CONCERT EVENT METADATA: DESCRIBING CONCERTS EFFECTIVELY IN A DIGITAL ENVIRONMENT

By

Michael J. Graves

A Master's paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Library Science.

Chapel Hill, North Carolina

April, 2003

Approved by:

____

Advisor

Michael J. Graves. Concert Event Metadata: Describing Concerts Effectively In a Digital Environment. A Master's paper for the M.S. in L.S. degree. April, 2003. 38 pages. Advisor: Jane Greenberg.

This research describes the creation of an ontology for web distribution of concert event information. A literature review provides a survey of current methods of encoding concert event metadata on the Web. The generation of sample records using these current standards highlights their strengths and limitations. Through an analysis of printed flyers and newspaper advertisements in the Chapel Hill area, a proposed metadata ontology is generated as an improvement over existing solutions. A prototype web database that would allow local musicians to inform the community of upcoming concerts is created as an example of how this ontology could be implemented.

Headings:

Metadata

Music

Ontologies

Resource Description Framework

Web Ontology Language

TABLE OF CONTENTS

Introduction

A reader can pick up a newspaper or magazine and find numerous advertisements for upcoming events ranging from cultural to academic to athletic and more. As society becomes more fast-paced there is less time to spend scanning print sources. The speed at which information concerning events is transmitted becomes an important factor in making more efficient use of our time. As a result, it is no surprise that the Internet has become a major avenue for informing people of upcoming events. The Web is replete with instances of schedules, calendars, show times and lists of upcoming concerts and athletic events. This research will focus on music concert events in particular and how metadata about music concert events is transferred.

How do people access concert information though? It is worth taking a step back from the digital medium for a moment to think about the role of print in disseminating information about concerts. Newspapers are a common source for this kind of information. In addition to being widely available, newspapers act as an aggregator, collecting information from a host of different sources and presenting it in a centralized and relatively familiar format. This provides a much more efficient alternative than calling every venue in town to find out which bands are playing in the near future. The complete concert listing in a newspaper is limited by space, editorial preferences, and cost. As well, newspapers are not responsible for organizing or performing in a concert and subsequently are a secondary source for this type of information. The band, the venue and the promoter are the parties with the most vested interests in providing access to accurate and up to date information. The newspaper merely serves as a channel for that information. As a result of these limitations, newspapers are often neither comprehensive nor completely accurate as sources of concert information.

Another somewhat more ephemeral form of print information is the flyer. In the case of small, local bands these flyers are often designed, printed and posted around town by the band members themselves. In contrast to the listings in newspapers, this information frequently comes from a primary source – the band itself. While the flyers themselves do not aggregate information about other shows the way that newspapers do, the places they are posted often function in this way. The physical location of these makeshift bulletin boards could be anywhere – inside a CD store, outside a theater or even on a telephone pole. In these conditions, flyers are subject to the weather, city maintenance, or the next band that comes along to staple their flyer on top of another one. The lack of any controlling authority for these postings introduces a number of problems such as, double postings and outdated information. These collections of concert listings will almost certainly be even less comprehensive than the newspaper.

The growth and increased availability of the Internet has opened up an entirely new medium to disseminating concert information though. Not only is it cheap to post concert information on the Web, but it is also quick and often instantaneous. In most cases though, concert listings have simply been moved from print to a digital equivalent without making use of the full potential of the World Wide Web. A newspaper's website might provide some basic search options but no new information has been presented in addition to the information in the print version. Because most venues have a homepage, the Internet makes it much easier to go straight to the source for more up-to-date information than could possibly be made available in a print edition newspaper. Still, in a medium sized city this could mean going to thirty or forty venue websites and looking through their entire listings. In a large city, this task would be enormous. The question then becomes how often does one need to check these pages? Once a week? Once a day? Depending on a user's location and musical preferences, one could spend hours per week checking and rechecking websites for current concert information. The ideal solution would be to automate the task. For this task to be automated, it would require the concert listings to be in a machine-readable format. Current practice makes very little of this information prime for computer understanding or automation, unfortunately. In most cases, this information exists only in HTML format. HTML simply serves to create a graphical representation of data. While this graphical representation is easy for humans to understand, it is nearly impossible for computers. As a result, these digital representations of concert listings become available and useful only when the user actively seeks them out to extract the needed information. Simply putting this information on the Web does not provide much improvement over the traditional methods. One glaring limitation of metadata encoded in an HTML format is the inability to perform searches such as keyword or band name for example. Presenting this information in a structured format would, however, greatly increase the ease with which this information could be exchanged and searched. Unfortunately, very little research has been performed that would address the issues involved in creating and evaluating an effectual framework for exchanging such information. This research aims to provide solutions to these problems.

Literature Review

Modular Metadata

When discussing available metadata standards for digital resource information the Dublin Core is often a common place to start. Increasing international acceptance and standardization efforts make this an attractive standard to work with (Dekkers and Weibel). The Dublin Core is relatively document-centric (Weibel) and thus limited in application to events. Music concerts are fundamentally different than documents (documents in a traditional sense), though. Many of the metadata elements that are used to describe documents such as title, author, and publisher will be ineffective at describing concert events. Likewise, elements used to describe events such as ticket price or a concert's start time will not be useful for describing documents. In addition, the important pieces of information that are needed to effectively describe a concert come from a number of different and separately maintained domains. A band often has no control over how much tickets cost or where tickets are available. Similarly, a venue does not maintain a listing of contact information or biographical information for every band. Yet, all these pieces of information are relevant to the full description of a concert. In order to have a useful element set for describing events a different approach to thinking about metadata is necessary.

Metadata is often described as data about data. This kind of definition is somewhat limiting though from the standpoint of describing events. If one takes a more "event aware" approach to metadata, the definition data about relationships might be more accurate (Lagoze. *Business As Usual*). The Indecs framework is a standard that takes such an approach. Though this standard is geared more towards e-commerce, the underlying idea behind its development provides a useful way of thinking about event aware metadata. Indecs defines metadata as "a relationship that someone claims to exist between two entities" (Rust and Bide 11). Events provide the basis for defining this relationship.

While parts of the metadata describing a concert event may describe more traditional document-like objects, such as a band's webpage or a physical address for a venue, the objects are often unrelated, except in the particular instance of the event being described. A concert event metadata record would tie each of these separate objects together into a single record. In this way, the metadata for a particular event becomes more like a map between different sets, or modules, of metadata. These modules could be maintained in different physical and/or logical areas. Since the metadata simply describes the relationships, if information within each module changes the original metadata record is automatically updated.

The Warwick Workshop that took place in 1996 began to address the idea of modular metadata (Lagoze. *The Warwick Framework*). The result of this workshop was a container architecture called the Warwick Framework. The Warwick

Framework allows the developer to import existing namespaces, or even just specific elements from different namespaces and combine these modules into one container. This type of approach has two major benefits. The reuse of existing schemas prevents the developer from having to create an entirely new namespace thereby saving time. In addition, if elements from well known standards such as the Dublin Core are used it is likely that others will be able to understand the new schema easier. Further development of the Warwick Framework led to the creation of application profiles. Application profiles provide a means to describe the container that is being used for a localized application (Heery and Patel).

As an example, consider the homepage for the Encyclopedia of Arda (http://www.glyphweb.com/arda/default.htm), a website about the works of J.R.R. Tolkien. A metadata record describing this resource might include a Dublin Core description of the website's title (Encyclopedia of Arda), creator (Mark Fisher), subject (J.R.R. Tolkien), etc. In this case, the website also includes a PICS (Platform for Internet Content Selection) description of the content rating, which could be included in the record. One might also consider including administrative metadata about the creator of the metadata record. In this picture, the metadata record is created by piecing together different instances of a number of different modules of metadata.

How does this apply to concert events, though? A concert event can be seen as a relationship between a number of different pairs of entities: a promoter and a concert attendee, a performer and his or her album, a venue and a performer, etc. The nature of each relationship is unique and will change over time; each relationship will also in some way relate to the other relationships. The relationship between the concert attendee and the venue might include information about the venue's location and contact information. The relationship between the concert attendee and the promoter might include information about where to buy tickets and how much they cost. The relationship between the concert attendee and the performer might include the performer's webpage and biographical information about the preformer. Each relationship is a metadata object. The complete metadata record describing an event can be seen as a composite of all of these metadata objects.

Existing Event-like Standards

A number of different event-like standards exist. Perhaps the most widely used and arguably most complex metadata standard for encoding event information is iCalendar (Dawson and Stenerson). Based on the vCalendar standard, iCalendar defines a new Multipurpose Internet Mail Extension (MIME) type and format for encoding and exchanging information about events. It provides a way not only to describe upcoming events but also to create a to do list and an alarm to notify of upcoming events. The standard is used extensively in wireless devices such as PDAs and mobile phones. Applications such as Microsoft Outlook and Lotus Notes have made iCalendar a popular standard in the desktop environment as well. Recently the open source browser, Mozilla, has implemented a calendar extension based on the iCalendar standard. Apple has also based their latest calendar application, iCal, on the iCalendar standard. This in particular has become a very popular implementation of iCalendar that is accessible to the general public (Kahney).

While iCalendar has quickly gained support for describing a wide range of events in the digital environment, it was designed originally with business needs in mind and falls short on a number of levels in describing cultural events such as music concerts (FitzPatrick). In order to address this limitation, the Swedish-based Structured Knowledge Initiative created SKiCal, an extension to the iCalendar standard (FitzPatrick, Lanner and Hjelm). SKiCal extends iCalendar by adding structure for information about directions, pricing and special needs access to name a few. These extensions make iCalendar a much more useful standard for describing event metadata, although SKiCal suffers from the same problem of complexity as does iCalendar, and maybe even more so since it simply adds another layer on top of iCalendar.

In addition to the complexity, another drawback to both iCalendar and SKiCal is the encoding format. Any applications that wish to exchange iCalendar information must use the format defined by the RFCs for iCalendar and SKiCal. While this format allows exchange of information, it lacks the extensibility and modularity that a framework such as XML (Extensible Markup Language) or RDF (Resource Description Framework) offers. Some basic work towards creating an RDF Schema and a DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) schema has been done (Miller, FitzPatrick, Brickley), but much of this remains incomplete and poorly documented. The Mozilla calendar application provides the ability to create an XML file from an iCalendar document, but this does not include files using SKiCal encoded information.

Another standard available for encoding event metadata is the RSS 1.0 (RDF Site Summary) specification. One obvious advantage of RSS 1.0 is that it is based on RDF. While a number of different versions of RSS exist, the 1.0 specification is the only version based on RDF. As a result, the core RSS description can be expanded through the addition of separate namespaces, or modules. One such module extends RSS with support for event information (Roug). In contrast to iCalendar, the RSS 1.0 specification and the event module are relatively simple and as such, much more accessible. The ease of use comes at the cost of decreased granularity, although, this shortcoming can be mitigated through the addition of relevant namespaces.

Perhaps the main advantage in using RSS to describe events is the efficiency in information exchange. Thousands of sites make available RSS feeds for the purpose of information aggregation. One popular news aggregation site, http://newsisfree.com, pulls news from thousands of sites publishing an RSS feed. RSS provides a simple way to automate the exchange of up to date information. Most high level programming languages have some sort of DOM (Document Object Model) or SAX (Simple API for XML) API which can easily manipulate an XML file. A simple XML or RDF parser is all that is required to extract information from an RSS document. In addition, RSS documents have the advantage of being easily generated on the fly from information contained in a database for example. If each concert venue or promoter provided an RSS feed of upcoming concerts, a desktop or server application could fetch up to date concert information tailored to each user or community's needs.

Each of the metadata standards discussed up to this point indeed offer various good points. SKiCal provides a very detailed set of elements focused mainly on describing cultural events while RSS 1.0 offers a simple, interoperable set of elements. Overall though, these standards fall short in their ability to describe music concert events in a useful and interoperable manner.

Objectives

Through an examination of the existing mediums used to describe concert events such as newspapers and flyers, this research will create a list of core elements needed to effectively describe concert events. Specifically, this research was performed to accomplish the following goals:

1. Identify the strengths and weaknesses of currently used event coding metadata standards,
2. Create a core list of elements that effectively describe concert events, and
3. Formulate a base ontology for the description of concert events.

This element set will be used to create a prototype implementation that allows local musicians to inform the public of upcoming concerts.

Methodology

This research involved four related stages. Each stage utilized different methods of approach, which are described in detail below. The following four stages to this research were identified:

- Collection and analysis of local concert advertisements resulting in a crosswalk comparing proposed metadata schema with existing schemas
- Analysis of currently used event coding standards for their effectiveness at describing concert events
- Formulation of an ontology for proposed metadata schema
- Creation of prototype implementation utilizing proposed metadata schema

A selective sample of flyers and concert advertisements from newspapers was gathered. This sample was collected over the span of approximately two months. A total of 36 flyers and advertisements were collected (8 flyers, 28 newspaper advertisements). The flyers were collected from the UNC Chapel Hill campus and Franklin Street in Chapel Hill, NC. The newspaper advertisements were collected from the *Independent* newspaper, which covers the Chapel Hill, Durham and Raleigh area. A content analysis of each flyer and advertisement was undertaken to extract the number of occurrences of specific pieces of information such as a band's name or a telephone number where tickets can be purchased. In order to assess the relative importance and need for repeatability of each element, the number of times that each piece of information occurred in a source was noted. The results of the content analysis were recorded in an Excel spreadsheet containing a list of all the different pieces of information contained within each source. This stage of the research culminated in the creation of an initial element set containing all the pieces of information that appeared in the flyers and advertisements examined. The complete list of core event elements appears in the first column of the crosswalk in Table 1. A crosswalk is a useful tool for identifying differences and similarities between various metadata schemas and for encouraging interoperability (Cromwell-Kessler 19). By mapping the desired elements for a new metadata standard to corresponding elements in existing standards, it becomes much easier for others to see how to interface with the new standard.

A survey of existing event coding standards was then undertaken to ascertain the strengths and weaknesses of currently used techniques. The generation of sample records using two of these standards (RSS and SKiCal) aided in the identification of specific limitations.

The mappings garnered from the crosswalk facilitated the creation of an OWL (Web Ontology Language) ontology to define the semantics of a proposed namespace as well as to provide a machine-readable mapping to other standards. The OWL ontology has the advantage of being a machine-readable alternative for understanding how the different elements within this schema relate to each other and to elements contained within other metadata standards. OWL provides a very powerful language for mapping between standards (Heflin); it was for this reason that OWL was chosen as opposed to RDF Schema. In particular, RDF Schema has no ability to declare two classes or properties as being similar in scope. The equivalentProperty and equivalentClass constructs that OWL provides effectively accomplish the task of mapping between schemas.

The final stage of this research involved the creation of a prototype database that utilized the newly created concert event metadata namespace.

Results and Discussion

Analysis of Flyers and Advertisements

Looking at an element's number of occurrences there were clearly some elements that were more important than others. Only the elements Band (headlining) and Date appeared in every flyer and advertisement. Venue (name) appeared in all but two. In one case, this was because the advertisement was for a house concert. It is not clear why a venue name was not given in the other case. Sponsor was the next most common element appearing in 29 of the flyers and advertisements. Time and Ticket purchase location (geographic) were also relatively common, appearing in over half of the instances. Each of the other elements occurred less than half of the time, some elements occurring only once (Ticket Limit per Customer and Benefit Organization [phone number]).

Crosswalk

What follows in Table 1 is a crosswalk mapping each element from the proposed metadata standard to the corresponding element from the previously discussed standards. If no element is available to describe the desired element the box is left blank. When multiple elements could be used, all are listed. iCalendar, SKiCal, and RSS 1.0 (with event module) are mapped because of their stated goals of being suitable for describing events. Dublin Core is mapped because it is such a widely used and familiar standard.

Table 1. Crosswalk comparing event metadata schemas.

Core Event Element	iCalendar	SKiCal	Dublin Core	RSS 1.0 (with event module)
Age restriction		PROHIBITED: *text* (in place of *text* any description of age restrictions can be given; ex: under 21.		
Band bio			Description	
Band (headlining)		PERSONS; SKiROLE= "PERFORMER"	Creator	
Band (opening)		PERSONS; SKiROLE= "PERFORMER"	Creator	
Band website			Relation	
Benefit organization (name)				
Benefit organization (phone number)			Relation	
Benefit organization (website)			Relation	
Contact information (website)	CONTACT or URL	WHURL	Relation	
Contact information (phone number)	CONTACT		Relation	
Date	DTSTART			ev:startdate (Restriction: required)
Event name		TITLE	Title	title
Sponsor			Publisher	ev:organizer
Ticket limit per customer				
Ticket price		PRICE;ITEM= "TICKET"; CURRENCY= USD:*float* (in place of *float* insert monetary value in form x.xx)		
Ticket purchase location (geographic)		BOOKINGS		

Ticket purchase location (phone number)		BOOKINGS		
Ticket purchase location (website)		BOOKINGS		
Ticket sale start date		BOOKINGS; OPREF="*id*" (Where *id* refers to a previously defined date)		
Time	DTSTART			ev:startdate (Restriction: required)
Time (doors open)	DTSTART (Non-repeatable element; if used for "Time" cannot be used for "Time (doors open)"			ev:startdate (Restriction: required)
Venue (address)	LOCATION		Coverage	ev:location
Venue (name)	LOCATION	PLACENAME	Coverage	ev:location
Venue (phone number)			Relation	ev:location
Venue (website)	LOCATION	WHURL	Relation	ev:location

Evaluation of Existing Event-like Standards

At first glance, it is clear that iCalendar alone is not sufficient for describing concert events. If used with the SKiCal extension it becomes a much more attractive alternative. Likewise, Dublin Core by itself appears unsuitable. RSS 1.0 used in conjunction with the Dublin Core and Event modules appears to be a viable solution though. If a suitable standard already exists it would be unnecessary and counterproductive to create an entirely new standard. Because both SKiCal and RSS seem to provide at least some ability to describe concert events, instances using both formats were created. A number of concert events were chosen at random. Each was encoded using both standards. Example 1 shows an event encoded using SKiCal. Example 2 shows the same event encoded using RSS 1.0.

Example 1. SKiCal encoded event.

```
BEGIN:VCALENDAR
VERSION:2.0
SKICALVER:1.0
PRODID:-//HandGenerated//EN
BEGIN:VEVENT
UID:component_demo_01
SKUID:event_demo_01
CREATED:20021118
DTSTAMP:20021118T163400
NAMESPACE;PREFIX=EX:"http://www.example.com/"
TITLE:Ampt Music Series
DTSTART:20021121T200000
PERSONS;PRXSKiROLE="EX:SPONSER":ASCAP
PERSONS;PRXSKiROLE="EX:SPONSER":Heineken
PERSONS;PRXSKiROLE="EX:BAND":Big Sky
PERSONS;PRXSKiROLE="EX:BAND":Emma Gibbs Band
PERSONS;PRXSKiROLE="EX:BAND":Jupiter Coyote
PLACENAME:Lincoln Theatre
LOCATION:126 East Cabarrus Street/n
  Raleigh, NC 27601-1832
PRICE;ITEM="Admission";CURRENCY=USD:8.00
CONTACT:919-821-4111
WHURL:http://www.ascap.com
END:VEVENT
END:VCALENDAR
```

Example 2. RSS 1.0 encoded event.

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:"http://purl.org/rss/1.0/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:ev="http://purl.org/rss/1.0/modules/event"
>
  <channel rdf:about="http://example.com/concerts.rss">
    <title>Upcoming Concerts</title>
    <link>http://example.com/</link>
    <description>An example concert channel</description>
    <items>
      <rdf:Seq>
        <rdf:li resource="http://example.com/2002/11/21"
/>
     </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://example.com/2002/11/21">
    <title>Ampt Music Series</title>
    <link>http://example.com/2002/11/21</link>
    <ev:organizer>ASCAP</ev:organizer>
    <ev:organizer>Heineken</ev:organizer>
    <ev:startdate>2002-11-21T20:00-05:00</ev:startdate>
    <ev:location>
      Lincoln Theatre
      126 East Cabarrus Street
      Raleigh,NC 27601-1823
    </ev:location>
    <dc:creator>Big Sky</dc:creator>
    <dc:creator>Emma Gibbs Band</dc:creator>
    <dc:creator>Jupiter Coyote</dc:creator>
    <dc:relation>www.ascap.com</dc:relation>
    <dc:relation>919-821-4111</dc:relation>
  </item>
</rdf>
```

One of the most significant limitations of SKiCal is the complexity. SKiCal is not simply a standard unto itself, but an extension of the iCalendar standard. In order to use SKiCal one must also be familiar with iCalendar. Between the two this requires a working knowledge of nearly 200 pages of standards definition and explanation. Many elements of the iCalendar RFC can be discarded off hand, which reduces the complexity, somewhat. The level of granularity available with this standard provides the opportunity to give a relatively accurate and complete description of a concert event. Some limitations and drawbacks were noted though. For instance, the DTSTART component property is a non-repeatable property. This prohibits including both a doors open time and concert start time in the same record. A few possible collisions between iCalendar and SKiCal surfaced as well. The difference between WHURL (SKiCal) and URL (iCalendar) is not clearly stated and there appears to be some overlap. Also, it is not entirely clear how WHURL and CONTACT overlap.

RSS proved to be much easier to work with than SkiCal for concert events. This is due, in some part, to the reduced granularity compared to SKiCal. The downside to this ease of use is that RSS is unable to describe certain important pieces of information, such as ticket price. Another hindrance is the required <title> element within each <item> description. While some concert listings had a readily identifiable title, others did not. One possible solution to this might be to simply use the name of the headlining band as the title. While the modularity of RSS is useful for extending and customizing its descriptive capabilities, the addition of multiple namespaces can quickly lead to overlap between elements. This was observed with the addition of just two modules (Dublin Core and Events). For instance, a concert producer could potentially be classified under ev:organizer, dc:creator, or dc:publisher.

While both of these existing metadata schemas might prove useful to various extents in describing events, neither is ideal. SKiCal has most of the desired elements, but the format is difficult to deal with and lacks extensibility. RSS makes use of a very interoperable format, but lacks important elements. An ideal schema should contain all the necessary elements and use a format that encourages interoperability. Interoperability is easily achieved if a metadata schema takes the relationship approach to metadata.

A closer examination of the new list of elements proposed by this study begins to reveal some possible groupings and relationships. There is information pertaining to the venue, such as name, address, phone number and website. There is information about different bands, such as band name, bio and website. Another grouping might contain ticket information such as cost, where tickets are available, and when tickets are available. A picture begins to take shape that no longer looks so much like a single object located at a specific point in time and space, but rather as a number of different objects each related to the other around a specific point in time. The ideal metadata record for a specific event should really be describing these relationships.

This sort of approach has obvious advantages. If the metadata record is simply a picture of relationships between different objects, the objects themselves do not have to be described they merely have to be identified. Reuse of existing descriptions that may change rarely or even not at all saves time and eliminates copying errors. In addition, if all that is needed is an identifier for these objects, the actual object description could be published and maintained elsewhere. A website functioning as an aggregator of concert information could pull information about a venue or a band directly from their respective websites. If a band's email address changes, the aggregating website will automatically reflect this because it is getting this information directly from the band's website. In this model, the information delivery is faster, more accurate and more efficient.

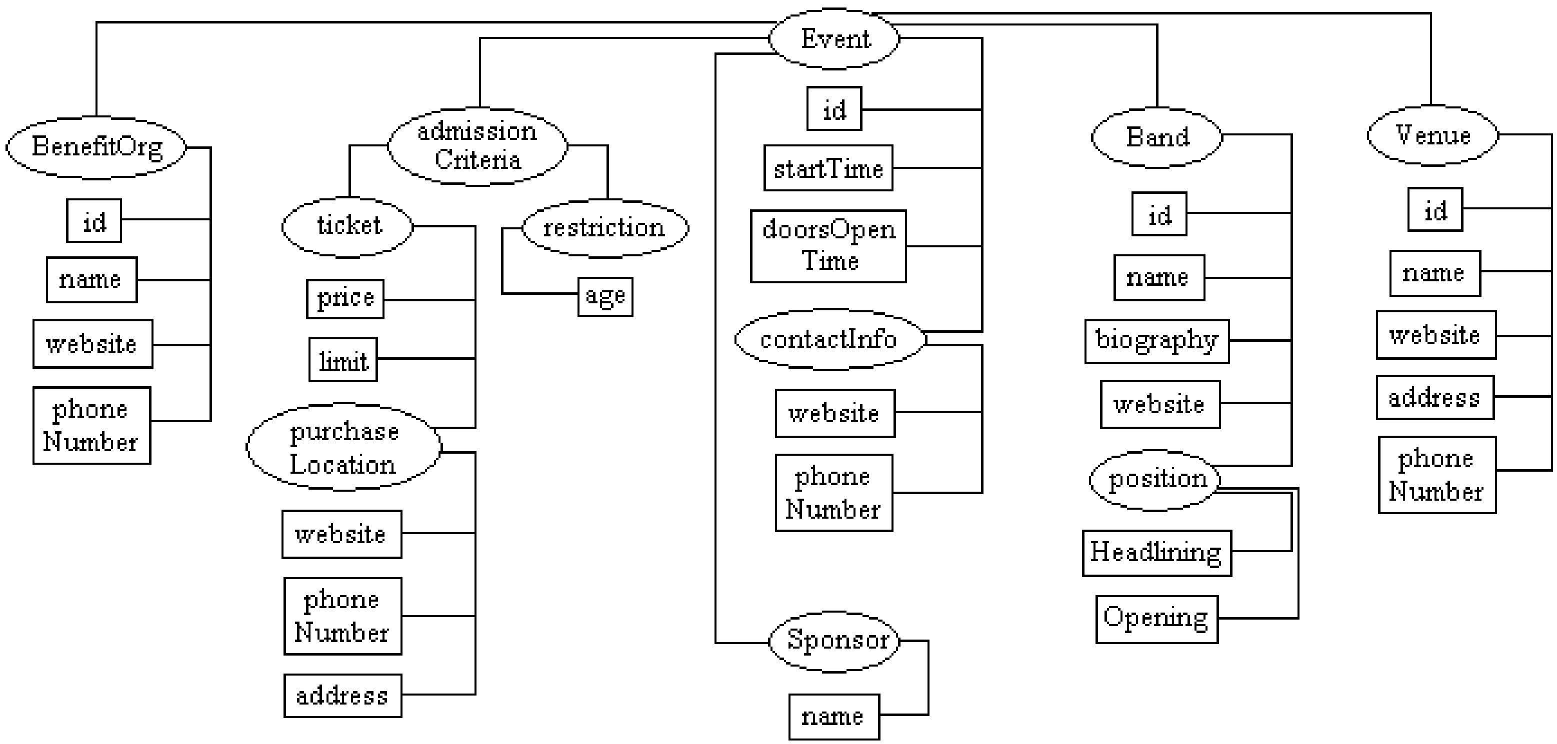
Concert Event Ontology

In order for such a model to function, the information needs to be presented in a structured framework that computers can understand (Berners-Lee, Hendler, Lassila). RDF provides the ability to make this type of model a realization. RDF in a general sense is simply a way of making statements through the use of triples. These triples consist of a subject, a predicate and an object. For example, a simple RDF statement could be made about the author of this document: the author of this document is Michael Graves. In this instance, the document is the subject, author is the predicate and Michael Graves is the object. The recommended format for representing these triples is in RDF/XML. The XML format is widely supported as a web language and so makes a perfect choice for exchanging RDF information on the Internet. Although OWL is still under development, already it provides a powerful way to make statements about relationships between different objects. An OWL ontology could define the semantics for a specific domain of knowledge, such as a namespace for concert events. OWL is also represented in an XML format building upon the existing RDF and RDF Schema languages.

An OWL ontology, which is provided in Appendix A, has been created to define the semantics of the proposed element set. The ontology also provides an effective way of refining and simplifying the element set. For instance, band name and venue name have been combined into the property simply called "name." The ontology dictates that this property can appear within a band description, venue description or event description.

The ontology defines the semantics of the namespace through the creation of hierarchies. These hierarchies also facilitate the modularization of data by grouping certain elements with closely related elements. Figure 1 shows an abstract representation of what a full record might look like using the refined namespace that is defined by the OWL ontology. The illustration makes it easy to see how the information could be grouped into separate modules.

Fig. 1 Concert Event Ontology



Prototype Implementation

Though still a relatively new standard, organizations are beginning to use RDF to encode metadata and design applications. One such application is MusicBrainz (http://musicbrainz.org/), an open source project created to provide a more accurate, interoperable, and open alternative to the commercially owned Internet Compact Disc Database (CDDB) (Swartz 76). MusicBrainz has decided to encode all its data in RDF format. Queries to the database server are also passed as RDF. This data is freely available to anyone. It is apparent how semantic interoperability could be exploited using this framework if concert information were likewise encoded and made available in RDF format. An RDF parser with access to the MusicBrainz database could be programmed to retrieve not simply a listing of a particular artist's albums or listing of that artist's upcoming concerts, but could be programmed to retrieve both sets of information on a particular artist, or even genre. Such an application could be run from a user's desktop and present the user with a list of all the bluegrass concerts in their area within the next week along with a listing of each performer's albums and a short bio on each performer, for example.

While the previous described application is beyond the scope of this research a prototype implementation has been created by the author that draws together many of the ideas discussed thus far. This particular implementation is a web database written in PHP. A MySQL database forms the backend and is used to store the raw data. The driving idea behind this application is to provide an easy and free way for local musicians in the Chapel Hill area to notify the public of upcoming concerts.

While the raw data has been stored in a database, a number of different methods of interfacing with this data may potentially be provided. The first step in presenting the data consists of returning an SQL result set from the database. An XML document using the schema defined by the ontology in Appendix A is generated via a PHP script. Depending on the mode of access, a particular XSL stylesheet is applied to the dynamically generated XML document in order to transform the XML.

The front end of the website provides an XHTML 1.0 representation of the data. Some basic searching and browsing options have been provided for the user to access the information. On the backend, two separate forms of information are available. An XML file is available for an entire event, a single band or a single venue. Due to the previously described limitations of RSS, it was decided that creating an RSS file for each concert would not be feasible or effective. Instead, a simplified RSS feed is made available listing all the concert events for that particular day. Only the title, link and description elements are used for each item. The headlining band's name is used for the title. This feed is generated dynamically so as to have the most up to date information.

While this application accomplishes what it was meant to do, there are a number of limitations and areas for improvement that were noted. A more useful system for managing which users have the ability to edit certain entries would help. The current system requires a user to create a user account before they post new material. For each event, venue or band that are posted the user who posted this information is also recorded. This is done mainly for editing purposes. The application will only allow the user that entered an event to edit it at a later point. There are circumstances where this might lead to problems. If for instance, a user that is not in a band enters information about a band, the real band members would not be able to edit the record describing their band. A more robust user control system might remedy this situation. In the interim an option to email the site admin is provided if anyone feels that there might be an error in any of the information.

Perhaps the biggest limitation of this application is the lack of authority control. For example, someone may enter a venue name as "The Cat's Cradle." Someone else may enter a venue name for the same venue as "Cat's Cradle." There would now be two records describing the same resource in the database. Dealing with this problem at the programming level is an extremely complicated task. It was therefore decided to try and limit the occurrences of duplicate entries through the layout of the site. A drop down list of venues and bands is provided when entering a new event. This not only simplifies the creation of a new event record but should hopefully limit the number of errors that might arise. Still, a future implementation that addresses this problem at a code level would have greater authority control.

Conclusion

This research is important because it highlights the shortcomings of existing standards' ability to describe concert events effectively. Very little research has been conducted in this area. By analyzing how concert event metadata is represented in traditional formats such as print, it becomes easier to create a useful means of representing this information in a digital environment. The advantages afforded by such an environment are numerous – searching can be optimized, data exchange can be automated, related information can be quickly updated and aggregated into a central logical location. The prototype designed as part of this research was an attempt to capitalize on many of these advantages. Future work should include usability studies to measure how well such an implementation exchanges concert information with both users and with other machines. The proposed ontology provides a good starting point for creating an interoperable standard that is useful for describing concert events. However, as with any standard, public input is an integral part of a standard's development and subsequent acceptance by the community. Further work in this area should solicit ideas from the public on how to improve the ontology.

As the amount of unstructured information on the Internet continues to grow it will become more and more difficult to find specific pieces of information. Designing and implementing useful metadata standards is a necessary step towards making that information accessible and useful.

WORKS CITED

Berners-Lee, Tim, James Hendler, and Ora Lassila. "The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities." *Scientific American* 284 (2001): 34-41.

Cromwell-Kessler, Willy. "Crosswalks, Metadata Mapping, and Interoperability: What Does It All Mean?" *Introduction to Metadata: Pathways to Digital Information*. Ed. Murtha Baca. Los Angeles: Getty Information Institute, 1998. 19-33.

Dawson, F., and D. Stenerson. "Internet Calendaring and Scheduling Core Object Specification (iCalendar)." Request for Comments, 2445. 22 Nov. 2002. <http://www.ietf.org/rfc/rfc2445.txt>.

Dekkers, Makx, and Stuart L. Weibel. "Dublin Core Metadata Initiative Progress Report and Workplan for 2002." *D-Lib Magazine* Feb. 2002 <http://www.dlib.org/dlib/february02/weibel/02weibel.html>.

FitzPatrick, G., P. Lanner, and N. Hjelm. "SkiCal – An Extension of iCalendar: draft-many-ical-ski-06." Working paper. 22 Nov. 2002. <http://www.ietf.org/internet-drafts/draft-many-ical-ski-06.txt>.

FitzPatrick, Greg. "SKI, the Swedish Calendar Initiative: Event Publishing in a Universal Marketplace." *INET '99 Proceedings*. 22 Nov. 2002 <http://www.isoc.org/inet99/proceedings/3b/3b_2.htm>.

Heery and Patel. *Application Profiles: Mixing and Matching Metadata Schemas.* 26 Mar. 2003 <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>.

Heflin, Jeff. "Web Ontology Language (OWL) Use Cases and Requirements." Working Draft. 3 Feb. 2003. <http://www.w3.org/TR/webont-req/>.

Kahney, Leander. "Want a Date? We All Cal With iCal." *Wired* 21 Sept. 2002. 22 Nov. 2002 <http://www.wired.com/news/mac/0,2125,55244,00.html>.

Lagoze, Carl. *Business Unusual: How "Event-Awareness" May Breathe Life Into the Catalog?* 22 Nov. 2002 <http://lcweb.loc.gov/catdir/bibcontrol/lagoze_paper.html>.

Lagoze, Carl. *The Warwick Framework: A Container Architecture for Diverse Sets of Metadata.* 26 Mar. 2003 <http://www.dlib.org/dlib/july96/lagoze/07lagoze.html>.

Miller, Libby, Greg FitzPatrick, and Dan Brickley. *SkiCal and iCalendar in DAML+oil: a case study*. 21 Nov. 2002 <http://www.ilrt.bris.ac.uk/discovery/2002/03/skical-daml/>.

Roug, Søren. "RDF Site Summary 1.0 Modules: Event." Working paper, Draft 0.2 2002-07-30. 22 Nov. 2002 <http://web.resource.org/rss/1.0/modules/event/>.

Rust, Godfrey, and Mark Bide. "The <indecs> Metadata Framework: Principles, Model and Data Dictionary." Working paper, WP1a-006-2.0. 22 Nov. 2002 <http://www.indecs.org/pdf/framework.pdf>.

Swartz, Aaron. "MusicBrainz: a Semantic Web Service." *IEEE Intelligent Systems* 17 (2002): 76-77.

Weibel, Stuart. "Metadata: the Foundations of Resource Description." *D-Lib Magazine* July 1995 <http://www.dlib.org/dlib/July95/07weibel.html>.

APPENDIX A: OWL ontology of concert event metadata schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE owl [
   <!ENTITY rdf
     'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
   <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
   <!ENTITY dc 'http://purl.org/dc/elements/1.1/'>
   <!ENTITY ical
     'http://www.imc.org/draft-many-ical-ski#'>
   <!ENTITY xsd 'http://www.w3.org/2000/10/XMLSchema#'>
   <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
   <!ENTITY ev
     'http://web.resource.org/rss/1.0/modules/event/'>
]>

<rdf:RDF
 xmlns:rdf="&rdf;"
 xmlns:rdfs="&rdfs;"
 xmlns:owl="&owl;"
 xmlns:dc="&dc;">

<owl:Ontology
 rdf:about="http://ruby.ils.unc.edu/~gravm/cem/">
  <rdfs:comment>
    An ontology for concert event metadata
  </rdfs:comment>
  <owl:versionInfo>
    $Id: cem-1.0.owl, v.0.5 2003/03/21 19:37:23 graves
    Exp $
  </owl:versionInfo>
  <dc:creator>Mike Graves</dc:creator>
  <dc:title>Concert Event Metadata Ontology</dc:title>
  <dc:date>2003-03-21T19:37:23-05:00</dc:date>
  <dc:format>text/xml</dc:format>
  <dc:identifier>
    http://www.unc.edu/~gravm/cem-1.0.owl
  </dc:identifier>
</owl:Ontology>
```

continued on next page

```
<!-- Class definitions !-->

<owl:Class rdf:ID="Event">
  <rdfs:label>Event</rdfs:label>
  <rdfs:comment>The class Event</rdfs:comment>
</owl:Class>

<owl:Class rdf:ID="Band">
  <rdfs:label>Band</rdfs:label>
  <rdfs:comment>The class Band</rdfs:comment>
  <owl:equivalentClass rdf:resource="&ical;SKiROLE" />
  <owl:equivalentClass rdf:resource="&dc;creator" />
</owl:Class>

<owl:Class rdf:ID="Venue">
  <rdfs:label>Venue</rdfs:label>
  <rdfs:comment>The class Venue</rdfs:comment>
  <owl:equivalentClass rdf:resource="&ev;location" />
  <owl:equivalentClass rdf:resoure="&dc;coverage" />
</owl:Class>

<owl:Class rdf:ID="BenefitOrg">
  <rdfs:label>Benefit Organization</rdfs:lable>
  <rdfs:comment>
     The class Benefit Organization
  </rdfs:comment>
</owl:Class>

<owl:Class rdf:ID="Sponsor">
  <rdfs:label>Sponsor</rdfs:label>
  <rdfs:comment>The class Sponsor</rdfs:comment>
</owl:Class>
```

continued on next page

```
<!-- Property definitions !-->

<owl:ObjectProperty rdf:ID="position">
  <rdfs:label>Position</rdfs:label>
  <rdfs:comment>
    The property defining a band's position in the
    lineup - can be one of either "opening" or
    "headlining"
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Band" />
  <rdfs:range>
    <rdf:Bag>
      <rdf:li rdf:parseType="Literal">
        Opening
      </rdf:li>
      <rdf:li rdf:parseType="Literal">
        Headlining
      </rdf:li>
    </rdf:Bag>
  </rdfs:range>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="admissionCriteria">
  <rdfs:label>Admission Criteria</rdfs:label>
  <rdfs:comment>
    The property defining the admission criteria for an
    event
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Event" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="ticket">
  <rdfs:label>Ticket</rdfs:label>
  <rdfs:comment>
    The property defining ticket information for an
    instance of "admissionCriteria
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#admissionCriteria"
  />
</owl:ObjectProperty>
```

continued on next page

```
<owl:ObjectProperty rdf:ID="purchaseLocation">
  <rdfs:label>Purchase Location</rdfs:label>
  <rdfs:comment>
    The property defining where to purchase tickets
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#ticket" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="restriction">
  <rdfs:label>Restriction</rdfs:label>
  <rdfs:comment>
    The property defining restrictions for concert
    admission
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#admissionCriteria"
  />
</owl:ObjectProperty>

<owl:DataTypeProperty rdf:ID="name">
  <rdfs:label>Name</rdfs:label>
  <rdfs:comment>
    The property defining the name of an object
  </rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&rdfs;Literal" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="website">
  <rdfs:label>Website</rdfs:label>
  <rdfs:comment>
    The property defining a website related to an object
  </rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&xsd;anyURI" />
  <owl:equivalentProperty rdf:resource="&dc;relation" />
  <owl:equivalentProperty rdf:resource="&ical;WHURL" />
</owl:DataTypeProperty>
```

continued on next page

```
<owl:DataTypeProperty rdf:ID="biography">
  <rdfs:label>Biography</rdfs:label>
  <rdfs:comment>
    The property defining a biography of a Band
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Band" />
  <rdfs:range rdf:resource="&rdfs;Literal" />
  <owl:equivalentProperty rdf:resource="&dc;description"
  />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="id">
  <rdfs:label>ID</rdfs:label>
  <rdfs:comment>
    The property defining an ID for an object
  </rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&xsd;nonNegativeInteger" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="address">
  <rdfs:label>Address</rdfs:label>
  <rdfs:comment>
    The property defining a physical address for an
    object
  </rdfs:comment>
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="&xsd;anyURI" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="phoneNumber">
  <rdfs:label>Phone Number</rdfs:label>
  <rdfs:comment>
    The property defining a phone number for an object
  </rdfs:comment>
  <rdfs:domain rdf:resource="&owl:Thing" />
  <rdfs:range rdf:resource="&rdfs;Literal" />
  <owl:equivalentProperty rdf:resource="&dc;relation" />
</owl:DataTypeProperty>
```

continued on next page

```
<owl:DataTypeProperty rdf:ID="price">
  <rdfs:label>Price</rdfs:label>
  <rdfs:comment>
    The property defining the price of a ticket
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#ticket" />
  <rdfs:range rdf:resource="&xsd;decimal" />
  <owl:equivalentProperty rdf:resource="&ical;PRICE" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="limit">
  <rdfs:label>Limit</rdfs:label>
  <rdfs:comment>
    The property defining how many tickets each customer
    is allowed to purchase
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#ticket" />
  <rdfs:range rdf:resource="&xsd;unsignedShort" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="age">
  <rdfs:label>Age</rdfs:label>
  <rdfs:comment>
    The property defining an age to be used for an
    instance of "restriction"
  </rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#restriction" />
  <rdfs:range rdf:resource="&xsd;unsignedShort" />
</owl:DataTypeProperty>

<owl:DataTypeProperty rdf:ID="startTime">
  <rdfs:label>Start Time</rdfs:label>
  <rdfs:comment>
    The property defining when a concert will start
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Event" />
  <rdfs:range rdf:resource="&xsd;dateTime" />
</owl:DataTypeProperty>
```

continued on next page

```
<owl:DataTypeProperty rdf:ID="doorsOpenTime">
  <rdfs:label>Doors Open Time</rdfs:label>
  <rdfs:comment>
    The property defining when doors will open for a
    concert
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Event" />
  <rdfs:range rdf:resource="&xsd;dateTime" />
</owl:DataTypeProperty>

</rdf:RDF>
```