DescribeX: Interacting with AxPRE Summaries

M. S. Ali, Mariano P. Consens, Shahan Khatchadourian, Flavio Rizzolo
University of Toronto, Canada
{sali, consens, shahan, flavio}@cs.toronto.edu

***Abstract*— DescribeX is a visual, interactive tool for exploring the underlying structure of an XML collection. DescribeX implements a framework for creating XML summaries described using axis path regular expressions (abbreviated AxPRE). AxPRE's capture all the bisimilarity-based proposals in the summary literature and they can be used to define new and more expressive summaries. This demonstration shows how DescribeX helps to analyze diverse XML collections in one particular scenario: the analysis of protein-protein interaction XML data from multiple providers that conform to the PSI-MI schema.**

I. Overview

XML has been adopted as the standard format for numerous applications in data exchange, web-based feeds (blogs, news feeds, podcasts), hypertext collections, and web services. XML schemas are used across different application domains for validating domain-specific XML instances. Schema validation provides a strong basis from which to structure, author and interpret XML data. However, even though two XML collections can be validated against a common schema, the actual structure of the XML instances may be quite different in each of the two collections. This situation may occur because the common schema is extended to allow different user communities to combine schemas freely (*e.g.*, RSS extensions like Yahoo! Media), or document designers may restrict themselves to use just a subset of a larger schema (*e.g.*, best practice guidelines of industry standards like those for IXRetail[1]). In these scenarios, schemas do not provide sufficient information for understanding the structural commonalities of a given collection.

DescribeX is a visual, interactive tool for exploring the underlying structure of an XML collection, capable of handling gigabyte-size datasets. DescribeX is based on a framework (presented in [1] and [2]) for creating XML summaries based on axis path regular expressions (AxPRE, for short). DescribeX summaries are specified by a partition created using the novel notion of bisimilarity applied to subgraphs described by an AxPRE. The elements in the extent of a given partition (represented by a node in the summary) can be computed by an XPath query that is constructed by DescribeX. By employing different AxPREs to define the summary partition, DescribeX can capture all the bisimilarity-based proposals in the existing literature, plus it can also define new and more expressive summaries.

The graph based visualization employed by DescribeX makes it straightforward to see the different path structures that are present in the collection. The application of local node refinements (ie, changing an AxPRE at a given summary node to a different, more detailed AxPRE) can reveal detailed substructure variations. DescribeX functionality helps a user in quickly understanding what parts of the schema are used in practice. Further analysis to find the most common structures and substructures can then be performed in DescribeX through the application of coverage. This provides a strong indication of the similarity of XML instances across a collection. These techniques can also be applied to several collections, in order to compare them against each other and highlight the differences in interpretation and usage of the underlying schema(s).

II. Demonstration of DescribeX

In this demonstration we consider the problem of analyzing multiple collections that use the same underlying schema. When separate communities publish XML data according to the same schema, it is the case that there are still multiple interpretations and varied usage of the underlying schema. In our demonstration of DescribeX, we show how we overcome the challenge of analyzing diverse collections in one such scenario: the analysis of protein-protein interaction (PPI) XML data from multiple providers that conform to the PSI-MI[2] schema. A more comprehensive analysis of several PSI-MI XML collections carried out using DescribeX is available in [3].

We showcase the tool from the point of view of Kelly, a standards designer that anticipates the need to evolve the standard as more PPI information becomes available. In order to act judiciously on how to modify the schema, she will need to understand how various data providers are structuring the information: what parts of the schema are actually in use, how providers are currently interpreting the schema, and the most common structures present. By collecting these pieces of knowledge, she can perform qualitative and quantitative analysis on how providers have structured their data over time. After performing this analysis, she has a better sense of what parts of the current standard are candidates for removal, addition, or to be deemed optional or mandatory.

One way Kelly can go about obtaining the information described above is by looking at some of the files, writing some XPath queries from structures that she would like to understand better, then running the XPath queries against

[1] `http://www.nrf-arts.org/`

[2] PSI-MI stands for Proteomics Standards Initiative, Molecular Interactions, see `http://www.psidev.info/`

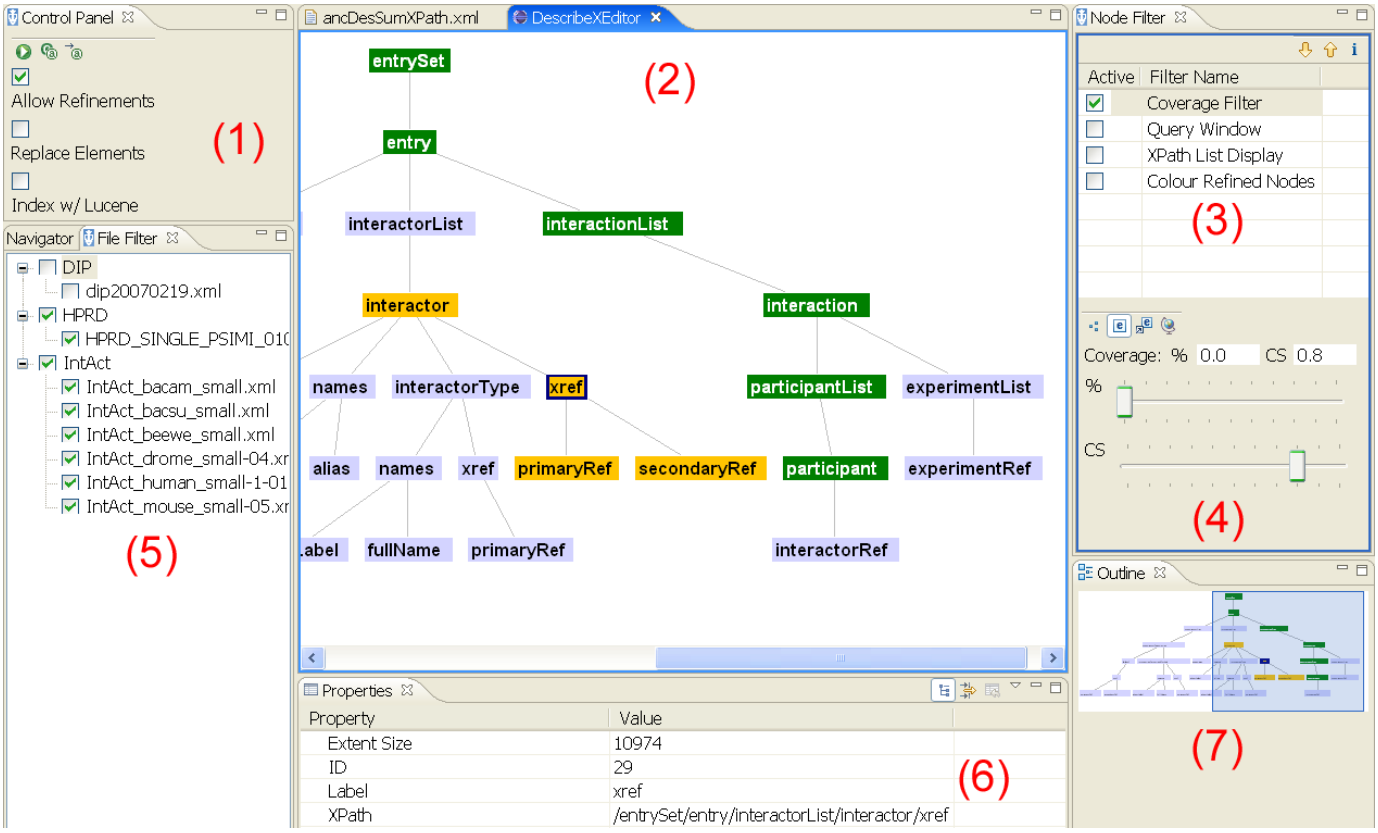

Fig. 1. Screenshot of DescribeX

collections of PPI XML data. While this approach can achieve the desired goal, she will need to create many XPath queries to understand the complete collection in a meaningful way (a very time-consuming process). This initial approach also leaves room to miss unanticipated structures that may exist in only a few files. Fortunately, Kelly has access to DescribeX to examine the collection as a whole. She can also examine specific structures (including unanticipated ones) in detail, as well as easily identify the most common patterns.

Kelly has decided to examine the PPI collections from two data providers; HPRD [4] and IntAct [5]. Figure 1 shows a screendump of DescribeX after she has created a summary of the files in the two PPI collections combined. Kelly selected the XML files to be summarized in the File Filter view (area (5) in Figure 1), and then clicked on the "Create Summary" button in the Control Panel (top leftmost button in area (1) of Figure 1).

The control panel provides access to additional functionality, such as: namespace recognition, replacing elements containing certain attribute values, creating a summary without support for further refinements (to decrease memory usage), exporting the element data for selected nodes, and indexing the XML collection with Lucene (enabling text search of CDATA and attribute content using a variety of scoring methods).

The summary created by Kelly is displayed in area (2) of Figure 1 as well as in a synchronized thumbnail Outline view in area (7) of Figure 1. DescribeX also saves summaries in a persistent format, so the tool can just reload a previously saved summary without re-creating it. The properties of a selected summary node, such as extent size (the number of elements in the corresponding partition) and the XPath expression that computes the elements in the extent, are listed in the Property view in area (6) of Figure 1. The specific summary shown has all nodes described by the same AxPRE $p*$ (p is an abbreviation for the parent axis, and $*$ is the Kleene closure operator), so the summary shown contains all the element paths from leaf to root that occur in the documents of the two collections.

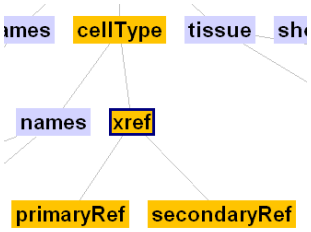

Fig. 2. Prior to Local Refinement of "xref" Element

Detailed substructures can be discerned through local refinement of nodes. For example, the detail in Figure 2 shows Kelly that the element "xref" has two potential subelements: "primaryRef" and "secondaryRef". Kelly would like to ex-

plore whether both subelements are always present within that particular xref substructure. Hence, she selects the ”xref” node and clicks on the “Refine Node” button in the Control Panel view. This action will add child (abbreviated c) to the AxPRE characterizing the subgraphs, partitioning the node extent based on the AxPRE $p * |c)$. DescribeX offers the user a suggested list of commonly used AxPREs for refining the summary. The newly created refined nodes are highlighted in a different colour (in our demonstration they are coloured red, but this can be turned off through an option in area (3) in Figure 1). The result appears in Figure 3 and shows that some “xref” elements contain both “primaryRef” and “secondaryRef” subelements while others contain only “primaryRef” subelements.

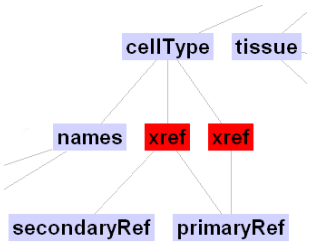

Fig. 3. After Local Refinement of “xref” Element

DescribeX summaries can have thousands of nodes, making it difficult to understand the graph visualization. As an example outside of the proteomics domain, consider the summary shown in Figure 6, representing a collection of several thousand Atom feeds collected from the Web. DescribeX offers the ability to reduce the number of displayed nodes by applying a coverage filter (the actual control used is displayed in area (4) of Figure 1). When the coverage filter is turned on, a property of the nodes is used as a metric to display only those nodes which satisfy the desired coverage percentage. For example, at 50% coverage using the extent size property of a node, the graph is reduced to display only the set of nodes with the highest extent size which together make up 50% of the sum of all extent sizes of the graph’s nodes. A coverage of 25% will display fewer nodes, while the complete graph will be visible at 100% coverage. Other metrics are available and can be switched through radio button selections.

Returning to our PSI-MI summary, coverage of 75% is shown in Figure 4, while coverages of 50% and 25% are shown in Figure 5. Consider how Kelly can make use of coverage in determining the popularity of “xref” elements within the collection. As can be seen through many of the screenshots, the “xref” element is available as a subelement in many substructures. While this may be permitted by the schema, Kelly may decide to disallow its presence in substructures where she sees only exception occurrences. Kelly moves the coverage filter value and the result in Figure 7 shows a summary graph at 91% coverage. While several “xref” nodes in the summary appear at a coverage of 91%, there are other “xref” nodes that would only appear if coverage is increased to capture the last 9% of remaining element occurrences.

Fig. 6. Atom Feed Summary

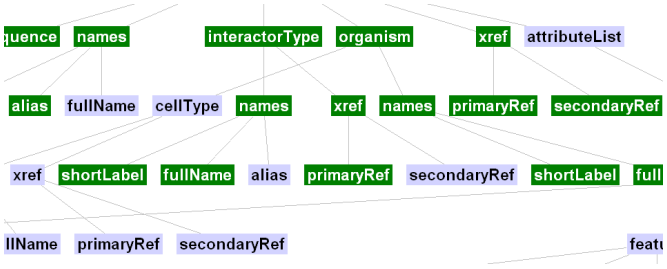

Fig. 7. Locating Popular “xref” Elements at 91% coverage

Now that Kelly has been able to examine the collection through the use of summaries, coverage, and local node refinements, she can now make judicious choices on how to modify the PPI schema standard for effective evolution.

III. DescribeX System

DescribeX is written in the Java programming language leveraging the Eclipse plug-in framework and its existing tools, views, and editors. DescribeX plug-in extension points allow developers to add novel coverage measures and summary techniques, as well as additional metrics (e.g., to help decide whether certain elements should be optional or mandatory). Since the core DescribeX engine is a Java library made accessible as a plug-in, other plug-ins can use the core summary processing API separate from the current user interface.

DescribeX processes XML collections one file at a time. Each file is parsed and summarized, and the file summary is then merged into the collection’s main summary (summaries created separately can also be merged). The summaries can be persisted using several mechanisms (XML files, Lucene indexes, relations). This allows the DescribeX engine to generate summaries of gigabyte-sized collections. As an example of the scale of summaries created with DescribeX, consider the 4.6 gigabyte Wikipedia collection made up of 659,388 files; its $p*$ summary has 245,099 nodes.

While summaries can be generated for gigabyte-sized collections, the actual graph visualization is useful only up to a few thousand nodes. To compensate for the limited usefulness of displaying extremely large graphs, DescribeX resorts to comprehensive coverage and filtering mechanisms.

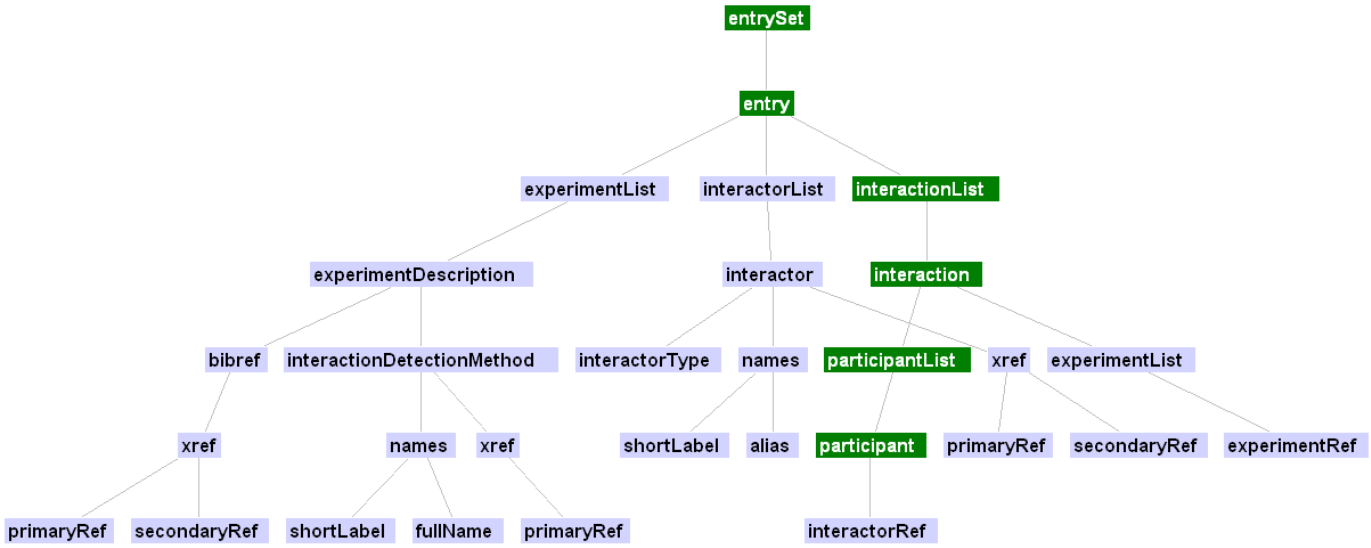

Fig. 4. Summary at 75% coverage

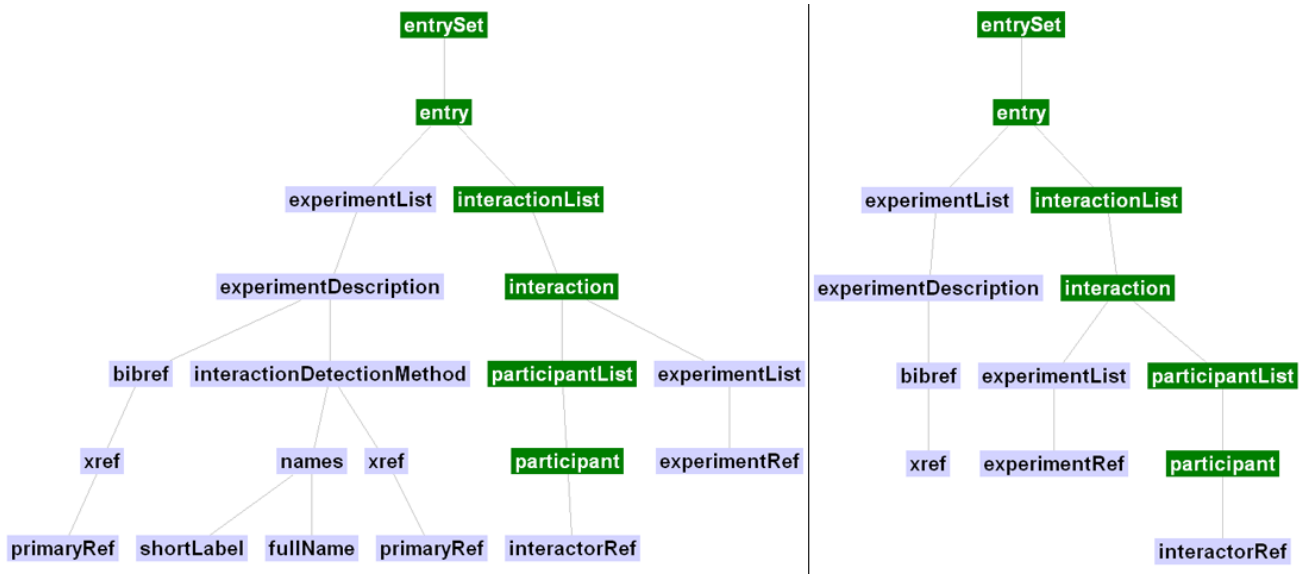

Fig. 5. Summary at 50% and 25% coverage

IV. Conclusions

DescribeX is a versatile tool for exploring XML collections that exploits the flexibility of AxPREs for controlling the appropriate level of detail in summary creation and visualization. The demonstration covers one possible use of the tool: understanding the differences in the instances contributed by multiple XML providers (even when all the providers conform to the same underlying schema).

V. Acknowledgments

Financial support was provided by the Natural Sciences and Engineering Research Council (NSERC), an IBM Faculty Award, and an IBM Center for Advanced Studies Fellowship.

References

[1] M. P. Consens, F. Rizzolo, and A. A. Vaisman, "AxPRE summaries: Exploring the (semi-)structure of XML web collections," in *ICDE*, 2008.

[2] M. P. Consens and F. Rizzolo, "Fast answering of XPath query workloads on web collections," in *XSym*. Springer LNCS 4704, 2007, pp. 31–45.

[3] R. Samavi, M. Consens, S. Khatchadourian, and T. Topaloglou, "Exploring PSI-MI XML collections using DescribeX," *Journal of Integrative Bioinformatics*, vol. 4(3):70, 2007.

[4] S. Peri *et al.*, "Development of human protein reference database as an initial platform for approaching systems biology in humans," *Genome Research*, vol. 13, pp. 2363–2371, 2003.

[5] H. Hermjakob *et al.*, "IntAct: an open source molecular interaction database," *Nucleic Acids Research*, vol. 32, pp. 452–455, 2004.