Quick Start Guide for

AllenGUI & AllenSPARQL

2021-01-11, Author: Sebastian Mate

1 Introduction

This document describes how to run temporal queries with AllenGUI and AllenSPARQL based on the minimal test dataset with 13 patients. This dataset has been constructed to test the correctness of the implemented software.

This document guides you through "Method 3: No D2RQ, i2b2 and Oracle database" as described in **`Test Dataset\README.txt`**. To make running the tools easier, we've automated the steps with batch files.

The overall workflow is a bit awkward (e.g., you have to manually restore the Fuseki triple store by uploading the RDF file), which is owed to the fact that there is no i2b2 Oracle database where the data can be pulled from. If you do have an Oracle database or even an i2b2-Wizard-based i2b2 installation, please refer to the above mentioned README file on how to use this. But for a small fun demo following this guide is easier.

This document assumes that you are using a recent version of Windows, therefore the helper files used in this script are Windows batch files (.bat). However, everything also works on any Linux or Mac computer, as long as you adopt and run these commands manually.

2 Prerequisites

To use AllenGUI and AllenSPARQL, you need to install the following software on your computer:

- Java SE Development Kit 8: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Apache Maven (for building the source code): https://maven.apache.org/download.cgi
- also see: https://maven.apache.org/install.html
- Python 3 (for running **`sagecell-client.py`**): https://www.python.org/downloads/windows/
  When installing Python, make sure the set PATH option is ticked!

3 Initialization

This section describes how to compile the software and how to upload data into the Fuseki triple store. To initialize the software stack, please execute the following batch files one after another:

`01-InstallPrerequisites.bat`

This installs two Python packages required by **`sagecell-client.py`**.

`02-InstallD2RQLibrary.bat`

This installs the D2RQ library into your local Maven repository (note: the JAR file provided for your convenience in the lib directory is a copy from the d2rq distribution that can be found here: https://github.com/downloads/d2rq/d2rq/d2rq-0.8.1.zip).

`03-BuildAllenSPARQL.bat`

This compiles AllenSPARQL using Maven.

`04-DownloadFuseki.bat`

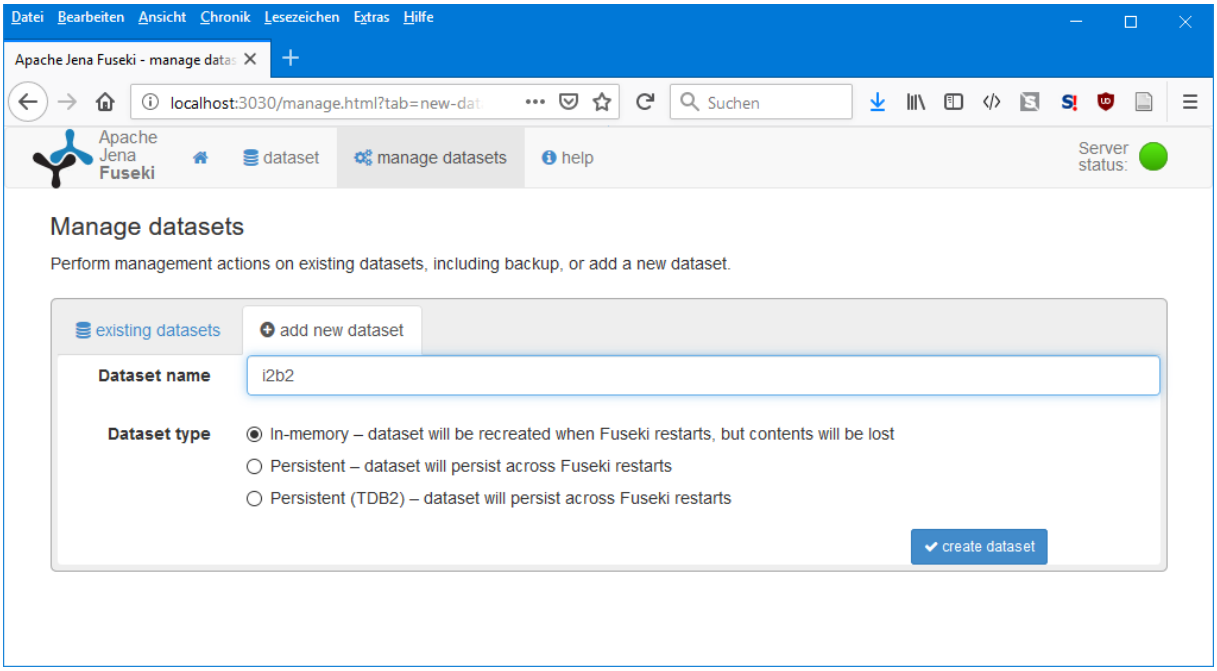
This downloads and extracts Apache Jena Fuseki, the RDF store. If it fails, you may need to modify the download URL in the file.

`05-StartFuseki.bat`

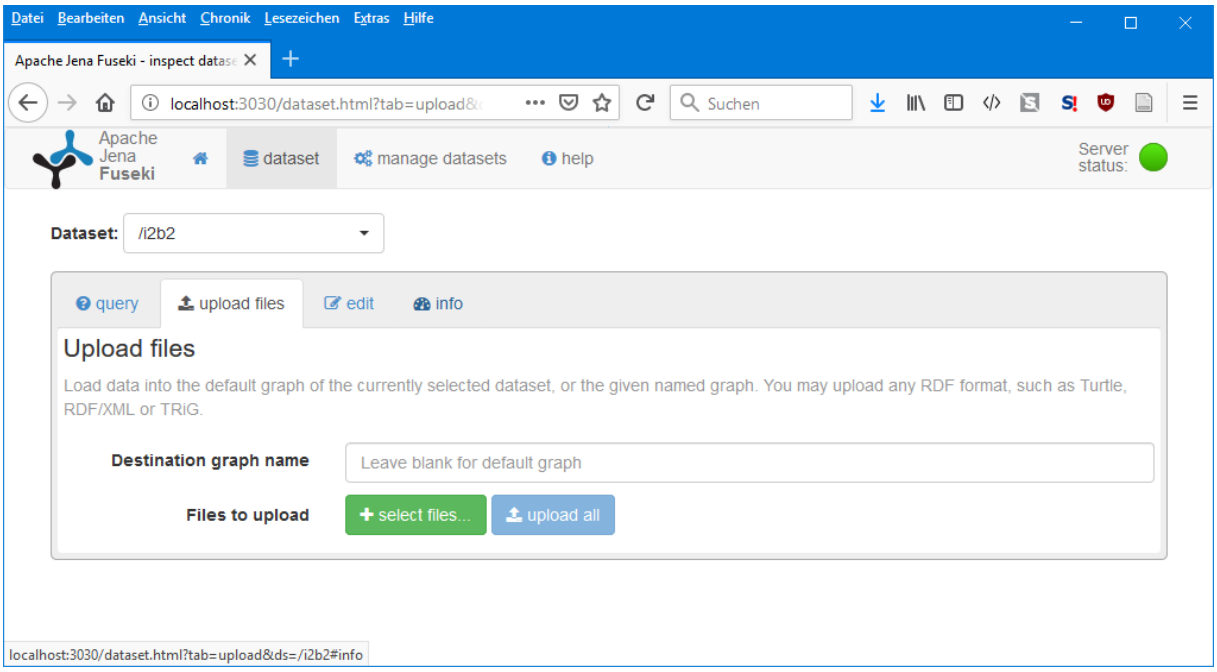
This starts the Fuseki server.

The next step is to upload the test data set into Fuseki. To do this, go to http://localhost:3030/ with your browser. If you can't access this page, Fuseki is not running.

Click on "manage datasets", "add new dataset" and enter "i2b2" as new dataset name. **Important:** Select "In-memory".

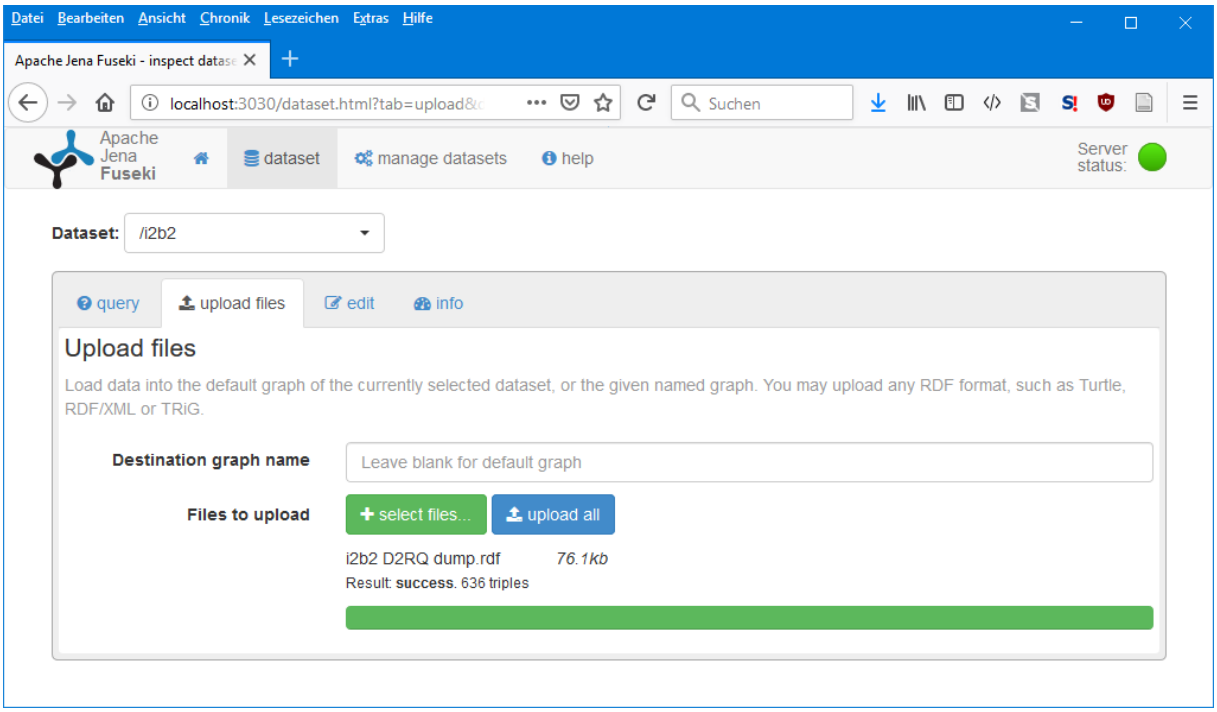


Click on "create dataset". On the next screen, click on "upload data" right to the "/i2b2" entry:



Click on “select files” and select **`Test Dataset\i2b2 D2RQ dump.rdf`**.

Click on “upload now”.



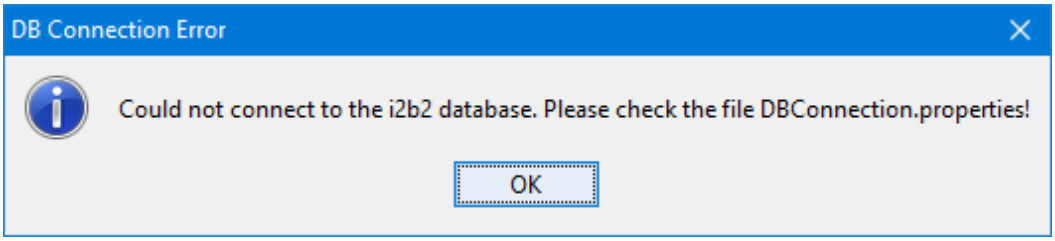
The data in Fuseki is now ready for use.

Finally, run **`06-StartTemporalTools.bat`**

This starts AllenSPARQL and AllenGUI.

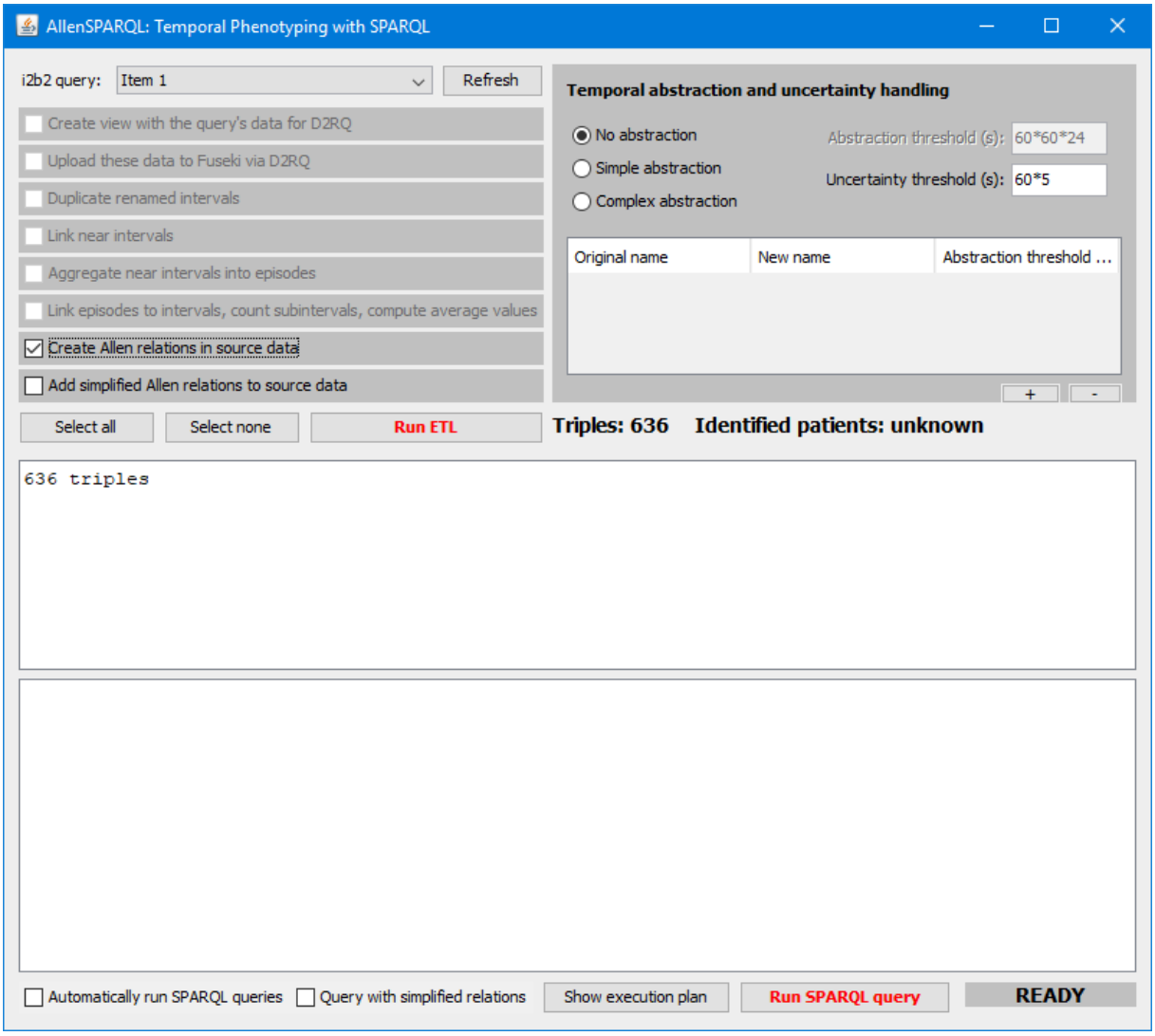
4 Querying without Temporal Abstraction

After starting AllenSPARQL with the default configuration, it will show a database connection error:



This is the expected and correct behavior, as this quick start guide does not use an i2b2 system. Instead we will be directly querying the Fuseki server. Ignore all errors and click on OK. ☺

The main window from AllenSPARQL should look like this:

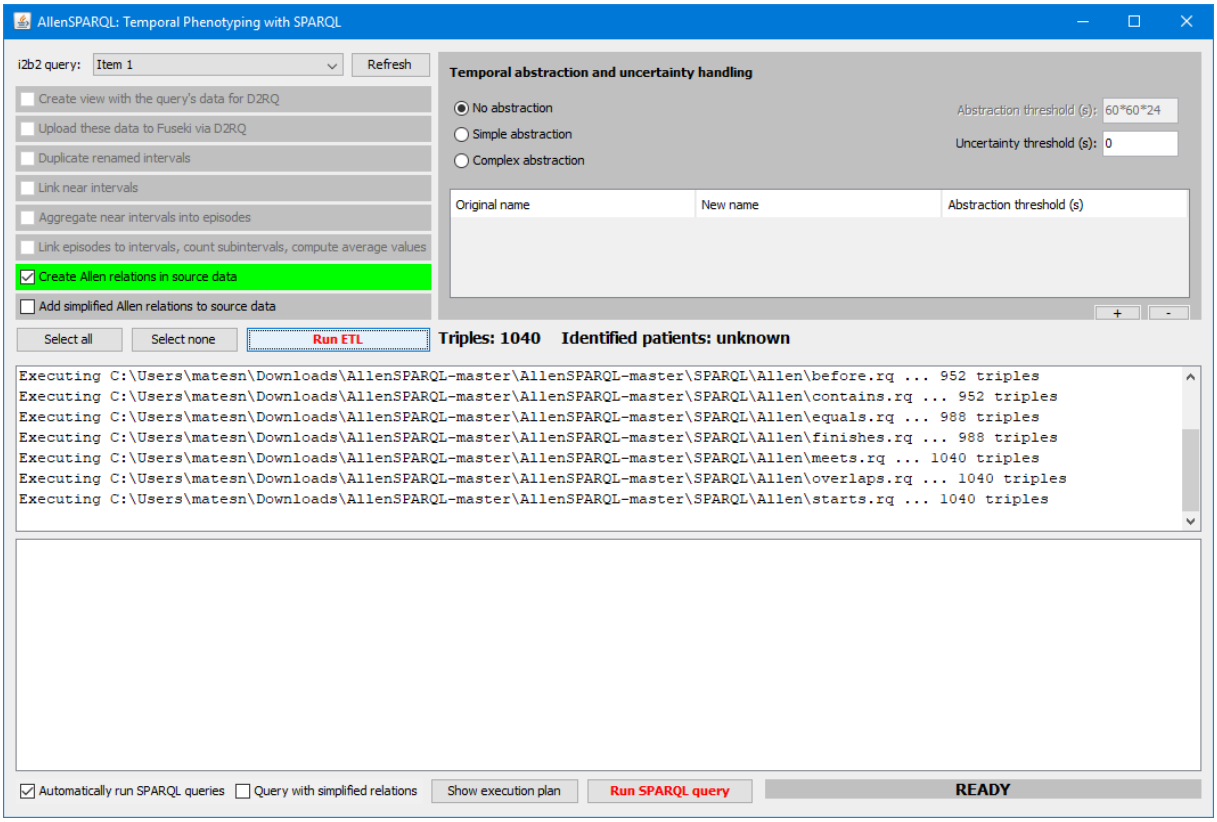


Notice that it displays 636 triples. If nothing is shown, the Fuseki server may not be running.

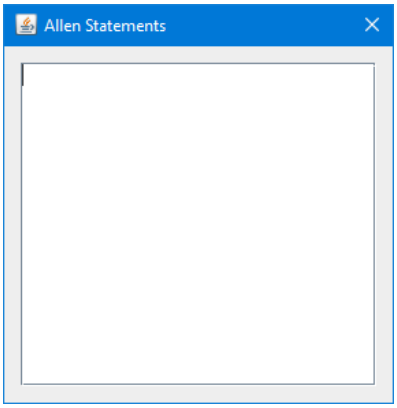
Tick “Automatically run SPARQL queries” in the lower left corner of the window.

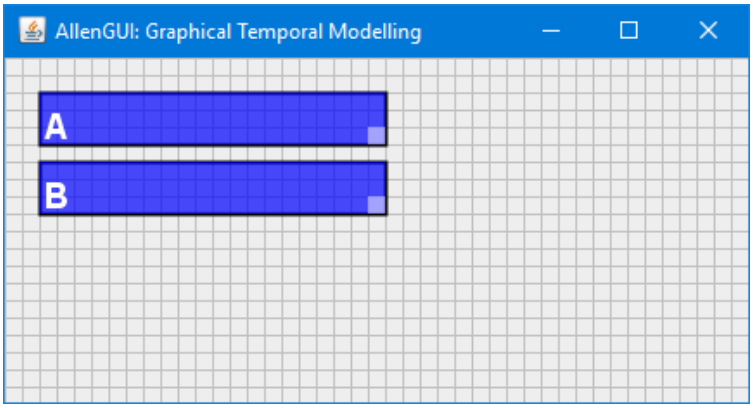
Important: Set “Uncertainty threshold (s)” (in the upper right corner) to “0”.

Tick “Create Allen relations in source data” and then press the button “Run ETL”. This creates the Allen relations among the data in the triple store. The ticked ETL step should turn green and the program should then report 1040 triples:

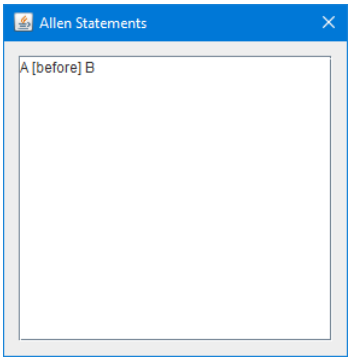


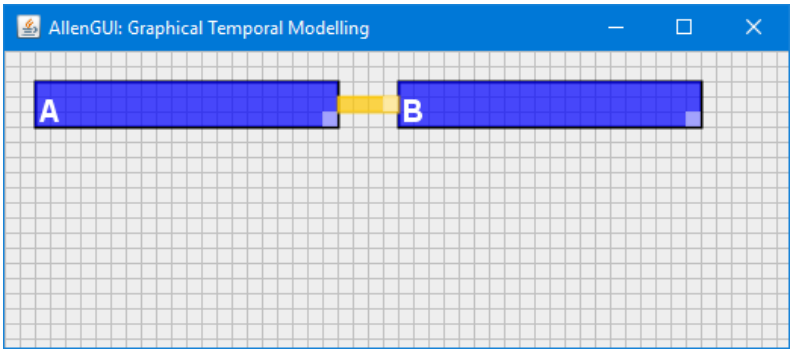
Now to AllenGUI. The tool should display the following two windows:





Use the main window with the grid to model temporal patterns, such as “A before B”:

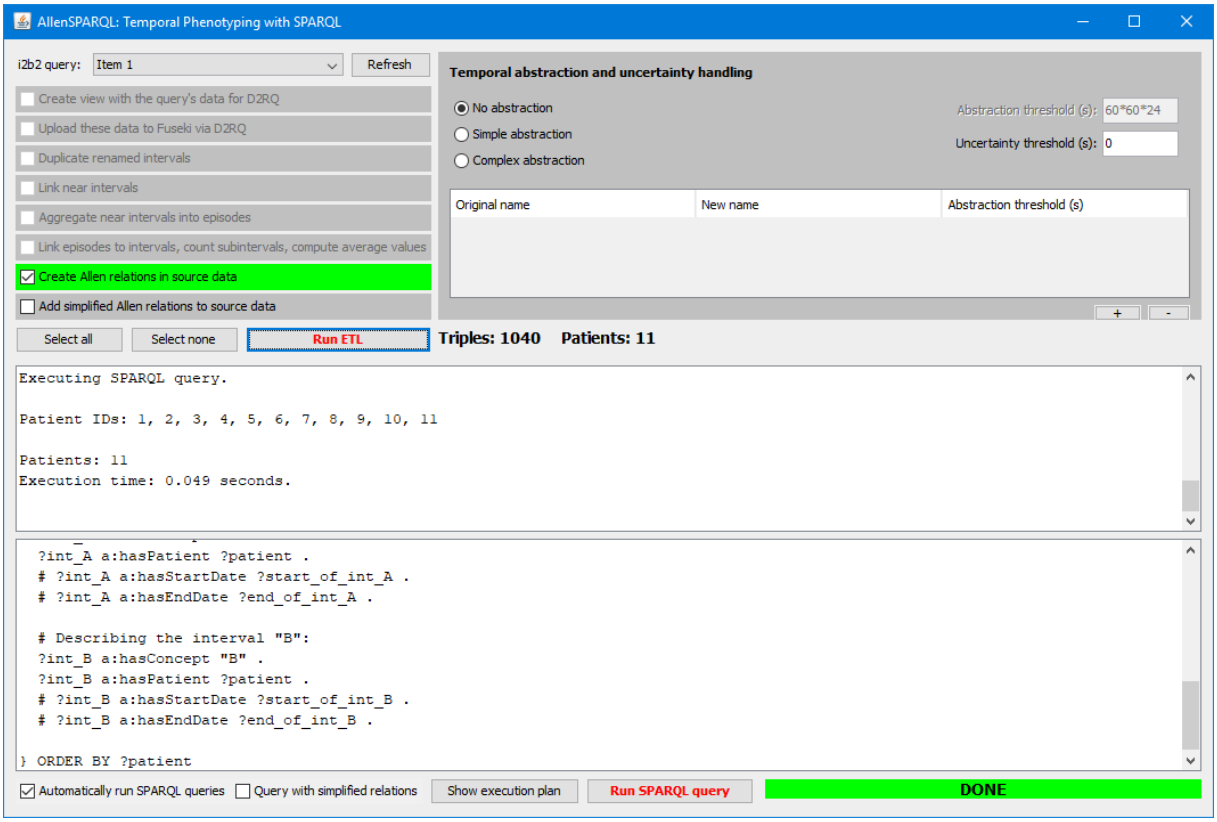




This can be done easily with the mouse:

- Use the left mouse button to move items on the canvas.
- To create a new block (or a yellow connector line), click on an empty space on the grid.
- To delete an item, click on it with the right mouse button.
- To rename an item, press the middle mouse button.

AllenSPARQL should have already executed this query. It should report 11 patients (IDs 1 - 11):



The following table shows the content of the database. There are thirteen fictional patients, one for each Allen relation. On three consecutive days (Day 1 = 2017-01-01, Day2 = 2017-01-02, Day 3 = 2017-01-03), there may be two different types of intervals, “A” and “B”. Whenever there are intervals of any type, these take place at the start of the day (00:00:00 to 00:01:00) and at the end of the day (23:59:00 to 00:00:00). Each interval has a length of 1 minute.

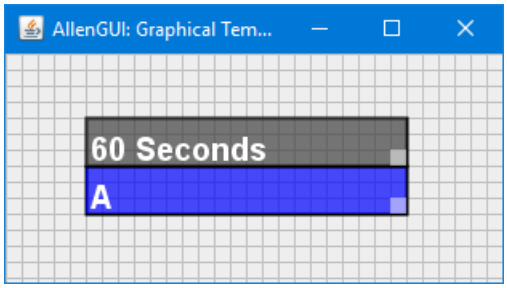
Before Temporal Abstraction				
Patient	Relation	Day 1	Day 2	Day 3
1	A before B	A A		B B
2	A meets B	A A	B B	
3	A overlaps B	A A	A A B B	B B
4	A finished by B	A A	A A B B	
5	A contains B	A A	A A B B	A A
6	A starts B		A A B B	B B
7	A equals B		A A B B	
8	A started by B		A A B B	A A
9	A during B	B B	A A B B	B B
10	A finishes B	B B	A A B B	
11	A overlapped by B	B B	A A B B	A A
12	A met by B	B B	A A	
13	A after B	B B		A A

00:00:00 to 00:01:00

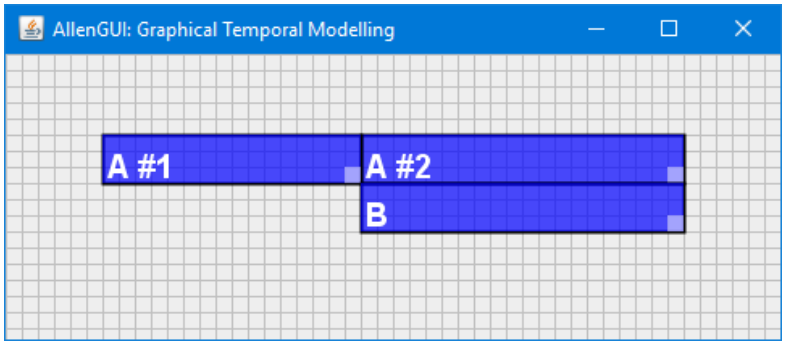
23:59:00 to 00:00:00

The “A before B” query from above returned all patients except 12 and 13, because the latter do not have any A before B (notice that for patients 12 and 13, all “A”s are after all “B”s).

Because each patient has at least one “A” interval that has a duration of one minute, the following query will also return all 13 patients:



This query will return patients 3, 4 and 5:



The reason for this is obvious when reviewing the data:

3	A overlaps B	A A A A B B B B
4	A finished by B	A A A A B B
5	A contains B	A A A A A B B A

5 Querying with Temporal Abstraction

For this you need to reset the Fuseki triple store. This can be done either by manually resetting or deleting the "i2b2" dataset in the Fuseki web interface or by deleting and re-installing Fuseki.

If an in-memory dataset was used previously (as described above in this text), simply restarting Fuseki should wipe the triple data of the "i2b2" dataset when restarting Fuseki. If this is the case, simply restart Fuseki and upload **`i2b2 D2RQ dump.rdf`** again. It should again report 636 triples.

If a persistent dataset was used previously, simply deleting and recreating the "i2b2" dataset may not be sufficient to actually delete the data. To be on the safe side, delete the whole Fuseki installation and recreate it:

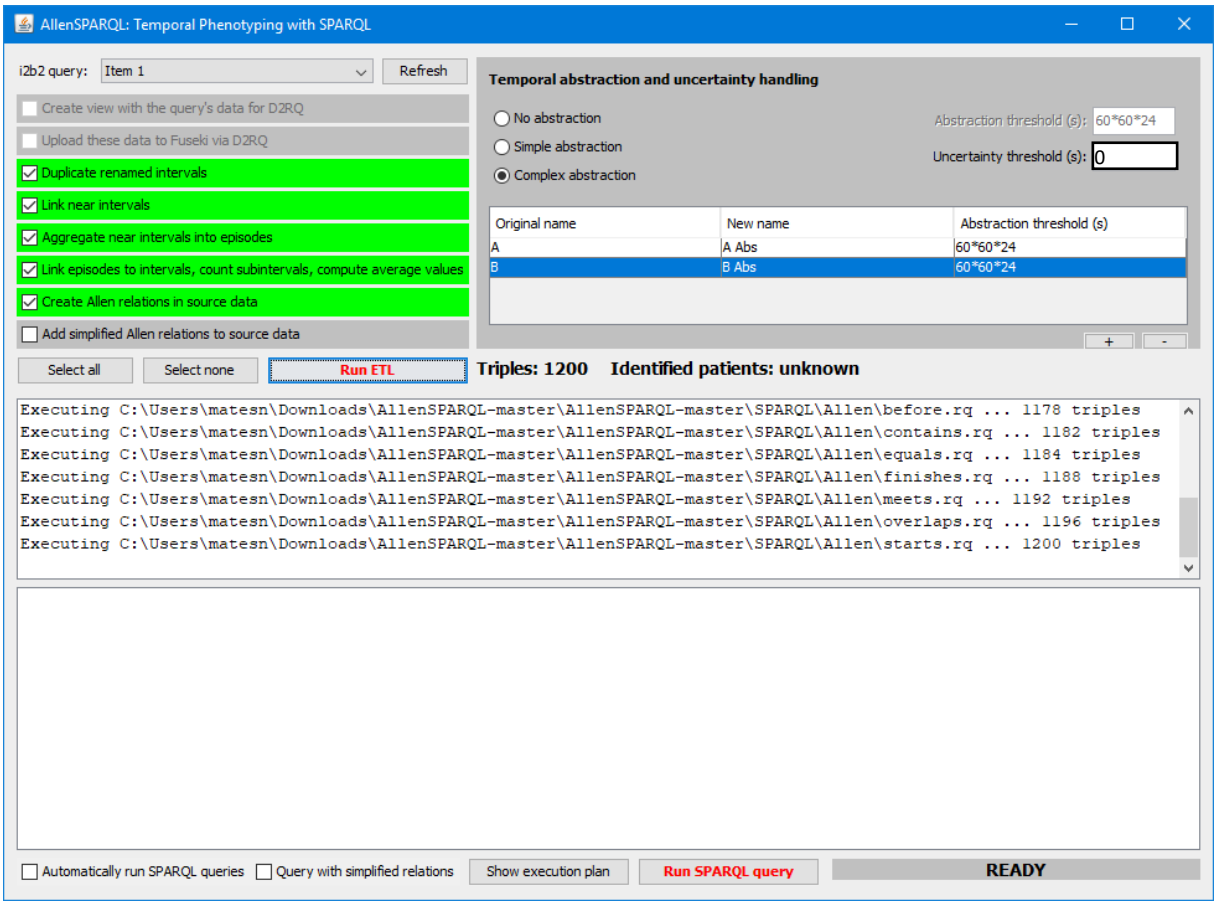
- Close all programs, including Fuseki. This will wipe the Fuseki "i2b2" dataset, as it was configured as "in-memory".
- Delete the directory **`apache-jena-fuseki-3.11.0`**
- Run **`04-DownloadFuseki.bat`**
- Run **`05-StartFuseki.bat`**
- Open the web interface of Fuseki and upload **`Test Dataset\i2b2 D2RQ dump.rdf`** again.
- Run **`06-StartTemporalTools.bat`**

Then, in AllenSPARQL:

- Tick "Automatically run SPARQL queries" in the lower left corner of the window.

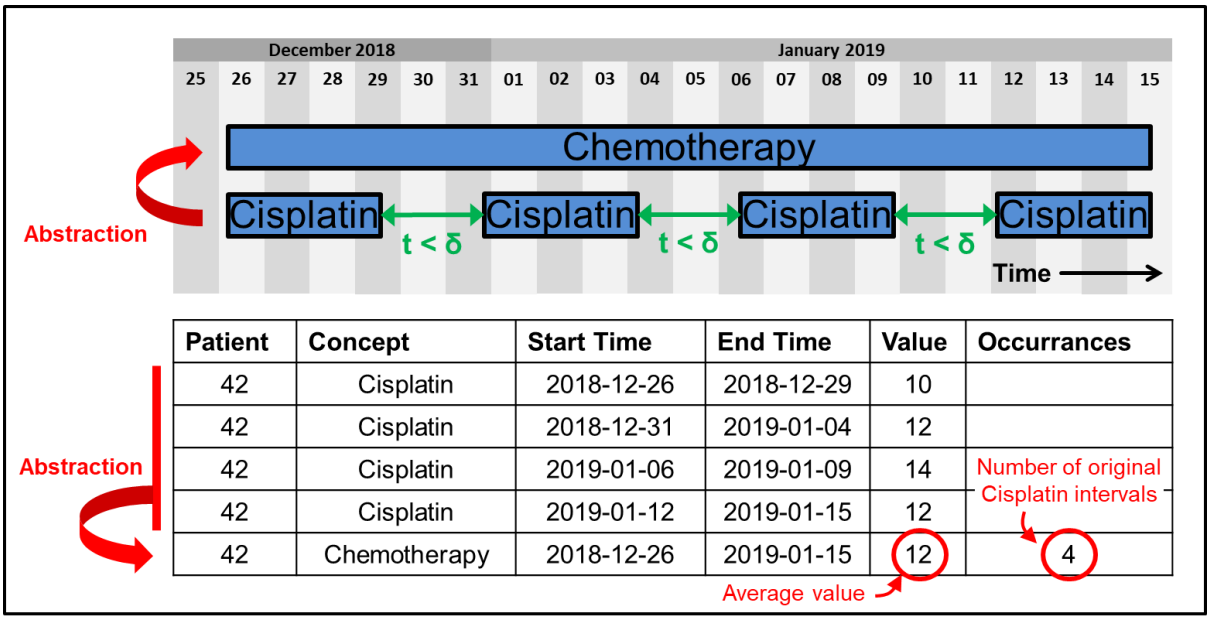
- Select “Complex Abstraction” on the right. With the small “+” Button, create two new entries that abstract all nearby “A” intervals into “A Abs” and all “B” intervals into “B Abs”.
- **Important:** Make sure that “Duplicate renamed intervals” is ticked
- **Important:** Set “Uncertainty threshold (s)” (in the upper right corner) to “0”.
- Click on “Run ETL”.

After the execution, the program should report 1200 triples:



This abstraction does the following: Whenever there is a series of intervals of one type, such as “A”, and whenever the temporal distance between any two of these intervals is less than δ, the “Abstraction threshold”, a new interval of the type “A Abs” is created that encompasses the original series.

As shown in the following illustration, this can be used to abstract a series of Cisplatin administrations into a chemotherapy cycle:



Patient	Concept	Start Time	End Time	Value	Occurrances
42	Cisplatin	2018-12-26	2018-12-29	10	
42	Cisplatin	2018-12-31	2019-01-04	12	
42	Cisplatin	2019-01-06	2019-01-09	14	Number of original Cisplatin intervals
42	Cisplatin	2019-01-12	2019-01-15	12	
42	Chemotherapy	2018-12-26	2019-01-15	12	4

Abstraction

Average value

In the test data set this is less spectacular. In the test dataset, “A” and “B” are technically equivalent to “Cisplatin” and “A Abs” and “B Abs” to “Chemotherapy”.

Using a threshold of 60*60*24 seconds = 24 hours for δ, as in the example above, this generates the “A Abs” and “B Abs” intervals as shown on the right:

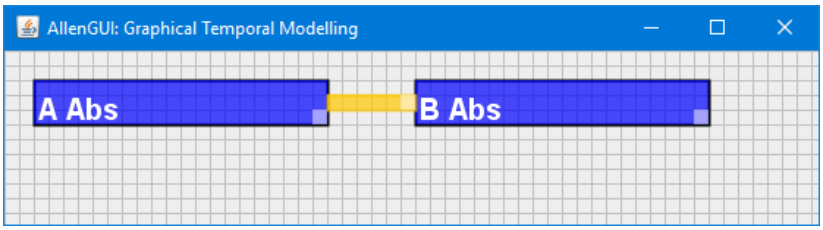
Before Temporal Abstraction

Patient	Relation	Day 1	Day 2	Day 3
1	A before B	A	A	B B
2	A meets B	A	A B	B
3	A overlaps B	A	A A B	A B B B
4	A finished by B	A	A A B	A B
5	A contains B	A	A A B	A B A A
6	A starts B	00:00:00 to 00:01:00	A B	A B B B
7	A equals B		A B	A B
8	A started by B	23:59:00 to 00:00:00	A B	A B A A
9	A during B	B	B A B	A B B B
10	A finishes B	B	B A B	A B
11	A overlapped by B	B	B A B	A B A
12	A met by B	B	B A	A
13	A after B	B	B	A A

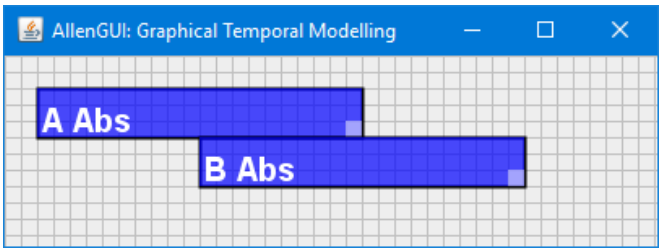
After Temporal Abstraction

Patient	Relation	Day 1	Day 2	Day 3
1	A before B	A Abs		B Abs
2	A meets B	A Abs	B Abs	
3	A overlaps B	A Abs	A Abs B Abs	
4	A finished by B	A Abs	A Abs B Abs	
5	A contains B	A Abs	A Abs B Abs	
6	A starts B		A Abs	A Abs B Abs
7	A equals B		A Abs B Abs	
8	A started by B		A Abs B Abs	
9	A during B		A Abs	A Abs B Abs
10	A finishes B		A Abs	A Abs B Abs
11	A overlapped by B		A Abs	A Abs B Abs
12	A met by B	B Abs	A Abs	
13	A after B	B Abs		A Abs

Now querying with AllenGUI (while using the abstracted intervals), the following query only returns patient 1:



This query returns patient 3:



Feel free to test all other Allen relations. ☺

Similarly, the following query will return patient 5, because this is the only patient that has a three-day interval of the type “A Abs”: