distributed Scholarly Resources

So why would I then still maintain, as I said above, that Mirador potentially plays a pivotal role in what may be nothing less than a paradigmatic shift in how we understand, approach and interact with cultural heritage resources? Mirador’s strength is in its architectural composition that a truly lazy reviewer could possibly attack as a mere patchwork of existing code pieces without much added value. But in fact this is exactly its strongest statement from an networked knowledge perspective, which enables it to be a part of a distributed model that would be able to leverage the epistemological benefits of resource quality I argued above. Mirador was built explicitly to do one job and one job very well: viewing digital images. The developers and designers stayed far away from every other temptation. On the

functional (or user facing) side of things they did not integrate image retouching functions, no dedicated transcription possibilities (although the annotation function has been used for simpler transcription tasks), no metadata editing capabilities, no annotation tools, no print-on-demand-service... no nothing. They just delivered a bare bones viewing, zooming, panning tool. This is an explicit design choice and by that an explicit assertion on how (scholarly) resources should be networked, namely not by building or integrating all software and data in one single (server) location integration, but through lightweight protocols that inform very thin tools—thin in the sense that they only serve one very particular task—where they can find data and what they should do with it. This strategy allows tools such as Mirador to be completely agnostic as to where some resource is located or how it is produced, served, and maintained. Mirador does not care about that, it just wants to see image data come as input and depict it. In this regard Mirador’s architectural make up can be read as an argument, just as editions (Cerquiglini 1999), interfaces (Andrews and Van Zundert 2016), and software code (Van Zundert 2016) can be seen as arguments in a wider debate on textual scholarship. Mirador’s argument favors distributed digital scholarly resources because it positions itself as a component that fits as a cog in such a distributed ecosystem of resources. Thus the epistemic argument of Mirador about the digital edition is that a digital edition ought to be a composition of various distributed resources.

Would the developers of Mirador have chosen any other strategy, then with every function they would have added there would have been a tighter integration with other software and stronger demands on the form (and possibly location) of data resources—and with every function Mirador would thus have become more an argument in favor of digital scholarly editions as monolithic data silos. In contrast, and quite on purpose, Mirador does not care whether one resource is in Madrid and another in San Francisco—as the developers explain themselves:

Users, such as scholars, researchers, students, and the general public, need to compare images hosted in multiple repositories across different institutions. They want a best-in-class experience with deep zoom capabilities, and viewing modalities optimized for single images, books and manuscripts, scrolls, or museum objects. End users want to create and view image annotations, comments, and transcriptions within a single user interface, *regardless of the system in which they were originally created or hosted*. (Sanderson, Snyderman, Winget, Albritton, and Cramer 2015; my emphasis)

This does not just make sense from the user’s perspective (who does not care where the

resource is). It makes sense from a technical point of view too: why duplicate the burden of maintenance and development for all resources? But most saliently: it makes sense from an epistemological point of view: it allows the object of expertise to reside with the expert. It allows us to have the responsibility for the quality of whatever is done with the object located in the place best equipped to that end. Obviously with a print publication this is harder as all epistemological objects (transcriptions, structure, contextualization, pictures, index, etc.) are solidified in it. One can update the publication, but it takes another expensive print run, and it is unlikely that this will be done in the case of individual changes—the long list of changes (<http://vangoghletters.org/vg/updates.html>) to the Web based Van Gogh Letters edition (Jansen, Luijten, and Bakker 2009) testifies to this. In the case of facsimiles an editor often has to make do with a lower quality photograph of a folio as an illustration (e.g. Figure 9). Arguably higher quality can be offered and maintained by an expert in an institution that takes the care for digital images and its sources to the core of its tasks. Chances are this will not be the scholar that made the transcriptions and edition. In the case of the edition of the Middledutch Comburg manuscript (Brinkman and Schenkel 1997) the repository of the source—the Württembergische Landesbibliothek—indeed did bring high resolution images online, according to the associated MARC21 information some thirteen years after the print edition was published (“Comburger Handschrift - mittelniederländische Sammelhandschrift - Cod.poet.et phil.fol.22” n.d.; “SWB Online-Katalog” n.d.).



Figure 9: Example of making do with a single photograph. These pages, taken from Brinkman and Schenkel's edition of the Comburg manuscript (Brinkman and Schenkel 1997) show one of the very few reproductions (cut and down scaled) in the print edition. (Image courtesy Verloren Publishers and authors.)

The facsimiles and the diplomatic transcript of the Comburg manuscript—one of the “flagships” of Middle Dutch literature—for the moment lead a scholarly unsatisfying divorced life. The facsimiles available on the Web at the site of the Württembergische Landesbibliothek, the diplomatic transcript available as offline print edition. Being able to link up both through an architecture as proposed by Mirador would no doubt greatly improve the epistemological value of both.

Mirador as Part of an Ecosystem of Digital Scholarly Resources

Even though it may be the case that an epistemological benefit can be expected from distributed digital scholarly editions, it remains to be seen if such an epistemological effect would actually be realized. The answer is highly dependent on the facility of the technical solution provided. That is: how well and how easy does Mirador let itself be used by scholars and developers alike?

If Mirador is set up for you in the right way and if the repository you want to connect to

supports the IIIF protocol, then Mirador brings you a long way. IIIF is short for International Image Interoperability Framework (<http://iiif.io/>). If you want a distributed ecosystem of scholarly resources—i.e. not having all resources in one place and still being able to reuse all resources—you need some kind of formal language that allows the different services that are resource consumers to know what the resources are, how they are structured, and how they may be used. That sounds high-tech, but in fact the core of it is rather very social: it comes down to a group of people agreeing on how things will be strictly written and these people (and eventually those they attract to their ideas and solutions) adhering to the agreed upon semiotics. If the semiotic signs and rules are rigid, algorithms can process them. For the online exposure of digital images the IIIF protocol is such a formal language. It developed grass roots from a community that saw the need for sharing digital image information. Meanwhile the IIIF community (<http://iiif.io/community/#participating-institutions>) looks impressive and thriving. The various adoptions of the IIIF protocol for specific viewing modalities, such as in the Universal Viewer (<http://universalviewer.io/Universal>) or in Leaflet-IIIF (<https://github.com/mejackreed/Leaflet-IIIF>) testify to IIIF's success. Mirador is one of the kernels in IIIF's developing ecosystem of open source image tools that facilitate the exposure of open access image resources on the web.

For an image resource on the Web to make its content known according to IIIF, it needs to serve a so called manifest file in JSON (<https://www.json.org/>) format that describes the content and structure of the given resource. The exact semiotics are all verbosely documented on the IIIF site (<http://iiif.io/api/presentation/2.1/#manifest>). The other essential part for an image repository is a server that will stream requested images to the application ('client') that wants to use the image resource. Image servers come off the shelf these days, as in the case of IIPImage (<http://iipimage.sourceforge.net/>). Not any image server will do, however. Again IIIF compliance is a prerequisite for the server to be part of a distributed network of resources that Mirador asserts. Several of such servers exist though (cf. <http://iiif.io/apps-demos/#image-servers>).

Any Mirador viewer can be pointed to a particular manifest by clicking the 'Replace Object' option (Figure 10). This will let the user choose from a list of pre-selected repositories (Figure 11), but also a manifest URL can be manually keyed in (Figure 12). Thus when you can point Mirador to a location such as http://sanddragon.bl.uk/IIIFMetadataService/Cotton_MS_Claudius_B_IV.json it will find all the information it needs to start streaming images towards the user. In this case the facsimiles of an incomplete Old English Hexateuch

(“Cotton MS Claudius B IV” n.d.). One can gauge from this how distributed Web resources work: all the client (Mirador) really needs to know is the one URL locating the manifest in the image repository. Because the image repository’s manifest behind that URL adheres to the IIIF protocol both image server and client can operate seamlessly without having any geographical or institutional constraints on each other.

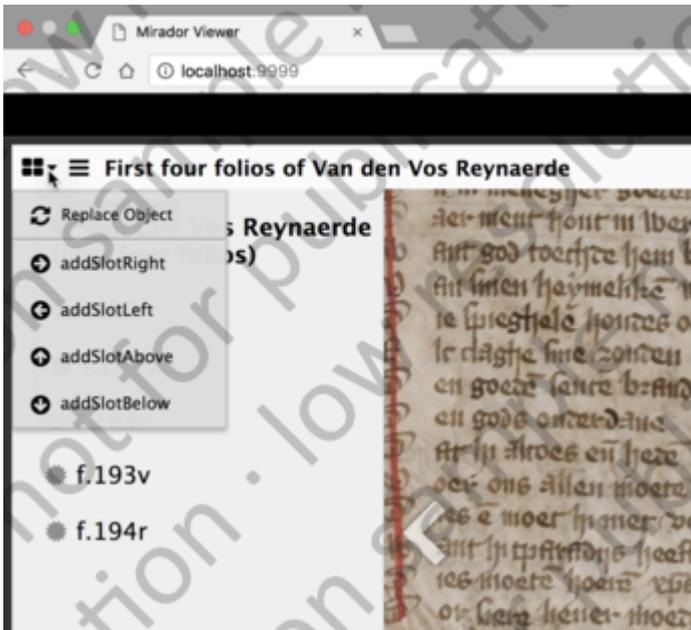


Figure 10: Mirador’s “Replace Object” function.



Figure 11: An Example of a list with image repositories.



Figure 12: Manually adding an image repository's manifest location.

Building a Digital Edition with Mirador

Suppose we have a world of distributed scholarly resources: there are facsimile repositories, and other repositories serve transcriptions of these facsimile, and yet other repositories may have annotations pertaining to these materials. In a world of distributed scholarly resources that connect to each other via APIs, talking certain protocol languages to each other, one should expect facsimiles, transcriptions, and annotations to be different independent resources that are polled and visualized together by a dedicated Web application. Depending on the resources polled, different editions may be created using in part the same resources. This situation is conceptually visualized in figure 13 (in comparison to the currently more common monolithic digital scholarly edition).

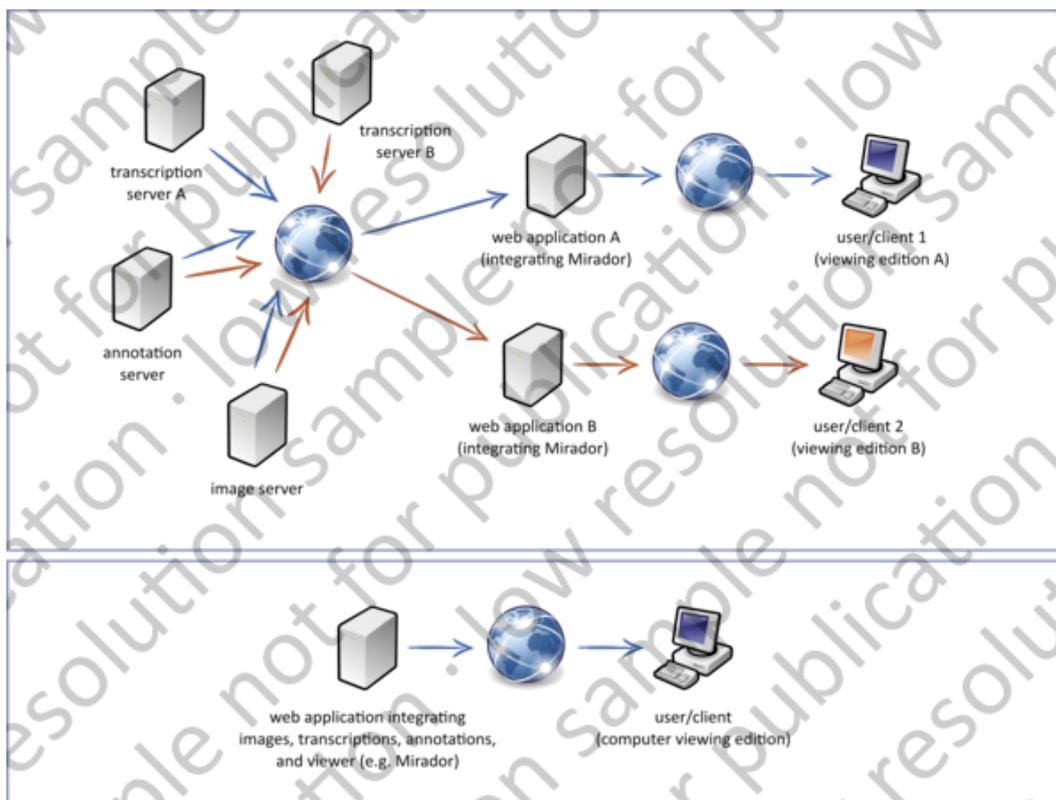


Figure 13: Distributed digital editions varying resources (top) vs. singular integrated monolithic edition (bottom).

Suppose a scholar would want to create a digital scholarly edition from such distributed resources, what would it take? To have that be more than just a theoretical wish, and more than just a thought experiment, I have implemented such an edition with a tiny sample of images and text as a demo (cf. Appendix 1). Mainly I wanted to know how hard it would be, because the more facile the work the more scholars will possibly take to developing digital editions from distributed resources. The experience points out that one needs to be quite an experienced (web) developer however, to be able to create all the servers and the integrating application. On a somewhat higher level of overview the following describes the implementation of the demo edition.

Setting up an image server and a Mirador instance is not trivial. You have to know your way around a Web server such as Apache (<https://httpd.apache.org/>) and how to run it securely. Installing an image server such as IIPImage (<http://iipimage.sourceforge.net/>) is relatively straightforward, but having it work properly with Apache involves less than trivial configuration. Apache and IIPImage together form the engine of an image repository. Once both are installed and properly working together we can admire the front page of our image server (Figure 14).



Figure 14: An empty image repository is born.

The fuel for the Apache-IIPImage engine is images. These images need to be prepared as so called pyramidal TIFFs that allow efficient and fast streaming image information to a client (such as Mirador). Effectively each image is stored in several sizes in one file to support zooming. Each size is associated with a layer, and each layer is broken up in many small tiles that travel the Internet easily and fast. To create pyramidal TIFFs one needs to be comfortable with a tool like ImageMagick (<https://www.imagemagick.org>) and commands such as `convert original_0001.tif -define tiff:tile-geometry=256x256 -compress jpeg 'ptif:pyramidal_0001.tif'`. Once the pyramidal images have been created the image server is able to show us facsimiles (Figure 15).

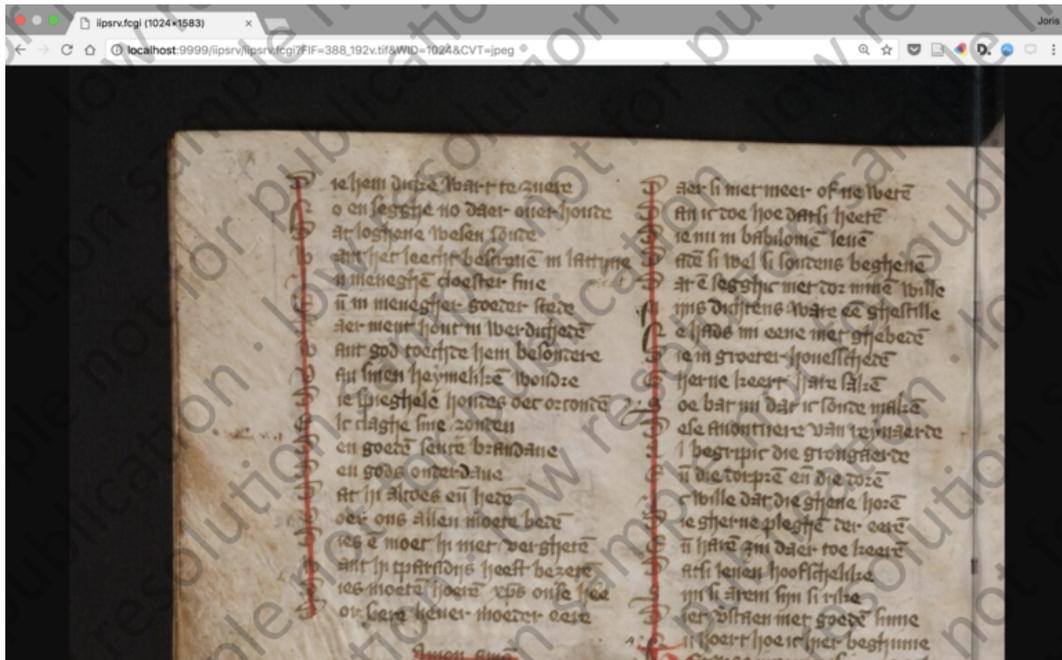


Figure 15: A first facsimile.

Lastly the manifest file needs to be created. Usually this would require a developer to generate this from some database of image information or by creating a script that is able to derive basic information from image files directly (using a tool like [exiftool](https://www.sno.phy.queensu.ca/~phil/exiftool/), <https://www.sno.phy.queensu.ca/~phil/exiftool/>). In the case of this demo the manifest file was put together by hand given the very few images it would describe. The hardest part of this was understanding the IIIF specification, thus the formal language used to describe the structure and metadata of image collections. The specification is not really complicated, but getting to know it still involves somewhat of a learning curve. Note that there is a tremendously helpful tool when building manifests by hand or when trying to familiarize oneself with the IIIF manifest specification, which is the Oxford Manifest Editor (<http://iiif.bodleian.ox.ac.uk/manifest-editor>). Once generated or written by hand the result is a manifest file (figure 16) that can be consulted by any computer running an IIIF client.



Figure 16: Manifest file sample.

At this point we have an image repository that contains, presumably, the facsimiles of the codex we want to turn into a digital edition (e.g. something like figure 15). Arguably this part of the work would be delegated to some specialized service or institution. Of course if no institution hosts the images you need already, the editor faces convincing such a service of hosting them and possibly funding the related work and maintenance. Or the editor could create a self-maintained repository as explained here.

Now an actual Web application is needed that uses Mirador to look at the image repository. These days Web application development means less and less writing “plain old HTML and JavaScript”. Mirador is not an exception, and reusing Mirador's source code while at the same time adhering to more formal and current development principles requires substantial Web development experience and knowledge. Mirador is developed using the Node.js runtime environment (<https://nodejs.org/en/>). This means it actually can be run out of the box on Node.js as a server, which is a solution one might opt for. The main reason for choosing Node.js however, seems to be the NPM package manager (<https://www.npmjs.com/>) that comes along and which protects developers from proverbial “version hell”—that is: you cannot use any version of a component with any version of another software component. A 19th century cart wheel will not fit your 21st century Tesla, even if it is all “wheels” and “vehicles”. Mirador uses a lot of third party JavaScript components, and so it needs to carefully check the versions of those components that it

combines. NPM is a highly convenient way to deal with this problem. However, the consequence is that if you want to use the Mirador source code from its official repository (<https://github.com/ProjectMirador/mirador>) the ‘proper’ way, you will first need to download and compile all components and sources that Mirador uses into one `mirador.js` file using Grunt (<https://gruntjs.com/>), which is another tool in the Node.js domain. Once this is done, the Mirador demo application provided by the original authors can be deployed by moving Mirador’s whole directory in the folder designated to serve files from by the Apache web server. Of course there is also a less formal but more convenient and speedier way to start using Mirador. If you do not require substantial changes and adaptations it is possible to reuse a “drop in” single pre-compiled `mirador.js` file and to refer to this inside a web page’s HTML source. Mirador’s official Github repository provides pre-compiled versions for this (<https://github.com/ProjectMirador/mirador/releases>).

At this point we can reach our Mirador instance via any web browser, and we can add the URL of our manifest upon which Mirador should show us the contents of our image repository (figure 17).

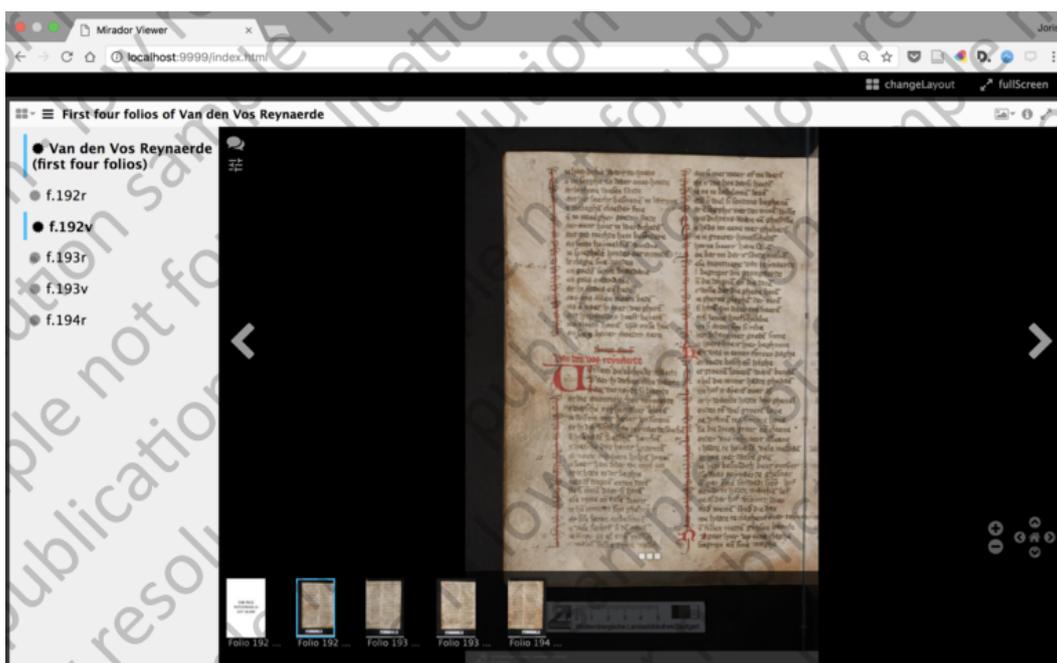


Figure 17: Mirador is up and showing images.

We need a source for our transcriptions too. This involves setting up another server that will on request provide the transcription of a particular page of the codex Mirador is looking at. One might possibly adapt one of the transcription/visualization environments named in the

beginning of this article. For the demo included with this article I created a basic transcription server myself (https://github.com/jorisvanzundert/mirador_review_demo). It uses a Sinatra Web server in the Ruby language (<http://sinatrarb.com/>; <https://www.ruby-lang.org/en/>) and serves a TEI-XML file that transcribes a tiny portion of the first facsimile (Figure 19; see Appendix 1 for the full source of the textual data). Using the Nokogiri HTML/XML parser in the background (<http://www.nokogiri.org/>), the same server will on request use an XSLT stylesheet to transform the TEI-XML into HTML, visualizing either a diplomatic (Figure 18) or critical visualization of the transcription.

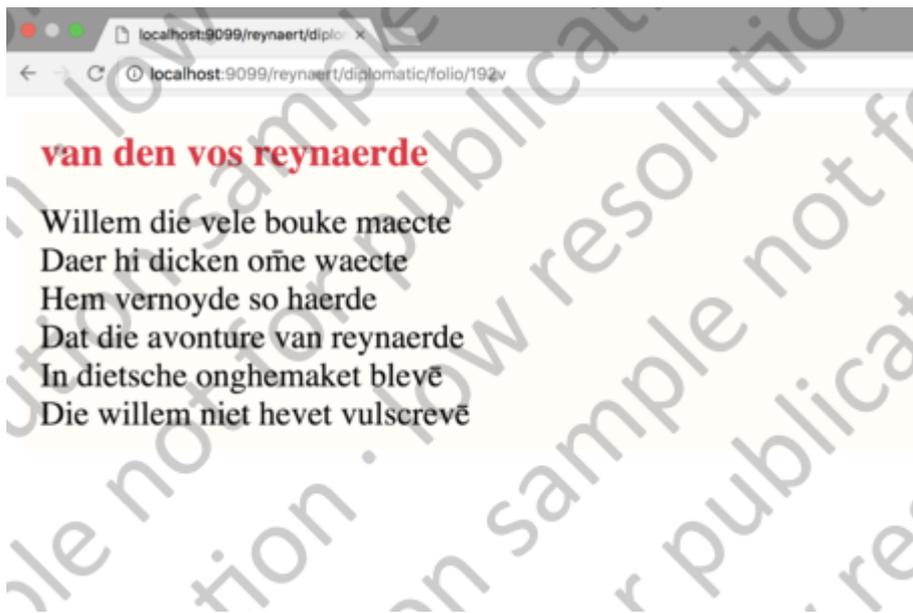


Figure 18: A diplomatic transcription served through the SimpleTranscriptionServer.

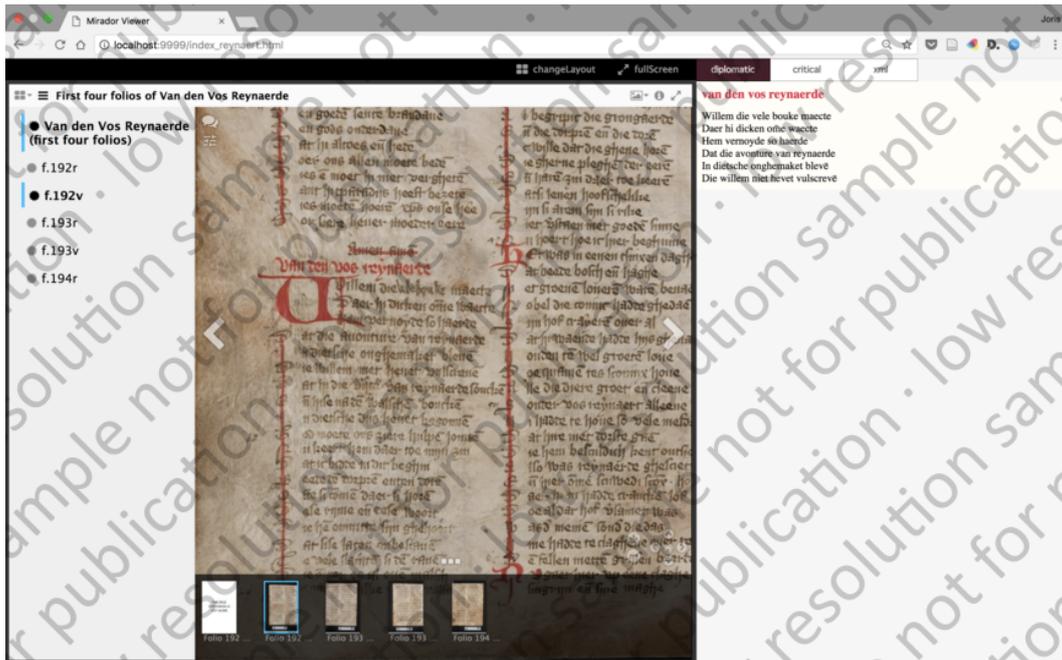


Figure 20: A client combining Mirador viewer for facsimiles and a text viewer for transcription.

Along the Seams of Mirador

This is where we start to stumble upon some of the limitations of Mirador. From the point of view of the scholar who studies manuscripts one would want a more granular connection between text and image. But as Mirador’s developers chose to pursue one task and one task very well, any extras one would want as a scholar will have to be added by someone with software development capabilities. All this is doable, but it is harder work than what I outlined above, and there I did not report the nitty gritty details of trying a number of non successful solutions that I abandoned in deep heaps of Linux system level error messages that also I—being an apt web application developer but not a very apt systems admin—could not solve quickly and for me conveniently.

Until this point development consisted of adding whole components together into services that could usefully speak to each other. To realize a more granular linking between text and image we will have to delve into the code of Mirador itself, however, to make some things possible it does not support out of the box.

And this is then also the point where we get a feel for the seams of Mirador, for the rough edges of its codework. At the same time that reusing and wrapping components is a most brilliant strategy to reduce maintenance and reinvention, it also has certain disadvantages.

Mirador uses code that has been made by many different bodies, and this shows. Programmers use different styles of coding—and there are many styles (cf. e.g. Croll 2014). In general it appears to me that the developers of Mirador are apt JavaScript programmers, which makes for well written code. However it is not like looking at a Rembrandt or a Van Gogh. It is more like Rembrandt, Van Gogh, and Picasso came together and decided to work concurrently on the same painting. For people wanting to integrate Mirador and add functionality this can be a very real difficulty. Another bother, but this may be more of a personal pet peeve, is Mirador's continuously use of object encapsulation through `jQuery.extend()` (<https://api.jquery.com/jquery.extend/>). This turns Mirador effectively into a God class (https://en.wikipedia.org/wiki/God_object) where everything is connected with everything else but not everything is necessarily clearly and consistently named. Finding the right hooks and slots to adapt Mirador to your wishes is therefore harder than might have been necessary. This is not helped either by the fact that Mirador's quick start documentation (<http://projectmirador.org/docs/docs/api-reference.html>) is very much in an beta phase and that its API documentation is none (<http://projectmirador.org/docs/docs/api-reference.html>). Even though being rudimentary the documentation gives a seasoned web developer just enough hints and insights that she might find her way through. If she would, this developer could hook into Mirador's code to achieve a more granular linkage between facsimile and transcription. In the demo I wanted a click on any verse in the transcription to cause Mirador to pan/zoom to the particular verse on the facsimile. For any digital scholarly edition of medieval text tying facsimile and text together this would seem to me a basic prerequisite of convenience, because either the transcription is used as a reading aid or the facsimile is used to verify the correctness of the transcription. Such linking requires some way of relating a particular TEI l-element (i.e. line element, <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-l.html>) to a particular area of the facsimile. And here we come upon a rough seem in the IIIF protocol that Mirador is applying.

According to the IIIF specification there are actually several ways how a more granular linking between text and image could be achieved. One is the ability to define ranges which may indicate a range of pages for instance, or a particular area on a page (<http://iiif.io/api/presentation/2.1/#range>). Another way is using a segment selector in an URI (<http://iiif.io/api/presentation/2.1/#segments>). Neither solution is very satisfying, however. For one thing, the IIIF specification is an evolving specification still and the ranges model is a good example of its current volatility. IIIF community discussion around ranges led to the deprecation of the current specification for ranges, which is to be fully replaced in IIIF version 3.0 (<https://github.com/IIIF/api/issues/1070>). More importantly, both

solutions assume knowledge about the transcription being part of the description of the facsimile. From the point of view of decoupled distributed scholarly resources maintained in a context of specific expertise, this is unsatisfying because it establishes a strong coupling between the facsimile resource and text resource: the facsimile description has rather very precise knowledge about a specific transcription in another specific resource. In keeping with the idea that services should be as agnostic of content as possible we would want our Mirador application, just like it reached out to an image repository, to register with some other service that would provide just enough information for it to link specific parts of the transcription with specific parts of the facsimile.

Fortunately the IIIF specification decided to follow the web annotation recommendations by the W3C that provide exactly this type of model (Cole 2017). The web annotation model became an official W3C recommendation on 23 February 2017 after years of preparation by the W3C Web Annotation Working Group (<https://www.w3.org/annotation/>), a group that derived from another grass roots initiative called the Open Annotation Collaboration (<http://www.openannotation.org/>). It is this Web Annotation Model that allows us to call another independent service into action where Mirador can indeed retrieve annotations on a facsimile independent of the description of that facsimile by the image resource. Conveniently—for the demo in any case—there exists a readymade SimpleAnnotationServer that will act as such an independent resource of annotations (<https://github.com/glenrobson/SimpleAnnotationServer>). This service provides the information to both my self rolled SimpleTranscriptionServer and to Mirador about annotations available for the facsimile. Mirador retrieves all annotations that exist for the facsimile. It is natively ("out of the box") able to depict these (see figure 21). However, to enable the user to click on a particular part of the transcription and to have Mirador then pan and zoom to the corresponding part of the facsimile, one needs to hack a few lines deep inside of Mirador's innards. Mirador encapsulates its own event handler mechanism that enables to publish events and to subscribe to the same—that is, one can signal an event happening (e.g. a click on some graphical object) and one's code can be notified of such events. If a user clicks on a verse the custom made text_viewer component publishes a `request_fit_bounds` event. After having modified its code a bit, Mirador listens to these events and subsequently pans and zooms to specific verses. Clicking on the first character of the transcription, for instance, zooms to the enlarged initial just to make the case (figure 22).

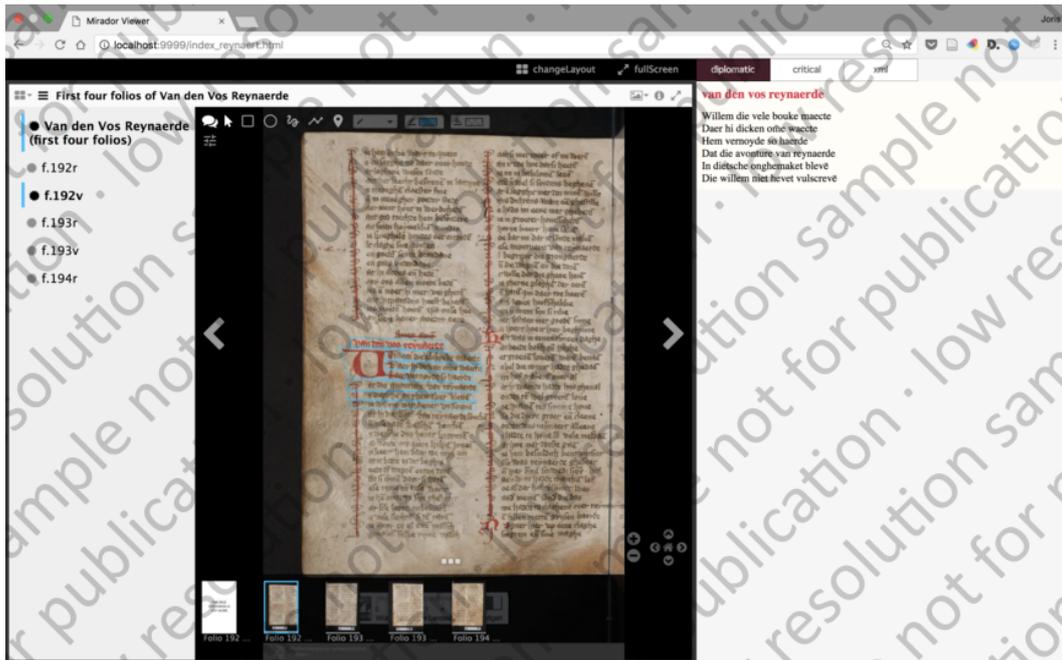


Figure 21: Mirador showing annotations, annotated areas with blue borders.

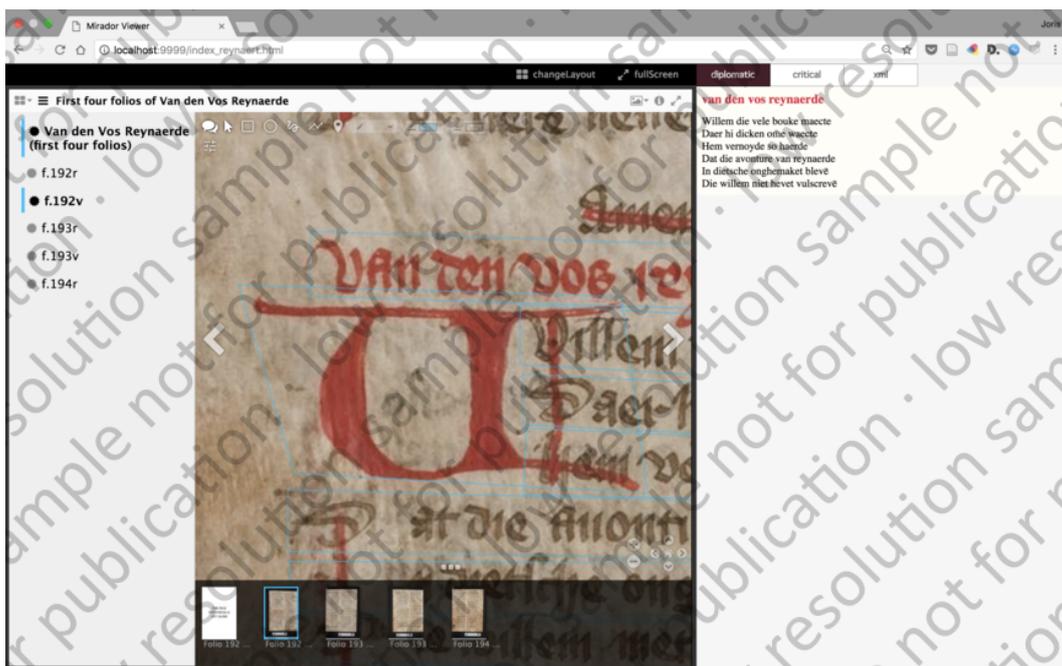


Figure 22: Mirador zoomed in on the enlarged initial of the manuscript.

The fact that IIIF relies on the Web Annotation model of the W3C is fortunate in more than one sense. Obviously it ties in with my contention that knowledge quality is best served when it resides with exactly the expertise it needs. That in turn serves to keep knowledge within repositories to the bare minimum needed to serve a very specific purpose, and with

this comes efficient maintenance and other practical benefits. But it is also fortunate in the sense that this way the IIF specification runs less a risk of bloating. What is a risk for integrated infrastructures (i.e. that they topple under the maintenance of ever more tools and data being integrated) is a risk for protocols and standards too: they may try to expand their coverage and expressiveness forever more. Indications of such bloating may be found in the resource structure specification of IIF that deals with what is actually on the images, how they form a collection etc. (<http://iif.io/api/presentation/2.1/#resource-structure>). The lure to over specify is a real pitfall, but judging from the technical community discussions (<https://github.com/IIF/api/issues/1070>) and deprecation warnings (<http://iif.io/api/presentation/2.1/#collection>), it looks like the community is veering towards a sparse and reluctant policy in adding specifications, which would be a tremendously good thing. Literally anything can be on an image, so carefully not wanting to describe such content seems essential to me in keeping a lean and effective protocol. Defining what is on the image would rather be left to more specific community standards and protocols. In the case of manuscripts and codices, one can imagine a very productive ‘hand off’ indeed of description according to the TEI model once the object of description is what is on the page. At the same time that this makes sense, it is also hard. Textual scholarship, and especially its digital adepts, have upheld a naive notion of unproblematic separation of materiality and textual content for a long time. DeRose’s publication on the OHCO (Ordered Hierarchy of Content Objects) can be taken as a convenient temporal marker for the emergence of this attitude (DeRose et al. 1990). In reality text and materiality are deeply intertwined and their separation is not unproblematic at all (Galey 2010:110–114). The trickeries of such an illusory separation almost immediately surface when one starts working with Mirador and codices. Mirador sports a “bookView” function (see Figure 23), which presents every two images as pages that face each other in a book. The “bookView” function however, assumes that a series of images is 1-to-1 related with a series of images of consecutive individual page sides, and that the first in a series of images is a depiction of a right-hand side page face. Neither needs to be true, even if these assumptions are in line with what is more or less a general convention. In the case of the demo presented, the chosen example text starts midway left column on a verso of a folio. Unmitigated however, the bookView function would depict it as a right-hand side page. Presently, the only way around this is inserting a page intentionally left blank (cf. Figure 24 and Figure 25). The IIF specification that Mirador adheres to has little way of expressing this structure that intersects material and textual dimensions. This is a case where Mirador’s design makes implicit assumptions that are not backed by its IIF model. Keeping with the technically sensible idea of separating

concerns a book view function ought only to have been implemented if there is actually a model (e.g. TEI) that informs it about the relation between text and pages, and between pages and images. This is not to say that Mirador’s developers were unaware of this—I simply do not know. The problem may have been that exactly on this issue the specification is still volatile. But this does show how easy it is to misconstrue the reach of a specification. With, I think, harmful epistemological consequences. The reader may want to point out that on the level of world history very little harm was done. Fair enough, but taking digital textual scholarship seriously hardly concerts with leaving potential confusion over basic documentary information about which pages were right and left.



Figure 23: Mirador’s “bookView” function.

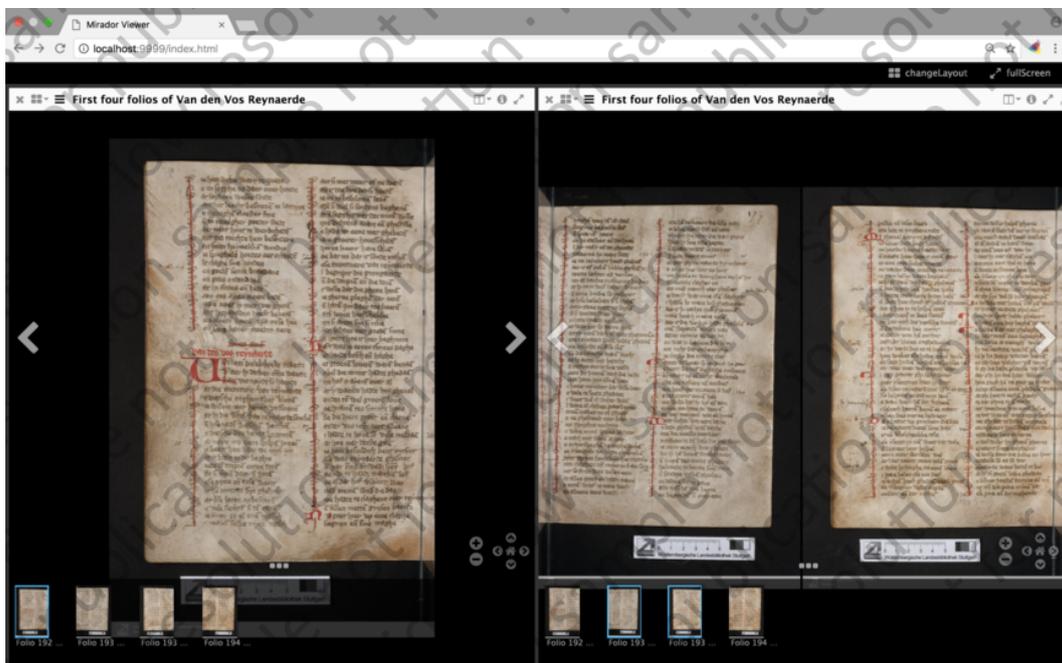


Figure 24: Mirador’s unmitigated book view, suggesting that this text starts on a recto.

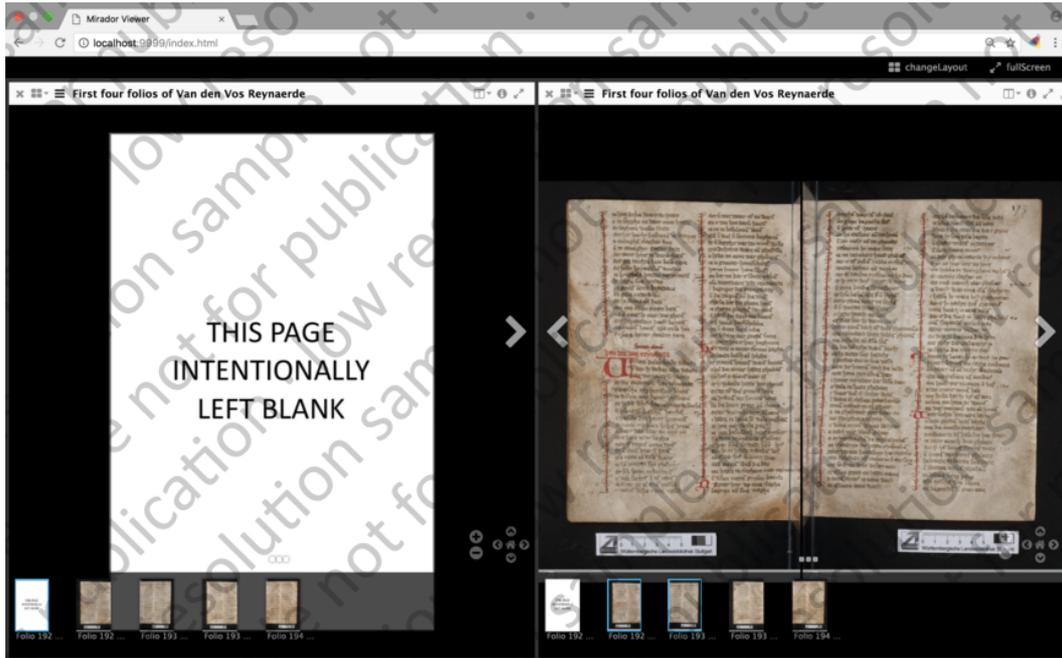


Figure 25: Mitigating Mirador's book view by inserting a blank page.

In the case of manuscript images and scholarly transcriptions IIIF and TEI stand much to gain from each other. But it remains to be seen how they should be aligned or connected. This is an issue that saw some discussion already recently on the TEI mailing list (cf. Stutzmann 2017). In the demo I challenged myself to make Mirador pan and zoom towards a particular area that is annotated (of sorts) by a particular part of the transcription. This was also the use case that brought to life the TEI-L discussion: how to implement sub page granular referencing between text (transcription) and image (page). From the point of view of distributed and decoupled scholarly resources one does not want strong integration of specific knowledge of the text description encapsulated within the image (IIIF) description, nor does one want vice versa knowledge about the image description tightly integrated inside of the text (TEI) description. Thus neither IIIF's proposal to link directly into a TEI file by using XPointer and XPath (<http://iiif.io/api/presentation/2.1/#segments>) queries:

```
{
  "@context": "http://iiif.io/api/presentation/2/context.json",
  "@id": "http://localhost:9999/annotations/annotation/annol",
  "@type": "oa:Annotation",
  "motivation": "sc:painting",
  "resource": {
    "@id": "http://localhost:9099/reynaert/diplomatic/folio192v#xpointer(tei:text/tei:body/tei:div[@type='folio']/tei:div[@type='part' and @n='1_1'])",
    "@type": "dctypes:Text",
    "format": "application/tei+xml"
  },
  "on": "http://localhost:9999/reynaert_fragment/folios/folio_192v#xywh=100,100,500,300"
}
```

nor what some supposed on the TEI list (cf. Holmes 2017), to strongly couple parts of transcriptions to segments (areas) of images using, for instance, a `facs` attribute:

```

<div type="part" n="1_1">
  <1><c facs="http://localhost:9999/reynaert_fragment/folios/folio_192v#xywh=100,100,500,300">W</c>illem die
  <subst>
    <del>
      <choice>
        <orig>m</orig>
        <reg reason="srcc;">M</reg>
      </choice>adockes</del>
    <add>vele bouke</add>
  </subst>
</div>

```

is really satisfying. Again the reason is that editors of scholarly texts best are not bothered with specific image related description, and a protocol or standard ought not to push this highly skilled knowledge on them—which is of course not to say that it is forbidden ground for the textual scholar, but if she wants that knowledge she ought to find it in the designated place, and should not be bothered by it normally.

Moreover the schemes above pretty much forbid a thing like competing transcriptions. Suppose you have two competing transcriptions for the same facsimile. With the strong coupling of the transcription fragment inside the image segment (`book1/canvas/p1#xywh=0,0,600,900`) in the IIF scheme, a viewing client like Mirador has no choice: the image description dictates that the client should go look for one specific transcription (that of the very specific XPointer denoted in the segment definition). This type of strong integration goes exactly against the nature of distributed resources and nullifies the ability to discover distributed knowledge. If there are multiple competing transcriptions for one particular facsimile then a viewer for that particular facsimile should be able to discover these transcriptions. The strong coupling above forces this work of discovery onto the creator and/or maintainer of the facsimile image. Thus probably a person who's immediate interest and maybe who's expertise is not geared to that task. So instead, and in the interest of epistemological gain, we ought to register the transcription with an additional independent service. A client like Mirador can in that case just ask from such a service: "Is there any transcription for this particular facsimile?" The service will then answer with the appropriate resources, and if there are competing transcriptions the viewer can choose one or present them as alternatives. Introducing such an intermediate service is called 'adding an indirection' or 'making resources dereferenceable' if you want it in stark information technology terminology.

To my knowledge there is no community based consensus about some formal protocol to support this type of service. It could be a tremendously productive idea if both IIF and TEI communities would enter in a dialectic on that topic. For the moment this type of behavior can be mimicked at best by utilizing the Open Annotation schemes that IIF adheres to, as demonstrated by the Mirador based application presented.

Conclusion: the Risks to Mirador's Distributed Worldview

Where do we stand after a long journey from Mirador along IT architecture, monoliths and epistemology to distributed knowledge and building a demonstrator scholarly edition from distributed resources? My conclusion is that things look pretty bleak with regard to the potential success of distributed resources in scholarship. Not because distributed resources are a bad idea obviously—in fact I would argue they are the only sort of IT information and knowledge architecture that makes sense from a scholarship point of view. They fit better with the tenets of scholarship that value multiple perspectives and intersubjective interpretation. But IT infrastructure is often overlooked as both a metaphor for and a formative agent of epistemological construction. That it can be a normative epistemological means is—I would argue—a hardly known fact in scholarship. Perceived as pretty much unrelated to the core of their epistemology, scholars are thus unlikely to meddle in much with the architecture of their IT infrastructure. As argued there should be a modest epistemological gain in distributed resources. However, as this is currently a technological promise at best, this is only a weak argument in favor of distributed digital scholarly resources.

The deep involvement of scholars with the requirements definition for Mirador and IIF have no doubt contributed to the success story of Mirador and IIF so far. My worry, however, is with the integration of these technologies in existing institutional contexts. Mirador makes a very strong statement for distributed architecture and connecting distributed knowledge and expertise. But this statement is in code, in technical architecture, and in the technical specification of a protocol. This makes it a statement that currently can only be really understood at all levels by developers, or scholars that are very apt developers too. As the building of the demo shows, a high level of IT expertise is needed to create scholarly tools using Mirador as a component. Of course this is not to suggest that developers do not carefully listen to scholars formulating requirements. Mirador is obviously an excellent case in this regard, for the developers understood very well the need for scholars to compare images from remote different sources. But this does not mean the distributed architectural vision carries over automatically to local contexts. As I have argued distributed knowledge is not part of a general scholarly worldview, rather distributed architecture emerges only as relevant to scholars in the very specific case where they cannot easily obtain a very specific resource. The generalized case of having all digital objects available as distributed resources remains an abstract idea for them, and moreover an idea of which the realization would require tremendous effort while having little concrete epistemological

appeal. It is therefore questionable if the majority of requirements put to developers by scholars when integrating technologies like Mirador and IIF in a local institutional context will actually forcefully lean towards a distributed architecture. I am pessimistic in this respect, and I think it is far more likely that both developer and scholar convenience will favor local repositories and local integrated tools connecting to those local repositories: linking cross institutional distributed resources requires a lot of overhead in meetings, discussions and collaboration.

In the case of IIF, as in many other cases, there may also be a discrepancy between who decides on the architecture of the technology and who is assumed to reap the epistemological benefits of it. The visitors and speakers lists of the 2017 IIF conference at the Vatican reveals a large majority of technologists, and a small minority of scholars (<https://2017iifconferencethevatican.sched.com/directory/speakers>). Some well known names in digital scholarship (Jeffrey Witt, Peter Robinson, John Bryant, Ben Brumfield, Frederik Kaplan) are represented, which is good, but that group should broaden and diversify if it is to avert the next futile technology push. Mirador and IIF may turn out to be typical technologies that came too early. It could very well be that the scholarship community is not well versed enough, and certainly not fluent enough yet in the computer and IT architecture languages that are needed to fully appreciate what distributed resources have to offer and how they constitute a different worldview from monolithic software solutions.

When all those risks of development, mutual understanding, and adoption can be mitigated, there remain still some caveats purely on the technological plane. Authentication for one can be a nightmare and the sudden death of any well argued infrastructure. And barring that there is still the CAP theorem to deal with if distributed scholarly resources indeed would take off (see https://en.wikipedia.org/wiki/CAP_theorem). But technical issues are usually the easiest to solve in the case of sociotechnical systems.

Somewhere in between the social and the technical is the question about Mirador's, and mostly IIF's, potential for adoption by existing repositories. Notwithstanding IIF's thriving community, it remains to be seen if repositories that heavily invested in other technologies such as, for instance, the DFG Viewer (<http://dfg-viewer.de/>) adopted by e.g. the Württembergische Landesbibliothek (<http://www.wlb-stuttgart.de/>) and the Universitäts- und Landesbibliothek Münster (<https://www.ulb.uni-muenster.de/>), will be inclined to support yet another protocol too. As often, the technology is not a showstopper in this case, but institutional politics, development capacity, funding, and policies may very well be.

Still, in all I like to think that Mirador got most things exactly right. Certainly the choice to limit the viewer to what is the bare minimum of functional essentials, built from reused components and software is wise. And most of all the developers succeeded in not bloating the tool under pressure of feature requests. Hopefully the same will go for IIF. Less is more, the leaner the specification the easier the adoption.

References

Andrews, Tara L., and Joris J. van Zundert. 2016. “What Are You Trying to Say? The Interface as an Integral Element of Argument.” In *Digital Scholarly Editions as Interfaces: Abstracts and Programme*, 29–30. Graz: Centre for Information Modelling – Graz University. https://static.uni-graz.at/fileadmin/gewi-zentren/Informationsmodellierung/PDF/dse-interfaces_BoA21092016.pdf.

Berners-Lee, Tim. 1995. “Hypertext and Our Collective Destiny.” Transcript of a talk. Berners-Lee: Talk at Bush Symposium: Notes. 1995. http://www.w3.org/Talks/9510_Bush/Talk.html.

Bouwman, André, and Bart Besamusca. 2009. *Of Reynaert the Fox: Text and Facing Translation of the Middle Dutch Beast Epic Van Den Vos Reynaerde*. Translated by Thea Summerfield. Amsterdam: Amsterdam University Press. <http://www.oopen.org/search?identifier=340003>.

Boot, Peter, and Joris Zundert van. 2011. “The Digital Edition 2.0 and The Digital Library: Services, Not Resources.” *Bibliothek Und Wissenschaft* 44: 141–152.

Borgman, Christine L. 2015. *Big Data, Little Data, No Data: Scholarship in the Networked World*. Cambridge Mass.: MIT Press.

Brinkman, Herman, and Janny Schenkel, eds. 1997. *Het Comburgse handschrift: Hs. Stuttgart, Württembergische Landesbibliothek, Cod. poet. et phil. 2°22*. 2 vols. *Middeleeuwse Verzamelhandschriften uit de Nederlanden* 4. Hilversum: Verloren.

Casey, John. 2012. “Iron Manning, Again.” Teaching resources blog. The Non Sequitor (blog). September 4, 2012. <http://thenonsequitur.com/?p=3527>.

Cerquiglini, Bernard. 1999. *In Praise of the Variant: A Critical History of Philology*. Baltimore: The Johns Hopkins University Press.

Cole, Timothy. 2017. "Making It Easier to Share Annotations on the Web." Institutional blog. *W3C Blog* (blog). February 23, 2017. <https://www.w3.org/blog/2017/02/making-it-easier-to-share-annotations-on-the-web/>.

"Comburger Handschrift - mittelniederländische Sammelhandschrift - Cod.poet.et phil.fol.22." n.d. Library catalogue. Württembergische Landesbibliothek Stuttgart. Accessed February 6, 2018. http://digital.wlb-stuttgart.de/sammlungen/sammlungsliste/werksansicht/?no_cache=1&tx_dlf%5Bid%5D=1880&tx_dlf%5Bpage%5D=1&tx_dlf%5Bdouble%5D=0&cHash=67470d81bfa46dfe0ac4e26d1ea1bf4b.

"Cotton MS Claudius B IV." n.d. Library catalogue. British Library: Digitised Manuscripts. Accessed February 6, 2018. http://www.bl.uk/manuscripts/FullDisplay.aspx?ref=Cotton_MS_Claudius_B_IV.

Croll, Angus. 2014. *If Hemmingway Wrote JavaScript*. San Francisco: No Starch Press.

DeRose, Steven J., David G. Durand, Elli Mylonas, and Allen H. Renear. 1990. "What Is Text, Really?" *Journal of Computing in Higher Education* 1 (2): 3–26.

Dombrowski, Quinn. 2014. "What Ever Happened to Project Bamboo?" *Literary and Linguistic Computing* 29 (3): 326–339. <https://doi.org/10.1093/lc/fqu026>.

Fagin, Ronald, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning About Knowledge*. Cambridge, Massachusetts: MIT Press.

Fecher, B., S. Friesike, and M. Hebing. 2015. "What Drives Academic Data Sharing?" *PLoS ONE* 10 (2): e0118053. <https://doi.org/10.1371/journal.pone.0118053>.

Fischer, Franz. 2017. "Digital Corpora and Scholarly Editions of Latin Texts: Features and Requirements of Textual Criticism." *Speculum* 92 (S1): S265–S287. <https://doi.org/10.1086/693823>.

Franzini, Greta. 2012. "A Catalogue of Digital Editions." Catalogue/Database. A Catalogue of Digital Editions. 2012. <https://sites.google.com/site/digitaleds/home>.

Galey, Alan. 2010. "The Human Presence in Digital Artifacts." In *Text and Genre in Reconstruction: Effects of Digitalization on Ideas, Behaviors, Products and Institutions*, edited by Willard McCarty, 93–118. Cambridge (UK): Open Book Publishers. <http://individual.utoronto.ca>

[/alangaley/files/publications/Galey_Human.pdf](#).

Gitelman, Lisa, ed. 2013. *“Raw Data” Is an Oxymoron*. Cambridge (MA), USA: The MIT Press.

Goldman, Alvin, and Thomas Blanchard. 2016. “Social Epistemology.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Winter 2016. Metaphysics Research Lab, Stanford University. <https://plato.stanford.edu/archives/win2016/entries/epistemology-social/>.

Holmes, Martin. 2017. “Re: IIF and Facs,” June 18, 2017. <https://listserv.brown.edu/archives/cgi-bin/wa?A2=TEI-L;d0eacf69.1706>.

Jansen, Leo, Hans Luijten, and Nienke Bakker, eds. 2009. *Vincent van Gogh: The Letters*. Amsterdam: Amsterdam University Press. <http://www.vangoghletters.org/>.

Kiernan, Kevin, Alex Dekhtyar, Jerzy W. Jaromczyk, and Dorothy Carr Porter. 2004. “Edition Production Technology (EPT) and the ARCHway Project.” *Digidult.Info*, August 2004.

McGann, Jerome. 2015. “Truth and Method: Humanities Scholarship as a Science of Exceptions.” *Interdisciplinary Science Review* 40 (2): 204–218.

Porter, Dot. 2013. “Medievalists and the Scholarly Digital Edition.” *Scholarly Editing: The Annual of the Association for Documentary Editing* 34. <http://www.scholarlyediting.org/2013/essays/essay.porter.html>.

Prescott, Andrew. 2016. “Beyond the Digital Humanities Center: The Administrative Landscapes of the Digital Humanities.” In *A New Companion to Digital Humanities*, edited by Susan Scheibman, Ray Siemens, and John Unsworth, 461–475. Malden (US), Oxford (UK), etc.: John Wiley & Sons, Ltd. <http://onlinelibrary.wiley.com/doi/10.1002/9781118680605.ch32/summary>.

Robinson, Peter. 2004. “Where We Are with Electronic Scholarly Editions, and Where We Want to Be.” March 24, 2004. <http://computerphilologie.uni-muenchen.de/jg03/robinson.html>.

———. 2013. “Why Digital Humanists Should Get out of Textual Scholarship. And If They

Don't, Why We Textual Scholars Should Throw Them Out.” Personal Blog. *Scholarly Digital Editions* (blog). July 29, 2013. <http://scholarlydigitaleditions.blogspot.nl/2013/07/why-digital-humanists-should-get-out-of.html>.

———. 2016. “The Digital Revolution in Scholarly Editing.” In *Ars Edendi Lecture Series, Vol. IV*, edited by Barbara Crostini, Gunilla Iversen, and Brian Jensen, 181–207. Stockholm: Stockholm University Press. <https://doi.org/10.16993/baj>.

Sanderson, Rob, Stuart Snyderman, Drew Winget, Ben Albritton, and Tom Cramer. 2015. “Mirador: A Cross-Repository Image Comparison and Annotation Platform.” In . Indianapolis. <https://program.or2015.net/sanderson-mirador-226.pdf>.

Siemens, Ray, Constance Crompton, Daniel Powell, and Alyssa Arbuckle. 2016. “Building A Social Edition of the Devonshire Manuscript.” In *Digital Scholarly Editing: Theories and Practices*, edited by Elena Pierazzo and Matthew James Driscoll, 137–160. Cambridge (UK): Open Book Publishers. <http://www.openbookpublishers.com/reader/483>.

Sondheim, D., G. Rockwell, S. Ruecker, M. Ilovan, L. Frizzera, and J. Windsor. 2016. “Scholarly Editions in Print and on the Screen: A Theoretical Comparison.” *Digital Studies/Le Champ Numérique* 6. <https://doi.org/10.16995/dscn.14>.

Stutzmann, Dominique. 2017. “Re: IIF and Facs (and TEI),” June 27, 2017. <https://listserv.brown.edu/archives/cgi-bin/wa?A2=TEI-L;205cb507.1706>.

“SWB Online-Katalog.” n.d. Library catalogue. SWB-Online Katalog. Accessed February 6, 2018. <http://swb.bsz-bw.de/DB=2.1/PPNSET?PPN=323970265&INDEXSET=1/PRS=MARC21>.

Thiruvathukal, George, Steven Jones, and Peter Shillingsburg. 2010. “E-Carrel: An Environment for Collaborative Textual Scholarship.” *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* 1 (2). <https://letterpress.uchicago.edu/index.php/jdhcs/article/view/54/65>.

Timpanaro, Sebastiano. 2005. *The Genesis of Lachmann's Method*. Translated by Glenn W. Most. Chicago, London: University of Chicago Press. <http://www.loc.gov/catdir/toc/ecip0513/2005015897.html>.

Zundert, Joris J. van. 2016. “Author, Editor, Engineer — Code & the Rewriting of

Authorship in Scholarly Editing.” *Interdisciplinary Science Reviews* 40 (4): 349–375.
<http://dx.doi.org/10.1080/03080188.2016.1165453>.

Appendix 1: Note on Running the Mirador Demo

The demo takes the form of a Docker image at https://hub.docker.com/r/jorisvanzundert/mirador_review_demo/. Docker is a virtualization tool, it recreates a complete computer system without changing any existing software and data on the computer that it is running on, so that one can safely install and test software. To run the demo, first make sure Docker is installed on your system. Docker for any platform (Mac, Windows, etc.) can be downloaded from <https://www.docker.com/community-edition>. Once installed open a terminal (or command prompt) and execute the following command:

```
docker pull jorisvanzundert/mirador_review_demo:sensitive_turing
```

This will download the Docker image. It is a 1 Gigabyte file, so this will take quite a while and you will not want to do this over a slow connection. Once downloaded recreate the complete environment I created using the following command (all on one line):

```
docker run -i -t -p 9999:80 -p 9090:8080 -p 9099:8088  
jorisvanzundert/mirador_review_demo:sensitive_turing /bin/bash
```

This will drop you in the command line of the virtual environment that I created the demo in, so you will see another prompt but in the same window; something like `root@957398268beb:/#` . Behind this new prompt type the following command:

```
start_mirador
```

The prompt will answer “Starting Mirador demo servers...” and will after a few seconds return to the prompt. You are all set now. Just navigate your browser to http://127.0.0.1:9999/index_reynaert.html and you should see the demo app appear. (I experienced some routing difficulties with Firefox. If you do too I would advise to use Chrome, which is less cumbersome than following the technical annex—which is that you need to put `127.0.0.1 localhost` in your `/etc/hosts` file).

After playing around, you can quit the demo by going back to the terminal that you typed `start_mirador` into and executing:

exit

The application has now stopped and you can close the terminal/command prompt.

All sources are available in a Github repository: https://github.com/jorisvanzundert/mirador_review_demo.

The textual data (a TEI-XML file) for the transcription can be found in the same:

https://github.com/jorisvanzundert/mirador_review_demo/blob/master/var/local/SimpleTranscriptionServer/public/reynaert_transcription_20170704_1529.xml.

As well as the source of the text-viewer component:

https://github.com/jorisvanzundert/mirador_review_demo/blob/master/var/www/html/text_viewer/text_viewer.js

--