SPIN! Data Mining System
Based on Component Architecture

Alexandr Savinov

Fraunhofer Institute for Autonomous Intelligent Systems
Schloss Birlinghoven, Sankt-Augustin, D-53754 Germany
savinov@ais.fraunhofer.de

Abstract. The SPIN! data mining system has a component-based architecture, where each component encapsulates some specific functionality, e.g., it can be a data source, an analysis algorithm or visualization module. Individual components can be visually linked within one workspace for solving different data mining tasks. The SPIN! convenient user interface and flexible underlying component architecture provide a powerful integrated environment for executing main tasks constituting a typical data mining cycle: data preparation, analysis, and visualization.

1 Component Architecture

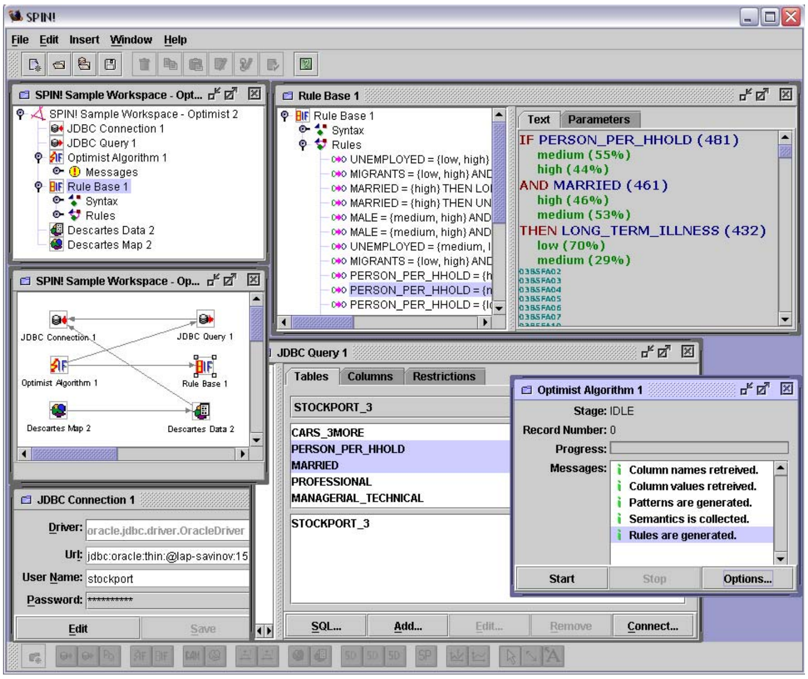
SPIN! data mining system has a component architecture. This means that it provides only an infrastructure and environment while all the system functionality comes from separate software modules called components. Components can be easily plugged-in the system thus allowing for an expansion of its capabilities. In this sense it is very similar to such general purpose environments as Eclipse. Each component is developed as an independent module for solving one or a limited number of tasks. For example, there may be components for data access, analysis or visualization. In order to solve complex problems components need to communicate and use each other.

All components are implemented on the basis of CoCon Common Connectivity Framework, which is a set of generic interfaces and objects in Java and allows components to communicate within one workspace.. The idea is that components can be connected by means of different types of connections. Currently there exist three connections: visual, hierarchical and user defined. Visual connections are used to link a component with its view (similar to Model-View-Controller architecture). Hierarchical connections are used to compose parent-child relationships among components within one workspace, e.g., between folder and its elements. The third and the main type is the user connection, which is used to arbitrary link components in the workspace according to the task to be solved (like in Clementine). It is important that components explicitly declare connectivity capabilities, i.e., how they can be connected and with what other components they can work.

2 Workspace Management

The SPIN! system is configured to include some set of components and then it automatically updates its main menu, tool bar and other functions (Fig. 1). An importance of such extensibility for data mining has been stressed in [5]. Workspace is a set of components and connections among them. It can be stored in or retrieved from a persistent storage. Workspace appears in two views: tree view and graph view. In tree view the hierarchical structure of the workspace is visualized where components are individual tree nodes, which can be expanded or collapsed. In graph view components are visualized as nodes of the graph while user connections are graph edges.

Components can be added to the workspace by choosing them either in menu or in tool bar. After a component has been added it should be connected with other relevant components. An easy and friendly way to do this consists in drawing an arrow from the source component to the target one. While adding connections between components the environment uses information about their connectivity so that only components, which are able to cooperate, can be really connected.



SPIN! Sample Workspace - Optimist 2
JDBC Connection 1
JDBC Query 1
Optimist Algorithm 1
Messages
Rule Base 1
Syntax
Rules
Descartes Data 2
Descartes Map 2

Rule Base 1
Syntax
Rules
UNEMPLOYED = {low, high}
MIGRANTS = {low, high} AND
MARRIED = {high} THEN LO
MARRIED = {high} THEN UN
MALE = {medium, high} AND
MALE = {medium, high} AND
UNEMPLOYED = {medium, l
MIGRANTS = {low, high} AND
PERSON_PER_HHOLD = {h
PERSON_PER_HHOLD = {n
PERSON_PER_HHOLD = {l

IF PERSON_PER_HHOLD (481)
medium (55%)
high (44%)
AND MARRIED (461)
high (46%)
medium (53%)
THEN LONG_TERM_ILLNESS (432)
low (70%)
medium (29%)
03B5FA02
03B5FA03
03B5FA04
03B5FA05
03B5FA06
03B5FA07

STOCKPORT_3
CARS_3MORE
PERSON_PER_HHOLD
MARRIED
PROFESSIONAL
MANAGERIAL_TECHNICAL
STOCKPORT_3

Stage: IDLE
Record Number: 0
Progress:
Messages:
Column names retreived.
Column values retreived.
Patterns are generated.
Semantics is collected.
Rules are generated.

Driver: oracle.jdbc.driver.OracleDriver
Url: jdbc:oracle:thin:@lap-savinov:15
User Name: stockport
Password: **********

Fig. 1. *The SPIN! Data mining system client interface: workspace (upper left window), rule base (upper right window), database connection (lower left window), database query and algorithm (lower right windows).*

Each component has an appropriate view, which is also a connectable component. Each component can be opened in a separate window so that the user can use its functions. When a workspace component is opened the system automatically creates a view, connects it with the model and then displays it within window.

3 Running Data Mining Algorithms

The typical data mining tasks include data preprocessing, analysis and visualization. For data access the SPIN! system includes Database Connection and Database Query components. The Database Connection is intended for storing information about the database where the data is stored. To use this database this component need to be connected with some other component, e.g., in graph view. The Database Query component describes one query, i.e., how its result set is generated from tables in the database. Essentially this component is a SQL query design tool, which allows for describing a result set by choosing tables, columns, restrictions, functions etc. Notice also that both Database Connection and Database Query components do not work by themselves and it is some other components that makes use of them. Such encapsulation of functionality and use of user connections to compose various aggregates has been one of the main design goals of the SPIN! component architecture.

Any knowledge discovery task includes data analysis step where the dataset obtained from preprocessing step is processed by some data mining algorithm. The SPIN! system currently includes several data mining algorithm components, e.g., subgroup discovery [1], rule induction based on empty intervals in data [4], spatial association rules [2], spatial cluster analysis, Bayesian analysis. To use some algorithm, say, Optimist rule induction [4], we need to add this component in the workspace and connect it with Database Query where the data is loaded from and Rule Base component where the result is stored.

The algorithm can be started by pressing the Start button in its view. After that it runs in a separate thread either on the client or within Enterprise Java Bean container on the server [3]. The rules generated by the algorithm are stored in Rule Base component connected to the algorithm. The rules can be visualized and studied by opening this component in a separate view.

References

1 Klösgen, W., May, M. Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database, PKDD 2002, Helsinki, Finland, August 2002, 275-286.

2 Lisi, F.A., Malerba, D., SPADA: A Spatial Association Discovery System. In A. Zanasi, C.A. Brebbia, N.F.F. Ebecken and P. Melli (Eds.), *Data Mining III*, Series: Management Information Systems, Vol. 6, 157-166, WIT Press, 2002.

3 May, M., Savinov, A. An integrated platform for spatial data mining and interactive visual analysis, Data Mining 2002, Third International Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields, 25-27 September 2002, Bologna, Italy, 51-60.

4 Savinov, A.: Mining Interesting Possibilistic Set-Valued Rules. In: Da Ruan and Etienne E. Kerre (eds.), Fuzzy If-Then Rules in Computational Intelligence: Theory and Applications, Kluwer, 2000, 107-133.

5 Wrobel, S., Wettschereck, D., Sommer, E., and Emde, W. (1996) Extensibility in Data Mining Systems. In *Proceedings of KDD'96 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, pp.214-219.