ORNL/TM-2016/762

EFDW: Energy Finance Data warehouse

Authors
Sangkeun Matt Lee
Supriya Chinthavali
Claire Zeng
Stephen Hendrickson
Mallikarjun Shankar

November 30th, 2016

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2016/762

Computational Sciences & Engineering Division
Electrical & Electronics Systems Research Division

EFDW: Energy Finance Data Warehouse

Authors:
Sangkeun Lee (Oak Ridge National Laboratory),
Supriya Chinthavali (Oak Ridge National Laboratory),
Mallikarjun Shankar (Oak Ridge National Laboratory),
Claire Zeng (Department of Energy).
Stephen Hendrickson (Department of Energy)

Date Published: November 30th 2016

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

ACKNOWLEDGMENTS

We would like to thank Eric Hsieh (US DOE) and Hugh Chen (US DOE) for their review and input. This work was financed by the Office of Finance of the Energy Policy and System Analysis Office (EPSA-51) of the US Department of Energy.

ABSTRACT

1. INTRODUCTION

1.1 CONTEXT AND GOALS

The Office of Energy Policy and Systems Analysis's finance team (EPSA-50) requires a suite of automated applications that can extract specific data from a flexible data warehouse (where datasets characterizing energy-related finance, economics and markets are maintained and integrated), perform relevant operations and creatively visualize them to provide a better understanding of what policy options affect various operators/sectors of the electricity system. In addition, the underlying data warehouse should be structured in the most effective and efficient way so that it can become increasingly valuable over time.

Automatic Sankey generator displaying money flows from consumers (end users such as residential, commercial, etc.) to distribution, generation and transmission operators is an example of one such application. In addition to the data sets themselves, the data warehouse must be able to capture calculations and estimations analysts (either from EPSA or external) have made in cases where data has been unavailable.

This report describes the Energy Finance Data Warehouse (EFDW) framework that has been developed to accomplish the defined requirement above. We also specifically dive into the Sankey generator use-case scenario to explain the components of the EFDW framework and their roles. An excel-based data warehouse was used in the creation of the energy finance Sankey diagram and other detailed data finance visualizations to support energy policy analysis. The framework also captures the methodology, calculations and estimations analysts used for the calculation as well as relevant sources so newer analysts can build on work done previously.

1.2 PRELIMINARY VISUALIZATION AND OUTSTANDING ISSUES

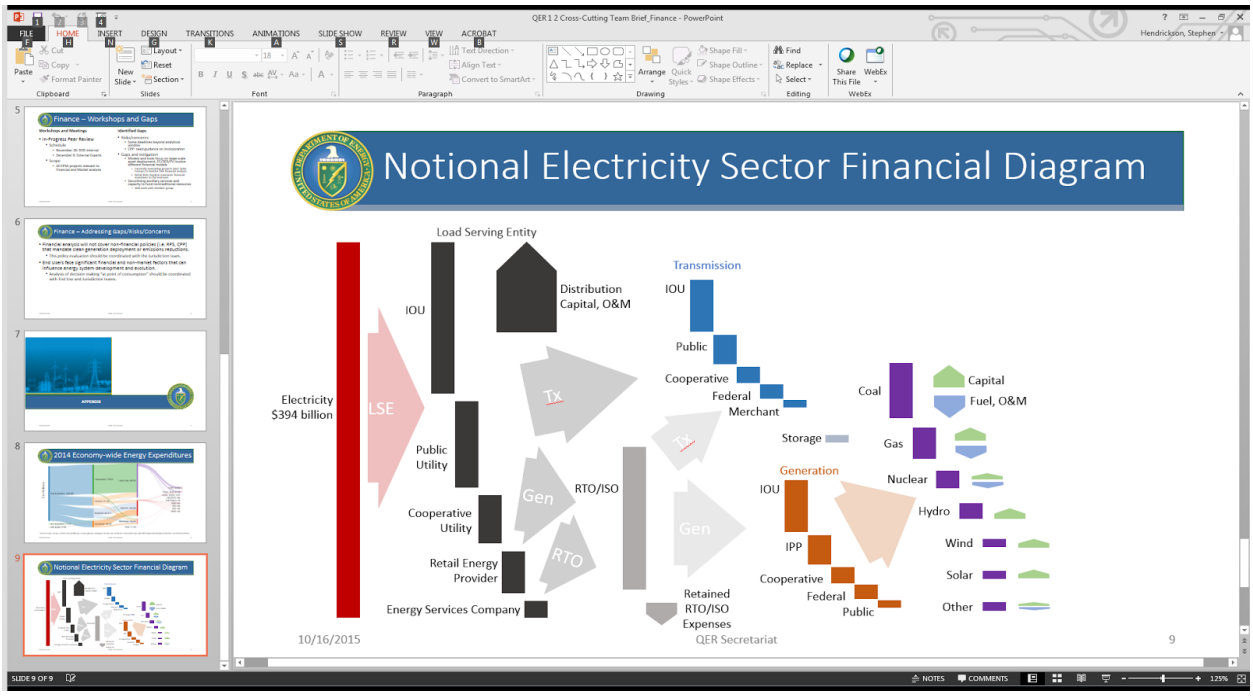


Figure 1 Preliminary visualization (Notional Electricity Sector Financial Diagram)

EPSA-50's goal was to show sizes of flows of revenue through the power sector. Such a visualization could help EPSA-50 understand market structures, compare sectors and actors, and identify points of potential policy applicability. Based on EPSA-50's previous use of the Sankey format for a tax revenues flow, EPSA-50 chose to use a Sankey format for power sector revenue flows.

The Sankey shows a simplified representation of the flows of revenues through the power sector value chain, from end users through load serving entities (LSE)/distribution, regional transmission organizations (RTO)/independent system operators (ISO) if applicable, transmission owners, generation owners, and fuel and operations and maintenance (O&M) providers. The financial flows mapped to physical flows of electricity as best as possible; for every dollar paid, a unit of electrical energy or delivery is received.

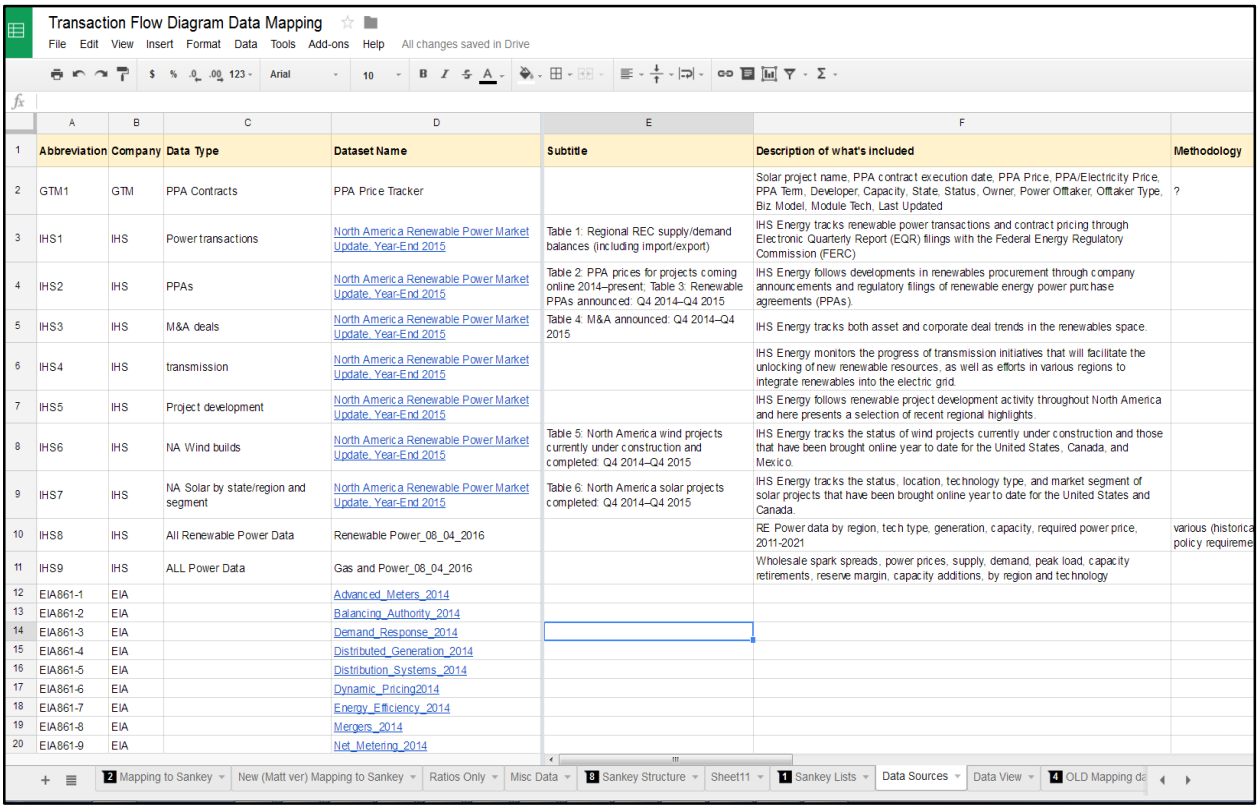
The structure of revenue flows required some unconventional Sankey structuring. For instance, some revenue flows from LSEs bypassed the RTO/ISO column and went straight to the next column. Additionally, the original notional electricity sector diagram (Figure 1) shows one Sankey, but for more accurate structuring, EPSA-50 decided to split it into two separate Sankey diagrams describing RTO/ISO and non-RTO/ISO regions, respectively.

The process of developing the Sankey required some simplifications of reality. For instance, to help illustrate the expense categories, the Sankey disaggregates functions that, from a business perspective, are not necessarily separate, such as IOU LSE and IOU Generation in vertically integrated markets. The LSE should be understood as a collector and disburser of payments from the end user, rather than a distinct entity from its distribution functions. Additionally, although revenue flows are shown from the LSE category to the Generation, Transmission, and Distribution categories, these are all within the same business for vertically integrated utilities.

1.3 DATA SOURCES AND USAGE CHALLENGES

EPSA's Office of Energy Finance and Incentives Analysis provided a list of data-sources described within the Transaction Flow Diagram Data Mapping spreadsheet. Initial datasets of relevance were chosen by the EPSA analyst to include in the tool.

The data source tab of this spreadsheet provides details about various attributes of the dataset such as name, data type, company, abbreviation, description of the data, frequency etc.



Transaction Flow Diagram Data Mapping

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E	F	
1	**Abbreviation**	**Company**	**Data Type**	**Dataset Name**	**Subtitle**	**Description of what's included**	**Methodology**
2	GTM1	GTM	PPA Contracts	PPA Price Tracker		Solar project name, PPA contract execution date, PPA Price, PPA/Electricity Price, PPA Term, Developer, Capacity, State, Status, Owner, Power Offtaker, Offtaker Type, Biz Model, Module Tech, Last Updated	?
3	IHS1	IHS	Power transactions	North America Renewable Power Market Update, Year-End 2015	Table 1: Regional REC supply/demand balances (including import/export)	IHS Energy tracks renewable power transactions and contract pricing through Electronic Quarterly Report (EQR) filings with the Federal Energy Regulatory Commission (FERC)	
4	IHS2	IHS	PPAs	North America Renewable Power Market Update, Year-End 2015	Table 2: PPA prices for projects coming online 2014–present; Table 3: Renewable PPAs announced: Q4 2014–Q4 2015	IHS Energy follows developments in renewables procurement through company announcements and regulatory filings of renewable energy power purchase agreements (PPAs).	
5	IHS3	IHS	M&A deals	North America Renewable Power Market Update, Year-End 2015	Table 4: M&A announced: Q4 2014–Q4 2015	IHS Energy tracks both asset and corporate deal trends in the renewables space.	
6	IHS4	IHS	transmission	North America Renewable Power Market Update, Year-End 2015		IHS Energy monitors the progress of transmission initiatives that will facilitate the unlocking of new renewable resources, as well as efforts in various regions to integrate renewables into the electric grid.	
7	IHS5	IHS	Project development	North America Renewable Power Market Update, Year-End 2015		IHS Energy follows renewable project development activity throughout North America and here presents a selection of recent regional highlights.	
8	IHS6	IHS	NA Wind builds	North America Renewable Power Market Update, Year-End 2015	Table 5: North America wind projects currently under construction and completed: Q4 2014–Q4 2015	IHS Energy tracks the status of wind projects currently under construction and those that have been brought online year to date for the United States, Canada, and Mexico.	
9	IHS7	IHS	NA Solar by state/region and segment	North America Renewable Power Market Update, Year-End 2015	Table 6: North America solar projects completed: Q4 2014–Q4 2015	IHS Energy tracks the status, location, technology type, and market segment of solar projects that have been brought online year to date for the United States and Canada.	
10	IHS8	IHS	All Renewable Power Data	Renewable Power_08_04_2016		RE Power data by region, tech type, generation, capacity, required power price, 2011-2021	various (historica
11	IHS9	IHS	ALL Power Data	Gas and Power_08_04_2016		Wholesale spark spreads, power prices, supply, demand, peak load, capacity retirements, reserve margin, capacity additions, by region and technology	
12	EIA861-1	EIA		Advanced_Meters_2014			
13	EIA861-2	EIA		Balancing_Authority_2014			
14	EIA861-3	EIA		Demand_Response_2014			
15	EIA861-4	EIA		Distributed_Generation_2014			
16	EIA861-5	EIA		Distribution_Systems_2014			
17	EIA861-6	EIA		Dynamic_Pricing2014			
18	EIA861-7	EIA		Energy_Efficiency_2014			
19	EIA861-8	EIA		Mergers_2014			
20	EIA861-9	EIA		Net_Metering_2014			

2 Mapping to Sankey | New (Matt ver) Mapping to Sankey | Ratios Only | Misc Data | 8 Sankey Structure | Sheet11 | 1 Sankey Lists | Data Sources | Data View | 4 OLD Mapping da

Figure 2 Examples of identified data sources

- EIA, Direct Federal Financial Interventions and Subsidies in Energy in Fiscal Year 2013 https://www.eia.gov/analysis/requests/subsidy/

- EIA, AEO, Total Energy Supply, Disposition and Price Summary (and Projections) ref2015.0219a A1,A2,A3 (multiply mmBTU x price projection for expenditure estimate)

- EIA, Table 1.5, Energy Consumption, Expenditures, and Emissions Indicators Estimates, 1949-2011

- historical data to complement AEO projections file: stb0105
- SNL, company level data

The process of finding datasets to fill the Sankey revealed that many revenues in the Sankey would be difficult to quantify. For instance, a major source of data, FERC, only has data for entities that are legally within its jurisdiction, excluding public power. As another example, FERC Form 1 and EIA Form 861, which were key datasets, are only required for utilities that meet a certain threshold of load served. Finally, non-power entities, which compose a small, but increasingly important part of the power sector, are not required to report data. Corporate procurements are generally kept confidential, and existing corporate procurement power purchase agreement (PPA) databases are often incomplete or inaccurate.

The flows were calculated bottom-up wherever possible. Calculations based on averages were used where bottom-up data was not available. Where bottom-up summation or calculations were not available, a ratio based on market share to distribute outflows was used.

2. RELATED WORK

2.1 SANKEY DIAGRAM VISUALIZATION TOOLS

Sankey diagrams are a type of flow diagram that is typically used to visualize cost transfers between processes. The capability of generating complex, interactive and customizable Sankey diagrams is one of the key objectives of the EFDW. Since there have been various tools to develop Sankey diagrams, we performed a lot of literature review to understand advantages and disadvantages of the existing tools. Some of the most popular software applications discovered during the review are explained below:

Software applications that focus on Sankey diagrams[1]:

2.1.1 Sankey Diagram Generator

Sankey Diagram Generator [1] is an online tool (http://sankey.csaladen.es). It supports various features such as self-loops, moving around nodes, changing opacity and the density of flows, which are necessary for creating sophisticated Sankey diagrams. A user needs to convert his/her own dataset into a JSON document following a specific structure to import dataset into the tool. This tool also supports explicitly assigning layer numbers to nodes in the diagram, which is not supported by most of the other tools. The downside of this tool is that the tool does not allow users to assign colors to nodes and flows.

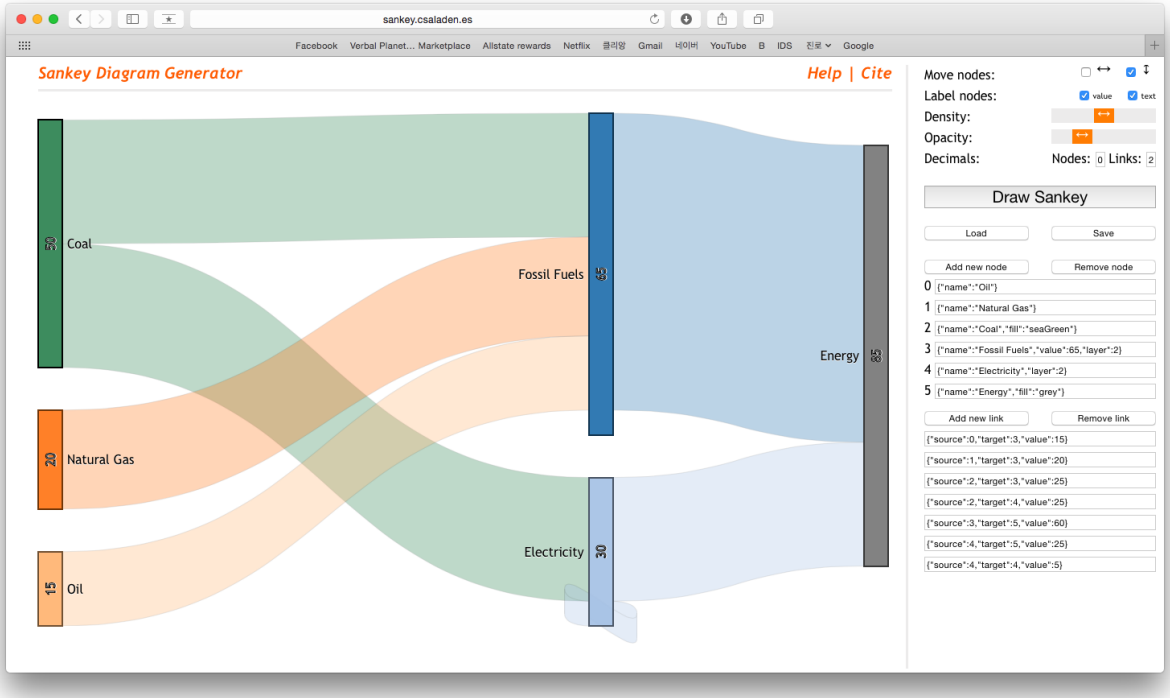


Figure 3 Screenshot of Sankey Diagram Generator main page

[1] http://www.it1me.com/learn?s=Sankey_diagram

http://www.sankey-diagrams.com/sankey-diagram-software/

2.1.2 SankeyMatic

SankeyMATIC [2] is another tool that is available on the web (http://sankeymatic.com). Like Sankey Diagram Generator, users need to prepare their datasets in a specific data format to import their dataset into the tool. It provides a web-based GUI and gives various options such as size, spacing, shape, label, etc. Unlike Sankey Diagram Generator, SankeyMATIC does not support assigning layers to the nodes, so users need to manually adjust the position of nodes, if layer positioning is necessary. However, this tool gives an option to users to assign colors to nodes and flows so the users can color code nodes and flows.

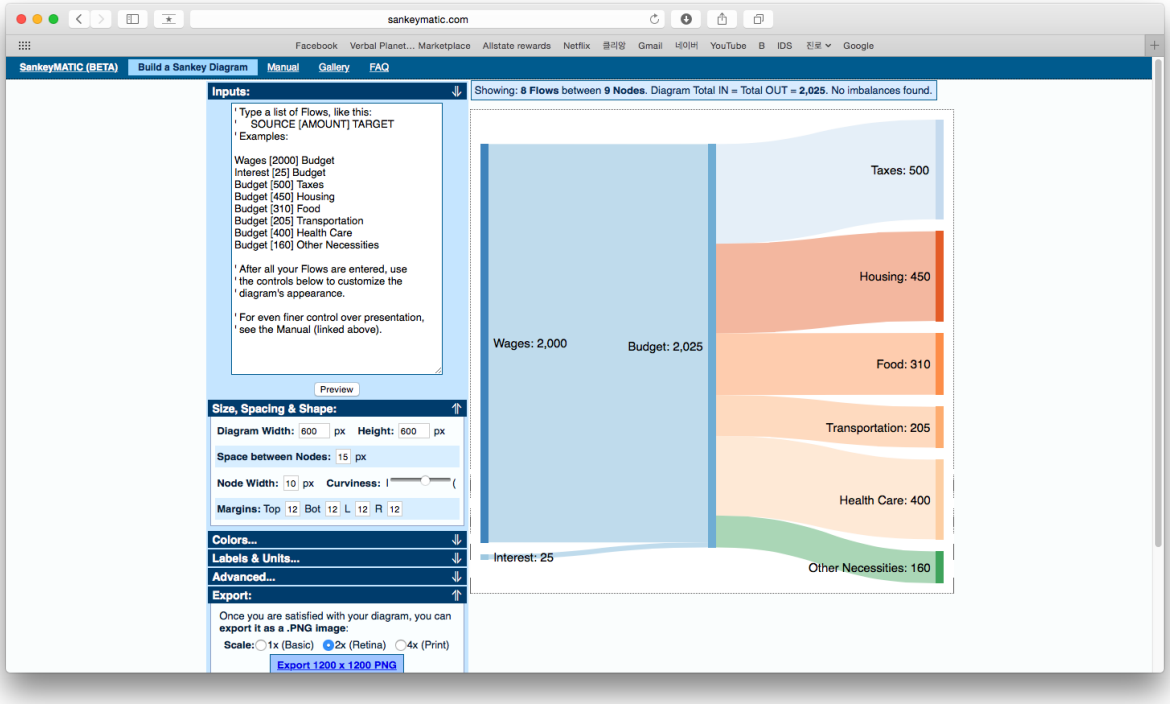


Figure 4 Screenshot of SankeyMATIC main page

2.1.3 Google Charts - Sankey Diagram

Google Charts [3] is a free tool that can generate various kinds of interactive charts including Sankey Diagram. To use Google Charts, users need to write JavaScript and embed the code in a HTML web-page; which can be a huge benefit in case of integrating with other web applications, but it can be also a disadvantage to most of users who have no programming experiences. Google Charts lacks some of important features such as assigning layers (levels) and positioning the nodes.

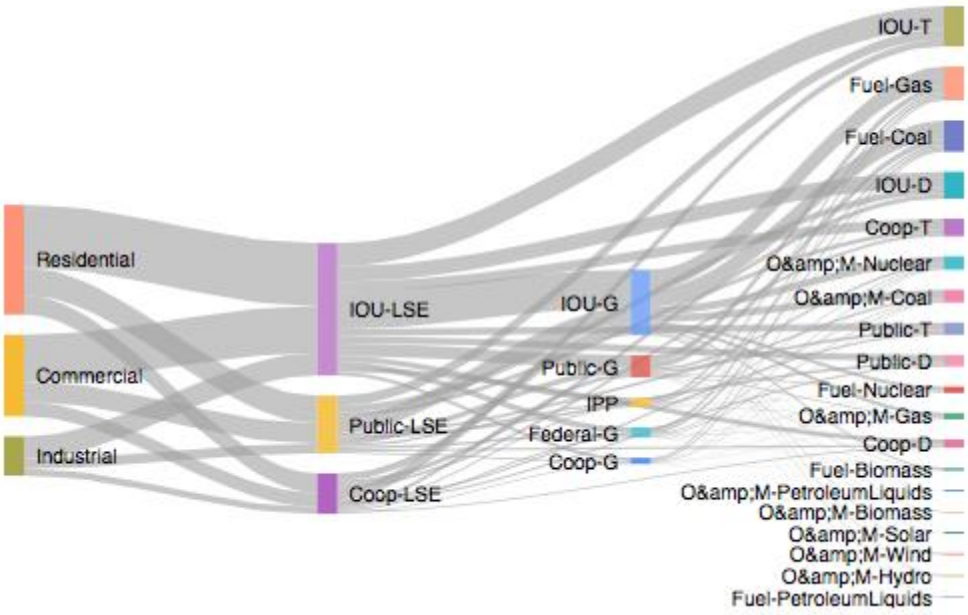


Figure 5 Example of Sankey Diagram generated using Google Charts

2.1.4 Python Library (matplotlib) for Making Sankey Diagrams

Matplotlib [4] (http://matplotlib.org), which is a python 2D plotting library, supports a wide range of plotting, and it also provides a module for creating Sankey diagrams. It is an advantage that the module allows users to set up a wide range of optional arguments (e.g., scale, unit, gap, radius, offset, margin), but users need to write a script to generate a Sankey diagram. The tool does not have any Graphical User Interface, so it can be very challenging for most of the users who are not familiar with programming.

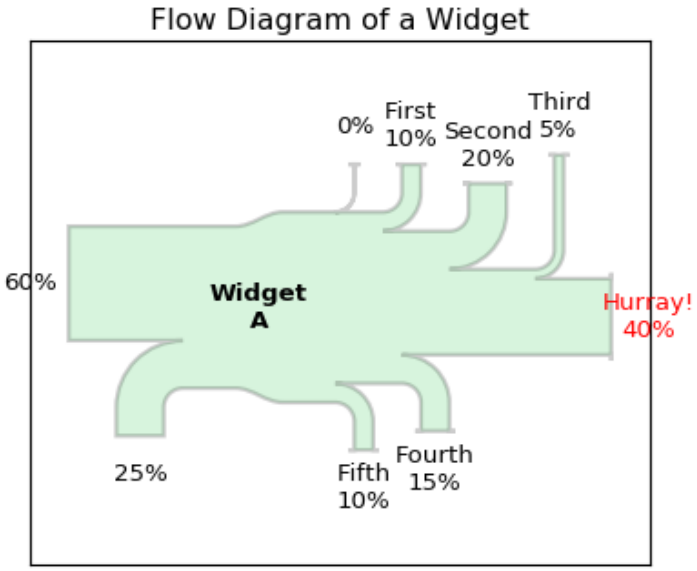


Figure 6 Example of Flow Diagram generated using matplotlib

2.2 VISUAL ANALYTIC TOOLS

2.2.1 Tableau Desktop/Server

Tableau Desktop/Server (http://www.tableau.com) is a commercial visual analytic software that can help users visualize data by connecting various types of data formats and databases. It supports various enterprise data sources such as Hadoop, cubes, AWS, etc., and can create various interactive visualizations and enable knowledge discovery by generating useful insights.

With its intuitive drag and drop interface, Tableau offers a lot of great features off the shelf visualization widgets that can be combined to generate interactive dashboards with drill-down views. However, Tableau does not natively support generating Sankey Diagram widget. There is a work-around (https://community.tableau.com/thread/152115) to build a Sankey Diagram with Tableau using Sigmoid function, but this approach comes with many restrictions. For instance, in the multi-level Sankey Diagram, the flow cannot skip levels, in other words, flows starting from level 0 are not allowed to directly go to level 2, and they must go to level 1. Also, the sum of amount in every layer should always sum up to 1.The data staging process also involves data-duplication which can be a significant overhead in terms of performance. This work-around may be useful for very simple Sankey Diagrams, but for most of the cases, Tableau is not suitable for generating sophisticated Sankey Diagrams for EFDW.

2.2.2 Infocaptor Enterprise

Infocaptor Enterprise is another web-based application which provides visual analytics. Like Tableau, it allows users to compose customized dashboards with various interactive visualizations. Unlike Tableau, Infocaptor natively supports Sankey diagram. Users can produce Sankey diagrams by drag and drop operations. Another advantage of Infocaptor is that it is possible to make the generated Sankey diagram interact with other kinds of visualizations or data widgets supported by the tool. For example, in a dashboard, we can make the dashboard show a corresponding data table when a node in the Sankey diagram is selected. Sankey diagram module of Infocaptor Enterprise can produce sophisticated Sankey diagrams, but it lacks features such as assigning levels to nodes, so users need to manually position nodes if assigning levels of nodes is needed. Also, the nodes and the edge flow colors cannot be configured or selected based on a criteria/attribute.

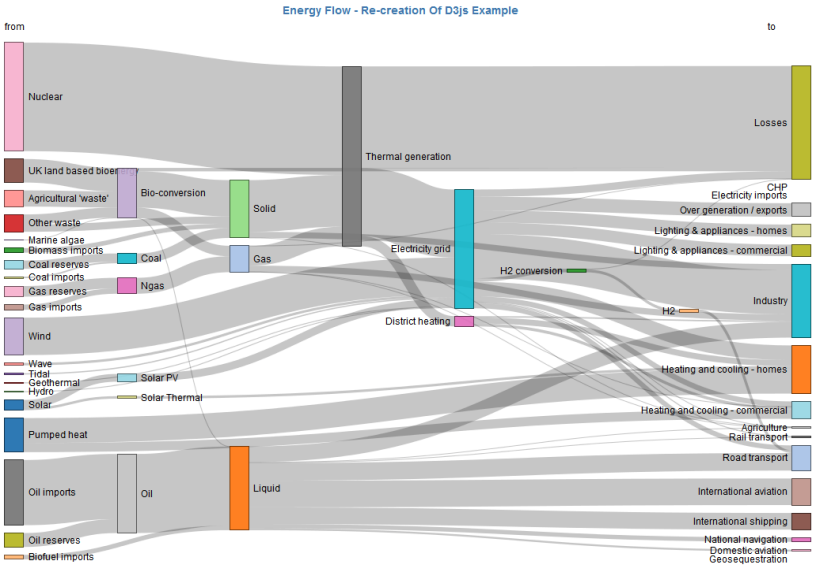


Figure 7 Example of Sankey Diagram generated by Infocaptor

2.3 VISUALIZATION TOOLS USED FOR THIS PROJECT

One of the main outputs from EFDW aims to present financial flows across the entire energy sector using Sankey diagrams and systematically track the provenance of the data and its relevant sources. Since none of the above discussed Sankey visualization and visual analytic tools offered all the desired features to represent the financial data, we used a combination of the tools described in the previous section to provide a comprehensive view. Infocaptor was used as the primary visual analytic application which in turn provides access to Tableau, SankeyMATIC, and Sankey Diagram Generator visualizations.

3. ENERGY FINANCE DATA WAREHOUSE (EFDW)

3.1 OVERVIEW

The Energy Finance Data Warehouse (EFDW) is a data warehouse where energy-related finance, economics, and market data can be maintained, integrated, expanded and analyzed so that it can become an increasingly valuable resource over time. EFDW aims to provide a way to systematically track methodology used for the calculation and estimation as well as relevant sources, so newer analysts can build on work done previously.

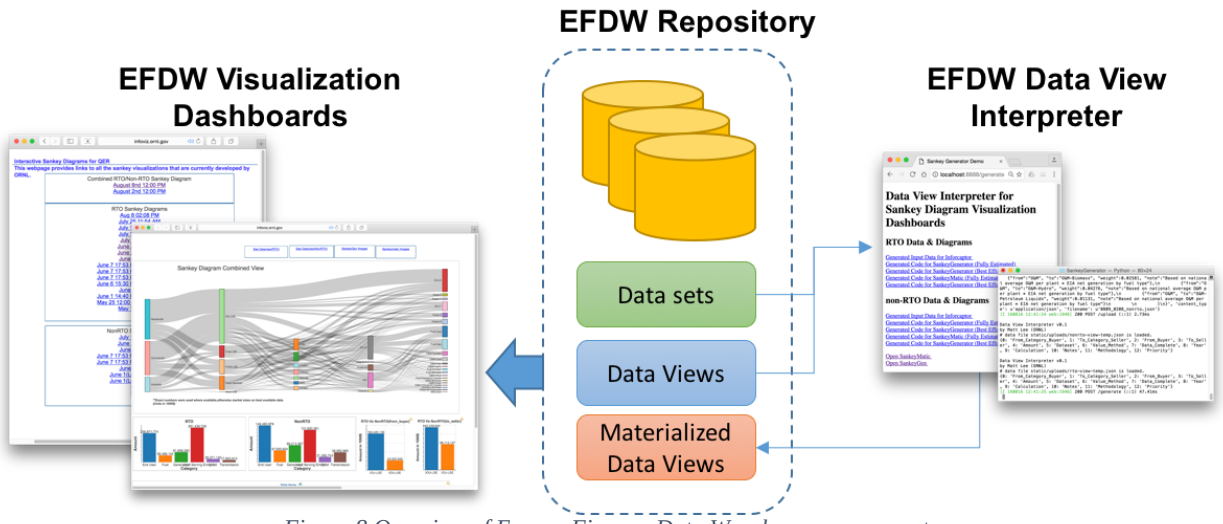


Figure 8 Overview of Energy Finance Data Warehouse components

Figure 8 shows the overview of Energy Finance Data Warehouse components. EFDW is composed of three components that are *EFDW Repository*, *EFDW Data View Interpreter*, and *EFDW Visualization Dashboards.* In the following, we give a brief explanation of each component.

EFDW repository is a storage where energy-related data along with calculated data by an analyst (either from EPSA or external) are accumulated over time. The data sets are organized and indexed so that EPSA analyst can quickly identify what data sets are available for analyses. In addition, EFDW repository contains *data views*, where a data view is a definition of a virtual data set constructed from other data sets in the repository. Analysts can describe their needs in their data view definitions to define various virtual datasets that are useful for analytic purposes. For instance, a data view can describe an input data set for EFDW's Sankey diagram visualization dashboards, where the data view includes the related available data, structure of Sankey diagram, and proportions for computing unavailable data points. In Section 3.2, we describe the EFDW repository in detail.

Data views describe how to construct data sets as structured JSON documents; however, data views themselves cannot be directly utilized for analyses. *EFDW Data View Interpreter* is a software module that is responsible for parsing data views and materializing them in physical data sets so that data views can be exploited for various analyses. The materialized data sets are also stored in the EFDW repository. For a proof-of-concept, we developed EFDW Data View Interpreter for EFDW's Sankey diagram visualization dashboards. In Section 3.2, we explain the syntax of data views and the detailed implementation of the tool.

EFDW Visualization Dashboards are a set of front-end graphical user interfaces for analysts to analyze data in the EFDW repository. Like EFDW Data View Interpreter, we focused on Sankey diagram dashboards for a proof-of-concept. The Sankey diagram dashboards allow analysts to interact with the diagrams displaying the context around the data points that they are interested in. One of the advantages is that the dashboards are linked with data views; so, analysts can track the provenance of data. The details are described in Section 3.4.

3.2 EFDW REPOSITORY

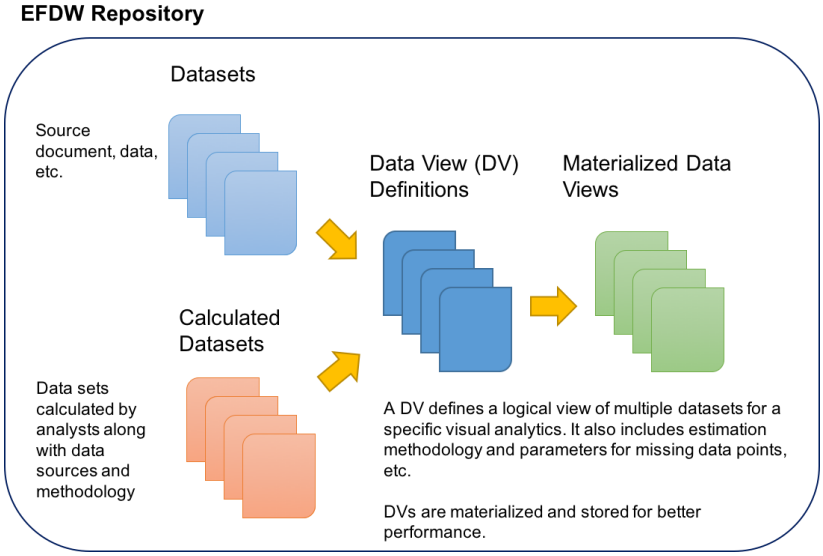


Figure 9 Various Data stored in the EFDW repository

Information stored in EFDW Repository would include, but not be limited to, data, metrics, and projections associated with company financials and physical assets; industry data in aggregate (such as capital expenditures per year, historical and projected); market data such as wholesale and retail electricity prices (such as those associated with the market analysis research effort); project finance data (such as those associated with the Project Finance Mapping Tool—capital costs by technology, power purchase agreement data); emissions data (to approximate the avoided emissions by asset type or region); and other economic data (such as interest rate or cost of capital projections) from various data sources. More specifically, the data stored in EFDW can be grouped into the following categories.

- (Generic) datasets - datasets that are imported from various other data sources without modifications. There can be a wide range of data formats; however, in this phase of the project, we focus on the tabular formatted structured data which is most energy-finance data sets.

- Calculated datasets - datasets that are created by EPSA analysts based on manual calculations or projections. The datasets in this category should contain the methodology and provenance of data used for calculation. We assume that analysts use tabular data representation like the generic data sets.

- Data views - data views are documents describing logical view of one of multiple data sets for a specific analytic purpose. They can be understood as a definition of a virtual data set that are constructed from other data sets and computations based on given parameters in the documents. Data views are represented as JSON (JavaScript Object Notation) files, which JSON is a lightweight data-interchange format. Detailed syntax of data view and examples will be presented in the Section 3.3.

- Materialized data views - data views only define how data sets can be constructed and data view themselves are not data sets that can be utilized for analysis. EFDW Data View Interpreter (Section 3.3.) is responsible for materializing data view documents in physical

data sets. Materialized data sets are represented and stored as CSV (comma separated values), which is a tabular data. Relationship between data views and materialized data sets are also maintained in the EFDW repository.

3.3 EFDW DATA VIEW INTERPRETER

EFDW Data View Interpreter is a software module which translates a data view file, which is a JSON document describing a virtual data set that can be constructed from data sources, into a CSV (Comma Separated Values), so that it can be used for various analytic purposes. For a proof-of-concept, we developed a EFDW data view interpreter focusing on specific visual analytics, which is interactive Sankey diagram. In Section 3.3.1., we explain the syntax of data view file and show how analysts can describe their needs for data set for Sankey diagram in a data view document.

3.3.1 Data View Definition for Sankey diagram

Sankey diagrams are a specific kind of flow diagram where the thickness of the arrows is shown proportionally to the flow amount, and they can nicely visualize energy cost transfers and transactions. Although many energy finance data sets are available in the EFDW, it is still challenging to visualize the data sets as a Sankey diagram for several reasons. First, Sankey diagrams require a very specific structure of input data. As Sankey diagrams are about the transaction flows; it is necessary to re-organize data in a flow-centric representation such as *From, To, Amount*. It is very unlikely that existing data sources are already organized that way. Second, there needs to be a way to deal with missing data points. Sankey diagrams make sense when there are no missing data points, as they intend to show overall picture of transaction flows across different categories, so missing data points can make the whole Sankey diagram structure semantically meaningless.

We envision that an EPSA analyst can define a data view which can construct a data set that can be used for Sankey diagram generation. A data view document for Sankey diagram include the following information:

- Data Source: Base data source that is used for data construction for a Sankey diagram
- General Information about the data view itself: Brief description of this data view (e.g., who created, title, etc.), Date when the file is created
- Sankey diagram structure: User can specify a Sankey diagram structure such as categories and subcategories, the list of categories that become the last layer, allowed flows, etc.
- Parameters for estimating missing data points: If there are missing data points, EFDW Data View Interpreter refers to these parameters to estimate and create missing data points.

More specifically, a data view document for Sankey diagram is a standard JSON file, and it is structured as follows.

```
{
    "data_source": location of base data file,
    "description": brief description of this data view,
    "date": date of last update,
    "Categories”: list of categories,
    "Layers”: layers for categories,
    "unit": unit of transaction amount,
    "first_box:(subcategory_name)" : the size of first box that will be distributed across
diagram when estimating missing data points,
    ...,
```

```
    "def_sub_category:(category_name)" : list of subcategories,
    ...,

    "allowed_flow": [
        {"from":"cat:(category_name)","to":"cat:(category_name)", "weight":proportion
distributed from source to target,"note":brief explanation how the proportion was decided},
        ...
    ]

    "cat_itm_weight" : [
        {"from":"(category_name)", "to":"(subcategory_name)", "weight":proportion distributed
from source to target, "note":brief explanation how the proportion was decided},
        ...
    ]
}
```

- **data_source** attribute is for the location of base data for Sankey diagram. The base data must be a CSV file stored in the EFDW repository having the schema described in Table 1. Note that the base data may not have the complete data points for a Sankey diagram.
- **description**, **date** attributes are for brief description of the data view. Users can specify the name of categories (e.g., "End User","Load Serving Entity","Generation","Transmission", etc.) using the categories attribute.
- The **layer** attribute describes in which layer of the Sankey diagram each of the defined categories should be visualized.
- The attribute **unit** specifies the unit of data visualized in the diagram. I.e., the amount in the base data is divided by the value of the **unit** attribute.
- There need to be the same number of **first_box:(subcategory_name)** attributes as the number of categories defined above. (subcategory_name) needs to be replaced with the actual subcategory names in the first layer.
- The value of these attributes is used for estimating the missing data points. There also should be the same number of **def_sub_category:(category_name)** attributes as the number of defined categories, where (category_name) is replaced with the actual category names. Each of these attributes defines the sub categories for each category; for instance, for the category Load Serving Entity, the attribute **def_sub_category:Load Serving Entity** will be defined as ["IOU-LSE","Public-LSE","Coop-LSE"].
- **allowed_flow** is an important attribute that defines the allowed transaction flows in a Sankey diagram. The value of the attribute is a list of JSON objects structured as `{"from":"cat:(category_name)","to":"cat:(category_name)", "weight":proportion distributed from source to target,"note":brief explanation how the proportion was decided}`. (category_name) must be one of the defined category names. The total weights from the same source category must sum up to 1. Data points describing transactions from the subcategory x to subcategory y will be included in the generated data only if the transaction from category X to category Y is allowed.
- The attribute **cat_itm_weight** defines the proportion for distributing incoming amount for a category to its own subcategories in case estimating the value.

Table 1 Schema of base data for Sankey diagram data view

Column Name	Description

From_Category_Buyer	Category of From_Buyer
To_Category_Seller	Category of To_Seller
From_Buyer	Subcategory name for transaction source (Buyer)
To_Seller	Subcategory name for transaction destination (Seller)
Amount	Transaction amount
Dataset	Dataset name where the data row originally came from
Value_Method	How this data point was achieved [Calculation: calculation by an analyst/Exact: exact number copied from an existing data set]
Data_Complete	Yes/No
Year	Year
Calculation	Equation used for analyst's calculation
Notes	Other description
Methodology	Methodology used for
Priority	In case there are multiple data points for the same transaction type with the same source and target, only one data point needs to have a value 'Y' for this column to specify which one will be mainly used for visualization.

3.3.2 Data View Interpreter for Sankey diagram

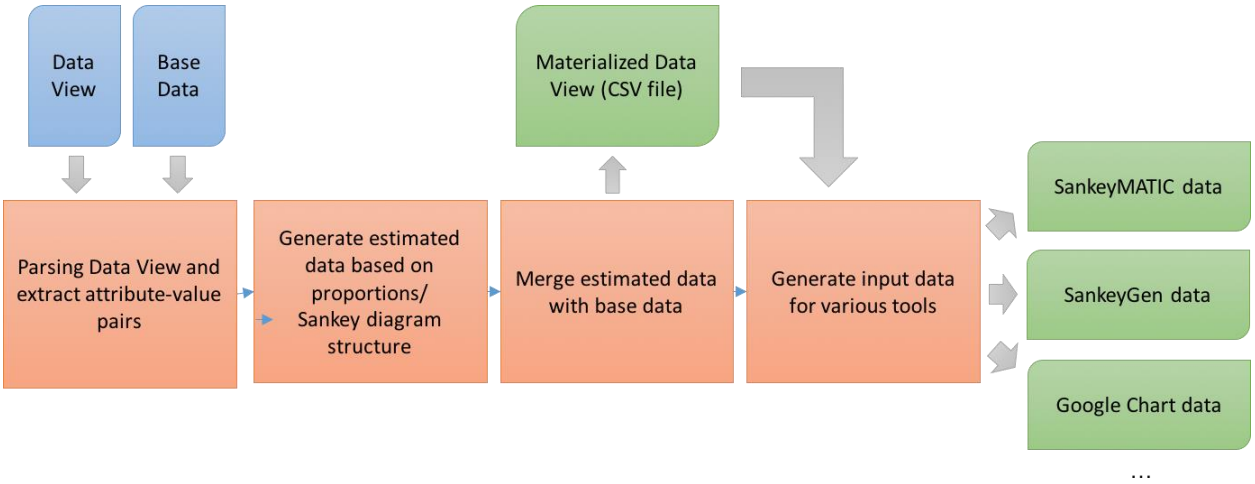


Figure 10 Data view interpretation process

Figure 10. Illustrates the process of data view document interpretation. First, data view file is parsed and attribute value pairs are extracted. Then the interpreter generates all the data points that are required for a Sankey diagram based on the defined Sankey diagram structure and the given proportions. Next, the generated data set is merged with the given base data to create the materialized data view file. In addition

to the generated tabular data, specifically for Sankey diagrams, we implemented one more step that generates input data for various tools. As discussed in Section 2, different existing Sankey diagram generators use different formats of input data. The data view interpreter allows users to use various Sankey diagram generator tools without having to manually convert the data formats from one to one.

The core of Data View Interpreter for Sankey diagram is implemented as a command line tool using Python 2.7, and it is intended to be compatible with Windows, Mac OSX, and Linux systems.

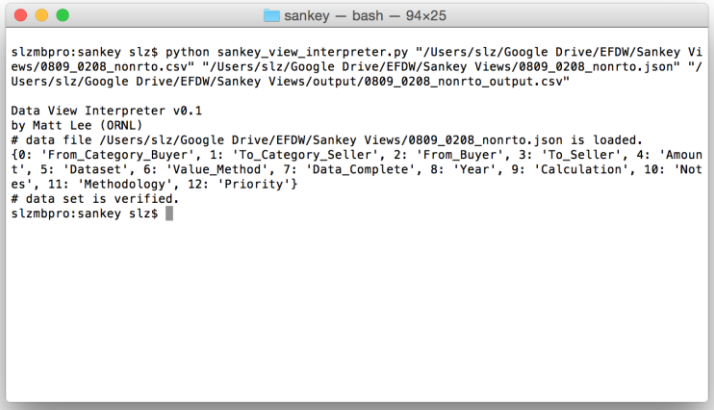


Figure 11 Screenshot of using the command line data view interpreter tool

Executing the Data View Interpreter is quite simple. A data view (a JSON file) and base data (a CSV file) needs to be prepared accordingly before the execution of the tool. Users can execute the data view interpreter by using the following command:

```
python sankey_view_interpreter.py (location_of_base_data)
(location_of_data_view_file) (location_of_output)
```

For instance, the following command

```
python sankey_view_interpreter.py 0805_base.csv 0805_dataview.json
0805_materialized.csv
```

will take the two files `0805_base.csv 0805_dataview.json` as inputs to create a file named `0805_materialized.csv`, which is the materialized data view file, in the same directory where the python script is located. The materialized view file not only contains the original base data points but also estimated data points that are computed based on the proportions defined in the data view file, and it can be imported to an Infocaptor dashboard for generating Sankey diagrams. In addition to the generated csv file, 4 additional several files such as follows will be created. Users can copy the contents of these files and paste them into the Sankey diagram generation tool's input window to generate a Sankey diagram.

- `0805_materialized.csv.sankeygen.estimated` : Input data for online Sankey diagram generator (SankeyGen, http://sankey.csaladen.es)/ Only estimated data points, base data not used

- `0805_materialized.csv.sankeygen.best_efforts` : Input data for online Sankey diagram generator (SankeyGen, http://sankey.csaladen.es)/ Estimated data points complement base data in case of missing data points.
- `0805_materialized.csv.sankeymatic.estimated`: Input data for SankeyMATIC (http://sankeymatic.com)/ Only estimated data points, base data not used
- `0805_materialized.csv.sankeymatic.best_efforts`: Input data for SankeyMATIC (http://sankeymatic.com)/ Estimated data points complement base data in case of missing data points

EFDW also provides a web-based interface for Data View Interpreter for the users. The software is developed using Tornado, which is a Python web framework. To be able to use the interface, the server module should be running on a system by executing the following command:

```
python sankey_gen.py
```

Then, open the address 'http://localhost:8888' in a web-browser (Safari, Chrome, Internet Explorer, etc.) will direct users to the user interface of the tool. Figure 12 shows a screenshot of the tool. Users can simply select a data file and data view file, then the tool provides a web-page where they can see previews of Sankey diagrams and download output files. The internal processing is done by the command-line tool that we described above. Examples and detailed usage of the developed tool will be explained in Section 4.

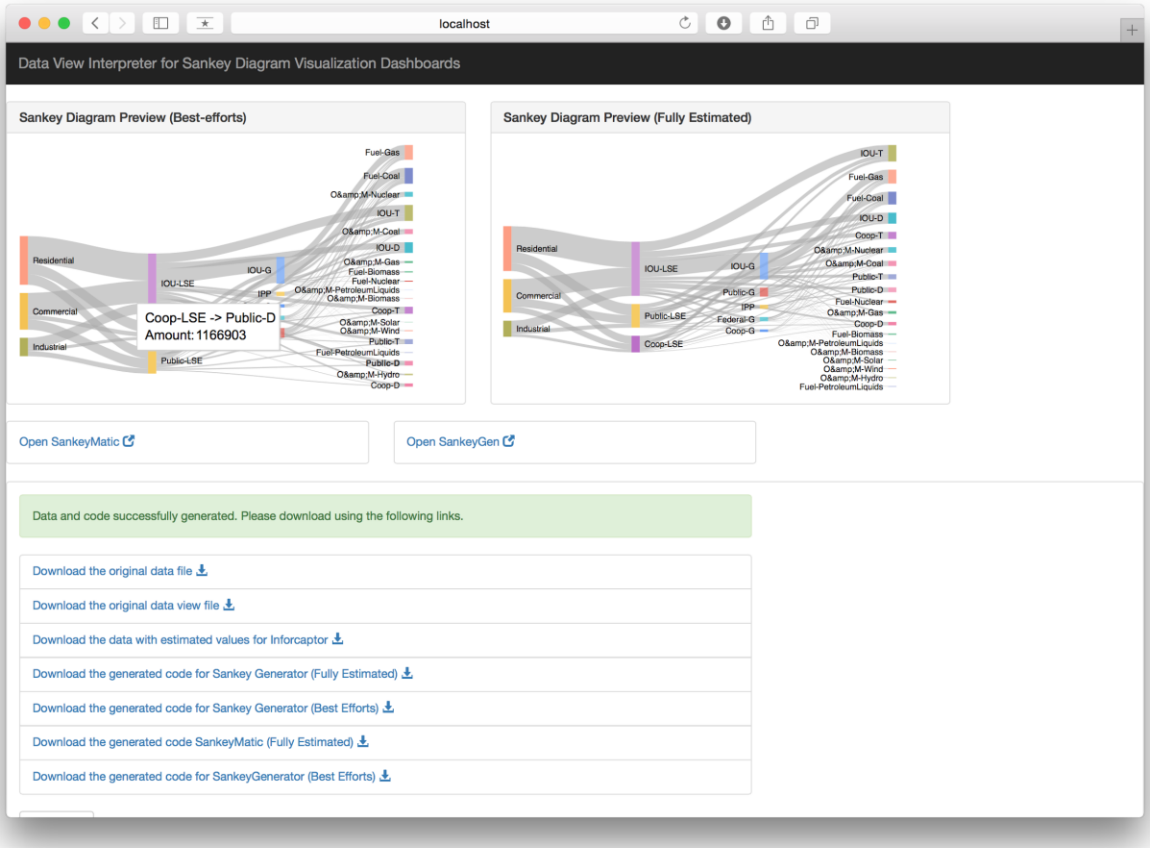


Figure 12 Screenshot of using the web-based user interface for data view interpreter.

3.4 EFDW VISUALIZATION DASHBOARDS

All the visualization dashboards developed under the EFDW framework uses the Infocaptor Server software as the main visualization applications. Additional visualizations created using Tableau (for drill down views), SankeyMATIC and Sankey Generator (for Sankey views with configurable coloring of nodes and flows) are provided as links from the Infocaptor dashboards. This section describes all the interactive Sankey dashboards created for the QER[2] [5], how to access them and the additional visualizations generated for providing more context around the data.

3.4.1 EFDW Repository Access Page

A main webpage has been created (link below) which provides access to the various visualization dashboards created by EFDW as selectable web links. The webpage was developed using the Infocaptor BI Software [6].
https://infoviz.ornl.gov/infocaptor_server/dash/mt.php?pa=sankey_visualizations_main_page_5745ec8aa81b7

The main web-page provides links to all the energy finance Sankey diagrams developed since May 17th 2016 and has been divided into 4 sections.

1. Tools
2. Combined Sankey Diagrams
3. RTO Sankey Diagrams
4. Non-RTO Sankey Diagrams

The Tools section provides access to the Infocaptor admin tool and the dataview interpreter. The next 3 sections provide access to the Sankey dashboards. Due to requests from the analysts to have both RTO and Non-RTO Sankey diagrams within the same view, the more recent versions of RTO dashboards incorporate 2 Sankey diagrams within the same view. The combined Sankey diagrams integrate the financial data for both RTO and Non-RTO into a single Sankey view.

[2] Quadrennial Energy review

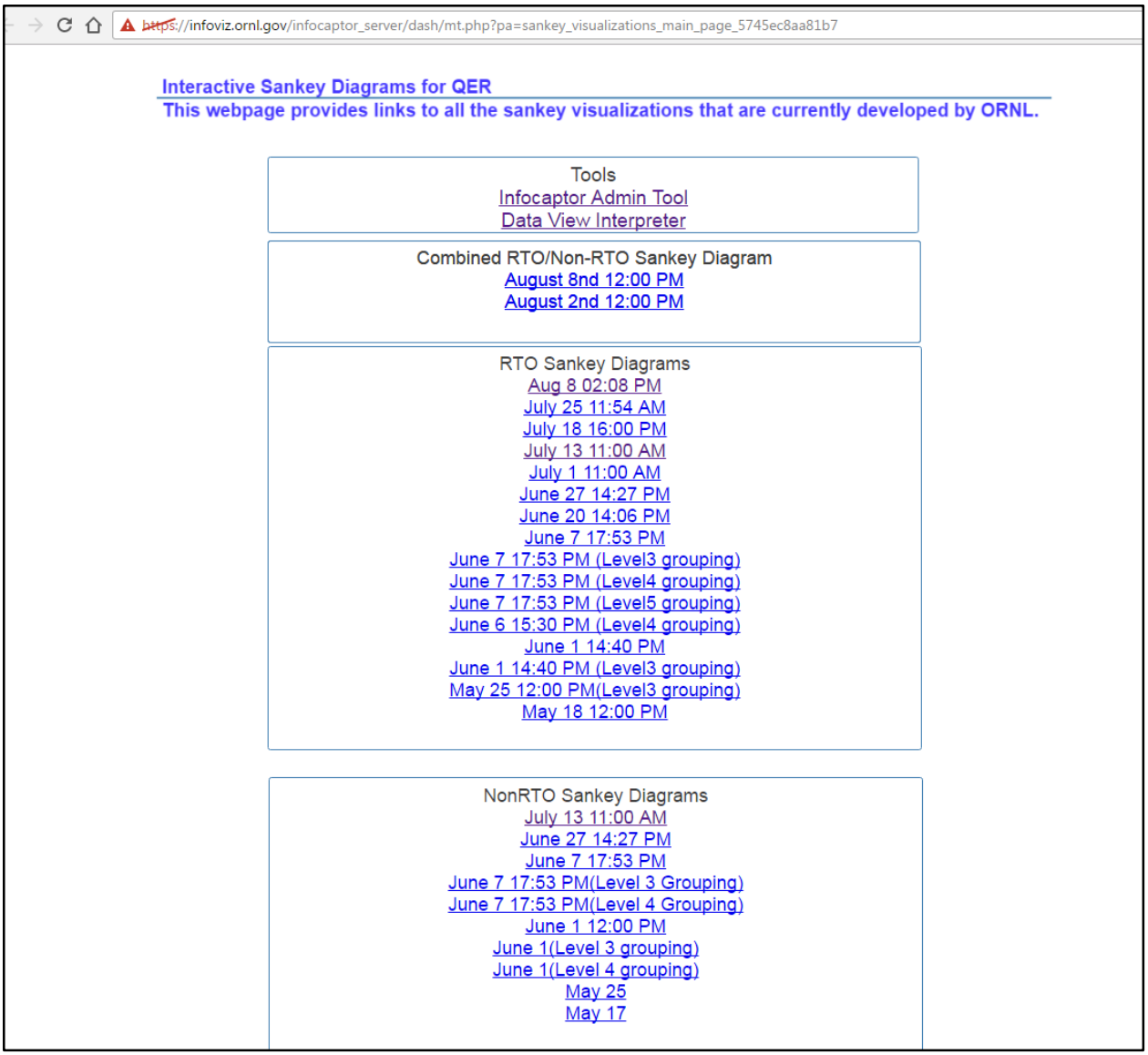


Figure 13 Main web page for Interactive Sankey Diagrams for QER

3.4.2 Sankey Diagram Visualization

We focused on Sankey Diagram, as a specific example of EFDW, and implemented the workflow for it first.

CREATING SANKEY DIAGRAM DASHBOARD PROCEDURE

Creating Sankey dashboards requires the following:

1. An input csv file in a specific format with the following attributes.
   From_Category_Buyer, To_Category_Seller, From_Buyer,To_Seller,Amount, Dataset

,Value_Method, Data_Complete,Year, Calculation, Notes, Methodology and Priority.

2. Infocaptor Software

Below is a step-by-step description for building a Sankey diagram using Infocaptor and the input csv file described above.
Note: The input csv file is the output file generated using the Web-based python tool described in section ? (rto_output.csv/ non_rto_output.csv)

1) Access the webpage https://infoviz.ornl.gov/infocaptor_server/dash/getin.php
2) Enter the username: admin and Password: XXX
3) Once you access the LaunchPad page, click on the dashboard editor icon as shown in Figure 14 below:

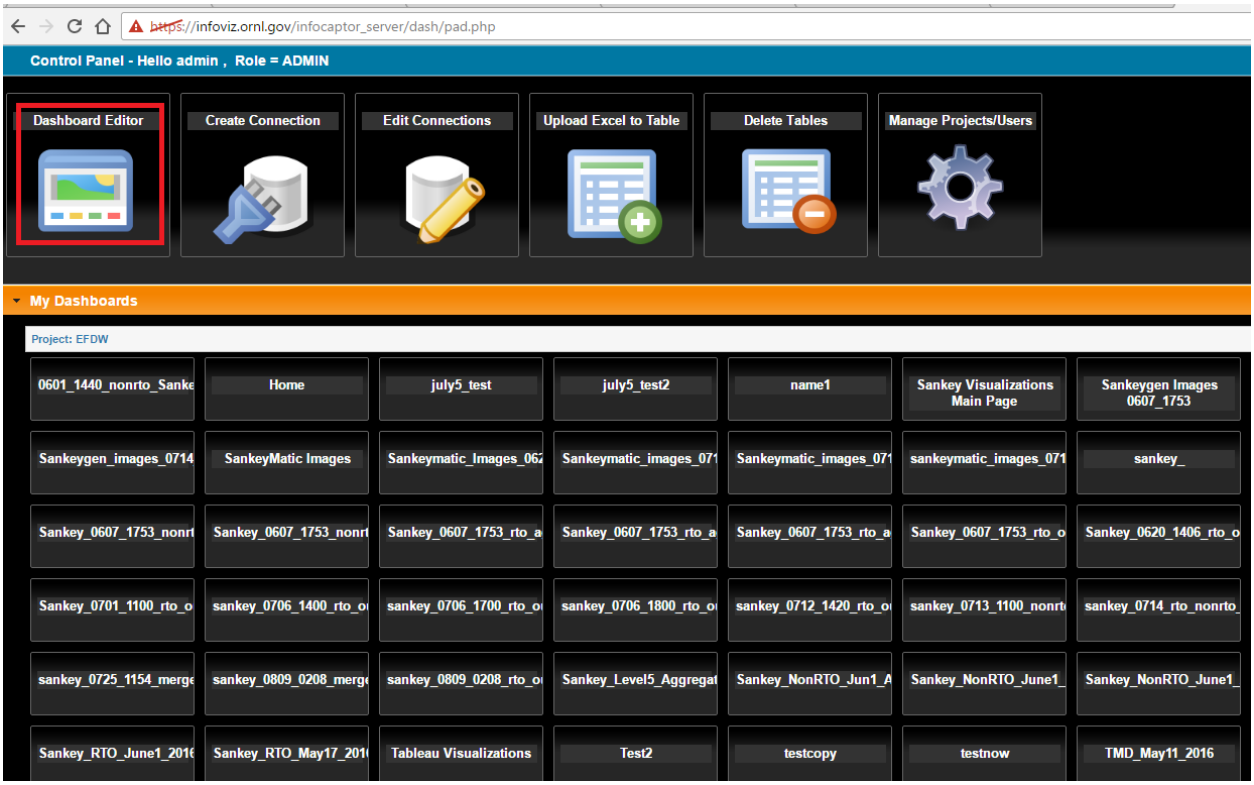


Figure 14 Admin page for Infocaptor

4) Under “Actions” menu, select “Upload Flat Files/Excel files”. Type a new MySQL database table name and paste the csv input file contents in the input box with a comment “Place XLXS Content here”. Then click on the “Start Upload Data” button.

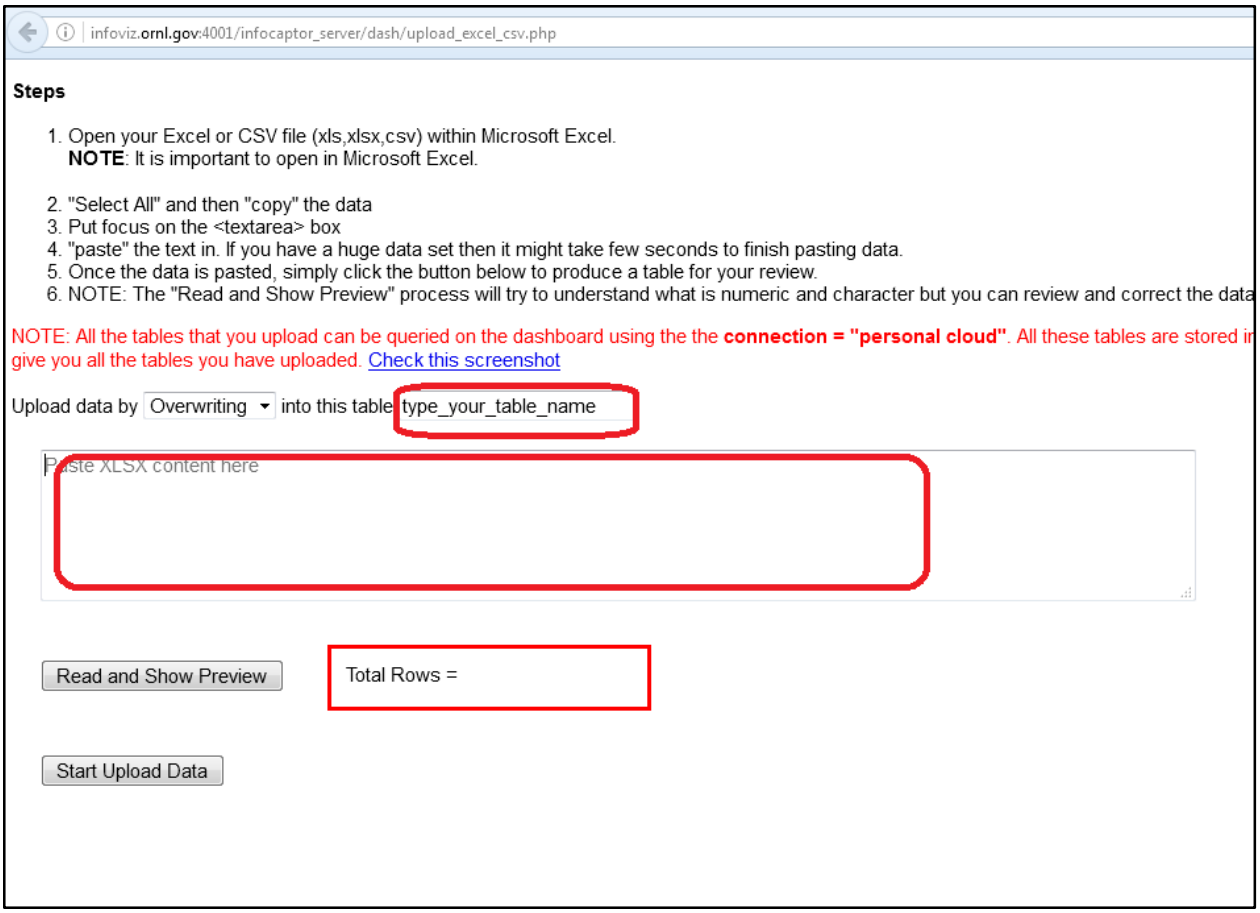


Figure 15 Uploading data sets to Infocaptor server

5) Within the dashboard Editor, click on the data tab on the top left of the page to upload the input.csv file contents into MySQL database table

Figure 16 Creating a dataset for Infocaptor

On the Data tab page, at the bottom of the page, click on
Infocaptor DataStore -> Personal Cloud -> Select the table name that has the data ->Click Analyze selected data from the Table.
Note: Check if the newly created table from step 4 appears in the table listing for personal cloud connection.

6) Clicking "Analyze selected data from the table" takes you to the visualizer tab page where you can drag and drop the "From_buyer" attribute to the rows shelf,the "To_seller" attribute to the cols shelf and "amount" attribute on the values shelf. Select the dropdown menu under Visualize As and select "Sankey Flow Levels".
   You should automatically see the Sankey widget showing up on the visualizer tab.

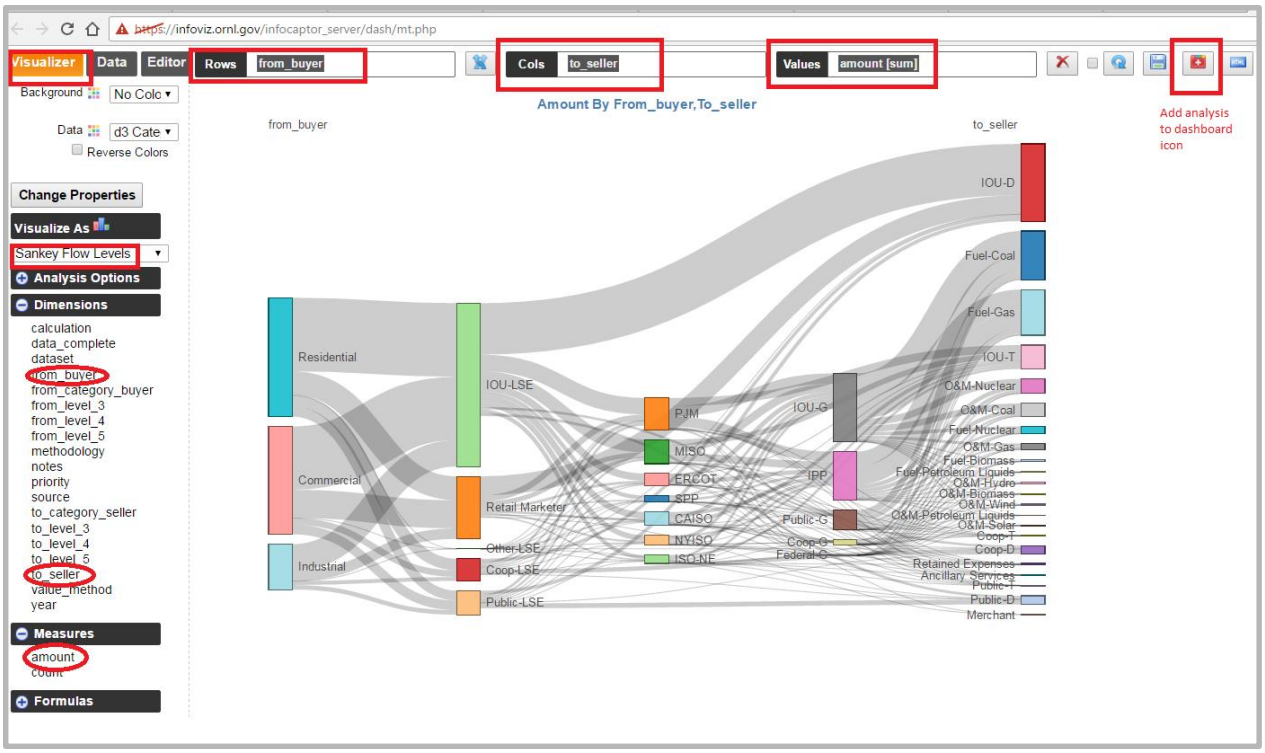


Figure 17 Creating a Sankey diagram with Infocaptor

7) Clicking on the "Add analysis to dashboard icon" will add the widget to a webpage that is being designed and takes you to the webpage.

Note: To create a webpage, go to the editor tab, and select "new page" from the "Actions dropdown menu" and give a name to the webpage as shown below:

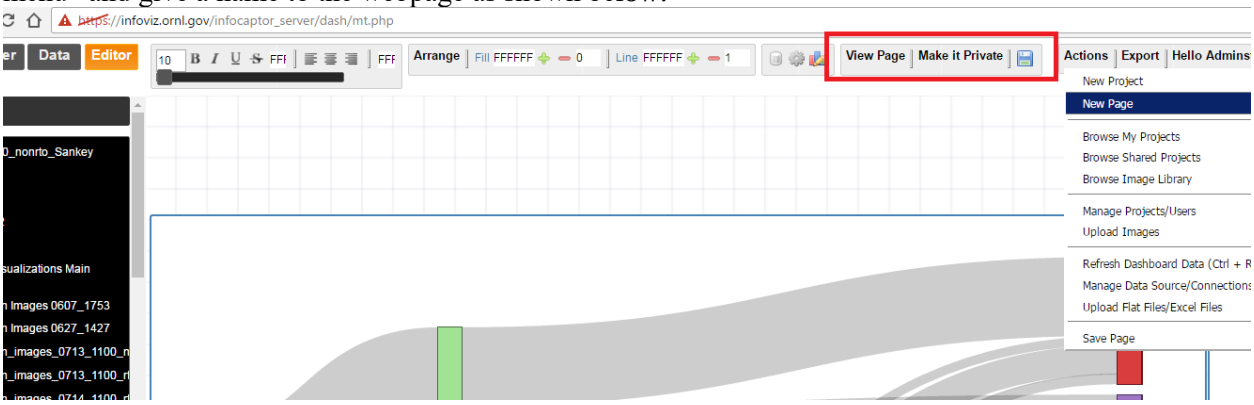


Figure 18 Publishing an Infocaptor dashboard

8) You can then save the webpage, view page and make the webpage private or public using the controls highlighted in the figure above.

9) Repeat the same procedure (step 1-8) to connect to the nonrto_output.csv file, create a sql table, click analyze data from that table using data tab, create another Sankey on the visualizer and then finally add the analysis to the existing webpage.

3.5 DESCRIPTION OF ENERGY FINANCE SANKEY DIAGRAM DASHBOARDS

The following energy finance Sankey diagram dashboards were created using the above described procedure

3.5.1 RTO/Non-RTO Dashboard (Best Efforts Sankey)

In this dashboard, 2 Sankey diagrams, one generated with RTO [7] data and the other for Non-RTO data using a combination of proportions and exact transactions data values from several data sets (termed as best efforts Sankey) are placed next to each other. Below the Sankey diagrams, there are 4 bar charts that provide aggregated amounts for various categories such as end user, generation, Load serving entity,O&M, transmission, Fuel etc for both RTO and Non-RTO. The 2 bar charts interactively display the money flowing out of and into the Sankey node selected on the first Sankey diagram. Below the barcharts, there are 2 tables that provide contextual data about their respective Sankey diagrams above them. When a Sankey node is selected, it displays all rows of transaction data flows coming into and leaving the Sankey node. To add tables which interact with the Sankey, refer to the section Adding Datatables to the webpage section.

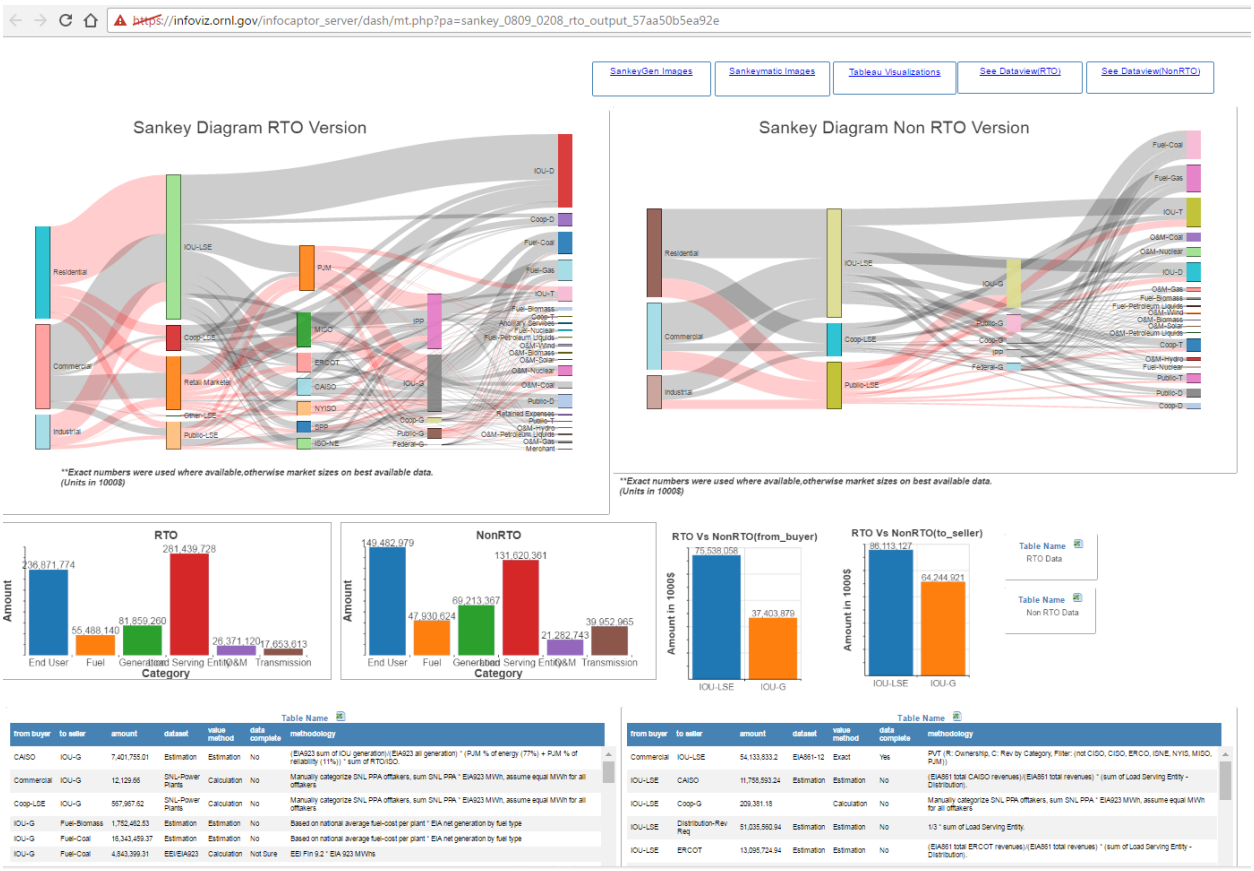


Figure 19 Infocaptor RTO/Non-RTO Dashboard for QER

3.5.2 RTO Dashboard (Best Efforts and Proportions Only)

In this dashboard, 2 Sankey diagrams (only RTO data), one generated using proportions only and the other using a combination of proportions and exact transactions data values from several data sets (termed as best efforts Sankey) are placed next to each other. Below the dashboard, there are 2 tables that provide contextual data about their respective Sankey diagrams above them. When a Sankey node is selected, it displays all rows of transaction data flows coming into and leaving the Sankey node.

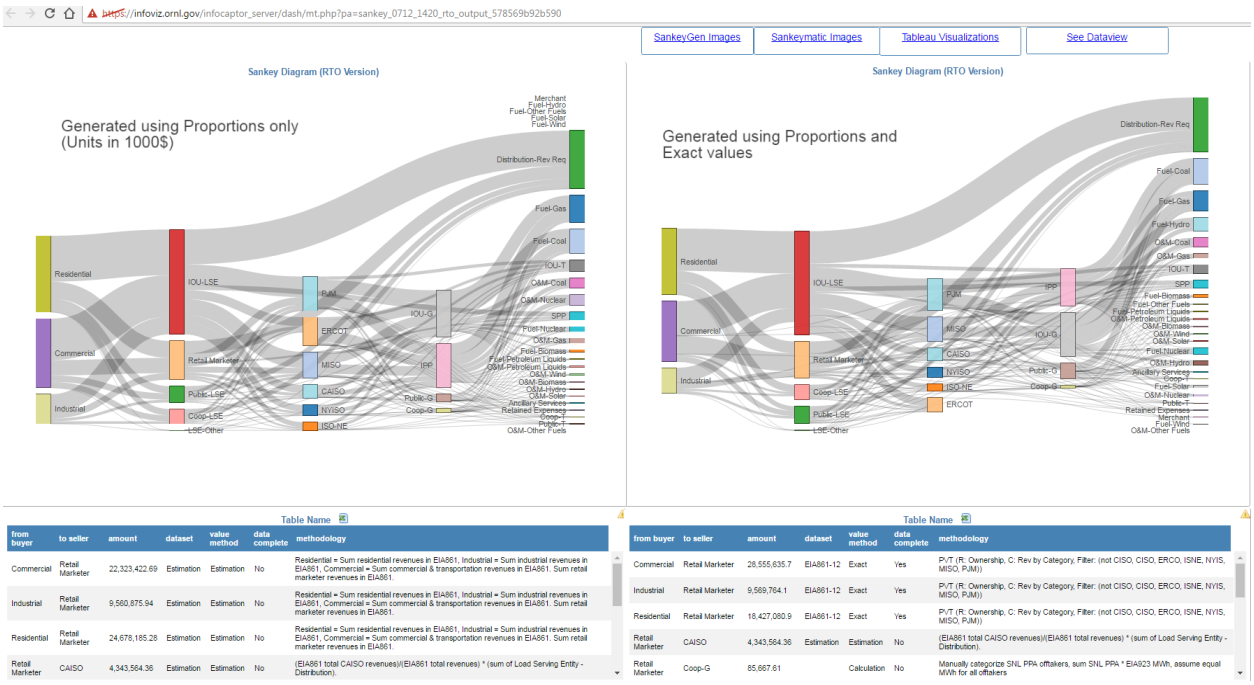


Figure 20 Infocaptor RTO Dashboard for QER

3.5.3 Non-RTO Dashboard (Best Efforts and Proportions Only)

In this dashboard, 2 Sankey diagrams (only Non-RTO data), one generated using proportions only and the other using a combination of proportions and exact transactions data values from several data sets (termed as best efforts Sankey) are placed next to each other. Below the dashboard, there are 2 tables that provide contextual data about their respective Sankey diagrams above them. When a Sankey node is selected, it displays all rows of transaction data flows coming into and leaving the Sankey node.

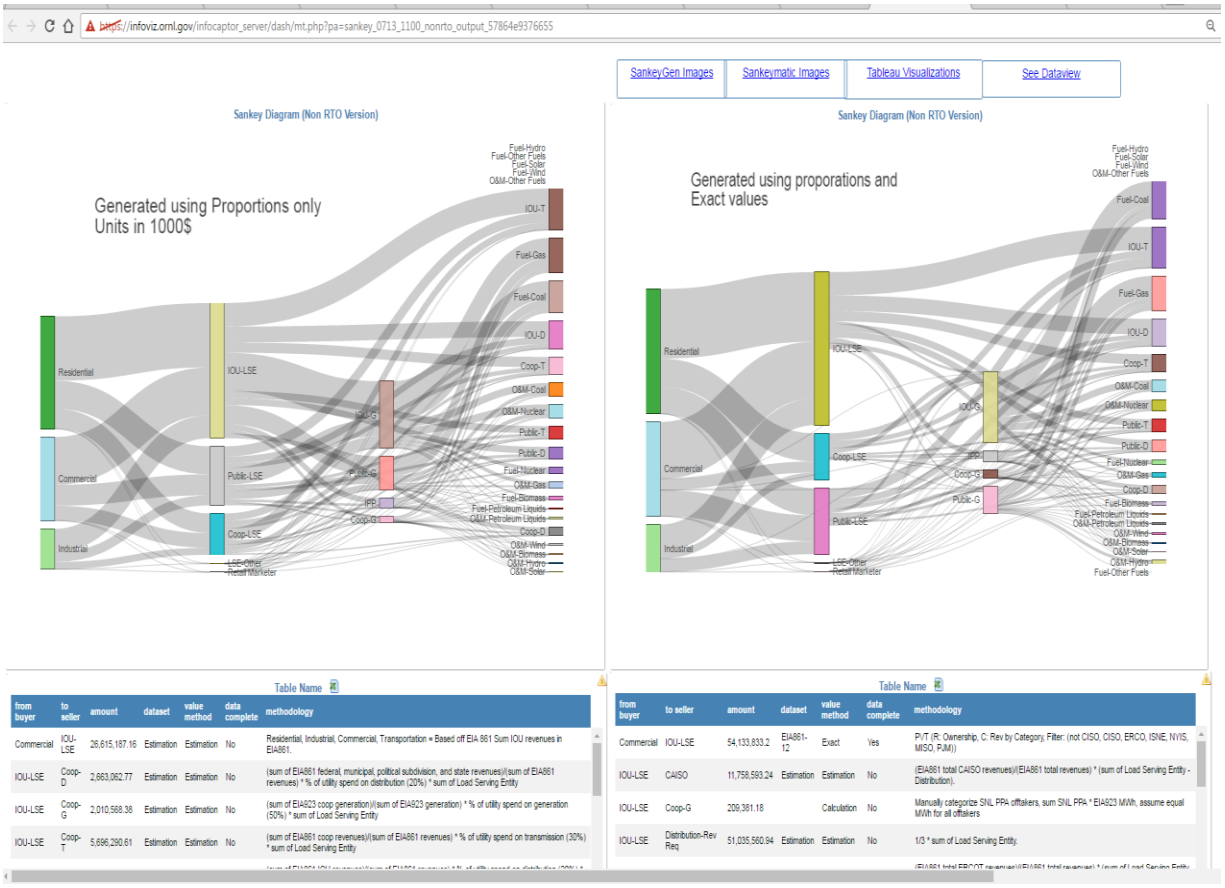


Figure 21 InfoCaptor non-RTO Dashboard for QER

3.5.4 Combined RTO/Non-RTO Sankey Diagram

In this dashboard, a single Sankey diagram that combines both RTO and Non-RTO data is displayed. Below the Sankey, there are 3 bar charts, one that provides aggregated amounts for various categories such as end user, generation, Load serving entity,O&M, transmission, Fuel etc for both RTO and Non-RTO. The other 2 bar charts interactively display the money flowing out of and into the Sankey node selected on the Sankey diagram.

Below the bar chart, there is a table that provides contextual data about the Sankey diagram flow data. When a Sankey node is selected, it displays all rows of transaction data flows coming into and leaving the Sankey node.

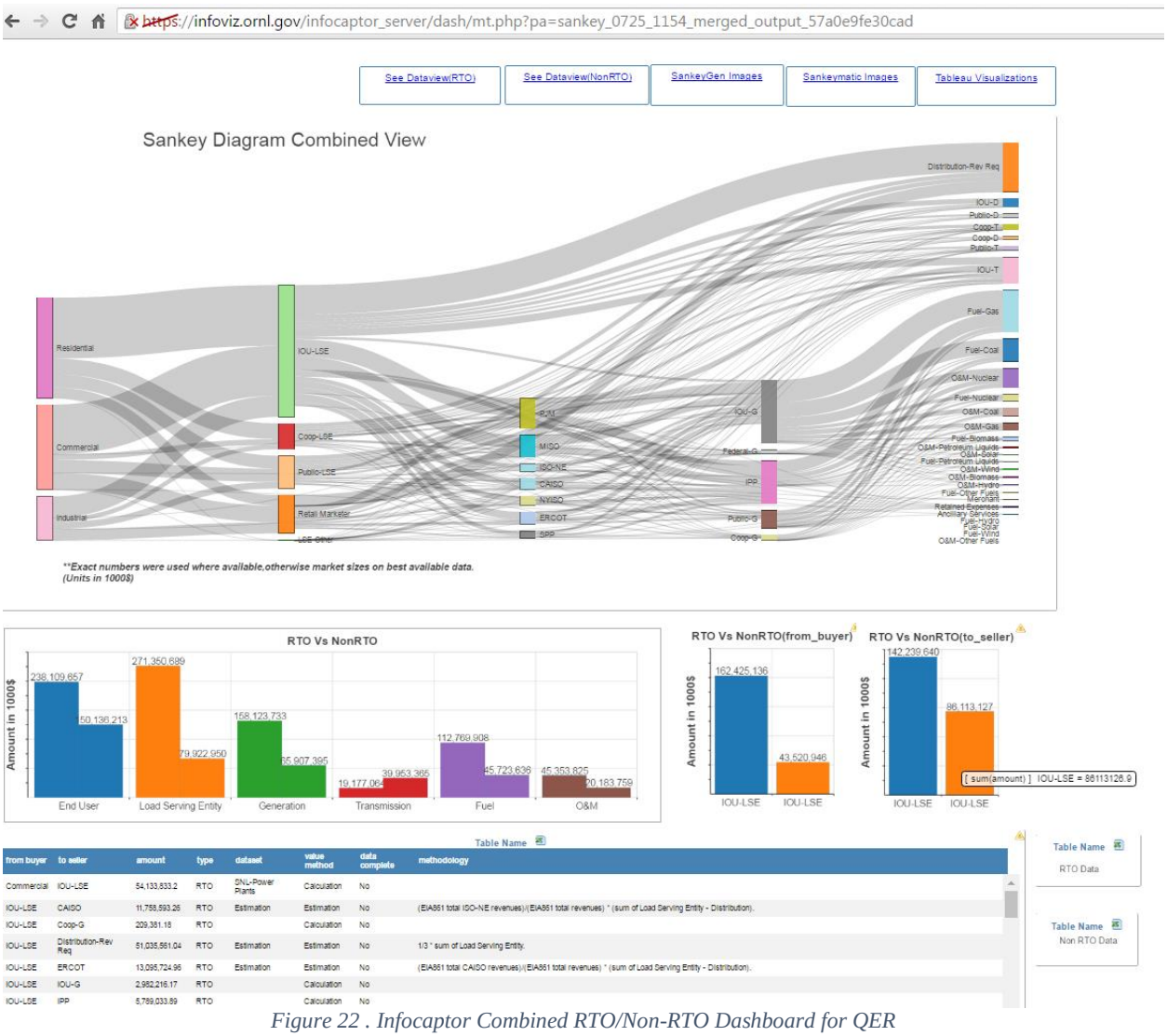


Figure 22 . Infocaptor Combined RTO/Non-RTO Dashboard for QER

3.6 ADDING DATA TABLES AND BAR CHARTS TO THE SANKEY DASHBOARDS

1. After logging into Infocaptor, on the editor tab select the web page on which you would like to add a barchart/data table. Scroll down on the top left menu to the dashboard widgets and drag and drop the bar chart on the webpage. Scroll down on the top left menu to the Container boxes and drag and drop the HTML grid widget on the webpage.
2. Right click on the HTML grid widget and select "Data Source". Ensure personal cloud is selected to access the data. Write an SQL query to get the table data.
   An example query is shown below in the figure. Note that param<pivot_grid_d21_node> is the Sankey node name whose object ID can be found by right clicking on the Sankey widget and selecting object ID.

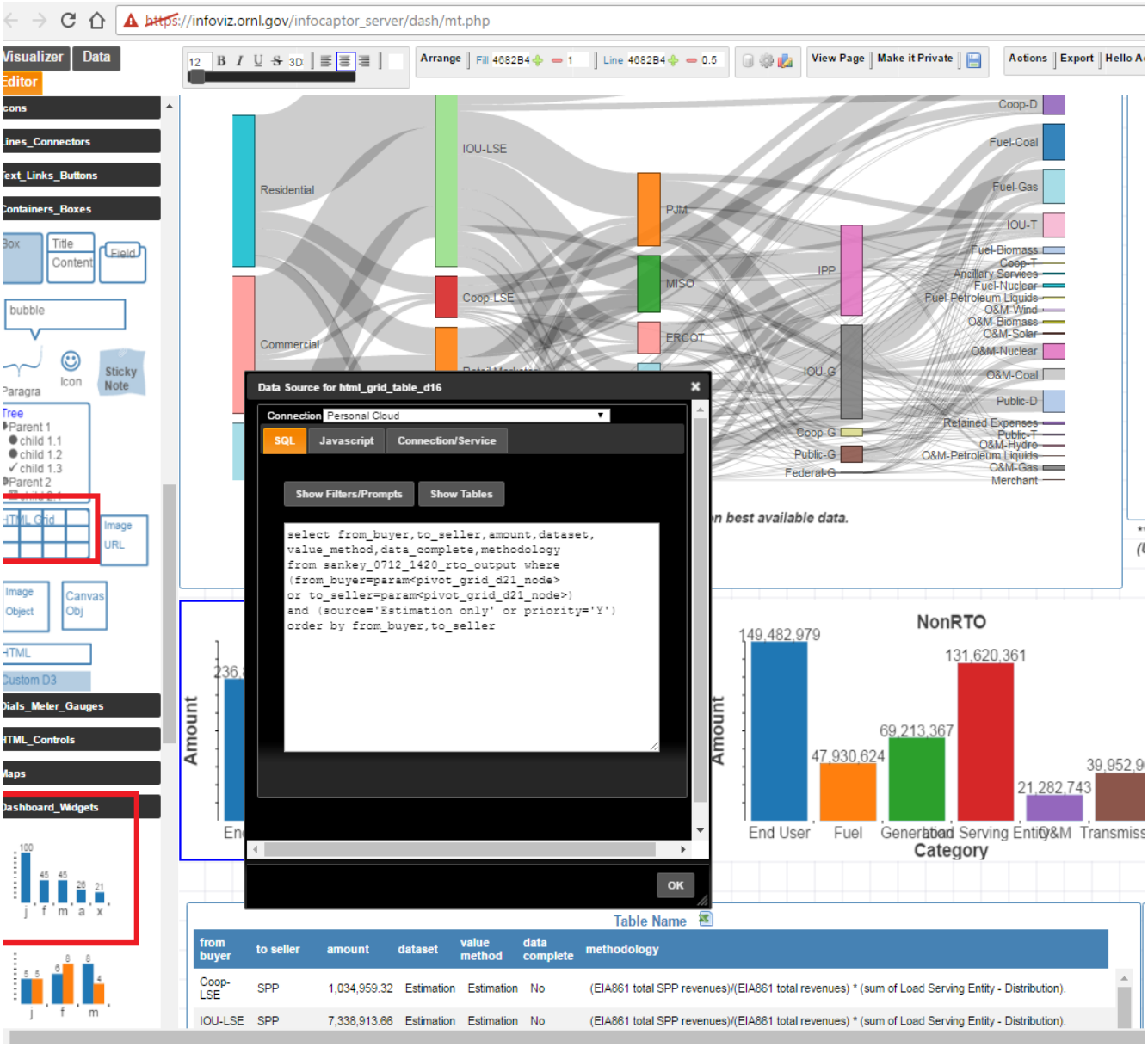


Figure 23 Adding data tables and bar charts to a Infocaptor dashboard

3.7 COMPLEMENTARY SANKEY DIAGRAM VISUALIZATIONS

Every Sankey dashboard provides links to other complementary Sankey visualizations generated using SankeyGen and SankeyMATIC. For e.g., clicking on these links will open a new webpage which shows Sankey diagram images generated using these tools. One of the advantages of using SankeyMATIC is that it allows coloring of nodes and flow links based on the attributes such as category, value method used to visually provide a sense of missing data for the data analysts. Below is a snapshot of the Sankey images

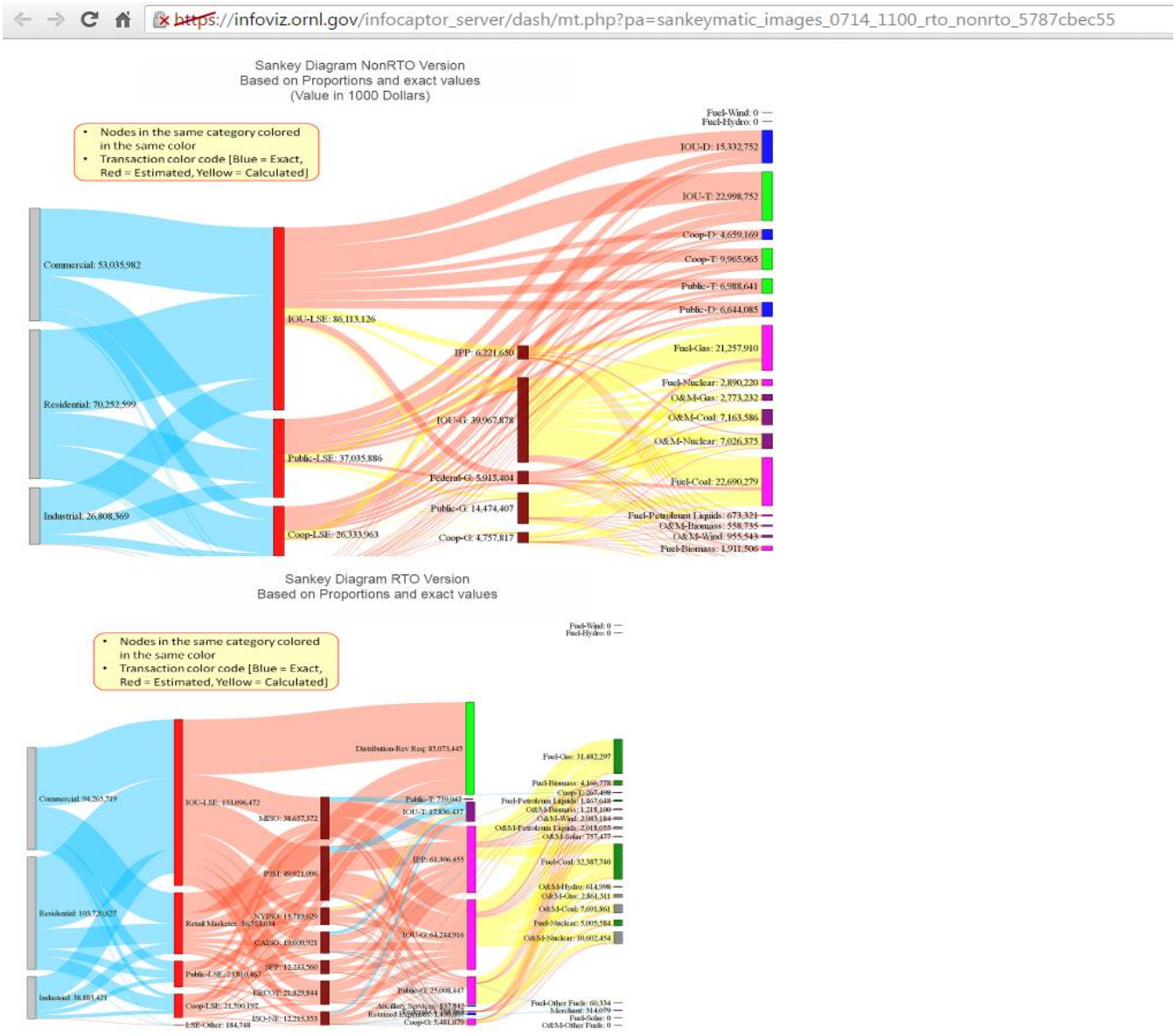


Figure 24 SankeyMATIC visualization for QER

Figure 25 SankeyGen visualization for QER

3.8 OTHER VISUALIZATIONS

3.8.1 Providing Drill-down View for Sankey Diagrams using Tableau

From the infocaptor Sankey dashboard, we can access the drill down visualizations of the Sankey data developed using Tableau software [8]. These visualizations are also deployed on a Tableau server, so all these visualizations are available via a web-browser as well.

The visualization dashboard below displays 2 treemaps for RTO and Non-RTO data where the color of the rectangle represents the seller category and the size of the rectangle represents the amount flowing from that category. The amount value is displayed along with the from category name,to_seller name and the value method used. The treemap can be dynamically updated based on the input selection options (highlighted in yellow).

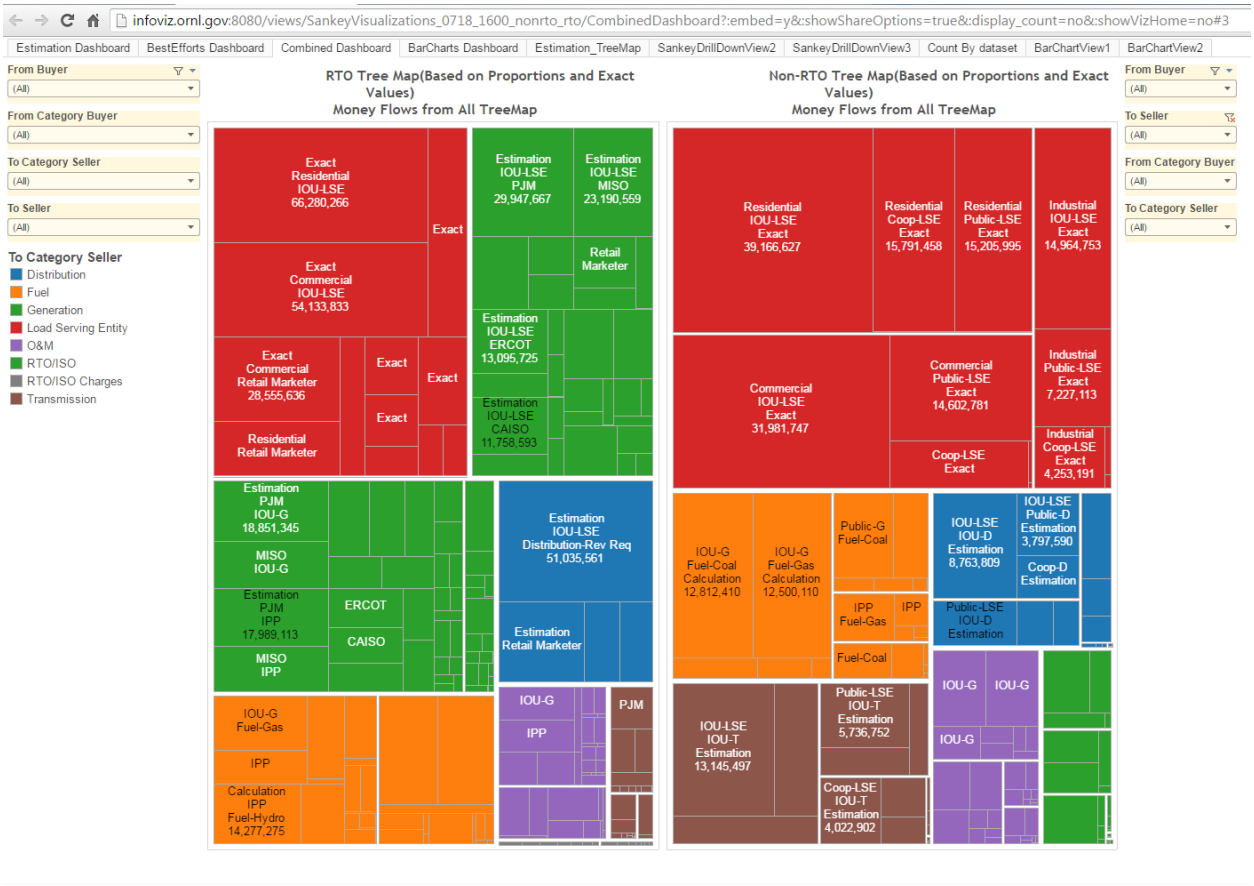


Figure 26 Tableau Tree-map visualization for QER

3.8.2 Tree Maps using Tableau

Figure below shows a drill down view of the treemap [9] where the user has selected the buyer as “Commercial”.

Figure 27 Tableau visualization for QER - Tree map drill down

In addition to the treemaps, we implemented the EFDW workflow for interactive pie charts and bar charts that are useful for drilldown views energy-finance data visualization within Tableau. Figure below shows input selection of “ISO-NE” for the buyer and “IOU_LSE” for the seller. The treemap dynamically updates to display financial flows from ISO-NE to other transmission and generation operators and to ancillary services. The pie charts and the barcharts also displays the money flows from ISO-NE and to IOU-LSE.

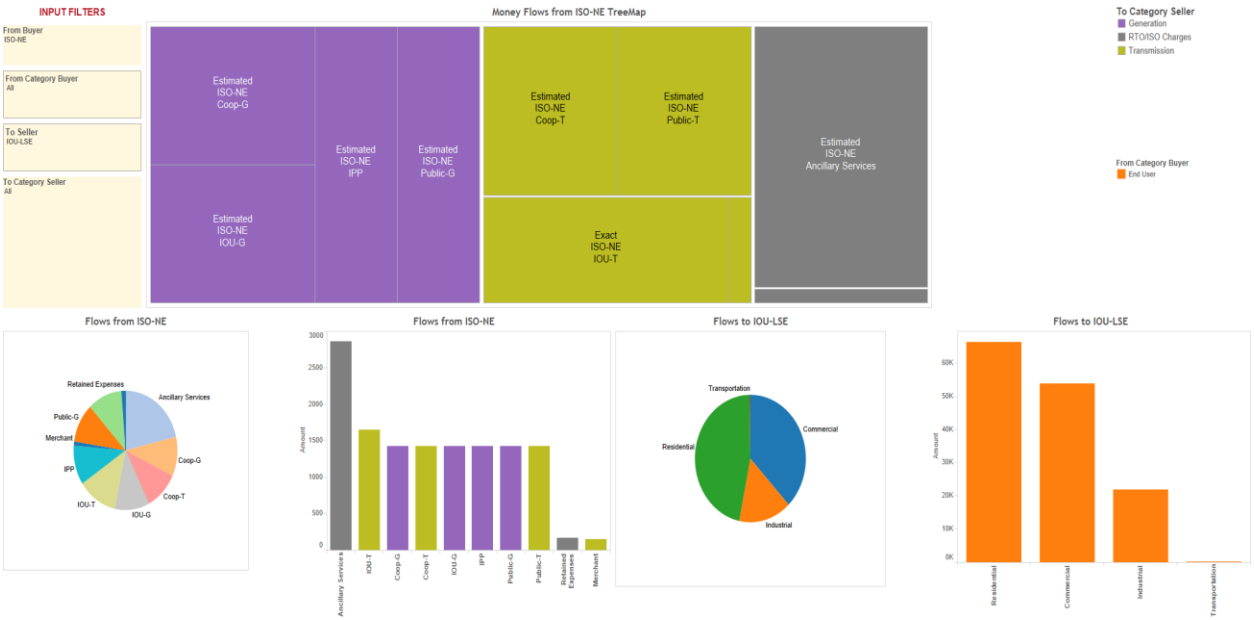


Figure 28 Tableau visualization for QER - other types of visualizations

4. USAGE OF THE DEVELOPED TOOLS

4.1 HOW TO USE WEB-BASED TOOL (AND COMMAND-LINE TOOL) FOR CREATING SANKEY DIAGRAMS

4.1.1 Understanding data, data view, and Sankey diagram previews

In this section, we first explain the tool's capabilities using several simple examples; then we show how we can generate even more complex data for Sankey diagrams. The tool is composed of the following files.

- `sankey_view_interpreter.py : Command-line python script that can be run from OSX/Linux/Windows terminal.`
- `sankey_gen.py : Python script to start up the Tornado server for web-based data view interpreter. This script uses the sankey_view_interpreter.py as its internal processing module.`
- `example/base_data.csv : An example base data file`
- `example/data_view.json : An example data view file`
- `static/*, template/* : HTML and Javascript files for web-based interface`

We will use the example base data file and the data view file in the example folder of the tool package. We need to understand these two example files.

- base_data.csv : This is the simplest base data file, as the file only contains columns and have no contents. In this case, all the transaction flows in the generated Sankey diagram will be estimated based on the proportion defined in used data view.

```
From_Category_Buyer,To_Category_Seller,From_Buyer,To_Seller,Amount,Dataset,Value_Method,Data_Complete,
```

```
Year,Calculation,Notes,Methodology,Priority
```

- data_view.json

```
{
    "comment: ": "test",
    "date" : "08/17/2016",
    "categories" : ["A","B","C"],
    "end_cat": ["C"],
    "layers": [0,1,2],

    "first_box:a" : 100,
    "first_box:b" : 200,
    "first_box:c" : 50,

    "def_sub_category:A" : ["a","b","c"],
    "def_sub_category:B" : ["x","y","z"],
    "def_sub_category:C" : ["1","2","3","4","5"],

    "grouping" : [],
    "grouping_names" : [],

    "unit": 1,

    "allowed_flow" : [
        {"from":"cat:A", "to":"cat:B", "weight":1.0, "note":""},
        {"from":"cat:B", "to":"cat:C", "weight":1.0, "note":""}
        ],

    "cat_itm_weight" : [

        {"from":"A", "to":"a", "weight":0.50, "note":""},
        {"from":"A", "to":"b", "weight":0.25, "note":""},
        {"from":"A", "to":"c", "weight":0.25, "note":""},

        {"from":"B", "to":"x", "weight":0.10, "note":""},
        {"from":"B", "to":"y", "weight":0.20, "note":""},
        {"from":"B", "to":"z", "weight":0.70, "note":""},

        {"from":"C", "to":"1", "weight":0.10, "note":""},
        {"from":"C", "to":"2", "weight":0.10, "note":""},
        {"from":"C", "to":"3", "weight":0.10, "note":""},
        {"from":"C", "to":"4", "weight":0.10, "note":""},
        {"from":"C", "to":"5", "weight":0.60, "note":""}

        ]
}
```

A data view file describes a Sankey diagram that we want to generate. In the data view file, users can describe when the file is created or updated, and what this data view is about. In this example, we can see that this data view file has been created (or updated) on 8/17/2016, and the comment says it's a 'test'. Categories and subcategories can be defined using the 'categories' and the 'def_sub_category:(category_name)' attributes. In this data view file, three categories are defined- A, B, and C where each one of category has subcategories {a,b,c}, {x,y,z} and {1,2,3,4,5}. Any strings can be a name of a category or a subcategory. Let us assume that we want to generate a Sankey diagram where the flow comes from category A to category B, then the amount is distributed to C. We don't want to allow flow start from A to C. In this example, in the 'allowed_flow' attribute, we include two flows that are

from category A to B, and category B to C. Since, we want to make the 100% of flow starting from A go to B, we assign weight 1.0 for the flow, and we do similar for the flow B to C. The total flow weight coming out of the same category should sum to 1. For instance, we can do such as;

```
    "allowed_flow" : [
        {"from":"cat:A", "to":"cat:B", "weight":0.5, "note":""},
        {"from":"cat:A", "to":"cat:C", "weight":0.5, "note":""},
        {"from":"cat:B", "to":"cat:C", "weight":1.0, "note":""}
        ],
```

The cat_itm_weight describe how much of amount coming into a category should be distributed across different subcategories. For example, in this example, the total amount that coming into A should be distributed to its subcategories a, b, c with the proportion 50%, 25%, and 25% respectively.

Let us create a Sankey diagram using the tool. Run the following command to execute data view interpreter.

```
python sankey_gen.py
```

The command will start up the Tornado web-server on the local computer where the script has been executed. Users can access to the web-based UI using a web-browser. In this tutorial, we use Chrome, but the tool is designed to be compatible with any modern web-browsers such as Safari, Internet Explorer, Firefox, etc.

The application is already being served by ORNL. Go to http://infoviz.ornl.gov using the web-browser. Then you will see the following screen.

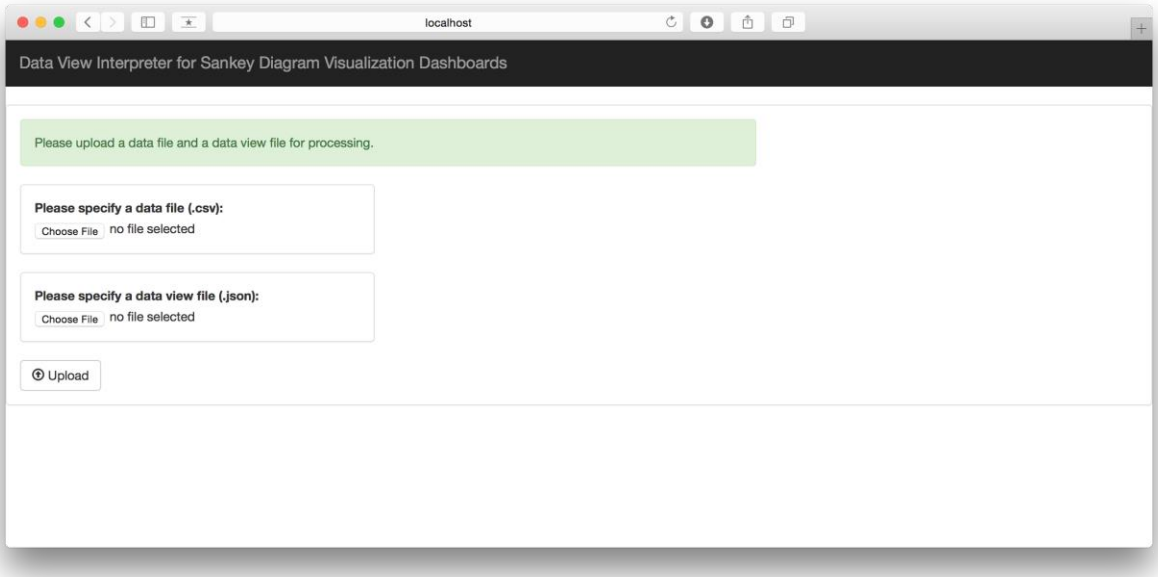


Figure 29 Data upload page for web-based Data View Interpreter

It is a very simple user interface that allows you to upload two files. Click the ‘Choose File’ buttons and select ‘base_data.csv’ and ‘data_view.json’ in the example folder. Then click the upload button. The the files will be sent to the server so that they can be processed. If successful, you will see the following screen.

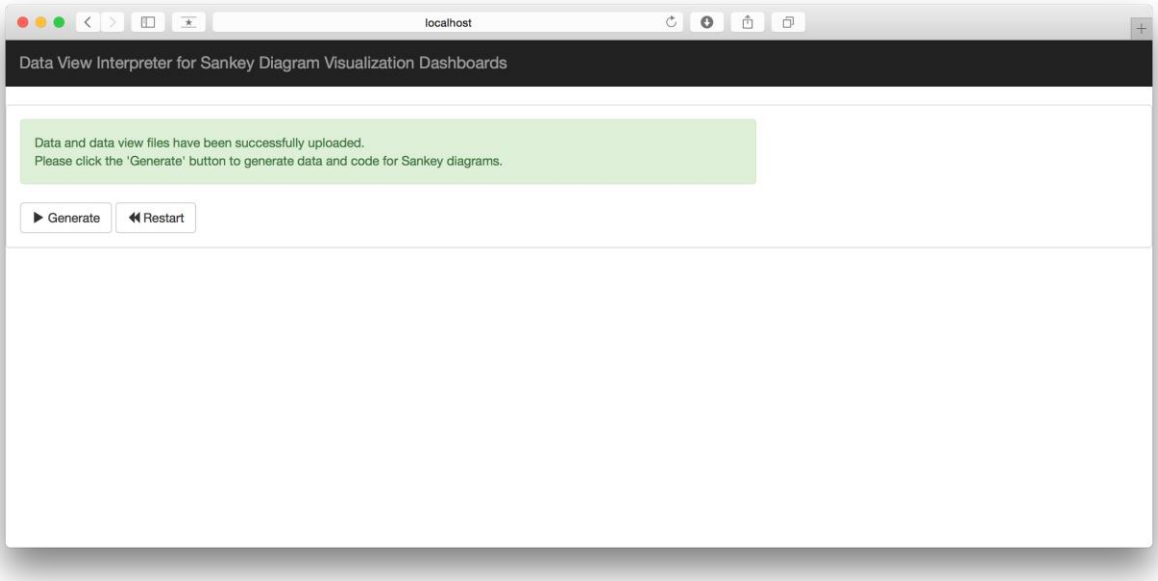


Figure 30 Data and data view files have been successfully uploaded. If data or data view files are not properly structured. Users will be directed to an error page.

If data and data view files are properly uploaded; the web-page will say that it is ready to generate the output files. Users can start over the process by clicking the ‘Restart’ button. Click the ‘Restart’ button to proceed.

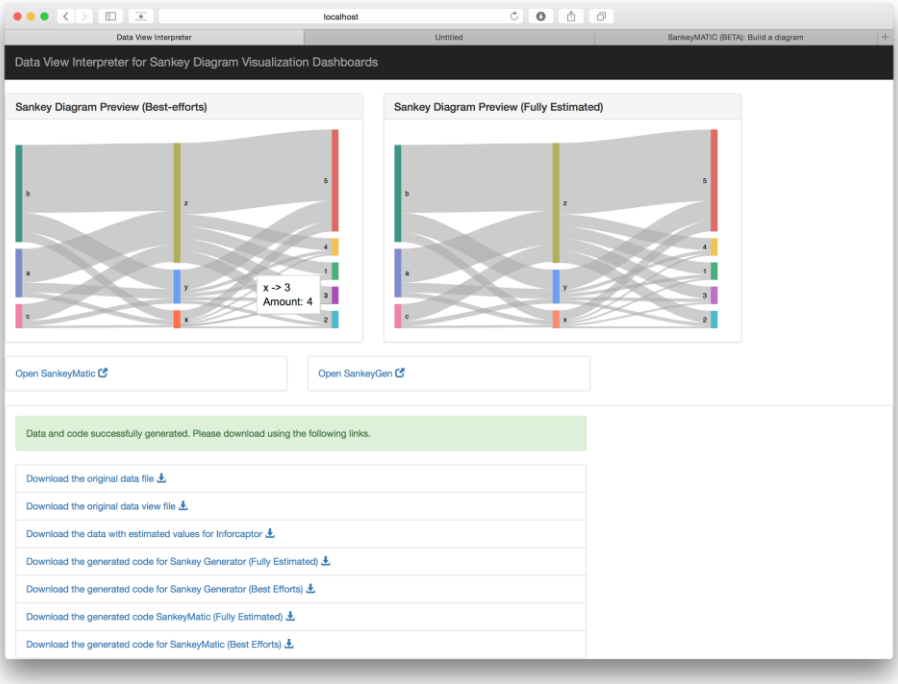


Figure 31 Download page for generated output files.

The web-based shows Sankey diagram previews - Sankey diagram preview (Best-efforts) and Sankey diagram preview (Fully-estimated). Best-efforts diagram will first include all of the available data in the base data file, then it will include estimated values for the unavailable data in the diagram. Fully-estimated diagram will not use any data in the base data file, and it will only include the data estimated by the tool based on the given proportions in the data view file.

In this example, note that there is no data in the base_data.csv, so both diagrams should be the same.

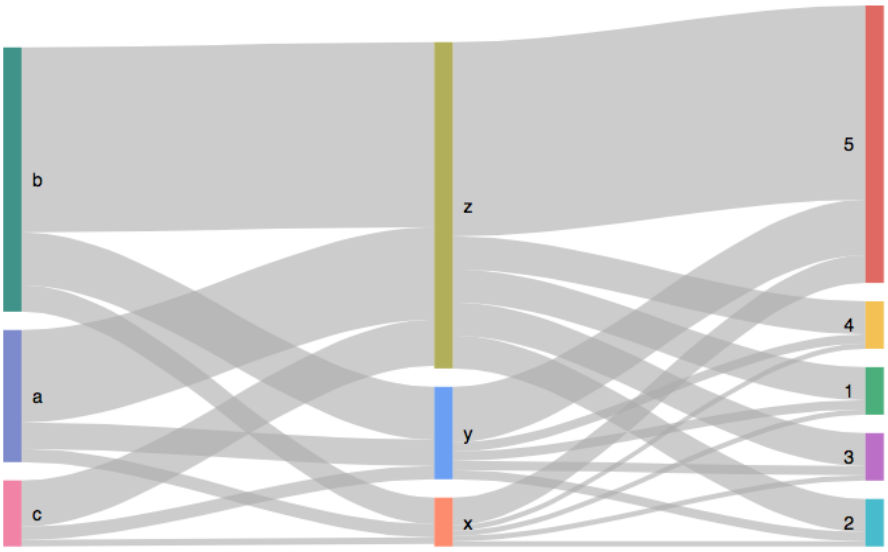


Figure 32 A simple Sankey diagram preview generated by using the example files base_data.csv and data_view.json

As we defined in the data_view.json file, in the first layer, boxes for the subcategories of the category ‘A’ are located. The size of first boxes - a,b, and c - are 100, 200, 50 respectively as we defined in the data view file. We can see that 100% of amount starting from the boxes in first layer goes to the second layer, where the boxes for subcategories of the category ‘B’ such as ‘x’,’y’,’z’ are located. The amount going into category B should be distributed to x, y, and z with the proportion of 10%, 20% and 70%. In other words, the amount from a, b, and c should be distributed to x, y, z with the proportion of 10%, 20%, and 70%. Thus, since the size of box a is 100, 10 goes to x, 20 goes to y, and 70 goes to z. Similarly, from the box b, 10% of 200 goes to x, 20% of 200 goes to y, and 70% of 200 goes to y. The transaction amount from the layer B to C is computed in the same way.

Modifying the categories, subcategories, proportions or allowed_flow will change the structure of generated Sankey diagrams. Let us modify some of the values in the data_view.json file and save the file as `example/data_view_modified.json` as follows. Modified values are color coded in red.

- data_view_modified.json

```
{
    "comment: ": "test",
    "date" : "08/17/2016",
    "categories" : ["A","B","C"],
    "end_cat": ["C"],
    "layers": [0,1,2],

    "first_box:a" : 100,
    "first_box:b" : 200,
    "first_box:c" : 50,
    "first_box:d" : 30,

    "def_sub_category:A" : ["a","b","c",”d”],
    "def_sub_category:B" : ["x","y","z"],
    "def_sub_category:C" : ["1","2","3","4","5"],

    "grouping" : [],
    "grouping_names" : [],

    "unit": 1,

    "allowed_flow" : [
        {"from":"cat:A", "to":"cat:B", "weight":0.4, "note":""},
        {"from":"cat:A", "to":"cat:C", "weight":0.6, "note":""},
        {"from":"cat:B", "to":"cat:C", "weight":1.0, "note":""}
        ],

    "cat_itm_weight" : [

        {"from":"A", "to":"a", "weight":0.50, "note":""},
        {"from":"A", "to":"b", "weight":0.25, "note":""},
        {"from":"A", "to":"c", "weight":0.10, "note":""},
        {"from":"A", "to":"d", "weight":0.15, "note":""},

        {"from":"B", "to":"x", "weight":0.10, "note":""},
        {"from":"B", "to":"y", "weight":0.20, "note":""},
```

```
        {"from":"B", "to":"z", "weight":0.70, "note":""},

        {"from":"C", "to":"1", "weight":0.10, "note":""},
        {"from":"C", "to":"2", "weight":0.10, "note":""},
        {"from":"C", "to":"3", "weight":0.10, "note":""},
        {"from":"C", "to":"4", "weight":0.10, "note":""},
        {"from":"C", "to":"5", "weight":0.60, "note":""}

        ]
}
```

We added one more subcategory named ‘d’ in the category A, and assigned 30 for the size of the new subcategory box. We modified that from A, 40% of amount goes to B, and 60% of amount goes to C. So, now the diagram should include flows that go directly from the first layer to the last layer. We also adjusted the weights for the cat_itm_weight attribute. If use the same base data file with the modified data view file, then the following preview diagram is generated.

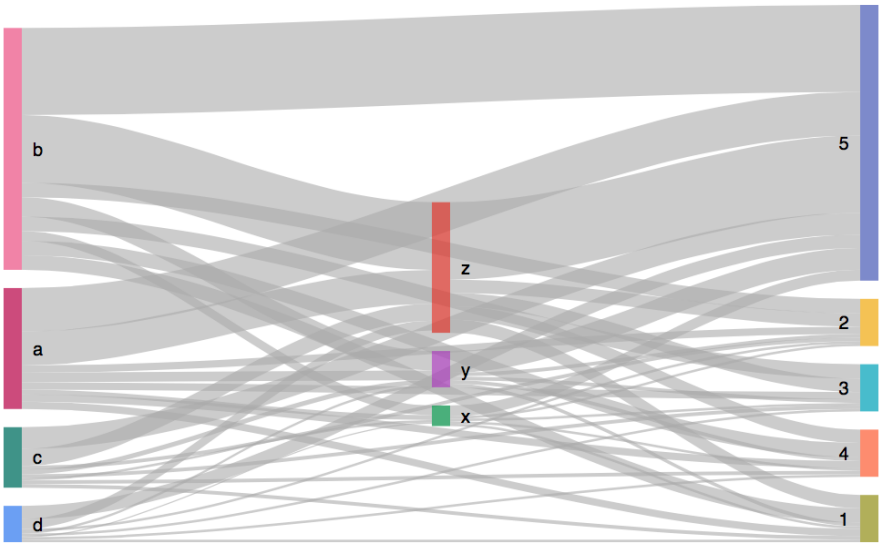
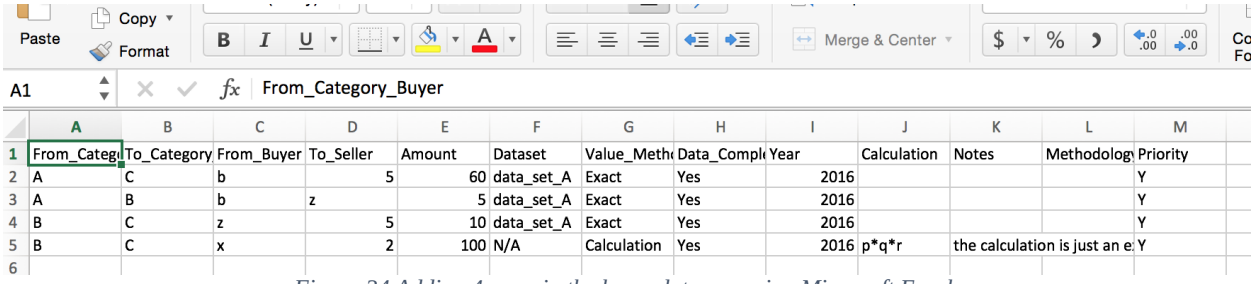


Figure 33 A simple Sankey diagram preview generated by using the example files base_data.csv and data_view_modified.json

We defined, the first layer now has a new box for the subcategory d, and the diagram include flows that goes directly from the first layer (category A) to the last layer (category C).

So far, we used an empty base data file; however, data file can contain data points for the Sankey diagram. Let us open the base_data.csv file and add some rows as shown in Figure 34. This process can be done by using the Microsoft excel or any text editors. After adding 4 rows, save the file as ‘example/base_data_modified.csv’



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	From_Categ	To_Category	From_Buyer	To_Seller	Amount	Dataset	Value_Meth	Data_Compl	Year	Calculation	Notes	Methodolog	Priority
2	A	C	b	5	60	data_set_A	Exact	Yes	2016				Y
3	A	B	b	z	5	data_set_A	Exact	Yes	2016				Y
4	B	C	z	5	10	data_set_A	Exact	Yes	2016				Y
5	B	C	x	2	100	N/A	Calculation	Yes	2016	p*q*r	the calculation is just an e		Y
6													

Figure 34 Adding 4 rows in the base_data.csv using Microsoft Excel.

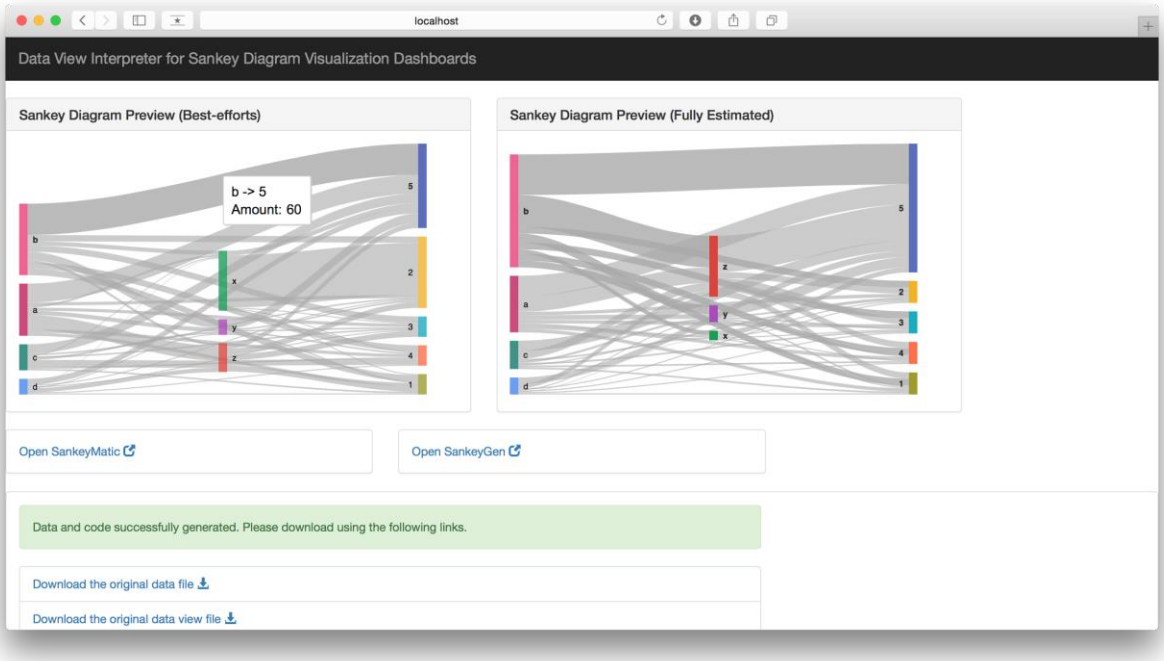


Figure 35 Figure 35. Best-efforts vs. Fully-estimated Sankey diagram

As shown in the Figure 35., since we included data in the base_data_modified.csv, two Sankey diagrams are differentiated. In the Best-efforts Sankey diagram preview, the amount for the flow from b (category A) to 5 (category C) is 60, as opposed to the fact that the amount for the same flow in the fully estimated value is 72. Note that in the Best-efforts Sankey diagram, the in/out amount for a box can be different as we use the given value and estimated value at the same time.

4.1.2 Generating more sophisticated Sankey diagrams using external tools

The web-based interface provides links for users to download the following five files plus to the original data/data_view files.

- Data with estimated values for Infocaptor: output.csv
- Generated data for Sankey Generator (Fully-estimated): output.csv.sankeygen.best_efforts
- Generated data for Sankey Generator (Best-efforts): output.csv.sankeygen.estimated
- Generated data for SankeyMATIC (Fully-estimated): output.csv.sankeymatic.best_efforts

- Generated data for SankeyMATIC (Best-efforts): output.csv.sankeymatic.estimated

output.csv is a materialized data view file, and it can be imported to a Infocaptor Sankey diagram dashboard. How to do that is explained in Section 4.2. In this section, we explain how we can create Sankey diagrams using Sankey Generator (http://sankey.csaladen.es) and SankeyMATIC (http://sankeymatic.com).

To use Sankey Generator, first, download the output.csv.sankeygen.* file, and open the file with any text editor. Select all and copy the entire contents. Then, by clicking the 'Open SankeyGen' button, open the Sankey Generator (http://sankey.csaladen.es). You will see the following web page.

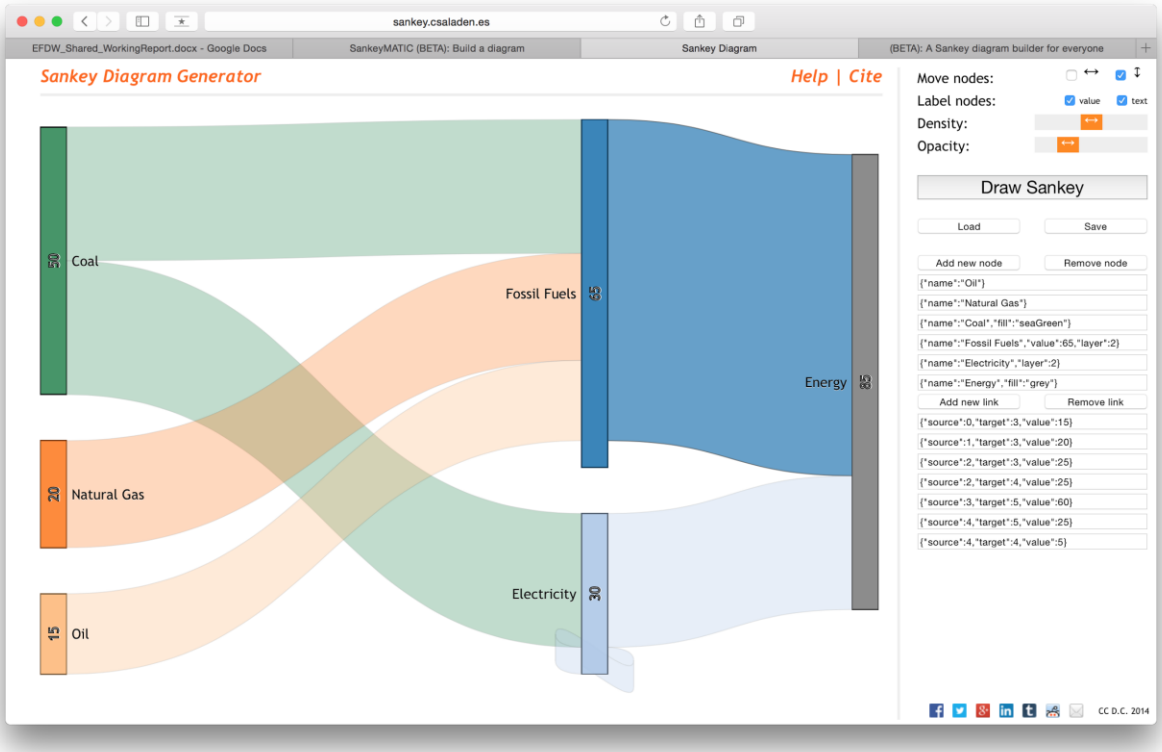


Figure 36 Sankey Generator interface

If you click, the 'load' button on the right panel, then you will see an empty box. Paste the copied contents of the output.csv.sankeygen.* file and click the 'done' button. Then, you can generate the Sankey diagram and interactively modify the diagram.

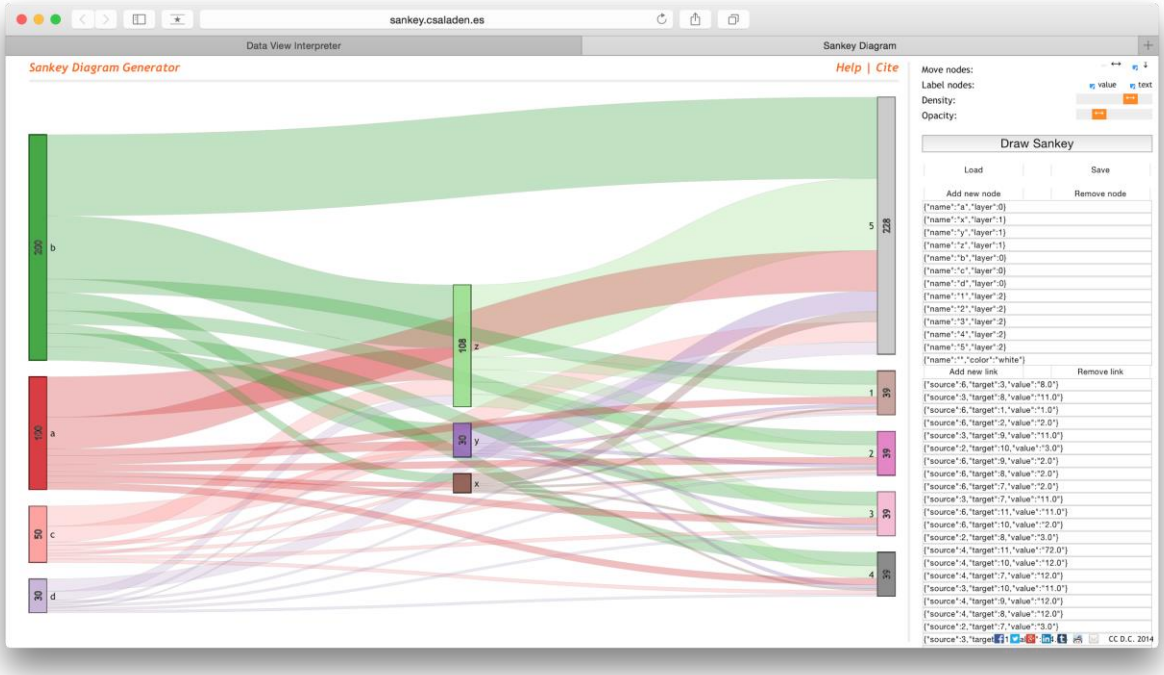


Figure 37 Generating a Sankey diagram with Sankey Generator (base_data.csv and data_view_modified.json)

Sankey Generator allows users to add/remove nodes and links, move nodes, change the density and opacity of the diagram.

To use SankeyMATIC, the process is similar. First, download the output.csv.sankeymatic.* file, and open the file with any text editor. Select all and copy the contents. Then, by clicking the 'Open SankeyMATIC' button, open the SankeyMATIC(http://sankeymatic.com). You will see the following web page.

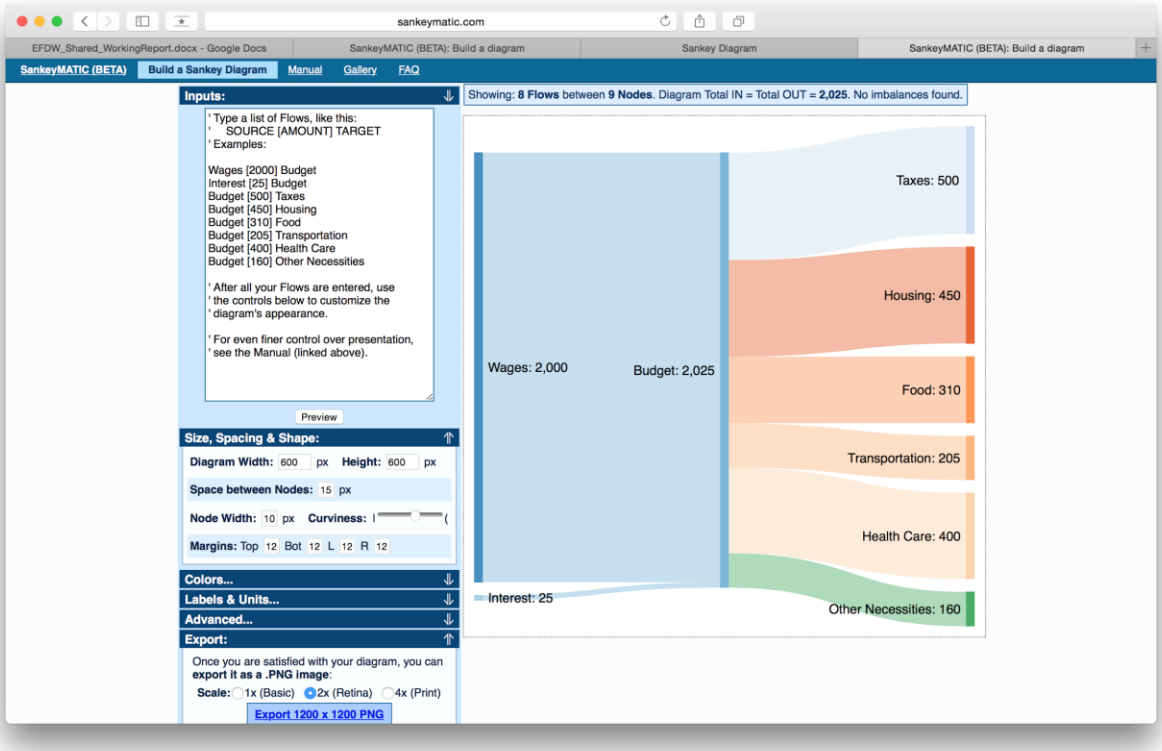


Figure 38 SankeyMATIC interface

Remove the contents of the Inputs box. Paste the copied contents of the output.csv.sankeygen.* file into the box, then, click the ‘Preview’ button. You can further adjust size, spacing, shape, colors etc.

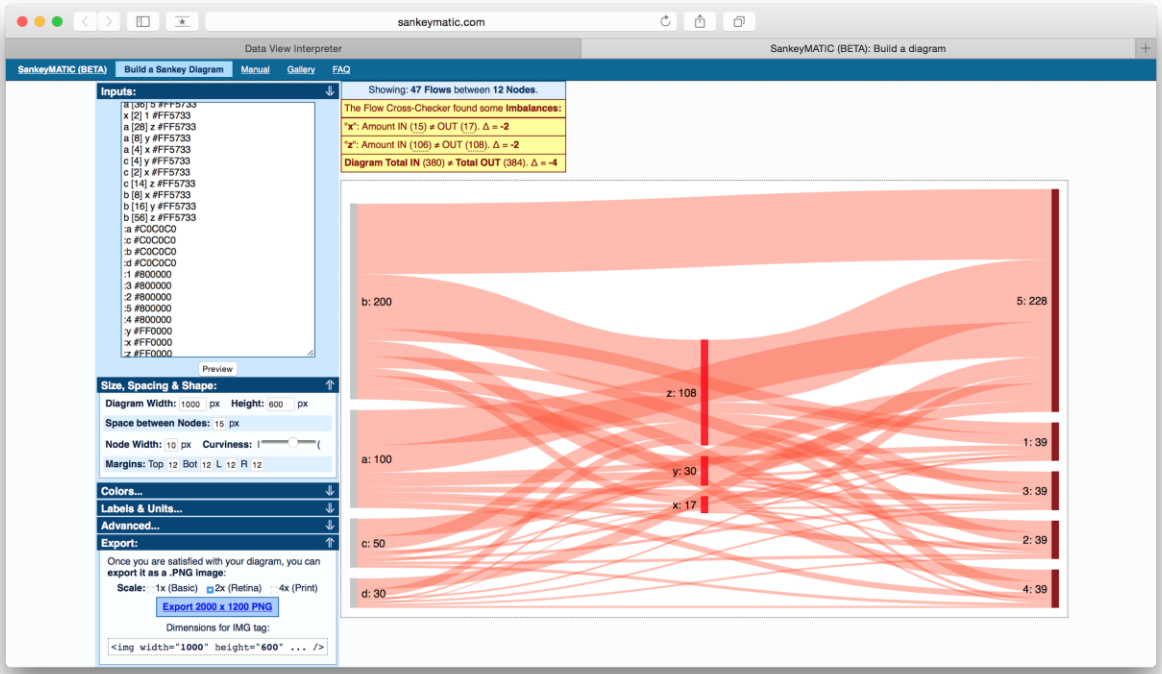


Figure 39 Generating a Sankey diagram with SankeyMATIC (base_data.csv and data_view_modified.json)

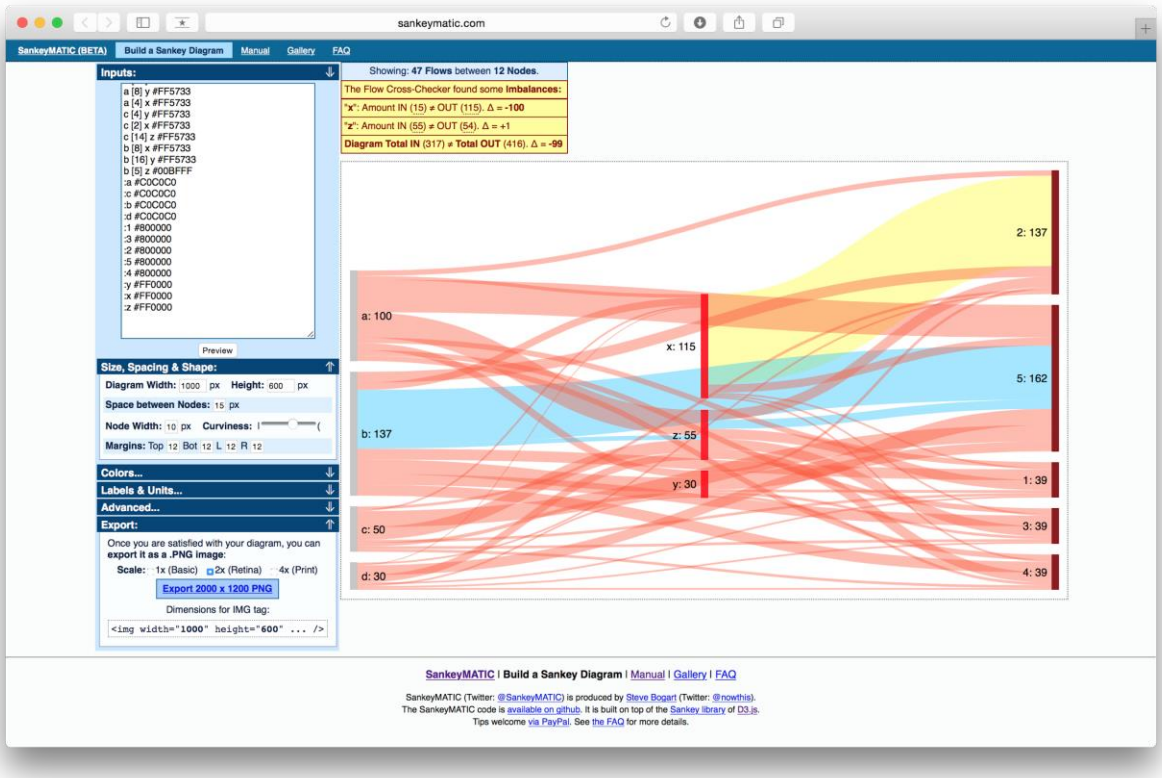


Figure 40 Generating a Sankey diagram with SankeyMATIC (Best-efforts version, base_data_modified.csv and data_view_modified.json)

By default, the Sankey diagram generated by the SankeyMATIC uses the following color-coding scheme: (1) Boxes for subcategories in the same category are colored with the same color; (2) Estimated flow amounts are color-coded in red/ Calculated flow amounts are color-coded in yellow/ Exact flow amounts are color-coded in blue.

4.2 HOW TO UPDATE INTERACTIVE DASHBOARD WITH NEW DATA

The 4 generated output files from the web-based tool serve as inputs to the infocaptor based interactive Sankey dashboards. Below are the list of steps for creating updated Sankey dashboards with new data.

1) Login to the infocaptor server using the link below:
   https://infoviz.ornl.gov/infocaptor_server/dash/getin.php
   UN: admin
   PW: XXX
2) Once your login is successful, the control panel home page will show up. Click on the "Dashboard Editor" icon to go into the web-page editor mode.

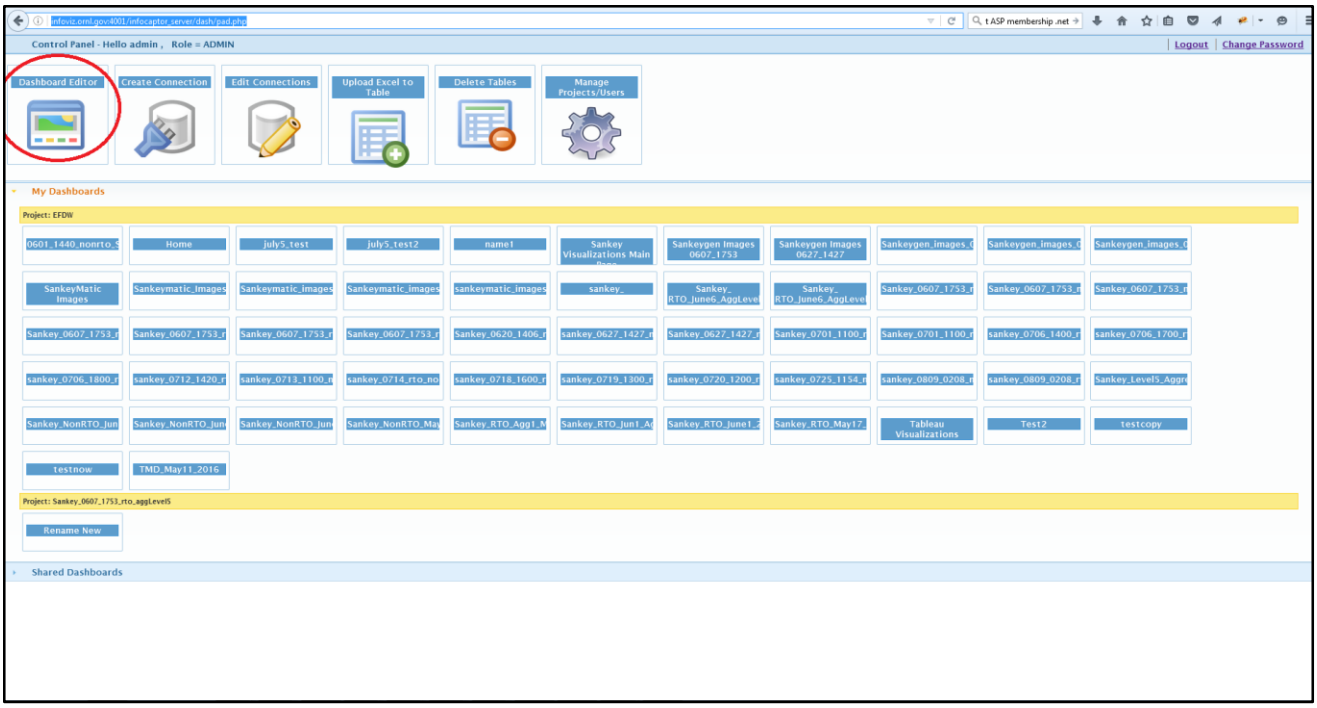


Figure 41 Opening an Infocaptor dashboard

3) Under “Actions” menu, select “Upload Flat Files/Excel files”. Type a new MySQL [10] database table name and paste the csv file contents in the input box with a comment “Place XLXS Content here”. Then click on the “Start Upload Data” button.

Note: To test if the table has been successfully created, select the data tab (next to editor tab-top-left corner of the webpage) and then select “Fetch Database Connections -> personal cloud” and see if the newly created table appears in the table listing for personal cloud connection.

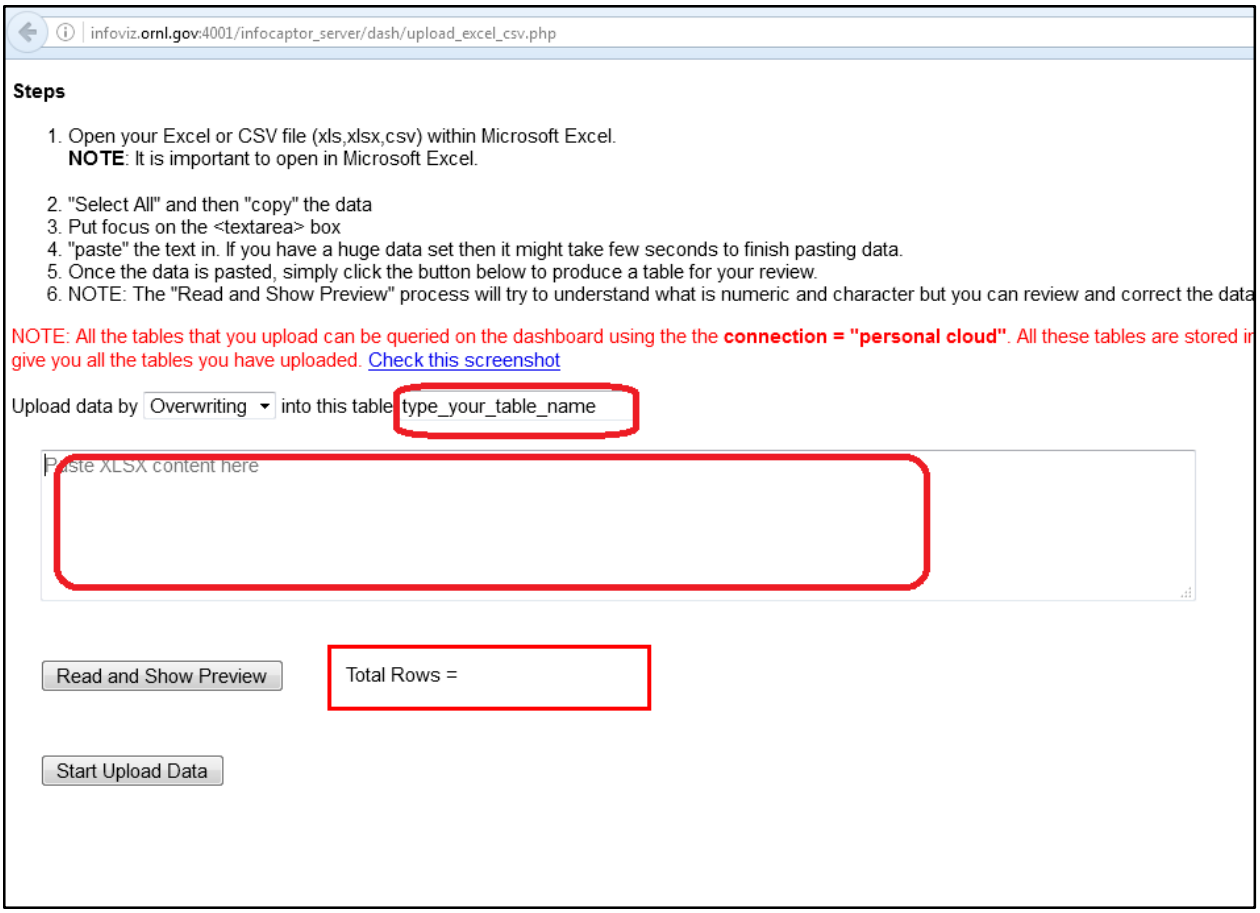


Figure 42 Upload a new dataset to Infocaptor

4) Under the Editor tab, select the webpage that needs to be updated.

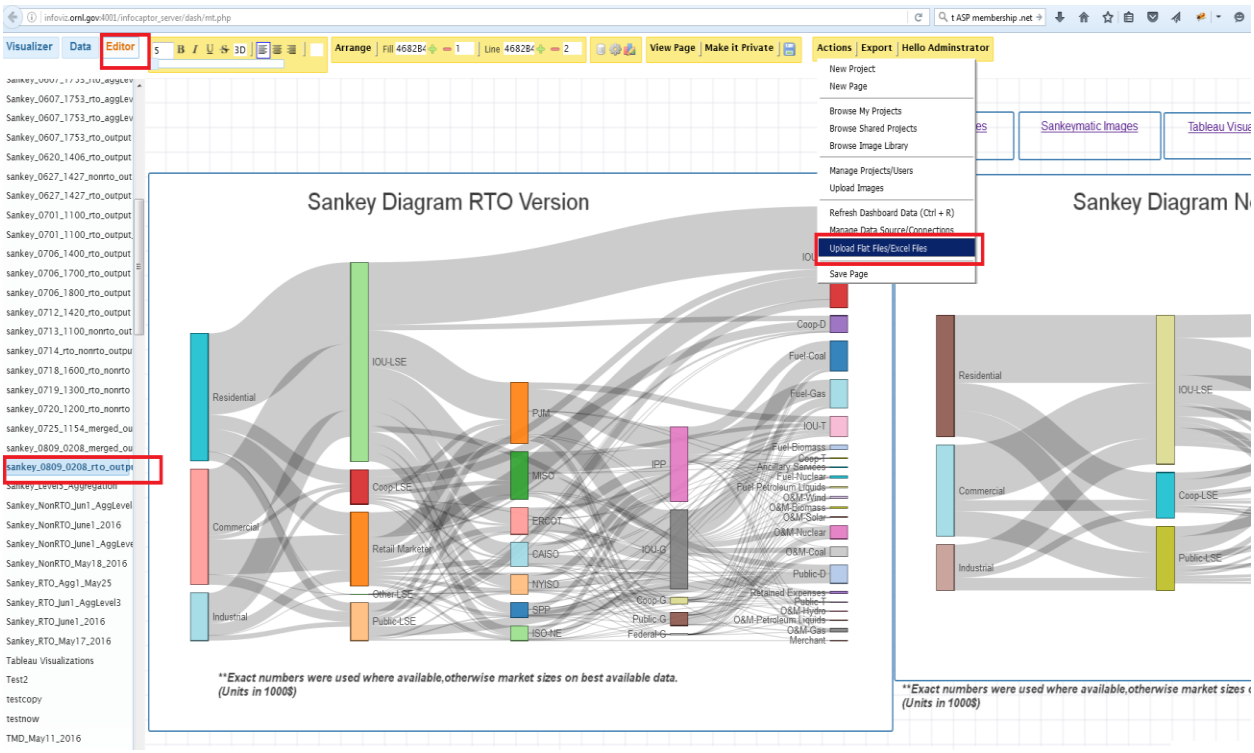


Figure 43 Updating a dataset for an existing dashboard- step 1

5) Select the Sankey diagram widget, right click to get the "Data source" option. The data source pop-up window has an SQL tab that contains an SQL query. Replace the table name with the newly created table as shown in the figure below. The widget will automatically get refreshed and update the visualization. This procedure needs to be repeated for each of the widgets within the dashboard.

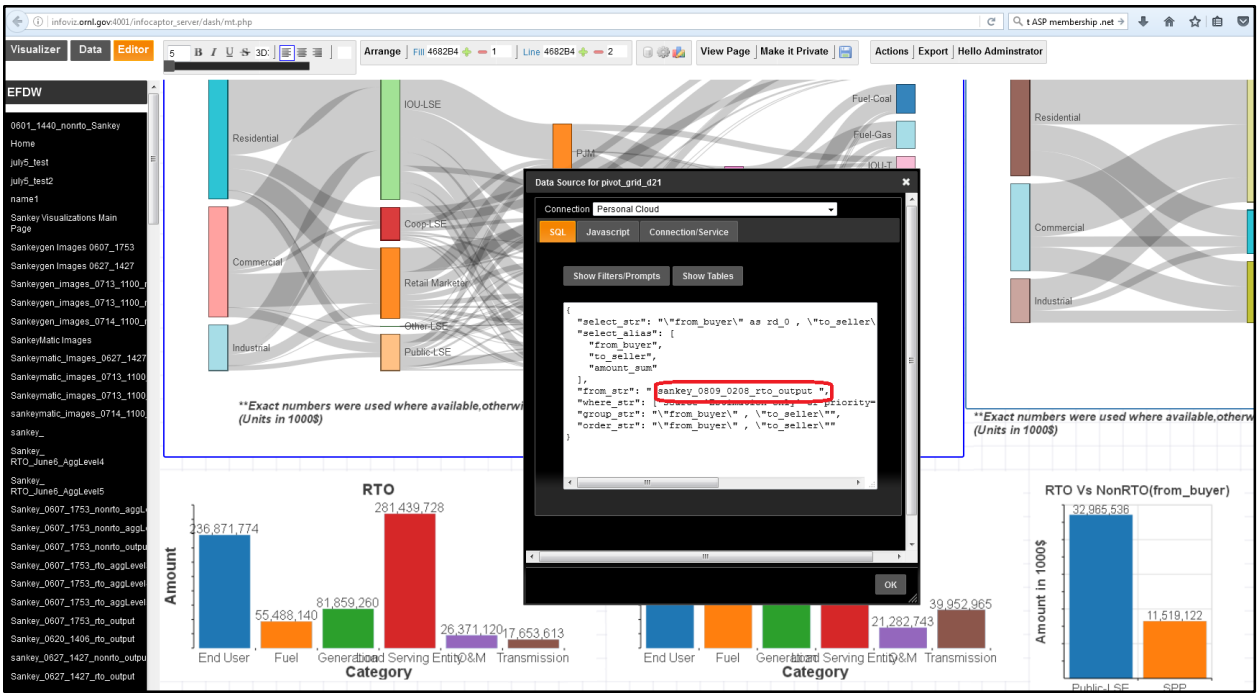


Figure 44 Updating a dataset for an existing dashboard - step 2

5. LESSONS LEARNED

In this section, we discuss challenges we faced during the project period and summarize lessons learned from experiences. Also, we included several recommendations that can be helpful for future projects. The goal of this section is that both ORNL and DOE analysts can collaborate more efficiently with each other in the future works by using this section as references. We grouped discussions into four different categories as follows.

5.1 COMMUNICATION AND DATA, DOCUMENT SHARING METHODS

During the project period, the team mainly used weekly conference calls as the main communication window; and it turned out to be very helpful, especially in the early phase of the project to clearly identify the goals of the project. In addition, holding weekly conference were also helpful for ORNL to understand the most immediate and urgent needs for the sponsors now. For this project, it was important to share screens as we were developing visualizations and visual analytic tools. BlueJeans tool [11] (https://bluejeans.com/) worked great for this purpose. Both organizations allowed flexibly adjusting time slots for the conference call depending on situations, and it was helpful to reduce the overhead of having weekly meetings.

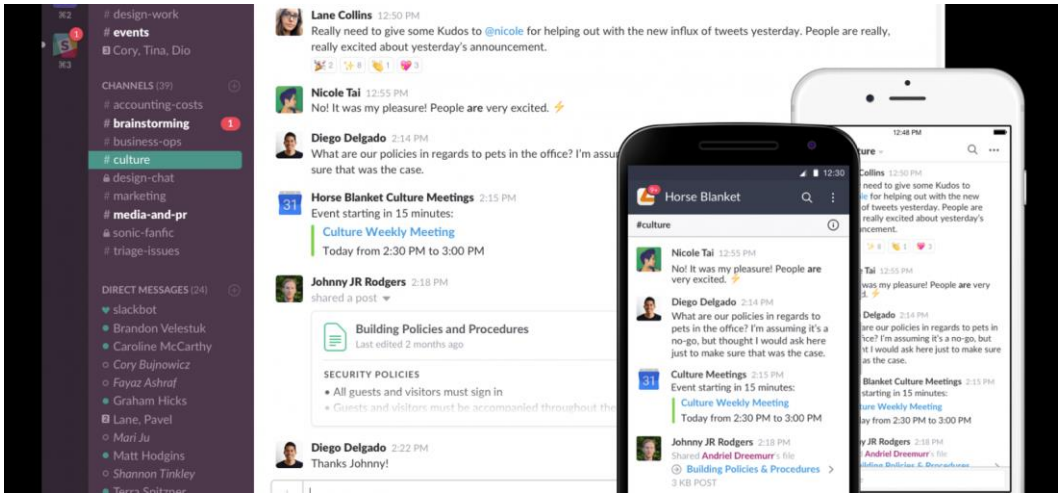


Figure 45 Slack (http://slack.com) can be an option for better communication method between ORNL and DOE

We mainly used Google Drive, Google Docs for sharing data, documents. These tools basically provided what we needed, and we used emails and phone calls to track what kind of changes or additions are made to the documents and data; however, it was challenging to track historical changes on data documents and conversations on various concurrent issues. For future work, it is recommended to adopt a cloud-based team collaboration tool like Slack [12] (http://slack.com). To be more specific, Slack allows integration services such as Google Drive, Dropbox, Github, and provides a chatroom where a private group can have conversations, include files, and things are searchable.

5.2 INITIAL PROJECT AGENDA VS. IMMEDIATE REQUESTS/NEEDS

Even though objectives of project are listed in the statement of work (SOW) document, it is usually hard to know what objective items among them will become most aligned with immediate requests and needs of sponsors ahead of time. For instance, in the case of this project, during the project period, it has become clear that visualization of Sankey diagram from energy finance datasets and development of related tools are the most urgent needs for The Quadrennial Energy Review (QER) report. As per the sponsor's requests, the team invested more efforts on these specific items.

When making decisions to focus on specific objectives, it is very important that both ORNL and DOE are clearly understanding how it can affect the initial agenda and proposed deliverables of the project. Stepping back from the on-going situations, carefully reviewing the initial project proposal is crucial to prevent the situation where some of important items are not delivered as proposed. Having a mid-term SOW document review is recommended.

5.3 SOFTWARE DEVELOPMENT AND DEPLOYMENT

As outcome of this project, we implemented several software to automate some of the processes that DOE analysts manually do. During the project period, there were some occasions that DOE analysts could leverage the capability of the tools that were being developed.

However, in most cases, ORNL had to be responsible for utilizing the tools to generate results. There are few reasons. First, deploying software on DOE machines for DOE analysts to run software themselves is not a straightforward task. Since many of modern software have dependencies between them (e.g., database server, web-server, Python modules, etc.), there is no easy way to simply deploy a software on

sponsor's side. Second, some tools were initially implemented as a command-line tools that runs on the Windows/linux/unix terminal; and it requires technical background.

We tried to resolve these issues by deploying the developed software on ORNL's side and providing intuitive web-based interfaces which are accessible via standard web-browsers from DOE's side. Regulations such as port and firewalls have been considered. It should be acknowledged that certain amount of time is necessarily required if DOE needs to deploy the developed software on DOE's side at the end of the project period. It is recommended that modern software containerization platform such as Docker (http://docker.com) is utilized to resolve difficulties of deploying software on systems with different environments. In addition, training sessions and descriptive manuals/tutorials are provided by ORNL to DOE analysts so that they can use the newly developed software more easily and independently for later usages.

6. FUTURE WORK

6.1 EXTENDING EFDW TO SUPPORT OTHER VISUALIZATIONS

In the current version of the tool, we implemented Sankey diagram as a specific case of EFDW workflow. Generalizing what we have accomplished, it is crucial to extend the concept of current versions EFDW to produce other kinds of visualization for energy-finance data. Like Sankey diagram, identification of data sources, and workflow to capture data provenance needs to be carefully considered.

6.2 ADVANCED EFDW REPOSITORY WITH SEARCH & NAVIGATE TOOL

Current version of EFDW uses the file system as a data repository; but to be able to achieve more advanced, search, navigate, and browse capability, it is necessary to utilize advanced database software and create appropriate indices for faster data lookups. On top of database services, implementing search and navigation user interface along with proper search result ranking function is crucial for users to identify data sources. In addition, allowing users to assign tags to data sources, capturing relationships between data sources, visualizing data sources will be very helpful for users.
This task involves with both backend and frontend software implementation.

6.3 ADVANCED DATA OPERATIONS FOR EFDW

Steps in EFDW workflow often involves with data operations such as joining table, schema mapping, data deduplication, data standardization, etc. So far, most of the tasks have been done manually by subject matter experts. Defining commonly used data operations for subject matter experts and providing systematic ways/tools will be helpful not only for speeding up the process by automation but also for tracking data provenance.

7. SUMMARY AND CONCLUSIONS

In this report, we described the Energy Finance Data Warehouse (EFDW) framework and its Sankey generator tool that we have developed to automatically display financial flows from consumers (end users such as residential, commercial etc.) to distribution, generation and transmission operators as a Sankey diagram. These tools can be utilized to provide a better understanding of how policy options can

financially affect various operators/sectors of the electricity system. Our approach also captures the methodology, calculations and estimations analysts used for the calculation as well as relevant sources so newer analysts can build on work done previously. We also summarized several challenges we faced during the project period and lessons learned from experiences along with recommendations so that can be helpful for future projects. As we could successfully use the tools for various use-case scenarios as described in separate documents, advancing EFDW data operations and visualization capabilities further is strongly recommended.

REFERENCES

[1] D. Csala, "Sankey Diagram Generator," 2014. [Online]. Available: http://sankey.csaladen.es.

[2] S. Bogart, "SankeyMATIC: A Sankey diagram builder for everyone," [Online]. Available: http://sankeymatic.com/.

[3] Google, "Google Charts," [Online]. Available: https://developers.google.com/chart/.

[4] J. Hunter, D. Dale, E. Firing and M. Droettboom, "matplotlib," 2016. [Online]. Available: http://matplotlib.org/.

[5] DOE, "THE QUADRENNIAL ENERGY REVIEW," 2015.

[6] Infocaptor, 2015. [Online]. Available: http://www.infocaptor.com/.

[7] "Regional transmission Organization," 2016. [Online]. Available: https://en.wikipedia.org/wiki/Regional_transmission_organization_(North_America).

[8] Tableau, "http://www.tableau.com/," 2016. [Online].

[9] B. Schniederman, "https://en.wikipedia.org/wiki/Treemapping," 2016. [Online].

[10] MySQL, "https://www.mysql.com/," 2016. [Online].

[11] "BlueJeans: Business Video Communications," [Online]. Available: https://www.bluejeans.com/.

[12] "Slack: Messaging for Teams," [Online]. Available: slack.com.