An Event-Aware Model for Metadata Interoperability

Carl Lagoze[1], Jane Hunter[2], Dan Brickley[3]

[1]Cornell University, lagoze@cs.cornell.edu
[2]DSTC Pty. Ltd., jane@dstc.edu.au
[3]ILRT, daniel.brickley@bristol.ac.uk

Abstract. We describe the ABC modeling work of the Harmony Project**.** The ABC model provides a foundation for understanding interoperability of individual metadata modules – as described in the Warwick Framework – and for developing mechanisms to translate among them. Of particular interest in this model is an *event*, which facilitates understanding of the lifecycle of resources and the association of metadata descriptions with points in this lifecycle.

1. Metadata Modularity and Interoperability

The Warwick Framework [22] describes the concept of modular metadata - individual metadata packages created and maintained by separate communities of expertise. A fundamental motivation for this modularity is to scope individual metadata efforts and encourage them to avoid attempts at developing a universal vocabulary. Instead, individual metadata efforts should concentrate on classifying and expressing semantics tailored toward focused functional and community needs. Warwick Framework-like modularity underlies the design of the W3C's Resource Description Framework (RDF) [15, 23], which is a modeling framework for the integration of diverse application and community-specific metadata vocabularies.

An outstanding challenge of such modularity is the interoperability of multiple metadata packages that may be associated with and across resources. Metadata packages are by nature not semantically distinct, but overlap and relate to each other in numerous ways. Achieving interoperability between these packages via one-to-one crosswalks [4] is useful, but this approach does not scale to the many metadata vocabularies that will continue to develop. A more scalable solution is to exploit the fact that many entities and relationships - for example, people, places, creations, organizations, events, and the like - are so frequently encountered that they do not fall clearly into the domain of any particular metadata vocabulary but apply across all of them.

The Harmony Project [6] is investigating this more general approach towards metadata interoperability and, in particular, its application in multimedia digital libraries. This approach, the *ABC model and vocabulary*, is an attempt to:

- formally define common entities and relationships underlying multiple metadata vocabularies;
- describe them (and their inter-relationships) in a simple logical model;
- provide the framework for extending these common semantics to domain and application-specific metadata vocabularies.

The concepts and inter-relationships modeled in ABC could be used in a number of ways. In particular, individual metadata communities could use these underlying concepts (the ABC model) to guide the development of community-specific vocabularies. These individual communities could use formalisms such as RDF to express the possibly complex relationships between the ABC model and their community-specific vocabularies. Furthermore, the formal expression of the relationships between community-specific vocabularies and the ABC model could provide the basis for a more scalable approach to interoperability among multiple metadata sets. Rather than one-to-one mappings among metadata vocabulary semantics, a more scalable basis for interoperability could be achieved by mapping through this common logical model.

This paper describes the initial results of our work on the ABC model, the main focus of which is describing *events* and their role in metadata descriptions. Briefly stated, our argument is as follows. Understanding the relationship among multiple metadata descriptions (and ultimately the vocabularies on which they are based) begins by understanding the entities (resources) they purport to describe. Understanding these entities entails a comprehension of their lifecycle and the events, and corresponding transitions and transformations, that make up this lifecycle.

This work is influenced by and builds on a number of foundations. The significance of events and processes in understanding knowledge has deep routes in philosophy [12]. The importance of processes and events in resource descriptions has been recognized by a number of communities including the bibliographic community [5], museums [21], the archival community [11], and those concerned with e-commerce and rights management [7]. Events, and their role in metadata interoperability, were recognized in [10]. Our modeling principles are influenced by work in the W3C and related communities, where both the XML Schema [13, 27] and RDF Schema [15] initiatives are evolving with the goal of formally modeling and representing data (and metadata) on the Web. These efforts and our own build on work in the database community to understand, model, and query semi-structured data [9].

The remainder of this paper is structured as follows. Section 2 describes the relevance of events for understanding the relationship between metadata vocabularies. Section 3 situates the concept of events and resource relationships within the broader ABC logical model. Section 4 then presents a formal model of events using UML [14] and then expresses this model using the XML Schema language. Section 5 uses this model to describe a compound multimedia example. The paper closes with Section 6 that describes future directions.

2. Event-Aware Metadata

In January, 2000 the Harmony Project sponsored a workshop [1] that brought together representatives of several metadata initiatives to discuss interoperability. Subsequent to establishing shared perspectives and goals, the workshop focused on the importance of events in understanding intellectual resources, the nature of various descriptions of them, and the relationships between these descriptions. There was general agreement that a model to facilitate mapping between metadata vocabularies needs to be *event-aware*. This requirement builds on a number of observations about resources and descriptions. These observations are as follows.

As described in the IFLA FRBR (Functional Requirements for Bibliographic Records) [5], intellectual content evolves over time. The taxonomy developed in FRBR is a useful foundation for understanding the lifecycle of a single resource: it begins as a conceptual *work*, it may evolve into one or more *expressions* (e.g., an opera, a story, a ballet), these expressions may be realized in one or more *manifestations* (e.g. an edition or printing of a story in book form), and eventually these manifestations are disseminated as individual items (e.g., an individual copy of a book). The FRBR model largely applies to the evolution of a single resource; the subtleties of inter-resource relationships and the derivative nature of relationships between them also need to be understood.

An important aid towards understanding this evolution of an individual resource and the derivative relationships between resources is to characterize the events that are implicit in the evolution or derivation. For example, the evolution from *work* to *expression* may contain an implicit *composing event*. The process of making implicit events explicit – making them *first-class objects* – may then provide attachment points for common descriptive concepts such as agency, dates, times, and roles. A model that explicitly represents the attachment of these concepts to events may be useful for mapping between metadata vocabularies that express these concepts.

Events are also important in understanding metadata descriptions because of the way that they transform "input" resources into "output" resources, and the respective descriptions (or metadata) for those input and output resources. In particular, an event is important from a certain descriptive community's perspective because of the way the event changes a property of a resource that is of interest to that community. While an event changes one or more properties of a resource, other properties remain unchanged. For example, a "translation event" of War and Peace" may change its language from Russian to English, but its author is still Fyodor Dostoyevsky.

Descriptive communities can be distinguished by the events that are of significance to them. For example, a community that focuses on the history of production of a film may consider the "event" associated with the insertion of a certain scene into a film significant. As a result that event may be explicit in their descriptive vocabulary – for example, that community may have a metadata attribute that describes the date of the scene insertion. Another community, say one concerned with the presentation of that film on a screen, may consider that event irrelevant and may consider the "is part of" relationship of the scene to the movie completely non-event related.

A particular metadata description is often a portrayal of a *snapshot* of some entity taken in a particular state - a perceived stability of the entity over a particular time and place that perforce elides events or lifecycle changes that are outside the domain of interest by the particular descriptive community. The granularity of that snapshot (and the number of elided or revealed events) varies across metadata vocabularies. For example, a Dublin Core description [3], intended for relatively basic resource discovery, is a particularly coarse granularity snapshot. A Dublin Core description of a postcard of the Mona Lisa might list Leonardo Da Vinci as the creator even though numerous events took place on the portrayal of the Mona Lisa since the depiction by Da Vinci. On the other hand, an INDECS [7] description, for which the events associated with transfers of rights are extremely important, might describe more fine-grained event snapshots.

These observations suggest the following intellectual, and ultimately, mechanical approach towards understanding the relationships between metadata vocabularies:

- Develop a consistent and extensible model for events. This is the main subject of the remainder of this paper.
- Analyze the nature of the snapshots underlying the descriptions. For example, a coarse granularity Dublin Core description of a resource may combine attributes that span a number of transitions in the lifecycle of the resource. An INDECS description of what may at first seem like the "same" resource may actually focus on a smaller snapshot such as the attributes associated with a single transfer of rights in a contractual transaction.
- Attempt to interpolate and model the events that are contained within these snapshots and model these event transitions. For example, a single Dublin Core record may contain information about an agent who is a creator, an agent who is a translator, and an agent who is a publisher. This implies that the DC record actually describes a snapshot that implicitly contains three events: creation, translation, and publishing. Modeling these events would then permit the explicit linkage between the attributes of the respective description with the corresponding event (e.g., associating the "Creator Agent" with the "agent event and associating the "Creation Date" of the description with the same event).
- Examine the overlap between the snapshots described by the individual descriptions. For example, the set of events implicit within an INDECS description may be fully contained within the broader snapshot of events within a DC description.
- Examine the relationship of the events in the event-aware models of the individual descriptions and of the properties that are associated with those events. Such event-aware analysis may make it possible to establish the relationship between the vocabulary-specific properties that "map down" to these events.

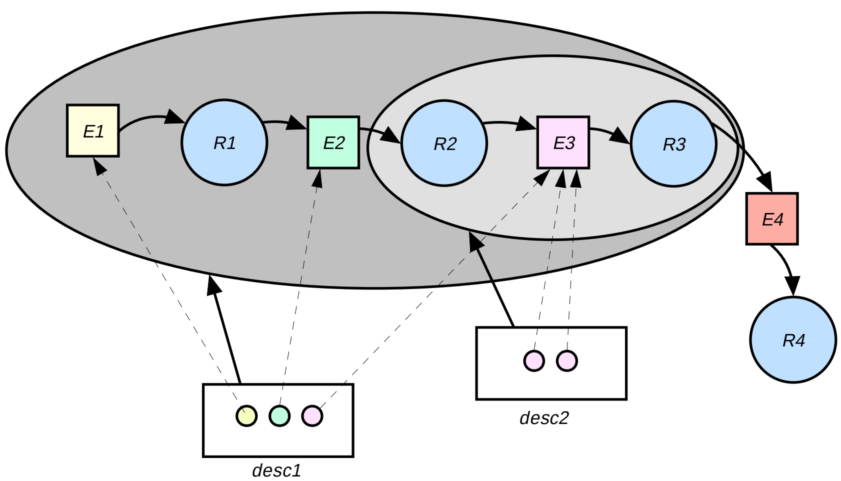

Figure 1 - Metadata and events

These concepts are illustrated in Figure 1. The larger circles represent manifestations of a resource as it moves through a set of event transitions; the events are represented by the squares interspersed between the circles. For example, event *E1* may be a creation event that produces resource *R1*. This resource may then be acted on by a translation event - event *E2* - producing resource *R2* and so on. The rectangles at the bottom of the figure represent metadata descriptions (instances of particular metadata vocabularies), and the ellipses that enclose part of the resource/event lifecycle represent the snapshot of the lifecycle addressed by that particular metadata description. For example, the larger dark-shaded ellipse represents the snapshot described by *desc1*, and the smaller light-shaded ellipse the snapshot described by *desc2*. The smaller circles within each descriptive record are the actual elements, or attributes, of the description. The dotted lines (and the color of each circle) indicate the linkage of the metadata element to an event - as shown the elements in d*esc1* are actually associated with three different events that are implicit in the snapshot. For example, the attributes (moving from left to right) may describe *creator, translator,* and *publisher,* which are actually “agents” of the events. As shown, the three rose colored elements are all associated with a single event *E3*, implying a relationship between them that can be exploited in mapping between the two descriptive vocabularies that form the basis for the different descriptions.

3. The ABC Logical Model for Metadata Interoperability

The ABC logical model is built on a number of fundamental concepts and assumptions including universally identified resources, properties (as a special type of resource), and classes that create sets of resources (and properties). The model also defines a set of fundamental classes (sets of resources) including creations, events, agents, and relationships. These fundamental classes provide the building blocks for expression (through sub-classing) of application-specific or domain-specific metadata vocabularies. The reader is referred to [16] for more details on the complete model.

Of particular applicability to this paper is the multiple-view modeling philosophy in ABC. This allows properties (relations between resources) to be expressed in a simple binary manner or in a more complex manner that promotes the relation to a first-class resource. These first-class resources then provide the locus for associating properties that describe the relation.

Resources are related in numerous ways: containment, translation, and derivation are but three of the more common relations. Describing these relations is an important aspect of metadata. In some vocabularies (e.g., Dublin Core) these relation descriptions are rather simple; in others there is the need for increased descriptive power. ABC (by adopting RDF's graph data model) allows us to move between simple and complex relation descriptions as follows. We create a model in which the entity that is the input to the relation, the entity that is the output of the relation, and the relationship between the two entities are all represented as *resources*. In order to more richly describe these resource, we can then associate properties with them. In this manner, we have promoted - “reified” - a simple *relationship arc* to a first class resource and associated properties with it. For certain applications, the complicated, explicit model is most useful; other times it is better to have a simple, flattened representation of the 'real' state of affairs. In both cases it is useful to understand how the two representations inter-relate.

The example in Figure 2 illustrates this point with the “hasTranslation” relation. We can take a simple view and say just that some document has a translation into another document.

Figure 2 - Simple Resource Relationship

An alternative is shown in Figure 3 where we take a complex view and promote the *hasTranslation* relationship to a first class event resource. We can associate properties with that event to describe its details, such as its agents and its inputs and outputs. These details are the subject of Section 4.

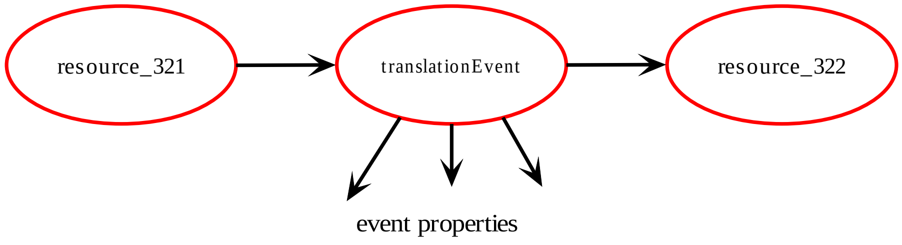

Figure 3 - Promoting Relationship to a First-Class Resource

This approach is applicable to a cross-section of events that have input and output resources or that describe an agent’s contribution to a resource. Examples of such event/relation pairings include...

- Modification event <=> *VersionOf* relation
- Compilation event <=> *CompiledFrom* relation;
- Extraction event <=> *ExtractedFrom* relation;
- Reformat event <=> *IsFormatOf* relation;

- Translation event <=> *TranslationOf* relation;
- Derivation event <=> *DerivedFrom* relation.

In such cases, ABC provides two representational options and recipes for inter-conversion. When rich information is required, ABC provides the event model. This involves describing the *event* through which that relationship was realised as an object in itself, describing the hidden detail implicit in a simple binary relation. When concise/simple metadata is needed, flatter relations are used.

4. Modeling Events using the ABC Vocabulary

The goal of the ABC vocabulary is to define and declare a core set of abstract base classes that are common across metadata communities. These base classes are intended to provide the attachment points for different properties (or metadata) that are associated with information content and its lifecycle. They will provide the fundamental infrastructure for modeling metadata and for refinement through sub-classing.

A review of a number of metadata models (including IFLA [5], CIDOC [21], INDECS [26], MPEG-7 [8], and Dublin Core[3]) reveals the following common entities:

- *Resources*
- *Events*
- *Inputs* and *Outputs*
- *Acts* (and associated *Acts* and *Roles*)
- *Context* (consisting of *Time* and *Place*)
- *Event Relations*

A UML model of these entities and their relationship to each other is shown in the UML [25] model illustrated in Figure 4. This model can be represented declaratively in a schema definition using XML DTDs, RDF Schema [15] or XML Schema Language [13, 27]. We have chosen to provide an XML Schema representation in Figure 5.

The remainder of this section describes the entities in this model and our basic approach to an underlying metadata modeling framework. This approach is not final and will continue to be refined through implementation and feedback.

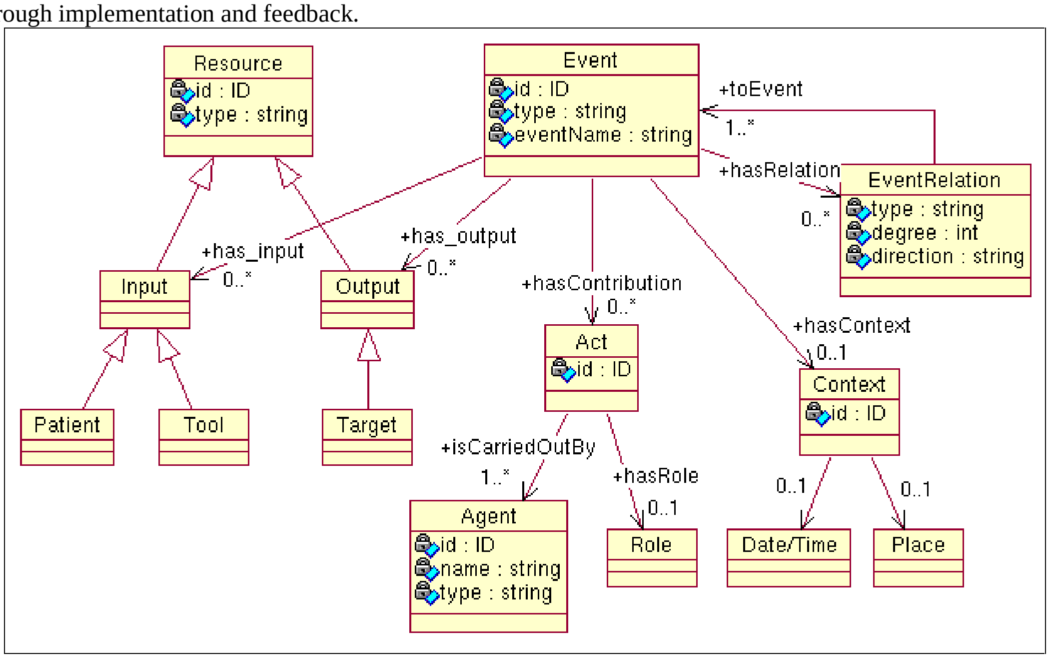

Figure 4: UML representation of the basic event model

```xml
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">

  <xsd:element name="Event" type="eventType"/>

  <xsd:complexType name="eventType">
     <xsd:element ref="Context" minOccurs="1" maxOccurs="1"/>
     <xsd:element ref="Act" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element name="EventRelations" type="EventRelations"
           minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element name="Input" type="InputType" minOccurs="0" maxOccurs="1"/>
     <xsd:element name="Output" type="OutputType" minOccurs="0" maxOccurs="1"/>
     <xsd:attribute name="id" type="xsd:ID"/>
     <xsd:attribute name="type" type="xsd:string"/>
     <xsd:attribute name="eventName" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="EventRelations">
     <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:attribute name="type" type="EventRelationType"/>
    <xsd:attribute name="degree" type="xsd:nonNegativeInteger" />
    <xsd:attribute name="direction" type="directionType"/>
  </xsd:complexType>

  <xsd:element name="Context">
    <xsd:complexType>
       <xsd:element name="Place" type="xsd:string" minOccurs="0" maxOccurs="1"/>
       <xsd:element name="Date" type="xsd:date" minOccurs="0" maxOccurs="1"/>
       <xsd:element name="Time" type="xsd:time" minOccurs="0" maxOccurs="1"/>
       <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Act">
     <xsd:complexType>
       <xsd:element ref="Agent" minOccurs="1" maxOccurs="1"/>
       <xsd:element name="Role" type="xsd:string" minOccurs="1" maxOccurs="1"/>
       <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Agent">
    <xsd:complexType>
       <xsd:element name="name" minOccurs="1" maxOccurs="1"/>
       <xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1"/>
       <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="ResourceType">
    <xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

  <xsd:complexType name="InputType" base="ResourceType" derivedBy="extension"/>
  <xsd:complexType name="Patient" base="InputType" derivedBy="extension"/>
  <xsd:complexType name="Tool" base="InputType" derivedBy="extension"/>

  <xsd:complexType name="OutputType" base="ResourceType" derivedBy="extension"/>
  <xsd:complexType name="Target" base="OutputType" derivedBy="extension"/>

</xsd:schema>
```

Figure 5: XML schema of event model

4.1. Resources

These represent the superclass of all of the possible things within our universe of discourse - they may be physical, digital or abstract. Every resource has a corresponding unique identifier.

4.2. Events

An event is an action or occurrence. Every event has a Context (Time and/or Place) associated with it (although it may not always be explicit). Events may also have inputs and/or outputs associated with them. For example, events which generate a new or transformed resource (e.g. translation, modification) will have both input(s) and output(s).

The event class has the following properties:

- An eventType property (which may be enumerated);
- An optional eventName property;
- Optional input and output resources;
- Zero or more Act properties - which describe the contributions made by various agencies to the event;
- Zero or more EventRelations. These are relationships with other events and include relations such as the *contains* relation to define subEvents.

4.3. Inputs and Outputs

Events can have *Input* resources and/or *Output* resources.

Input resources vary in that some inputs are actually operated on during the event (Patients) whilst others are simply tools or references which are used during the event (Tools). The *Patient* and *Tool* subclasses of Input have been provided to support this distinction. This is important to avoid ambiguity during the complex-to-simple transformation when there are multiple inputs. Sometimes it may be difficult to determine when a resource should be defined as an Input Tool and when it should be defined as an Agent. If there is a need to define the *Role* of the Input resource, then it must be defined as an Agent class.

Output resources vary in that some resources are the primary target outputs whilst others (e.g. messages) are of secondary importance. This distinction is important during metadata simplification in order to determine which Inputs and Acts are associated with which Output resources. The *Target* subclass is provided to prevent ambiguity and clearly specify the target output resources. Target output resources are assigned the Role/Agent property/value pairs during the complex-to-simple transformation.

4.4. Acts

An Act is a contribution to an event which is carried out by one or more actors or agents playing particular roles. An Act can only exist as a property of an Event. Each Act has one or more Agent properties and an optional Role property.

4.5. Agents

Agents represent the resources which act in an event - or the "actors" in an event. *Agents* are properties of *Acts* and usually have (through those Acts) an associated *Role* which defines the role that this actor plays in the particular event. The precise model by which agent roles are described is an area of ongoing research within Harmony. Some commonly-used agent types are:

- person/human being;
- organisation;
- instrument (hardware,software, machine).

In reality, any resource may take a causative role, thus allowing it to act as an agent. Additional possible agent types include: animals, fictional animals (Teletubbies), aliens, supernatural beings, imaginary creatures, inanimate objects (e.g., a painting that falls from a wall and strikes a sculpture, which shatters and then is presented as a new resource in a museum show), natural or environmental processes (storms, plagues, erosion, decay etc.).

4.6. Context

Date/Time. Time can be specified in a variety of ways. It can be either free text describing a period or event or a specific date/time format. It may also be either an instantaneous time or a time span. It may be GMT, local time or a time relative to a particular object's scope e.g. a time stamp in a video. Some examples include:

- The Battle of Hastings
- The 20th Century
- Next Year
- 21-10-99
- 00:07:14;09 - 00:12:36;21

Place. The place entity describes a spatial location. It can be free text or formatted. It can be absolute or relative. It can be a point, line, 2D or 3D region. Similarly to time, place can vary enormously in granularity. It may be a real world spatial location or a spatial location relative to a particular origin, coordinate system or objects' dimensions. Some examples of valid place values are shown below:

- 24 Whynot St, West End
- 0, 0, 100, 100
- Mars
- latitude, longitude
- the bottom left hand corner i.e. a section of a digital or physical object

4.7. EventRelations

EventRelations are provided to express relationships between Events. Typical top-level subtypes of EventRelations include: temporal, spatial, spatio-temporal, causal, conditional. Each of these may have enumerated subtypes e.g. temporal relations may include: *precedes, meets, overlaps, equals, contains, follows*. EventRelations may also have direction (*uni-directional, bi-directional*) and degree (unary, binary, n-ary) attributes associated with them. Conditional relations will have one or more condition statements associated with them.

5. Applying the Model to a Complex Object

The following example of a complex object was developed at the January 2000 Harmony workshop.
A 65 min video (VHS) of a "Live at Lincoln Center Performance". The conductor is Kurt Masur. The Orchestra is the New York Philharmonic. The performance was on April 7, 1998 at 8PM Eastern Time. The performance was broadcast live and recorded by the BBC. The direction and program notes (in English) were by Brian Large. The two pieces performed are:

- The Rite of Spring by Igor Stravinsky, written in 1911. Its length is 35 minutes
- Concerto for Violin by Phillip Glass written in 1992. With Robert McDuffie solo on the Violin. Its length is 25 minutes.

Figure 6 is an RDF model of the scenario based on the ABC vocabulary. The performance actually consists of 3 parts or sub-events, *event1_1, event1_2, event1_3*. *event1_1* and *event1_2* are the sequential performance sub-parts which are expressions of separate *concepts* or *works*. Figure 7 illustrates the preceding creation event which produced the composition concept which was input to *event1_2*. *Event1_3* is the ProgramNotesProduction Event. It needs to be separately defined to ensure that the Agent/Role pair of Brian Large/Note Producer is associated with the "ProgramNotes" output resource. Appendix A contains an XML instantiation of this scenario based on the XML Schema in Figure 5.

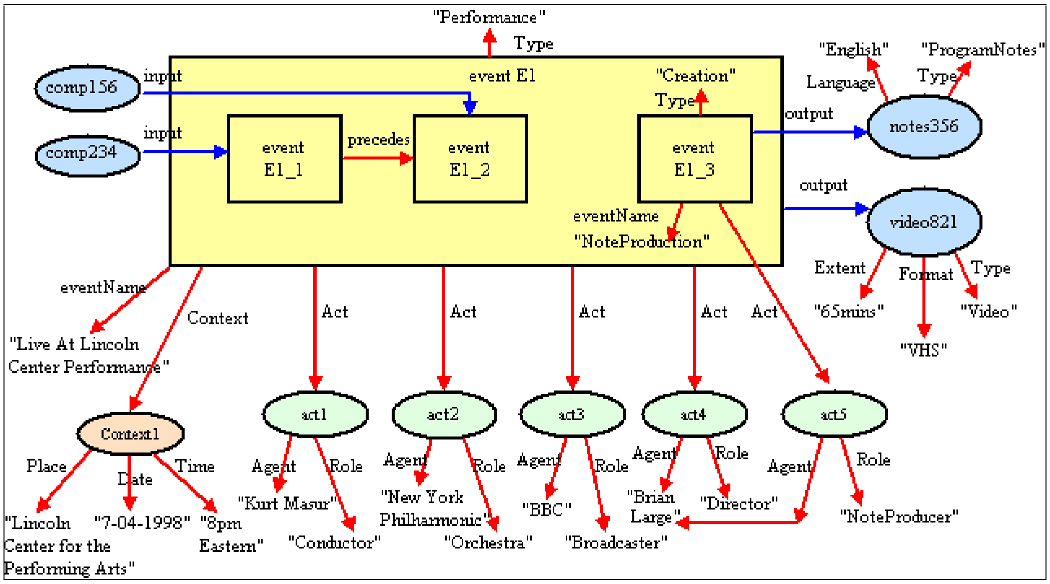

Figure 6: Event-aware Model Representation of the Scenario

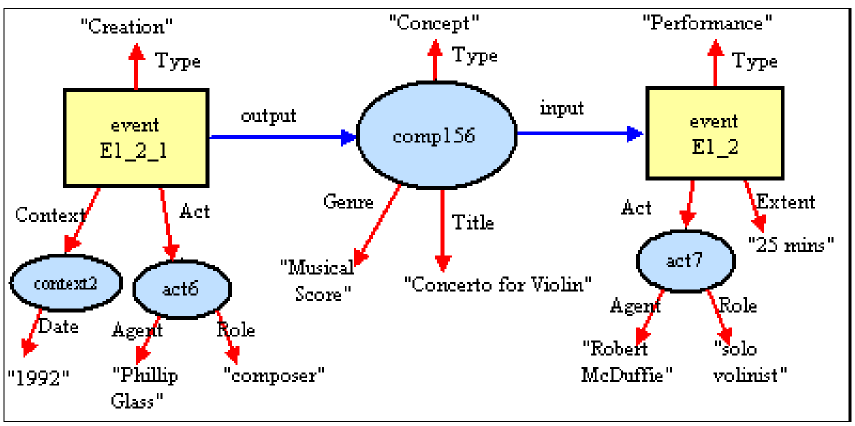

Figure 7: Event-aware Model of A Performance of 'Concerto for Violin'

6. Next Steps

The modeling concepts described in this paper are the first stage of our work within Harmony to understanding mappings among the metadata schemas from different domains. We close with some observations on how these mappings might work for some representative metadata vocabularies and on the possible mechanisms for performing such mappings.

The complexity of the mapping process varies according to the metadata vocabulary. For domains that use a flat unstructured resource-centric metadata model (e.g., Dublin Core, AACR [20]), the mapping process can be broken down into two steps: transformation from the ABC event-aware model to the resource-centric model, and mapping of the ABC semantic elements to the specific domain's semantic elements. Figure 8 illustrates these steps in mapping from an ABC description of the (simplified) scenario performance to a Dublin Core description of the video. Although the MPEG-7 data model is not explicitly event-aware, it does support the concept of time-based segmentation within audiovisual documents, which reflects the sequence of the original events which were recorded. One approach is to map the ABC model's descriptions of actual real-world events to descriptions of segments within the audiovisual content. Since the CIDOC/CRM and INDECS models both use an event-aware metadata model, it is expected that the structural mapping process from ABC to these schemes (step 1 in Figure 8) will be relatively simple.

There are a number of possible mechanisms available for the mapping process. Some of these are non-procedural, including:

- merging XML Infosets into a single composite Infoset [18, 24];
- using Equivalence classes within XML Schema Language to define mappings [27];
- using XSLT (XSL Transformation Language) [17] to transform an XML description from one domain to another.

We expect, however, that none of these approaches will be able to cope with mapping between the broad range of community-specific semantics which can be "dropped in" within the unifying framework provided by ABC. Recognizing this, we also plan to investigate a number of proposals for a logic language expressed over the RDF data model, which may be useful for this purpose, such as [2, 19].

In the end these investigations and mechanisms will need to take into account a theme common across the metadata field. Expressive power is often desirable for metadata descriptions, but expressiveness comes at the cost of complexity. The success of any model and mechanisms for mapping among multiple descriptive vocabularies will be measured by whether it is feasible to build usable and deployable systems that implement them.

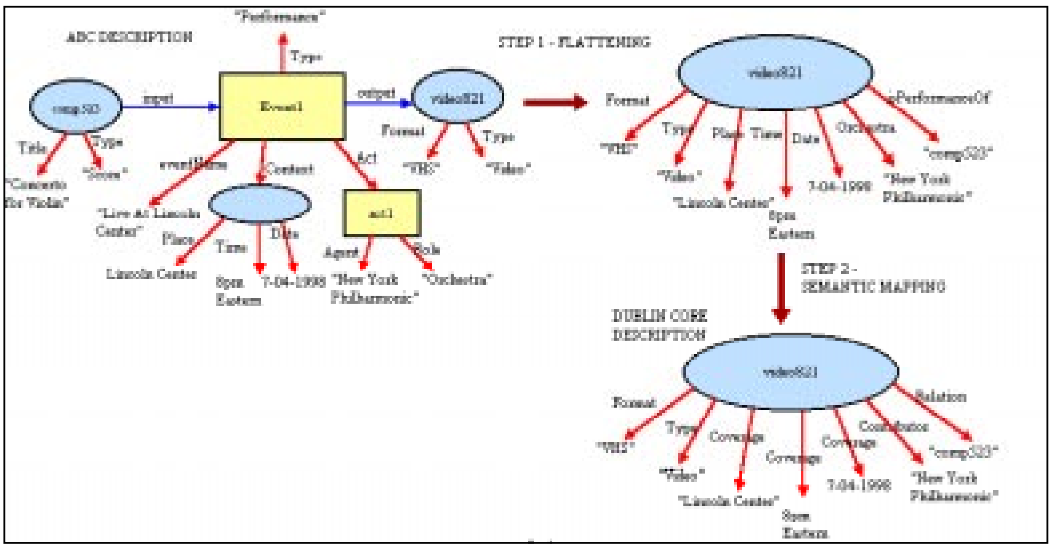

Figure 8: The 2-step Mapping Process from ABC to Dublin Core

Acknowledgements

The authors wish to acknowledge the contributions to this work by the participants at the ABC Workshop in January, 2000: Tom Baker, Mark Bide, David Bearman, Elliot Christian, Tom Delsey, Arthur Haynes, John Kunze, Clifford Lynch, Eric Miller, Paul Miller, Godfrey Rust, Ralph Swick, and Jennifer Trant. Support for the work in this document came from a number of sources including NSF Grant 9905955, JISC Grant 9906, and DSTC Pty Ltd.

References

[1] *ABC Workshop*, http://www.ilrt.bris.ac.uk/discovery/harmony/abc_workshop.htm.

[2] *DARPA Agent Mark up Language (DAML)*, http://www.oasis-open.org/cover/daml.html.

[3] *Dublin Core Metadata Initiative*, http://purl.org/DC.

[4] *Dublin Core/MARC/GILS Crosswalk*, http://lcweb.loc.gov/marc/dccross.html.

[5] "Functional Requirements for Bibliographic Records," International Federation of Library Associations and Institutions, http://www.ifla.org/VII/s13/frbr/frbr.pdf, March 1998.

[6] *The Harmony Project*, http://www.ilrt.bris.ac.uk/discovery/harmony/.

[7] *INDECS Home Page: Interoperability of Data in E-Commerce Systems*, http://www.indecs.org/.

[8] "MPEG-7 Requirements Document," International Organisation for Standardisation, Requirements IDO/IEC JTC1/SC29/WG11, October 1998.

[9] S. Abiteboul, P. Buneman, and D. Suciu, *Data on the web: from relations to semistructured data and XML*. San Francisco: Morgan Kaufmann, 2000.

[10] D. Bearman, G. Rust, S. Weibel, E. Miller, and J. Trant, "A Common Model to Support Interoperable Metadata. Progress report on reconciling metadata requirements from the Dublin Core and INDECS/DOI Communities," *D-Lib Magazine*, 5 (January 1999), http://www.dlib.org/dlib/january99/bearman/01bearman.html, 1999.

[11] D. Bearman and K. Sochats, "Metadata Requirements for Evidence.," Archives & Museum Informatics, University of Pittsburgh, School of Information Science, Pittsburgh, PA http://www.lis.pitt.edu/~nhprc/BACartic.html, 1996.

[12] J. F. Bennett, *Events and their names*. Indianapolis: Hackett Pub. Co., 1988.

[13] P. V. Biron and A. Malhotra, "XML Schema Part 2: Datatypes," World Wide Consortium, W3C Working Draft WD-xmlschema-2-2000025, http://www.w3.org/TR/xmlschema-2/, April 7 2000.

[14] G. Booch, J. Rumbaugh, and I. Jacobson, *The unified modeling language user guide*. Reading Mass.: Addison-Wesley, 1999.

[15] D. Brickley and R. V. Guha, "Resource Description Framework (RDF) Schema Specification," World Wide Web Consortium, W3C Candidate Recommendation CR-rdf-schema-20000327, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/, March 27 2000.

[16] D. Brickley, J. Hunter, and C. Lagoze, "ABC: A Logical Model for Metadata Interoperability," Harmony Project, Working Paper, http://www.ilrt.bris.ac.uk/discovery/harmony/docs/abc/abc_draft.html, 1999.

[17] J. Clark, "XSL Transformations (XSLT)," World Wide Web Consortium, W3C Recommendation REC-xslt-19991116, http://www.w3.org/TR/xslt, November 16 1999.

[18] J. Cowan and D. Megginson, "XML Information Set," World Wide Web Consortium, W3C Working Draft WD-xml-infoset-19991220, http://www.w3.org/TR/xml-infoset, December 20 1999.

[19] D. Fensel, I. Horrocks, F. Van Marmelen, S. Decker, M. Erdmann, and M. Klein, "OIL in a Nutshell," Vrije Universiteit Amsterdam, Amsterdam, http://www.cs.vu.nl/~dieter/oil/oil.nutshell.pdf, 1999.

[20] M. Gorman, *The concise AACR2, 1988 revision*. Chicago: American Library Association, 1989.

[21] ICOM/CIDOC Documentation Standards Group, *CIDOC Conceptual Reference Model*, http://www.ville-ge.ch/musinfo/cidoc/oomodel/.

[22] C. Lagoze, C. A. Lynch, and R. D. Jr., "The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata," Cornell University Computer Science, Technical Report TR96-1593, http://cs-tr.cs.cornell.edu:80/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593?abstract=, June 1996.

[23] O. Lassila and R. R. Swick, “Resource Description Framework: (RDF) Model and Syntax Specification,” World Wide Web Consortium, W3C Proposed Recommendation PR-rdf-syntax-19990105, http://www.w3.org/TR/PR-rdf-syntax/, January 1999.

[24] J. Marsh and D. Orchard, “XML Inclusions,” World Wide Web Consortium, W3C Working Draft WD-xinclude-20000322, http://www.w3.org/TR/2000/WD-xinclude-20000322, March 22 2000.

[25] Object Management Group, “OMG Unified Modeling Language Specification Version 1.3,” , OMG Specification, http://www.omg.org/cgi-bin/doc?ad/99-06-08.pdf, June.

[26] G. Rust and M. Bide, “The INDECS Metadata Model,” http://www.indecs.org/pdf/model3.pdf, July 1999 1999.

[27] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, “XML Schema Part 1: Structures,” World Wide Web Consortium, W3C Working Draft WD-xmlschema-1-2000225, http://www.w3.org/TR/xmlschema-1/, April 7 2000.

Appendix A: XML Schema Instantiation of the Scenario

The instance document below is a description of the scenario (Event E1) in Figure 6, which conforms to the XML Schema representation of the ABC model in Figure 5.

```
<Event id="E1" type="Performance" eventName="Live At the Lincoln Centre">

   <Context>
        <Date>7/4/98</Date><Time>20:00</Time>
        <Place>Lincoln Centre for the Performing Arts</Place>
   </Context>

   <Act id ="Act1">
        <Agent>
            <name>Kurt Masur</Name>
            <type>Person</Type>
        </Agent>
        <Role>Conductor</Role>
   </Act>

   <Act id="Act2">
        <Agent>
            <name>New York Philharmonic</Name>
            <type>Group of Persons</Type>
        </Agent>
        <Role>Orchestra</Role>
   </Act>

   <Act id="Act3">
        <Agent>
            <name>BBC</Name>
            <type>Producer</Type>
        </Agent>
     <Role>Producer</Role>
   </Act>

   <Act id="Act4">
        <Agent>
            <name>Brian Large</Name>
            <type>Person</Type>
        </Agent>
        <Role>Director</Role>
   </Act>

   <EventRelation type="contains">

     <sequence>

       <Event id="E1_1" type="Performance" eventName="The Rite of Spring">
         <Extent>35 mins</Extent>
         <Input id="comp234">
            <type>Musical Score</Type>
            <title>The Rite of Spring</Title>
         </Input>
       </Event>
```

```
       <Event id="E1_2" type="Performance" eventName="Concertofor Violin">
         <Extent>25 mins</Extent>
         <Input id="comp156">
            <type>Musical Score</Type>
            <title>Concerto for Violin</Title>
         </Input>
         <Act id="Act7">
            <Agent>
               <name>Robert McDuffie</Name>
               <type>Person</Type>
            </Agent>
            <Role>Solo Violinist</Role>
         </Act>
       </Event>

     </sequence>

     <Event id="E1_3" type="Creation" eventName="ProgramNotesProduction">
         <Act id="Act5">
            <Agent>
               <name>Brian Large</Name>
               <type>Person</Type>
            </Agent>
            <Role>Notes Producer</Role>
         </Act>
         <Output id="notes356">
            <type>Program Notes</Type>
            <language>English</Language>
         </Output>
     </Event>

   </EventRelation>

   <Output id="video821">
        <type>video</Type>
        <extent>65 mins</Duration>
        <format>VHS</Format>
   </Output>

</Event>
```