

SWEETInfo: a Web-Based System for Visualizing and Querying Temporal Data

Martin J. O'Connor, Amanda Richards, Susana Martins, Mike Bingen, Samson W. Tu, Amar Das

Stanford Center for Biomedical Informatics Research
Stanford, CA 94305, U.S.A. martin.oconnor@stanford.edu

Abstract. We describe SWEETInfo (Semantic Web-Enabled Exploration of Temporal Information), a highly-interactive web-based system for querying and visualizing time-oriented data. SWEETInfo was developed as an open-source Web-based infrastructure and combines Semantic Web technologies, such as OWL and SWRL, with standard Web development software, such as the Google Web Toolkit. Investigators can use SWEETInfo to collaboratively manipulate, explore, and visualize temporal data. It allows users to interactively generate complex temporal analyses and to share these analyses with other SWEETInfo users. It also supports importation of data from spreadsheets and relational databases and the final export of analysis results. SWEETInfo was publicly released in September 2011 and is available at www.sweetinfo.org.

1 Introduction

The temporal dimension of data is central in many domains. Answering non-trivial temporal questions in these domains typically requires generating complex temporal criteria. For example, a question in a medical domain might combine temporal durations at multiple granularities (“Did a patient have elevated blood sugar levels for more than two days in the past week?”), queries with aggregates (“What was the average post-breakfast blood sugar level of a patient over the last week?”), and the use of periodic patterns (“Did a patient have more than one three-week interval of drug treatment followed by a suppressed viral load lasting at least a week?”). Encoding these temporal queries is difficult using currently available tools. Although commonly-used tools such as SAS and R support complex analysis strategies, they offer very poor support for expressing temporal criteria. They generally support only simple instant-based timestamps and a small set of basic temporal functions. A key shortcoming is that they lack a temporal model. Hence, there is no principled means of associating a piece of information with its temporal dimension. The end result is that temporal criteria have to be expressed at a very low level and customized to the details of the source data layout and structure. Expressing non-trivial criteria, and, in particular, developing complex multi-layered analyses, requires the time-consuming development of customized analysis routines, a process that often requires specialist knowledge. Sharing these analyses with others is also problematic.

There is a clear need for tools to support development and sharing of complex temporal analyses by non specialist users. To address this need, we developed a system that allows researchers to interactively develop temporal analyses that can be published and shared with other researchers. The application, Semantic Web-Enabled Exploration of Temporal Information (SWEETInfo) [1], is a web-based tool that lets investigators collaboratively manipulate, explore and visualize temporal data.

2 SWEETInfo System Architecture

SWEETInfo combines semantic web technologies such as the Web Ontology Language (OWL) [2] and the Semantic Web Rule Language (SWRL) [3] with a variety of standard web technologies. It was developed as a web-based application using standard open source technologies. Its architecture consists of a fairly standard n-tier web layering (Figure 1). The three primary layers are: (1) a presentation layer; (2) a business logic layer; and (3) a data access layer. The presentation layer uses a combination of CSS and the Google Web Toolkit (GWT) [4]. GWT provides a convenient Java wrapper around Javascript, allowing an application to be developed completely in Java. The business logic layer was developed using semantic web technologies. In particular, it uses an OWL-based temporal information model to provide a standardized means of representing all information in a system. SWRL-based temporal reasoning methods are used in SWEETInfo to perform operations on data in this information model. The data access layer uses Protégé-OWL APIs (protege.stanford.edu) to manipulate OWL ontologies. Its SWRL APIs (protege.cim3.net/cgi-bin/wiki.pl?SWRLTab) provide the ability to execute SWRL rules. These APIs use the Jess rule engine (www.jessrules.com) to execute the SWRL rules.

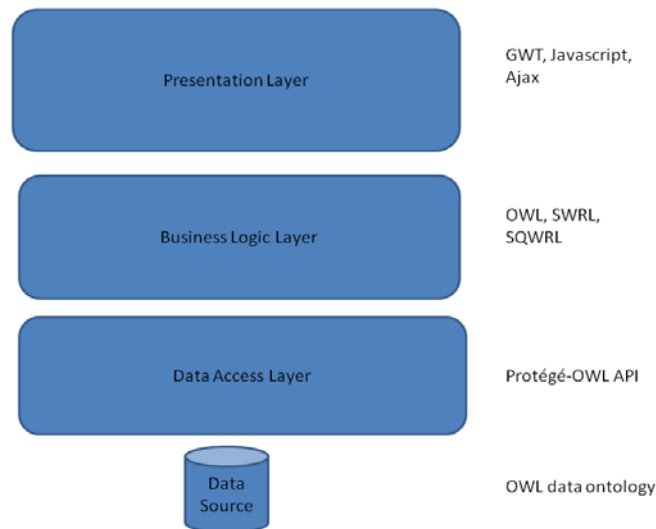


Figure 1. Layers in SWEETInfo System Architecture.

3 SWEETInfo Information Architecture

A principled temporal model is key when performing temporal analysis because it enforces a consistent representation of temporal information in a system. One important result of research in the area of temporal data representation is convergence on the valid-time temporal model [5]. Although numerous models have been used to represent temporal information in relational databases and other information systems, we selected this one because it couples simplicity with considerable expressivity. We adopted an OWL-based representation of this model in SWEETInfo [6]. This model is used to encode the temporal dimension of all user data, thus allowing standardized approaches to temporal reasoning and querying.

Using this temporal model as a base, we developed an information model in OWL to provide a consistent representation of all user data in our system. Our model provides a consistent representation of the data's temporal dimension. All data are represented using *variables*, which are atomic pieces of data. Each variable has a value and a temporal dimension. Variables can be numeric, string-based, or include ontology term references. Example variables include viral load values and blood pressure values. All variables are associated with an object of interest (e.g., a patient). All operations and displays in SWEETInfo are defined in terms of objects of interest and the variables associated with them.

Once all temporal information is represented consistently in an information model, it can be manipulated using reusable methods. While OWL has very limited temporal operators for manipulating time values, its associated rule language SWRL provides a small set. However, these operators are very basic, and provide simple instant-based comparisons only. SWRL provides built-ins, which are a mechanism for creating user-defined libraries of custom methods and using them in rules. We have used them to define a library of methods that implement Allen's interval-based temporal operators [7]. The library also has a native understanding of our temporal information model and supports an array of temporal operations on entities defined using it. It can thus be used to directly reason about data defined using our information model.

A SWRL-based query language called SQWRL (Semantic Query-Enhanced Web Rule Language) [8] provides querying support in the system. SQWRL defines a set of SQL-like query operators that can be used to construct retrieval specifications for information in an OWL ontology. SQWRL uses our temporal library to provide complex temporal selection of results. For example, it can make queries such as List the first three doses of the drug DDI or Return the most recent dose of the drug DDI. SQWRL's ability to work directly with our temporal model allows expressive, yet relatively concise temporal queries for expressing criteria. All criteria in SWEETInfo can be expressed directly in SQWRL. Some operations are implemented by combining rules and queries to generate intermediate results incrementally at successively higher levels of abstraction.

A set of predefined temporal operations provide the system's basic set of data transformations. Each operation was designed to be simple enough to be defined using straightforward forms-based dialogs. They were also designed to be easily combined with other operations in a sequence. In this way, users can build complex functionality from relatively simple operations. We identified an initial set of four generalized operations necessary to express temporal criteria. They are:

Filter Operations. These select a defined subset of data out of a larger dataset. We currently provide four filtering criteria for operations. They are *value criteria*, which can express basic equality or inequality criteria on numeric or string data, *aggregate criteria*, which can filter on the count, average or maximum or minimum of data, and two forms of temporal constraints. They are *duration criteria*, which can select data based on the length of time between two time points, and *timing criteria*, which support a range of temporal criteria using standard Allen operators. Instances of these four criteria can be composed in a single filter operation (conjunctively or disjunctively) to build complex selections of data. The four criteria can also be used when defining the other three operations.

Grouping Operations. Dividing data into groups is common when performing analysis. For example, patients can be grouped into related categories (e.g., short, medium, tall), which are then analyzed in different ways. A grouping operation allows users to define a set of named groups meeting different sets of criteria. Each group is specified using a criteria set, and data for each group is generated from the subset of input to a grouping operation that meet its associated criteria.

New Variable Operations. SWEETInfo uses variables to define data elements (e.g., blood pressure, viral load; see Section 3). A new variable operation defines a new variable by creating a restriction on an existing variable. For example, a High RNA variable is defined by selecting RNA values that are greater than some number, using a value criterion. Alternatively, a temporal duration criterion could be used to find patients who had been treated with a particular drug for longer than one month.

Temporal Context Operations. Defining temporal patterns is a central requirement when working with temporal data. Temporal context operations are basic building blocks in this regard because they allow users to specify periods that meet a certain pattern. For example, if a user is interested in post-surgical patient outcomes, he could create a new temporal context to represent the period from the beginning of a hospitalization to 30 days after release. This operation allows complex temporal criteria to be built iteratively from a smaller set of simple criteria.

4 SWEETInfo System Features

SWEETInfo uses this information architecture to provide five core functionalities: (1) data importation; (2) temporal data analysis; (3) data visualization; (4) publication and sharing; and (5) analysis results exporting.

4.1 Data Importation

SWEETInfo allows users to import data into its information model from relational databases or spreadsheets. It provides an interactive wizard that allows users to first select a data source and then to select subsets of that data that are to be represented as SWEETInfo variables (Figure 2).

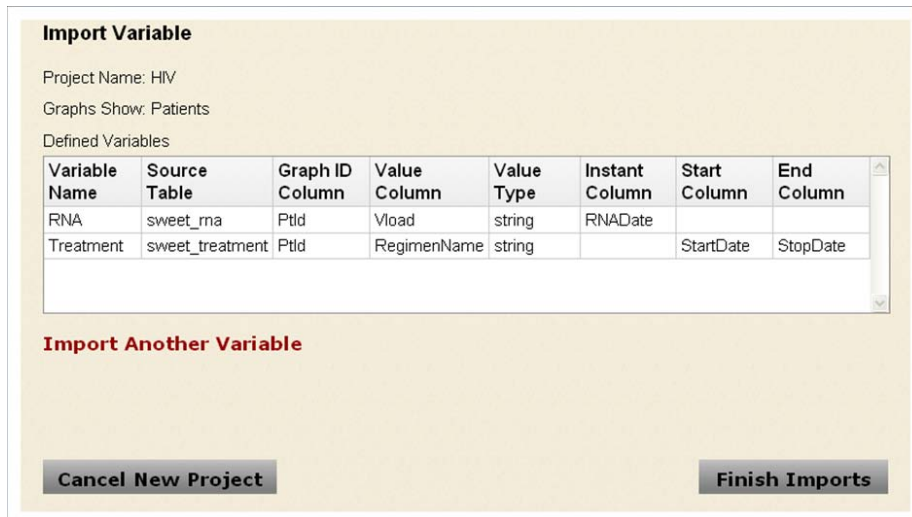


Figure 2. Screen shot of a preview stage of SWEETInfo import wizard showing user-specified mapping of relational tables to SWEETInfo variables.

4.2 Temporal Data Analysis

As mentioned, operations in SWEETInfo ultimately specify the action taken when certain conditions are met. The conditions are specified using a criteria set. Each operation has an associated creation dialog, which has four subtabs, one for each criteria type. Users can specify criteria interactively, and, on creation, they are displayed in summary form.

To support a typical analysis workflow, we used a pipeline-based representation of analyses (Figure 3). A pipeline is composed of chained operation-view pairs. This approach allows users to see intermediate results of the step-by-step operations that generate the overall analysis. Multiple parallel paths can be defined with filter and grouping operations. A visualization node is automatically produced by each operation and can be used to explore data. It can also be used to summarize the number of patients meeting criteria at each stage of the pipeline. Each view node can also be opened to view detailed displays of the data.

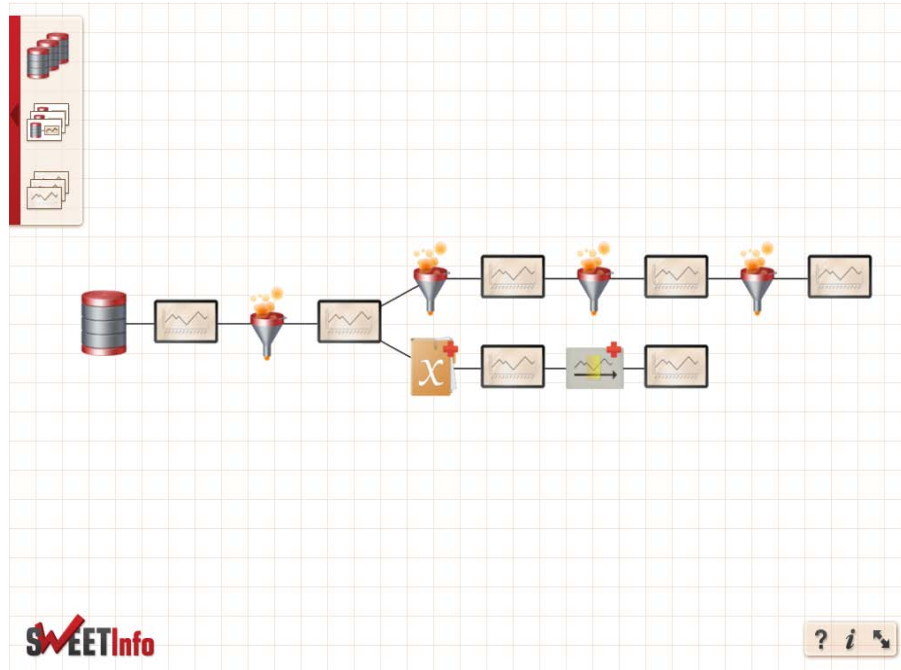


Figure 3. Screen shot of SWEETInfo showing a pipeline definition.

4.3 Data Visualization

Within a pipeline, users can immediately execute an operation that they have defined and examine the results in a set of graphical displays, which show population-level data. They can also drill down to examine individual data elements (Figure 4). The population-level view provides a simultaneous display of data for multiple data elements. The view node can also provide summary statistics for an operation. SWEETInfo provides customized displays for different types of temporal data, and allows users to customize display options. After viewing the data, a user can modify operations in a pipeline immediately or define additional operations. Users can also define branching points using filter and group operations to define parallel analysis paths. The immediate visual feedback and the ability to quickly modify or extend pipeline operations facilitate rapid development of analyses.

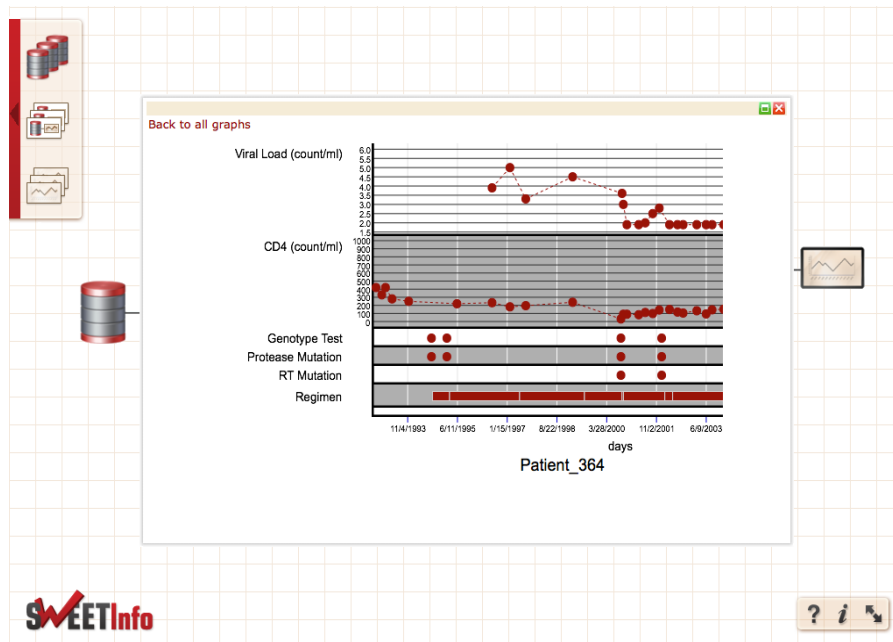


Figure 4. Screen shot of example SWEETInfo visualization.

4.4 Publication and Sharing

SWEETInfo includes a project-based mechanism that allows users to save their analyses and pipeline definitions. It supports multiple projects per user, with each project holding a data set and a pipeline. This mechanism allows users to store, reload, and execute the analyses defined in a pipeline. SWEETInfo also allows users to share pipelines with other users. Additional fine-grained sharing is also provided. Each operation node, which can define a set of complex constraints, can be shared. A library of operations can be maintained for each user and selectively shared. Shared operations can be easily modified to deal with similar data sets.

4.5 Results Exporting

After running an analysis users may wish to export their data for further analysis by other tools. SWEETInfo allows users to export data from any view node in a pipeline. These data are exported as a set of ZIPped CSV files, with one CSV file per variable.

5 Conclusion

A key feature of SWEETInfo is its use of semantic web technologies, particularly OWL, SWRL, and SQWRL. OWL is used to represent an information model for storing all user data. It is also used to represent all system information, such as user accounts, data transformations, and pipeline configurations. Additionally, all data transformations are executed using SWRL and SQWRL. These operations make use of an expressive temporal model and library developed for these languages [6]. The system demonstrates that these technologies can support the demands of complex highly interactive web-based applications, and that they can be successfully combined with traditional web technologies. The information model ontology and data analysis modules that result from this combination are reusable and can provide temporal representation and analysis functionality in other semantic web applications. SWEETInfo was released in September 2011 and is available at www.sweetinfo.org.

We are currently adding more expressivity through additional pipeline operations, such as, for example, statistical analysis operations. These operations will invoke external packages like R or SAS. We also intend to allow users to assemble individual operations into higher level components and then to share them. Our ultimate goal is to produce a library of reusable analysis modules, much like those provided by standard statistical packages.

Performance is also a key goal. While the current system performs satisfactorily, it does not scale to a large number of users. In particular, performing simultaneous analyses on a single server with SWRL and SQWRL is computationally expensive. We are investigating the use of a variety of strategies to tackle this problem.

Acknowledgements. This research was supported by grant 1R01LM009607 from the National Library of Medicine.

References

1. O'Connor, M.J., Bingen, M., Tu, S.W., Richards, A, Das, A.K. Web-Based Exploration of Temporal Data in Biomedicine. 7th International Conference on Web Information Systems and Technologies (WEBIST), Noordwijkerhout, Netherlands, 352-359 (2011)
2. OWL Overview: <http://www.w3.org/TR/owl-features/>
3. SWRL Submission: <http://www.w3.org/Submission/SWRL/>
4. Google Web Toolkit: <http://code.google.com/webtoolkit/>
5. Snodgrass, R.T. The TSQL2 Temporal Query Language, Boston, MA: Kluwer. (1995)
6. O'Connor, M.J. and Das, A.K. A Method for Representing and Querying Temporal Information in OWL. Springer-Verlag, CCIS 127, 97-110 (2011)
7. Allen, J.F. Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26(11) (1983)
8. O'Connor, M.J. and Das, A.K. SQWRL: a Query Language for OWL. OWL: Experiences and Directions, 6th International Workshop, Chantilly, VA (2009)