This session isn’t about VRA Core worship.
It’s about taking the Core...
...and adapting it...
...interpreting it...
...to work within limitations...
...,or extending it, to meet local needs...
...without leaving it unrecognizable as VRA Core.
This is the current version of VRA Core - version 4.0, first released in 2007.
Version 1.0 was launched in 1996 and ever since it has adapted in response to developments in technology and community input. The intended community has been cultural heritage visual resource professionals, but that is changing. Notice the URL there, VRA Core is now hosted at the Library of Congress and is more visible to a wider range of users seeking a standard to catalog digital media from many disciplines and within many different systems. The original authors of the Core couldn’t have imagined the all the ways people want to use it today. The presentations in this session demonstrate the process of resolving problems and conflicts in local implementations.
Many of you are already familiar with the work of the VRA Embedded Metadata group has done to incorporate Core 4 into digital image embedded metadata, but how many of you know the backstory – the dirty, behind the scenes work that went into making it all work? I’m still working on my memoires, so I think not many of you. What follows is a little taste of the story.
I checked the fine print in the VRA Bylaws – it’s not required to use Core 4 or to support it. I think it is still illegal to disparage it or use the Core logo in an unflattering manner, so I’m probably in some sort of trouble here. The EMwg chose to use it in embedded metadata because it has a lot of useful features.
By the way, what users were we trying to help?

- Limited access to central database
- No IT support
- Shared image project
Plenty of tools exist for entering basic photo metadata: title, caption, keywords. We wanted VR quality metadata, so that meant creating custom input tools. Most of our users are not programmers, so the tools had to be familiar and as easy to use as possible. Pretty much all our users have Adobe Photoshop, Bridge, and Excel. They do most of their data manipulation in Excel – the lingua franca of data. Our task was to figure out what these tools were capable of in terms of Core 4 metadata. What could they create? What could they import and export? We also wanted to produce data that was transferrable to other programs.
Currently, the best way to embed metadata in images is Adobe’s XMP standard. It is open source, widely supported and extensible. XMP is not perfect though – it has limitations. XMP is serialized in RDF/XML, but it is a subset of RDF, not all RDF properties and attributes are available.
How do we jam Core into it? What will we loose? What matters?
The first step was to figure out how XMP data works – format, serialization, limitations. Then we had to figure out how to write plugins and info panels for Photoshop and Bridge. What is possible in the UI? What functions can be built in?
When choosing fields, like Work Title or Image Copyright, the idea is to start with schemas that are most widely used by the majority of photo applications and web services and then move down the list, using specialized schemas last. This places as much of the metadata as possible in properties that will be read by common tools.
There’s VRA all the way at the bottom looking like the least favored of the children. It turns out however, that it has an important role to fill, stepping when the other schemas can’t fulfill our needs.
Most of all is the distinction between the original work and the image of the work. Standards have been in place for digital image metadata for a long time. The EXIF and IPTC standards cover the who what and where of the photograph, but they are a bit muddy on the details of creative works shown in the photo.
IPTC Extension does have fields specifically for artworks and, while a good start, they don’t meet the needs of the users EMwg was trying to help.
For Instance, “Date Created” is a single calendar date only and doesn’t allow for a range of dates or a complex free-text date such as “built 1298 – 1310, destroyed 1673”
Another thing we tried, and the method that would be the most reliable and computer friendly, would be to nest VRA within IPTC. This keeps all the artwork data together in one structure and makes it possible to describe multiple artworks using multiple arrays, each one being a completely discrete packet. This method is supported by XMP.
Unfortunately, most applications don’t recognize the nested VRA data and delete it. So, I can make a custom info panel that qualifies IPTC with VRA, but if you open that metadata with another tool, the VRA data will be stripped out. This is far too dangerous – we can’t risk having VRA metadata erased.
One solution would be to tell people that they can only edit embedded metadata in the VRA tool. This would never work. We can’t tell people to only open their file in one particular application. The metadata we create has to be safe to open in the tools people are likely to use.
Just to reiterate: this is a problem with the way software developers have implemented IPTC Extension, not in XMP. Most developers don’t think about people customizing IPTC metadata so they don’t build their software to handle it. This is a reality that we have to live with which means extending IPTC Extension artwork with VRA work is not practical – VRA data has to be separate.
Our practical solution was to use IPTC Core for the Image and simple Core 4 display values for all the Work.
Core 4 does have a lot of fields. Do we need an image file to carry them all?
Here is a snippet of Core 4 XML. Could we replicate this structure without losing meaning in XMP? Yes, we can and we did.
Here is a snippet of Core 4 XMP RDF/XML. Sure, it looks great. It’s got lots of arrows, colons, and slashes and everything is nicely indented, so it should work well. The question is, do we really need this complexity?
The answer was, no. This is the use case we targeted. This is how many database exports and imports work in Excel. You can parse the data if you want using semicolons as delimiters. Of course there are limitations with Excel – you can’t easily include subfields or types, but many people seem to be OK with that because they only display simple lists of values in their user interface. Most importantly, users are comfortable with Excel, they aren’t programmers and aren’t going to build an XSLT to transform complex data.
We decided to keep it simple and use a flattened version of Core 4 display fields. This means that we will eliminate the nested arrays of parsed terms and use single free text display values instead. These are very easy to export to Excel. A spreadsheet is the easiest and most common way people will use it. This does mean that curators will have to do some work before they can ingest the data into their database. For instance, if multiple artworks are present in an image, the display values must reflect this through the use of identifying labels and delimiters.
Locations were tough. We saw a need for more granularity there because works can have several Locations worthy of mention.
Choosing a namespace. We went in assuming others would use our version of VRA in XMP. If we were successful, it wouldn't be just us using it, so we wanted to be careful about what it conveyed. We didn't want it to conflict or suggest that it was a complete representation for VRA Core.

We came up with the name vrae – VRA Essentials
If we wanted to park big ‘olCore 4 in XMP...
...we would have to adapt it while leaving it drivable. A lowrider big rig – I guess that is kinda cool. Sure people pointed and laughed, but we didn't mind, were able to go where we wanted. And then, Sheryl Frisch asked me if we could add some complexity to the VRA Panel to accommodate the California State University’s Dspace shared image project.
They used qualified Dublin Core to cover the fields they needed, then they designed a matching Excel template for contributors to use. Our first thought was to just make a completely new XMP panel using their qualified Dublin Core field, but then we thought we could use this as an opportunity to expand the VRA XMP panel and build something everyone could use. Again we were faced with obstacles from XMP and the tools we were using.
The VRA panel was kept simple by design - a single display text field for every Core 4 element. Obviously for the CalState Creator section we had to add a lot of fields...
... 10 to be exact. We also added an auto-complete feature for the Creator Label so a uniformly formatted ULAN-type Label would be produced.
We also allowed for multiple Creators. After we finished this and it was working I thought back on the many times people told me, “No one will ever want to enter granular data in Bridge.”
CSU wanted a GETTY ULAN-like name display and we thought it would be easier for the user if we provided individual boxes to enter the data and then let the panel build a properly formatted label. Core 4 doesn't have all the individual fields to do this – for instance, it doesn’t have first and last name or date display ...
...so we used FOAF and CDWA lite properties.
What ended up with - a very faithful expression of VRA Core 4.0 in XMP. Yeah, it’s kind of complex.
An example of how we retained Core structures in XMP. We determined which Core elements were repeatable, then used an RDF structure for these. The fields here are not repeatable, so they are a simple structure: the are just a simple list of non-repeatable properties within the Agent Details. For instance, Name, earliest, latest dates are not repeatable.
Agent Role and Culture need to be repeatable, so we placed them inside containers – in this case RDF arrays. This allows us to have as many of them as desired.
For example, a single role would be written like this.
But additional roles can be added to the array.
And if desired, VRA attributes, like vocab and refid, can be added.
OK, I’m going to lay some technical XMP information on you now. I’m doing it to illustrate how you can replicate meaning from a schema to a new data format. We wanted to keep our XMP Core 4 data as lean as possible so we looked at using inherent meaning in XMP RDF to express Core 4 concepts such as preferred Title. Core 4 uses the XML attribute pref=true/false
XMP uses RDF arrays (lists) to hold repeating data values. These lists must be given a type to indicate if the order of the items is meaningful. Why? So the UI knows if the list should be displayed in order. A list that is meant to always be displayed in a certain order is called Sequential.
For example, if you enter “Mona Lisa” in the top position of the Title, it always stays there and will be displayed on top every time. We used this inherent meaning in XMP RDF to express the Core 4 attribute pref=“true”. Sequential array item 1 = XML pref=“true”. This simplifies the data because we didn’t have to add a new Preference property.
The vrae structure accurately reflected Core 4 and the custom file info panel was beautiful, and the export-import tool worked. There was just one problem...
...the users. Yes, we want to make the users happy and what made them happy was ease of use and speed.
They found the metadata palette in Bridge faster and easier to use than the file info panel window. We set out adapting the CSU panel and vrae to the Bridge palette and quickly discovered that it was not possible. It turned out that custom palettes are very limited, most crucially for us, they cannot create structured data...
...like vrae. Those nested structures that match VRA Core? Forget them, they can’t be created in the Bridge palette.
A somewhat faithful expression of VRA Core 4.0 in XMP. It is very easy to program and to import and export to Excel. But, it has some strict limitations.
To handle repeating fields, like multiple Agents, we numbered them, 1, 2, and three in this case, then added extensions for the subproperties and types we needed.
But can the flat version work for other people? We predefined three Agents, but what is someone needs a fourth?
This flat expression of Core 4 works very well for CSU’s project because there is a known template. The flat fields drop right into their Excel template. Import and export work very well and the data entry tool was easy to code.
Considerations

theory vs. reality
ideal vs. practical
So we have a stable, simple, Core 4 display values info panel which has gained a following.
We have sorted out how to express complex Core 4 in XMP and have a working info panel for it.
We have adapted that to a flat format that is easier to work with, but has some strict limitations. So what to we do with these, choose one and make it public? We could, but now there is another complication (a good one): widespread interest in setting an XMP standard for cultural heritage images. The VRA EMwg doesn’t want to put out an XMP standard and tools that might become obsolete or might compete with an international effort – we want to be part of that effort.
First, IPTC have acknowledged the need for more heritage fields and have approved a group to put forward a list of candidate fields. We are in the process of doing this right now. The museums, libraries and visual resource collections we surveyed expressed an interest in a more granular XMP standard, one that picks up where IPTC leaves off, so we are also pursuing that through the SCREM project.

What is SCREM?
It stands for Schema for Rich Embedded Metadata for Media Files and is the working name for a wider schema to enable interchange of heritage metadata between and within heritage organisations, to be supported by a custom XMP interface.
Obviously we have solved a lot of the problems with vrae and it could serve a starting point for SCREM. However, we want buy-in form as many people as possible, so we will consult with a wide range of experts on what schema will serve as the foundation and how it will be structured. We will also get a broad consensus on what fields to include.
Matthias Arnold, Heidelberg Research Architecture – Visual Resources
Marta Bustillo, National College of Art and Design (Ireland)
Heidi Eyestone, Carleton College
Sheryl Frisch, Cal Poly, San Luis Obispo
Stephen Jennings, Fine Arts Library, Harvard
Heather Lowe, California State University San Bernardino
Heidi Raatz, Minneapolis Institute of Arts
Amanda Rybin, The University of Chicago Department of Art History
Greg Reser, UCSD
Steve Tatum, Virginia Tech