Hyperbolic Tree for Effective Visualization of Large Extensible Data Standards

Research-in-Progress

Yinghua Ma
Shanghai Jiaotong University

Hongwei Zhu
Old Dominion University

Guiyang SU
Shanghai Jiaotong University

Tubagus Mohammad Akhriza
Shanghai Jiaotong University

Abstract

Large data standards specify tens of thousands of data elements that have intricate relationships. Without effective visualization, it is extremely difficult to understand such large data standards. This is further exacerbated when users are allowed to extend data standards, which in effect produces multiple versions of data standards. In this research, we develop a hyperbolic tree based visualization technique that uses different colors of node labels to distinguish different groups of relationships. Edge colors are also differentiated to visualize extensions to a given standard. The technique is applied to a real-world financial reporting data standard called the XBRL GAAP Taxonomy. Ongoing research will further enhance the technique and evaluate its effectiveness in helping users understand large data standards.

Keywords. Data standards, visualization, hyperbolic tree.

Hyperbolic Tree for Effective Visualization of Large Data Standards

Research-in-Progress

Introduction

Large data standards specify many data elements for multiple users to create semantically interoperable data. When a user adopts a data standard, the user must understand the standard and determine the correspondence between data elements defined in the standard and the data instances the user wants to create. For example, when a user wants to create a data instance for total current assets, the user may find that it corresponds to *AssetsCurrent* in a given data standard. Although the task of finding appropriate data elements in a data standard can be partially facilitated by semantic schema matching tools (Madhavan et al. 2005; Madhavan et al. 2001; Rahm and Bernstein 2001; Rahm et al. 2004), this is largely a manual process because of limitations of these tools. The complexity of large data standards exerts significant cognitive cost on users of the standards. Effective visualization tools can potentially reduce cognitive costs, thereby helping users understand and make appropriate use of the data standards.

In this ongoing research, we develop a hyperbolic tree visualization technique and demonstrate its value by applying it to a large data standard in the financial domain. The standard is the US Generally Accepted Accounting Principle (GAAP) Taxonomy, which has been encoded in eXtensible Business Reporting Language (XBRL) (XBRL International 2006). The Taxonomy has been adopted by the Securities and Exchange Commission (SEC) as a data standard for publically traded companies to use and create financial statements. The 2009 version of the standard specifies more than 13,000 data elements with intricate relationships among the elements. This level of complexity has presented substantial challenges to experts to fully understand the Taxonomy.

Despite the large size of the GAAP Taxonomy, certain information that a company wants to report may not be present in the GAAP Taxonomy. Thus the SEC allows companies to extend the GAAP Taxonomy by creating their own custom data elements. The number of custom elements quickly surpasses the number of GAAP elements. We have observed that by July 31, 2010, more than 30,000 custom elements had been introduced. Thus, it is also desirable to understand how custom elements are related to GAAP elements. To address these needs, we have developed a hyperbolic tree visualization technique for exploring large data standards such as the GAAP Taxonomy and understanding relationships between custom data elements and standard elements.

Background – XBRL, GAAP Taxonomy, and Limitations of Conventional Tee Visualization

In this section, we provide necessary background on XBRL-based GAAP Taxonomy, which we use to demonstrate how the visualization technique can help users understand this complex data standard. We also discuss the limitations of exiting conventional tree visualization.

XBRL and GAAP Taxonomy

XBRL is a *technology* based on XML Schema and XML Linking. It defines a business reporting language by specifying a set of data types, XML elements, and attributes for each element. For example, XBRL defines data types such as *monetaryItemType* and *sharesItemType* that are often used in business reporting. Using XBRL, any jurisdiction can develop its own reporting taxonomy as a *data standard* for companies to exchange business data.

An XBRL taxonomy consists of *taxonomy schemas* that define data elements and *linkbases* that specify various relationships between data elements or between a data element and other resources. For example, below is the specification of the *Assets* data element in the schema of the GAAP Taxonomy:

```
<xs:element id='us-gaap_Assets' name='Assets' nillable='true' substitutionGroup='xbrli:item'
type='xbrli:monetaryItemType' xbrli:balance='debit' xbrli:periodType='instant' />
```

There are two types of elements: *concrete* (by default) and *abstract* (specified using the *abstract* attribute). A concrete element such as *Assets* can be used in data instances (financial statements) with actual values. An abstract element is used by the taxonomy only to conceptually group other elements that usually have a part-of or is-a relationship with the abstract element. Below is how a company uses the Assets element to report its total assets in its financial statement:

```
<us-gaap:Assets id="Item-0039" contextRef="As_Of_12_31_2010" unitRef="Unit12" decimals= "-6">
35067000000</us-gaap:Assets>
```

In addition to data elements, an XBRL taxonomy usually uses five types of linkbases to specify relationships between data elements or between a data element and another type of recourse. A *definition* linkbase specifies conceptual relationships between elements such as generalization-specialization or parent-child relationship. A *label* linkbase provides human-readable descriptions for the elements defined in the taxonomy schema. A *reference* linkbase provides further explanations to the elements by linking them to authoritative references (e.g., SEC regulations or certain accounting standards) that define the meaning of the elements. A calculation linkbase specifies numeric relationships between concrete elements. A presentation linkbase specifies the hierarchical grouping (mainly the parent-child relationship) and the order in which the elements are presented in a report. For example, the following fragment in the GAAP Taxonomy's calculation linkbase specifies that *Assets* is the sum of *Current Assets* and *Non-current Assets*:

```
<calculationArc order='10' use='optional' weight='1.0'
xlink:arcrole='http://www.xbrl.org/2003/arcrole/summation-item' xlink:from='loc_Assets'
xlink:to='loc_AssetsCurrent' xlink:type='arc' />

<calculationArc order='20' use='optional' weight='1.0'
xlink:arcrole='http://www.xbrl.org/2003/arcrole/summation-item' xlink:from='loc_Assets'
xlink:to='loc_AssetsNoncurrent' xlink:type='arc' />
```

When this relationship is represented graphically, each data element, identified by a location ID using either xlink:from or xlink:to attribute, corresponds to a node. Each link, specified by both xlink:arcrole and xlink:type attributes, corresponds to an edge. Thus the above excerpt of calculation linkbase can be represented with three nodes and two edges. A weight of 1.0 indicates addition, while a weight of -1.0 indicates subtraction.

Characteristics of Calculation Relationships in GAAP Taxonomy

Understanding each data element and various relationships in which it participate is a daunting task. In this paper, we focus on calculation relationships and offer observations about them as specified in the 2009 version of the GAAP Taxonomy to help the reader appreciate the complexity of the data standard.

Among 13,452 data elements specified, 10,799 are concrete, of which, 4,703 elements participate in one or more calculation relationships defined by 1,316 formulas. For example, Assets = AssetsCurrent + AssetsNoncurrent is a formula. Conversely, we can think graphically by making Assets a parent node that has two child nodes. Both AssetsCurrent and AssetsNoncurrent have their own child nodes (e.g., many different types of specific assets make up AssetsCurrent). Overall, the relationship structure is large in both width and depth: the maximum number of child nodes is 312, and the maximum of depth is 14. As a result, a data element that serves as a root node can have many descendants. Table 1 shows the top 10 root data elements and the number descendants they have.

Table 1. Top 10 elements have most descendants		
No.	Name	Number of descendants
1	CashAndCashEquivalentsPeriodIncreaseDecrease	1540
2	NetIncomeLossAvailableToCommonStockholdersDiluted	1060
3	Assets	705
4	LiabilitiesAndStockholdersEquity	554
5	StockholdersEquityPeriodIncreaseDecrease	114
6	LoansReceivableNet	109
7	IncomeTaxExpenseBenefitIntraperiodTaxAllocation	107
8	AlternativeExcessNetCapital	105
9	DeferredTaxAssetsGross	58
10	EffectiveIncomeTaxRateContinuingOperations	55

There are two cases when a data element participates in multiple calculation relationships: (1) the element is the "whole" and there are multiple ways of deriving the whole from various "parts" (e.g., two ways of obtaining the whole W: W=A+B and W=X+Y+Z); (2) the element is a part of several whole's (e.g, E is a part of both V and W: V=A+E, W=X+E). Table 2 lists the top 10 data elements with the most formulas in which they serve as the whole (case 1).

Table 2. Top 10 elements have most formulas		
No.	Name	Number of Formulas
1	PaymentsForProceedsFromInvestments	8
2	Assets	6
3	AmortizationOfDeferredCharges	5
4	Revenues	5
5	IncomeLossFromContinuingOperationsBeforeIncome TaxesMinorityInterestAndIncomeLossFromEquityMethodInvestments	5
6	AdjustmentsNoncashItemsToReconcileNetIncome LossToCashProvidedByUsedInOperatingActivities	5
7	loc_IncreaseDecreaseInReceivables	5
8	IncreaseDecreaseInOperatingAssets	5
9	IncreaseDecreaseInOperatingLiabilities	5
10	ShortTermBorrowings	5

Limitations of Existing Visualization Approach

In practices, most tree-like structures are represented using a conventional tree view with clickable nodes to expand or collapse the hierarchical structure. For example, Financial Accounting Standards Board (FASB), the organization tasked by the SEC to maintain the GAAP Taxonomy, uses a conventional tree view for users to browse and search the Taxonomy. Figure 1 is a screenshot of the visualization tool, with two formulas of Assets partially visible. The tool organizes data elements using different financial statements that are familiar to accounting professionals. Data elements are recognized by their labels (as opposed to their names).

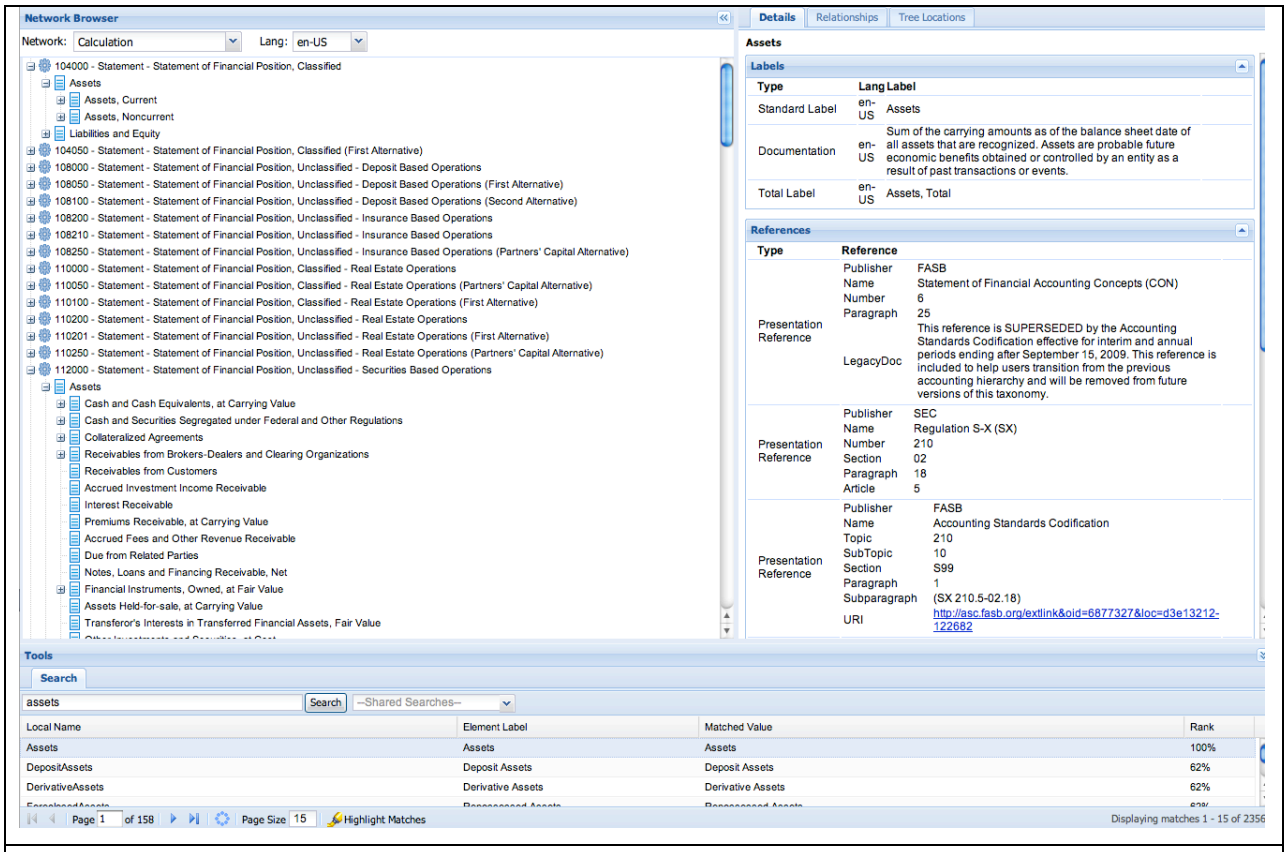

Figure 1. Conventional Tree View

There are several limitations of conventional tree visualization. When an element such as Assets has several formulas, the element appears multiple times and is scattered in several places of the tree. It takes many clicking steps to expand the tree to see related elements. The number of data elements visible is quite limited. Overall, it is very difficult to see the big picture of how a data element is related to other data elements.

Visualization using Hyperbolic Tree

Numerous visualization techniques have been developed, among which various tree-based node-connection techniques are effective for visualizing intricate relationships among a large number of nodes. Comparing with other tree-based techniques such as ConeTrees (Robertson et al. 1991), DOITrees (Heer and Card 2004), and Space Trees (Plaisant et al. 2002) Hyperbolic Trees (Lamping et al. 1995) are the most effective for displaying a large number of nodes in limited space. Coupled with Focus+ technology, a hyperbolic tree allows for smooth zooming to reveal details and at the same time to retain a global view (Sarkar and Brown 1992).

We choose to use the hyperbolic tree technique to address the limitations of the conventional tree technique. With hyperbolic tree technique, a tree is projected into a cured surface called hyperboloid. At initialization, the root is put in the center. Each child of the root is assigned a wedge shaped area, which is shared by descendants recursively. Descendants are located away from the center. The user can drag the tree to move any node to the center. As this happens, branches close to the center are expanded and enlarged automatically to reveal details and branches away from the center are automatically collapsed and shrunk.

We make the following design choices when implementing the hyperbolic tree technique to visualize the GAAP Taxonomy and custom data elements:

1. When an element appears as the "whole" in more than one relationship, the element appears only once and the multiple relationships are distinguished using different background colors of descendants' labels.
2. When an element appears in more than one relationship as a "part", the element appears more than once so that the resulting graph is maintained as a tree as opposed to a network.
3. When custom elements are shown along with the Taxonomy, custom elements are represented as red nodes and Taxonomy elements are represented as black nodes. An edge's color varies between red and black. The "amount" of redness is proportional to the percentage of custom data elements among siblings.
4. A virtual root node is added to integrate multiple trees
5. Use Poincare projection model to place nodes and draw arcs. This model preserves angles between edges and is relatively easy to implement.

The first design choice allows the user to see all relationships of the element in a single place. The second design choice ensures that the edges do not entangle to clutter the screen. The third design choice allows the user to see the relationship between a company's extension taxonomy in the context of the GAAP Taxonomy. The fourth and fifth choices are mainly to facilitate the implementation of the hyperbolic tree technique.

With design choices 1 and 2, the calculation relationships of the 2009 GAAP Taxonomy are a forest with 323 trees. With design choice 4, these trees are connected to form a single tree. Figure 2 shows the resulting tree, with a company's custom elements included (red links). The company is CenturyLink, Inc. (whose stock ticker symbol is CTL).

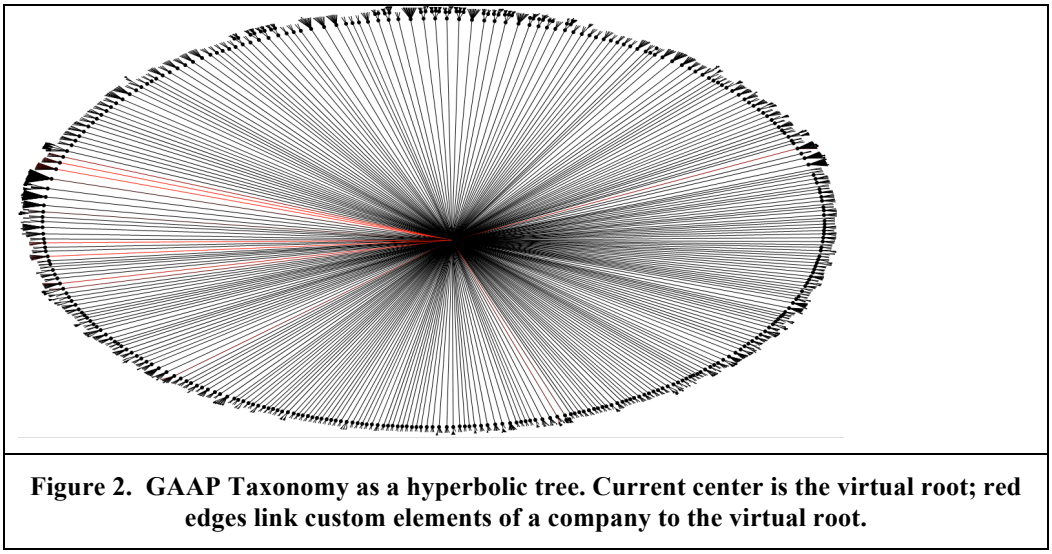

Figure 2. GAAP Taxonomy as a hyperbolic tree. Current center is the virtual root; red edges link custom elements of a company to the virtual root.

Recall that the GAAP element "Assets" has six formulas in which it is the "whole" of different parts. Figure 3 shows how three of the six formulas are visualized when the tree is re-centered (note that prefix loc_ will be removed in next version). Different background colors of labels are used to show different formulas. In Figure 3, formula "Assets=AssetsCurrent+AssetsNoncurrent" is shown in orange color. Two other formulas are shown in green and pink. The other three formulas will be revealed when the tree is re-centered to make the descendant nodes of these formulas closer to the center. In Figure 3, we also observe that CTL introduced a custom element to represent a specific type of current asset (note the red node descending from loc_AssetsCurrent node). When there is enough space to show details of custom data elements, their labels are also in red font color, as shown in Figure 4.

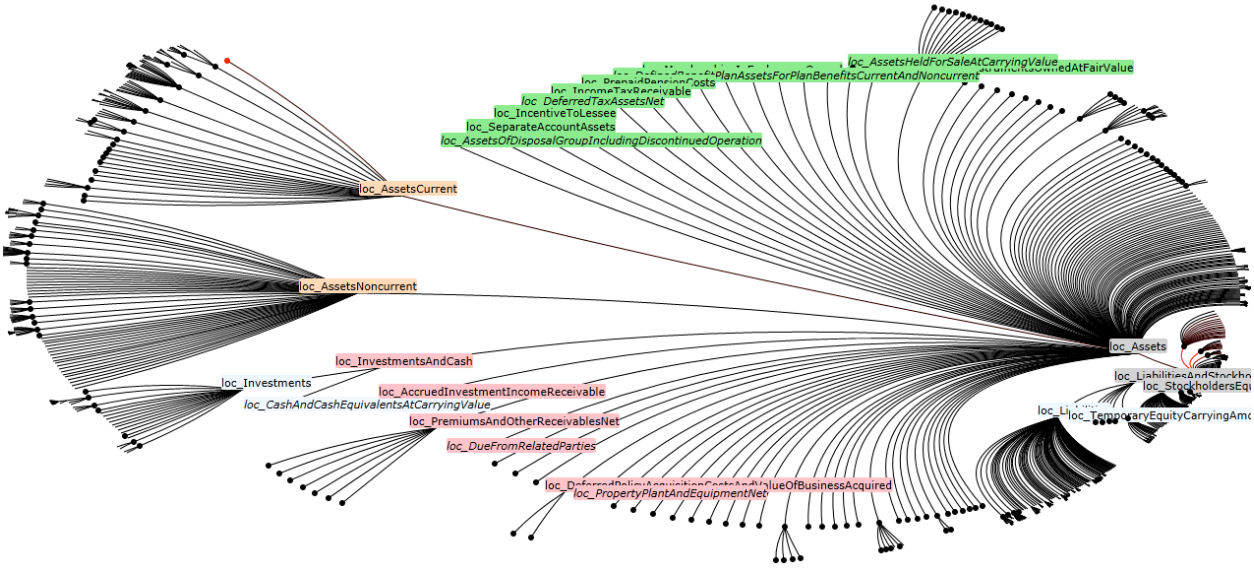

Figure 3. Elements in Different Formula are shown in Different Colors

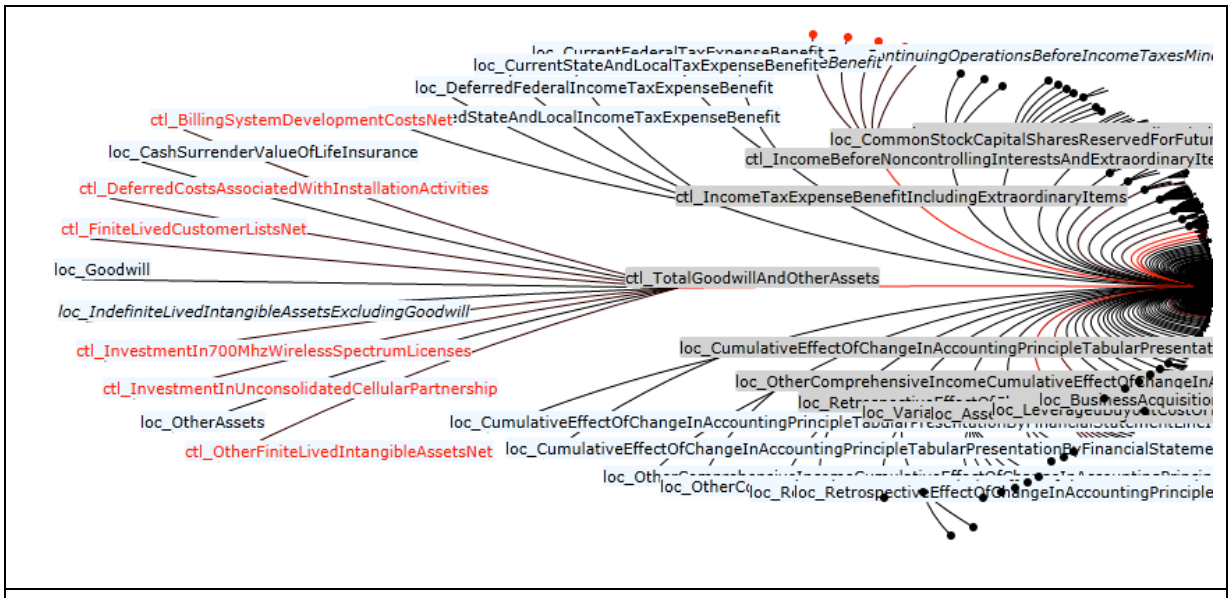

Figure 4. Red font color is for custom data elements.

Conclusion and Future Research

We have developed a hyperbolic tree technique for visualizing large extensible data standard. Although we have not done systematic usability evaluation, the effectiveness of the technique is observable when it is applied to visualize the GAAP Taxonomy and company extensions to the Taxonomy. For example, despite our extensive experience in studying the quality and usage patterns the GAAP Taxonomy, we did not "see" the various ways of calculating total assets until the visualization tool was used. Neither were we able to "visualize" where custom data elements fit in the GAAP Taxonomy without using the tool.

In future research, we will conduct usability tests. These tests will provide feedback on additional features to be incorporated into the visualization tool. For example, perhaps different edge colors and different node shapes can be used in conjunction with label background colors to distinguish different calculation relationships of a data element. The advantage over using just label background colors is that labels are not revealed as often as edges and nodes. Thus by coloring edges and nodes, we provide the user with increased awareness of the existence of multiple relationships. The handling of label display needs to be improved to reduce clutter. We will also improve this visualization tool by adding more interactive features for the user, such as allowing the user to hide certain branches or nodes and hop from one node to a linked node. Additionally, we will add search functions so that nodes can be easily located within the tree. Nevertheless, we believe we have made significant progress towards providing an effective visualization tool to help the user understand large data standards.

Acknowledgement

To be inserted.

References

Heer, J., and Card, S.K. "DOITrees revisited: scalable, space-constrained visualization of hierarchical data," in: *Proceedings of the working conference on Advanced visual interfaces*, ACM, Gallipoli, Italy, 2004, pp. 421-424.

Lamping, J., Rao, R., and Pirolli, P. "A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, United States, 1995, pp. 401-408.

Madhavan, J., Bernstein, P.A., Doan, A., and Alon Halevy "Corpus-Based Schema Matching," 21st International Conference on Data Engineering (ICDE'05), Tokyo, Japan, 2005, pp. 57-68.

Madhavan, J., Bernstein, P.A., and Rahm, E. "Generic Schema Matching with Cupid," 27th International Conference on Very Large Data Bases (VLDB), Morgan Kaufmann Publishers Inc.,, 2001, pp. 49--58.

Plaisant, C., Grosjean, J., and Bederson, B.B. "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation," in: *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, IEEE Computer Society, 2002, p. 57.

Rahm, E., and Bernstein, P.A. "A Survey of Approaches to Automatic Schema Matching," *VLDB Journal* (10:4) 2001, pp 334-350.

Rahm, E., Do, H.-H., and Maßmann, S. "Matching Large XML Schemas," *ACM SIGMOD Record* (33:4) 2004, pp 26-31.

Robertson, G.G., Mackinlay, J.D., and Card, S.K. "Cone Trees: animated 3D visualizations of hierarchical information," in: *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, ACM, New Orleans, Louisiana, United States, 1991, pp. 189-194.

Sarkar, M., and Brown, M.H. "Graphical fisheye views of graphs," in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Monterey, California, United States, 1992, pp. 83-91.

XBRL International "Extensible Business Reporting Language (XBRL) 2.1," XBRL International, 2006.