This is a transcript for a special presentation co-sponsored by
The National Archives Assembly
and
The National Archives Center for Advanced Systems and Technologies (NCAST)
(www.archives.gov/ncast)

“Archivematica: Creating a Comprehensive Digital Preservation System”

Featuring NCAST Guest Speaker
Peter Van Garderen, President and System Archivist
*Artefactual Systems, Inc. (*http://archivematica.org)

Monday, May 24, 2010 (1-2:00 P.M. EST) - Lecture Room B
National Archives at College Park, MD

Introduction - Jim Cassedy (co-Chair, Archives Assembly, Technology Applications Committee)

Good afternoon, how are we this fine Monday rainy day? Fantastic, I can tell. Welcome to the session. “*Archivematica* – creating a Comprehensive Digital Preservation System.”

My name is Jim Cassedy and I’m the brand newest co-chair of the technology applications committee. We sponsor programs especially on, relating to, automation in archives and workplace applications of technology. I should mention that the national archives assembly – which is pleased to cosponsor this event – is an organization of current and past employees of the National Archives who seek to learn of new archival advances while at the same time advocating for a strong national archives.

It is my great pleasure to introduce Dr. Kenneth Thibodeau of the Electronic Records Archives. Ken has asked me to say little more than that he too is an old fogie, but he has a far more distinguished record than that. And certainly his service with the National Archives is notable and I am happy to introduce Ken to introduce our speaker

Ken Thibodeau, NCAST Director

Thanks, Jim. Good afternoon, everybody. A lot of people don't know I'm really a closet teacher. I started my career as a teacher so I love pop quizzes and I'm going to start today with a pop quiz. What do you get when you cross an archivist with a geek? The answer is you get today's guest speaker, Peter Van Garderen. Peter's a graduate of the archival studies program at the University of British Columbia and he's currently working for his doctorate in archival science at the University of Amsterdam, but he also has a certificate in software engineering. In fact, in addition to that cross, Peter has another cross in which he's both a Canadian and a Dutchman, so you get the combination of the Canadian laid back and the Dutch habit of speaking fast. In fact, I don't know anyone –

Peter Van Garderen

And loud.

Ken Thibodeau

Who speaks faster than Peter. One of my other Dutch friends pointed out that the reason for that is if you have a country that exists below sea level, you want to talk fast before you drown. But anyway, it's a pleasure for me to introduce Peter today. We've worked together for more than ten years, starting with the first InterPARES project where Peter had some involvement as a student and as the project manager. Peter today has what I think is the best job title in the world: President and Chief System *Archivist* – not system architect, but system archivist – I don't know if there's anyone else that has that job description, but from a company called Artefactual Systems, that does development and consulting services in the IT area, specifically primarily for libraries and archives. Among other things Peter's company has developed some software called ICA-AtoM, which they're doing for the International Council on Archives, which is software for description that conforms with the ICA standards on description – the ISAD(G) and ISAAR and so on – and other products. But I don't want to take too much time up with Peter, so I'm going to turn the mic over to him.

Peter Van Garderen

Great, thanks, Ken. Thank you.

Ken Thibodeau

And Peter did ask us earlier to encourage you if you have questions or comments, interrupt him at any time.

Peter Van Garderen

Yeah, I prefer just to have a discussion while we're talking about stuff, so please raise your hand and let's discuss. I guess we've got an hour and a half – or an hour and twenty minutes now – so I'll get right to it.

First of all I want to thank Ken very much for extending the invite. And as well to the National Archives Assembly for asking me to come and speak. So what is Archivematica?

Oh, sorry, before I do that. I understand all as well there's people following along from a webcast. What I'm doing right now is I'm actually running a virtual machine on my laptop. I'm running the Archivematica system from a USB key and unfortunately that means I'm not able to use the web-conferencing software to do a presentation.

So anybody listening on the telephone, I've been told you've got these links already. I don't know if – the presentation slides, the workflow instructions as well as a whole series of screen captures from the system. So between all of that you should be able to kind of follow along and see what we're talking about here back in DC.

So, Archivematica itself is an integrated suite of completely free and open source tools that allow users – you know typically archivists – to process digital objects from ingest to access and apply format-specific preservation policies. The Archivematica project too adopts an agile software development method, which one of the key components of that is having time-based release schedules so we release no matter what on this date – whatever we've got is what goes into that release.

And so it forces a certain discipline in the software engineering process. And what that means is that we've had six very rapid iterative releases already over the past fourteen months leading to the 06 alpha release which was done last week – about a few days ago – and that's what I'm demoing to you today.

Each iteration in an agile development method leads to improved requirements, obviously improved software, as well as updates to documentation, improving the scope, depth and breadth of all of those, as well as development resources – the resources that are available to developers and people working with the software. So the big part of that is that we're not going to get it right the first time; we just pretty much assume that.

It's basically the exact opposite of this would be a waterfall methodology where you try to get all of your requirements just right, you spend a year writing a giant spec document and then you go and try to make it perfect the first time out.

So, the agile development methodology really is very well suited to the digital preservation field, where for all intents and purposes we're never actually going to finish making our system because the technology we're trying to preserve is constantly changing and the technology we have available to preserve digital objects is constantly changing so we try to just accept that right from the start as a principle.

Where did it come from? Ken mentioned my company Artefactual Systems. I've been consulting now for about ten years. After InterPARES, I worked on the InterPARES project and that's where I met Ken, I went and started my own consulting business doing electronic records strategies, digital preservation strategies, but more and more I became very interested in the opportunities for open source to be used in archives for a number of reasons which I'll come back to at the end of the talk, which I think – There are a number of reasons why I think open source software is a good fit and a right fit for the archival community.

So over the past few years the main focus of the company has been developing and supporting open source software tools for the archival community. ICA Atom and Archivematica in particular.

The City of Vancouver Archives is one of our clients and they essentially started on the path as one of the smaller – like a city, a medium sized archival institution essentially – wanted to implement a digital preservation solution because of the same problem that all archives have: 90% of the world's information is – over 90% – is now being produced in digital format. That's tomorrow's archives; tomorrow's archives are here today – we've been creating them for the last 20 years.

Unfortunately, a lot of institutions don't have very practical solutions in hand yet; they don't necessarily have the budgets of a NARA, let's say, to go and implement an enterprise system. So it's a very practical need to transfer records from electronic records document management systems as well as now to transfer over the electronic records from the Vancouver Olympic Organizing Committee. We need to be able to do something now. Like, what can we do today?

So with that in mind, about two years ago we started with the premise that there's enough open source – there's been a lot of research, I mean InterPARES kicked off in '98 I guess – and there's – sorry? Well if you count the UBC PROJECT **–** there's been a lot of research around for a long time. And there's also been a lot of one off, ad hoc sort of tools have been created over the last few years as well that let archivists deal with certain particular problems – parts of the pieces of the digital preservation problem – but nowhere if you're a small, medium-sized institution could you say two or three years ago, "hey, where's something I could plop down and start doing OAIS compliant digital preservation today?"

Part of our theory was that there's enough of these tools around, we should be able to stitch them all together and to create one comprehensive digital preservation system. And that was actually the premise of another paper that was published shortly afterwards by the UNESCO Memory of the World Subcommittee on Technology, which essentially had the same premise, saying that there's enough of this around, can't you put all this stuff together to try to make you know a free and open source archival description – sorry archival digital preservation/digital archive system?

So over a period of time we got in touch with UNESCO Memory of the World committee and they became a sponsor for this project as well to take the technology we were developing hand in hand with the City of Vancouver and open source it and make it available to the community.

Another client of ours, the International Monetary Fund Archives was on the same track as the city of Vancouver and going through a digital preservation strategy and essentially over the last half year has been working with us to do a proof of concept project using Archivematica and contributing back to the Archivematica code base as part of the time and resource that we're investing in it. Essentially, work out their full spectrum requirements, so doing an early iteration essentially as a proof of concept project to get ready for their own full implementation of a production ready digital preservation program.

So I mentioned the OAIS, presumably everybody is familiar with the OAIS model and I don't have to go into any more detail for it. Key concepts there – and again it's a default language – we talk in the digital preservation world and it's the default language we use within the Archivematica project.

Key concepts are the mandatory responsibilities, the functional entities, the information packages – the submission, the dissemination packages and the archival information packages, the content information, the preservation, the descriptive information packages.

And then the actors, the consumers, the producer, and the management that plays certain roles in the system. We focus specifically on the functional entities as the things that describe what an OAIS system needs to do first. And this is typically on a high level OAIS diagram – you see when people talk with OAIS, of course when you drill down into each functional entity it gets a lot more complicated. And who has actually peeled through and read OAIS from beginning to end? Okay, and don't – people that work for Ken will have. This gives me a good idea of who's the audience as well then.

It's a beast, there's a lot in there, it's a great standard, it's really been one of the major advancements – one of the first things we needed in the digital preservation community so we could all start talking the same language, put everything in the right box in the right place, figure out where components are going to integrate.

But the standard itself is not perfect either, and there's lots of inconsistencies when we did our analysis. And there's just a lot to it so we need to be able to filter this down, to create a simple system, to be able to say "in order for us to be OAIS compliant – which everybody wants to be – what do we have to do?"

So we started by first of all using a use case methodology in the City of Vancouver project and breaking down each of the functional components and doing a detailed use case analysis, breaking it down hierarchically, to figure out what comes in, what goes where, and start translating the language a little bit to something that's more practical and that fit more archival business – standard archival business processes.

Those use cases still weren't enough, they were still too abstract at that point, so then we started developing UML activity diagrams, which is this you know very specific workflow methodology. And those actually went through three iterations as well because the pure adaptation from the OAIS model still was too abstract for us to be able to apply it directly to match the requirements of the archival businesses processes in the City archives.

So the third iteration is one that we could use as the baseline system requirements for development and that's what we've been using over several iterations of the software development now. It gets updated each time we do the software development because, again, the actual deployment of software – trying to integrate it into some case studies, use case studies – inside the archival institutions and the limitations of the technology itself as they exist today in 2009/2010, actually influences the requirements because it'd be great if we could theoretically we can say it'd be great if we can do this, if we can't do it, what's the point?

So it really is the focus using agile methodology is really to be as realistic and as practical – as pragmatic – as possible to get something working today that still meets best practices and standards. So in any system releasewe end up with a set of system workflow instructions that essentially take these functional requirements, which say specifically what the system needs to be able to do, and because we don't necessarily have all the technology – it isn't mature, it isn't necessarily fully integrated yet – there may be technical gaps, it may be simply development gaps we haven't had time to get to.

But what we want to do is be able to say even with the very first iteration one proof of concept, we were able to identify specific steps that either technology, a technical tool, or archivist performing a manual step would complete so that from point A to B we would have fulfilled all the OAIS requirements and that's really for us is a critical, again another critical principle in the project.

That the system is not just technology, it's an integrated whole of people, procedures, and software. So that with every iteration, we're confident that, if that's all you had, if development stopped today, we are convinced that we could still take those instructions, take that technology, and complete OAIS SIP processing, get the AIP, get it out to the consumers, and still be fully compliant with the OAIS functional model.

So this is the latest set of the workflow instructions that's available for download. I see somebody photocopied for you a page or two of it as well. And this is what the archivists do, the user would use to then follow along and actually complete the steps within the system. With the white boxes being the automated steps and the other being the manual steps here.

So we took each of the steps in the process and we mapped them, we essentially made them a what's called a micro-service. The micro-services approach to digital preservation is turning out to be quite a legitimate and quite effective alternative to a large repositories…

[somebody's phone…audience giggles]

Peter Van Garderen

I was saying that the micro-services approach is turning to be quite an interesting and I think very effective alternative to large scale repository based digital preservation systems. Where the system is built around the capabilities and of the technology stack first. It starts with the repository stack.

It starts with … essentially the framework stack which almost always is J2E, like, you know, JBoss servers, that kind of thing. And then works its way forward from there, saying this is the technology we have, how are we going to meet the requirements?

Alternatively, the micro-services approach says these are the actual granular things that need to happen along the way of an OAIS workflow and here's some tools, or here's some manual processes we can map to that to get this job done.

So our – and this is over the last couple years, the California Digital Libraries has been doing a lot of work to kind of standardize the micro-services approach and they've published a number of specs in collaboration with the Library of Congress as well, and I know just recently iRODS itself (one of *the* research collaborators here with NARA) has also started to define… again, it was one of those things where we were already doing it that way, it just didn't have a name yet, and iRODS was already doing it that way as well, it just didn't have a name yet.

Now we've got a proper name, we've got a lot of theory around it now that helps us kind of essentially establishes a legitimate alternative to repository based digital preservation systems.

Okay, what does that mean?

Rita Cacas (NCAST Communications and Assembly President; monitoring web attendees)

[Peter, there is a question]

Unknown Male 1

Hello? I was just wondering. This seems to be a nice micro-services idea, and this seems to be a nice immitation of modularity in the system. How much have you guys tested that as a proof of concept in terms of swapping out different tools for the various micro-services. Is that…

Peter Van Garderen

Yeah, it's working out quite well so far. And that's obviously - yes, yes, thank you for bringing that up. That's one of the key – again one of the key principles is the idea that we're not married to a giant technology stack, so that if we have one tool providing say normalization services, or providing unique identifiers, if that tool, for whatever reason, we have a better tool or that tool has limitations, we should be able to swap it out of the stack and put a new one in and carry on processing like we did before.

And that's – we rarely had that with the Xena. Xena is a normalization tool put up by the National Archives of Australia and our very first iteration started using – we used Xena to do our normalization of office documents to OpenOffice format. And there were certain limitations with that tool that we just were unhappy with and we ended up actually swapping that out over this last release and putting in, just using it's called the UNICOM, it's a service engine that uses the OpenOffice engine directly.

So Xena was using OpenOffice as well, but it introduced a whole bunch of layers and wraps around it that wasn’t really working with our workflow. So in fact we did that, we pulled out Xena and we dropped in UNICOM instead. And that’s one of the beauties of the micro-services model. And we did that in like two days, with a bit of testing and so forth.

So this is definitely I think one of the strengths of the micro-services approach. And particularly for our problem in digital preservation where again, the technology that’s creating the digital objects that we’re ingesting and the technology that’s available to us to manage that stuff is constantly changing.

And essentially what we’re trying to do with the Archivematica project is not just develop software but develop kind of a methodology that micro-services help to kind of define that theoretically but practically as well we want to develop a methodology that makes it very easy for us to constantly be adapting to that change.

So what Archivematica is then is a system. And again the California Digital Library guys just did a great paper they’re going to present at the Open Repositories in Spain in the summer where they talk about the Unix pipeline that affords – this has been around for a long time already, since the 70s, is this idea of the Unix pipeline is essentially you take the standard output of one process and you make it the standard input for the next process.

And that’s essentially what Archivematica is. It’s a classic mixed pipeline of OAIS defined micro-services and we map, you see the micro-services at the top there? Each of those is mapped to one of the existing open source tools that we’ve integrated into the application.

If you go to Archivematica.org/software, it’ll give you a listing of all the tools that are in the current release as well as a link to all their licenses because of course we want to make sure all the licenses are compatible so that we can continue to give the entire system, the entire stack, away fully free and open source.

And so we’ve got digital information objects working their way through the various micro-services workflows and simply being passed on from one process to the next using standard Unix pipeline approach, where we’ve got a combination of Python scripts and BASH scripts that simply move the stuff along, queue it, make sure it’s locked so that we don’t get clashes in the pipeline and so forth, making it possible, again, to run the entire system from a USB key for demonstration purposes, obviously a 4 gigabyte key is what I’m running the software off right now.

And what we end up doing is we end up bundling, we've been working with the Ubuntu operating system – we went to a specific flavor of it that uses the XFCE desktop called X-Ubuntu – but it's essentially the full power of Linux, the Linux operating system, that we're building and integrating this off of.

It gives us a nice user-friendly desktop to work from and we're bundling all these tools on top of it and we're allowing the user to come in and bring in their digital objects externally, so either through - from a network directory like the DOD standard requirements of 50152 what is it? I think I got it right – is that right?

Ken Thibodeau

15 dash –

Peter Van Garderen

Yeah, you know the one. It talks about the transferring records, making them available on the network directory. So Archivematica would have a watch directory. It would see the submission information packages that appear in the network directory and then the archivist or the system would pull them in and start triggering – and trigger the workflow process.

Or what we're doing in the City of Vancouver for example, the Bannock records. They don't want to give us their server, so we have to go in with a bunch of one terabyte hard drives, use CRC tools to – we're using RSINK – to get the stuff off their drives, confirm that we've gotten it on the hard drive, bring it back to the City of Vancouver, plug it in to a transfer station, and use external hard drives to then move it over and kick start the workflow process.

The entire system is packaged as a virtual appliance that combines all of this – combines here we've got the operating system, we've got all the tools, we've got the integration code living as one system. We can make a virtual appliance that runs inside a VM player like Virtual Box or VMWorks, so we can put it down on servers.

At the IMF for example, it's running off a VMWorks server. I can run it off a bootable USB key like I'm doing now as a demonstration system, but we can make much larger USB keys, so I'm planning a little project on my holiday to do my own family digital archives using a completely USB-run system.

You can then also put a dedicated PC and servers. So at the City of Vancouver Archives, they're actually more this set up here where we've got about five or six workstations now that are networked in a totally private area network so that there's no issues with security or with external records coming in until they're actually ready to go into the network storage. And we're able to use the same disk image that we use to create the virtual appliance and the USB key as a completely bare metal install on the machine.

So it’s just here’s a machine, boom we install it on there. It’s got 4 gigabytes of space, we blow it up to a terabyte that’s available and now we’ve got a fully ready, functional Archivematica node. And then we can connect various nodes over the network.

So we use that to replicate itself over the network, and then we can – the archivist, there’ll be two archivists working at the beginning, so they can do their own SIP processing at a time, they can share archival storage, they can share the access system by a network connection. That’s all happening from the same virtual appliance.

Okay, so, when the user boots up from the USB key or starts the machine that has a dedicated Archivematica install, what they’re going to see is a desktop. And again, part of our agile iterative approach was that we can either interface this thing or we can use the operating system or we can use a file manager as the --- we’re moving files through a pipeline so a file manager makes a perfect user interface for this. And so we’ve got a number of scripts imbedded in the OS inside the file manager to help the archivist move stuff from one place to the next.

Over time, as we get the system gets more sophisticated, a lot of this will actually move to a web-based dashboard. We’re already starting to develop that right now. So the archivist primary interface will become a web-based dashboard but for the time being it’s the archivist works with the desktop interface. So the first folder is the receiveSIP folder and this is a watch folder.

You notice the arrow and that’s just to simulate if this is a live system, we have it watching a network directory through a SIP share or an NFS share, web database GP, there’s a number of ways we can watch external directories and have the notification come up to the archivist – when a SIP has arrived – from external media, or from the network drop.

See here I’ve got three sample SIPs. And we’ve only got time to go through one so I’m going to go to the one that’s got more detail to it. And so here the use case we’re simulating here is a submission information package coming from the electronic records document management system.

So it has its retention schedules, every year, every month, it has a number of records that come up for archival preservation. Those, the system exports and puts in this specific drop space.

The other examples: images, multimedia, maybe the archivist has gone and done a retention evaluation of the shared directories. I mean that’s one of the first places these types of pilots typically start before we go through the more sophisticated system integrations.

So the archivist makes a copy of the SIP. And that’s of course, for example, in the City of Vancouver, we have two backups so that if anything happens during the processing we can go to one of the external copies, we can go to another backup copy because we’re not going to sign off and destroy this SIP until we’re happy, we’ve got a fully ready AIP and DIP loaded to the access system.

So as that package that comes into the system, it gets converted into an Archival Information Package – an AIP, what we call “apes” – and then the dissemination information package is the package that’s – the information package that’s made available to the end user.

And that’s called a “dip” for short. And all of them are essentially a combination of the actual digital objects and the bit streams as well as any metadata that describes, i.e. technical metadata and descriptive metadata.

The combination of the two makes an information package and depending on where it is in the process it’s either a SIP, an AIP, or a DIP. So they come in as SIPs and again, keep in mind this is designed for archival business processes so we have a review SIP step where the – either the system at some future iteration will actually go an check the manifest to make sure it’s compliant with the submission agreement that they’ve established with the producer, or the archivists themselves can do a check and make sure that the metadata that they were expecting was the right kind for this type of system transfer, do a ND5 check on received to make sure that the files weren’t corrupted between transfer.

And so here for example is… we’re starting this SIP - the descriptive metadata with essentially qualified Dublin core, so in this particular scenario, we say that the electronic records document management system gave us some metadata that we’ve mapped to the appropriate Dublin core elements.

If the system didn’t come with the metadata at this stage then the archivist could right-click and add a blank Dublin core XML template, which then they would fill in with any descriptive metadata they have.

We expect to make – add EAD as an option and we very likely are going to have to have some kind of – we’re looking at some other projects that have basically, like California Digital Library, the Tipper Project – essentially qualified Dublin Core doesn’t give us enough elements to capture everything we want to handle transfer and appraisal.

But then the other option is of course, at the City of Vancouver, they use the TRIM document management system and it gives us a whole bunch of metadata – a lot of it is quite useful obviously for description and so forth.

And so what we'll do is we'll end up taking the custom metadata that comes from whatever target system there is and we'll end up bundling that as well in what ends up being a METS XML profile.

So for all intents and purposes we'll say that this one checks out and we're happy with it. So we move the SIP to quarantine. This is a practice that we've adopted from the National Archives of Australia, the theory being that when you take in a bunch of records that have been transferred, there very likely could be a bunch infected with a virus.

If you put the typical virus tools that don't have the definitions updated until a few days later or a few weeks later as the threat becomes known and they put a patch in for it. So the idea is that – I think in the national archives it's really – they're standardized at thirty days. So you put your stuff away for thirty days, when it comes out you run a virus check, you update your virus check tool, and you run a virus check on it.

Assuming everything clears, you start posting your records. So that's a step that we've incorporated into Archivematica as well. For demonstration purposes, it's just set to a minute right now. So it's crunching away. And as soon as the minute's up it's going to start processing and preparing the SIP for appraisal.

And so what it's done already is it's assigned a unique identifier – obviously we have to have a unique identifier assigned to all our information packages, to package itself as well as all its contents. There it goes, it's just finished quarantine, so now it's starting appraisal. Notice it says quarantine completed.

And now it's extracting packages, so one of the things we found in very early iterations is that you end up with lots of zip files and – you don't know what you're going to get when you start pulling stuff off a shared directory for example. And part of our design goal is to be able to handle anything that gets thrown at it. So one of the very practical things we have to do is clean up file names and extract packages.

The other thing is you have ampersands, weird combinations of characters that are prohibited in Unix where we're running all these tools. So we do a cleanup of the file names, of course we keep a log of any file name changes that we're making.

So our unique identifiers that we're using are UUIDs – Universally unique identifiers, which we think is a very simple and elegant solution to the identification problem.

There's been a lot of effort put into creating global registries for unique identifiers and I think using UUIDs is actually a much simple, more elegant solution. It's an algorithm that's available as a standard Linux utility tool, it's available for every programming language available – it's an algorithm that makes it very, very, very unique – it's very difficult to replicate.

I've got a little quote there, it's something about – let me see it – it's very unlikely in our known space and time that we're going to create a duplicate UUID, which totally eliminates the need to go register somewhere globally because if I use this tool to create a UUID over here, if you're creating one over there, they're going to be unique – there's no point, they're not going to clash. Then we can use things like archival resource keys to like put name spaces on them and so forth.

Okay so I'll get – so the appraisal just finished, you saw the notifications popping up as it's going through and we'll take a look at it in a second. One of the critical things we did was, after we did the - assigned the UUID, we checked the checksums to make sure whatever got sent to us was actually what was received.

We extract the packages between the file names, and then we started identifying, validating, and extracting metadata from the digital objects, so this is one of the places where the actual practical tools first started to emerge about five years ago. There was the project out of – it started at Harvard I guess – was JHOVE – the National Archives of the UK had DROID, and the National Library of New Zealand had a Metadata Extractor.

And these are tools that were - the whole purpose was to do identification validation, so yes you sent us something called blahblahblah.mpeg, but is it really an mpeg file? Is it really .doc? Is it really a Microsoft Word document? So what these tools are designed to do is actually look at the bit streams, look at their headers, and say yes, this in fact a document, and then validate it against published standards.

Say this particular document actually meets this spec and it is a valid Microsoft Word document, a valid MPEG file, and then typically most digital objects have a lot of imbedded metadata that's either explicitly in the header or we're able to pull out using certain tools so there's a whole wealth of technical metadata that we can pull out of existing digital objects that can help us with arrangement and description, with authentication and so forth. And of course, digital preservation –

it tells us what the file formats are and what we need to do with them. That’s low resolution there.

So the very first iteration of Archivematica was essentially taking these tools, which is – one of the things, most archivists that have been involved in digital preservation have heard about these tools, but it’s just actually getting these installed, like on typically they have a Windows desktop that’s like locked down by their network administration and they can’t even get their hands on the tools.

So our very first goal is to just get these tools on a platform that we made alive, you can easily pop it in anywhere and work with it, so archivists can actually start working with the tools they’ve been hearing about, and going to conferences and hearing about.

And so we did the same thing and presume that this is going to be able to give us the very first step in the process – we can do identification validation. The problem we’ve run into – and this is all on the Vancouver Archives Project wiki and there is a test set of about 30 different file format types – this is only the first four or three.

We found very very conflicting results between all the various toolsets and the end result is they’re all very early stage tools still, they’re all excellent projects and they’re all trying really hard with the resources they have but they’re actually not necessarily very reliable quite yet. If you compare all the various test results. And this is actually a real problem still I think in the digital preservation domain is getting good reliable identification validation and comprehensive – that’s the other problem.

It’s like, you know, we can target a few specific file formats, but to get all of the thousands of potentially known file formats that we’re going to have thrown at it, to be able to identify, that’s just a logistical problem. So that was, we kind of hit the wall on that one and fortunately out of nowhere came what I think is probably the most underrated digital preservation project out there today, is –

[Rita: Mark Conrad has a question]

Mark Conrad (NCAST Research staff in Rocket Center, WVA, and co-Chair, Archives Assembly, Technology Applications Committee)

Peter, I’m looking at your side and you say that the services validate format, at least in the case of DROID, all it’s doing is doing stringchecks within the header, does this other tool do actual validation?

Peter Van Garderen

No, not very well. And I think the file tool does. But you're correct Mark and it's really only doing – it's trying it's best at identification. And validation – for us the defacto validation is happening when we throw it through the normalization tool because the normalization tool will either choke or it won't. And for us right now that's the only real reliable way to do validation.

So, you're correct in that it's really only identification that's working, although the tools profess to do validation, right? And it depends on the file format. Again I think there's a very limited set where they will validate, but again that's a known limitation of this whole area, this whole tool site area right now.

Mark Conrad

Okay, thanks.

Peter Van Garderen

I mean in your experience, is there a tool that does that properly right now?

Mark Conrad

There are individual tools for individual formats.

Peter Van Garderen

Exactly

Mark Conrad

But that's it.

Peter Van Garderen

Yeah, so that's the problem. So I think FITS is a really really good start. Are you familiar with FITS, Mark?

Mark Conrad

Yup

Peter Van Garderen

And so I think this is a really great start on starting to solve this problem and essentially what it is is a lot like Archivematica – it's a wrapper around existing tools. And what it does is it takes the best practice tools we have so far as well integrates a few other known Unix utilities that do this kind of work and what it does is it will output a report, which is what's happened here after we fed them through FITS.

So for each file, we get a FITS log report and it essentially reports on the various conflicts. So it tells us what each – first of all it tells if there's consensus and unfortunately quite often there's not – and again these tools are still very raw. Sometimes very simple problems, like one using – it's just a namespace issue like where you're not using the correct file extension and so forth and so the two consider them to be two different things.

There's issues with identifying the correct version and so forth. But it does actually, you know in the end what it does is it takes all the output from all these tools and publishes it in a report and what you can do is you can go to the FITS tool and tell it I trust this tool more than the other or I want to base whatever I do I want to give this tool a higher ranking or I want you to use this tool last to evaluate the conflict and use it as the deciding vote.

So it gives you more flexibility in mixing and matching, but we haven't ourselves, we haven't gone really that far yet in doing that. But it certainly is to my opinion is definitely the way forward and we're relying heavily now on the FITS project which is in very active development and there's a lot of work being done.

This is out of Harvard University. And I think it was really – it saved our butt because we had a serious problem with how do we go forward. We basically were at the point where we were going to have to develop our own FITS tool.

But of course this is the beauty of the open source model – somebody has gone, made and developed as open source and boom we can integrate it into our project and we can move onto other things like finishing the work flow automation.

Rita Cacas

[Peter, we have another question…]

Richard Marciano, UNC/DICE group

Hi, it's Richard Marciano.

Peter Van Garderen

Hi Richard.

Richard Marciano

I had a question regarding your workflow framework, since you explain it as sort of being a generalization of Unix pipes, typically that means that there's no space for global state information and that you're passing information from a previous stage to the next stage. Could you comment on that and say a little bit more about how all these tools actually coordinate and if there's any notion of state that's built into this framework.

Peter Van Garderen

Yeah, that state is captured in the log files. So the state of the actual digital object never changes, right? We will normalize it, which means we make a copy of it right on top of this box here like here – your disembodied voice coming from a black box on the table.

The digital object never changes state and that's the whole point. We want to authentically preserve it. The only time it changes state is when we make a normalized preservation or access copy of it.

All the things that are happening to the object and all the information we're able to pull out of it – and again all the things that we are doing it to it – those are being captured in log files.

In fact, if you look at your screen captures, there's probably a couple of shots of the log file directory and including the FITS output, including things – the log for when we extracted things and so forth. And that's later on sort of gets imbedded back in a METS doc, and we're still working on getting all of the log file into a METS document.

But the standard input output is that we pass - the output typically is the file - and then we pass it on to the next process where something – again something either pulls information from it or does something to test it or in some cases again logs – so I guess I lied.

We do change the file name so you can argue that's a state change, but typically I guess the separation is that the object just keeps getting passed through the pipeline and the actual information about things that are happening to it are being captured in the log file. Does that answer your question?

Richard Marciano

Yes, thank you.

Peter Van Garderen

So, here's a log for the UUIDs that are being assigned to the objects. Then we check the virus scan. So I didn't mention either that we're using the ClamAV tool which is being run on most – I think the majority of email servers worldwide right now to, you know, essentially because a lot of people pass attached documents, so there's a big problem with viruses coming through email servers.

So it's a very active and again fully free and open source project. So Archivematica, as long as it's got an internet connection, it's constantly pulling down the ClamAV virus definition updates.

Okay, so our next step is the archivist now has all the information in front of them. – Oops, I closed the, uh – So another step in our workflow now is that the archivist is getting ready to appraise the SIP, so again this could be there's this rule that says don't bother if it's coming from this kind of system, or the archivist could actually manually go in now and take a look at the objects.

They know there's no virus issue. Again, the virus issue, for us, it's almost like a feel good factor for us. It's more of an issue when we pass the object back to the consumer. Most of the viruses that are being written aren't going to be an issue on the Linux system that we're using to manage Archivematica.

It's got to be some pretty tricky viruses to actually cause a problem for us inside the system, but it's more an issue and a courtesy not to pass infected objects on to the consumer if they ever ask for an AIP with the object in it. So the objects have been virus-checked, we've pulled as much metadata as we can about it. That information is available right now in the log files.

And at this point the archivist could make decisions about whether certain objects actually meet their appraisal requirements for historical value, informational value, for legal value, or they can assess the technical capabilities of the archives to preserve – whether they're happy with the default normalization policy that's going to kick into effect depending on the file formats that were identified – whether they want to change those at this point. Any number of things could be happening in this appraisal stage depending on the institutional policy.

One thing, we didn't do – one thing we're leaving as optional right now is we unzip the zip files, so you notice here those got extracted, but we kept the original zip files as well. So here's an example of where I might say, okay, we don't as a policy we don't actually want to keep those, we'll trust that the extraction worked properly.

And you'll notice that there's a manifest for the SIP itself, and here's where we're starting to assemble all of the information: the descriptive metadata that came along for the ride in the Dublin Core, XML that was included is in the descriptive metadata section, we have the AMD and file sections describing the actual contents of the information package, and we're starting – we have the starting bits of inserting PREMIS metadata into the administrative metadata section, in this case the UUID and original file naming if the file name was changed, and something for 07 is to get all of the log metadata into PREMIS events, it's a major deliverable for us that we're working on in 07 release.

So, I'll take the – let's say I'm happy with it as is, so I'll drag the office doc SIP and move it over to the prepare AIP folder. So again it's a watch folder – as soon as the file hits it, it knows to send it to the next step in the pipeline and essentially the message we're getting is it's normalizing and it's now converting the files to preservation and access formats.

And this is again one of the critical components of the Archivematica system is the – I'm getting a message now that the Archival Information Package is getting prepared. So essentially what we've done is our default preservation policy in the Archivematica system is to use normalization.

So essentially I think after all these years I think we're still down to basically four primary strategies: it's technology preservation – keep all technology running to keep your information accessible on the system that created it; emulation – so recreate the operating system application environment in which the digital objects were originally created so at some point in the future, you could emulate that environment and bring the original digital objects back so that people can read and use them at that point in time; thirdly it's migration – it's that take all the stuff and keep a close eye on it and if you think that some of the file formats are at risk so that people will not be able to read and use them at some point in the future or in the present, then go and migrate those to a format that you can view and use them on; or normalization, which says right at the point that we get them let's figure out what's our best bet for long term preservation and convert the files to that format and make that our primary preservation format that we preserve it as.

So Archivematica is for all three of the latter, we support emulation, we support migration, we support normalization. So we'll always keep the original file format – we've got to cover our bets so ideally the emulation technology advances, we'll always be able to pull up the original object and be able to emulate that. Migration is, well, we do normalization by default, so as stuff gets ingested, we see a file format, we map it to a preservation file format. If at some point in the future – five, ten years down the road – we say okay, we're not happy with TIFF, we're not happy with MPEG, we're going to migrate those?

Then we have a migration alert and the system would use the same normalization process to then do a migration and convert those files to the new preservation file format. But our default policy is normalization and that is to convert upon ingest – and again that is the default strategy used by the National Archives of Australia as well. And part of their rationale as well is that we figure we’re going to have limited amounts of time to actually pay attention to these SIPs as they’re coming in and it’s probably at the point of ingest that we have the attention span of the archivist.

After that we just have such a large volume of information that we’re never going to necessarily have the time and the resources to go back and do the detailed migration analysis and so forth. So the point is let’s get as much done as we can at the point where we’re actually paying attention to this system, at the point in time where we’re actually bringing objects in.

So a big part of our work over the last year is to define our media type preservation plans, so essentially to take specific file formats that we’re going to expect to be getting in – and again we started with the City of Vancouver and the IMF as our case studies and seeing the typical files that they’re getting that they’re expected to ingest over there – and trying to group them into media types so that we can standardize and have a preservation file format for a specific media type.

And the other thing we’re doing is we’re creating access formats. So the OAIS model specifies that you go and get – you create a dissemination information package when the consumer requests it and you go and you get an AIP and you convert it into the DIP, which just isn’t very practical I think. And one of the things – the Family Search guys from the Latter Day Saints did a presentation – they probably have the highest requests per day for any kind of we’ll call it digital archives system.

They’ve got all the digitized genealogy records and so forth – they’re getting millions of hits a day and they’re trying to - they’ve applied an OAIS reference model to their own digital archiving system but they’re saying this whole thing of pulling the AIP off every time you get a request on a website is just not practical and it really isn’t.

So the idea here is that we anticipate what we expect the good access format to be for the particular file format at the point of normalization we create both the preservation format and an access format. And that access format gets cached in the web access system so that it’s going to be able to take care of 90% of the requests we get from consumers will be met by those access formats.

Of course we still want to have a process in place where they can request the AIP and get at the original file format for example. But again, we expect those to be the very minority of cases. So what we're doing is, well we're defining our media type preservation plans, and again trying to do it in a very practical way, it's just sit down, figure out what file format we have, do our research on what's a good preservation format.

And that's been difficult to do this – a lot of people aren't necessarily publishing or making, you know there's bits and pieces around, but it's not really in a systematic or structured way are people publishing their format policies. And that's been a bit of an issue. So we're doing a lot of work to try to assemble all that information. And access for preservation formats it has to be in open format, so an openly published format, ideally managed by an open – some community committee or community process.

It has to be a format that is able to preserve most of the significant characteristics of that file format and, in our case or a limitation we put on our own project, is that we have to have a free and open source tool that can normalize that format. If we don't, then it's not really an option for us because one of our design goals is to be able to give this system away as a fully free and open source system.

And that creates some problems for us, for example, in converting Microsoft Office documents without any kind of noticeable loss. So as an example, we've got – so this is all on the Archivematica wiki, so the page I'm showing you here, if you go follow the link to media type preservation plans, it shows you the overview of the file formats, what the preservation format is, what the access format is, what tool we use to normalize it.

You can drill down to each individual file format, there's a link to its PRONOM information until we have the Universal format – UFDR – PRONOM's the best thing going we've got right now for kind of giving unique identifiers to actual file formats. There's a link to the significant characteristics, so for audio, the Florida Digital Archives has done a lot of good work on identifying essential characteristics in a very practical way, so we're borrowing a lot of their published information about their essential characteristics analysis.

And so what we do is for each media type we talk about what we consider to be the core essential characteristics – these are the characteristics that we have to be able to preserve when we go from the source format to the preservation format. Then we do actual tests so we use tools – FFmpeg makes a great tool for doing audio and visual conversions, very well established in the open source community – so we run tests, we compare, make sure the tool does what we expect it to do.

And then we set up an actual media type preservation plan for that particular media type. And this is all a work in progress, and it'll be a work in progress for as long as you know the Archivematica system's around, which is going to be for a long, long time. But the point is that it's all accessible. We're making all this research, we're not saying we're perfect, we're just doing the best we can and every time we make a decision we try to document where we're getting this information from, whether it's from us running our own tests or pulling it in from other sources.

And then we convert the media type preservation plan into an actionable configuration file. So by that we mean when you go into the Archivematica system, there's a folder called format policies and for each file extension, there is a very simple format policy that captures the decision made for that particular release on the preservation and access format.

So if you go to WMA, you'll notice that the very first value, it says inherit audio. And so what that means is that we can have rules specifically for when is a media audio here, but in this case we said okay the media object belongs to audio and we're going to inherit the preservation rule we've established for audio. And likewise what we could do here is we could then simply, if there's multiple variations of an extension name, we could simply put the one that's actually containing the rule.

And here we go to the audio XML file and we've got a very simple definition of what our access format, our preservation format, and then the actual conversion command that Archivematica will pull out and apply to the normalization tool. So institutions can actually go right into this XML file if they want to change bit rate or other kind of values that they want to change. They're able to alter their own normalization and preservation format policies by simply editing an XML file.

So we're doing our best to come up with default policies and say this is what we think is appropriate for Archivematica but we're not locking anybody down to saying this is the one that you have to apply to your institution.

[How are we doing for time? Looking good so far, okay]

And I think this is actually a very practical contribution that we're making in the Archivematica product. Whether it's useful to anybody else, well time will tell. Again, and our own experience is that it's great that we've got PRONOM out there. Like, you know so we can have - we essentially have a registry to say definitively what's the correct name to use, the correct extension to use for a particular file format.

We've got things now like the PLANETS project has the PLATO tool and the test bed which lets you do the similar testing to what we've done but in a much more – you know it's a bit more heavy duty in that it's done within a JAVA framework; if you want to test a tool, you've got to build a JAVA wrapper for it. Us we just go out and get it and it's there.

You can do that - I can add it right from this USB key right now to do my test and record it in a wiki. It's slightly different approaches to the same problem – two compatible approaches. So I guess what we'd like to do over time is actually contribute and make this useful – with those gaps we can contribute some of the tests we've been doing to that test bed repository. But after that, so you've done tests, you've said okay, these are finished tests and this is good, or we lost some characteristics here, we lost some characteristics here.

At the end of the day, the small to medium sized institution just wants somebody to tell them what do we do? Like what do we do now? That's all great, very theoretically sound, but you haven't told us how to solve our problem. I got thousands of objects coming in; I need to do something with them. So, for what we're doing here is we're just explicitly publishing the preservation format for this iteration of Archivematica.

In this case, in these rules here, we also want to – what we're going to do is establish an external RDF registry so that the system itself can go create the external repository. And this is how we would trigger any migration processes. So if we look back at the high level architecture diagram, you'll notice that under the monitor preservation service, it's going and checking the format registry and all the format registry is, essentially, is an online repository of those format policies.

And if there's any changes for a particular file format, the Archivematica itself will get that information back and it can trigger process using the actual normalization rules to do a mass migration. At the same time, other projects can actually query that online repository as well.

And there's some really interesting work being done at the University of Southampton with the Preserve2 Project where they're using RDF graph technology as well to go and compare and do risk analysis for different file format policies. But again, it's all in this very early native stage. But we're looking ahead and we're hoping to be interoperable with those kinds of initiatives.

Okay, so going back to the workflow process then, we'll notice that the prepareAIP process is finished, it's demoed the normalization using the rules that are defined here and it's created a BagIt – a zip file using the BagIt format – so BagIt is both a specification and a set of a number of different types of scripts.

We're using the one from Library of Congress – the JAVA script – and what it does essentially is it's just a very basic specification that just has some very basic specifications about how we create information packages. And it was originally designed for exchanging information packages between institutions, but we think it's ideal also for actually creating archival information packages and the Tipper Project was a collaboration between New York University, Florida, and I'm missing one of them, but they're testing out some of this conceptually to see if I have a BagIt file and I got it from my Fedora repository and I'm sending it to your custom-made repository, is your repository able to ingest it and receive it in?

And theoretically that is what we should be able to do, but I'm really quite, I think, confident that BagIt is the best format that we have today to start creating these information packages. There's a lot of research being done around it as well. And again the whole point of it is that it's simple and that's really something that we try to focus on constantly, like let's not over-engineer this, let's keep it as simple as possible to reduce the layers of complexity, to make it possible to – easy to get at the information at some point in the future.

So what it does, it has a few rules about simply packaging up your information and having a manifest, having the information about what version of BagIt you're using, putting checksums in it, and then having a payload directory which is called data, which in Archivematica we divide into the logs directory, where we keep all our raw logs, all the information about what's happened to the information objects through the workflow, the actual objects, and then again our own manifest, which is a METS XML file.

You'll notice that the datavibe job vacancy – the rich text format – it's been converted to ODT (Open Document Format). The Word document's converted to Open Document Format and in some cases where we don't have a good preservation file format we just don't do normalization, we preserve the original.

So this is our – for our office documents example – this is our archival information packages, this is the thing we're going to put away into storage now. It's again identified through UUID, what we've put stuff in the storage we want to use a modification of the California Digital Libraries Pairtree specification, which all it does is takes two digits of your identifier and makes that a subdirectory, so you can actually manually navigate the directory using just the ID to get at your package.

We're going to do 4 because a UUID's a little bit longer. And at this point the system's agnostic as to what storage system is connected to it. So you notice here it's a link directory N, at this point this would be a shared directory that's either connected to - for the City of Vancouver we're connecting it to a network-attached storage device. So it's just - to them it's just an NFS share that they see as one giant directory where they put their stuff and the guys in the back just keep pot swapping new boxes in as the terabytes get piled on.

We gave them a fright because we wanted a hundred terabytes right off the bat. There's a lot of Bannock stuff coming in, Olympic work, essentially community stuff coming in, as well as a giant – one of the animation companies in the City of Vancouver passed, basically donated a whole bunch of stuff, so that's terabytes of digital media. But the point is that from the archivist's point of view, they don't want to manage storage, from an Archivematica point of view we don't necessarily want to manage storage either, we just want to make sure that we're compatible with as many types of storage, archival storage options, as possible.

So for my own home digital archives I've got an Amazon S3 cloud storage account, where I can put all the stuff into Amazon buckets. And for other situations you can have it go – specifically external hard drives. We're very interested in looking at iRODS technology, having a data grid available to it and using iRODS policies to then manage the geographically – because we have an issue now with Canadian archives interested in this kind of thing but they don't want to use an Amazon S3 account because they don't want their data living in your wonderful country because of certain legislation that allows certain people to look at the data if they so desire.

And that's not just an issue with Canada. I think it's actually – I've been talking to some people – it's just a comfort zone that most countries aren't willing to cross that border, so to speak. So in a lot of cases they're looking at they like cloud storage, they like grid storage, but they want it to be national.

They don't want it to be across the national boundaries. And again that's I think setting up an iRODS network is certainly a good option to look at for Canadian deployment.

Okay, so that's the bit on the storage. So it's off in whatever storage we've connected it to. At the same time that the normalization happened, it spit out the access copies into reviewDIP.

So at this point the archivist can have another look and decide whether there's certain files that for let's say copyright or access reasons they don't actually want to put up into the access system, they can take them out at this point.

But you notice here that we've normalized pretty much – we're dealing with office documents so almost everything is normalized to PDF here, the multimedia example, most of it would have been MPEG for audio and MPEG for video. If it was the image was supposed to be JPEG at this point so whatever type of SIP we have, whatever the media type is, it gets converted to the access format. So this is now going to get uploaded into the access system.

Ken Thibodeau

Peter, does the DIP know about its AIP?

Peter Van Garderen

Yes, it does. And in a very, very rudimentary way right now. One of our main deliverables for the 08 release is syncing the DIP and AIP information. So but especially because we use the UUID, the DIP knows about where its AIP is, but it's very, very limited right now. It's very loose string.

So we want – we need entire integration between them and of course if somebody sees an object in the access system and says now I want to request the AIP we have to have that process in place to actually go get the AIP and so forth. So when we build that piece we'll tighten the integration between the two. But I do want to note that we do think – we do want to keep the descriptive information in our descriptive system and technical metadata in the AIP.

So we don't want to get in the business of taking – and we're having this debate right now – how much of the technical metadata – like you saw the FITS report, right? – how much of the technical metadata belongs in the descriptive access system in a public access system. Well somebody may want to search on resolution.

Okay, resolution I could see. Somebody may want to - or bit rate, but there's a whole list of other technical metadata that are pretty much useless that people aren't going to be searching on. And on the same taken, it makes – it's much easier to manage your descriptions in a descriptive metadata system rather than in zips, BagIt packages in archival storage, which in a lot of cases you're going to make near-line type storage as well.

So once you start talking about terabytes of storage it's a lot – you know we don't necessarily want to have to have highly available spinning disk storage, although that's nice to have, but it just becomes a cost issue. So I think there's a lot of logical reasons not to have everything synced 100%, like everything that's in the AIP is supposed to be in the DIP? I think from what we're kind of dealing with right now where we're kind of reviewing our requirements is that, what can we put in the DIP, what can we put in the AIP – and as long as the two are inextricably linked, that's acceptable to us.

Unknown Female 1

Do you have to create a DIP?

Peter Van Garderen

No, you could stop right now.

Unknown Female 1

And with just -

Peter Van Garderen

I could delete this right now and we would be done.

Unknown Female 1

And you could still search the stored?

Peter Van Garderen

Yes, and that's what we're working on the dashboard, exactly. Right now we're looking at everything through the file manager interface. At some point (well actually, we've got it already) we have the basic prototype working for it – is, this is an earlier version – but this is the web-based interface, what we call the dashboard. So the dashboard will give you the opportunity to search the archival storage, search the DIPS and so forth. A lot of, almost all the log information will be accessible here.

So that's all going to be fully indexed and searchable through the dashboard, but the dashboard is not publicly accessible – this is the thing that the archivists use. And we want to put stuff up to a web access system and the two are completely separate systems.

And right now we're integrating with ICA-AtoM but what we would like to be able to do is – you know, because we're just using HTTP Post and basically a REST API, so that you can start – we're talking to the ArCon people as well? The ArCon project? Which is now combined with the archivist's toolkit – but we'd like to be able to have the opportunity to use that as your web access interface.

And then, you know, if somebody used ContentDM or whatever else, we want to be able to have multiple accessing components. We want to be agnostic to what you want to use for your access system. Because I think that's one of the other big hurdles to get across is that people need the OAIS processing piece, but everybody already has an archival description and access component, or most people do. They have their preference, so we don't want to dictate what that's going to be.

Unknown Male 2

Are you recording relationship information or do you make preservation copies? Derivations?

Peter Van Garderen

Uh, yes, yes, yeah. Well it's not in there now but it will be in the 07, that's one of the more detailed technical things that anyone talked about. Right now the connection's purely by the identifier. So by following that collection, then you get the metadata over there which tells you all the derivations that have been made. And that is something, I think, arguably, should belong in the descriptive system as well. So you say hey, you're looking at a PDF, but by the way this came from a Word document.

So what I've done now is I've dragged it into the next Watch directory, which is uploadDIP. And I'm running a local copy of ICA-AtoM here; for the same intents and purposes, that's running somewhere completely over the web.

And in this case, the Dublin Core XML metadata that came with the SIP in the part-of element identified what *fonds* – you guys use record groups – what record group it was part of, so it knows as it's uploading.

It knows we've got one sample record group in here right now, and so we've told it that all of the SIP is essentially part of that particular record group, so it's going to find it and attach itself to it.

Otherwise you would just create a brand new top level collection essentially. The level description by default is set to the series right now. So Archivematica is uploading the various files to this *fonds* right now.

We don't see any of it because – oh there it is, I'm to log in. I see it coming in. ICA-AtoM has a publication workflow – oops – which essentially allows archivists to create descriptions while they're working on it that's not accessible to the public search and browse.

So while it's getting uploaded it is not available. It's set to draft. But this is using the title from Dublin Core XML that came in and we're using – we've got validation now as well.

This is an ISAD description, we see the top sort of telling the user what elements are still required to do proper ISAD description. If you notice here, the various files you're getting uploaded. If it had a multi-page PDF it split it all onto multi-pages for browsing.

The cover flow viewer starts to show all of the documents as they're coming in. and we took whatever metadata we got from the Dublin Core XML file, but again we like to use EAD for that to make it take in as much as rich archival description as we can if it's already been created.

And then for each individual system we'll have to do a mapping where we say this is the metadata that comes out of your particular EAD METs or whatever other sources that you have or this one legacy data migration project we're doing for your shared directories and this is the metadata we pulled off of it.

And we map that to these archival description elements for the purposes of when it gets to upload it to an archival description application.

And then once it's inside Archivematica – oh I'm sorry, inside ICA-AtoM – then we can essentially carry on with our archival description and add all of these various elements that are available to whatever standard we're using.

So we can switch templates in ICA-AtoM, go from ISAD to Dublin Core. And we've also got the rules for archival description and, by default we'd like to have DAX in there as well for the U.S. users. Very close to ISAD so that's not a big stretch. So it's gone over the output folder so all the files should be here now.

Refresh that. So what we're looking at here is the metadata that came along for the ride and just a few little sample elements. The archivist now can carry on and do full archival description.

So again, how much of - in this case we're using the ISAD templates and I can do things now like instead of series I want to make it a subseries, add any of the elements for this particular – at this level, at the SIP level or at the admin level. I'll also change the published status so as soon as I do that, the archivist has time to look at it, take stuff out, add descriptions, and then when they're happy with it they can publish it and now it becomes available on the search and browse index for the public.

Ken Thibodeau

Peter, if your SIP had metadata saved from TRIM or some other records management application, would it pull that?

Peter Van Garderen

Yeah, that's what I was talking about. We have to define profiles. We're going to have generic profiles – say our submission agreement is that you give us Dublin Core XML or you give us EAD XML and if you do, we know how to handle it. Or we say for TRIM we know TRIM spits out this kind of metadata, so we're going to take it and this is what we're going to do with it.

So we have to system by system – and ideally what would happen is at a project like City of Vancouver we create a TRIM profile – and TRIM's used a lot of different places - we had some Malaysian archivists visit, they're using it for example – so the idea is we created it once, as soon as we created it, it becomes available as a profile that ships with the Archivematica.

The next time somebody has a document in - or whatever interface - we create a profile for that and whatever project we have or a user out in the field creates that and contributes it back to the project.

So there's no way we would know without – there's ways we can make a generic profile and say we expect Dublin Core and EAD and it could do this is what we're going to with it. And otherwise we have to do an analysis for each various target system that's contributing content.

So here I just – so the archivist can do arrangement and description once it's in the system by dragging and dropping in the hierarchy tree. Where did I drop that… There it is. And then do things like export as Dublin Core or EAD after I've got a description and so forth. That's pretty much it, like this is where Archivematica 06 is at right now.

The release we finished last week - we essentially came out of a proof of concept. Until a year ago we weren't sure whether this was going to fly, we were just like – well there's nothing else going on, so in the meanwhile we've got to do something so let's see.

The idea or the concept was: can we stitch together all these tools to make something work. And the good news is that it's working, we're able to do it, we're confident now that the archives – they're as confident that you know we can be compliant with OAIS, we can create archival information packages, put them away for long term storage, using the tools we have right now.

Even this very, very early, raw prototype system. Until 05 it was essentially proof of concept, now we have this raw prototype you can actually put down in front of an archivist and they can get some work done. So there's a lot of stuff we crammed into 06 release. Asking about ingest, we want to make sure that we can properly take in BagIt SIPs as well, so again if we're getting stuff from other systems or we make ours kind of standardized on the BagIt as our SIP standard, so not just standards in the metadata. Like it'd be nice to have the qualified Dublin Core or EAD and then the way you put it all together, we'd like you put it together as a BagIt.

That would be our preferred submission information package and if not, then we will create templates to map to whatever you're sending us. We want to be able – the other big one is moving the log data into PREMIS elements and then completing that work – doing work on the first iteration of that web dashboard I showed you the screen capture for.

And then the big one for 08 we mentioned was getting the syncing between the access system and the archival storage and we're still making decisions there about what's our default metadata that we would want to share between the two.

And having a way to manage the processes. So from the dashboard, we want to be able to say – like right now we're using lock and queuing utilities, which is great. So I could throw a bunch of SIPs at the system at once and it will just kind of – it will queue them and it won't choke on them, which is great.

But now we want to be able to have it so that we have multiple nodes. Let's say I have three or four archivists working on ingest at the same time, and we want to be able – we've got so that if we're doing the video conversions – like the animation company accession I was talking about has very large video files – so we know that machine's going to be busy for the next six hours converting this giant video file for example.

So we want to be able to start – the same archivist wants to keep continuing to process the SIP, but it will be able to thread processes on various machines. That's something – you know that's pretty advanced step we can do to essentially make it a production ready system that can handle high volumes of ingest. And that's the other big thing we want to work on.

And of course the preservation monitoring piece. We've already got a bit of a prototype out for the format policy registry I was talking about, but we want to actually incorporate that into the working system.

So since we started this project now there is a couple of vendors that are offering similar type solutions. So essentially, you know - one vendor specifically - talked the OAIS language and are meant to do the same thing – taking information packages from ingest to access. But the one difference between Archivematica and them is that we're a fully free and open source project.

And what that means is the product is free as in free beer, so if I buy you a beer – it's an analogy, so I can buy you a beer, so it's free to you – it didn't cost you anything, there's no monetary charge for you.

Most importantly though, it's free as in free as a dove, so free software – there's four fundamental freedoms that are at the core of what free and open software is and these are defined by Richard Stallman who wrote the GPL license.

There's variations of this and there's various open source definitions, but the four core freedoms is that you can – You have the freedom to download this software and run it for any purpose.

You have the freedom to study it and essentially figure out how to adapt it to your own needs, so here's Archivematica, here's a default, here's what we did with it. See what you can do with it. If you want to make customizations to it, you want to do something different with it, you're more than welcome to do it.

And a critical piece of that is to give access to the source code, so there's a lot of projects around that say they're free and open source, but you can only download a tarball.

Or you can download an executable file that they've built with their own system so you can't get at the source code. So I think that in order for the free software to be free you have to have easy access to the source code.

You're allowed to redistribute it to anybody, or society in general, for the society's benefit. And of course you can improve it and contribute it back to the community as a whole.

There's no such thing as a free lunch, it always costs money to run technology, including free and opens source software, including Archivematica.

The big thing is though we can improve quite a bit on the total cost of ownership, especially for the archival community where there's limited budgets. So I think the best analogy is kind of like a free kitten – so it's free to you, but you got to pay to feed it, you know?

But it will grow up and it'll grow to love you and can sit in your lap and purr and you'll have a good relationship with it. So the point I want to end on is that I think you know Archivematica is in this very early stage but I think that we've already proven that we're onto something and I think that in a very short period of time, with very, very limited resources, we've been able to put together a good raw working prototype and within the year's time we expect to be at a point where we have production ready – or we expect systems that are able to go into production-ready digital preservation processing.

The big key difference again is that I think we're able to maximize limited amount of funding and resources that are available in the archival community. Where this – a lot early development done by lead institutions, essentially to have all the knowledge about what archival preservation is, hire a contractor, they pass that knowledge onto the contractor, they pain-stakingly co-develop digital preservation solutions over a period of years and then can't share that technology or that technical knowledge with the community at large because the contractor now has a license on it and they'll resell it back to the community.

So that knowledge that – it's essentially public money being spent to create technology to preserve public records for the public trust but it's not in the public domain. And personally I have a problem with that, as a tax payer I have a problem with that. Of course as an open source software developer it's also my business model is to offer an alternative to that. And so you know one of the big things we want to do is encourage as much participation in the project as possible.

And there's multiple way for us to find partners and collaborators and the open source business model is one that's very legitimate these days – there's lots of success stories and lots of places where it works.

But I think in particular for the archival community, where we're dealing with – essentially one of our major problems is obsolete software, incompatible software, and proprietary software, so it's a little ironic that we would be funding solutions to create proprietary solutions for it.

That's my open source spiel so I hope everybody – that's my little bit about open source software and this is why I think open source software is – at the very least I want there to be a legitimate open source alternative for archival institutions and that's part of the rationale, one of the driving points, behind the Archivematica project is to build a solid core of software code that works well, it competes just as well with proprietary products but also it becomes a base for sharing knowledge within the community and building that community.

And this again, there's lots of pieces to the puzzle, a project like this started in our case started with us a service provider. In some cases it starts with lead institutions. And as the projects mature, typically they'll establish some kind of foundation or steering committee and this is exactly the kind of ecosystem that we're looking to build around the Archivematica project.

So if you want to get involved, learn more, that's where to find it – the website where to find more. Thanks for your time, I'll take more questions if we have time.

[CLAPPING]

Ken Thibodeau

I have another question. Go back to the beginning, you said the archival object is a file.

Peter Van Garderen

It is a SIP.

Ken Thibodeau

But you're also equipped to deal with share drives, so if the content of the share drive is an end user's file or files, as opposed to something, I don't know, coming out of TRIM or some other RMA, the default arrangement of those files is the file – the paths that the user established. Does Archivematica preserve information about that structure?

Peter Van Garderen

Yeah, it keeps the same structure. So, one of the initial debates we had was whether every file is a SIP or every file is an AIP and it just wasn't practical. And the OAIS specification is flexible on that point and we tried getting some debate on EXL about it and some of that stuff but none of the people were talking about it.

In any point, the decision – the design decision that we made is that whatever the user drops down as a SIP, and that could be multiple files with nested directories, that's what we maintain as the AIP. And this is part of the design decision up front when we do integration with the existing systems, is what's a reasonable size for how many files we want to SIP or what's the rule for creating a SIP? The logical arrangement?

So for the electronic records document management system it's all the files put in the same classification code that have the same retention rule, that's the stuff comes in as individual SIPs. In your example,y ou say okay, I'm going to drag and drop all of my documents directory with all of its various nested directories – that's what you drag and drop into receiveSIP.

Archivematica will start processing that as a SIP, as one giant information package with multiple information objects, sorry multiple files within it. Or you can parse it out and say okay, I want to take these objects and so forth. But it will respect whatever nesting you've assigned to it right from the beginning. But again from a logical point of view you probably don't want to have a hundred directories with several thousand files as one giant information package.

It just makes it more difficult over time to manage as archival information packages and so forth. But at this point it's totally agnostic in what you decide to feed it.

Mark Conrad

Peter, what would you do with a web crawler? I saw you had a HTTrack on one your tools, and you are going to have multiple levels of hierarchies and file formats all over the place.

Peter Van Garderen

And that's what we decided to do with the web crawl. I don't know of any web archiving project where they actually have somebody go through and rearrange it after the crawl. It just is too time intensive.

So yes, that's a really good example, similar to what Ken was saying, we will take the entire web crawl, starting at the roots at a URL, which is you know, it's like HTTrack, it relates it all to nested relative directories. And we will take that in as the SIP. Does that answer your question? It's the silence. To be honest, we haven't done much testing with it yet, so we don't know at what point and/or if it will choke, whether it's the – at this point in time that's our design decision.

And if you've got the minutes – is that Richard or is that Mark?

Mark Conrad

It's Mark. We've got quite a few web crawls sitting on our test bed. One of them is, for example, the KODIAK system investigation board website, which has somewhere in the neighborhood of 4000 files and a dozen formats, and how many levels of hierarchy.

Peter Van Garderen

Yeah, 4000 files seems okay to me, like I think we'd be okay with that. It's just more processor cycle time. It just might take a long time to crunch through it all. I don't think that will be a problem, I mean that's a small website. When you think like 4000 files.

Mark Conrad

Yeah, that's relatively small, but what do you do with the normalization.

Peter Van Garderen

Normalization is just going to go side by side with the original format and it's going to get mapped in the METS document and then we can apply rules and logic to the METS document later on. So like, you know, it will create the file groups, you know the METS in the file section. And that we can use that to map, use its use attributes to figure out which was the original and which was the normalized copy.

So it's – we are applying preservation formats where appropriate, so that's one step better where you just doing a crawl and you put it away into storage. Or you're crawling and I think – correct me if I'm wrong – the Heritrix approach is you do the crawl and essentially you stuff it into ARC files as a compression package but you're not really doing any preservation of the actual files that are coming back from the crawl. Is that right in your experience? Do you have much experience with the Heritrix approach?

Mark Conrad

I haven't played with Heritrix, I have played with HTTrack and just the way you maintain the links, you know, depending on the settings that you use on HTTrack you'll get some very different results. And you will also get very different results in terms of what you can disseminate from that.

Peter Van Garderen

Oh yeah, absolutely. So there's like a black magic to getting HTTrack to work the way you want and a lot of days invested in it usually. But we still like it as the only option. For example the reason we integrated it is because we wanted to crawl the Vancouver Olympic Committee websites because they were going to go down at some point. We've got to do something, what can we do today? Well we can run HTTrack.

So a couple of the archivists at the City archives played with the settings for quite some time to get it to the point where we were happy with the scope – what we were getting back – and now we know we've got something. We've always got that original crawl. We're going to run it through Archivematica so we've got some normalized access copies.

The settings you can use, you can use them to create a completely relative website so when you start with the source page, you can – as long as there's hyperlinks in the nested pages – you can follow the hyperlinks around. Of course JAVA script and search indexes won't work. But the site itself is preserved, assuming that you've got file readers for all the various file content that was on there.

What we would be able to do using the METS document we'll be able to say, if you hit one of these – you know, create an alternate mapping so if you're going to hit files – let's say it's ten years down the road and you're going to hit certain files in there that we know we no longer have viewers for anymore or file versions, we could use the mapping and the METS file to swap those out with the ones sitting side by side.

So it leaves us more options, I'm not saying it's a complete solution but it's as good as anything else out there as far as I'm concerned.

Mark Conrad

Okay, thanks.

Unknown Male 3

I don't want to get too far into the weeds which is usually a warning that I'm about to…

Peter Van Garderen

I've got lots of time, I don't know about your colleagues.

Unknown Male 3

I was wondering if you had any thoughts on the tools, approaches that you might use toward the problem of withdrawing material and redacting material that you publish.

Peter Van Garderen

With drawing material? So you mean like scalable vector graphics?

Unknown Male 3

Well, no, I mean withdrawing material…

Peter Van Garderen

Oh withdrawing, okay, pardon me, okay yeah. Well I think we would leave that open to the access system. And like for example with the ICA-AtoM, we're able to leave stuff as un-public.

First of all you get to decide before you upload whether you even want to upload stuff altogether. If you have problems with – there's access issues, you wouldn't upload that to your access system anyway. And in that case you would probably integrate a redaction tool at that point and redact it before it gets uploaded to the access system. It’s probably the way we’d want to do it.

For the time being, we typically have access policy set at the series level or at the higher level, not at the item by item level, but I know that’s not always the case. So we would withhold publishing, like we might have the descriptive metadata but we might not upload the digital objects if that’s an issue.

But it really is something I don’t necessarily see – other than integrating - again maybe adding a micro-service for – we probably wouldn’t even make it an Archivematica micro-service, just make it a stop where you would then be able to apply a redaction tool and do what you needed to do before you upload it. That’s probably how we would handle it but it hasn’t been an issue yet. We haven’t spent much time thinking about it.

Unknown Male 3

Okay, thank you.

Peter Van Garderen

We’re good? Thanks everybody for your time.

CLAPPING