Andrew J. Roback
COM 541
Dr. Otterbacher
27 February 2010

Project One Written Report

The purpose of my project is to model the XML markup for a quotation aggregator that combines player, coach, and front office staff name information with team information and quotations found within the body of news articles on the Chicago Blackhawks news webpage. This webpage is a division of the Blackhawks website and contains articles about Blackhawks players, coaches, front office staff, as well as affiliated teams and persons. The news page itself has an RSS feed which exports certain article information to feed readers, but is has no additional XML markup other than the basic markup necessary to ensure proper placement of exported elements by the feed reader client. My project is designed to include XML markup that provides not only additional information about the persons mentioned in the articles (in the form of team names) but also provides searchability for articles that contain quotations from players, coaches, or front office staff members as well as the content of those quotations.

My rationale for arranging information around quotations is that persons quoted in articles are often the primary subjects of those articles. Many sports articles mention players or coaches repeatedly by name, but only in the context of other players and coaches who are the primary subject of the article. Often, the primary subject of the article is discernable from the title of the article; however, interviews are sometimes conducted with persons who are not named in the article title, but whose interviews comprise a significant portion of the article. Therefore, discerning between persons mentioned in an article versus persons actually quoted within the article is tantamount to discerning between main subject persons in the article and persons who are merely contextual to those main subject persons.

Some articles have no quotations at all, and exist only to describe actions surrounding players; these articles are typically press releases that describe official moves of the hockey organization and do not have interviews or reactions from persons involved. If a user were to search for the name of a player, however, all articles that mention that player, including organizational press releases, will be returned as results.

The goal of creating an instance document that attributes quotations to people mentioned in the article is to separate interview rich news content about individual persons from content where those persons are merely mentioned in passing.

Users, Uses, and Context

I envision that users of such an XML markup scheme would primarily be consumers who read these articles and want a higher degree of searchability and granularity within the news articles. Fans of individual players would benefit from being able to search for news stories where the player is interviewed, and could also potentially follow an RSS feed that utilizes my markup to aggregate articles about one specific player, as opposed to receiving a feed that contains articles about an entire team. Journalists, researchers, and sports writers could search for rich news content to pull quotes from past articles. Additionally, users from the National Hockey League (NHL) or team front office workers could utilize articles marked up in this way to promote and preserve rich content over dryer, press release style articles so that casual browsers would encounter more interesting articles more frequently on NHL or team websites.

A search for XML endeavors in journalism revealed that a Journalistic Markup Language vocabulary was proposed, but never took off as a standard. I believe that a contextual markup language for the users I described would be beneficial in that it would support sports-based

aggregators. Potentially, the markup structure I created for this project would allow sports-based aggregators to collect and display highly specific sports-related content.

Design Choices

The first thing you will notice as absent are XML namespace prefixes. Referencing *Beginning XML 4th Edition*, I found that you can eliminate prefixes by waiting to declare default namespaces in child elements rather than declaring all namespaces up front in the root element. The drawback is that namespaces declared in the child elements only apply to that element and its descendants, and must be declared again if that namespace is reused. Also, in grandchild elements where default namespaces are declared in the child element, an empty namespace string (xmlns="") must be used to cancel the child element default namespace and return the descendants to the default namespace declared in the root element. However, since this does not occur in my instance document (all child element default namespaces persist through the closing of the child tag), it was possible to declare default namespaces within the <player>, <coach>, and <front_office> children in order to avoid the problem of creating a new prefix for each individual person and applying them to the elements. The result of declaring default namespaces in child elements is a prefix free document that is still well-formed XML.

I chose the article URL on the Blackhawks news webpage as the default namespace for each <news_item> element since that URL best describes the location of the article. I could have created my own namespace URI, but the uniqueness of the article ID in each URL leads me to believe that duplicates are not likely since it each is a unique six digit number. Additionally, even if these unique identifiers were to change at some point and a duplicate appeared, it would not be difficult to replace the front end of the URL using the find and replace feature of a text editor in order to reset the URI's to a namespace that was under the author's control (although

the author would ostensibly be the NHL or Blackhawks organization, so this too would be unlikely).

I allocated some pieces of information as attributes rather than elements. The <news_item> element contains an ordinal number attribute as well as the title of the news item. I made the article title an attribute since it will never have children elements, and also to avoid a namespace conflict with the <title> child of the <byline> element, which refers to the professional title of the author of the article. For the <player>, <coach>, and <front_office> tags, the name of the respective individual is an attribute since those pieces of information will never have child elements associated with them.

In order for me to test whether my markup was valid, I chose to utilize a DTD rather than an XML Schema. The DTD I wrote is marked up for cardinality, the most restrictive aspects being that all of the attributes are required and that a person must have at least one team associated with their name (that being one of the goals of the XML markup). Child elements in the <dateline> and <byline> elements were left unrestricted intentionally since they are sometimes absent in press release articles, however I chose to include absent information with self-closing tags; this was a personal preference.

Conclusion

By marking up persons and associating team information and quotations with those persons, users can decide what type of content they wish their search to return rather than sifting through every result returned by a text only search. Additionally, RDF markups could be employed to direct users to official sites for persons mentioned in these articles (although, official NHL sites currently exist only for players). The inclusion of semantic elements such as

the markup developed for this project means greater usability and searchability of news articles for users and the inclusion of journalistic articles in future, specialized aggregator clients.