Decision Support Tool Suite Version 1

Deliverable D7.3.1

Version FINAL

Authors: W.R. Van Hage[1], T. Ploeger[1]

Affiliation: (1) SynerScope B.V.

Building structured event indexes of large volumes of financial and economic data for decision making

ICT 316404

Grant Agreement No.	316404
Project Acronym	NEWSREADER
Project Full Title	Building structured event indexes of large volumes of financial and economic data for decision making.
Funding Scheme	FP7-ICT-2011-8
Project Website	http://www.newsreader-project.eu/
Project Coordinator	Prof. dr. Piek T.J.M. Vossen VU University Amsterdam Tel. + 31 (0) 20 5986466 Fax. + 31 (0) 20 5986500 Email: piek.vossen@vu.nl
Document Number	Deliverable D7.3.1
Status & Version	FINAL
Contractual Date of Delivery	March 2014
Actual Date of Delivery	May 7, 2014
Type	Report
Security (distribution level)	Public
Number of Pages	53
WP Contributing to the Deliverable	WP7
WP Responsible	WP7
EC Project Officer	Susan Fraser

Authors: W.R. Van Hage[1], T. Ploeger[1]

Keywords: decision support, first version, knowledgestore exporter

Abstract: In this deliverable we describe the first version of the Decision Support Tool Suite (DSTS), the main point of end-user interaction with the event indexes extracted from news in the NewsReader project. We describe SynerScope, a visual analytics application that delivers real time interation with dynamic network-centric data. We describe the KnowledgeStore exporter which gets and converts the NewsReader data for loading into SynerScope. We also explain how to interact hands-on with the NewsReader data using SynerScope.

Table of Revisions

Version	Date	Description and reason	By	Affected sections
0.1	03 Mar 2014	Deliverable skeleton and content outline	T. Ploeger, W.R. van Hage	All
0.2	04 Mar 2014	First draft of 'Introduction' and 'The Decision Support Tool Suite'	T. Ploeger, W.R. van Hage	1, 2
0.3	05 Mar 2014	Expanded draft of 'The Decision Support Tool Suite', first draft of 'The NewsReader Data'	T. Ploeger, W.R. van Hage	2, 3
0.4	06 Mar 2014	First drafts of 'Importing NewsReader Data' and 'Interacting with NewsReader Data'	T. Ploeger, W.R. van Hage	4, 5
0.5	07 Mar 2014	Expanded draft of 'Interacting with NewsReader Data', added abstract and executive summary, ready for internal review	T. Ploeger, W.R. van Hage	5
0.6	19 Mar 2014	Internal review	T. Ploeger, W.R. van Hage	All

Executive Summary

The NewsReader project aims to support decision making by building structured event indexes of large volumes of news articles and financial data. The main point of interaction with the NewsReader event indexes will be the Decision Support Tool Suite (DSTS). The DSTS is a graphical user interface that is meant to support users when making strategic decisions. The DSTS does this by providing insight into the sequences of events that led up to a current situation so that a user can extrapolate to what might happen in the future.

The technical implemenation of the DSTS largely consists of the SynerScope tool, a visual analytics application that delivers real time interaction with dynamic network-centric data. SynerScope interfaces with the NewsReader KnowledgeStore through an export tool.

SynerScope allows the user to display several visualization methods ('views') simultaneously. Interactions with one view (e.g. highlighting, selecting) are replicated in all other open views. The Hierarchical Edge Bundling View (HEB) is the primary network visualization in SynerScope. Each Node is visualized as a point on a circle, and each Link is visualized as a curved line between its source and target Node. The Nodes are grouped hierarchically, based on an ordering defined by the user.

The NewsReader event data can be imported into SynerScope through a Web-based exporter tool. The exported data must then be imported into SynerScope, where it can be navigated through search, network expansion, and slicing and dicing to look for interesting patterns or irregularities.

Contents

List of Tables

1 Introduction

The NewsReader project aims to support decision making by building structured event indexes of large volumes of news articles and financial data. The purpose of this deliverable is to describe *the first version* of the Decision Support Tool Suite (DSTS), the main point of interaction with the NewsReader event indexes for the end user.

The *design* of the DSTS was described in detail in Deliverable 7.1. In this deliverable, we focus on the first version of the actual DSTS loaded with NewsReader data. We start by giving an overview of the functionality of the DSTS (Section 2) and the NewsReader data (Section 3) that can be loaded into the system. The next logical step is to show specifically how to import the NewsReader data into the DSTS (Section 4).

We then demonstrate how the NewsReader data can be interacted with once loaded into DSTS (Section 5) and subsequently conclude the deliverable (Section 6) with an overview of future work.

The authors have tried to make this deliverable as self-contained as possible, but will sometimes refer the reader to other project deliverables for more detailed information. Still, this deliverable should provide the reader with enough information to be able to use the first version of the DSTS with the NewsReader event data.

2 The Decision Support Tool Suite

The Decision Support Tool Suite (DSTS) is a visual analytics application that is meant to support users in making strategic decisions. The DSTS provides insight into the sequences of events that led up to a current situation, allowing the user to extrapolate to what might happen in the future. The system will not predict future events, but rather aid users in drawing their own conclusions.

Within the NewsReader project, the DSTS will interface with the KnowledgeStore (see Deliverable 6.2.1 for more details on the KnowledgeStore) and with external Web sources. The event data produced in other work packages is collected in the KnowledgeStore. The DSTS will import selections of this data for investigation. External Web-based knowledge sources that are not processed in other work packages can be accessed directly from the DSTS.

The technical implemenation of the DSTS largely consists of the SynerScope application (Figure 1), a visual analytics application that delivers real-time interaction with dynamic, network-centric data. SynerScope supports simultaneous views and coordinates user interaction, enabling identification of causal relationships and to uncover unforeseen connections. These features are explained in detail in the rest of this section.

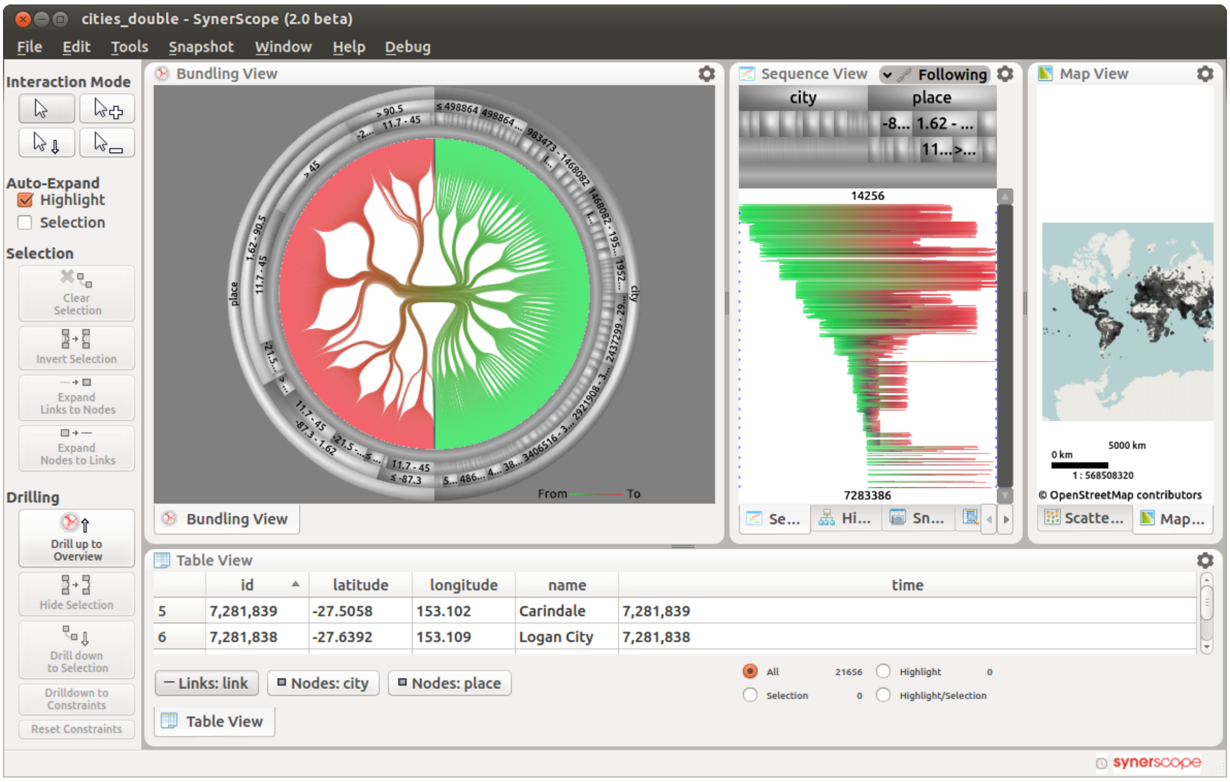


Figure 1: The SynerScope visual analytics application.

As stated in Section 1, the *design* of the DSTS - and thus also the design of SynerScope

Figure 2: Example of nodes representing persons.

- was described in Deliverable 7.1. For more details on the design principles that lie at the foundation of SynerScope, its visualizations, and its interaction methods, the authors refer the reader to that deliverable.

In earlier deliverables, 'SynerScope' was called 'SynerScope Marcato'. This name is no longer used. From this point forward the authors may also use the name 'SynerScope' to refer to the DSTS, as they are in essence one and the same system.

In the coming subsections, we start by giving an overview of the primary concepts used in SynerScope (Section 2.1). We continue with the data workflow used in SynerScope (Section 2.2). We then show how data may be visualized (Section 2.3) and interacted with (Section 2.4). We conclude with how SynerScope can be deployed (Section 2.5).

2.1 SynerScope Overview

SynerScope is designed to work with a very basic information schema, called the SynerScope Interface Schema (SIS). SIS consists of two object types: nodes and links.

Nodes are data entities under investigation, such as persons, bank accounts, IP addresses, insurance policies, or any other kind of entity that can be related to another entity by means of a link. An example is shown in Figure 2.

A **link** connects two nodes and indicates a directional event or relationship from one node to another. Links are also referred to as relationship, transaction, or event. Examples are phone calls, bank transactions, and network communication messages. Links often have a temporal attribute that indicates a link's occurrence in time. An example is shown in Figure 3.

Both nodes and links can have additional attributes of a number of data types, including integers, floating point numbers, free text, date and time, latitude and longitude.

A **hierarchy** is a classification of nodes by repeatingly dividing nodes into subcategories based on the values or bins of a single node attribute. In case of geographical data - where nodes represent cities, a valid hierarchy (shown from top to bottom) would be: continent - country - city. A city node is correctly placed within its hierarchy based on its continent and country attributes. An example is shown Figure 4.

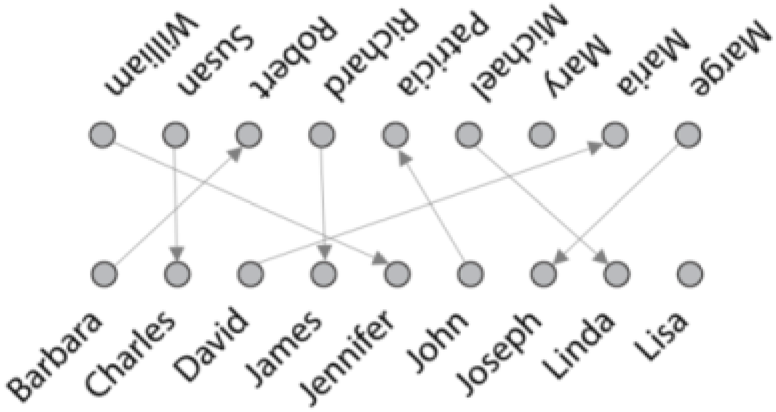


Figure 3: Example of links between persons representing phone calls.

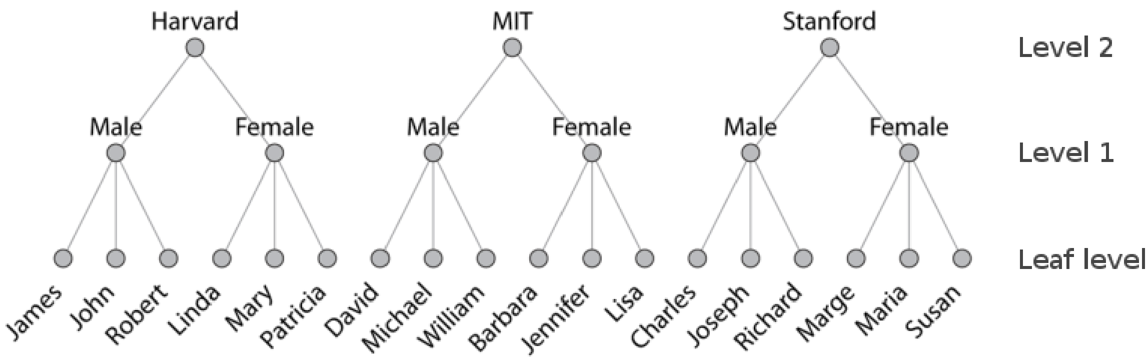


Figure 4: Example hierarchy on top of nodes representing persons, with gender at the first level and institute at the second level.

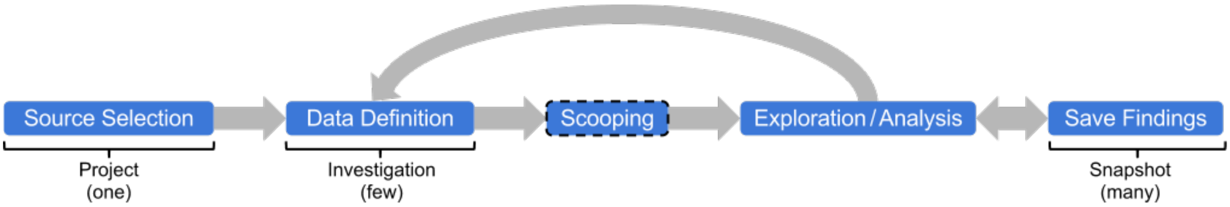


Figure 5: Overview of the SynerScope data workflow.

2.2 Data Workflow

SynerScope gathers node-link data from tabular data sources (e.g. relational databases or CSV files) using the following workflow:

First, a **project** is created to select the data source. Then with an **investigation** the node type(s) and link type are defined. Optionally, prefilters are defined in the investigation which filter the data from the original data source and transfers this filtered data to a local database using scooping. At this moment the data is available in the SynerScope tool and is ready to be analyzed. When a finding is discovered it can be saved using **snapshots**. An overview of this workflow is shown in Figure 5.

This workflow remains more or less the same (at least for this first version of the DSTS) when importing data from the NewsReader KnowledgeStore, as we will see in Section 4, where step-by-step example of the workflow using actual NewsReader data is included.

2.3 Visualization Techniques

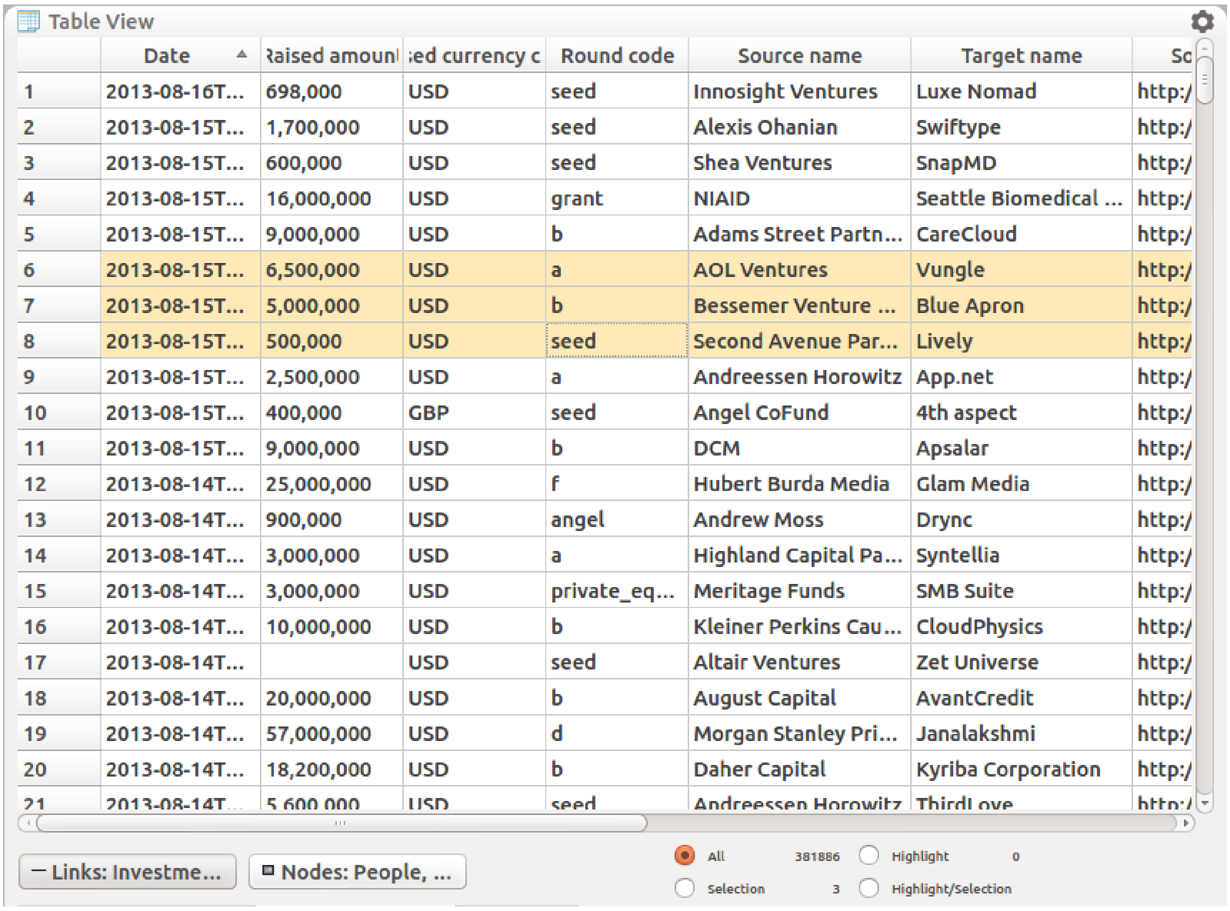
After constructing a new or opening an existing investigation, the user has access to all visualizations of SynerScope. The visualizations take advantage of hardware-accelerated rendering. Each visualization is described in detail in its own subsection below.

2.3.1 Table View

The Table View (Figure 6) provides a traditional spreadsheet view on the data. There is a separate sheet for nodes and a separate sheet for links. The Table View shows all the data as a table of values. Rows in the table correspond to either Links or Nodes, depending on which sheet is selected. The columns represent the attributes of the Nodes or Links. The Table View can be used to sort Nodes or Links by the values of a certain field by clicking the header cells of the columns in the table. Furthermore, from the Table View, the user can export the current Selection to a spreadsheet file for later processing with other analysis software such as Microsoft Excel, SAS, or R.

2.3.2 Hierarchical Edge Bundling View

The Hierarchical Edge Bundling View (HEB) (Figure 7) is the primary network view in SynerScope. Each Node is visualized as a point on a circle, and each Link is visualized as



	Date	Raised amount	Raised currency c	Round code	Source name	Target name	So
1	2013-08-16T...	698,000	USD	seed	Innosight Ventures	Luxe Nomad	http:/
2	2013-08-15T...	1,700,000	USD	seed	Alexis Ohanian	Swiftype	http:/
3	2013-08-15T...	600,000	USD	seed	Shea Ventures	SnapMD	http:/
4	2013-08-15T...	16,000,000	USD	grant	NIAID	Seattle Biomedical ...	http:/
5	2013-08-15T...	9,000,000	USD	b	Adams Street Partn...	CareCloud	http:/
6	2013-08-15T...	6,500,000	USD	a	AOL Ventures	Vungle	http:/
7	2013-08-15T...	5,000,000	USD	b	Bessemer Venture ...	Blue Apron	http:/
8	2013-08-15T...	500,000	USD	seed	Second Avenue Par...	Lively	http:/
9	2013-08-15T...	2,500,000	USD	a	Andreessen Horowitz	App.net	http:/
10	2013-08-15T...	400,000	GBP	seed	Angel CoFund	4th aspect	http:/
11	2013-08-15T...	9,000,000	USD	b	DCM	Apsalar	http:/
12	2013-08-14T...	25,000,000	USD	f	Hubert Burda Media	Glam Media	http:/
13	2013-08-14T...	900,000	USD	angel	Andrew Moss	Drync	http:/
14	2013-08-14T...	3,000,000	USD	a	Highland Capital Pa...	Syntellia	http:/
15	2013-08-14T...	3,000,000	USD	private_eq...	Meritage Funds	SMB Suite	http:/
16	2013-08-14T...	10,000,000	USD	b	Kleiner Perkins Cau...	CloudPhysics	http:/
17	2013-08-14T...		USD	seed	Altair Ventures	Zet Universe	http:/
18	2013-08-14T...	20,000,000	USD	b	August Capital	AvantCredit	http:/
19	2013-08-14T...	57,000,000	USD	d	Morgan Stanley Pri...	Janalakshmi	http:/
20	2013-08-14T...	18,200,000	USD	b	Daher Capital	Kyriba Corporation	http:/
21	2013-08-14T...	5,600,000	USD	seed	Andreessen Horowitz	ThirdLove	http:/

Figure 6: Table View.

a curved line between its source and target Node. The Nodes are sorted so that changes in the network over time have a minimal impact on the location of the Nodes on the circle. This makes for a stable view of the network.

The Nodes are grouped hierarchically, based on the ordering defined in the Hierarchy . The Links between Nodes of the same hierarchical category are bundled together (as if they were tied together with a cable tie). The thickness of the bundle is caused by overlapping lines (each line is still shown individually) and shows the amount of interaction between categories. Bundles can be selected by making a slicing gesture by clicking and dragging the mouse over a bundle.

2.3.3 Massive Sequence View

The Massive Sequence View (MSV) (Figure 8) is the primary temporal view in SynerScope. Each Node gets a fixed position on the horizontal axis. Nodes are grouped hierarchically in the same fashion as in the HEB. Links between Nodes are represented by a horizontal line between the respective positions of the Nodes. On the vertical axis the user can select a scalar attribute, typically a time or date. This orders the Links temporally. The MSV is typically used to inspect patterns in time. When certain categories of Nodes interact with each other in some temporal pattern, this becomes instantly recognizable in the MSV.

2.3.4 Map View

The Map View (Figure 9) is the primary spatial view in SynerScope. The user can select two attributes from any Node or Link data source to interpret as WGS84 latitude and longitude coordinates. These attributes are used to plot the Nodes (not the Links) on a map as points. The points are plotted in such a way that the user can see the point density. An additional scalar attribute can be selected to indicate point size on the map.

2.3.5 Scatter Plot View

The Scatter Plot View (Figure 10) uses Cartesian coordinates to relate the values of two attributes of either Nodes or Links. Dots are drawn on a two-dimensional chart, the positioning relative to the horizontal and vertical axis being determined by the attribute's values. A third attribute can used to set the size of the dots.

2.3.6 Other Views

In addition to the visualization views described above, SynerScope also has several 'supporting' views:

Search and Filter The Search and Filter View is an interactive view that allows the user to select Nodes or Links by searching by value. The user can type in a search query that is matched against the value of a certain attribute for a given Node or Link type.

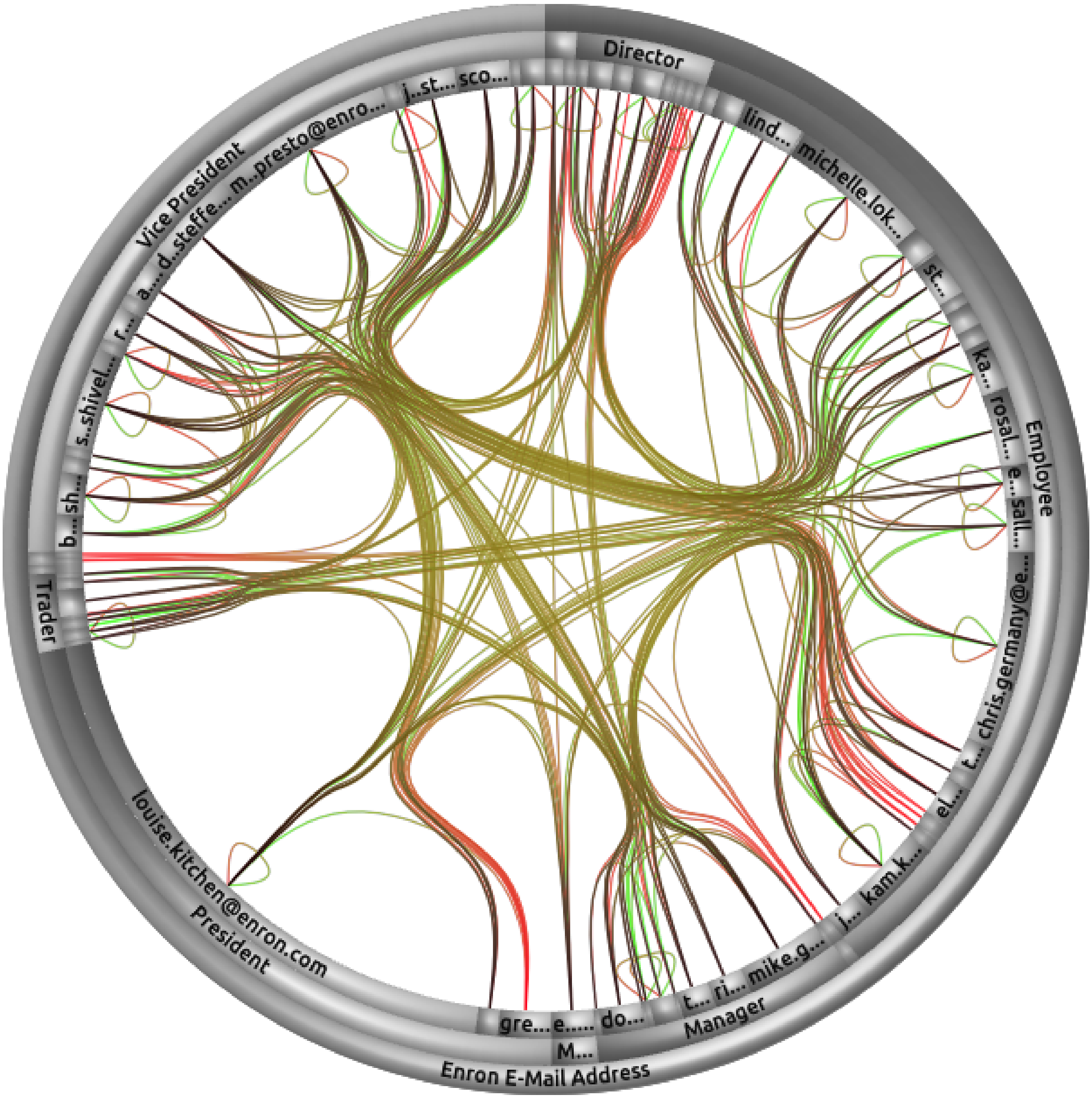


Figure 7: Hierarchical Edge Bundling View.

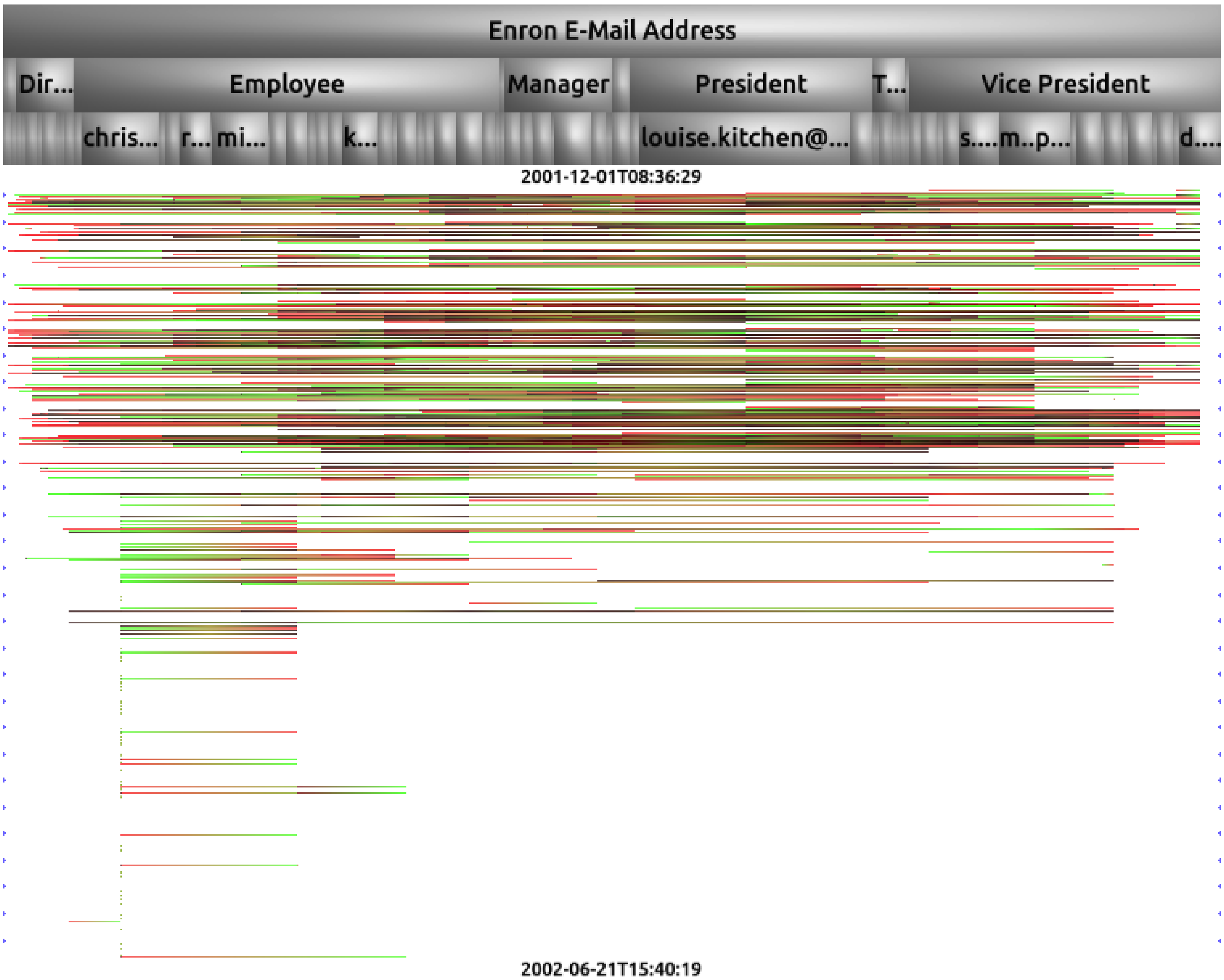


Figure 8: Massive Sequence View.

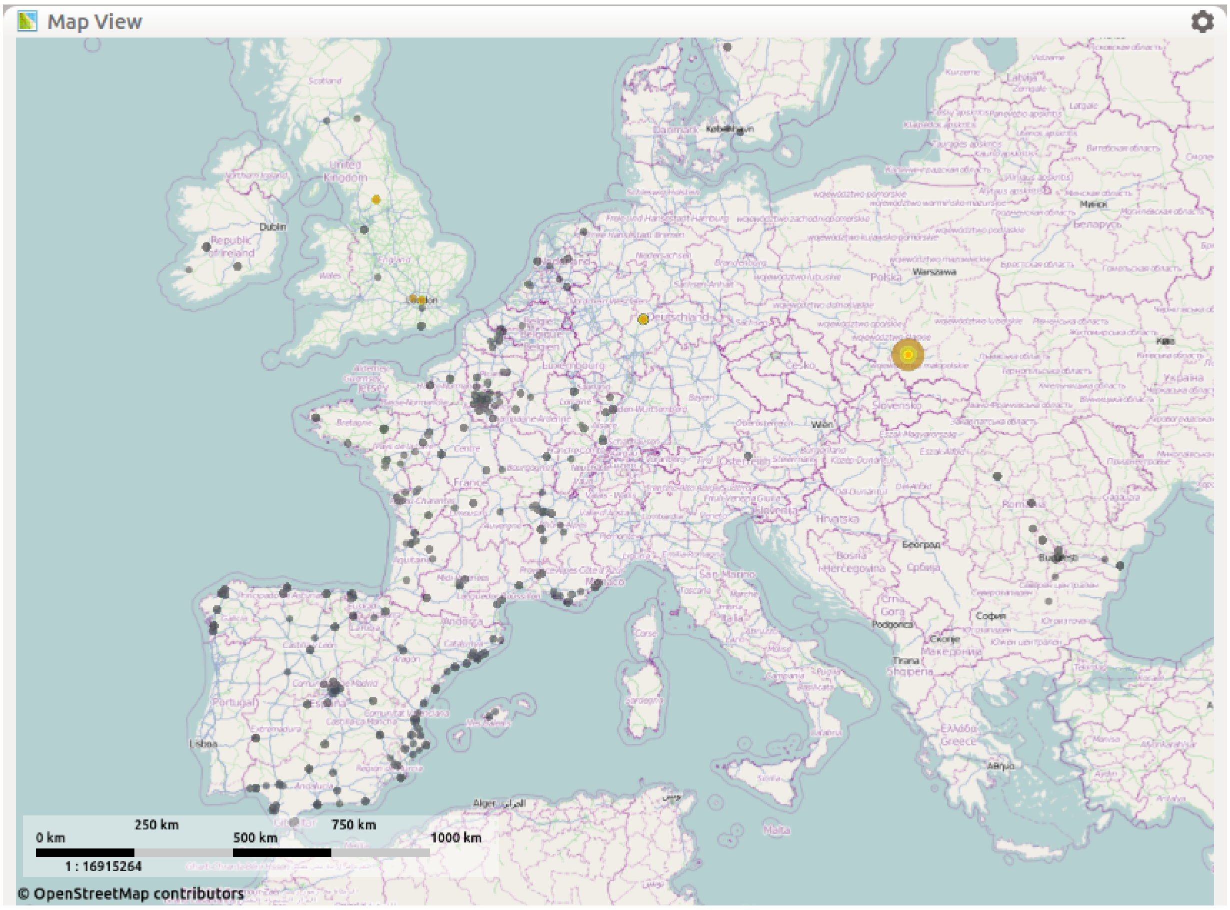


Figure 9: Map View.

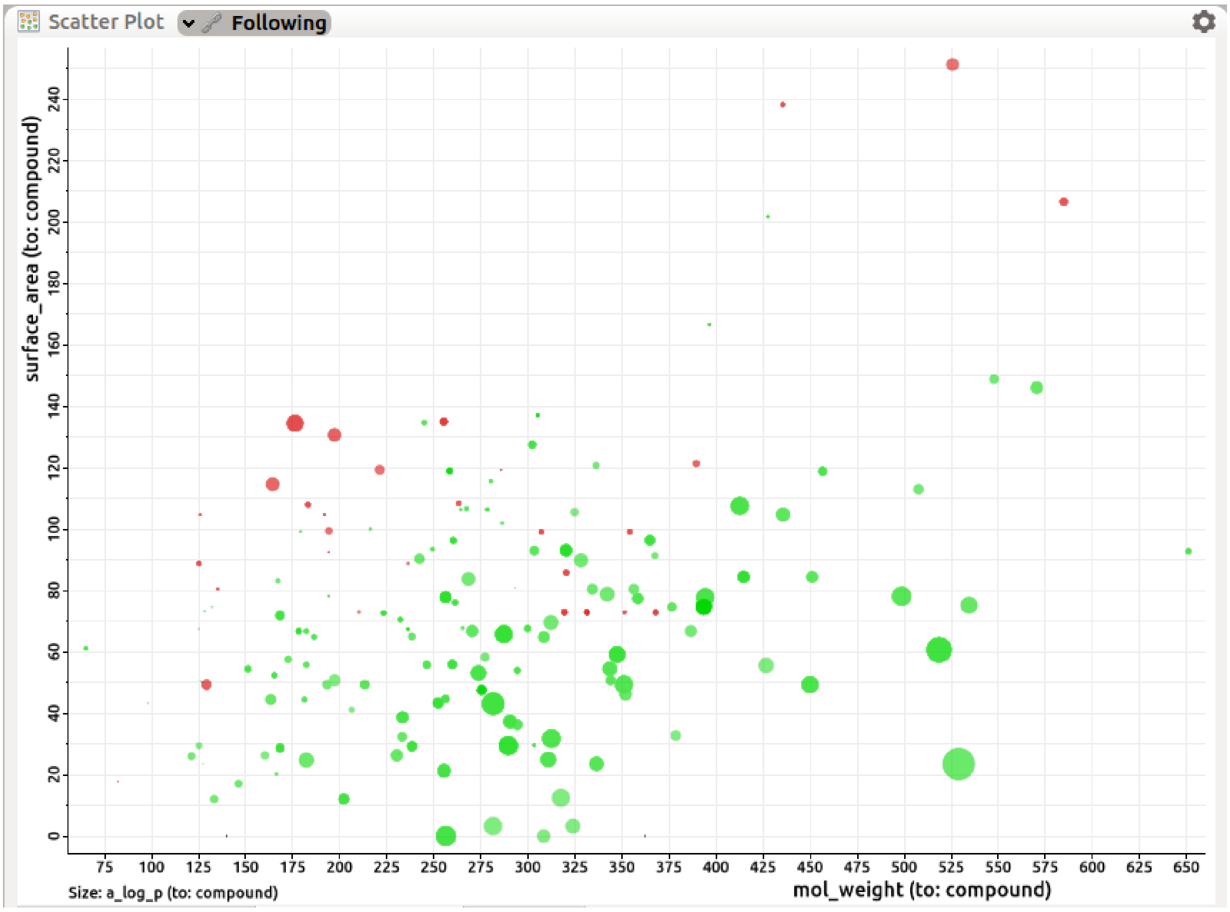


Figure 10: Scatter Plot View.

Hierarchy Editor In the Hierarchy Editor View it is possible to define a hierarchical ordering (of arbitrary depth) on nodes. Each layer of the hierarchical ordering groups nodes together based on some common attribute value.

Web View The Web View is used to display user-developed, specialized views created with HTML and JavaScript.

2.4 Interaction Methods

The user can interact with SynerScope views in several ways: By **selecting** and **highlighting**, **drilling down** or **up**, and **expanding** selections. Every interaction method is **coordinated** across multiple views and over the node hierarchy. This principle, together with the details of each interaction method, is described below.

The actual buttons to use these interaction methods are located in the left sidebar of SynerScope's main window and are explicitly labeled with their function.

2.4.1 Selection and Highlight

SynerScope supports the combined use of Selections and Highlights. When a user hovers over a certain Node, Link, or category/bundle of Nodes or Links, SynerScope highlights these objects with a color (Figure 12) that makes them stand out from the other objects. When a user confirms that the Highlight is of interest by clicking, the Highlight turns into a more static Selection. The Selection stays the same until a new Selection is made. The Selection gets a different color (Figure 11) than the Highlight, which allows the user to contrast the two sets to see commonalities and differences. Any overlap between the Selection and the Highlight get their own color that makes it stand out from the other objects.

2.4.2 Drill Down/Up

Drilling down rescopes the domain of investigation to the current Selection. Essentially, it allows the user to zoom in on an interesting part of the data set to get a more detailed view. SynerScope always shows all the data, therefore drilling down does not add more detail, but by hiding the rest of the data it provides the Selection with more room to be visualized, which provides a clearer view of the individual data points.

Drilling up performs the converse operation. It rescopes the domain of investigation to the entire data set, preserving the current Selection. This can allow the user to select small parts of the data that are of interest in a Drilled down view, after which the user can Drill up to see the context of the Selection in the entire data set.

2.4.3 Expanding

Network expansion allows the user to extend the currently Selected set of Nodes or Links with respectively Links or Nodes that are directly connected. This way it is possible to

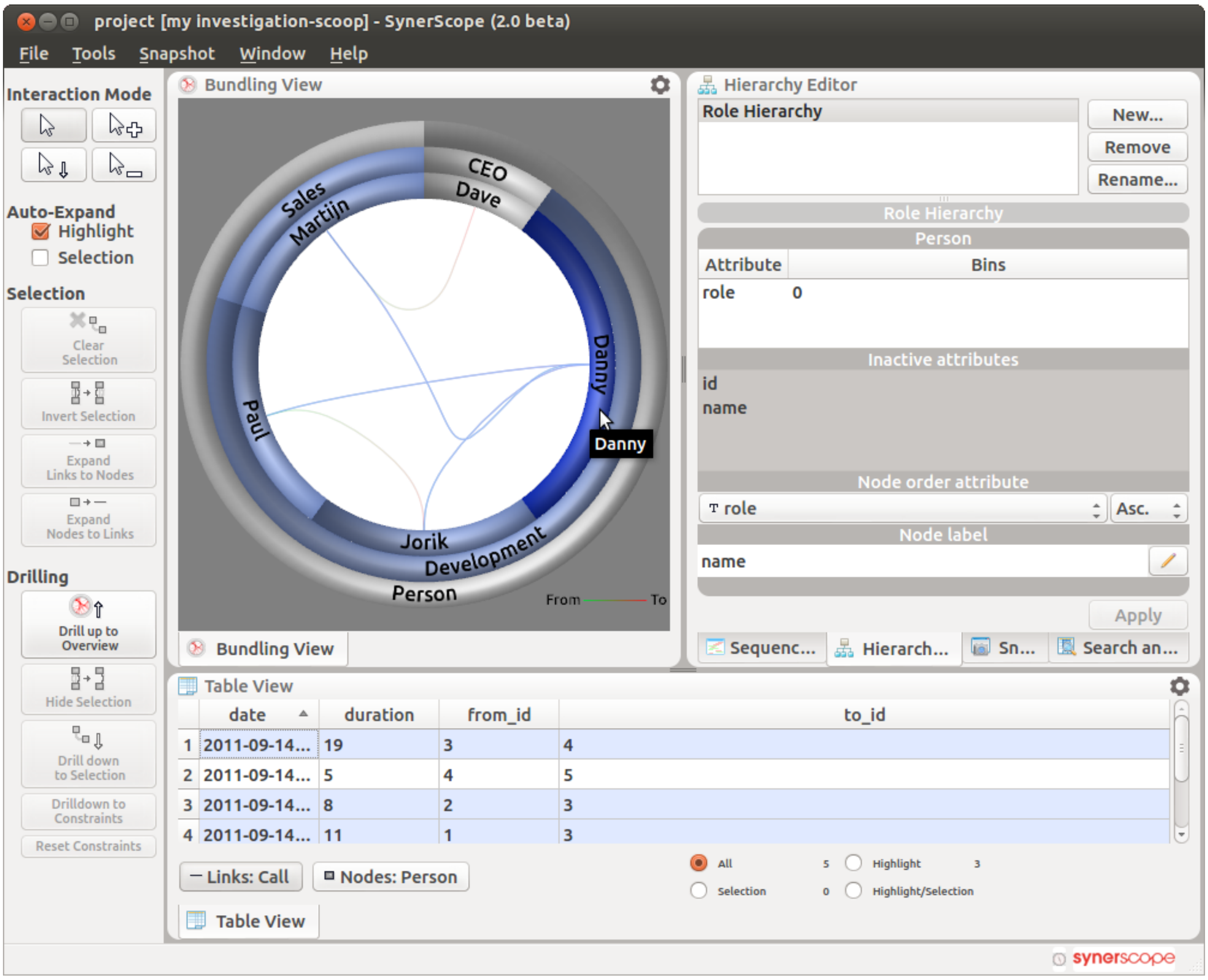


Figure 11: Highlight.

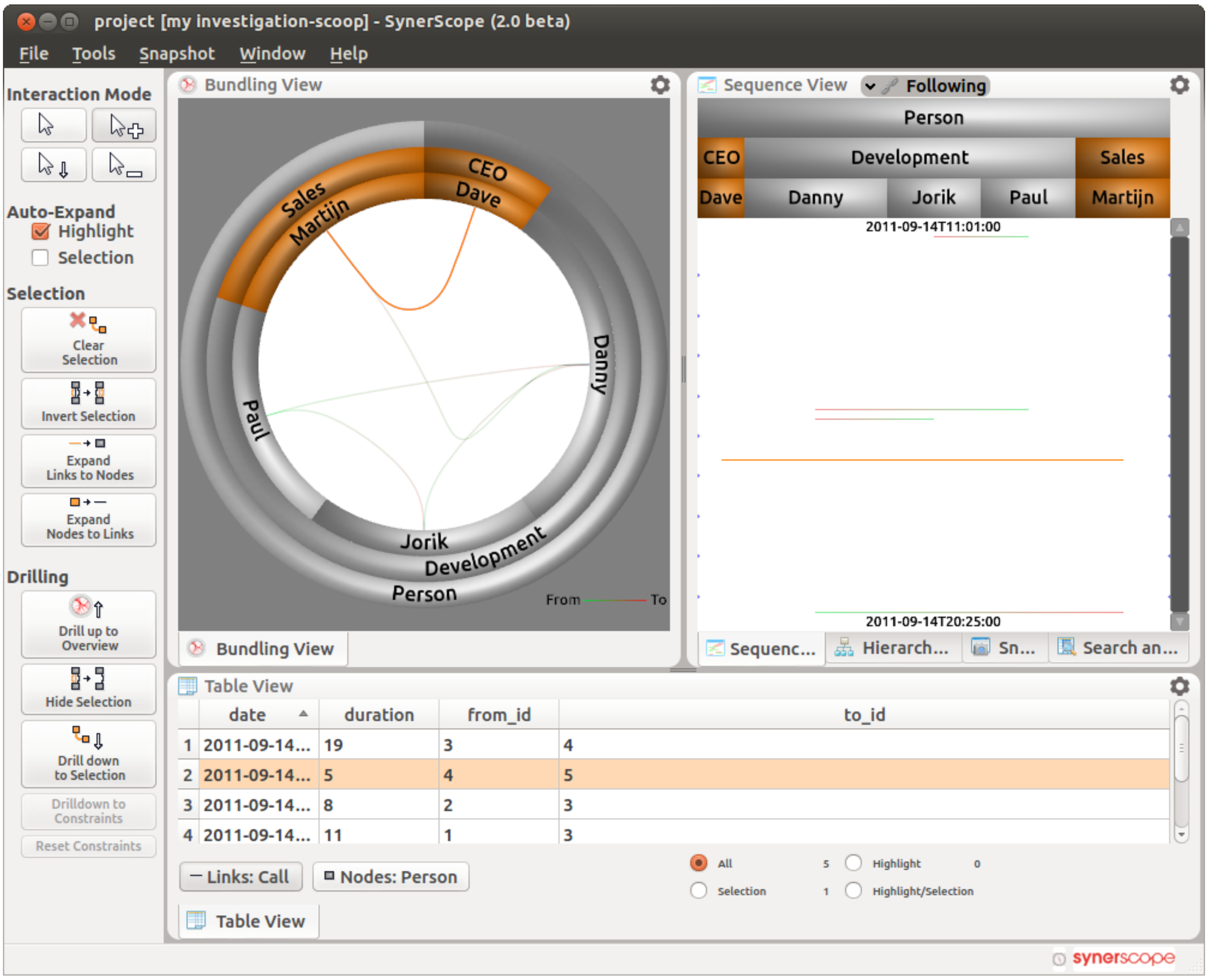


Figure 12: Selection.

let the entire Selection grow by 'hopping' over the Link network. Network connectivity, or the lack of connectivity, can be investigated in this way.

Drilling down provides more detail, but it also cuts away the context. Therefore another common use of expansion is to gather the local network context of a Selection of Nodes before Drilling down. This can provide the user with a minimal relevant collection of contextual information to judge the patterns in the Drilled down view.

A number of fine-grained expansion tools are also available, such as an 'internal' expansion within the domain of the current selection. This expands the currently Selected set of Nodes and Links with all the Links between currently selected Nodes that are not already selected.

2.4.4 Multiple Coordinated Views

The central interaction paradigm of SynerScope is its **Multiple Coordinated Views**. SynerScope can show different perspectives on the same data simultaneously using the different visualization techniques described in Section 2.3. A selection or highlight made in one of the views causes an equivalent selection to be made in all other views.

An example of this principle can be seen in Figure 11 and 12, where a highlight and selection in the HEB are propagated to the Table View below the HEB (and any other open views).

This principle enables the user to explore correlations between different facets of the data. For instance, it can be used to explore whether actors that interact are geographically colocated by selecting part of a social network in a network view, then checking which corresponding locations are selected in a geographical view and whether these are near to each other.

Each view lives in a 'pane' within the SynerScope main window. In the bottom right corner, the user can select several different pane layouts, such as one big pane or two smaller panes side-by-side. One pane can have multiple views. In that case, each view lives in its own tab (just like a modern Web browser) but only one of them can be visible. Each pane has its own button for opening a new type of view within that pane.

2.5 Deployment

SynerScope can be run locally on the end users own machine, if said machine meets the minimum hardware and software requirements specified below. Alternatively, SynerScope can be run remotely on a virtual machine in the cloud and streamed to the end users machine.

2.5.1 Local Deployment

For a dataset consisting of one hundred thousand Nodes and one million links, SynerScope runs smoothly on a machine that meets or exceeds the following specifications:

Processor Intel Core i5 or equivalent. Intel Core i7 is recommended.

Memory At least 2GB–4GB RAM is recommended for large datasets.

Screen Resolution At least 1024x768, 1440x900 or higher is highly recommended.

Graphics An NVIDIA Quadro 4000 or equivalent graphics card is recommended, although experimental support for consumer-level graphics (NVIDIA GeForce) is present.

2.5.2 Cloud Deployment

SynerScope can run in a “private” cloud environment, as a graphically accellerated virtualized application. This allows users to user the Decision Support Tool Suite from their own computer regardless of the technical specification, even a tablet computer is sufficient, as long as the network access to the cloud server is sufficiently stable.

A setup in a public cloud environment is also possible. Specifically, the authors have had some success in running SynerScope on graphically accelereated virtual servers on Amazon Web Services[1] (AWS).

[1] http://aws.amazon.com

3 The NewsReader Data

The NewsReader linguistic pipeline is fed raw news articles and outputs layered event data. This event data is then stored in the KnowledgeStore[2]. The Decision Support Tool Suite then imports the data from the KnowledgeStore and visualizes it.

This section is only a succinct description of the data formats used in NewsReader. The goal is to demonstrate the origins and properties of the data that can be loaded and visualized by the DSTS. For in-depth details on the linguistic pipeline and its output formats, see Deliverable 4.2.1. and Deliverable D 5.1.1

3.1 Data Formats Overview

The NewsReader linguistic pipeline takes raw text as an input. The result consists of multiple layers of raw linguistic information in the NLP Annotation Format[3] (NAF). This data is then transformed into actual event data by converting it to the Simple Event Model Plus[4] (SEM+). The Grounded Annotation Framework[5] (GAF) is used to link event instances (and their constituents) to their original mention in the raw text from which they were extracted. Each of these data formats is described in their own subsection below.

3.2 NAF

The Named Annotation Format represents linguistic annotations. NAF is an XML-based, multi-layered format. Each individual module of the linguistic pipeline adds its own layer without touching the data added by other modules. NAF consists of the following 10 layers.

The *raw layer* contains the raw input text. The *text layer* contains the individual tokens of the raw input text. The *terms layer* contains the individual words and multi-words. The *chunks layer* contains chunks such as noun phrases and prepositional phrases. The *dependency layer* contains relations between words. The *entity layer* contains entity mentions such as persons, organizations. The *coreference layer* contains terms that refer to entities. The *semantic role layer* contains roles of entities. The *time expression layer* contains time expressions. Finally, the *factuality layer* contains information on whether events mentioned in the text actually happend, did not happen, or whether this is uncertain. Examples of each of these layers are shown in Deliverable 2.1.

3.3 SEM+

NAF's raw linguistic data is interpreted by a probabilistic, rule-based system to generate actual event data in the Simple Event Format Plus.

[2]For in-depth details on the KnowledgeStore, see Deliverable 6.2.1.
[3]http://wordpress.let.vupr.nl/naf
[4]http://wordpress.let.vupr.nl/sem
[5]http://groundedannotationframework.org

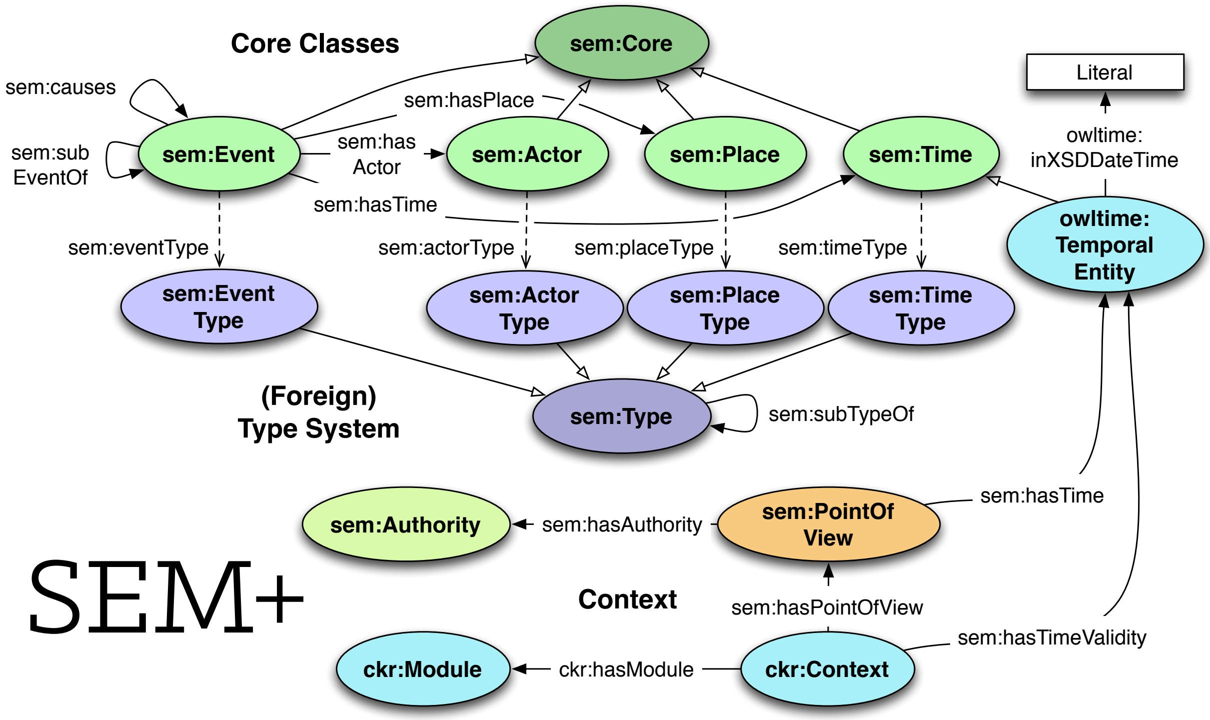


Figure 13: The Simple Event Model Plus.

SEM+ is designed to represent events in the broad sense of the word. They may be derived from various sources (from the web, sensory data, historical documents, etc.). They may also be incomplete (e.g. missing values) and partial (e.g. missing entire facets).

More specifically, SEM+ models events in terms of “who did what to whom, where and when?”, respectively modeled as **Actors**, **Events**, and **Places**, which are all **Time**-stamped. Figure 13 gives an overview of these core classes together with their relationships. Each of the classes may have a type, which may be specified by a foreign type system. A unique feature of SEM is that it allows specifying multiple views on a certain event, which hold according to a certain authority.

3.4 GAF

The Grounded Annotation Framework links the event data of SEM+ to mentions in the the original text the events were denoted by and derived from using the linguistic pipeline. In essence, GAF consists of ‘denoted by’ links from SEM+ events back to raw linguistic data of NAF.

GAF groups information based on provenance and (temporal) validity. The former allows any user of the event data to exactly see how and from what basis it was generated. The latter enables the representation of conflicting information, such as 2 events derived from 2 news articles by different authors.

GAF also allows linking of event data to non-textual sources. Events are not only described in textual documents, they are also represented in many other non-textual sources. These sources include videos, pictures, sensors or evidence from data registration such as mobile phone data, financial transactions and hospital registrations.

4 Importing NewsReader Data

For data to be imported into the DSTS, it first has to be exported from the KnowledgeStore and transformed to the SynerScope Interface Schema (SIS) (see Section 2 for more details on this schema).

As a part of the first version of the DSTS a basic Web application has been built to perform the export and transformation steps of this process. This allows SynerScope's existing importer to be used.

For this first version of the DSTS, of the 3 data layers mentioned in Section 3, we use the SEM+ layer of event data and part of the GAF provenance data. In future revisions of the DSTS, we will use all 3 data layers.

After describing how the KnowledgeStore exporter works and how to operate it, we show how to import the exported data using SynerScope's importer.

4.1 KnowledgeStore Exporter

The KnowledgeStore exporter (Figure 14) is a Web application written in Python[6] using the Flask[7] Web framework. The exporter can run locally on the users own machine, or it can be hosted on a server so it can be used by multiple users simultaneously.

Using the exporter is very straightforward. First, the user must select an actor he or she is interested in. An autocomplete function helps the user find the relevant actor. Second, the export must be limited to a certain number of events. Upon pressing the 'Export nodes and links'-button, the exporter goes to work and will eventually return a .zip archive. This archive contains 2 CSV-files formatted in SIS: One contains nodes, and the other contains links.

The conversion to CSV performed by the importer for this first version of the DSTS is very simple. The importer first queries the KnowledgeStore for the RDF[8] representation of the events.

Each of the triples returned by the KnowledgeStore in response to this query are separated into their subject, predicate, and object. Subjects and objects become nodes in SIS and are added to the nodes CSV file. Predicates naturally map to the links between the nodes and therefore become links, which are added to the links CSV file.

We will demonstrate how to import these CSV's in Section 4.2.

The export must be constrained by a certain actor and a maximum number of events because there is such a large amount of events in the KnowledgeStore (approximately 1.783.991 at the time of writing), that it is impossible to load them all into this first version of DSTS at once.

The authors recommend that users export 5000 events at most. Future revisions of the DSTS will most likely be able to support larger numbers of events. Future revisions of the exporter will support additional constraints, such as on place and time. We are also

[6]http://www.python.org
[7]http://flask.pocoo.org
[8]http://www.w3.org/RDF

looking into ways of reducing the number manual steps required to import KnowledgeStore data into the DSTS.

Figure 14: The KnowledgeStore Exporter.

4.2 Example Import

We start our example by creating an export with events about the actor 'Volkswagen' limited to 500 events, using the steps described in Section 4.1.

Now, start SynerScope. You will be greeted by the project overview (Figure 15). Let us create a new project for the data we just exported.

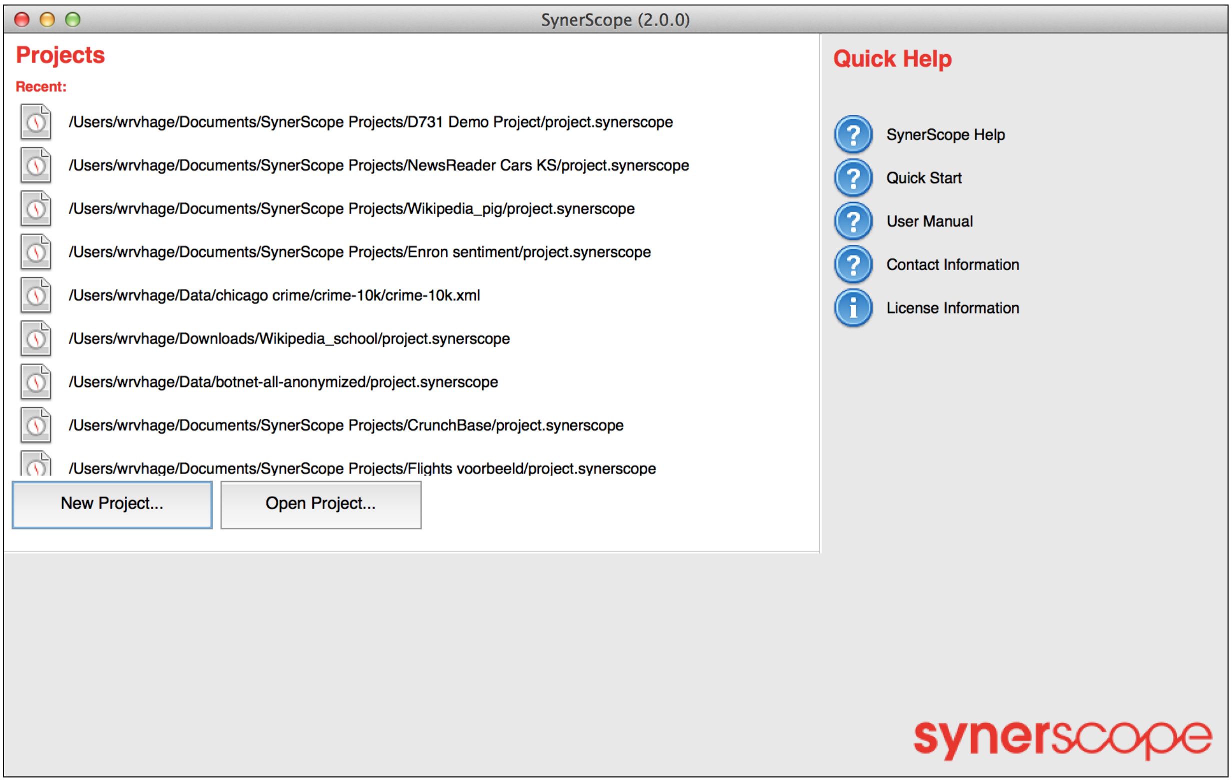


Figure 15: SynerScope project overview.

Upon clicking the 'New Project' button, you will be able to specify a name and a location on your computer for the project (Figure 16). Choose something that you like.

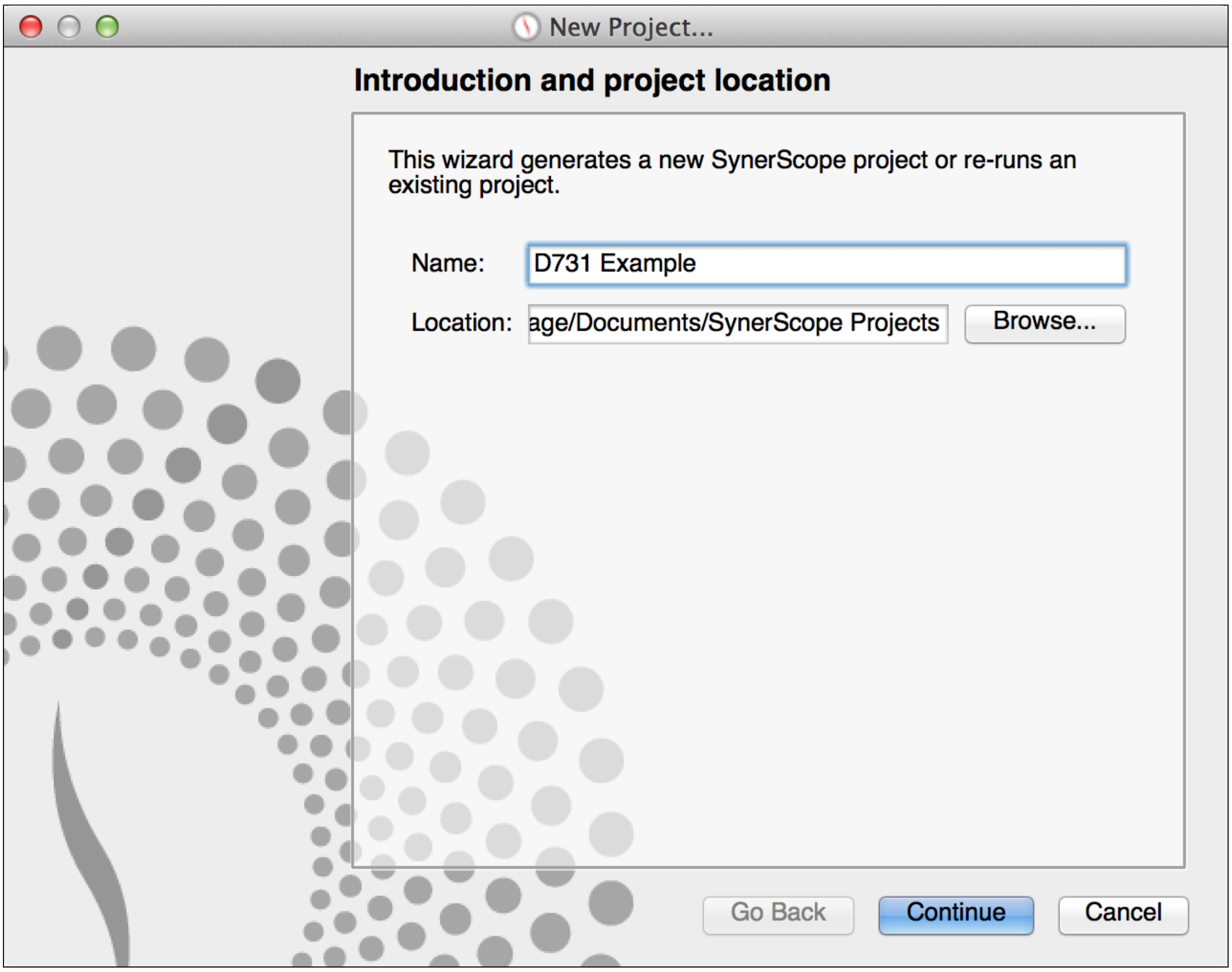


Figure 16: Create a new SynerScope project.

Next, SynerScope asks you whether you would like to use a database or CSV files as data for this project (Figure 17). In this case, we select CSV-files.

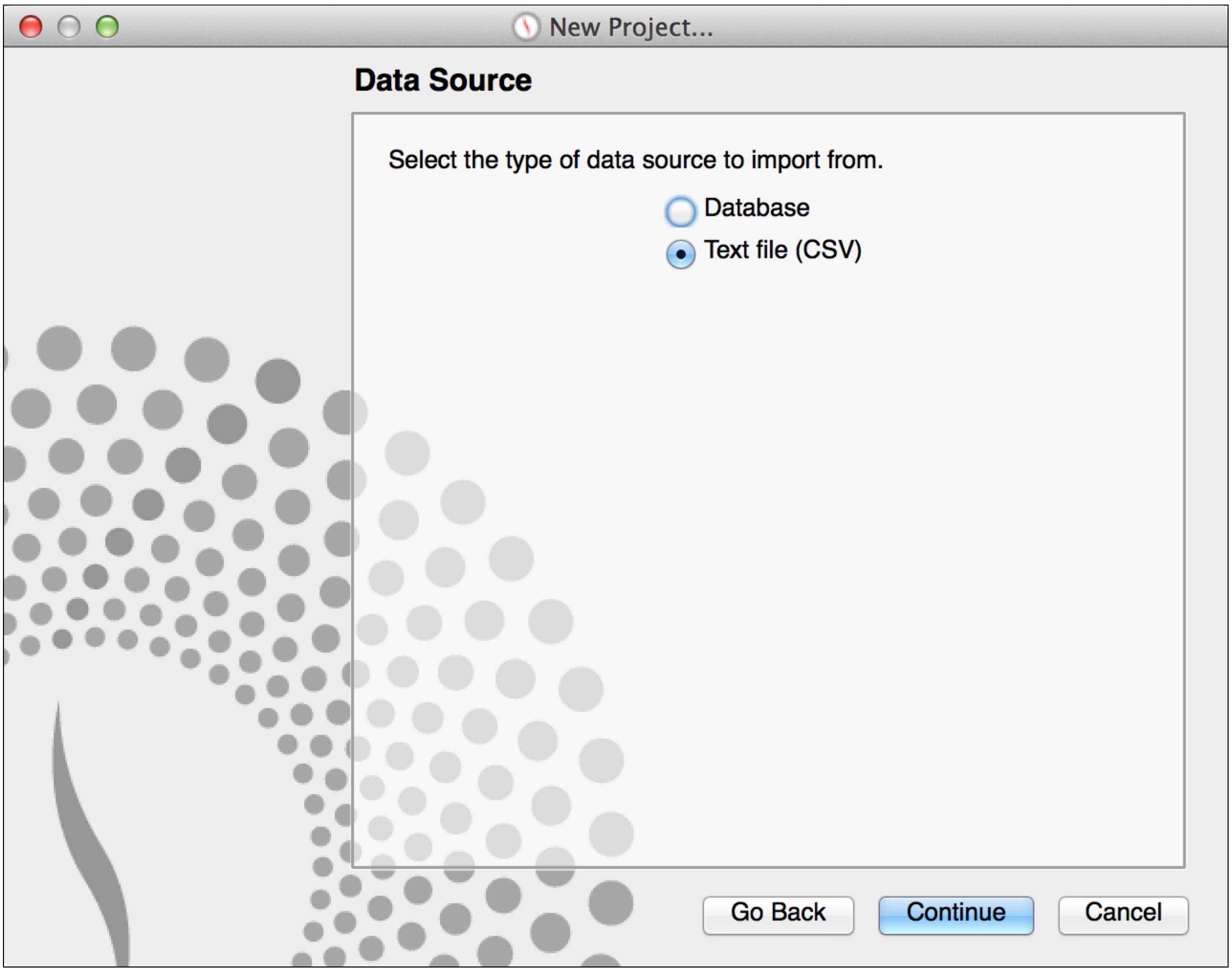


Figure 17: Selecting the data source for a new SynerScope project.

SynerScope will then ask you where to store the internal database for this project (Figure 18). You should not need to change anything here.

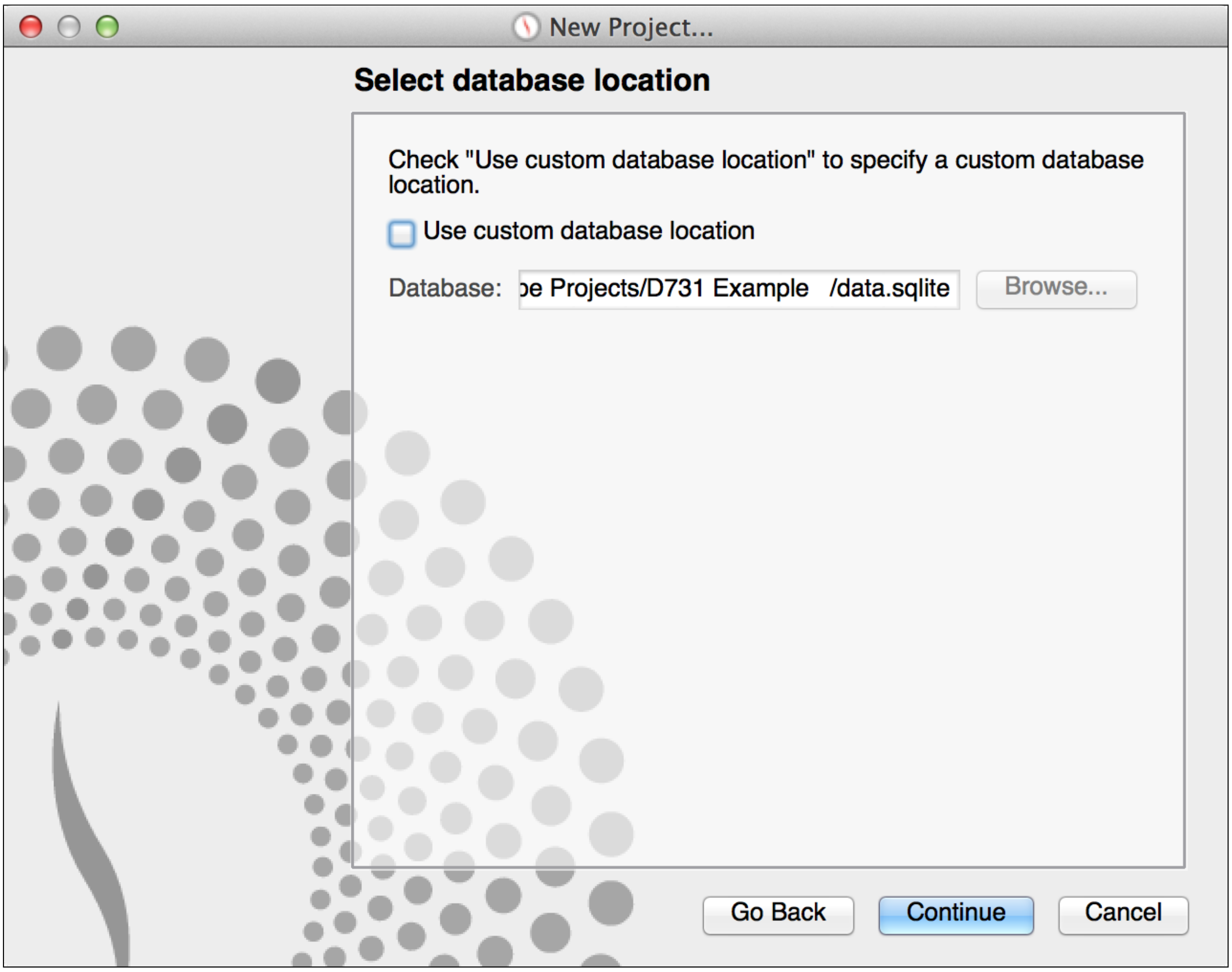


Figure 18: Selecting the data source for a new SynerScope project.

You have successfully created a SynerScope project. Click 'Done' (Figure 19).

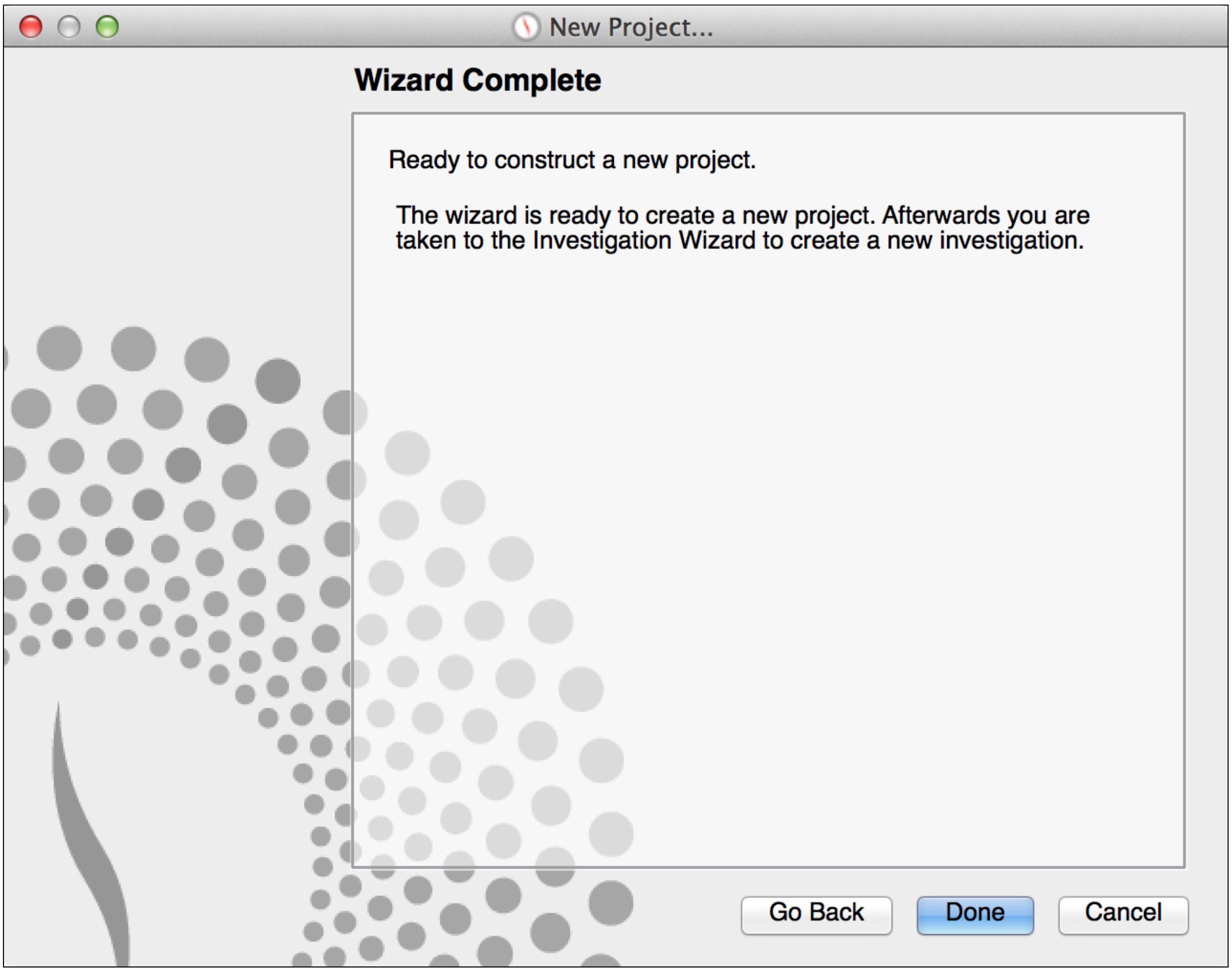


Figure 19: Successfully created a SynerScope project.

The wizard will now ask you to create an investigation within your new project. Pick a name, leave the template drop-down blank, and continue (Figure 20).

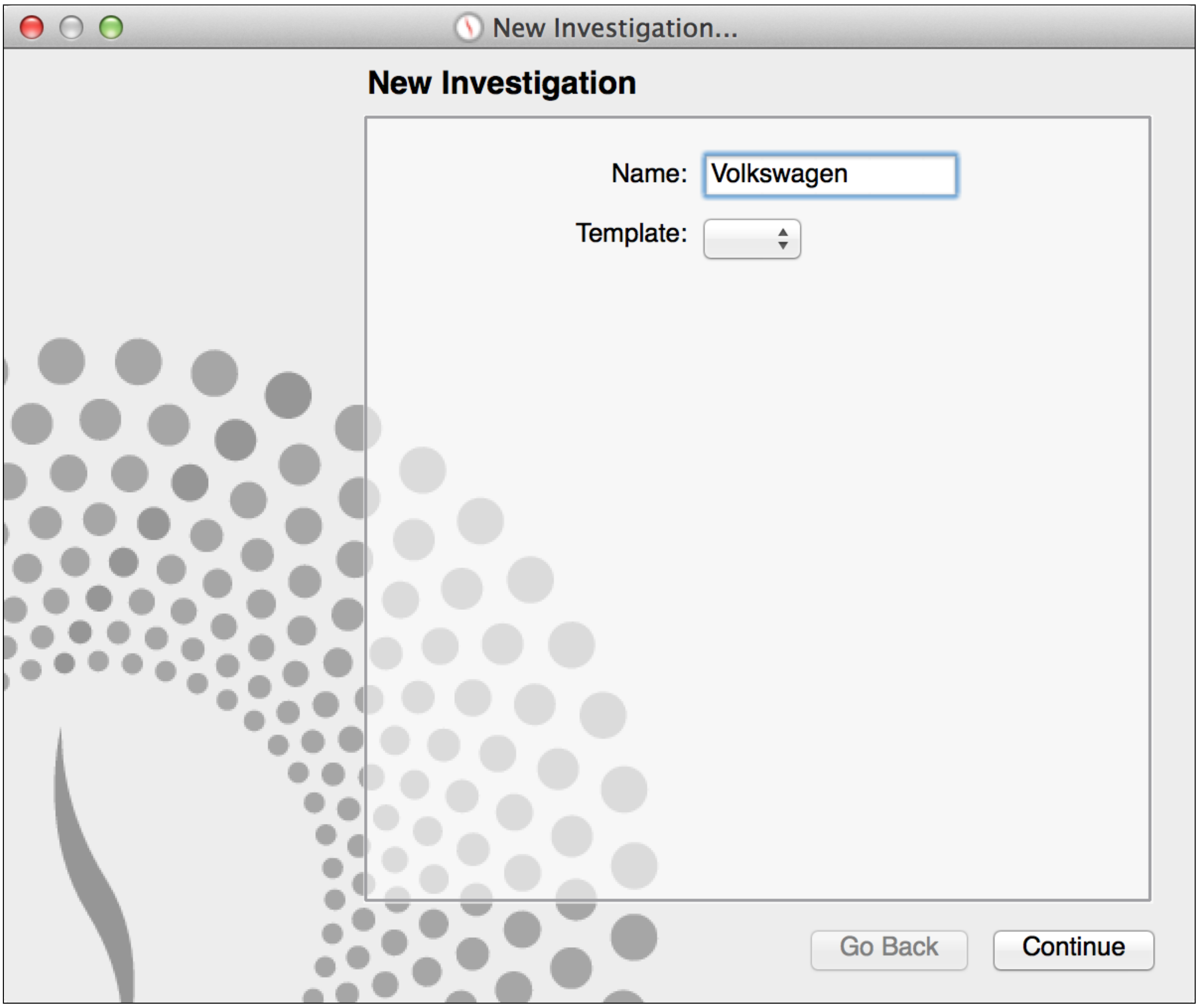


Figure 20: Creating an investigation within a project.

You have successfully created an investigation. Click 'Done' (Figure 21).

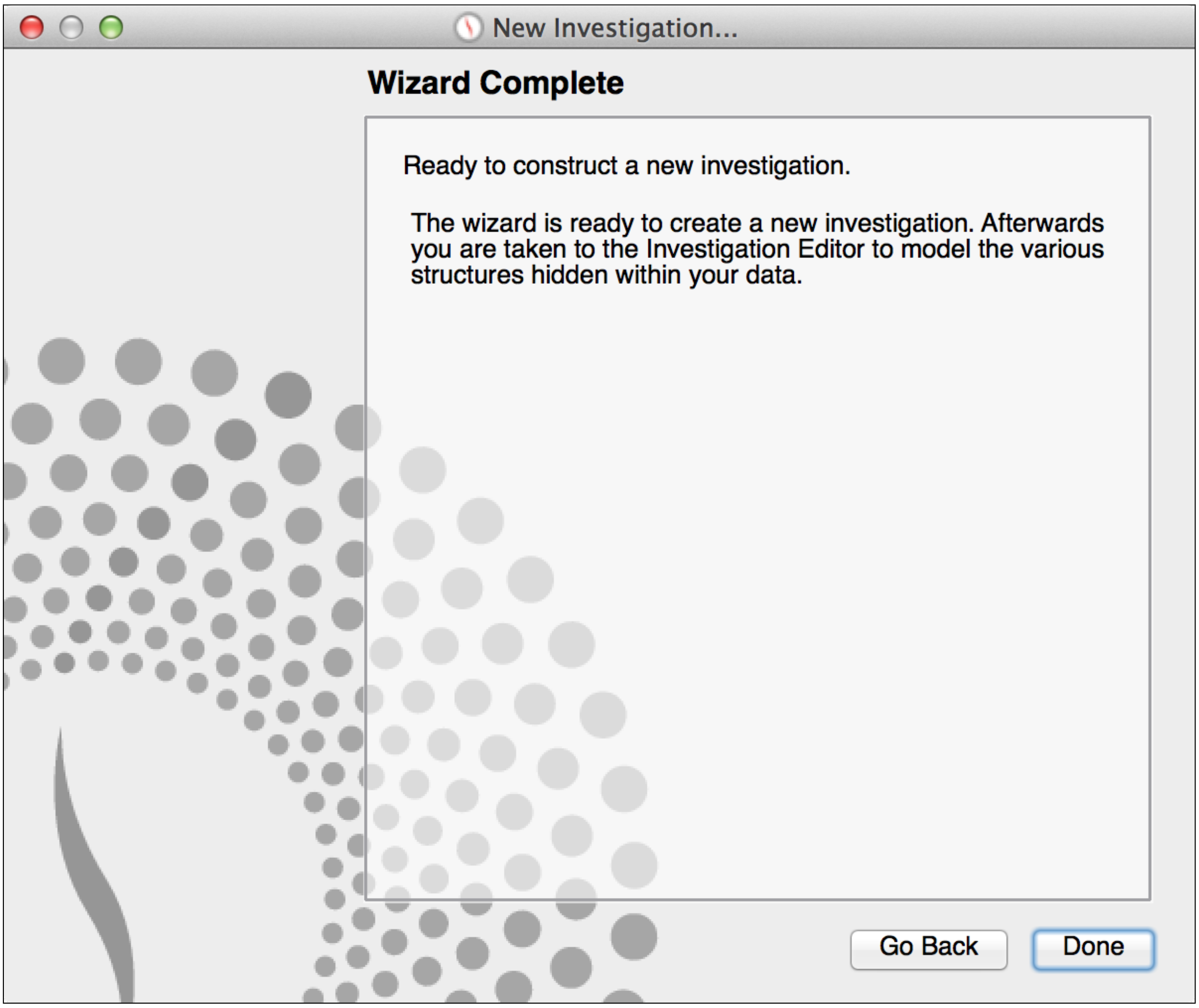


Figure 21: Creating an investigation within a project.

Next, you will be taken to the investigation editor. Here we will define a mapping from the CSV files to nodes and links (Figure 22).

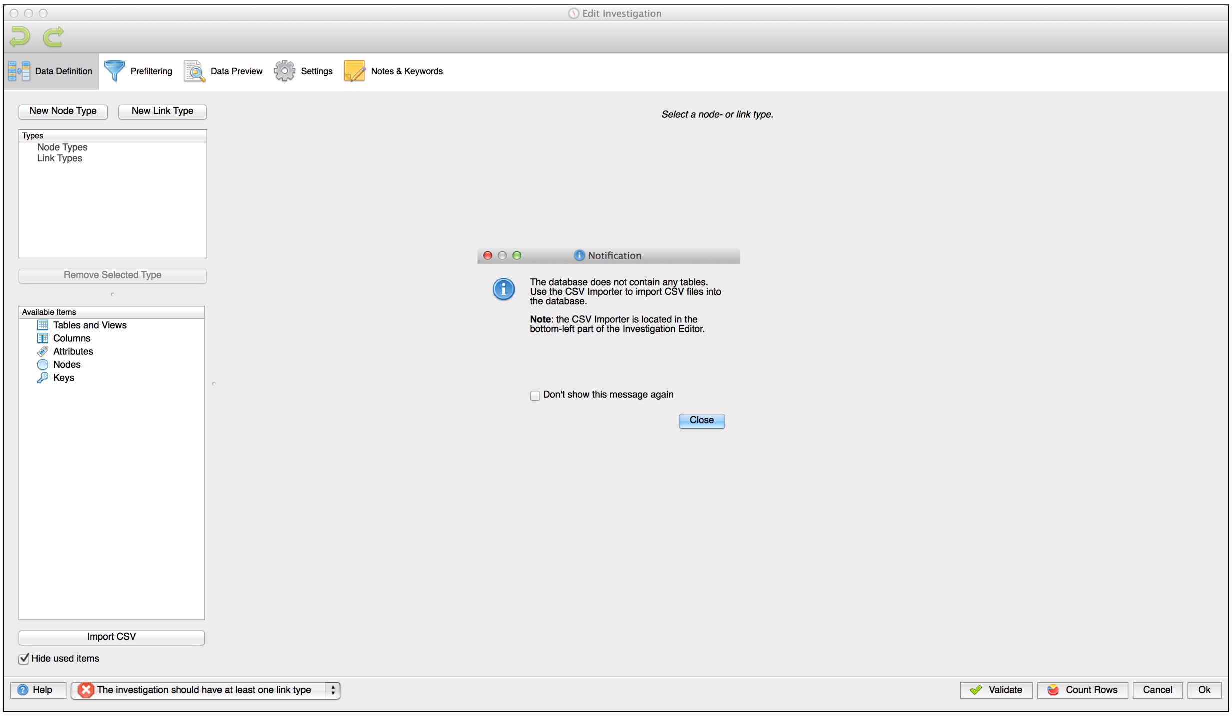


Figure 22: The SynerScope investigation editor.

First, we must import the actual CSV files that we got from the exporter. Click 'Import CSV' in the lower left of the Investigation Editor. This will open the CSV Import Wizard (Figure 23). Click 'Browse' and navigate to either the node or link CSV file. The CSV importer will then ask you about the CSV format, the column definition, and finally about a table name. You should not have to change anything in these steps. Just click continue until you get back to the investigation editor (Figure 22). Repeat the process for the other CSV file.

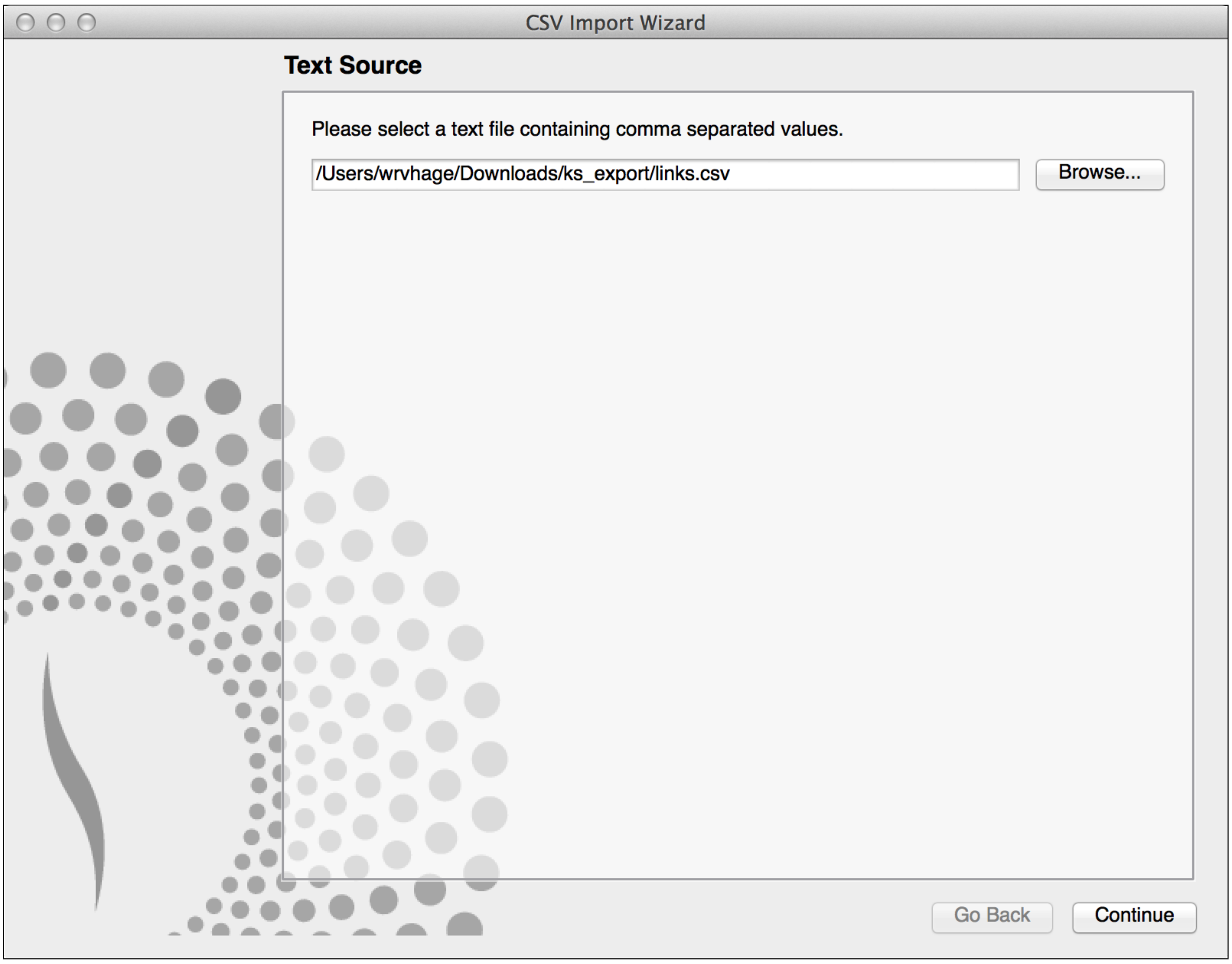


Figure 23: The SynerScope CSV importer.

Now it is time to define a new node type. Click 'New Node Type' in the top left of the investigation editor. Next, enter a name for the node (suggestion: 'Node') and drag the green 'nodes' table from the box in the lower left to the green 'Drop Table Here' area. Then drag all purple columns from the box in the lower left to the purple 'Drop Column Here' area. Finally, drag the blue 'node_URI' attribute from the box in the lower left to the blue 'Drop Attribute Here' area. The result should look like Figure 24.

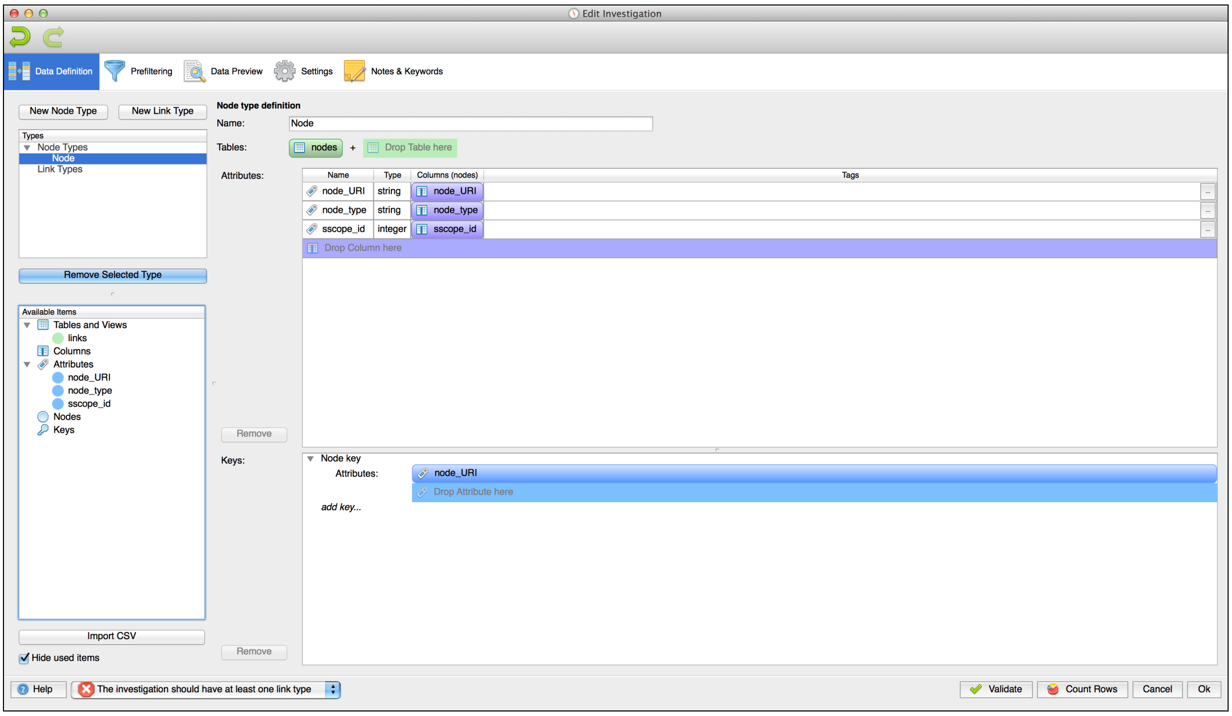


Figure 24: Creating a new node type.

The next step is to create the links between nodes. Click 'New Link Type' in the top left of the investigation editor. Next, enter a name for the link (suggestion: 'Link') and drag the green 'links' table from the box in the lower left to the green 'Drop Table Here' area. Then drag all purple columns from the box in the lower left to the purple 'Drop Column Here' area. Next, drag the red 'Node' from the box in the lower left to both red 'Drop Node Here' fields. Also, drag the yellow 'Node key' from the box in the lower left to both yellow 'Drop Key Here' fields. Finally, drag the blue 'source' attribute from the box in the lower left to the left blue 'Drop Attribute Here' field and drag the blue 'target' attribute from the box in the lower left to the right blue 'Drop Attribute Here' field.

The result should look like Figure 25. Click 'Ok' and you will be shown a list of investigations, among which should be the one you just created. Select it and click 'Open'. You should see a few progress bars and eventually a message like the one in Figure 26 will pop up, letting you know everything is done! Click 'OK'.

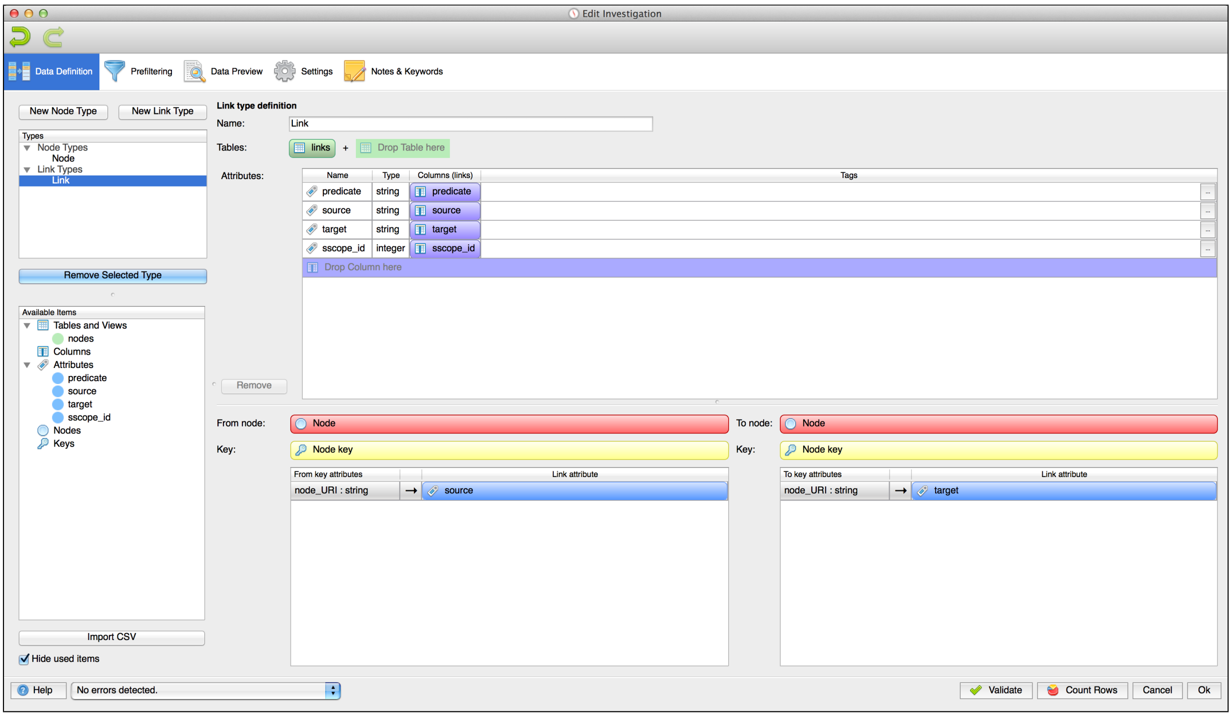


Figure 25: Creating a new link type.

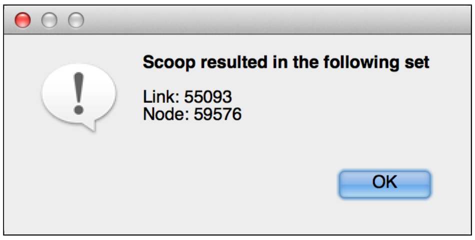


Figure 26: Import complete!

SynerScope's main window should now launch. We will describe how to interact with it in Section 5. Note that the next time you launch SynerScope, you can simply open the project and investigation you just created. Re-running the import process is only necessary when you want to look at a new export from the KnowledgeStore.

5 Interacting with NewsReader Data

After successfully creating a project and investigation (as described in Section 4) or opening an existing one, you will be greeted by SynerScope's default window layout. In this section, we will first show you how to customize this setup for the NewsReader data and then demonstrate few sample interactions.

5.1 The Setup

Figure 27 shows SynerScope's default window layout. To simplify things, lets switch to the split window layout. You can do this by clicking the layout buttons on the lower right or by pressing ALT+2. Leave the HEB view open in the left pane, and open the hierarchy editor in the right pane by clicking the green '+'-button and selecting it from the available views (Figure 28) in the window that pops up. The end result should look something like Figure 29.

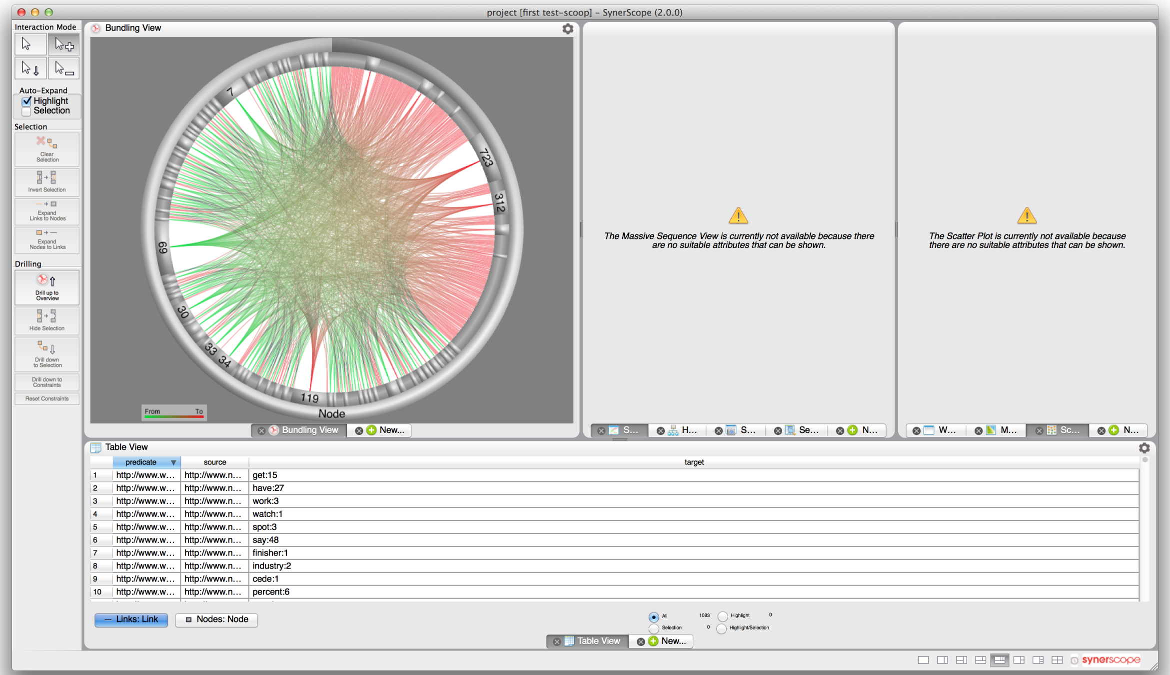


Figure 27: SynerScope default window layout.

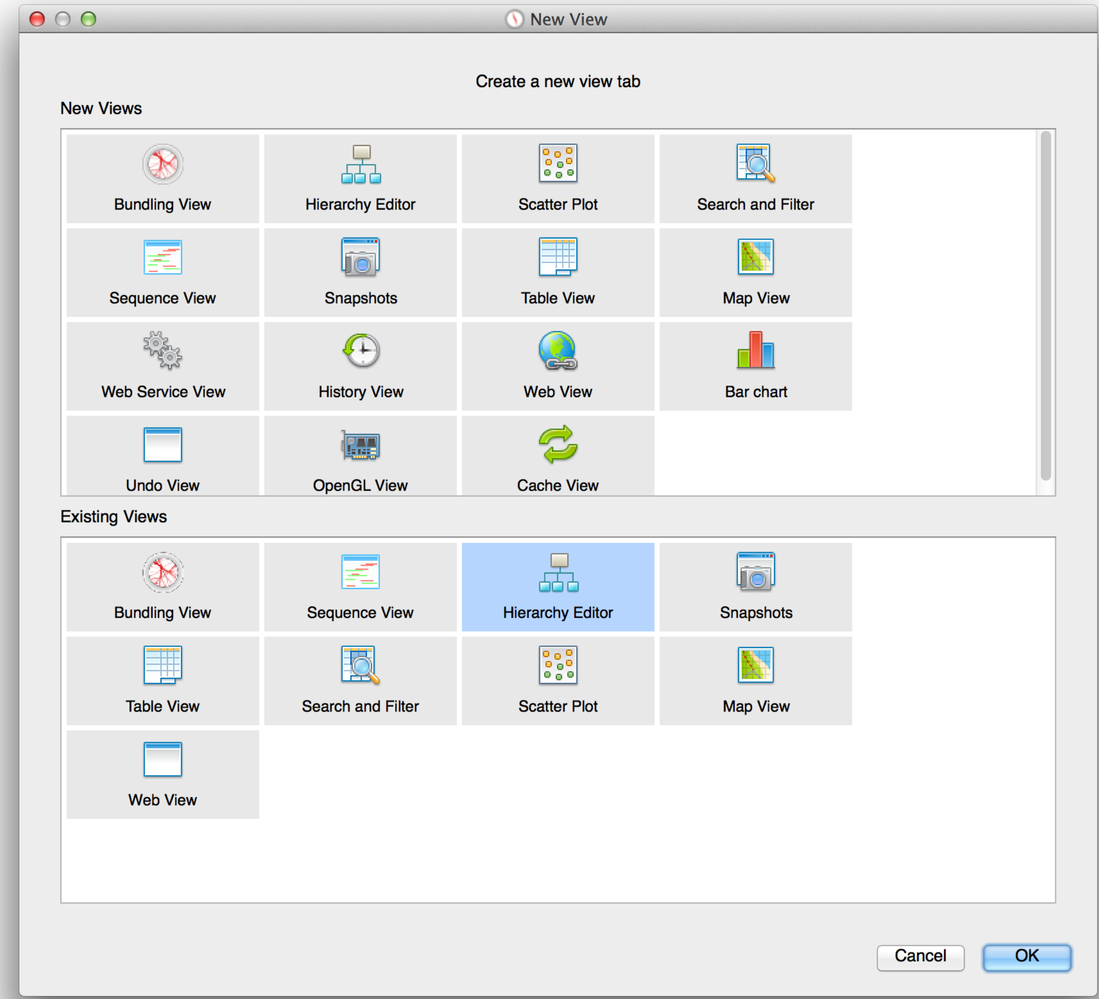


Figure 28: Opening a new view in SynerScope.

Right now, the HEB looks like a massacre. This is because we have not told SynerScope how to hierarchically order the nodes. As you might expect from its name, we can use the Hierarchy Editor to do this.

In the Hierarchy Editor (on the right in Figure 29), click the 'New'-button in the top left and give your hierarchy a name (or use the default). Drag the 'node_type' from the light gray box of inactive attributes to the white box above it. Next, click on the pencil icon in the bottom right to set the node labels. A dialog with a drop-down box should appear. Select the 'node_URI' attribute and click 'OK'. The end result should resemble Figure 29.

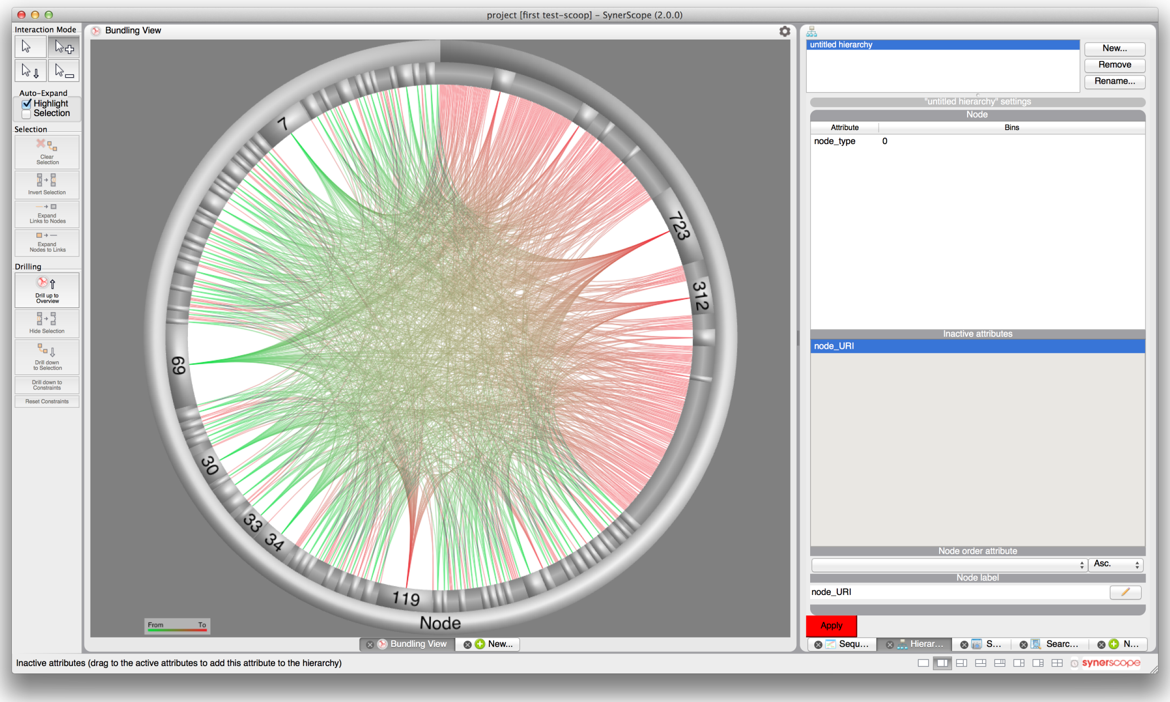


Figure 29: Creating a hierarchy in SynerScope.

After clicking the red 'Apply' button in the hierarchy editor, the HEB should change to resemble the HEB shown in Figure 30. The nodes are now grouped according to their 'node_URI' value and the links between whole groups of nodes are bundled together.

To complete the setup, we will change to window pane layout to include two smaller side panes instead of one large side pane. Switch to this layout by clicking the layout buttons on the lower right or by pressing ALT+6. Then, using the green '+'-buttons, open the table view in the bottom pane and the search & filter view in the top pane. The result should look like Figure 30. We are now ready to continue with some interaction examples in the next section.

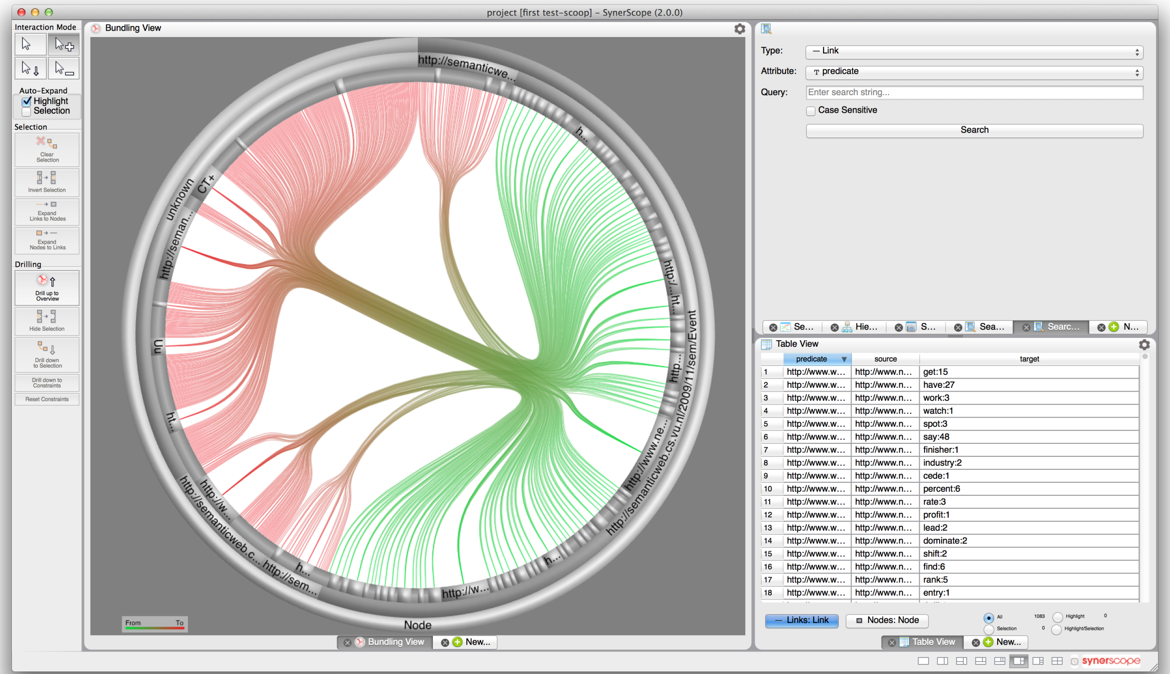


Figure 30: The final SynerScope layout we will use for the interaction examples.

5.2 Interaction Examples

For the interaction examples we have used an export using the actor 'Volkswagen' limited to 500 events. Make sure SynerScope is setup as described in Section 5.1 Using the visualizations described in Section 2.3 and the interaction methods outlined in Section 2.4 we will demonstrate how to find a specific actor and the events it is involved in; and how to find the properties of those events (e.g. place, time).

It is worth mentioning that future revisions of the DSTS and the NewsReader data may render the instructions for these examples invalid. Future versions of this deliverable (D7.3.2 and D7.3.3) will therefore contain updated (and more) examples.

Suppose we want to select all events that involve the actor 'BMW'. First, in the search and filter view, make sure the 'Type'-dropdown is set to 'Node'. Then, in the 'Attribute' dropdown, select 'node_URI'. Unsurprisingly, the query should be 'BMW'.

When you press search, SynerScope will select all nodes that have the characters 'BMW' in their node_URI field. You can see this by switching the table view to nodes and by choosing to show only the current selection.

We now want to see the events BMW is involved in. Actors and events are associated by links. This means we can use the 'Expand Nodes to Links' button to add all links connected to our current selection of BMW nodes to our selection.

Now we have only selected the links, but we want the actual events connected to them to be added to our selection as well. For this we can use the 'Expand Links to Nodes' button. This will add all nodes connected to our currently selected links to the selection. You can now drill down to view this sub-network in isolation. The end result of this example can be seen in Figure 31.

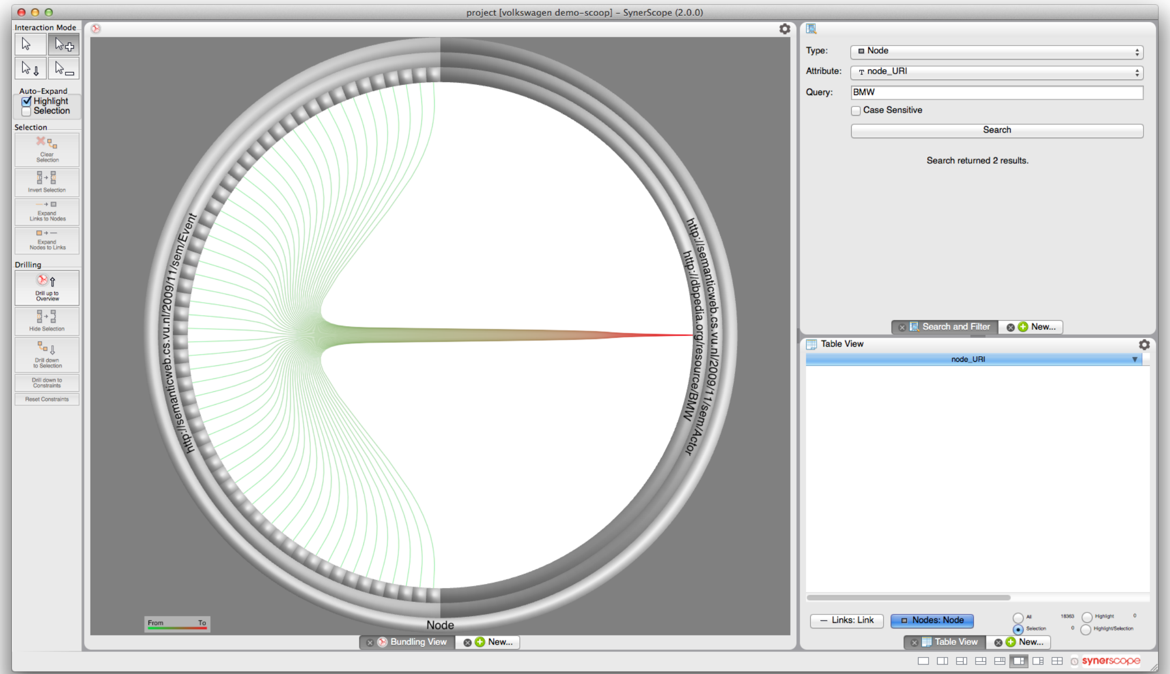


Figure 31: The simple sub-network of events BMW is involved in from example 1.

Just knowing which events BMW is involved in is not terribly interesting. We would like to see some more information on the events we just drilled down to. We can do this by selecting all the events that are currently visibile and then drilling up, as the selection is maintained across drill downs/ups.

Select all event nodes by dragging across all the currently visible links and expanding the links to nodes. Now drill up and notice that the links and nodes from our drill down are still selected. Now that all other links are visible, we can expand our selection once more to gather all other links connected to our events.

After expanding the selection, drill down to zoom in and get a better view of the data. For each event, we can now see which other actors are involved, if it occurred in a particular place, or at a particular time by looking at the actor, place, and time nodes that are linked to the event nodes (Figure 32).

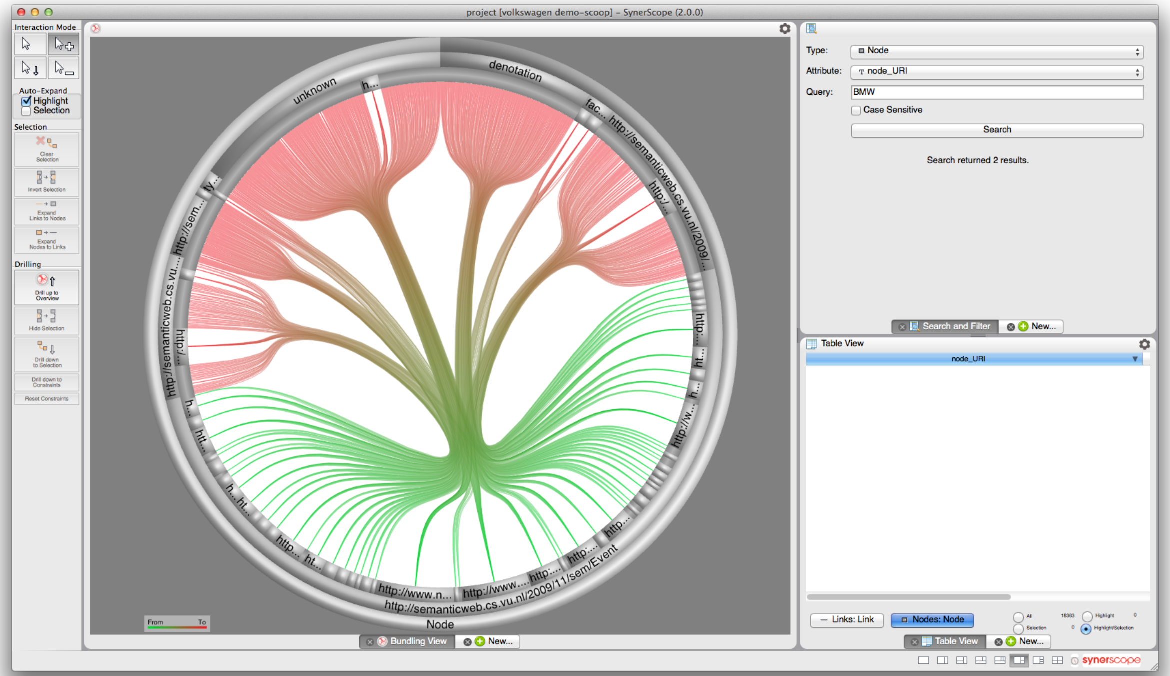


Figure 32: The detailed sub-network of events BMW is involved in from example 2.

6 Conclusion

In this deliverable we documented the design of the Decision Support Tool Suite, a graphical user interface that is meant to support users when making strategic decisions. The DSTS provides insight into the sequences of events that led up to a current situation so that a user can extrapolate to what might happen in the future.

The DSTS will be the main point of end user interaction with the NewsReader event indexes. The technical implemenation of the DSTS largely consists of the SynerScope tool, a visual analytics application that delivers real time interaction with dynamic network-centric data.

SynerScope allows the user to display several visualization methods ('views') simultaneously. Interactions with one view (e.g. highlighting, selecting) are replicated in all other open views. The Hierarchical Edge Bundling View (HEB) is the primary network visualization in SynerScope. Each Node is visualized as a point on a circle, and each Link is visualized as a curved line between its source and target Node. The Nodes are grouped hierarchically, based on an ordering defined by the user.

The NewsReader event data can be imported into SynerScope through a Web-based exporter tool. The exported data must then be imported into SynerScope, where it can be navigated through search, network expansion, and slicing and dicing to look for interesting patterns or irregularities.

6.1 Future Work

In future revisions of the Decision Support Tool Suite (D7.3.2, D7.3.3) we would like to simplify the export-import process as described in Section 4. We also would like to add additional features to the exporter, such as additional types of constraints.

We will also look into taking full advantage of all data layers (i.e. NAF and GAF), going beyond just the event data (SEM+) that is currently loadable. This means we will probably have to adapt our current subject-predicate-object to node-link mapping to a different structure.

With a new mapping, we also expect to be able to take advantage of more of SynerScope's visualizations, such as the Massive Sequence View and the Map View. Finally, we will look into developing custom visualizations using SynerScope's web view, as outlined in Deliverable 7.1.

References