LOUD: Linked Open Usable Data and linked.art

David Newbury

September 15, 2018

Background

Cultural heritage has been trying to solve the problem of the machine-readable semantic expression of cultural information for decades. Earlier work such as LIDO[1] or CDWA[2] worked to do this in the context of cataloging or resource discovery–descriptions of objects, formalized in such a way that it would be possible to harvest and display metadata about those objects.

The description of objects is, if not solved, at least reasonably understood. However, object-centered description comes from a culture of museum practice where the primacy of the object becomes the most essential characteristic. Because of this, the data structures are designed to reproduce the cataloging practices of the museum professionals.

While not an incorrect way to deal with objects, when we want instead begin to understand the context of the object we need a more holistic strategy for describing objects as entities that exist within social contexts. These social contexts are made up of interactions between people, mediated via objects, taking place at specific times and locations. Treating these interactions (or events) as the locus of documentary effort rather than the object allows us to begin to express more a more nuanced and interesting version of cultural heritage.

The CIDOC-CRM provides a semantic framework that allows us to formally express this event-based model[3]. While man-made objects remain a core entity, people, places, events, and concepts also emerge, and the CIDOC-CRM, by focusing on events and activities as a connective reification, begins to allow us not only to describe objects but to contextualize them.

However, one of the main critiques of the CIDOC-CRM is that its expansive scope and logical formalism means that the practical application of it within computer systems is a complex and unfamiliar task to most software developers. While the editors have worked to ensure that the logical formalism remain

[1] http://network.icom.museum/cidoc/working-groups/lido/
[2] http://www.getty.edu/research/publications/electronic_publications/cdwa/
[3] http://www.cidoc-crm.org/

compatible with the W3C Resource Description Format (RDF), even that is an unfamiliar technology for the vast majority of software developers. To broaden the adoption of the standard, we feel a more developer-friendly expression of the CIDOC-CRM is needed, and have thus created linked.art[4].

Precursors to linked.art

Linked.art is not the only attempt within cultural heritage to make use of RDF and the CIDOC-CRM for the description of cultural heritage information. The Yale Center for British Art, the Smithsonian American Art Museum, and the British Museum in particular were significant early adopters of the standard, and provided an excellent foundation and practical experience in what is needed to publish a RDF serialization of museum information. However, while these institutions provided access to their data in this form, none of them consumed the data as part of their core digital practice. Rather, it was seen as an publication format for consumption by others[5].

In 2014, a major advance for the field was the formal publication of the JSON-LD serialization for RDF[6]. JSON-LD provided a way to express the constructs of RDF in a way that could be interpreted by software developers as standard JSON, an extremely common syntax used on the web, without needing to understand the formalisms of RDF used "behind the scenes". Additionally, the International Image Interoperability Framework (IIIF)[7], while not CRM-based, demonstrated that a significant JSON-LD standard could achieve acceptance in the cultural heritage space. Arguably, much of the success of IIIF came from the way that it used JSON-LD to hide the use of RDF from implementers, while preserving the formalism and underlying semantic structure.

Between 2014 and 2017, three US-based projects began to look at JSON-LD and the CIDOC-CRM as an underlying foundation for software applications that would not just transform and republish existing data, but would generate RDF-native content and, significantly, consume and display it in custom web applications. Art Tracks, a project of the Carnegie Museum of Art active between 2014-2017, used the CIDOC-CRM to describe the provenance of art objects[8]. The Getty Provenance Index Remodel is re-expressing the 1.7 million records within the Provenance Index as CIDOC-CRM [9]. And the American Art Collaborative (AAC), between in 2015-2018, worked to reconcile and combine the collections of 14 museums and archives across the US using the CIDOC-CRM as their base model[10]. These three projects had significant overlap, both in scope

[4] http://linked.art

[5] ResearchSpace, once complete, will be a significant application built to consume this data, but to date has not seen wide public release.

[6] https://www.w3.org/TR/json-ld/

[7] https://iiif.io/

[8] http://www.museumprovenance.org/

[9] http://www.getty.edu/research/tools/provenance/provenance_remodel/index.html

[10] http://americanartcollaborative.org/

and in personnel.

In particular, the American Art Collaborative revealed two significant requirements for the success of this type of work. While modeling the collections of the 14 institutions, it became obvious that without documented, consistent patterns for using the CIDOC-CRM, different interpretations would produce different results for the same sorts of data. While semantically valid, differences in the level of completeness in the data, differing interpretations of the CRM scope notes, and different ways of mapping the reference model into RDF produced structurally-incompatible expressions of similar underlying information. For example, it might be arguably correct to model the gender of a person as an `E55 Type`, it would also be arguably correct to model it as a membership within an `E74 Group`.

Second, through the implementation of a consuming application in parallel with the data modeling work, it became clear that semantically correct modeling of the data was often insufficiently precise to allow for the "roundtripping" of information implicit within the structure of the original data. This information would often not be expressed explicitly in the RDF, making it impossible to programmatically distinguish between two semantically different, but structurally identical constructs. As an example, an early AAC model treated both material statements and dimension statements as instances of `E33 Linguistic Object`. This is semantically accurate, but the consuming application was unable to distinguish between the two. Without explicitly classifying them as "material statement" and "dimensions statement", there was no way for a software application to display them independently.

Neither of these problems were barriers to publication, but they were both barriers to consumption and to interoperability. In order to resolve these sorts of issues the AAC Target Model and Review Application[11] were created. David Newbury (working for Design for Context) and Rob Sanderson (at Stanford University and then the J. Paul Getty Trust) collaboratively developed a set of patterns for expressing core CDWA concepts in the CIDOC-CRM in a way that allowed them to be published and consumed as JSON-LD. These patterns, informed by their work on Art Tracks, IIIF, and the Getty Provenance Index, became the foundation for the American Art Collaborative model and the basis for linked.art.

What is linked.art?

Linked.art is a RDF profile of the CIDOC-CRM that uses JSON-LD and the Getty Vocabularies to describe object-based cultural heritage in an event-based framework for consumption by software applications. It uses a subset of classes from the CIDOC-CRM ontology along with other commonly-used RDF ontologies to provide interoperable patterns and models that can be interpreted

[11] http://review.americanartcollaborative.org/

either as JSON or as RDF. It focuses on usability and consistency, rather than completeness: as a design principle, it tries to cover "90% of the use cases of 90% of the organizations, with only 10% of the complexity of the full CRM ontology with all of its approved extensions."[12]

Linked.art is designed with the skills and knowledge of web developers in mind. The biggest implication of this is that it is designed to not require a full RDF platform to use–experience has shown that requiring understanding of the unusual tools required to make use of the semantic web significantly reduces the number of developers who are willing to engage with the data. By expressing the RDF as JSON-LD, a developer can interpret complex graphs as a collection of JSON documents describing hierarchal data structures, a far-more familiar pattern within the software development community.

This decision is not without downsides: one trade-off is that linked.art loses some of the capabilities that RDF brings–most significantly, inferencing and implicit inverses. We cannot assume that developers will make use of the information contained within the ontology description to interpret the data–each document must be self-contained. It also means that any specific API will be optimized for specific use-cases: the choice of what data is included inline and what data is only referenced has a significant impact on the performance of consuming software applications.

It is not without compromises for developers, either. In general, the desire for the information presented to be semantic, without implicit information expressed within the structure of the document, means that the JSON documents tend to be more verbose than a non-semantic API would otherwise require. That complexity also means that consuming applications often require more network requests than would be needed in a fully-customized API. But these compromises are, in our opinion, worthwhile trade-offs to allow for the far richer interpretation of knowledge enabled by the underlying semantic framework.

How is linked.art implemented?

By reviewing the American Art Collaboration data, along with Art Tracks, the Getty Provenance Index, the Pharos Consortium[13], and other data sets provided by the community, linked.art has identified a subset of the CIDOC-CRM classes that covers the vast majority of real world use cases for object-based art collections[14].

Linked.art uses the JSON-LD concept of a "context"[15] to map those selected classes and their respective relationships from the published CIDOC-CRM RDF ontology into JSON property names using a set of established

[12] https://linked.art/model/profile/
[13] http://pharosartresearch.org/
[14] https://linked.art/model/profile/class_analysis.html
[15] https://w3c.github.io/json-ld-syntax/#the-context

rules[16]. This process provides developer-friendly names for RDF URIs: `http://www.cidoc-crm.org/cidoc-crm/E22_Man-Made_Object` becomes `ManMadeObject`, and `http://www.cidoc-crm.org/cidoc-crm/P2_has_type` becomes `classified_as`. By removing the property numbers, we avoid developers needing to memorize or look up CRM numbers. These simplifications also allow the properties to be accessed using dot notation with JavaScript, something that would not be possible when directly using the predicates. The context is published at https://linked.art/ns/v1/linked-art.json, along with a second context providing the full set of CIDOC-CRM predicates.

In addition, linked.art imports a small number of other classes and predicates from well-known RDF vocabularies such as RDF[17], RDFS[18], SKOS[19], Dublin Core[20], and ORE[21] to accommodate concepts that are either as-of-yet inexpressible in CRM, or are, as in the case of `rdfs:label`, in such common use as to be a de-facto standard. It also defined several new classes to handle concepts that are common in practice, but are not yet describable in the CRM: given the profile's history with art provenance, many of these concepts have to do with the complex legal practices surrounding property interest.

These technical decisions allow linked.art to express complex information in a form that can be expressed in a form easily interpretable using standard software libraries. While necessary infrastructure, they do not, on their own, accomplish what is needed to enable Linked Open *Usable* Data. Along with this, and building on the experiences of the IIIF consortium, design principles are also needed that allow both publishers and developers to understand how to create and interpret these documents consistently.

Much of the work of the linked.art project has been around identifying common patterns in real-world data and formalizing these patterns into design principles that can be reused. These reusable patterns benefit publishers, in that they help ensure consistency between various implementations of linked.art, and they benefit consumers, as software libraries can be written that can interpret these patterns and consistently extract information from them. What follows is a discussion of several of these patterns.

Pattern #1: Class Specificity and Classifications

One of the most consistent problems when describing the real world has much to do with the granularity possible when classifying human knowledge. Humanity excels at creating systems for ordering the world, but managing the semantics of

[16] https://linked.art/model/jsonld/#context
[17] https://www.w3.org/RDF/
[18] https://www.w3.org/TR/2014/REC-rdf-schema-20140225/
[19] https://www.w3.org/TR/skos-primer/
[20] http://dublincore.org
[21] http://www.openarchives.org/ore

that complexity in a way that maintains the ability of machines to interpret it remains a hard problem.

In our specific use case, the problem manifests itself through a mismatch between knowledge representation and knowledge retrieval. When describing an entity, you would like to be as precise as possible. It is more useful to tell someone that an object is a "Ford F150 Pickup Truck", than just a "Truck", to say nothing of the CRM class "Man-Made Object". However, when searching for an object, the opposite is true: Often we would like to retrieve trucks without exhaustively listing every possible type of truck, particularly when it is difficult or impossible to know what options are available beforehand.

RDF typically uses RDFS and OWL as a mechanism for managing that complexity via inferencing and ontology. By describing the world using object-oriented inheritance patterns, you are able to build complex models of the universe, and, through inferencing, retrieve simpler models. However, there are difficulties with this model: one is that linked.art assumes that most developers will not inspect the ontology, and from the document itself it is difficult or impossible to determine what the super-classes of any given object might be. In addition, these classification schemes are rarely strict hierarchies: our truck has a make, a model, and a type, and the complexity of understanding the class relationships between all of those is difficult to predict without a deep understanding of the specific ontological decisions that went into the modeling.

A second solution, one used by the British Museum, was to avoid precisely defining each entity, and instead rely on precisely describing the relationships between the entities. While this minimizes the need for inferencing, it still requires enumerating many, many properties, and enabling interoperability means that the community needs to choose specific extension properties.

The CIDOC-CRM, however, provides a useful mechanism for working with this through Types, using the `crm:P2_has_type` predicate. We found it useful to think of these as 'tags' or 'classifications', hence the linked.art use of `classified_as` as our property name.[22] This allows us to use very broad classifications such as "Man-Made Object" for things, but still allow for highly precise description through the use of many types. Our truck would then instead be a Man-Made Object classified as a "Ford", a "F150", and a "Pickup Truck".

Given a suitable vocabulary, we could then discover that a Pickup Truck is a type of Truck, which is a type of Vehicle. This moves the responsibility of maintaining class structure away from the ontology and into the vocabulary. In the abstract, this seems only to move complexity from one area to the other, but hierarchal vocabularies are a far more common concept for developers. In addition, within our specific domain we have the benefit of the Getty Vocabularies: in particular the Art and Architecture Thesaurus, or AAT[23]. AAT began compiling a thesaurus

[22] The property `type` was already being used for `rdf:type`, so needed to find another property for P2, as our rules for constructing properties remove the "has" from "has_type".

[23] http://www.getty.edu/research/tools/vocabularies/aat/

of domain-specific terms in the late 1970s and has been in continuous development since. In 2014 they were released as RDF and are an extraordinarily useful mechanism to classify entities within the cultural heritage domain.

This pattern of filtering arrays of objects based on a property of that object is extremely common in software development, so while slightly more complex than a discrete property it is not difficult to implement. The other benefit of this pattern is that it makes extensibility trivial–a consumer does not need to understand every included type to make use of the data and a publisher can add additional types at any time without invalidating any existing implementations. This ability to incrementally improve data without breaking working implementations is essential for interoperability.

Pattern #2: Strings, Things, and Linguistic Objects

Another reoccurring pattern that came out of our investigations is that legacy data is often not semantically precise enough to be modeled using the CIDOC-CRM. Much of the early modeling advice given during the AAC work was semantically correct, however it did not take into account the level of human effort it would take to transform existing data to match the patterns described. Material statements are an obvious example: while "Oil on Canvas" is not particularly difficult to represent semantically, "Mahogany desk, inlaid with ebony and rosewood, with brass and ivory fittings" would be extraordinarily complicated to express using the CRM, and often the nuance present in the text cannot be expressed in CRM at all.

Instead, what we found was that there was a reoccurring need to maintain both the human-readable string and the machine-readable structured data, and that these two structures could co-exist side by side. By treating the descriptive strings as E33 Linguistic Objects we preserve the ability to present the information to humans. We can than, over time, parse that data into more complex structures for machine use without having to perfectly replicate all the knowledge contained within the string. This pattern of increasing levels of complexity reoccurs regularly within linked.art.

In implementing this, we found that it was surprisingly difficult to include these texts within the RDF. Natively, there is no well-defined mechanism for including strings within an RDF serialization of the CIDOC-CRM. The semantics of suggested solution, "The text of an instance of E33 Linguistic Object can be documented in a note by P3 has note: E62 String"[24], are not explicitly correct–it is not about the Linguistic Object, rather it *is* the Linguistic Object. Instead, for Linguistic Objects, we use `rdf:value` to indicate the explicit, machine-readable content of an entity.[25]

[24] http://www.cidoc-crm.org/sites/default/files/cidoc_crm_version_5.0.4.pdf

[25] This is currently under discussion on the CRM mailing list, and it is likely that a formal solution to this will be discussed at the next SIG meeting.

Finally, to distinguish between various classes of texts, this pattern makes heavy use of the `classified_as` pattern. By typing Linguistic Objects against the AAT we can appropriately display known types of text in appropriate context while still maintaining the ability to display unknown text to the user if desired.

Pattern #3: Partitioning and Membership

The third reoccurring pattern used across linked.art is the partitioning of entities. Often we would like to group specific statements about an entity–to be able to say that Currier engraved the print, while Ives printed it; or that the painting consists of oil paint, while the support is canvas on panel. While the CIDOC-CRM provides several patterns for expressing these sorts of statements, linked.art has chosen entity partitioning as its preferred pattern for any instance where we would like to group specific properties of an entity together.

To aid in this, the linked.art context uses scoped contexts[26] to reuse the JSON property `part` to represent each of the partitioning relationships such as "P106 is composed of" or "P89i contains" within the CIDOC-CRM. This allows the semantic richness and interoperability of the CRM to remain, while making it significantly easier for publishers and consumers to make use of the resulting data.

This pattern is used for Things, Places, and Periods, as well as their children. Group membership has an alternate pattern, to maintain the consistency between membership of groups (which may be partitioned) and people (who cannot).

For publishing, this works quite well, allowing complex structures to be recorded at varying levels of complexity. For consuming, however, it creates an issue–some properties make sense to inherit upwards via partitioning, but others do not. For instance, a desk made of oak that has as a part a drawer made of brass is indeed made of oak and brass. However, France contains a city named Paris, but the country is not named both France and Paris. As far as we can tell, there is no general rule around which properties can be inherited from parts–individual implementations are possible, but future work in this area is needed to help aid in discovering and codifying general patterns.

One problem unsolved by partitioning is the conceptual groupings of entities. The CRM provide "E78 Curated Holding" to handle a very specific form of such grouping, but doesn't have a general pattern to reflect which are associated intellectually, but not through some containment structure. For example, an office may contain many chairs, and as such the chairs could be considered part of the office. But I own many chairs, and while the chairs are related to each other, they are not part of any larger entity–instead, they are intellectually associated with each other. Similarly, each of the instances of an exhibition are linked together, but they are not part of a larger event–instead, they are

[26] https://w3c.github.io/json-ld-syntax/#scoped-contexts

conceptually grouped together. Given the lack of a suitable class for this sort of grouping, linked.art imports the ORE Aggregation class, and declares it a subclass of E28 Conceptual Object.

A Practical Example: Provenance of Objects

Together, these patterns allow for an enormous amount of precision, even when working with data of varying completeness. An object's acquisition via auction, for example, can be partitioned into more granular activities, allowing us to describe purchases by groups of individuals. Each of these activities can be classified using terms from the AAT, such as `aat:300077989` (purchasing) or `aat:300393199` (commissioning) to help distinguish the precise details of the transaction. And complex transactions involving multiple objects can be treated as a conceptual grouping, associating all objects within an auction lot together intellectually, while not implying that they are somehow the same object. These transactions can also be documented within Linguistic Objects which refer to the transaction–either to the acquisition as a whole, or to individual parts (as in the case of lot descriptions or bid histories). These documents themselves can be classified with AAT concepts such as `aat:300026068` (auction catalogs).

Using a small number of simple patterns, a complex object history can be recorded, and can be interpreted at varying levels of granularity–a user only interested in the ownership of objects can ignore that the acquisition event is contained within a larger auction event, and an individual interested in the auctions themselves can ignore the details of individual transactions–but they can be linked together to present a complete picture to scholars of the art market.

The Problems of Context and Change

While these patterns have proven themselves in practice, there are additional areas where patterns have not yet been fully defined. The area with the most complexity comes when trying to do more than just describe the real world, but instead to describe the process of description, or to begin to talk about the provenance of data.

The mechanism for doing so within the CIDOC-CRM, E13 Attribute Assignment, is somewhat difficult to use. It mirrors the reification pattern from RDF in that it identifies the subject and the object, but does not allow one to explicitly state what the nature of that relationship might be. Instead, it prefers to use the `classified_as` pattern to indicate the nature of the activity in which the attribute assignment was. In order to use this to provide contextual statements about when specific properties were assigned, Linked.art (after discussion with the CIDOC-CRM SIG), uses the CRM predicates themselves as Types, allowing

an explicit description of what relationship is being defined between these two entities[27].

This pattern both allows us to add data provenance to assertions, which is particularly useful when we would like to record information that was formerly believed to be true, but is no longer held. It also allows us to record information within the context of a larger event–by partitioning an Event (such as an exhibition) and including as a part an Attribute Assignment, we can include the act of assigning exhibition-specific information.

Often, what is wanted is not the data provenance, but instead only the contextual information–we don't know anything about the activity of assigning the information, but we do know that several statements about an entity held true within a particular context. An exhibition number, perhaps, as well as a description within the catalog and an exhibition-specific title, are all assigned to an object within the context of that exhibition.

In this case, we can reuse the Conceptual Grouping pattern, and make use of the ORE pattern of proxies[28]. This pattern allows us to group many related statements within a proxy, and, only within the context of that proxy, assign those statements to the entity. This pattern is significantly simpler to consume, but loses the data provenance of those statements. However, as long as the link between the property and the Type is computable, the conversion from a collection of attribute assignments to a proxy can be automated. Conversely, a proxy can be converted to a series of attribute assignments without data loss.

There are issues with this pattern, however, when we begin to deal with changes over time. A static context, such as an exhibition, does not present problems–but a changing context, such as an inventory, does: once an object has been included within an aggregation, it is unclear whether or not it is *always* within that aggregation. The question is: are aggregations mutable or immutable? Is the context of "My favorite paintings" fixed in time, such that any description of that what is included in that context is only valid at a particular instant? Does it include every object that has ever been my favorite, even if it is no longer so? Or, instead, does that context have an identity that persists over time, and instead the membership of that context shifts over time (a pattern which is not supported by ORE proxies)[29]?

At it's core, this feels very similar to a data provenance problem. However, it is not the problem of recording change in knowledge over time–there are well-defined solutions to that problem such as Activity Streams[30], PROV-O[31] and Memento [32]. The problem of data provenance is that our knowledge is

[27] For a detailed description of this pattern, see https://linked.art/model/assertion/

[28] http://www.openarchives.org/ore/1.0/datamodel#Proxy

[29] This issue has been discussed at length in https://github.com/linked-art/linked.art/issues/147

[30] https://www.w3.org/TR/activitystreams-core/

[31] https://www.w3.org/TR/prov-o/

[32] http://www.mementoweb.org/guide/quick-intro/

constantly changing, and it is important to know whose knowledge it is and when that knowledge was held–PROV-O and Memento help us record and access documents that change over time. The problem of context, though, is that the objects we are describing are not themselves fixed or immutable–once they exist within a social context, they are themselves interpreted and change. Just because an artwork is no longer attributed to Titian doesn't mean that it wasn't at some point a Titian, at least in the context of that moment in time. To be able to truly ask the sorts of questions we would like to enable, we have to both be able to ask "What did we know about the object last year?" and "What was thought about the object in 1850?" Those two histories are different, and to truly enable understanding beyond the description of objects, and instead understand objects in context, we will need to be able to look at our objects from both perspectives at the same time.

Conclusion: Linked Open Usable Data

Together, these patterns provide a structure for using the CIDOC-CRM within the context of object-based memory institutions to publish and consume complex descriptions of event-based object histories. They follow a principle that we call "LOUD": Linked Open *Usable* Data: maintaining the expressibility and precision of the semantic web and the CIDOC-CRM, but using patterns, tools, and best practices to hide the complexity from our end users, software developers. The complexity is still there–not elided, merely waiting for the moment when it will need to be resurfaced to answer a question or enrich an interface in a way that cannot be done without use of the semantic underpinnings.

The semantic data is also there for a different audience: not the audience interested in the description, discovery, or display of information, but instead researchers who wish to interrogate the information to answer questions unanticipated by the publishers. This scholarly audience, working across large datasets, will take advantage of the rich capabilities enabled by the underlying graph, but they should not be privileged, since each researcher's needs will vary and each question will be unique.

The complexity of our information must be managed, and we must allow consumers to consume at the level of complexity they understand and need. Linked.Art attempts to demonstrate a technical and social solution for finding an appropriate balance between the needs of the developer and the needs of the scholar. We must work to make the information as simple as possible, because in order to fully make use of object descriptions within social contexts, the complexity of our description will only increase. The only way to build higher is to ensure that the foundations that we have are both strong and well-understood.