The ABC Ontology and Model

Carl Lagoze
Cornell University
Ithaca, NY
lagoze@cs.cornell.edu

Jane Hunter
DSTC Pty, Ltd.
Brisbane, Australia
jane@dstc.edu.au

Abstract

This paper describes the latest version of the ABC metadata model. This model has been developed within the Harmony international digital library project to provide a common conceptual model to facilitate interoperability between metadata ontologies from different domains. This updated ABC model is the result of collaboration with the CIMI consortium whereby earlier versions of the ABC model were applied to metadata descriptions of complex objects provided by CIMI museums and libraries. The result is a metadata model with more logically grounded time and entity semantics. Based on this model we have been able to build a metadata repository of RDF descriptions and a search interface which is capable of more sophisticated queries than less-expressive, object-centric metadata models will allow.

Keywords: Metadata, Modeling, Ontologies

1 Introduction

The Harmony Project [10] is an international digital library project funded by DSTC (Australia), JISC (U.K.), and the NSF (U.S.). The broad goal of the project is to research methods and models for describing the variety of rich content that increasingly populates the Web and digital libraries.

The paper describes recent results in the development of a metadata model and ontology. The so-called *ABC Model* (a purposely innocuous name) was first articulated in an early Harmony working paper [21] and was later documented in a number of conference papers [25, 26, 30]. The initial and ongoing goal of work on the ABC model is three-fold:

- To provide a conceptual basis for understanding and analyzing existing metadata ontologies and instances.
- To give guidance to communities beginning to examine and develop descriptive ontologies.
- To develop a conceptual basis for automated mapping amongst metadata ontologies.

As such, the ABC ontology is not intended as a metadata vocabulary per se, but as a basic model and ontology that provides the notional basis for developing domain, role, or community specific ontologies. In this spirit, the ABC model incorporates a number of basic entities and relationships common across other metadata ontologies including time and object modification, agency, places, concepts, and tangible objects. Communities wishing to build their own metadata ontologies and models may then extend the ABC entities and relationships as needed.

The initial version of the model, described in [21] benefited from contacts and collaborations with a number of communities and efforts including the Dublin Core Metadata Initiative [7], the IFLA Functional Requirements for Bibliographic Records [9], the INDECS E-Commerce Metadata Model [12], and the work among many of those communities to articulate a common conceptual model [17]. The updated version of the model, described in this paper, benefits from collaborations with the museum community represented by both the CIDOC/CRM [3] and the CIMI [5] consortium. The modeling methodology continues to build on concepts from the Resource Description Framework (RDF) [31] of the World Wide Web Consortium [14], but should also be applicable to other modeling paradigms with roots in first-order logic such as UML [19] or Conceptual Graphs [36].

The remainder of this paper is structured as follows:

- Section 2 describes the scope and intended purpose of the ABC model;
- Section 3 summarizes the basic components of the model, paying special attention to its temporal components that permit modeling the lifecycle aspects of entities;
- Section 4 is a complete reference of the classes and properties that constitute the ABC model;
- Section 5 illustrates the application of the ABC model on both fictional and actual (CIMI) examples;

- Section 6 describes the metadata repository and search interface developed to enable sophisticated queries across all of the CIMI images from a single user interface;
- Section 7 concludes with our anticipated future work directions.

2 Purpose and Scope of the ABC Model

2.1. Origins

Work on the ABC data model and ontology is motivated by the recognition that many existing metadata efforts often proceed with insufficient attention to underlying modeling principles. Failure to pay due attention to more formal principles has frequently led to attempts to express complex resource descriptions without a clear exposition of the entities and relationships necessary for such descriptions. Such informality may be appropriate for simple "pidgin" metadata [16], such as Dublin Core, but lacks precision for detailed descriptions [29].

We argue that one essential test of a descriptive model should be the specificity of queries that it supports. If the intent is to support simple Boolean queries on fields such as *"return all documents authored by Tom Baker and with 'Grammar' in the title"* then it is reasonable to build descriptions as a record of attributes and their *appropriate literal* values [16]. We have found that, especially in our work with the museum community, creators of metadata frequently want more advanced query semantics, which include attributes of multiple entities and ask questions about *"who was responsible for what, when and where"*. In order to support such queries, a metadata model must provide a logical foundation for temporal semantics and consistent attachment points for the agents, actions, and situations involved in these temporal semantics.

As a parenthetical remark, we emphasize that support for more advanced queries almost certainly increases the human effort and the expense of creating resource descriptions (i.e., the Arms cost-functionality curve [15]). Therefore, communities considering building on more complex models, such as ABC or the CIDOC/CRM [27], should carefully consider the costs and benefits. In many cases, it may be more expedient to supply high-volume pidgin metadata (e.g., simple Dublin Core) rather than constructing highly expressive but expensive descriptions for fewer objects. These are decisions that should be based on careful analysis of desired functionality and economic realities.

The modeling principles upon which ABC builds are described in the original strawman document [21]. The notion of temporality deserves emphasis here due to its importance in the model and its implications for its scope. A core intent in ABC is the ability to model the creation, evolution, and transition of objects over time. Traditional bibliographic cataloging has generally assumed that the objects being described, and therefore their attributes, are more or less stable. Time and object transition has generally been relegated to "second class" status. This has made traditional resource-based cataloging inadequate in a number of contexts [28], for example:

- *museums*, where describing the temporal transitions of an object (e.g., its discovery, classification, exhibit history) is considered essential.
- *archives*, where provenance of an object is fundamental to establishing its integrity.
- *digital resources*, which unlike physical content are fundamentally malleable and derivable.
- *rights management*, where questions about "who did what, when, where, and of what nature?" are essential to assigning proper attribution.

2.2. Targeted Objects

The ability to model change makes ABC appropriate for describing a wide variety of entities and the relationships between them. In particular, it has been designed to model physical, digital and analogue objects held in libraries, archives, and museums and on the Internet. This includes objects of all media types – text, image, video, audio, web pages, and multimedia. It can also be used to model abstract concepts such as intellectual content and temporal entities such as performances or lifecycle events that happen to an object. In addition the model can be used to describe other fundamental entities that occur across many domains such as: agents (people, organizations, instruments), places and times.

2.3. Intended Use and Users

The ABC model has not been constrained by the design principle that it be comprehensible by the standard "user", or creator of metadata. Rather, it is intended as a conceptual foundation with two communities of use in mind:

Individual metadata communities might use the principles demonstrated in ABC as the basis for building domain or purpose specific metadata ontologies and models. While these ontologies might not include all aspects of ABC, our experience has shown that awareness of the principles in ABC, especially the clean separation of entities and the conceptualization of object transition, can prove valuable in avoiding common pitfalls of metadata design. ABC has been deliberately designed as a primitive ontology so that individual communities are able to build on top of it. A set of base classes has been provided to act as either attachment points for domain-specific properties or super classes which can be sub-classed to create domain-specific classes.

System builders might use the ABC principles as the basis for implementing tools that permit mapping across descriptions in multiple metadata formats. Our experience has shown that the possibility of mapping automatically is often mitigated by the undisciplined use of existing metadata formats, or by the non-regular semantics of many metadata ontologies. However, it is arguably true that a foundation model such as ABC may provide a knowledge framework to assist in metadata mapping. Our own experiments with ABC [30] have demonstrated this mapping.

3 Narrative Overview of the ABC Model

This section introduces the elements of the ABC model. It is intended as a complement and entry point to Section 4, which includes a detailed specification of all ABC classes and Properties, Appendix A, which presents the ABC model as an RDF schema, and Appendix B, which illustrate the ABC class hierarchy in graphical form.

The primitive category at the core of the ABC ontology is an *entity*. Three categories lie at the next level of the ontology: *temporality, actuality*, and *abstraction.*

3.1. *Temporality* Category

A distinguished aspect of the ABC model is the manner in which it explicitly models time and the way in which properties of objects are transformed over time. Other descriptive models such as AACR2 [23] and Dublin Core [6] imply some time semantics. For example, the DC *date* element and its qualifiers [8] *created* and *modified* express events in the lifecycle of a resource. Note, that expressing these events in this second-class manner (i.e., not making the temporal entities ontological entities) makes it difficult to associate agent responsibility with those events and connect them with changes in state of the resource. In contrast, the ABC model makes it possible to unambiguously express situations in which object properties exist, the transitions that demark those situations, and the actions and agency that participate in those transitions.

The theoretical foundations for the ABC temporal notions can be found in process models such as Petri Nets [35] or extensions to first-order logic such as Situational Calculus [32]. In brief summary, ABC models time as follows:

- A *situation* provides the context for framing time-dependent properties of (possibly multiple) entities. Entities, such as a person or a document, may have properties that exist only in the context of a situation and other properties that are constant across the context of a description. For example, in a description of the "my car", the property "has make Honda" is constant across the entire description, but the property "has color red" applies before I paint it, and the property "has color green" applies after I paint it. Concurrently, the green paint can has the property "is full" before I paint the car and the property "is empty" after I paint it, but always has the property "produced by Dulux". ABC models this through the use of situations to which are bound *existential* facets of entities, which provide the attachment points for situation-specific properties of entities (the color of the car and the fullness of the paint can). These existential facets can co-exist with a single *universal* facet of each entity, to which the time-independent properties are bound (e.g., the model of the car or the producer of the paint can). From the perspective of first-order logic, the *existential* facet corresponds to "there exists a situation in which an instance of the entity exists with a property set", and the *universal* facet corresponds to "for all situations in the description the entity exists with certain property set". More details on entities and their contextual and non-contextual properties are given below in Section 3.2.
- An *event* marks a transition from one situation to another. Events always have time properties. The effect is that a *situation* implicitly has time duration as defined by its bounding events (associated via *precedes* and *follows* properties). As an example, the model could express the loan of the *Mona Lisa* to the Metropolitan Museum for a fixed period (e.g., May 1, 2000 through May 30, 2001) as follows: an existential facet of the *Mona Lisa* with a property "located at the Metropolitan" could be associated with a situation that is related via *precedes* and *follows* properties with two events, one of which gives the time of the loan, the other the time of the return. The use of the *hasPresence* property with an *Event* models the fuzzy concept of the participation of an *Agent* in the *Event* – more precise notions of participation require the *Action* concept as described below.
- An A*ction* provides the mechanism for modeling increased knowledge about the involvement and responsibility of agents in events. Specifically, it denotes a verb in the context of event – the *hasAction* property connects an *Action* to an *Event*. Returning to the example of painting the car, one might model this using one *Event* (the painting) and two *Situations* to provide the context for the existential facets of the paint can and car as described above. One might also want to express the fact that John held the paint can and Sue did the actual painting. *Actions* provide the ontological framework for expressing these "verbs" and associating the specific agency with them. In addition, the *involves* property (and its sub-properties) makes

it possible to explicitly associate actions with effects on existential facets of entities. Finally, the *hasParticipant* property, and its possible domain-specific sub-properties, makes it possible to precisely specify the association of an *Agent* with an *Action.* The combination of these makes it possible to clearly state entity derivations (e.g., translations, reformatting, etc.) and modifications and who or what is responsible for them.

As shown in the examples in Section 5, these three temporal classes make it possible to unambiguously express statements like: *"In 1998 Quentin Blake acted as Illustrator in the event that led to a situation where a soft cover edition of 'Charlie and the Chocolate Factory' existed".*

3.2. Actuality Category

The *Actuality* ontology category encompasses entities that are sensible – they can be heard, seen, smelled, or touched. This contrasts with the *Abstraction* category, which encompasses concepts.

As described in Section 3.1 entities that are Actualities, can have one *universal* or time-independent facet and many *existential* or time-dependent facets. ABC expresses this notion through the *inContext* property that associates an *Actuality* with a *Situation*. For example, an ABC description of Bill Clinton might have an *existential Actuality* with property "President of the United States" that is related via the *phaseOf* property to one universal *Actuality* with the property "born in Arkansas". The existential facet would be related via the *inContext* property to a *Situation* that *follows* an *Event* representing Clinton's election in 1992. The result is a statement that expresses the "sameness" of the two entities (they are both "Bill Clinton"), but the fact that one is an existential facet and one is a universal facet.

The ABC model also incorporates intellectual creation semantics influenced by the IFLA FRBR [9]. A sub-category of Actuality, *Artifact*, expresses sensible entities that are tangible realizations of concepts, and that can be manifested in multiple ways; *e.g.*, as *Manifestations* and *Items* as expressed in the FRBR.

3.3. Abstraction Category

The *Abstraction* category makes it possible to express concepts or ideas. Entities in this category have two notable characteristics:

1. They are never in the context of a situation. While it can be argued that an idea is "born" at some time, ABC treats the "birth of an idea" when it is manifested in some sensible way; e.g., when it is told, demonstrated, or shown in some manner.
2. Correspondingly, ideas cannot exist in isolation in the model. They must be bound to some Actuality through the *hasRealization* property.

The main use of the Abstraction category is to express the notion of *Work* in the FRBR sense; that is, as a means of binding together several *Manifestations* of an intellectual *Expression*. For example, an ABC description of the Hamlet might instantiate a *Work* that binds the folio manifestation, a Stratford performance, and a Penguin edition.

4 ABC Classes and Properties

This section describes the elements of the ABC ontology, expressed using RDF primitives [20, 31]. The section is divided into a list of the basic classes of the model and the properties that relate instances of those classes. The notions of *SubClass, SubProperty, Domain*, and *Range* are used in the manner of RDF. An RDF schema representation of this class and property structure is available in Appendix A.

4.1. Classes

The figure in Appendix B shows the hierarchical relationships between the ABC classes, which are described below. Classes are shown in rectangles and sub-class relationships are indicated by solid lines.

Name *Entity*
Subclass of none
Description
The primitive category having no differentiae.

Name *Temporality*
Subclass of *Entity*
Description
A primitive ontology category for sub-classing categories of entities that provide time existential contexts.

Name *Actuality*
Subclass of *Entity*
Description
A primitive ontology category for sub-classing categories of entities that have a tangible existence in some world view. *Actualities* as identities and properties associated with those properties have a duality as universal - entities whose identity as a set of properties/characteristics are time independent, or *universal*, relative to the world view of a model - and existential - entities whose identity as a set of properties/characteristics are local, or *existential*, to situations/contexts in a model. The *phaseOf* and *inContext* properties are means of co-relating an existential facet of an *Actuality* and its universal facet and of associating the existential facet with a specific situation.

Name *Abstraction*
Subclass of *Entity*
Description

A primitive ontology category for sub-classing categories of entities that are pure information or concepts (stands in contrast to the *Actuality* category).

Name *Artifact*
Subclass of *Actuality*
Description

A type of *Actuality* that is the tangible realization of some consciously conceived *Abstraction* - a prototypical example is intellectual content. The primary distinguishing characteristics of *Artifacts* is that they can be manifested in a number of ways and copied - for example the book "Hamlet" is an *Artifact* (of the *Abstraction* Hamlet) since it is one of many possible *Actualities*. This contrasts to the *Actuality* William Shakespeare who admittedly may have been pre-conceived by his parents but can not be manifested in various ways. Similarly a historical museum object such as a dinosaur bone is an *Actuality* but not an *Artifact*.

Name *Event*
Subclass of *Temporality*
Description

An *Event* marks a transition between *Situation*s, one that is associated with the event through a *precedes* property and another through a *follows* property. The time granularity of the transition is variable - for example some *Events* are truly an instant (a point in time). However, an *Event* may have coarser granularity such as span of time during which some situation change was undertaken (for example, the painting of the Sistine Chapel Ceiling). The granularity of the snapshot is associated with the *Event* via a single *atTime* property.

Name *Situation*
Subclass of *Temporality*
Description

A *Situation* is a context for making time-dependent or *existential* assertions about *Actualities*. Each *Situation* can act as context for existential facets of multiple *Actualities.* The time certainty for a *Situation* is implicitly within the time contexts for the events that enclose it (i.e. a *Situation* can serve as the *precedes* of one *Event* and the *follows* of another). However, the time certainty can be explicitly stated via a *atTime* property on the *Situation*. The purpose of this is to make the model as closed as possible - where the time certainty of *Events* and *Situations* is known.

Name *Action*
Subclass of *Temporality*
Description

An activity or verb performed by some *Agent* or *Agents* in the context of an *Event*. *Actions* may involve an *Actuality*, which may be in its existential or universal facet, and may have result that is another *Actuality*, which always must be in its existential facet.

Name *Agent*
Subclass of *Actuality*
Description

An *Actuality* that is present during an *Event* or is the party of some *Action*. *Agents* may be persons, instruments, organizations, etc.

Name *Work*
Subclass of *Abstraction*
Description

An *Abstraction* that is intellectual property in the IFLA FRBR sense. A *Work* is an abstract concept which can not exist in a model in isolation, but is only revealed when it has been actualized in some *Manifestation*.

Name *Manifestation*
Subclass of *Artifact*
Description

A form of an *Artifact* that stands as the sensible realization of a *Work*. *Works* and *Manifestations* stand in a one to many relationship. The *hasRealization* property associates a *Work* with its *Manifestation*(s). Associating several *Manifestations* with a *Work* through the *hasRealization* property defines those *Manifestations* as members of a (fuzzy) equivalence class implicitly identified by the common Work.

Name *Item*
Subclass of *Artifact*
Description

A form of an *Artifact* used to establish a set of identical copies. *Manifestations* and *Items* stand in a one to many relationship. The *hasCopy* property associates a *Manifestation* with its *Items*. Associating several *Items* with a *Manifestation* through the *hasCopy* property defines those *Items* as members of an exact equivalence class.

Name *Time*
Subclass of *Entity*
Description

An entity which represents either a time span or point in time and which can be used to confine the temporal extent of *Temporalities* (*Events* or *Situations)*. The *Time* entity provides the range constraint for the *atTime* property.

Name *Place*
Subclass of *Entity*
Description
An entity which represents spatial location. It can be used to specify the location of either *Temporalities* (*Events* and *Situations*) or *Actualities*. The *Place* entity provides the range constraint for the *inPlace* property.

4.2. Properties

The figure in Appendix B illustrates the relationship of the ABC properties to the ABC classes. Classes are shown in rectangles and solid lines indicate sub-class relationships. Properties are down as dashed lines directed from their domain class(es) to their range class. A property that does not have a defined range is indicated by an oval at the end of a dashed arc. Finally, property/sub-property relationships are indicated by dotted arcs with an oval of the sub-property at the end of the arc. The definitions of the ABC properties illustrated in Appendix C are provided below.

Name *precedes*
Subproperty of none
Domain *Event*
Range *Situation*
Description
Binds a *Situation* and the *Actualities* within its context as existing before an *Event*.

Name *follows*
Subproperty of none
Domain *Event*
Range *Situation*
Description
Binds a *Situation* and the *Actualities* within its context as existing after an *Event*. There is no explicit implication of causality between the *Event* and *Actualities* existing in the *Situation* that is the value of the *follows* property. Causality between *Events* or *Actions* in *Events* and *Actualities* is established through the *hasResult* property and its sub-properties.

Name *isPartOf*
Subproperty of none
Domain *Entity*
Range *Entity*
Description
Establishes an "is Part Of" relationship between one *Entity* and another (it is the inverse of the *contains* Relationship).

Name *contains*
Subproperty of none
Domain *Entity*
Range *Entity*
Description
Establishes a "contains" relationship between one Entity and another (inverse of *isPartOf*). The inverse properties are explicitly expressed due to the need to model the notion of an Actuality that does not makes sense without its "parts" and, equally, the notion of an Actuality that does not makes sense without its containment.

Name *isSubEventOf*
Subproperty of *isPartOf*
Domain *Event*
Range *Event*
Description
Establishes an "is Part of" relationship between one *Event* and another; e.g., the relationship between "D-Day" and "World War II". The relationship does not imply semantic constraints such as containment of the *atTime* property of one *Event* within another or relationships between the associated *Situations*. Note the distinction between two *Events* related by the *isSubEventOf* property and an *Event* and an *Action* related by the *hasAction* property. In the former case, each *Event* acts as transition points between different preceding and following *Situations.* In the latter case, there is a single *Event* with one preceding and following *Situation*, but individual verbs or *Actions* within that *Event.*

Name *inContext*
Subproperty of none
Domain *Actuality*
Range *Situation*
Description
Establishes an *Actuality* as an existential, which means that its property set exists within the context of *Situation* that is associated as the value of this property.

Name *phaseOf*
Subproperty of none
Domain *Actuality*
Range *Acuality*
Description
Establishes the relationship between an existential facet and a universal facet of an *Actuality*.

Name *hasRealization*
Subproperty of none
Domain *Work*
Range *Manifestation*
Description
Binds a *Manifestation* within the conceptual umbrella of a *Work*. A *Work* may have several *hasRealization* properties, which establishes a fuzzy equivalence set among *Manifestations,* implicitly stating that the properties of the subject

Work are shared across the object *Manifestation*(s).

Name *hasCopy*
Subproperty of none
Domain *Manifestation*
Range *Item*
Description
Binds an *Item* as one of several copies of a *Manifestation*. A *Manifestation* may have several *hasCopy* properties, which establishes an exact equivalence set among *Manifestations*, implicitly stating that the properties of the subject *Manifestation* are shared across the object *Item*(s).

Name *involves*
Subproperty of none
Domain *Action*, *Event*
Range *Actuality*
Description
Expresses the involvement of an *Actuality* in the performance of an *Action* or an *Event*. There is no implication of transformation or lack thereof in this use. (Such specialization of *involves* is expressed using the *hasPatient* and *usesTool* properties.)

Name *hasPatient*
Subproperty of *involves*
Domain *Action* ,*Event*
Range *Actuality*
Description
Strengthens the notion of *involves* to the classic patient sense stating that the *Actuality* that is the value of this property is transformed by the *Action* or *Event*. For example, the action expressing the rebinding of a book might have the book related via a *hasPatient* property.

Name *usesTool*
Subproperty of *involves*
Domain *Action*, *Event*
Range *Actuality*
Description
A specialization of involves that in effect weakens the notion of involvement of the *Actuality* in the *Action* or *Event* - e.g., it is used but not transformed as in the case of a camera in the production of a picture.

Name *hasResult*
Subproperty of none
Domain *Action*, *Event*
Range *Actuality*
Description
Expresses the result of an *Actuality*, which always must be in an existential facet, in the performance of an *Action* (in the context of an *Event*).

Name *destroys*
Subproperty of *hasPatient*
Domain *Action*, *Event*
Range *Actuality*
Description
A specialization of *hasPatient* that indicates that the value *Actuality* ceases to exist in *Situation*(s) that follow the *Event*. Any *Actuality* that is not explicitly destroyed can be assumed to exist in subsequent *Situations* even though it might not be explicitly represented.

Name *creates*
Subproperty of *hasResult*
Domain *Action*, *Event*
Range *Actuality*
Description
Specializes *hasResult* to mean the coming into existence of the *Actuality* that is the value of this property. This means that the *Actuality* can be assumed to not exist in *Situations* prior to the one in which the created instance of the *Actuality* appears.

Name *hasAction*
Subproperty of none
Domain *Event*
Range *Action*
Description
An *Event* can have one or more *Actions*, which are verbs performed by *Agents* in the context of the *Event*.

Name *hasPresence*
Subproperty of none
Domain *Event*
Range *Agent*
Description
Associates an *Agent* as being present in the context of an *Event*. The notion of "presence" is purposely weak – there is no implication that the target Agent is an active participant in the transition marked by the *Event*.

Name *hasParticipant*
Subproperty of *hasPresence*
Domain *Event, Action*
Range *Agent*
Description
Refines *hasPresence* to associate an *Agent* as an active participant in an *Event* or *Action*. Combined with the *hasPatient* and *hasResult* properties attached to an *Action*, this permits definite statements of subject and causality – e.g., "John did the painting that turned the car from red to green". The inverse property is *participatesIn*.

Name *atTime*
Subproperty of none
Domain *Temporality*
Range *Time*
Description

Associates a time with an entity that is a sub-category of a *Temporality*. The range for this property is the *Time* entity.

Name *inPlace*
Subproperty of none
Domain *Actuality*, *Temporality*
Range *Place*
Description

Associates a location with an entity. The entity can be an *Actuality* or *Temporality*. The range for this property is the *Place* entity.

5 ABC Modeling Experiments and Examples

As a result of a collaboration between the Harmony project and the CIMI Consortium [5], a Call for Participation (CfP) [34] was issued to CIMI members in October 2000. Interested CIMI members were invited to contribute approximately 100 museum metadata records and the associated multimedia digital objects. The goal of the experiment was to evaluate the ABC model and its usability as means of mapping among disparate metadata ontologies. Four organizations responded to the CfP:

1. Australian Museums Online (AMOL);
2. Natural History Museum of London;
3. Research Libraries Group/Library of Congress;
4. National Museum of Denmark.

A detailed description of the images and data provided by the CIMI members is available at [11]. Detailed results of the experiment are available at [4].

This section summarizes some of those results. The complexity of many of the CIMI examples makes them impractical as introductory examples of the application of the ABC model. Instead, we have chosen to illustrate three fictional, but realistic, examples in Sections 5.1, 5.2, and 5.3 and then include one of the simpler CIMI examples in Section 5.4. All of the examples are illustrated as RDF-like node and arc diagrams. An XML serialization of these graphs would also be possible, but space limitations prevent including these.

5.1. Children's Book

Example Narrative: The book, "Charlie and the Chocolate Factory" was written by Roald Dahl in 1964. The first edition (a hardcover, illustrated by Joseph Shindleman) was published in 1985 by Knopf. A second edition was published in 1998 by Puffin. It was a paperback illustrated by Quentin Blake. In 1995, a 3 hour audiocassette recording of the book was produced by Caedmon. It was narrated by Robert Powell and the caterer during production was "Sam 'n Ella's Catering".

The graphical representation corresponding to the ABC model for this example is shown in Appendix C.

5.2. Dinosaur Bone

Example Narrative: A dinosaur bone was discovered by Richard Leakey in 1995 in Kenya. In 1971 it was acquired by the British Museum in London and added to its collection. In 1991, Jean Smith, the curator of the British Museum, classified the bone as part of a plesiosaur. In 1998, Richard Hill photographed the bone using a digital camera (a Nikon 990). In 1999, this image of the dinosaur bone was published on the museum's web site.

The graphical representation corresponding to the ABC model for this example is shown in Appendix D.

5.3. Birth

Example Narrative: On June 14 2001 at the Wesley Hospital, an 8lb 11oz baby girl was delivered to parents Jill and John Smith. The obstetrician at the delivery was Jane Jekyl and the midwife was Carl Hyde.

The graphical representation corresponding to the ABC model for this example is shown in Appendix E. This example demonstrates how the *Action type* refines or more narrowly specifies the actions which occur within an *Event*. The *Agent's ParticipationType* further refines or more narrowly specifies the role of the specific agents within an action.

5.4. AMOL Vase Example

This final example is based on a metadata record from the Powerhouse Museum in Sydney, describing the physical characteristics, life history and digital surrogates of a vase in the collection. The actual metadata record is available at [2]. This record demonstrates the difficulty of automated mappings from existing metadata descriptions. As shown, the value for each metadata tag is a natural language paragraph, in which is embedded complex information on the lifecycle events of the vase. Automated processing of this record would require natural language processing techniques.

The graphical representation corresponding to the ABC model for this record is shown in Appendix F.

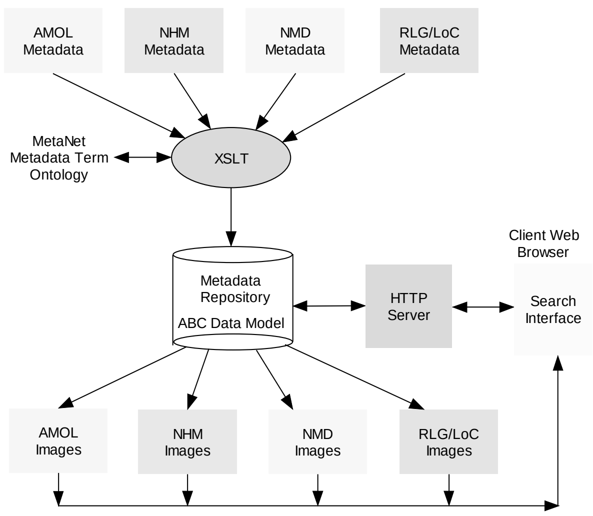
6 Searching over the model

The ABC model allows users to ask much more sophisticated queries than is possible via less expressive metadata models such as Dublin Core e.g., "Tell me all of the previous owners of an object", "Give me all of those objects which were acquired as gifts and the donor's name and address". By using the ABC model, one is able to record and retrieve the history of an object from its creation, through to its use, change of ownership, relocation, modification, digitization and repurposing.

The ABC model as described heretofore provides an abstract, syntax-neutral conceptual framework for modeling metadata. However in order to create, store and query the metadata descriptions, a concrete syntax is required. RDF provides one possible XML syntax for encoding and exchanging metadata descriptions. Although alternate XML encodings of the data models are possible, we have chosen to encode the graphical ABC models of the CIMI data at [4] in RDF/XML syntax to maximize interoperability, since RDF provides a well-defined mechanism for encoding in XML instances of ontologies. An RDF/XML representation of the AMOL example in 5.4 is provided at [1].

Although fully-automated mapping of the existing CIMI metadata records to the ABC model is an unrealistic goal, we are currently working on organization-specific XSLT programs capable of mapping each set of CIMI records into RDF descriptions based on the ABC model. Using XSLT, combined with the semantic knowledge provided by MetaNet, a metadata term ontology [24], it is possible to streamline the generation of the ABC metadata descriptions; making use of automatic facilities where possible and augmenting them with some human effort.

Given the resulting collection of RDF/XML descriptions, we then use the Squish RDF query engine developed at ILRT as part of the Harmony project, to query the RDF files directly. Squish is a generalizable Java RDF query engine [33] which can run SQL-like query strings over many instances of



RDF descriptions. For example, the Squish query below requests all of the events (and their type, time and place) that occur in the previous AMOL example [1]:

```
SELECT ?event, ?type, ?time, ?place FROM
        http://ilrt.org/discovery/harmony/amol.rdf
        WHERE
        (web::type ?event abc::Event)
        (abc::context ?event ?context)
        (dc::type ?event ?type)
        (abc::time ?context ?time)
        (abc::place ?context ?place)
        USING web FOR
http://www.w3.org/1999/02/22-rdf-syntax-ns#
        abc FOR http://ilrt.org/discovery/harmony/abc-
0.1#
        dc for http://purl.org/dc/elements/1.1/
```

A Squish web search interface demo to the AMOL images is available at: [13]. Once the RDF descriptions for all of the image/data sets have been generated, then this search interface will be extended to search across all of the CIMI images, as illustrated in Figure 1.

7 Future work and Conclusions

Because the ABC model has been specifically designed to model the creation, evolution and transition of objects over time, we are particularly interested in investigating its application to multimedia asset management metadata within organizational workflows. Hence, a future goal is to design a workflow management system that automatically invokes the appropriate metadata editing and generation tools as objects proceed through an organization's workflow, from creation or acquisition, to editing, reuse, copying, resale and preservation. Such a tool would realize some of the record-keeping goals articulated by Bearman and Trant in [18]. In theory, the ABC model should provide the ideal underlying schema for modeling, validating, storing, navigating and searching the different types of metadata generated from the sequence of event-triggered metadata input tools.

Finally, in recognition of the extensive overlap of the goals of the Harmony project and the CIDOC/CRM, a DELOS Working Group on Ontology Harmonization has been established. The first workshop was held in Rome in May, 2001 [22]. A second workshop is planned for September, 2001 in Darmstadt. The objective of this working group is to investigate merging the concepts of the ABC model and the CIDOC CRM into a single ontology and in the process, to determine:

- methodologies for comparing, merging and sharing ontologies;
- representational alternatives for ontologies;
- the optimum approach to the management of sharable or merged ontologies and the future merging of additional ontologies.

In closing, our work on developing the ABC model has been extremely useful in elevating our understanding of the metadata landscape and in comprehending what people are trying to accomplish with their resource descriptions. For instance, work with Dublin Core records from the CIMI community demonstrates a desire to represent relatively complex lifecycle information for which the simple Dublin Core model is inadequate. As mentioned earlier, the appropriateness of any metadata model must be measured by balancing the specificity of the knowledge that can be represented in it and queried from it and the expense of creating the descriptions. Our experiments with ABC demonstrate the usefulness of metadata models with temporal semantics for the class of descriptions where that level of knowledge representation is deemed appropriate.

8 Acknowledgements

The work described in this paper has been carried out within the Harmony International Digital Library Project. It has been funded by NSF Grant 9905955, JISC Grant 9906 and the Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme. The authors wish to acknowledge the valuable contributions made to this work by Dan Brickley and Libby Miller from ILRT.

9 References

[[1] *AMOL RDF Record*, 2001 http://metadata.net/harmony/amol.rdf.

[[2] *AMOL Vase Description*, 2001 http://www.metadata.net/harmony/amol.html.

[[3] *CIDOC Conceptual Reference Model Special Interest Group*, http://cidoc.ics.forth.gr/.

[[4] *CIMI ABC Modeling Examples*, 2001 http://metadata.net/harmony/cimi_modelling.htm.

[[5] *CIMI Consortium*, 2001 http://www.cimi.org.

[[6] *Dublin Core Metadata Element Set, Version 1.1*, 1999 http://www.dublincore.org/documents/dces/

[[7] *Dublin Core Metadata Initiative*, http://dublincore.org.

[[8] *Dublin Core Qualifiers*, 2000 http://purl.org/DC/documents/rec/dcmes-qualifiers-20000711.htm.

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

[[10] *The Harmony Project*, http://metadata.net/harmony.

[[11] *Images and Metadata Records Provided by CIMI Members for Harmony/CIDOC CRM Museum Interoperability Research*, 2001 http://archive.dstc.edu.au/CIMI/.

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

[[13] *RDF: Image Metadata*, 2001 http://swordfish.rdfweb.org/discovery/2001/03/demos/image.html.

[[14] *Semantic Web Activity: Resourse Description Framework (RDF)*, 2001 http://www.w3.org/RDF.

[[15] W. Y. Arms, *Digital libraries*. Cambridge, Ma.: MIT Press, 2000, p. 208.

[[16] T. Baker, "A Grammar of Dublin Core," *D-Lib Magazine*, 6 (10), 2000, http://www.dlib.org/dlib/october00/baker/10baker.html.

[[17] 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.

[[18] D. Bearman and J. Trant, " Electronic Records Research Working Meeting May 28-30, 1997, A Report from the Archives Community," *D-Lib Magazine*, 1997, http://www.dlib.org/dlib/july97/07bearman.html.

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

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

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

[[22] M. Doerr and J. Hunter, *DELOS Working Group on Ontology Harmonization: Full report from the first meeting at CNR Rome*, 2001 http://metadata.net/harmony/Rome_Full_Rep_V2.doc.

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

[[24] J. Hunter, "MetaNet - A Metadata Term Thesaurus to Enable Semantic Interoperability Between Metadata Domans," *Journal of Digital Information*, 1 (8), 2001,

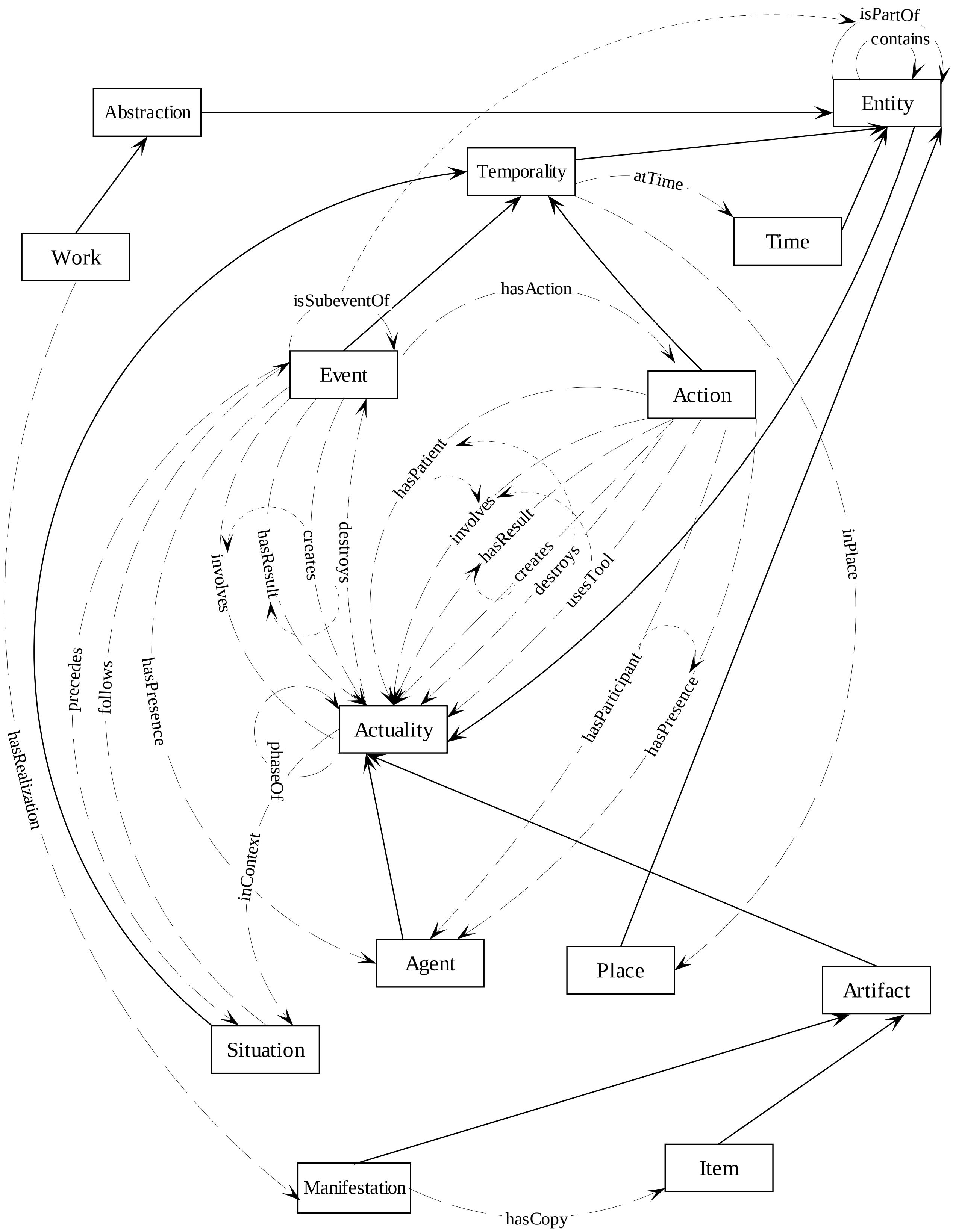
http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Hunter/.
[[25] J. Hunter and D. James, “Application of an Event-Aware Metadata Model to an Online Oral History Archive,” presented at ECDL 2000, Lisbon, 2000, http://archive.dstc.edu.au/RDU/staff/jane-hunter/OralHistory/paper.html.
[[26] J. Hunter and C. Lagoze, “Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles,” presented at WWW10, Hong Kong, 2001, http://www.cs.cornell.edu/lagoze/papers/HunterLagozeWWW10.pdf.
[[27] ICOM/CIDOC Documentation Standards Group, *CIDOC Conceptual Reference Model*, 1998 http://www.ville-ge.ch/musinfo/cidoc/oomodel/.
[[28] C. Lagoze, “Business Unusual; How "event awareness" may breathe life into the catalog,” presented at Bicentennial Conference on Bibliographic Control in the New Millennium, Library of Congress, Washington DC, 2000, http://lcweb.loc.gov/catdir/bibcontrol/lagoze_paper.html.
[[29] C. Lagoze, “Keeping Dublin Core Simple: Cross Domain Discovery or Resource Description?,” *D-Lib Magazine*, 7 (1), 2001, http://www.dlib.org/dlib/january01/lagoze/01lagoze.html.
[[30] C. Lagoze, J. Hunter, and D. Brickley, “An Event-Aware Model for Metadata Interoperability,” presented at ECDL 2000, Lisbon, 2000, http://archive.dstc.edu.au/RDU/staff/jane-hunter/harmony/harmony_ECDL2000.zip.
[[31] 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, January 1999. http://www.w3.org/TR/PR-rdf-syntax/.
[[32] J. McCarthy, “Programs with common sense,” in *Semantic Information Procession*, M. Minsky, Ed.: MIT Press, 1968, pp. 403-418.
[[33] L. Miller, *RDF querying using Squish*, 2000 http://swordfish.rdfweb.org/rdfquery/.
[[34] J. Perkins, *ABC/Harmony CIMI Collaboration Project Description*, 2000 http://www.cimi.org/public_docs/Harmony_long_desc.html.
[[35] J. L. Peterson, *Petri net theory and the modeling of systems*. Englewood Cliffs N.J.: Prentice-Hall, 1981.
[[36] J. F. Sowa, *Conceptual Graphs*, 2001 http://www.bestweb.net/~sowa/cg/.

Appendix A ABC Model Expressed as an RDF Schema

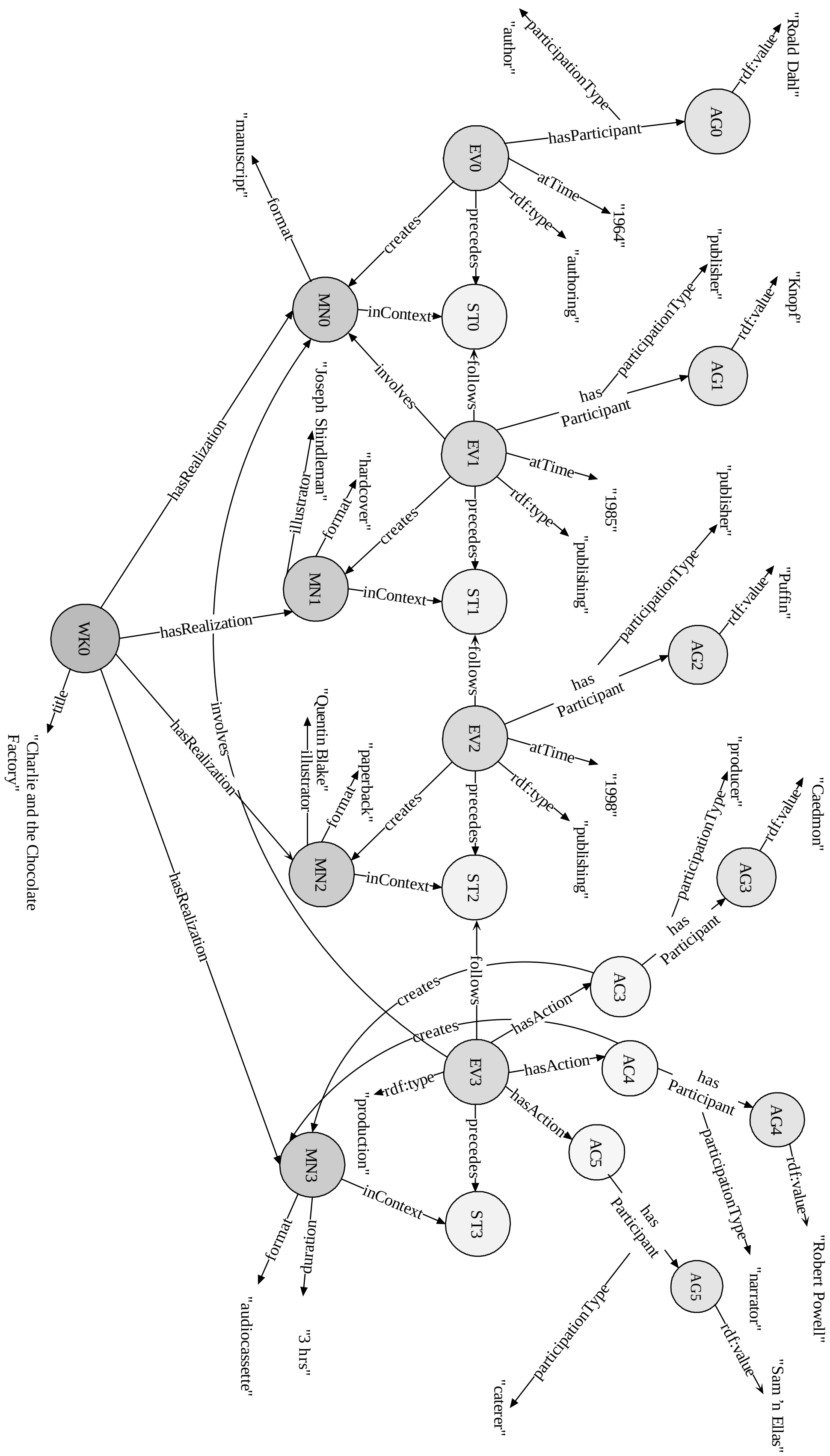
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0 U (http://www.xmlspy.com) by Carl Lagoze (Cornell University) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
	<rdfs:Class rdf:ID="Entity"/>
	<rdfs:Class rdf:ID="Temporality">
		<rdfs:subClassOf rdf:resource="Entity"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Actuality">
		<rdfs:subClassOf rdf:resource="Entity"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Abstraction">
		<rdfs:subClassOf rdf:resource="Entity"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Time">
		<rdfs:subClassOf rdf:resource="Entity"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Place">
		<rdfs:subClassOf rdf:resource="Entity"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Artifact">
		<rdfs:subClassOf rdf:resource="Actuality"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Event">
		<rdfs:subClassOf rdf:resource="Temporality"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Situation">
		<rdfs:subClassOf rdf:resource="Temporality"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Action">
		<rdfs:subClassOf rdf:resource="Temporality"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Agent">
		<rdfs:subClassOf rdf:resource="Actuality"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Work">
		<rdfs:subClassOf rdf:resource="Abstraction"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Manifestation">
		<rdfs:subClassOf rdf:resource="Artifact"/>
	</rdfs:Class>
	<rdfs:Class rdf:ID="Item">
		<rdfs:subClassOf rdf:resource="Artifact"/>
	</rdfs:Class>
	<rdf:Property rdf:ID="precedes">
		<rdfs:domain rdf:resource="Event"/>
		<rdfs:range rdf:resource="Situation"/>
	</rdf:Property>
	<rdf:Property rdf:ID="follows">
		<rdfs:domain rdf:resource="Event"/>
		<rdfs:range rdf:resource="Situation"/>
	</rdf:Property>
	<rdf:Property rdf:ID="isPartOf">
		<rdfs:domain rdf:resource="Entity"/>
		<rdfs:range rdf:resource="Entity"/>
	</rdf:Property>
	<rdf:Property rdf:ID="contains">
		<rdfs:domain rdf:resource="Entity"/>
		<rdfs:range rdf:resource="Entity"/>
	</rdf:Property>
	<rdf:Property rdf:ID="isSubEventOf">
		<rdfs:domain rdf:resource="Event"/>
		<rdfs:range rdf:resource="Event"/>
		<rdfs:subPropertyOf rdf:resource="isPartOf"/>
	</rdf:Property>
	<rdf:Property rdf:ID="inContext">
		<rdfs:domain rdf:resource="Actuality"/>
```

```
        <rdfs:range rdf:resource="Situation"/>
    </rdf:Property>
    <rdf:Property rdf:ID="phaseOf">
        <rdfs:domain rdf:resource="Actuality"/>
        <rdfs:range rdf:resource="Actuality"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasRealization">
        <rdfs:domain rdf:resource="Work"/>
        <rdfs:range rdf:resource="Manifestation"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasCopy">
        <rdfs:domain rdf:resource="Manifestation"/>
        <rdfs:range rdf:resource="Item"/>
    </rdf:Property>
    <rdf:Property rdf:ID="involves">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:range rdf:resource="Actuality"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasPatient">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:range rdf:resource="Actuality"/>
        <rdfs:subPropertyOf rdf:resource="involves"/>
    </rdf:Property>
    <rdf:Property rdf:ID="usesTool">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:range rdf:resource="Actuality"/>
        <rdfs:subPropertyOf rdf:resource="involves"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasResult">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:range rdf:resource="Actuality"/>
    </rdf:Property>
    <rdf:Property rdf:ID="destroys">
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:range rdf:resource="Actuality"/>
        <rdfs:subPropertyOf rdf:resource="hasPatient"/>
    </rdf:Property>
    <rdf:Property rdf:ID="creates">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:range rdf:resource="Actuality"/>
        <rdfs:subPropertyOf rdf:resource="hasResult"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasAction">
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:range rdf:resource="Action"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasPresence">
        <rdfs:domain rdf:resource="Event"/>
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:range rdf:resource="Agent"/>
    </rdf:Property>
    <rdf:Property rdf:ID="hasParticipant">
        <rdfs:domain rdf:resource="Action"/>
        <rdfs:range rdf:resource="Agent"/>
        <rdfs:subPropertyOf rdf:resource="hasPresence"/>
    </rdf:Property>
    <rdf:Property rdf:ID="atTime">
        <rdfs:domain rdf:resource="Temporality"/>
        <rdfs:range rdf:resource="Time"/>
    </rdf:Property>
    <rdf:Property rdf:ID="inPlace">
        <rdfs:domain rdf:resource="Actuality"/>
        <rdfs:domain rdf:resource="Temporality"/>
        <rdfs:range rdf:resource="Place"/>
    </rdf:Property>
</rdf:RDF>
```

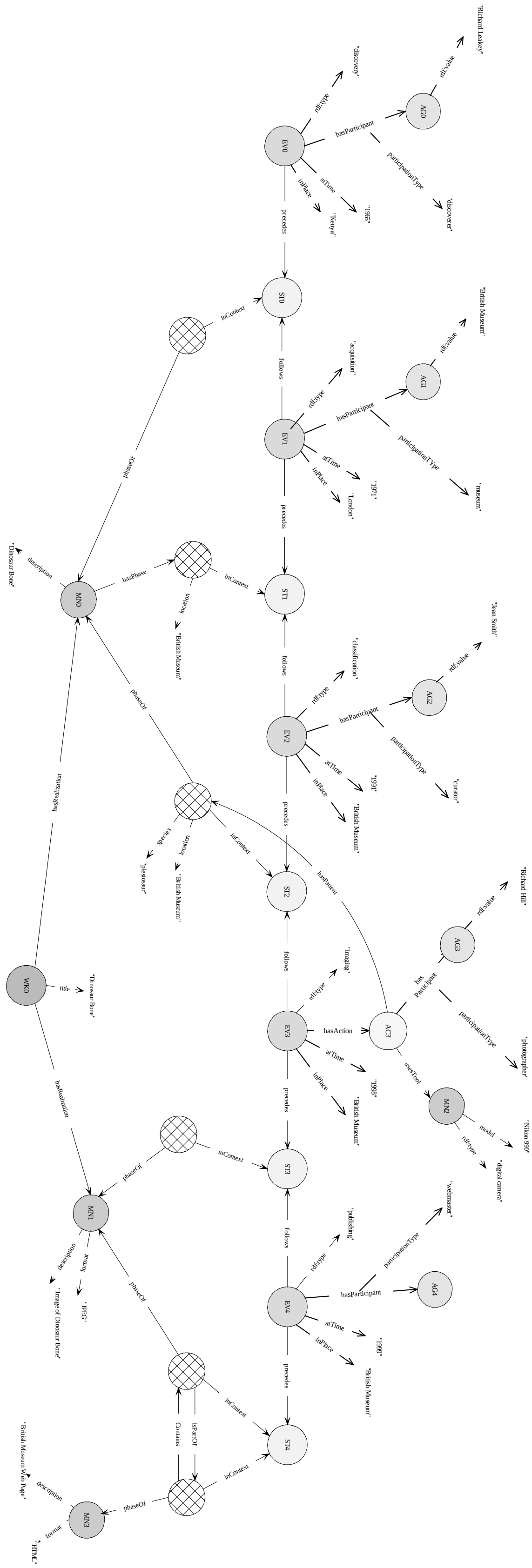
Appendix B ABC Class Hierarchy with Property Relationships



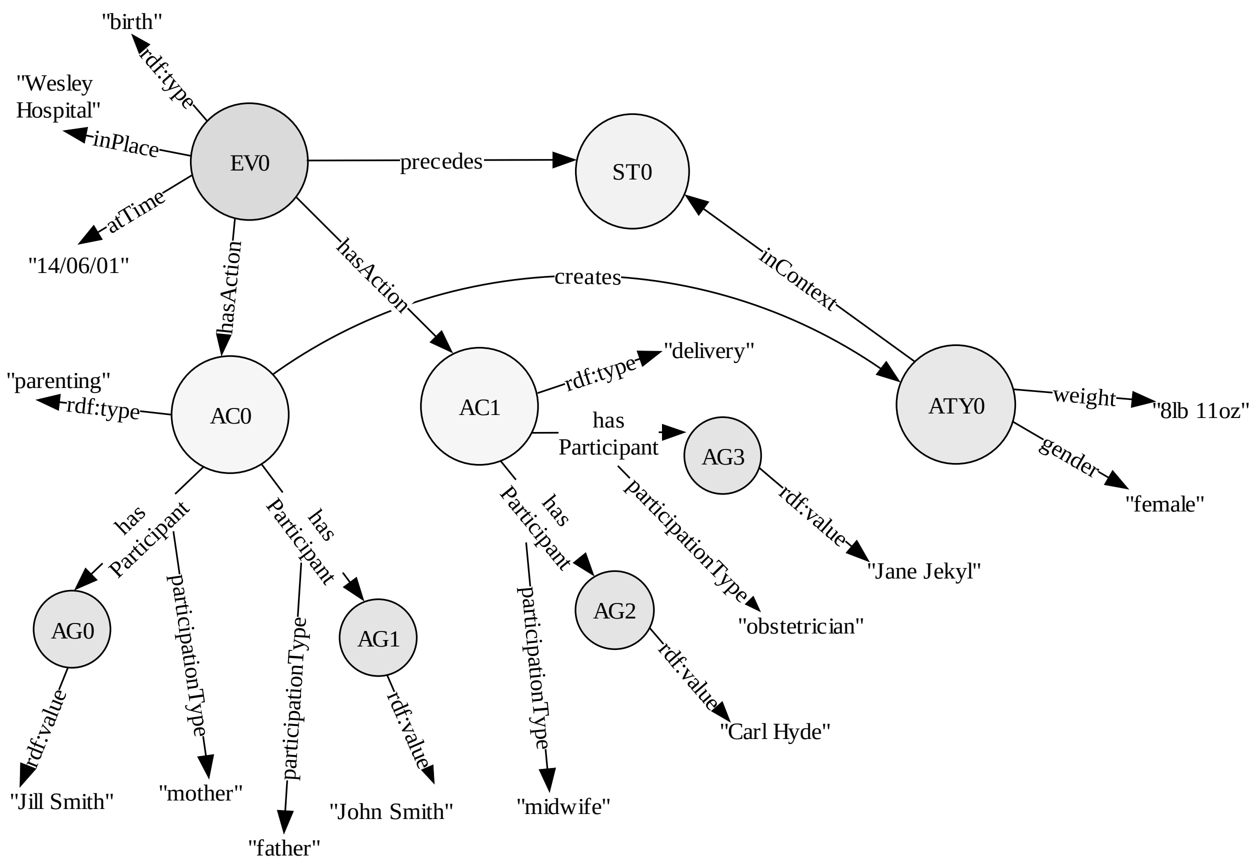
Appendix C Charlie and the Chocolate Factory Model



Appendix D Dinosaur Bone Example



Appendix E Birth Example



Appendix F AMOL Vase Example