Dissertation Proposal

Curran Kelleher

3/13/2014

Abstract

There is immense potential value in public data that is not being realized. While publicly available data sets are published on the Web, it is difficult to realize their full value in practice because they are made available using numerous different formats and protocols. The heterogeneity of formats and protocols used makes it difficult to combine and analyze data sets together, and hinders the development of analysis and visualization tools. In this proposal, we present preliminary designs for novel data structures and algorithms supporting integration and interactive visualization of data sets from multiple sources, based on the data cube concept. The proposed data representation framework will allow many data sets from different sources to be combined together and visualized using interactive visualization dashboards with multiple linked views. The focus will be on publicly available data, however the proposed framework can be applied to any data collection whose components can be conceptually modeled as data cubes.

Contents

1 Introduction

Consider the data from the US Census that covers population statistics for US States from 1950 to 2010. Consider also population statistics from the United Nations covering World Countries from 1970 to 2012. These two data sets may use different identifiers for years and geographic regions, but they cover an overlapping conceptual data space of time, geography and population. From these two data sets it is possible to create a visualization dashboard with a map of the world showing population as color and a corresponding line graph showing population for each region as lines. If the user views the whole world, the UN population data is shown for each country. If the user zooms into the US, US Census data is shown for each state. If the user selects a point of time in the line graph, the data shown on the map is from that point in time. If the user pans and zooms on the map, the lines in the line graph update to only show the regions visible on the map.

The contributions of this dissertation are novel data structures and algorithms for integration and interactive visualization of many data sets from multiple sources, based on the data cube concept. The focus will be on public data, however the techniques can be applied to any data collection. The proposed data representation framework will allow data sets to be combined together and visualized using interactive visualization dashboards like the one described, giving users the sense that the data exists within a single unified structure. The framework is designed to be able to represent and inte-

grate an arbitrary number of data sets created independently of one another, and expose the integrated structure to reusable visualization tools that can be combined together in dashboard layouts with multiple linked views. The proposed data representation and visualization framework is fundamentally new, and will allow heterogeneous data sets to be explored in a unified way that was never before possible.

Data cubes, also known as OLAP (OnLine Analytical Processing) cubes, can represent data that contains measures aggregated (typically using sum or average) along categorical hierarchies. The data cube concept emerged from the field of data warehousing as a way to summarize transactional data, allowing analysts to get a bird's eye view of company activities. The term OLAP stands in contrast to OLTP (OnLine Transaction Processing), which is the part of the data warehouse system that ingests and stores data at the level of individual transactions or events. After the ETL (Extract, Transform and Load) phase of the data warehouse flow, the data is analyzed by computing a data cube from the transactional data.

The data cube concept and structure can be used to model existing data as well. Publicly available data sets (often termed "statistical data") may be considered as pre-computed data cubes if they contain aggregated measures (also called "indicators", "metrics" or "statistics") across time, geographic space, and other dimensions such as gender, age range, ethnicity or industry sector. With this approach, it is possible to model many data sets together using shared dimensions and measures which will allow integration of many

data sets together in a single structure. Existing OLAP technologies assume that the data cubes will be computed from a relational source, and are not designed to handle integration of pre-computed data cubes that may use inconsistent identifiers for common dimensions and measures. Therefore the application of the data cube concept to integration and visualization of many pre-computed data cubes, while theoretically plausible, requires the development of novel data structures and algorithms that extend the data cube model to handle integration of pre-computed data cubes that may use inconsistent identifiers for common dimensions and measures.

The envisioned data representation and visualization framework can serve as a digital telescope into the universe of phenomena on Earth via publicly available data. For example, consider public data sources such as the United Nations, the US Census, the US Bureau of Labor Statistics, or the US Centers for Disease Control. These organizations and hundreds of others around the world provide publicly available data about various topics including population statistics, public health, distribution of wealth, quality of life, economics, the environment, and many others. By unifying these data sources and providing users with tools to explore them visually, a deeper understanding of the world can be gleaned by anyone through the lens of public data.

There is immense potential value in public data that is not being realized. The ability to visually explore public data lends itself to applications in education, journalism, and public policy. Especially in the era of "Big Data", it is increasingly valuable for organizations and individuals to have the ability

to analyze large quantities of data from various sources that varies across time, space, and other dimensions. In addition, publicly available data can provide context for business-centric proprietary data analysis activities.

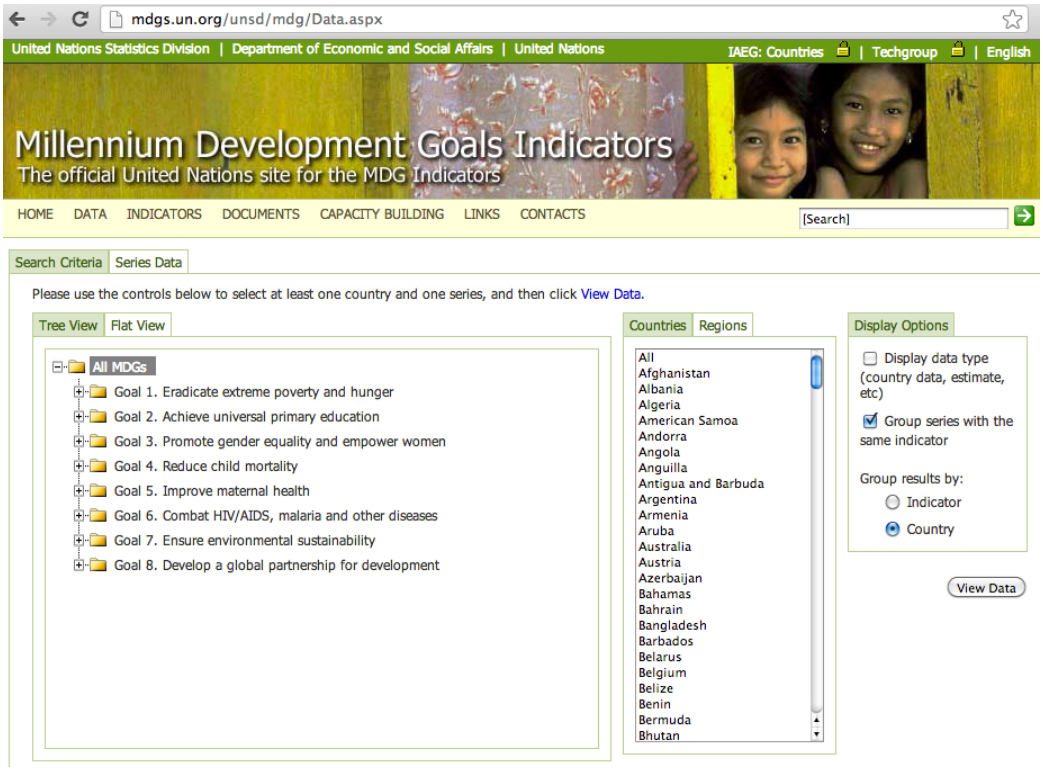


Figure 1: The Web-based interface provided for navigating the United Nations Millenium Development Goals Indicator data sets [85]. This is one example of the variety of formats and protocols used for making public data available on the Web.

While publicly available data sets are available on the Web, it is difficult to realize their full value in practice. The difficulty stems from the fact that they are made available using numerous different representations and protocols. For example, some data sets are made available as CSV files,

Figure 2: The Web-based interface for downloading data from the CIA World Factbook [2]. A data download link is provided that yields a text file using a nonstandard table format. This is a second example of the variety of formats and protocols used for making public data available on the Web.

Excel spreadsheets (such as the one shown in figure 6), or must be navigated using a Web-based user interface such as the ones shown in figures 1, 2, 3 and 4. Sometimes visualization interfaces are provided for public data, such as in figure 5, however these tools are typically extremely limited in scope and hard-coded to the data set at hand. The heterogeneity of formats and protocols used makes it difficult to combine and analyze data sets together, and hinders the development of analysis and visualization tools. With the tools available today such as D3.js, creation of Web-based interactive data visualizations involves hard coding one-off projects to a particular data set.

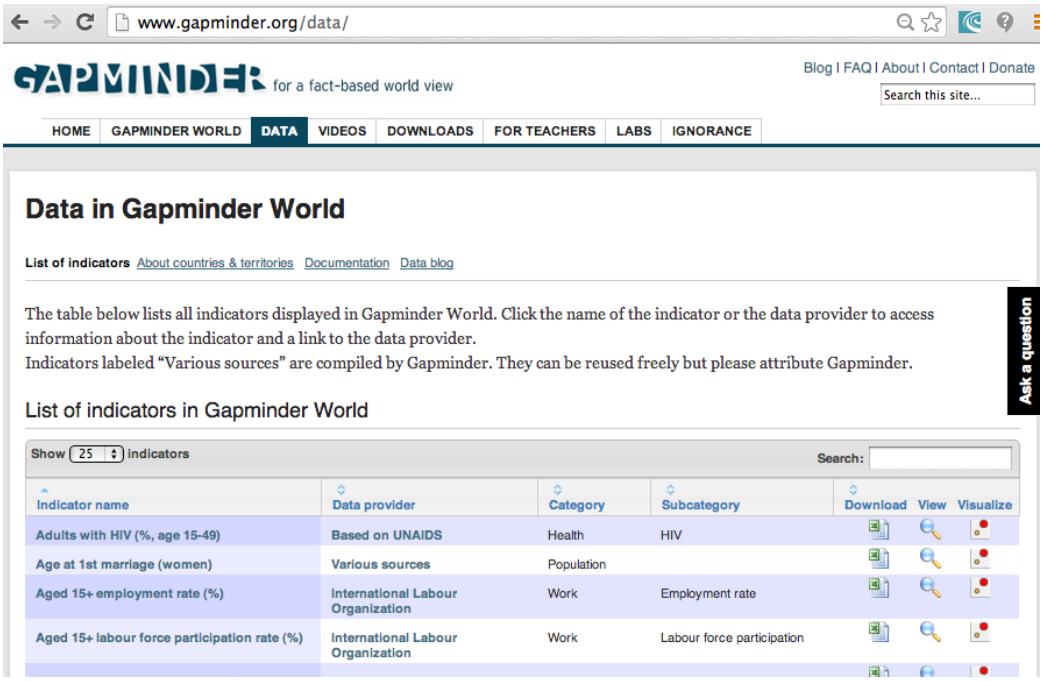
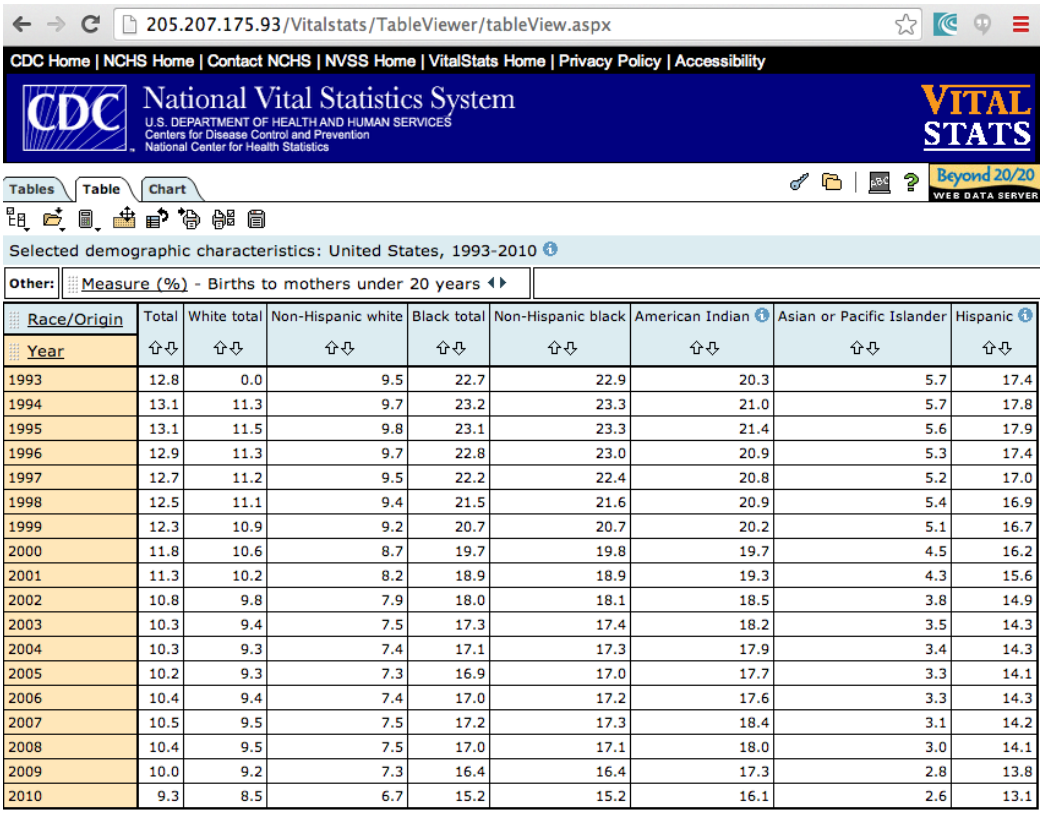


Figure 3: The Web-based interface for downloading data harvested by the GapMinder project [45]. A data download link is provided for each indicator that yields an Excel spreadsheet hosted using Google Docs.

Ideally, anyone should be able to apply interactive visualization techniques to public data easily. This dissertation focuses on the challenges in making this a reality, and offers a solution based on the data cube concept. The proposed framework will reduce the effort required to create Web-based data visualizations by linking reusable visualization templates with public data sets that have been imported into our generalized data representation framework.

Many, but not all, data sets can be modeled as data cubes. Since data cubes are only capable of representing data that has been aggregated along



Race/Origin / Year	Total	White total	Non-Hispanic white	Black total	Non-Hispanic black	American Indian	Asian or Pacific Islander	Hispanic
1993	12.8	0.0	9.5	22.7	22.9	20.3	5.7	17.4
1994	13.1	11.3	9.7	23.2	23.3	21.0	5.7	17.8
1995	13.1	11.5	9.8	23.1	23.3	21.4	5.6	17.9
1996	12.9	11.3	9.7	22.8	23.0	20.9	5.3	17.4
1997	12.7	11.2	9.5	22.2	22.4	20.8	5.2	17.0
1998	12.5	11.1	9.4	21.5	21.6	20.9	5.4	16.9
1999	12.3	10.9	9.2	20.7	20.7	20.2	5.1	16.7
2000	11.8	10.6	8.7	19.7	19.8	19.7	4.5	16.2
2001	11.3	10.2	8.2	18.9	18.9	19.3	4.3	15.6
2002	10.8	9.8	7.9	18.0	18.1	18.5	3.8	14.9
2003	10.3	9.4	7.5	17.3	17.4	18.2	3.5	14.3
2004	10.3	9.3	7.4	17.1	17.3	17.9	3.4	14.3
2005	10.2	9.3	7.3	16.9	17.0	17.7	3.3	14.1
2006	10.4	9.4	7.4	17.0	17.2	17.6	3.3	14.3
2007	10.5	9.5	7.5	17.2	17.3	18.4	3.1	14.2
2008	10.4	9.5	7.5	17.0	17.1	18.0	3.0	14.1
2009	10.0	9.2	7.3	16.4	16.4	17.3	2.8	13.8
2010	9.3	8.5	6.7	15.2	15.2	16.1	2.6	13.1

Figure 4: The Web-based pivot table user interface for downloading data from the US Centers for Disease Control about births to mothers under age 20 by demographic and year [42]. The product powering this interface is the Beyond 20/20 Web Data Server [1]. In this interface a "download" button is provided that yields data in CSV (Comma Separated Value) format.

categorical dimensions, there are many classes of data that do not fit within the model. For example, a database containing the details of transactions in a supermarket would not be appropriate to model as a data cube. Each entry of a customer purchase may contain a listing of items purchased, how it was paid for, and the date and time the purchase was made. This kind of data fits well into the relational model, but is not appropriate to model as a

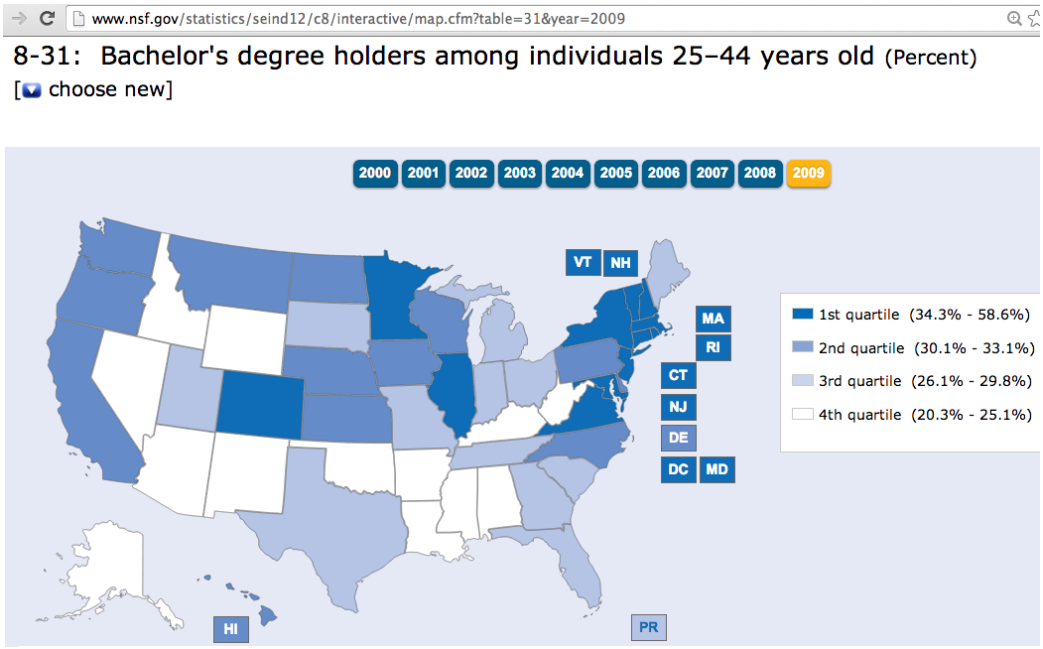


Figure 5: An interactive visualization of bachelors degree statistics provided by the National Science Foundation site [43]. This is an example of an extremely limited visualization tool provided along with a public data set.

data cube. Data cubes represent only aggregated summaries, not individual events. In the case of grocery store database containing the transactions for many grocery stores in different regions, while the individual transaction entries cannot be modeled as a data cube directly, a data cube can be constructed from the transactional data by aggregating measures such as "amount paid" and "number of items purchased" along dimensions such as "time", "region" and "product category". This is a typical data warehouse scenario, where a business aggregates transactional data into a data cube in order to analyze company activities in a summary view.

The key characteristics that allow a given data set to be modeled as a

data cube are as follows:

- The data set contains numeric fields (termed "measures") that represent aggregated summaries using sum, average, or some other aggregation operator.
- The measures of the data set are aggregated along one or more sets of discrete categories or entities (termed "dimensions"). These dimensions can be either unordered, ordered, or hierarchical.

Data sets that have the following qualities may not be modeled as a data cube (although it may be possible to compute data cubes that summarize data sets like these):

- The data set represents a graph. Graph data such as social network connections or links between Web pages is not supported by the data cube model.
- The data set contains relational data with a one-to-many relation. For example, a database of transactions in a grocery store where one transaction has many items. Items containing lists of other items cannot be represented using the data cube model. However, one may consider transforming data sets like this such that the nested lists are summarized by some measures (such as total cost or number of items) so the data cube model can be applied.

- The data set contains entries for individual discrete events or transactions. Data sets with this quality cannot be modeled directly as data cubes, however it may be possible to compute data cubes by aggregating them using OLAP techniques from data warehousing.

Public data tends to be particularly well suited to the data cube model because it typically contains measures about people (or byproducts of human activities) distributed across time, space (geographic regions) and other dimensions such as gender or age range. For example, the public data available in the Gapminder visualization tool contains measures (such as "number of adults with HIV/AIDS" and "child mortality") aggregated across countries and years [45]. This partitioning of space into countries and time into years is one choice of levels in the space and time hierarchies, but the data cube model is more general in that it can support multiple levels of detail in both the Space dimension (e.g. Continents, Countries, States, Counties, and Metropolitan Areas) and the Time dimension (e.g. millenia, centuries, years, months, days, hours and minutes). Therefore any public data sets that contain measures (also called statistics, indicators or metrics) aggregated along any resolution of time and space can be modeled as data cubes.

When multiple data sets are modeled as data cubes, they can be integrated into a single structure. Based on the common dimensions and measures shared between data sets, an integrated heterogeneous data cube structure can be created from an arbitrary number of data sets from multiple sources. Interactive visualization techniques can be applied to this integrated

structure, yielding fundamentally new ways of exploring and presenting multiple data sets.

2 Expected Contributions

The expected contributions of this dissertation include the following:

- Novel data structures and algorithms for data cube integration. Existing formats, protocols and models only consider the case of homogeneous data cubes computed from a single source of relational data, and do not handle the case of integrating many pre-computed data cubes from multiple sources. Data integration has been well studied for relational data, but data integration methods have not been applied to OLAP cubes, which present unique challenges including management of dimension hierarchies and measures that are "universal", or shared by many data sources.

- A conceptual framework that links the integrated data cube structure with existing data visualization theory and techniques. Much work has been done concerning "Visual OLAP" [82], however the visualization approaches for OLAP cubes have not been extended to handle the rich heterogeneous structure introduced by integrating many data cubes from multiple sources.

- A framework for defining visualization dashboards with multiple linked views for interactively exploring integrated data cubes. Interactions between multiple views for OLAP cubes have been considered, but our proposed integrated data cube structure affords a richer set of interac-

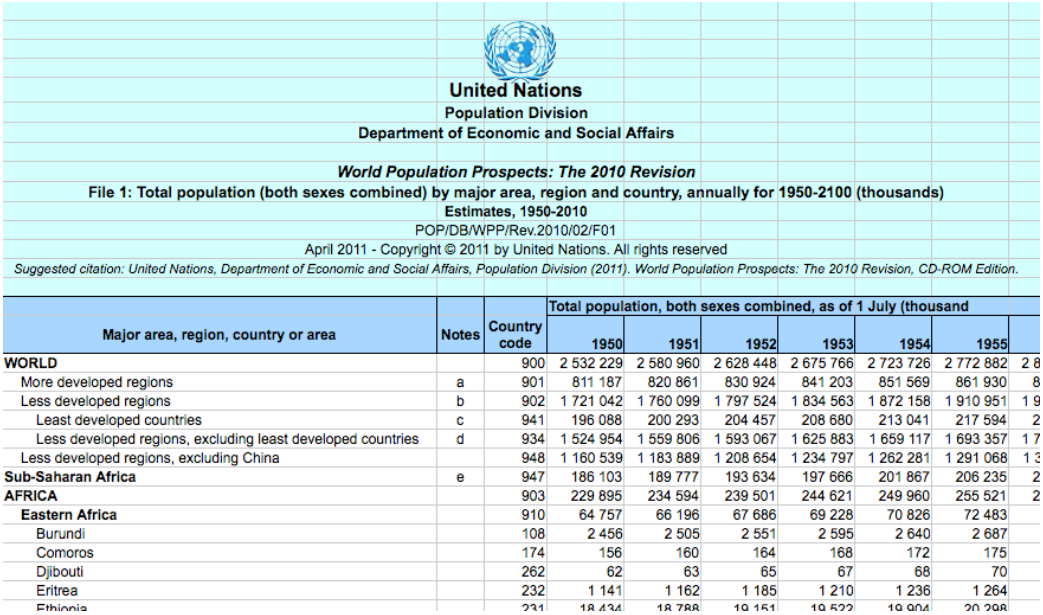
tions that goes beyond traditional OLAP operations such as drill-down, roll-up, slice and dice.

These contributions will advance the field of computing and data visualization by enabling the development of tools for integrating and visualizing heterogeneous data sets in ways never before possible.

3 Related Work

3.1 Data Representation

In today's world of "information overload", data takes many forms. Perhaps the most familiar data representation system today is Microsoft Excel, which is capable of representing data tables as well as complex operations across the data values [38]. Many organizations use Excel to manage data or make data available as Excel spreadsheets. For example, the United Nations Department of Economic and Social Affairs makes their population statistics available in Excel format (see figure 6).



United Nations
Population Division
Department of Economic and Social Affairs

World Population Prospects: The 2010 Revision
File 1: Total population (both sexes combined) by major area, region and country, annually for 1950-2100 (thousands)
Estimates, 1950-2010
POP/DB/WPP/Rev.2010/02/F01
April 2011 - Copyright © 2011 by United Nations. All rights reserved
Suggested citation: United Nations, Department of Economic and Social Affairs, Population Division (2011). World Population Prospects: The 2010 Revision, CD-ROM Edition.

Major area, region, country or area	Notes	Country code	Total population, both sexes combined, as of 1 July (thousand						
			1950	1951	1952	1953	1954	1955	
WORLD		900	2 532 229	2 580 960	2 628 448	2 675 766	2 723 726	2 772 882	2 8
More developed regions	a	901	811 187	820 861	830 924	841 203	851 569	861 930	8
Less developed regions	b	902	1 721 042	1 760 099	1 797 524	1 834 563	1 872 158	1 910 951	1 9
Least developed countries	c	941	196 088	200 293	204 457	208 680	213 041	217 594	2
Less developed regions, excluding least developed countries	d	934	1 524 954	1 559 806	1 593 067	1 625 883	1 659 117	1 693 357	1 7
Less developed regions, excluding China		948	1 160 539	1 183 889	1 208 654	1 234 797	1 262 281	1 291 068	1 3
Sub-Saharan Africa	e	947	186 103	189 777	193 634	197 666	201 867	206 235	2
AFRICA		903	229 895	234 594	239 501	244 621	249 960	255 521	2
Eastern Africa		910	64 757	66 196	67 686	69 228	70 826	72 483	
Burundi		108	2 456	2 505	2 551	2 595	2 640	2 687	
Comoros		174	156	160	164	168	172	175	
Djibouti		262	62	63	65	67	68	70	
Eritrea		232	1 141	1 162	1 185	1 210	1 236	1 264	
Ethiopia		231	18 434	18 788	19 151	19 522	19 904	20 298	

Figure 6: The United Nations Population Prospects data set [86], made available in Excel format. This is another example of a public data set that could be imported into our data representation framework.

Relational database systems provide a mature data management solution and are widely adopted [94]. The relational model has well understood theoretical underpinnings such as the relational algebra [23]. Data warehouse systems are typically built on the relational model, and augmented by multi-scale aggregated data structures called data cubes, also known as OLAP (On-Line Analytical Processing) cubes [47] [24]. Data cubes contain summaries of the collection of facts stored in a relational database [19]. For example, a data cube may contain how much profit was made from month to month subdivided by product category, while the relational database may contain the information associated with each individual transaction. Because data cubes provide a higher level of abstraction, they are a widely used method of data abstraction for supporting visualization and analysis tasks. Kimball pioneered the area of "Dimensional Modeling", which concerns constructing data warehouse schemas amenable to OLAP based analysis [63]. Data cubes have been implemented in a variety of different systems, so effort has been made to discover unified conceptual or mathematical models that can characterize many implementations [32] [116] [115] [70] [3] [49] [13].

NoSQL systems are modern databases that are designed to go beyond the scalability limitations of relational systems [18]. While NoSQL systems sacrifice some of the integrity constraints upheld by relational database systems [109], they are gaining traction in industry because they can handle the scale of data demanded by applications of the "Big Data" era [67]. NoSQL systems provide flexible storage systems that do not necessarily require the

definition of a schema. This makes it arguably easier to modify and update the type of content stored over time as compared to relational systems.

The Semantic Web is a vision of a "Web of Data" coexisting with the World Wide Web [8]. The basis of the Semantic Web is the RDF (Resource Description Framework) data model, which represents a graph of data in the form of (subject, predicate, object) triples. The Semantic Web vision has evolved into the concept of Linked Data, which refers to data that is available as RDF and made available according to common conventions [12] [10]. Any data that can be represented using a relational database can also be represented using RDF [11]. The SPARQL query language for RDF can be used to query and integrate data from multiple sources [92]. Lopez et al. developed an information management system for integrating and analyzing heterogeneous information sources characterizing urban areas [73]. The Semantic Web technology stack contains a method for declaring when different identifiers refer to the same entity and processing queries appropriately to integrate data [51] [33]. While the Semantic Web provides a compelling vision, its adoption is not as widespread as one might expect [74].

The RDF Data Cube Vocabulary is capable of representing data cubes using Semantic Web technologies [30]. The intention of the RDF Data Cube Vocabulary is to provide a common representation and interchange format for statistical data. The RDF Data Cube Vocabulary draws from a previous effort called the Statistical Data and Metadata eXchange (SDMX) initiative that was launched in 2001 by seven organizations working on statistics at the

international level [29]. The primary challenges faced when using the RDF Data Cube Vocabulary include transforming to and from well known formats and data models. Salas et al. discussed how data can be transformed from existing OLAP systems or flat files into RDF using the Data Cube Vocabulary, and also introduced a faceted visualization tool for RDF data cubes [97]. Kämpgen et al. investigated how data represented using the RDF Data Cube Vocabulary can be transformed for analysis using traditional OLAP systems [60]. Maali et al. proposed a pipeline for converting government data into high quality Linked Data utilizing the Data Cube Vocabulary [75].

Datta et al. introduce a conceptual model for data cubes [32]. In this formalization, a data cube (or, in the terms of the authors, a data cube instance) is defined as a 6-tuple (D, M, A, f, V, g) where D is a set of dimensions, M is a set of measures, A is a set of attributes, f is a function that maps dimensions to sets of attributes (the levels of the dimension), V is a set of tuples that assign concrete numeric values for each measure, and g is a function that maps data cube cells to tuples in V. The authors formalize common OLAP operators including slice, drill-down, roll-up, and pivot, as well as operators over multiple cubes including join, union, intersection and difference. This formalism captures the essence of data cubes, but is limited in that it does not deal with integrating data cubes from multiple sources where names used for dimensions, attributes and measures may not match.

Kuznetsov et al. introduce a mathematical formalism of data cubes based on lattice theory [66]. This work focuses primarily on characterizing the

lattice structure of hierarchical data cubes and relating the structure to established mathematics in lattice theory. The contribution of this work is primarily mathematical, and the structures introduced do not cover the entire problem area of representing, structuring and querying complete data cubes. This characterization is similar to the zoom graph concept introduced by stolte et al. [108]. Usman et al. introduced a conceptual model for OLAP enhanced for coupling with data mining and visualization techniques [114].

3.2 Data Integration

The field of data integration offers many techniques for combining data from multiple sources based on the relational model [36] as well as from a theoretical perspective [69] [50] [124]. Schema matching is the area of data integration that concerns semantic matching between the attributes of data tables from different sources [93] [41]. Schema matching may be performed manually, however it must be automated in order to scale to hundreds or thousands of different sources. Numerous approaches for automated schema matching have been proposed [104] [34] [62] [84] [77] [35]. Schema matching approaches aimed specifically at Web and Ontology based data integration have also been proposed [54] [87] [37] [78] [59] [88] [113] [118] [89] [40]. Data matching (also known as record linkage) is the area of data integration focusing on resolving different identifiers to the same real-world entity [121] [122] [65] [4] [48]. Record linkage has been applied extensively to public data [58]

[57] [56]. Several tools have been introduced that aid users in data integration tasks via a graphical user interface [22] [61] [39]. Techniques from both of these areas must be applied in order to integrate data sets from multiple sources and utilize our proposed unified data model.

3.3 Visualization

The field of information visualization offers several compelling theoretical approaches for visualizing data. Arguably the first significant work concerning data visualization was William Playfair's "Commercial and Political Atlas", published in 1786 [91]. In this work, Playfair introduced the Bar Chart, Pie Chart, and Line Graph. The first attempt at a systematic formalization of data visualization was Jacques Bertin's "Semiology of Graphics" [9]. In this work, Bertin relates data types to visual marks and channels in a coherent system that takes visual perception into account. Bertin's work has influenced many future theoretical underpinnings of visualization, including Leland Wilkinson's "Grammar of Graphics" [120] and Jock Mackinlay's APT (A Presentation Tool) system [76], which led to the development of the commercial visualization package Tableau [52].

As a more concrete manifestation of visualization theory, much effort has been placed on generating taxonomies of visualization techniques. Chi et al. introduced a concrete taxonomy of visualizations [20] based on the Data State Reference Model [21]. Shneiderman introduced a more general taxonomy based on tasks and data types [103]. Card et al. made steps

toward characterizing the entire design space of data visualizations based on Bertin's theory [16]. Tufte explored numerous visualization techniques for quantitative information in general, many of which can be applied to visualization of data cubes [112].

Much work has been done regarding visualization of data cubes. Stolte et al. introduced a formalism for defining multi-scale visualizations of data cubes throughout their work on the Polaris system [108] [107] [106]. In this work the authors introduce theoretical underpinnings of a visualization system capable of navigating hierarchical data cubes with a combination of data abstraction and visual abstraction. One fundamental concept in this system is the "zoom graph", a lattice of data cubes that supports arbitrary interactive zoom paths through the multidimensional data cube hierarchies. Cuzzocrea et al. surveyed the area of data cube visualization in depth [26] and have made several contributions regarding semantics-aware OLAP visualization [28] and a hierarchy driven compression technique for OLAP visualization [27]. Mansmann coined the term "Visual OLAP", framed it as a fundamentally new paradigm for exploring multidimensional aggregates [82], explored applications of hierarchical visualization techniques to OLAP cubes [81] and extended Visual OLAP to support irregular hierarchies [80]. Scotch et al. developed and evaluated SOVAT, a Spatial OLAP visualization and analysis tool applied to community health assessments [100] [99]. Lee and Ong introduced a visualisation technique for knowledge discovery in OLAP combining elements of bar charts and parallel coordinates [68]. Maniatis et

al. explored how OLAP cubes can be visualized using TableLens and other techniques [79]. Data cubes have also been utilized as the foundational data structure for several "Big Data" visualization systems [71] [72].

Interactions within data visualization environments have been well studied. Becker et al. investigated brushing in scatter plots [7]. Shneiderman et al. explored dynamic queries in general and how these operations fit into a larger context of visual information seeking [102]. Ward introduced a visualization system based on multiple linked views with direct manipulation techniques including brushing and linking [119]. Anselin discussed how interactive visualization systems with linked views can be applied to Geographic Information Systems [5]. Yi et al. conducted a thorough survey of existing taxonomies for visualization and interactions and developed a set of generalized classes of interactions for visualization [123]. Techapichetvanich et al. explored how visualization interactions pertain to OLAP cubes in particular [110]. Sifer et al. introduced a visual interface utilizing coordinated dimension hierarchies for OLAP cubes [105]. Tegarden formulates some requirements for information visualization relevant for business applications, and highlights some unconventional interactive visualizations with potential application to data cube visualization [111].

3.4 Web Graphics Technology

The World Wide Web has evolved to become a full fledged application development platform. HTML5 is the latest set of standards and APIs (Ap-

plication Programming Interfaces) from the World Wide Web Consortium that define the capabilities of modern Web browsers [55]. HTML5 applications are able to run across multiple platforms (albeit requiring some effort from developers). HTML5 has eclipsed Java Applets and Flash in fulfilling the dream of "write once, run anywhere". HTML5 contains three graphics technologies that can support interactive Web-based visualizations: Canvas, SVG (Scalable Vector Graphics), and WebGL.

HTML5 Canvas provides a 2D immediate mode graphics API [44]. When using the Canvas API, developers must work with a stateful graphics context by issuing commands to manipulate the raster image of a Canvas element within the HTML page. This approach requires developers to manage rendering logic at a low level and manage data structures that correspond to graphical representations. The Canvas API has seen wide adoption for HTML5-based games, however for visualization applications the higher level SVG API has seen wider adoption.

SVG (Scalable Vector Graphics) provides a 2D retained mode Graphics API [31]. SVG uses the HTML DOM (Document Object Model) to represent the definition of persistent graphical elements. When using SVG, developers need only be concerned with updating the DOM. The SVG engine within the browser is responsible for updating the display to correspond with the SVG DOM. In this way, SVG is a higher level API than Canvas. This makes SVG a preferred platform for developing visualizations. However, SVG is less optimizable than Canvas, because developers do not have access

to the rendering logic at all. SVG has performance limitations relating to performance limitations and DOM manipulation overhead.

WebGL provides a 3D graphics API that is essentially an interface to OpenGL ES [83]. OpenGL ES is a subset of OpenGL designed for use in embedded systems and mobile devices. Developers using WebGL must use programming techniques inherited from OpenGL such as buffer management, vertex management, shader definition, 3D projection, and lighting techniques. WebGL enables developers to take advantage of the GPU (Graphics Processing Unit) for massively parallel computation using shaders. WebGL supports high performance 2D and 3D graphics, but is much more complicated to use than Canvas or SVG.

Many high level libraries have been built for supporting use of Canvas, SVG, and WebGL. Three.js is a 3D scene graph library that includes rendering engines for all three graphics technologies [15]. Highcharts is a high level visualization library that provides pre-packaged chart types that can be customized to a limited extent. Leaflet is a library for creating tile-based geographic maps with zooming and panning. hBrowse is a generic framework introduced for Web-based hierarchy visualization [64]. Processing.js is a JavaScript port of the graphics language Processing using HTML5 Canvas. Many more libraries for Web-based graphics and visualization exist, but none have come close to the widespread adoption of D3.js.

D3.js is a flexible and powerful visualization library that uses SVG and has a strong community of users [14]. D3 at its core is a DOM manipula-

tion library with heavy use of functional programming. D3 allows concise declarative statements to define the core logic of visualizations. D3 provides additional APIs for performing common visualization tasks such as defining and using scales, generating labeled axes, and computing layouts from graphs and trees. D3 is at the center of a vibrant developer ecosystem and has seen wide adoption in industry. There are plentiful examples of D3.js usage for creating visualizations, some of which are shown in figure 7. Many supporting libraries have been created including NVD3 reusable charts, Chart.js for composing visualization elements, Crossfilter.js for interactive multidimensional filtering, and DC.js for multiple linked views.

Several projects have focused explicitly on visualization of public data on the Web. ManyEyes was an experiment in scaling the audience for visualizations by empowering users to create visualizations of their own data [117]. ManyEyes provided a fixed set of pre-packaged visualization tools and allowed users to visualize their own data tables using the provided visualizations. GapMinder is a project aimed at exposing public data (primarily the United Nations Millenium Development Goals Indicators) using visualization [96]. GapMinder includes an animated scatter plot with an interactive time slider, a line chart showing statistics over time, and a world map (see figure 8). The Google Public Data Explorer provides a visual interface to selected public data sets similar to GapMinder, however it does not make the data available to users in a machine-readable format.

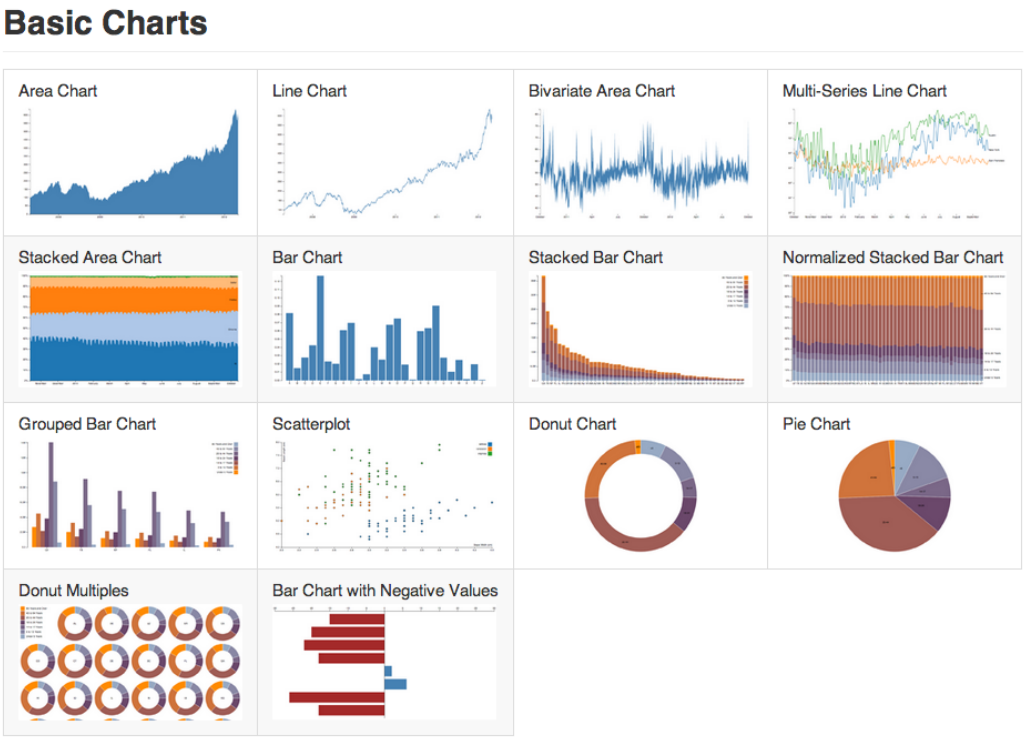


Figure 7: Basic chart examples using D3.js. These are some examples of visualizations that can be tailored to visualize and interact with data cube projections in a generalized manner, building on our proposed framework.

4 Vision

In order to motivate research in data cube integration and visualization, one must consider a larger picture. The public data available today can paint a vivid picture of the world if it is exposed in a meaningful way. Data visualization augments human cognition by enabling users to glean knowledge from data using visual perception rather than detailed mental analysis [17]. Data cubes provide a well structured common representation that captures the essence of many data sets. Data visualization augments human cognition

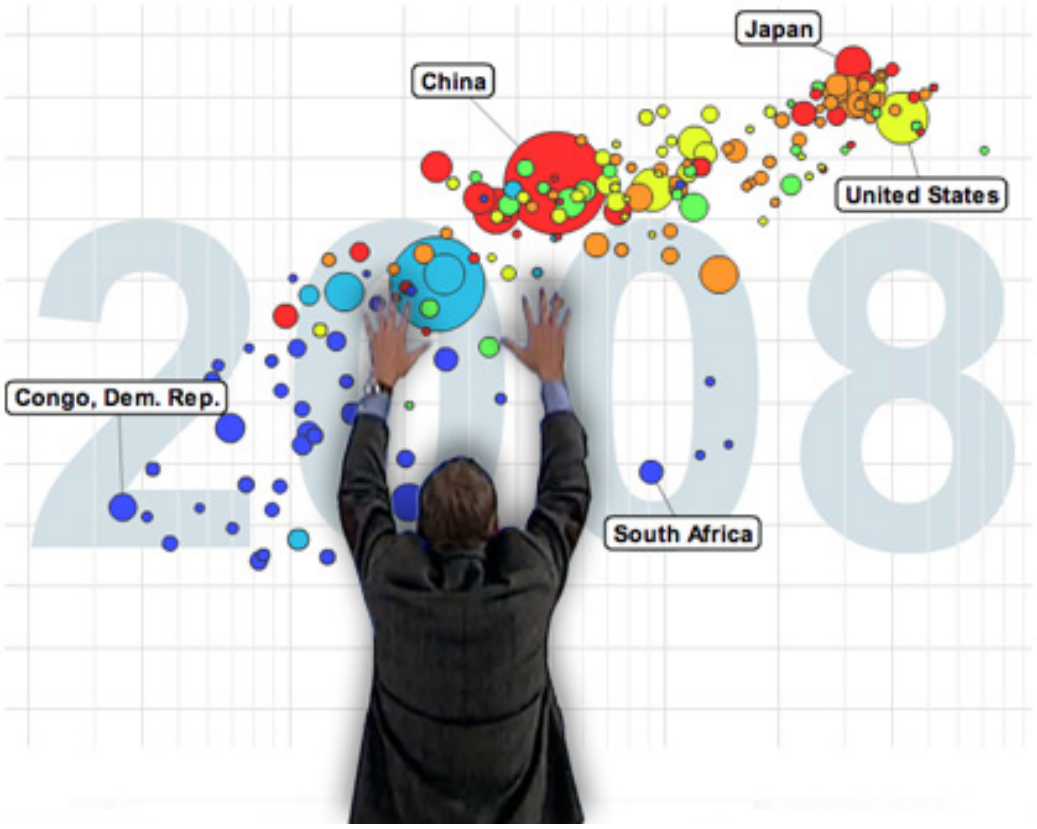


Figure 8: Gapminder, a public data visualization tool based on an animated scatter plot, timeline, and map. Here, Professor Hans Rosling, the creator of GapMinder, is shown gesturing the motion of the plot while presenting the visualization.

by offloading data analysis tasks to tasks of visual perception. The synthesis of public data with visualization through data cubes can lead to a technology platform that changes the world by bringing the power of data visualization to the public.

The main problem this work addresses is the gap between heterogeneous data sets and information visualization software. The reality of the current

data visualization landscape contains many disparate data sets, data formats, visualization tools (specific implementations), and visualization techniques (abstract conceptual visualization approaches). The problem with this situation is that it requires an immense amount of manual work to establish a complete pipeline from any given data source to an instantiation of a visualization technique. This situation is summarized in figure 9.

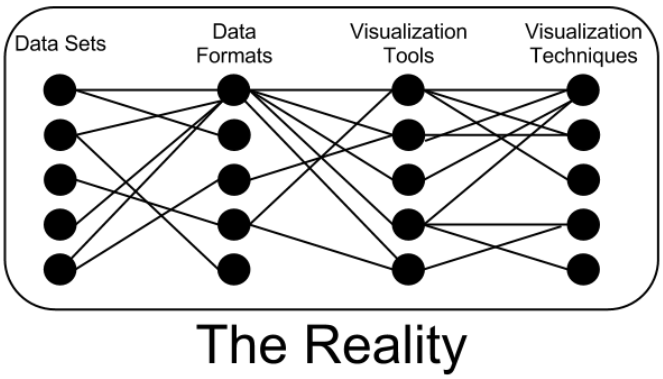


Figure 9: The fragmented reality of public data visualization. This is the primary problem addressed by this work.

An ideal solution to this problem would allow any target data set to be visualized using any target visualization technique. For example, the task "Visualize the US Census Population Statistics on a Choropleth Map" should be possible to execute in a straightforward way, ideally by a simple process in which the target data set is selected (US Census Population Statistics), the target visualization technique is selected (Choropleth Map), and the mapping from the data set to the visualization technique is configured (total population maps to region area color by a linear color ramp). This ideal is

summarized in figure 10.

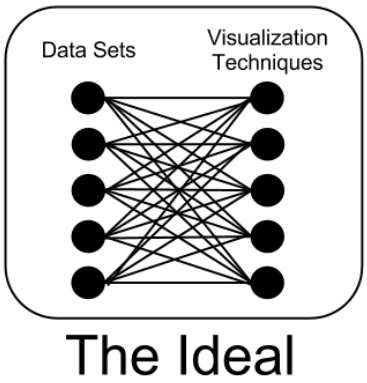


Figure 10: The ideal solution to the gulf between data sets and visualizations.

Our proposed solution to address the gulf between data sets and visualization techniques involves the introduction of a generic data representation and a visualization pipeline based on it. The generic data representation should be capable of representing most public data sets. For this we chose to use the data cube concept as a foundation, as it captures the essential structure of most public data sets we considered. Additional metadata not captured by traditional OLAP systems is also required for generating satisfactory visualizations, such as provenance information and human-readable descriptions of the dimensions and measures involved. This solution is summarized in figure 11.

4.1 Application Areas

Imagine what it would be like if any person could readily access or construct interactive visualizations of public data. Public data and its visualization has

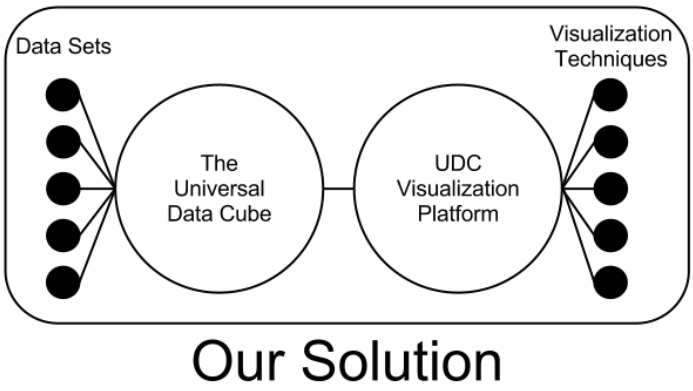


Figure 11: Our proposed solution; introduce generic intermediate representations conducive to data visualization.

relevance to many application areas including but not limited to education, journalism, and public policy.

Educational material is ripe with opportunity for augmentation by interactive visualizations. For example, consider the next generation of textbooks as eBooks running on tablets. Textbooks covering historical trends can use visualization to represent data about, for example, the distribution of various demographics across Earth and how they have shifted over the centuries. Economics courses could use interactive visualizations of global economic data to help students better understand economic dynamics. Environmental studies can include visualizations of public data on climate and pollution. Medical studies can take advantage of public health data. There is no end to the potential applications of public data visualization in education.

Journalism requires an in-depth understanding of stories as they evolve. Public data can provide context for those stories, and interactive visualiza-

tions of relevant data can be placed in digital publications alongside article text. Visualizations are already being used for this purpose today by publishers such as the New York Times and the Boston Globe.

In the area of public policy, people need to make decisions that are complex and can benefit from data analysis insights. However, public policy teams often lack the specialized skill set required to analyze available public data relevant to the decisions at hand. If tools were available that made it easy for anyone to visualize public data, policy makers could utilize such tools to great effect during the policy making process. Discussions could be augmented by explorations of public data, and policy decisions could be backed by visual data presentations that clearly and objectively make a point.

4.2 Data

Consider public data sources such as the United Nations, the US Census, the US Bureau of Labor Statistics, and the US Centers for Disease Control. These organizations and hundreds of others around the world are providing publicly available data about various topics including population statistics, public health, distribution of wealth, quality of life, economics, the environment, and many others. These data sets can be exposed to the general public using interactive visualization.

Most data sources provide essentially data tables. While these data tables can be mapped to visualizations, the potential for automating the visualization generation process is limited because essential metadata is missing from

the table representation. For example, each column in the table may have a name, but the meaning of each column may only exist within documentation for the table, which requires manual effort to track down and integrate into the visualization. Also, with tables it is up to the visualization author to distinguish nominal (dimension) columns from quantitative (measure) columns and choose visual encodings appropriate for each. The data cube model

The proposed data cube based data model is superior to simple data tables for several reasons. By modeling each data set as a data cube using the proposed model, the essential metadata required for automatic visualization generation is present in the data. For example, each dimension and measure of the data cube is explicit in the data representation, eliminating the need for a human to refer to external documentation as part of the visualization process. The proposed data model also explicitly represents four kinds of data relevant to visualization theory. Dimensions can be ordinal, nominal or hierarchical. Measures are quantitative. Visualization theory for ordinal, nominal and quantitative fields has been developed by Bertin [9], while visualization of hierarchical data has been explored subsequently in the literature [6]. The explicit representation of these factors within the data model enables partial automation of visualization creation by providing visual encoding options to users based directly on visualization theory.

In addition to supporting static visualizations, our proposed data model supports interactions within and between visualizations. While tables provide minimal guidance for visualization authors in terms of possible interac-

tions, elements of the data cube model correspond directly to visualization interactions. Measures are quantitative fields that can be used for brushing with interactive filters (for example, with rectangular selection in a scatter plot). Dimensions can be visually selected, then used for defining the slice shown in a linked visualization (for example, selecting bars in a bar chart can drive the data shown in a choropleth map). Hierarchies afford tree-based interactions such as drill-down and roll-up, which can also be used for linking visualizations. For the hierarchical dimensions of space (geographic regions) and time, pan and zoom interactions can define dimension slices.

The proposed work includes a survey of prominent public data sources. This survey will include a listing of data providers, their data access mechanisms, and data cube models of the data sets they provide. This survey of data sets will inform the development of novel data structures and algorithms capable of integrating and querying data sets from multiple sources based on the data cube model. Several data sets surveyed will be transformed into the novel data structure and integrated together as proof-of-concept examples.

4.3 Visualizations

There are many established data visualization techniques such as bar charts, scatter plots, and choropleth maps. Many of these techniques can be understood in terms of the data cube structure they are capable of representing. For example, a simple bar chart is capable of representing a single dimension (defining the meaning of each bar), and a single measure (defining the

height of each bar). As another example, a simple scatter plot is capable of representing a single dimension (defining the meaning of each dot), and two measures (one for the X position and one for the Y position).

The proposed work includes a survey of established visualization techniques. This survey will include a listing of well known data visualization techniques and a characterization of the data cube structure they are capable of representing. For several of these visualization techniques, concrete algorithms will be developed that implement the visualization in a generic manner, building on the novel data structures and algorithms introduced for data cube integration.

4.4 User Tasks

End users of the system should be able to visually explore and present data for their own purposes. The data visualization process involves many steps such as importing the raw data set into the framework, identifying common dimensions and measures between data sets (schema matching), identifying when different identifiers refer to the same entity (data matching), choosing which data sets to use as input, identifying a subset (data cube projection) to use as input to a visualization, choosing which visualization technique to apply, and defining a mapping between the data cube structure and the visualization.

The proposed work includes demonstration of a complete data visualization workflow based on the data representation and visualization framework

introduced. For each step in the process, it is possible to develop a user interface. However, developing a user interface for every step is beyond the scope of this dissertation. Importing data sets into the framework will require programming effort. User interface approaches will be introduced for data visualization steps that use data sets already imported into the framework, such as data set selection, querying, and visualization mapping.

5 Data Cube Representation and Integration

The core contributions of this dissertation are novel data structures and algorithms for data cube integration and visualization. In this section these contributions are formalized. Representation and transformation of multiple data cubes lies at the heart of this project. Therefore a data structure capable of representing multiple data cubes is introduced. Algorithms are introduced for integration of multiple data cubes and for querying the integrated structure for the purpose of interactive visualization.

5.1 Core Data Structures

In the literature on data cubes, terminology is not always consistent. For example, Datta et al. use the term "attribute" to refer to dimension hierarchy levels [32], while the RDF Data Cube Vocabulary uses the term "attribute" to refer to annotations on observations that may relate to scaling factors or the status of the observation [30]. Let us begin our formalization of data cubes with a discussion of the terms that will be used: data set, dimension, level, member, cell, measure and observation.

The conceptual model of our proposed framework is shown in figure 12. In this figure, Crow's Foot notation [53] is used to represent the concepts and their relationships. When two concepts are connected with a line that branches into three lines at one end (the "crow's foot"), it means that there is a one-to-many relationship between those concepts. To be precise, the

concept that the crow's foot points at has a cardinality constraint "one or more" with respect to the concept connected with the single line end. A perpendicular line to the crow's foot represents optionality (whether the relationship is mandatory or optional). When the perpendicular line is present, it means it is mandatory (not optional) that there be exactly one instance of the concept connected with the single line end present. For example, in the connection between `DataSource` and `DataSet`, the crow's foot means "A `DataSet` has many `Observations`", and the perpendicular line means "Every `Observation` is associated with exactly one `DataSet`".

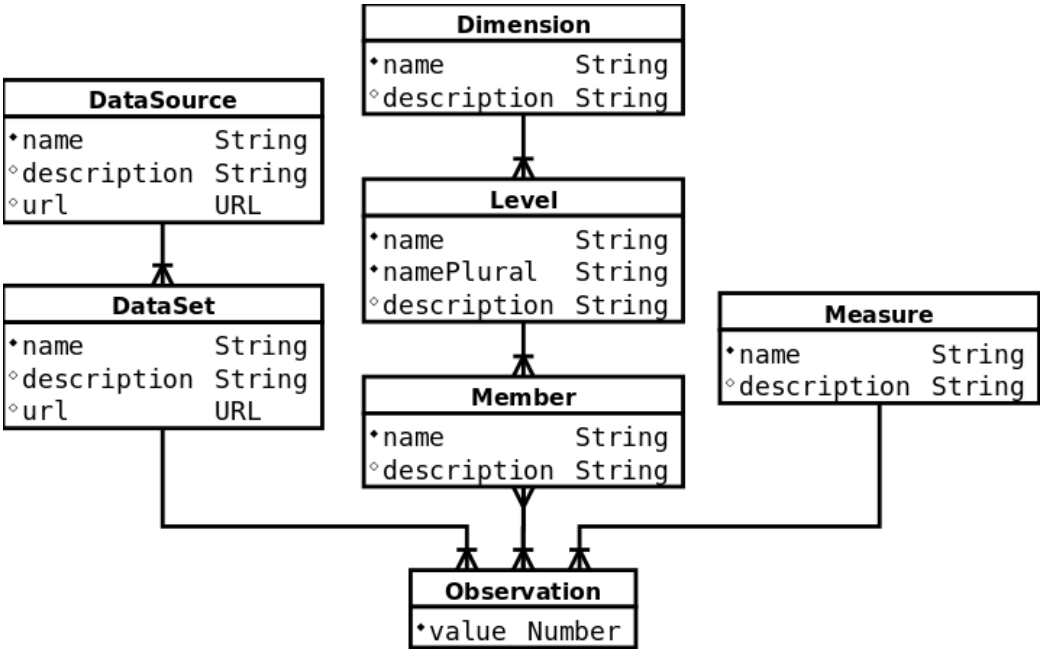


Figure 12: The preliminary conceptual model of our proposed unified data representation framework.

The term "data set" will be used to refer to a collection of observations

from the same source where each observation shares a common set of dimensions and measures. A data set can be annotated with metadata such as the title of the data set, its source, and a human readable text description. This is similar to the term `DataSet` in the RDF Data Cube Vocabulary [30]. One example data set would be "United Nations Population Estimates", which contains population values at the country level since 1950.

The term "dimension" will be used to refer to a set of entities that may be arranged in a hierarchy. The entities within a dimension hierarchy are called "members" of the dimension. Members may fall into a particular "level", or depth of the hierarchy. One example dimension would be "Time", having levels "Year", "Month", and "Day". Members of the Time dimension at the Year level might include "1950", "1951", and so on. Space and Time are dimensions that occur in most data sets. Additional dimension examples include "Gender", "Ethnicity", "Age range", and "Industry".

The term "cell" will be used to refer to a set of members. Each member in a cell comes from a distinct dimension. Data sets typically include multiple dimensions. For example, the UN Population Estimates data set includes both the Space and Time dimensions. In this data set, population values are assigned to combinations of Space and Time. A specific combination of members, one from Space and one from Time, defines a cell in this data set. One example of a cell would be "France in 1950". The number of members defining a cell must correspond to the dimensions of the data set within which the cell is used. If the data set covers, for example, Space, Time and Gender,

an example of a cell would be "Females in France in 1950".

The term "measure" will be used to refer to the properties for which numeric values can be assigned to cells. Measures almost always represent a count, sum or average over the abstract space defined by cells. For example, "Population" is a measure (the count of people). Other example measures include "Average income", "Number of people with AIDS", "Child Mortality Rate", "Gross Domestic Product", "Area", and "Population Density".

Some instances of these terms are universal (can be referenced by many data sets) while some are local (exist only within a data set). Dimensions, levels, members and measures are universal. Instances of these may be referenced in many data sets. Observations are local to particular data sets. This distinction of universal versus local scope is necessary when considering the challenge of integrating many data sets together, because the reconciliation of entities across independently published data sets is critical for data set integration.

Let us define universal data cube metadata as a tuple (D, L, E, l, e, p, M) where

- D is a set of dimensions.
- L is a set of levels.
- E is a set of members.
- $l : E \to L$ is a one-to-one mapping that assigns a level to a member. Although in practice it may not make sense to assign levels to all mem-

bers (for example, consider the "Gender" dimension), let us make the simplifying assumption that every member has a level.

- $e : D \to E$ is a one-to-many mapping that defines the set of members that fall within a given dimension. Every member should only fall within a single dimension.

- $p : E \to E$ is a one-to-one mapping that defines the parent-child relationship between members. For a given member m, $p(m)$ is its parent member in the dimension hierarchy.

- M is a set of measures.

Let us define a data set S as a tuple $(D_S, L_S, E_S, M_S, C, V, g)$ where

- $D_S = \{d_1, d_2, \ldots, d_n\}$ is the set of n dimensions present in the data set.

- L_S is the set of levels present in the data set.

- E_S is the set of members present in the data set.

- $M_S = \{m_1, m_2, \ldots, m_k\}$ is the set of k measures present in the data set.

- C is the set of cells in the data set, defined by the cartesian product of member sets for each dimension as follows: $C = e(d_1) \times e(d_2) \times \ldots \times e(d_n)$ where $e(d)$ represents the set of members in E_S that fall within dimension d.

- V is the set of values corresponding to observations in the data set. Each value $v \in V$ is a tuple $(u_1, u_2, \ldots, u_k)$ where each u_i is a numeric value corresponding to the measure $m_i \in M_S$.

The proposed work involves extending this mathematical formalism of our unified data representation framework to support all operations necessary for visualization of data. More specifically, the operations introduced in the OLAP model and algebra from Datta et. al [32] including restriction (slicing), aggregation (roll-up), join, and union will be considered for extension into our model. The extension will involve handling multiple data cubes simultaneously, and dealing with the heterogeneous data cube structures that result when querying across multiple cubes.

5.2 Integrating Multiple Data Cubes

The data structure introduced separates the universal structural elements from the values specific to each data set. This means that when importing a data set into this data structure, a mapping between dimension and measure identifiers found in the raw data set and universal dimension and measure definitions must be established. When a data set is imported that includes dimensions or measures not found in any other data set already imported, universal dimensions and measures must be created from the data set. When a data set is imported that includes dimensions and measures shared between it and previously imported data sets, a mapping between the identifiers in

the data set being imported and the existing universal definitions must be established. The process of establishing such mappings can draw from well known data integration processes of schema matching and data matching.

This approach implies that the data integration tasks of schema matching and data matching must take place before any data set is imported into the data structure. In other words, it is not possible for the data structure to contain multiple data sets for which matching has not already been done, unless duplicate universal dimensions or measures have been inadvertently created.

When querying the integrated data cube structure, heterogeneities may arise that would not occur in conventional OLAP systems. For example, two data sets may provide different values for the same measure in the same cell for dimension subsets in which they overlap. In processing query results for visualization, one approach that can be taken is to take the average of the values. This would mean that for areas in which the data sets do not overlap, values would be taken only from the single data set that provides the value, where in overlapping areas the average value from each data set will be provided. This gives the user the impression of a single uniform larger data set. The overlapping areas may also reveal discrepancies between data sets (by analyzing the differences in value), but this is an area of future research beyond the scope of this dissertation.

5.3 Crowdsourcing Data Experiment

Rather than manually curating data, a crowdsourcing approach can be taken to data collection for the UDC. We have performed an initial experiment to test the feasibility of this approach. Amazon Mechanical Turk supports assignment of tasks, called "Human Intelligence Tasks" or HITs, to workers who get paid small amounts (on the order of cents) to execute the tasks. To populate the UDC using Mechanical Turk, HITs can be devised that ask workers to find an answer to a simple question like "What was the population of India in 1950?". This question is an instance of a more general form "What was the `${measure}` of `${place}` in `${time}`". By enumerating possible values for `${measure}`, `${place}`, and `${time}`, responses to such HITs can populate large regions of the UDC.

To test the crowdsourcing data collection approach, an experiment was performed using Amazon Mechanical Turk. In this experiment, `${measure}` = population, `${place}` = {India, China, United States}, and `${time}` = {1950, 2010}. The results contained between 7 and 10 responses from multiple workers for each combination of place and time. By taking the mode (most frequently occurring value) of the worker submissions for each combination of place and time, the following table was generated:

In the table shown in figure 13, each row represents an observation within the data cube. The values in the Place column refer to members of the Space dimension. The values in the Time column refer to members of the Time dimension. The values in the Population column assign numeric values to

Place	Time	Population	Source URL
India	1950	369880000	www.geohive.com/earth/population3.aspx
China	1950	563000000	geography.about.com/od/populationgeography/a/chinapopulation.htm
USA	1950	150697361	en.wikipedia.org/wiki/1950_United_States_Census
India	2010	1150000000	www.indiaonlinepages.com/population/india-population.html
China	2010	1339724852	en.wikipedia.org/wiki/Demographics_of_China
USA	2010	308745538	en.wikipedia.org/wiki/United_States_Census

Figure 13: Initial results from an experiment in crowdsourcing public data using Amazon Mechanical Turk.

cells (combinations of Space and Time members) for the Population measure. This initial result demonstrates the feasibility of crowdsourced data collection for the UDC.

6 Data Cube Visualization

The purpose of introducing novel data structures and algorithms for data cube integration is to provide a foundation for development of interactive data visualization software. Data visualization theory (the relationships between data, graphics and perception) has been explored by Bertin [9], Wilkinson [120], Mackinlay [76] and others. Visualization of homogeneous data cubes has also been explored by Stolte et al. [108], Cuzzocrea et al. [26], and others. In the sections that follow, the possibilities for interactive visualizations based on the novel data cube integration framework are explored.

6.1 Data Cubes and Visualization Theory

Data cubes map well to data visualization. Bertin developed a significant part of his visualization theory based on data tables and the various kinds of fields (also referred to as attributes or columns) that one may encounter [9]. Dimensions and measures of data cubes, when projected into a table via querying, produce fields that correspond to the types of fields identified by Bertin. Based on an understanding of this correspondence, visualizations can be understood in terms of the data cube structures they are capable of representing.

The kinds of data table fields identified by Bertin include Nominal, Ordered, and Quantitative. Nominal fields contain references to distinct categories that have no intrinsic ordering. For example, a column in a table

referring to World Countries may be considered Nominal. Ordered fields contain references to distinct categories that have an intrinsic ordering. For example a field contain the values "Small", "Medium", and "Large" may be considered Ordered. Quantitative fields contain numeric values. For example, a field containing the population of each country may be considered Quantitative. Note that any Quantitative field may be converted into an Ordered field by binning.

According to Bertin, the appropriate visual encoding of a given field in a data table is determined by whether it is Nominal, Ordered or Quantitative (from Semiology of Graphics [9] p. 69). The planar X and Y dimensions are the most powerful in that they are capable of representing any kind of field. Size is also capable of representing any kind of field. Value (also called luminance or brightness) is capable of precisely representing only Nominal and Ordered fields. Texture (variation in the pattern used to fill visual marks) is capable of representing Nominal or Ordered fields. Color, Orientation and Shape are only capable of representing Nominal fields.

Visualization of trees is considered in another part of Bertin's theory (as a subset of networks) and has been explored in depth in visualization literature [46] [101] [98] [6]. The primary means of tree visualization include node-link diagrams, nested shapes, adjacent shapes, indented lists, and matrix representations. Node-link diagrams of trees have separate marks for each node, and each node is connected by a line. Nested shape tree visualizations such as TreeMaps or nested circles use containment to represent the tree

structure. Tree visualizations using adjacent shapes such as icicle plots or tree rings (hierarchical pie charts) use adjacency to represent connections between nodes in the tree. Indented lists, as in a file system tree view or a table of contents outline, use indentation level and linear ordering to present the tree. Matrix representations from graph theory can also be used to represent trees, however this is not common.

Data cube projections produce tables whose field types derive directly from the dimensions and measures involved. Data cube dimensions may be hierarchical, ordered, or unordered. These dimensions, when projected into tables, yield trees, Ordered fields, and Nominal fields, respectively. Data cube measures always project to Quantitative fields. Using these correspondences between data cubes and data types linked with visualization methods, a taxonomy of data cube visualizations can be formulated.

Based on an understanding of how data cube projections can map to visual encodings, visualization techniques can be characterized in terms of the data cube structure they are capable of representing. The relationship between data cube structures and visual encodings provides the basis upon which generalized visualization tools can be developed. These generalized visualization tools can then be used to visualize any data cube projection that adheres to the limitations of the input data cube structure they are capable of representing. Examples of such reusable visualizations include bar chart, grouped bar chart, stacked bar chart, area plot, streamgraph, scatter plot, parallel coordinates, choropleth map, treemap, radial tree, and

icicle plot.

The proposed work involves precise characterization of the above mentioned visualizations in terms of the data cube structure they are capable of representing. This categorization provides the basis upon which interactions on the visualizations can be formulated.

6.2 Visualization Taxonomy

Visualization Technique	Data Cube Structure
Bar Chart	1D, 1M
Pie Chart	1D, 1M
Scatter Plot	1D, 2M
Stacked Area Chart	2D, 1M
Line Chart	2D, 1M
Choropleth Map	1D (Geo), 1M
Node-Link Tree	1D (Hierarchical), 0M
TreeMap	1D (Hierarchical), 1M
Circular TreeMap	1D (Hierarchical), 1M
Icicle Plot	1D (Hierarchical), 1M
Hierarchical Pie Chart	1D (Hierarchical), 1M
Parallel Coordinates	1D, nM
Pivot Table	nD, nM
Small Multiples	+1D or +2D

Figure 14: Our taxonomy of visualization techniques based on the data cube structures they are capable of representing.

We introduce a taxonomy of visualization techniques based on the data cube structures they are capable of representing, as shown in the table in figure 14. In the "Data Cube Structure" column, "D" stands for Dimension and "M" stands for Measure. The number preceding "D" or "M" is the

number of Dimensions or Measures that the visualization technique is capable of representing. When "n" is used instead of a number, it means an arbitrary number of Dimensions or Measures can be represented by the visualization. "Small Multiples" is considered here as a special case that can be applied to any existing visualization by replicating the visualization while changing the slice in

6.3 Data Set Index

We introduce an index of public data sets characterizing their data cube structure in terms of dimension and measure coverage. Figure 15 shows a preliminary index of public data sets showing data cube structure only in terms of dimensions, levels, and measures. Once data sets are imported into our framework, the construction of this index can be automated. Additional detail will be included such as the members covered for each dimension. This index can be used in conjunction with our visualization taxonomy to match data sets (and combinations thereof) with approproate visual representations.

6.4 Prototypes

As a first prototype, I implemented a timeline visualization of the United Nations Population Estimates data set (see figure 16). To create this visualization, I manually cleaned the data originally made available as an Excel

Data Set	Dimensions	Measures
UN Population	Space (Countries) Time (Years)	Population
US Census Population	Space (US States, US Counties) Time (Years)	Population
US Bureau of Labor Statistics Employment	Space (US States, US Counties) Time (Years, Quarters, Months) Industry (NAICS Hierarchy) Ownership (Private, Government)	Employment Average Income Total Wages
UN Millenium Development Goals	Space (Countries) Time (Years)	Employment Literacy Infant mortality ... many more
CIA World Factbook	Space (Countries) Time (Years)	Life Expectancy Median Age GDP Per Capita ... many more
CDC Causes of Death	Causes of Death (Hierarchy) Time (Years)	Percentage of Deaths
PEW Global Religious Landscape	Space (Countries) Time (Years) Religion	Population

Figure 15: A preliminary index of public data sets characterizing their data cube structure in terms of dimension and measure coverage.

file and exported it as a CSV file. D3.js was used to create the timeline visualization from the CSV file. This example was created in order to fully understand the steps involved in visualizing a real-world public data set. This implementation has some aspects that are hard-coded to the specific data set, however this implementation can serve as a starting point for developing the generalized data representation and visualization framework incrementally.

As a second prototype, I implemented a stacked area chart of mortality

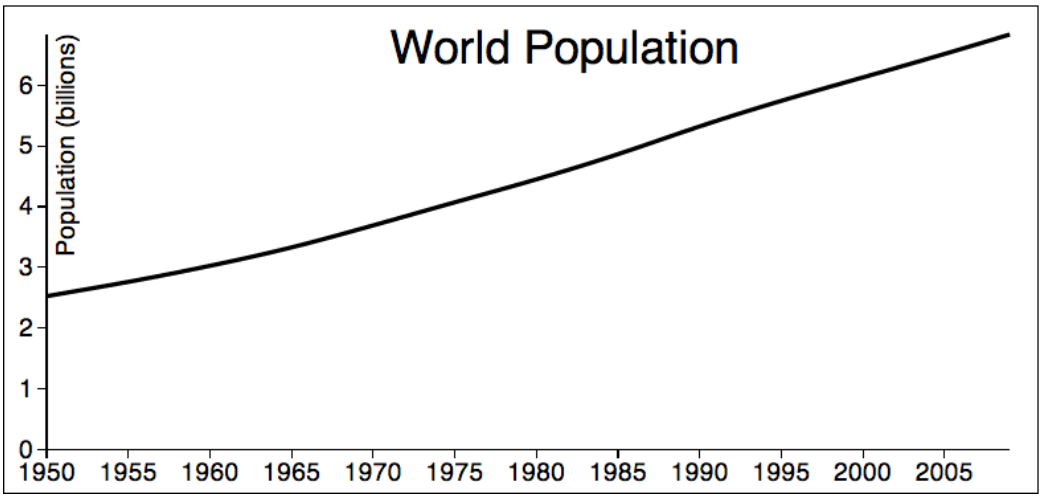


Figure 16: A timeline visualization of the United Nations Population Estimates data set. I implemented this visualization using D3.js and data downloaded from the United Nations Web site as a first prototype for visualizing public data.

data downloaded from the Centers for Disease Control, shown in figure 17. The data originally downloaded CSV (Comma Separated Value) file was not valid CSV, and had to be manually corrected using a text editor. The table contained a hierarchy of diseases, and all but the top-level disease categories were removed manually. Selecting the subtree of causes of death to include in the visualization is one example of a task that would be automated with our data representation framework. Next, a JavaScript program was written that pivoted the table from a format where each column was a year to a format where each row is a year, making the table usable by D3.js. This table contained an entry for "all causes", which was removed manually because it was not appropriate to include visualize.

The mortality data set was published using GitHub Pages using a JSON

(JavaScript Object Notation [25]) structure compatible with D3.js [14]. AMD (Asynchronous Module Definition) is a JavaScript pattern for publishing and consuming reusable modules across domains [90]. The mortality data set was published as an AMD module containing JSON data rather than as a CSV or JSON file in order to circumvent the same-origin policy. This allows any Web page to consume the data set, not only pages within the same domain. This method of publishing was chosen because it is a simple way to publish data publicly with zero cost (as GitHub Pages is a free service for Open Source code), longevity (as GitHub is less likely to go offline in the future than a private server relying on my bank account), and cross-domain availability (any page can load the module using an AMD loader such as `require.js`). This method of publishing data is also developer-friendly, as most modern developers are familiar with GitHub.

The mortality stacked area visualization highlights several issues that will be faced in general when visualzing data that must be addressed by our proposed data representation framework. The causes of death extracted from the raw data are sometimes too long to use in the visualization. For example "Symptoms, signs, and abnormal clinical and laboratory findings, not elsewhere classified" is too long, and could be simplified to "Unclassified conditions". In a data cube model, causes of death would be members in a dimension hierarchy. The labeling issue encountered in the mortality visualization indicates a need to support renaming of members for use as textual elements within visualizations. Since each label refers to a dimension

member which also may be a generally well-known concept, the labels on the visualization could, for example, be links to the Wikipedia pages about the various causes of death, such as Cardiovascular Disease. Also, there are 24 causes of death presented in this visualization using different colors, however D3 color scales only support up to 20 colors. This issue indicates that it may be useful to be able to automatically aggregate dimension members together as a new "Other" category in certain cases, or allow users to manually select only a subtree of a dimension hierarchy for visualization.

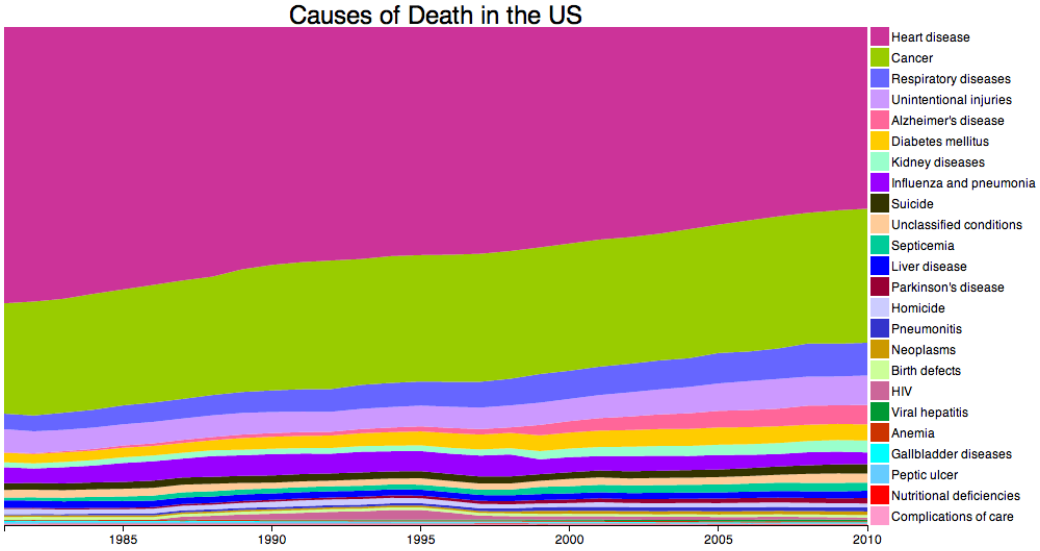


Figure 17: A second pass at a stacked area visualization of Mortality data from the US Centers for Disease Control. This version has 25 hand-picked distinguishable colors, a color legend to spread out labels, and shortened labels in some cases.

Figure 18 shows a sample of the raw data from which the hierarchy of causes of death must be gleaned. In this data the hierarchy is encoded as an indented tree. Two different characters are used as indentation charac-

ters, ASCII codes 32 (space) and 65533 (unknown character). The level of indentation does not use a consistent number of indentation characters per indentation level. For example, the indentation level jumps from 0 to 4 to 7 to 10 to 13. I implemented an algorithm that parses the tree structure from an indented list and outputs the tree in a the JSON tree data structure compatible with D3.js hierarchical layouts.

Several D3 example were drawn from to implement the cause of death tree visualization shown in figure 19, which uses the Reingold–Tilford "tidy" tree layout algorithm [95]. This visualization shows only the hierarchy of causes of death, but no numerical values associated with each node. Notice that the two causes of death that show the highest percentages in our stacked area visualization, Cancer and Cardiovascular Diseases, are the two nodes in the hierarchy that have the two largest subtrees of categorization.

The node-link tree visualization in figure 19 is an example of a visualization technique that can be applied to visualize dimension hierarchies in general. This implementation shows the structure of the hierarchy clearly, but has several drawbacks. Due to the size of the hierarchy, the inclusion of labels for all nodes necessitates small labels that are only legible at high resolution. When a hierarchy scales above certain thresholds of width and depth, this visualization becomes unwieldy, and labels must be truncated or omitted entirely. This is one example of scalability issues that must be addressed when developing general-purpose visualization technique implementations.

The pair of visualizations shown in figure 20 is an example of a visualiza-

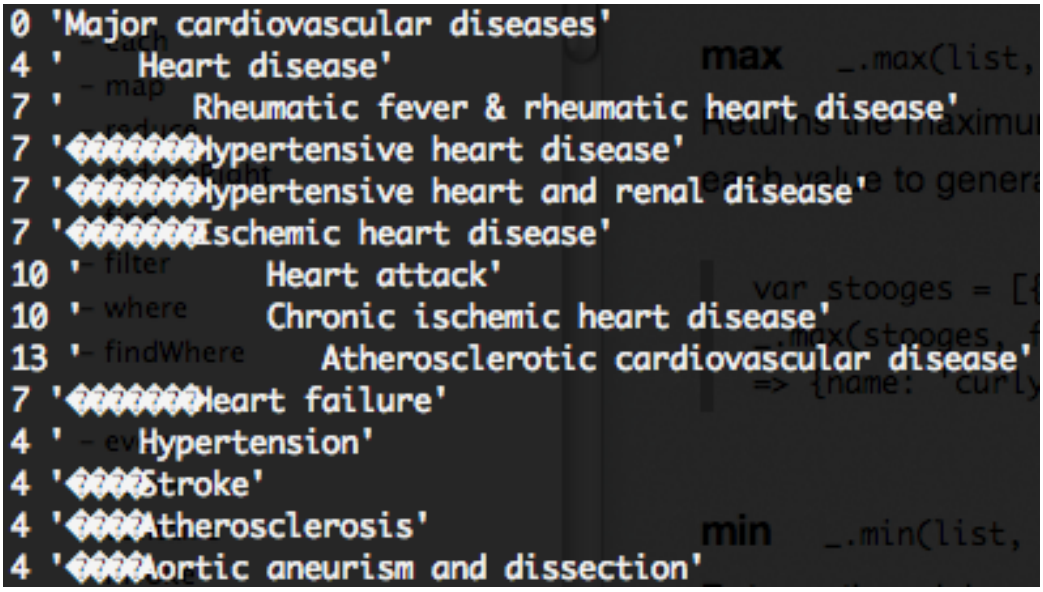


Figure 18: A portion of the raw data from the Centers for Disease Control encoding the hierarchy of causes of death. The number of indentation characters is shown on the left, and the content of the "Cause" field from the original CSV file is shown on the right in quotes. Note that there are two different indentation characters used, and the indentation level is not of a consistent multiple. This is one example of an unconventional format that must be parsed into a dimension hierarchy for use within our data representation framework.

tion dashboard with multiple linked views. The tree view shows a single level subtree. Black nodes have children, while white nodes do not. Clicking on a black node causes the tree to drill down into the subtree with the clicked node as its root. When this interaction is executed, the stacked area visualization is recomputed to show the new set of disease causes that correspond to the children of the newly selected tree node. In this way, the tree visualization provides interactions for drill down and roll up that define the slice of the data shown in the stacked area chart.

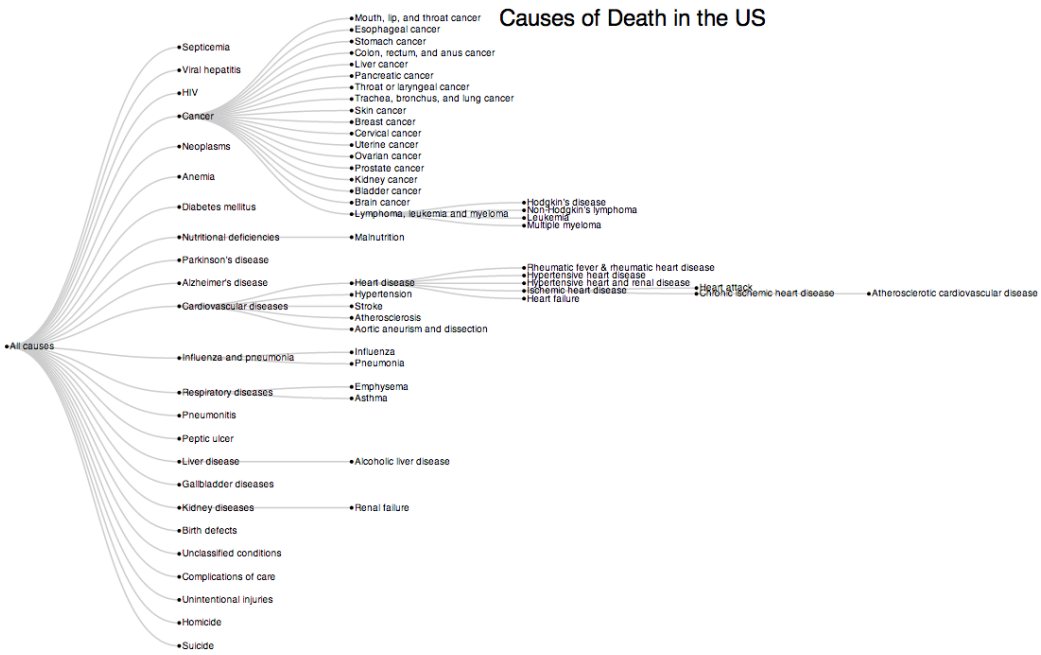


Figure 19: A tree visualization of data from the Centers for Disease Control showing the hierarchy of causes of death. This is one example of a visualization that shows the structure of a dimension hierarchy.

6.5 Interactive Data Cube Visualization Dashboards

Based on the data cube structure represented by a given visualization, interactions on the visualization can be related back to the data elements represented. For example, a rectangular selection on a scatter plot determines a subset of points. The subset of points, by inverting the visual encoding, defines a subset of the dimension members used to create the points. This subset of dimension members can be used as input to queries that define other visualizations or overlays on other visualizations. These kinds of linked interactions can be used for implementing well-known linked interactions such

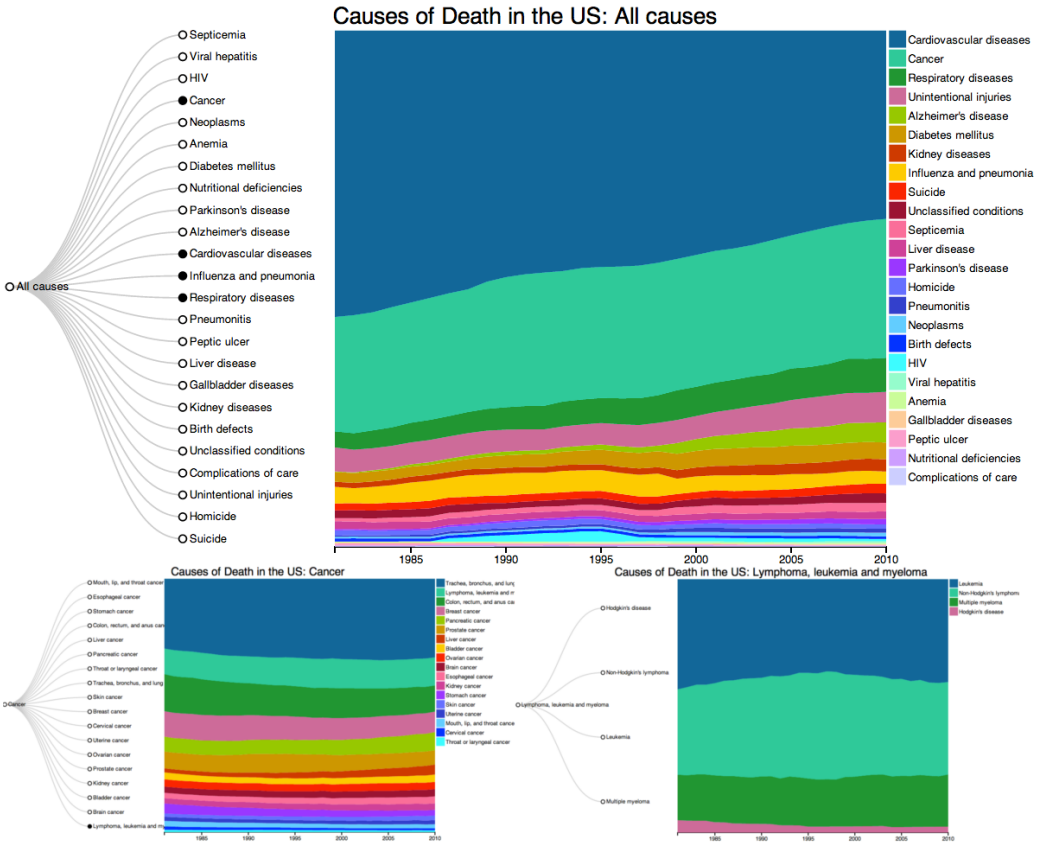


Figure 20: Cause of death visualization with two linked views. Navigating up and down the hierarchy by clicking on nodes changes the slice of data shown in the stacked area visualization. The top view shows the top-level causes of death. Clicking on the "Cancer" node yields the view on the bottom left, which shows types of cancer in the stacked area visualization. Further drilling down to "Lymphoma, leukemia and myeloma" yields the view on the bottom right.

as brushing, linking, probing (details on demand) and interactive filtering. The proposed work includes a characterization of the interaction techniques available in each of the reusable data visualizations explored. Based on the interaction techniques available and the types of query fragments they can

define, a framework for defining interactive visualization dashboards will be introduced.

Initial prototypes indicate a promising approach for representing and configuring interactive visualization dashboards. The configuration of a visualization dashboard can be represented by a tree structure corresponding to nested boxes, as well as visualization configuration options. The configuration options may contain references to elements of other visualizations on the dashboard. This is how, for example, zooming and panning on a map can be configured to interactively drive the subset of data shown in a timeline. Figure 21 shows a simple dashboard layout configuration. Figure 22 shows a more advanced prototype including a map component.

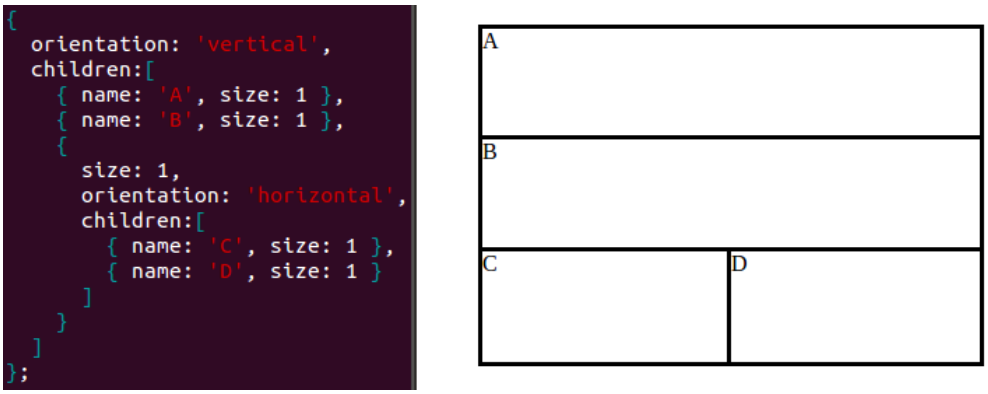


Figure 21: Basic dashboard layout configuration. The nested box layout on the right is determined by the configuration definition on the left.

The prototype dashboard layout system was used during a summer internship at Rapid7, a cybersecurity company, to create an interactive visualization dashboard with multiple linked views for analyzing corporate login

Figure 22: An example dashboard layout configuration with customized elements.

activity (see figure 23). Technologies used for visualization include D3.js, a visualization framework that uses SVG, and Leaflet.js, a framework for geographic maps. The map shows where users have logged into the network, aggregated geographically using the Leaflet MarkerCluster plugin and visualized using D3's Pie Chart layout. Black represents successful logins and blue represents failed logins. This industry application of our dashboard layout framework demonstrates its capability to define dashboards with multiple linked views. This framework has been released as Open Source and is available at `github.com/curran/dashboardScaffold`.

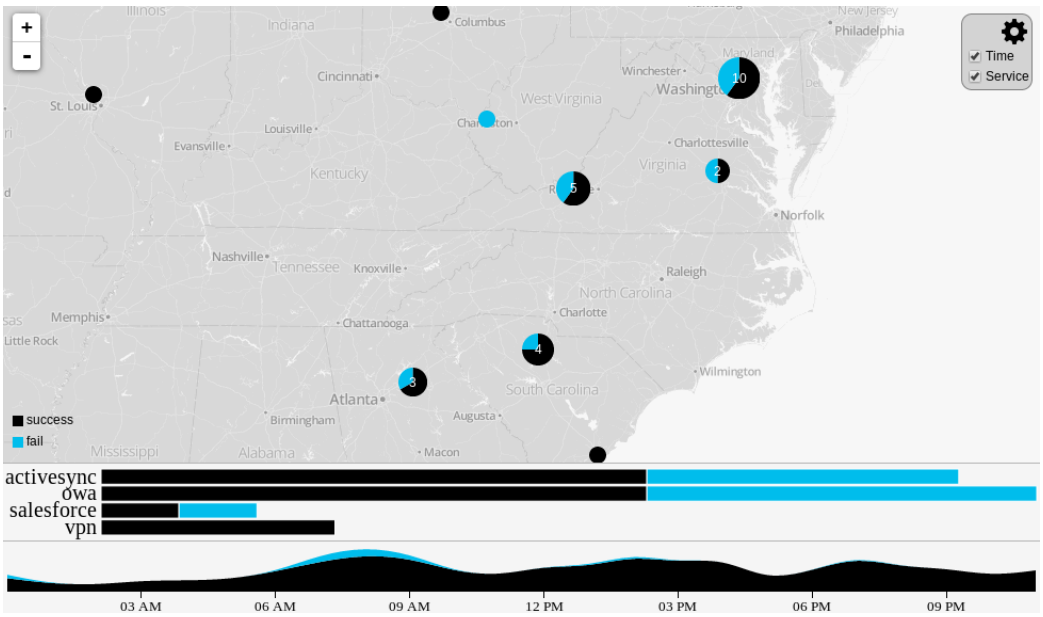


Figure 23: Our open source dashboard configuration framework in use in industry. This dashboard shows corporate login data, and is integrated into the Rapid7 product called UserInsight.

7 Plan of Action

In order to complete the proposed dissertation project, the following tasks will be completed:

- Introduce and refine a data representation framework consisting of novel algorithms and data structures capable of representing, integrating and querying multiple data cubes for the purpose of visualization.
- Survey a substantial sample of public data sets and characterize them in terms of data access mechanisms (user interfaces, data formats and data delivery protocols) and their coverage over universal dimensions

and measures.

- Survey a substantial sample of visualization techniques and characterize them in terms of the data cube structure they are capable of representing and the interactions they afford.

- Develop a conceptual framework for composing interactive data cube visualizations into dashboards with multiple linked views.

- For several of the public data sets surveyed, load the data into a proof-of-concept implementation of the data representation framework that uses Web technologies.

- For several of the visualization techniques surveyed, implement the visualization techniques in a generic manner, building on the novel data structures and algorithms introduced for data cube integration.

- Use the generalized techniques implemented to visualize the data sets imported into the framework.

- Generate an interactive visualization dashboard with multiple linked views that demonstrates using interactions in one visualization to define the slice of the data shown in another.

8 Expected Contributions

In summary, the expected contributions of this dissertation include the following:

- Novel data structures and algorithms for data cube integration. Existing formats, protocols and models only consider the case of homogeneous data cubes computed from a single source of relational data, and do not handle the case of integrating many pre-computed data cubes from multiple sources. Data integration has been well studied for relational data, but data integration methods have not been applied to OLAP cubes, which present unique challenges including management of dimension hierarchies and measures that are "universal", or shared by many data sources.

- A conceptual framework that links the integrated data cube structure with existing data visualization theory and techniques. Much work has been done concerning "Visual OLAP" [82], however the visualization approaches for OLAP cubes have not been extended to handle the rich heterogeneous structure introduced by integrating many data cubes from multiple sources.

- A framework for defining visualization dashboards with multiple linked views for interactively exploring integrated data cubes. Interactions between multiple views for OLAP cubes have been considered, but our

proposed integrated data cube structure affords a richer set of interactions that goes beyond traditional OLAP operations such as drill-down, roll-up, slice and dice.

These contributions will advance the field of computing and data visualization by enabling the development of tools for integrating and visualizing heterogeneous data sets in ways never before possible.

References

[1] Beyond 20/20. Product: Web data server. `http://www.beyond2020.com/index.php/data-solutions/products/web-data-server`, February 2014.

[2] US Central Intelligence Agency. Country comparison :: Gdp - real growth rate. `https://www.cia.gov/library/publications/the-world-factbook/rankorder/2003rank.html`, February 2014.

[3] Rakesh Agrawal, Ashish Gupta, and Sunita Sarawagi. Modeling multidimensional databases. In *Data Engineering, 1997. Proceedings. 13th International Conference on*, pages 232–243. IEEE, 1997.

[4] Akiko Aizawa and Keizo Oyama. A fast linkage detection scheme for multi-source information integration. In *Web Information Retrieval and Integration, 2005. WIRI'05. Proceedings. International Workshop on Challenges in*, pages 30–39. IEEE, 2005.

[5] Luc Anselin. Interactive techniques and exploratory spatial data analysis. *Geographical Information Systems: principles, techniques, management and applications*, 1:251–264, 1999.

[6] S Todd Barlow and Padraic Neville. A comparison of 2-d visualizations of hierarchies. In *infovis*, volume 1, page 131, 2001.

[7] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[8] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[9] Jacques Bertin. Semiology of graphics: diagrams, networks, maps. 1983.

[10] Chris Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web. *Retrieved June*, 20:2008, 2007.

[11] Christian Bizer and Richard Cyganiak. D2r server-publishing relational databases on the semantic web. In *5th international Semantic Web conference*, page 26, 2006.

[12] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[13] Markus Blaschka, Carsten Sapia, Gabriele Hofling, and Barbara Dinter. Finding your way through multidimensional data models. In *Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on*, pages 198–203. IEEE, 1998.

[14] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[15] Ricardo Cabello. Three. js. *URL: https://github. com/mrdoob/three. js*, 2013.

[16] Stuart K Card and Jock Mackinlay. The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 92–99. IEEE, 1997.

[17] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[18] Rick Cattell. Scalable sql and nosql data stores. *ACM SIGMOD Record*, 39(4):12–27, 2011.

[19] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and olap technology. *ACM Sigmod record*, 26(1):65–74, 1997.

[20] Ed Huai-hsin Chi. A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 69–75. IEEE, 2000.

[21] Ed Huai-hsin Chi and John T Riedl. An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 63–70. IEEE, 1998.

[22] Peter Christen. Febrl-: an open source data cleaning, deduplication and record linkage system with a graphical user interface. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1065–1068. ACM, 2008.

[23] James Clifford and Abdullah Uz Tansel. On an algebra for historical relational databases: two views. In *ACM SIGMOD Record*, volume 14, pages 247–265. ACM, 1985.

[24] Edgar F Codd, Sharon B Codd, and Clynch T Salley. Providing olap (on-line analytical processing) to user-analysts: An it mandate. *Codd and Date*, 32, 1993.

[25] Douglas Crockford. The application/json media type for javascript object notation (json). `http://tools.ietf.org/html/rfc4627`, 2006.

[26] Alfredo Cuzzocrea and Svetlana Mansmann. Olap visualization: models, issues, and techniques. *Encyclopedia of Data Warehousing and Mining,*, pages 1439–1446, 2009.

[27] Alfredo Cuzzocrea, Domenico Saccà, and Paolo Serafino. A hierarchy-driven compression technique for advanced olap visualization of multidimensional data cubes. In *Data Warehousing and Knowledge Discovery*, pages 106–119. Springer, 2006.

[28] Alfredo Cuzzocrea, Domenico Sacca, and Paolo Serafino. Semantics-aware advanced olap visualization of multidimensional data cubes. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(4):1–30, 2007.

[29] Richard Cyganiak, Simon Field, Arofan Gregory, Wolfgang Halb, and Jeni Tennison. Semantic statistics: Bringing together sdmx and scovo. *LDOW*, 628, 2010.

[30] Richard Cyganiak, Dave Reynolds, and Jeni Tennison. The rdf data cube vocabulary. Technical Report http://www.w3.org/TR/vocab-data-cube, W3C, December 2013.

[31] Erik Dahlström, Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, Jonathan Watt, Jon Ferraiolo, Fujisawa Jun, and Dean Jackson. Scalable vector graphics (svg) 1.1 (second edition). *W3C Recommendation*, 2011.

[32] Anindya Datta and Helen Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. *Decision Support Systems*, 27(3):289–301, 1999.

[33] Li Ding, Joshua Shinavier, Zhenning Shangguan, and Deborah L McGuinness. Sameas networks and beyond: analyzing deployment status and implications of owl: sameas in linked data. In *The Semantic Web–ISWC 2010*, pages 145–160. Springer, 2010.

[34] AnHai Doan, Pedro Domingos, and Alon Y Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *ACM Sigmod Record*, volume 30, pages 509–520. ACM, 2001.

[35] AnHai Doan, Pedro Domingos, and Alon Y Levy. Learning source description for data integration. In *WebDB (Informal Proceedings)*, pages 81–86, 2000.

[36] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[37] AnHai Doan and Alon Y Halevy. Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1):83, 2005.

[38] Stephen G Eick. Visualizing multi-dimensional data. *ACM SIGGRAPH computer graphics*, 34(1):61–67, 2000.

[39] Mohamed G Elfeky, Vassilios S Verykios, and Ahmed K Elmagarmid. Tailor: A record linkage toolbox. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 17–28. IEEE, 2002.

[40] Jérôme Euzenat, Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*. Springer, 2007.

[41] Ronald Fagin, Phokion G Kolaitis, Renée J Miller, and Lucian Popa. Data exchange: Semantics and query answering. In *Database Theory—ICDT 2003*, pages 207–224. Springer, 2003.

[42] US Centers for Disease Control. National vital statistics system. `http://205.207.175.93/Vitalstats/TableViewer/tableView.aspx`, February 2014.

[43] National Science Foundation. Science and engineering indicators 2012 state data tool. `http://www.nsf.gov/statistics/seind12/c8/interactive/map.cfm?table=31&year=2009`, February 2014.

[44] Steve Fulton and Jeff Fulton. *HTML5 Canvas*. O'Reilly Media, 2013.

[45] Gapminder. Data in gapminder world. `http://www.gapminder.org/data/`, February 2014.

[46] Martin Graham and Jessie Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

[47] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997.

[48] Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford. Record linkage: Current practice and future directions. *CSIRO Mathematical and Information Sciences Technical Report*, 3:83, 2003.

[49] Marc Gyssens and Laks VS Lakshmanan. A foundation for multi-dimensional databases. In *VLDB*, volume 97, pages 106–115, 1997.

[50] Alon Halevy, Anand Rajaraman, and Joann Ordille. Data integration: the teenage years. In *Proceedings of the 32nd international conference on Very large data bases*, pages 9–16. VLDB Endowment, 2006.

[51] Harry Halpin, Patrick J Hayes, James P McCusker, Deborah L McGuinness, and Henry S Thompson. When owl: sameas isn't the same: An analysis of identity in linked data. In *The Semantic Web–ISWC 2010*, pages 305–320. Springer, 2010.

[52] Pat Hanrahan, Chris Stolte, and Jock Mackinlay. visual analysis for everyone. *Tableau White paper*, 4, 2007.

[53] David C Hay. Making data models readable. *Information Systems Management*, 15:21–33, 1998.

[54] Bin He and Kevin Chen-Chuan Chang. Statistical schema matching across web query interfaces. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 217–228. ACM, 2003.

[55] Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, and Silvia Pfeiffer. Html5: A vocabulary and associated apis for html and xhtml. *W3C Working Draft edition*, 2013.

[56] CD'Arcy J Holman, A John Bass, Ian L Rouse, and Michael ST Hobbs. Population-based linkage of health records in western australia: development of a health services research linked database. *Australian and New Zealand journal of public health*, 23(5):453–459, 1999.

[57] Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[58] Matthew A Jaro. Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7):491–498, 1995.

[59] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31, 2003.

[60] Benedikt Kämpgen and Andreas Harth. Transforming statistical linked data for use in olap systems. In *Proceedings of the 7th international conference on Semantic systems*, pages 33–40. ACM, 2011.

[61] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *PART 5———Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 3363–3372. ACM, 2011.

[62] Jaewoo Kang and Jeffrey F Naughton. On schema matching with opaque column names and data values. In *SIGMOD Conference*, pages 205–216, 2003.

[63] Ralph Kimball. *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses*. Wiley. com, 1998.

[64] Lukasz Kokoszkiewicz, Julia ANDREEVA, Ivan Antoniev DZHUNOV, Edward KARAVAKIS, Massimo LAMANNA, Jakub MOSCICKI, and Laura SARGSYAN. hbrowse-generic framework for hierarchical data visualization. In *Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference*, 2012.

[65] Nick Koudas, Sunita Sarawagi, and Divesh Srivastava. Record linkage: similarity measures and algorithms. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 802–803. ACM, 2006.

[66] SD Kuznetsov and Yu A Kudryavtsev. A mathematical model of the olap cubes. *Programming and Computer Software*, 35(5):257–265, 2009.

[67] Neal Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2):12–14, 2010.

[68] Hing-Yan Lee and Hwee-Leng Ong. A new visualisation technique for knowledge discovery in olap. In *KDOOD/TDOOD*, pages 23–25. Citeseer, 1995.

[69] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.

[70] Chang Li and X Sean Wang. A data model for supporting on-line analytical processing. In *Proceedings of the fifth international conference on Information and knowledge management*, pages 81–88. ACM, 1996.

[71] Lauro Lins, James T Klosowski, and Carlos Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2456–2465, 2013.

[72] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. *Eurovis*, 2013.

[73] Vanessa Lopez, Spyros Kotoulas, Marco Luca Sbodio, Martin Stephenson, Aris Gkoulalas-Divanis, and Pól Mac Aonghusa. Queriocity: A linked data platform for urban information management. In *The Semantic Web–ISWC 2012*, pages 148–163. Springer, 2012.

[74] Miltiadis D Lytras and Roberto García. Semantic web applications: a framework for industry and business exploitation–what is needed for the adoption of the semantic web from the market and industry. *International Journal of Knowledge and Learning*, 4(1):93–108, 2008.

[75] Fadi Maali, Richard Cyganiak, and Vassilios Peristeras. A publishing pipeline for linked government data. In *The Semantic Web: Research and Applications*, pages 778–792. Springer, 2012.

[76] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics (TOG)*, 5(2):110–141, 1986.

[77] Jayant Madhavan, Philip A Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *VLDB*, volume 1, pages 49–58, 2001.

[78] Jayant Madhavan, S Jeffery, Shirley Cohen, X Dong, David Ko, Cong Yu, and Alon Halevy. Web-scale data integration: You can only afford to pay as you go. In *Proceedings of CIDR*, pages 342–350, 2007.

[79] Andreas S Maniatis, Panos Vassiliadis, Spiros Skiadopoulos, and Yannis Vassiliou. Advanced visualization for olap. In *Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 9–16. ACM, 2003.

[80] Svetlana Mansmann and Marc H Scholl. *Extending visual OLAP for handling irregular dimensional hierarchies*. Springer, 2006.

[81] Svetlana Mansmann and Marc H Scholl. Exploring olap aggregates with hierarchical visualization techniques. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1067–1073. ACM, 2007.

[82] Svetlana Mansmann and Marc H Scholl. Visual olap: A new paradigm for exploring multidimensional aggregates. In *Proc. of IADIS Int'l Conf. on Computer Graphics and Visualization (CGV)*, pages 59–66, 2008.

[83] Kouichi Matsuda and Rodger Lea. *WebGL programming guide: interactive 3D graphics programming with WebGL*. Pearson Education, 2013.

[84] Tova Milo and Sagit Zohar. Using schema matching to simplify heterogeneous data translation. In *VLDB*, volume 98, pages 24–27. Citeseer, 1998.

[85] United Nations. Millenium development goals indicators; country level data. `http://unstats.un.org/unsd/mdg/Data.aspx`, February 2014.

[86] United Nations. World population prospects: The 2012 revision. `http://esa.un.org/unpd/wpp/Excel-Data/population.htm`, February 2014.

[87] Natalya F Noy. Semantic integration: a survey of ontology-based approaches. *ACM Sigmod Record*, 33(4):65–70, 2004.

[88] Natalya F Noy. Ontology mapping. In *Handbook on ontologies*, pages 573–590. Springer, 2009.

[89] Natalya F Noy and Mark A Musen. The prompt suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[90] Addy Osmani. *Learning JavaScript Design Patterns*. " O'Reilly Media, Inc.", 2012.

[91] William Playfair. Commercial and political atlas: Representing, by copper-plate charts, the progress of the commerce, revenues, expenditure, and debts of england, during the whole of the eighteenth century. *London: Corry*, 1786.

[92] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *The Semantic Web: Research and Applications*, pages 524–538. Springer, 2008.

[93] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[94] Raghu Ramakrishnan and Johannes Gehrke. *Database management systems*. Osborne/McGraw-Hill, 2000.

[95] Edward M Reingold and John S. Tilford. Tidier drawings of trees. *Software Engineering, IEEE Transactions on*, (2):223–228, 1981.

[96] Hans Rosling, Rönnlund A Rosling, and Ola Rosling. New software brings statistics beyond the eye. *Statistics, Knowledge and Policy: Key Indicators to Inform Decision Making. Paris, France: OECD Publishing*, pages 522–530, 2005.

[97] Percy E Salas, Michael Martin, Fernando Maia Da Mota, Karin Breitman, Sören Auer, and Marco A Casanova. Publishing statistical data on the web. In *Proceedings of 6th International IEEE Conference on Semantic Computing*, IEEE 2012, Palermo, Italy, 2012. IEEE.

[98] H-J Schulz. Treevis. net: A tree visualization reference. *Computer Graphics and Applications, IEEE*, 31(6):11–15, 2011.

[99] Mathew Scotch, Bambang Parmanto, and Valerie Monaco. Usability evaluation of the spatial olap visualization and analysis tool (sovat). *Journal of Usability Studies*, 2(2):76–95, 2007.

[100] Matthew Scotch and Bambang Parmanto. Sovat: Spatial olap visualization and analysis tool. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 142b–142b. IEEE, 2005.

[101] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG)*, 11(1):92–99, 1992.

[102] Ben Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6):70–77, 1994.

[103] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[104] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. In *Journal on Data Semantics IV*, pages 146–171. Springer, 2005.

[105] Mark Sifer. A visual interface technique for exploring olap data with coordinated dimension hierarchies. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 532–535. ACM, 2003.

[106] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):52–65, 2002.

[107] Chris Stolte, Diane Tang, and Pat Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *KDD*, volume 2, pages 112–122. Citeseer, 2002.

[108] Chris Stolte, Diane Tang, and Pat Hanrahan. Multiscale visualization using data cubes. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):176–187, 2003.

[109] Michael Stonebraker. Sql databases v. nosql databases. *Communications of the ACM*, 53(4):10–11, 2010.

[110] Kesaraporn Techapichetvanich and Amitava Datta. Interactive visualization for olap. In *Computational Science and Its Applications–ICCSA 2005*, pages 206–214. Springer, 2005.

[111] David P Tegarden. Business information visualization. *Communications of the AIS*, 1(1es):4, 1999.

[112] Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[113] Michael Uschold and Michael Gruninger. Ontologies and semantics for seamless connectivity. *ACM SIGMod Record*, 33(4):58–64, 2004.

[114] Muhammad Usman, Sohail Asghar, and Simon Fong. A conceptual model for combining enhanced olap and data mining systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 1958–1963. IEEE, 2009.

[115] Panos Vassiliadis. Modeling multidimensional databases, cubes and cube operations. In *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, pages 53–62. IEEE, 1998.

[116] Panos Vassiliadis and Timos Sellis. A survey of logical models for olap databases. *ACM Sigmod Record*, 28(4):64–69, 1999.

[117] Fernanda B Viegas, Martin Wattenberg, Frank Van Ham, Jesse Kriss, and Matt McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.

[118] Holger Wache, Thomas Voegele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 workshop: ontologies and information sharing*, volume 2001, pages 108–117. Citeseer, 2001.

[119] Matthew O Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94*, pages 326–333. IEEE Computer Society Press, 1994.

[120] Leland Wilkinson. *The grammar of graphics*. Springer, 2005.

[121] William E Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[122] William E Winkler. Overview of record linkage and current research directions. In *Bureau of the Census*. Citeseer, 2006.

[123] Ji Soo Yi, Youn ah Kang, John T Stasko, and Julie A Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.

[124] Patrick Ziegler and Klaus R Dittrich. Three decades of data integration-all problems solved? In *IFIP congress topical sessions*, pages 3–12. Springer, 2004.