

# DSPROLOG

Quick Reference

by

Sebastian V. Huebner

Version 1.1, April 2007

# Chapter 1

## The DSPROLOG IDE

DSProlog is an experimental knowledge discovery support environment (KDSE) for KDD in PCM-data. Also, it may be seen as an integrated development environment (IDE) for design, implementation and test of subsymbolic and symbolic classifier systems for patterns in PCM-signals. Subsymbolic or first-level-classifiers (FLC) are based on time-frequency signatures of events in PCM-signals. Symbolic or second-level-classifiers (SLC) are based on symbolic signatures of patterns in sets of FLC-decisions. Tools provided by DSProlog cover the following areas:

- integrated project management,
- filtering, visualization, browsing, editing and audio-playing of PCM-data,
- time-frequency based feature extraction,
- subsymbolic and symbolic signature design,
- template matching and refinement,
- subsymbolic and symbolic data mining,
- similarity vector visualization,
- subsymbolic pattern search,
- hierarchical resolution of symbolic classifier systems,
- visualization of classifier decisions,
- classifier decision based extraction of signals and signatures,
- database management of classifier decisions,
- symbolic representation of events in signals,
- an interface to ISO-Prolog.

The name DSProlog is a combination of DSP (digital signal processing) and Prolog, a logic programming language created around 1972 by Alain Colmerauer and Philippe Roussel. The idea behind DSProlog is, that classifiers for complex patterns in signals may be represented as expressions in declarative logic and be computed in a similar way than Prolog predicates are. DSProlog is a way to bridge the gap between dense PCM-data and symbolic descriptions of deterministic patterns in the signal.

## 1.1 IDE Layout

DSProlog provides three different editors for three different types of objects: An editor for visual objects (VO editor), an editor for first level classifiers (FLC editor) and an editor for second level classifiers (SLC editor). The general editor layout is shown in figure 1.2. After launching DSProlog the VO editor is active. The other editors are activated by clicking one of the buttons of the toggle panel (figure 1.1):

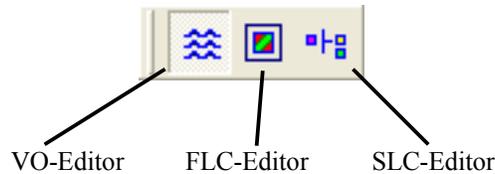


Figure 1.1: Toggle panel

## 1.2 Project Management



Select a DSProlog project file (.dsp) and open it by double clicking.



Show the 'Create Project' dialog window (see figure 1.3). Enter the name of the project and paths for the project itself, for the default audio directory and for the default export directory. Click the OK button to create the new DSProlog project. The project is written into a separate folder containing a project file (ProjectName.dsp) and a number of subfolders for different types of objects and datasets. Automatically created subfolders are 'Visual Objects', 'FL-Classifiers', 'SL-Classifiers', 'FL-Facts', 'SL-Facts' and 'Profiles' (for user defined profiles). After creation of a new project, all folders except the 'Profiles' folder are empty. While working, the different types of objects are saved in their specific folders by default. All file objects may be organized manually into hierarchies by making use of the windows file system and a file manager.

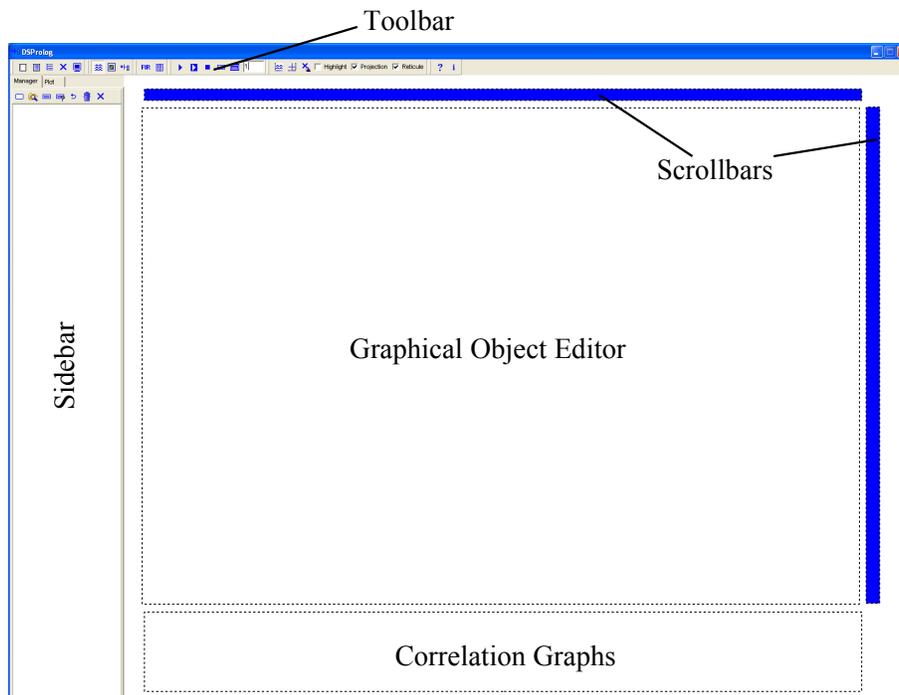


Figure 1.2: General editor layout

-  Show the project settings window. Options are the default paths for audio files and export data.
-  Close all project specific files and database connections.
-  Show the IDE settings window (figure 1.5). In this window IDE specific settings may be changed: Under the tab 'Export Graphs' the resolution and appearance of exported figures may be changed. Under the tab 'Audio' options for filtering audio data and reading headerless .dat files may be changed. Under the tab 'FLC signatures' settings for displaying FLC signatures may be changed. Under the tab 'General' settings for opening objects and other general settings may be changed.

Other typical project management operations need to be done manually:

- Move a DSProlog project by moving the entire project folder.
- Delete a DSProlog project by deleting the entire project folder.
- Rename a DSProlog project by renaming project folder and project file.

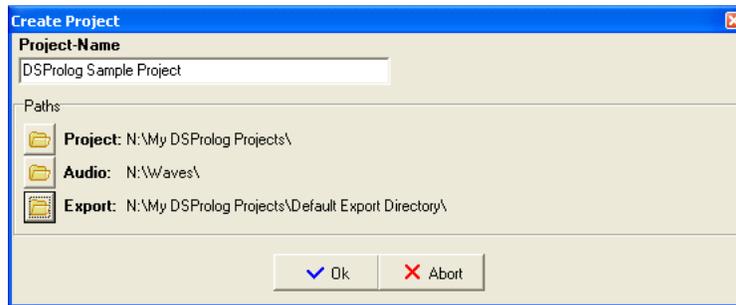


Figure 1.3: Create a new project

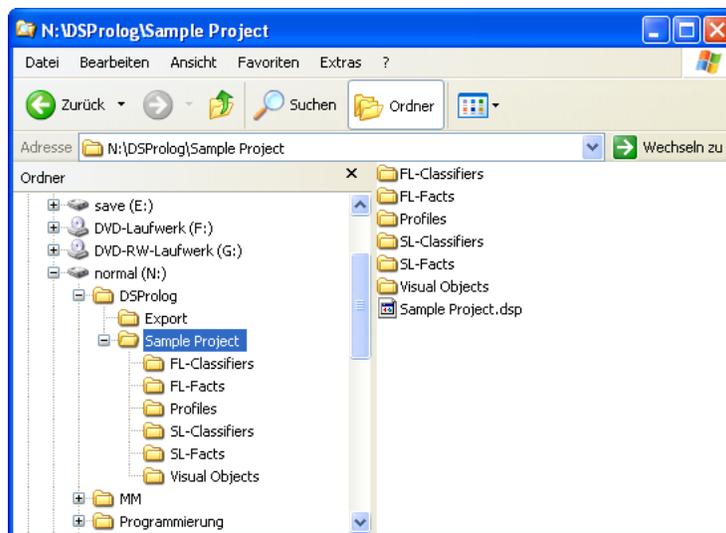


Figure 1.4: Structure of a project

### 1.3 Help and Info

- ?** Show the manual in the default PDF reader.
- i** Show the software license and additional information.

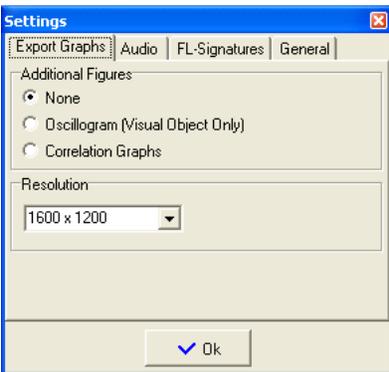


Figure 1.5: IDE settings window

## Chapter 2

# The Visual Object Editor

At first view, the visual object editor (VO editor) resembles popular audio editors, but it is distinct in many ways. It's main purpose is to solve two problems within the process of modeling audio classifiers: To produce exact visual representations of acoustical phenomena and to visualize classifier decisions. Currently the VO Editor supports three different types of visualizations: oscillograms, energy diagrams and DFT based spectrograms. Visual representations of acoustical phenomena may serve as templates or reference for the initialization of classifier signatures. Interactive visualization of classifier decisions is vital within the process of testing classifiers.

## 2.1 Files & Types

### 2.1.1 VO Files

Visualizations of PCM data are internally represented as objects and saved in VO files (.vo). These files are written by default into the 'Visual Objects' folder. It is possible to arrange VO files into hierarchies via the Windows file system. The files do not store PCM data but only those parameters, which are necessary for the generation of a visualization.

### 2.1.2 PCM Files

Currently the following PCM file formats are supported:

- Uncompressed Microsoft PCM (.wav) files, mono, 16 Bit with any sample rate.
- Uncompressed PCM (.dat) files, mono, 16 Bit with any sample rate. If you use .dat-files, you need to specify the sample rate, the number of channels and the encoding depth in the IDE settings window under the 'Audio' tab.

### 2.1.3 Profile Files

Certain parameters of VO's may be stored in profile files. Profile files may be applied to and driven from VO's (see below). There are two types of profiles located in the 'Profiles' folder:

- VO profiles (.vop) for an entire set of VO parameters.
- Color palette profiles (.ecp) for color palette parameters.

## 2.2 Management of Visual Objects

All VO files of a project are displayed in the object manager (see figure 2.1) located at the left sidebar. Open VO files by clicking onto them. The object manager provides the following tools:

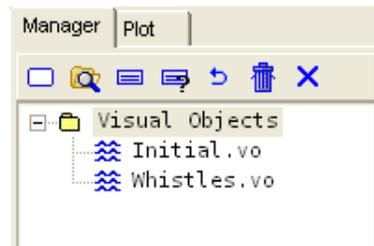


Figure 2.1: Visual object manager

-  Create a new visual object.
-  Open the 'Windows Explorer' in the 'Visual Objects' folder of the current project. You may arrange all files inside this folder via the windows file system. Any changes made here are displayed automatically in the tree view of the object manager.
-  Save the current VO.
-  Rename and save the current VO.
-  Undo changes by opening the last saved instance of the current VO.
-  Delete all objects and folders that are selected in the tree view of the object manager. Please note that files and folders cannot be restored after deletion!
-  Close the current VO and free memory.



### 2.3.2 Graphical Tools

 Export the current VO as a bitmap (24 Bit). Options may be changed in the IDE settings window under the tab 'Export Graphs'. An oscillogram or correlation graph may be added on top of the main graph (see 3.4).

 Show the mouse position.

 Delete all active search records.

 Highlight: Show search results in bright colors. Applies only if the 'Projection' check box is checked.

 Projection: Show FL and SL signatures of instantiated classifiers at the time-frequency position of search results.

 Reticule: Show reticules at the time-frequency position of search results.

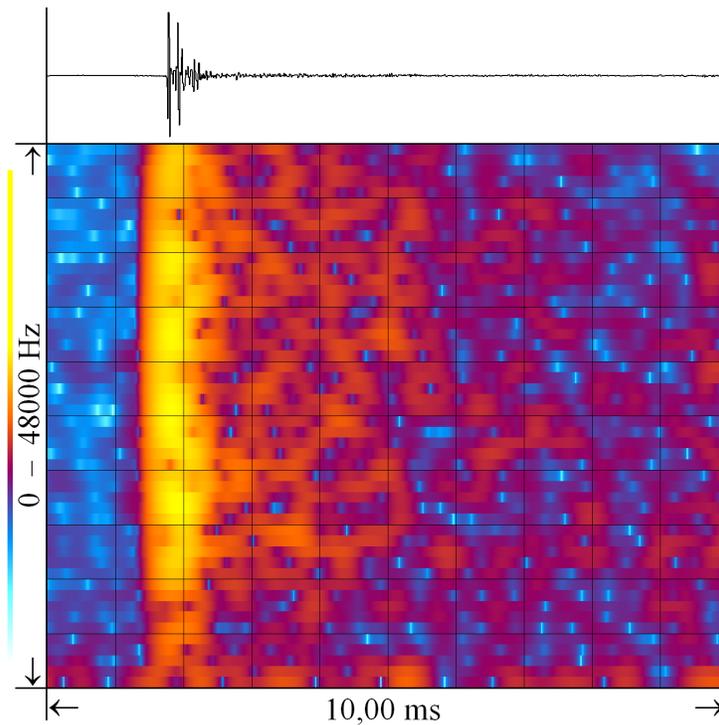


Figure 2.3: Exported graph with added oscillogram

### 2.3.3 Selection

An area in a VO may be selected by the two mouse buttons and the 'Ctrl' key (see figure 2.4):

**Left Button:** Set left time;

**Right Button:** Set right time;

**Ctrl + Left Button:** Set bottom frequency;

**Ctrl + Right Button:** Set top frequency;

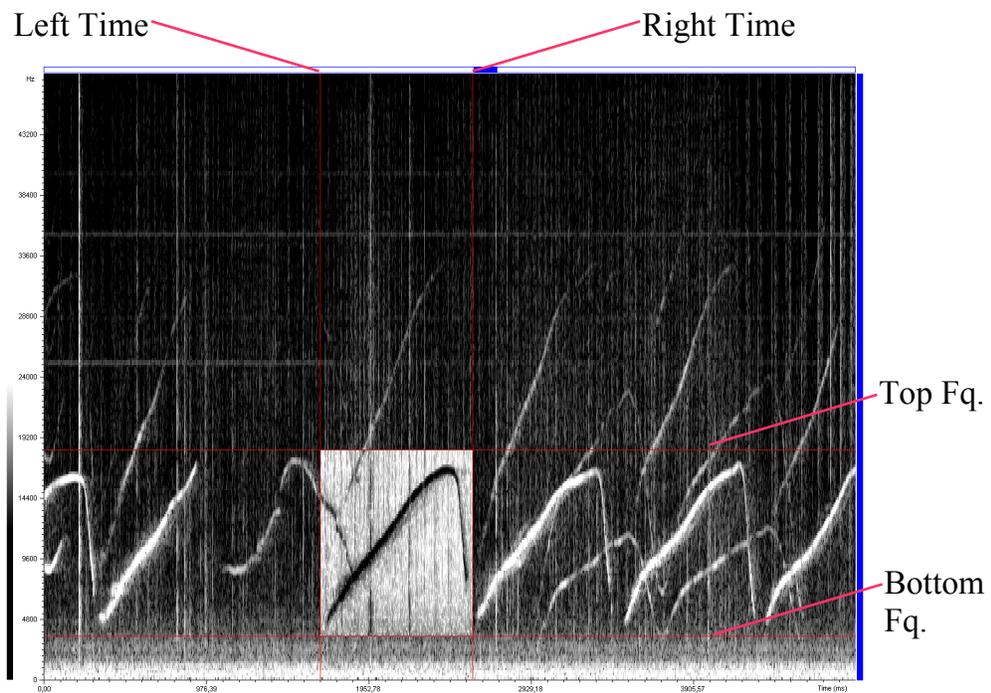


Figure 2.4: Selection

### 2.3.4 Playing and Extracting Audio Files

Audiofiles may be played or extracted either unfiltered or filtered. Filtering is done by a FIR bandpass filter. The filter window may be changed in the IDE settings window under the 'Audio' tab.

- ▶ Play the current VO unfiltered.

-  Play the current VO or the selection inside the current VO filtered.
-  Stop playing.
-  Extract an unfiltered PCM file from the current VO.
-  Extract a filtered PCM file from the current VO or from the selection inside the current VO.
-  Slow down factor valid both for playing and extraction of PCM files.

### 2.3.5 Adjusting Properties

Properties of visual objects may be changed in the property editor (see figure 2.1). The editor is divided into two sections: A navigation toolbar and a key-value editor. By using the buttons of the navigation toolbar you can 'navigate' inside PCM files, i.e. manipulate time and frequency properties of the current VO. By altering the values in the key-value editor you can change the appearance of the current VO in many ways. Also, you may save and apply certain settings in form of user defined profiles.

Time and frequency properties of the current VO may be changed either by setting values explicitly or by clicking the following buttons:

-  Zoom in along the x-axis.
-  Zoom out along the x-axis.
-  Move one step left.
-  Move one screen left.
-  Move one step right.
-  Move one screen right.
-  Zoom out to full along the x-axis.
-  Zoom into the current selection.

If the current VO is a spectrogram, frequency properties may be changed by clicking the following buttons:

-  Zoom in along the y-axis.

 Zoom out along the y-axis.

 Move one step up.

 Move one screen up.

 Move one step down.

 Move one screen down.

 Zoom out to full along the y-axis.

 Zoom out to full along both x- and y-axis.

VO properties may be changed by setting their values explicitly in the key–value editor. Please note that some of the properties shown in the editor are read only.

**Name:** The name of the VO (read only).

**Root:** The PCM file the VO relates to (read only).

**Profile:** Apply or save a VO profile containing all values except: Name, Root, Time 1, Time 2, Length, Frequency 1, Frequency 2 and Bandwidth.

**Graph:** {Oscillogram, Energy Plot, Spectrogram}: Visualization mode.

**Energy:** {Linear, Log10}: Display mode for amplitude, energy and Fourier coefficients.

**DFT Window:** {None, Hann, Hamming, Bartlett, Blackman, BlackmanStd, BlackmanOPT}: Windows for the DFT.

**DFT Width:** Size of the DFT window.

**Scaling %:** Multiplier for amplitude, energy and Fourier coefficients.

**Gain %:** Multiplier for the PCM signal.

**Grid:** {True, False}: Show/hide grid.

**Timescale:** {smp Local, ms Local, smp Global, ms Global}: x-scale mode.

**Inverted:** {True, False}: Display mode.

**GridColor:** Double click to change the color of the grid.

**BackgndColor:** Double click to change the color of the background.

**GraphColor:** Double click to change the color of the graph.

**Time 1:** First sample displayed by the VO.

**Time 2:** Last sample displayed by the VO.

**Length:** Length of VO in samples.

**Frequency 1 (Hz):** Bottom frequency of the current VO.

**Frequency 2 (Hz):** Top frequency of the current VO.

**Bandwidth:** Bandwidth of the current VO.

**dx/dt:** x-pixels of screen / number of samples (read only).

**dy/db:** y-pixels of screen / number of bands (read only). The box is displayed in red, when the number of bands exceeds the number of y-pixels.

**Phase:** {True, False}: Show the imaginary values of the DFT coefficients.

**Palette (DFT):** Apply or save a color palette defined by the values 'Blue 1', 'Blue 2', ..., 'Red 2'.

**Blue 1, Blue 2, ..., Red 2:** Color palette parameters.

### 2.3.6 Wizards



Wizard that guides through the process of selecting and applying VO profiles and color palette profiles to a set of VO files.

**FIR** Wizard that guides through the process of filtering a set of PCM files with a FIR bandpass filter. Please note that changes in the PCM files are irreversible!

### 2.3.7 Shortcuts

Most shortcuts are accessible from all editors. Changing time or frequency properties of a FLC object results in regenerating its underlying visualization matrix.

**Ctrl + N:** New project.

**Ctrl + O:** Open project.

**Ctrl + Q:** Toggle editors.

**Ctrl + S:** Save object.

**Ctrl + W:** Close object or project.

**Ctrl + Arrow Left:** Move screen left.

**Ctrl + Arrow Right:** Move screen right.  
**Ctrl + Arrow Up:** Move screen up.  
**Ctrl + Arrow Down:** Move screen down.  
**Arrow Left:** Move step left.  
**Arrow Up:** Move step up.  
**Arrow Right:** Move step right.  
**Arrow Down:** Move step down.  
**S:** Zoom into Selection.  
**Esc:** Redraw object and erase selection.  
**Del:** Delete SLC constituent (in SLC editor only).

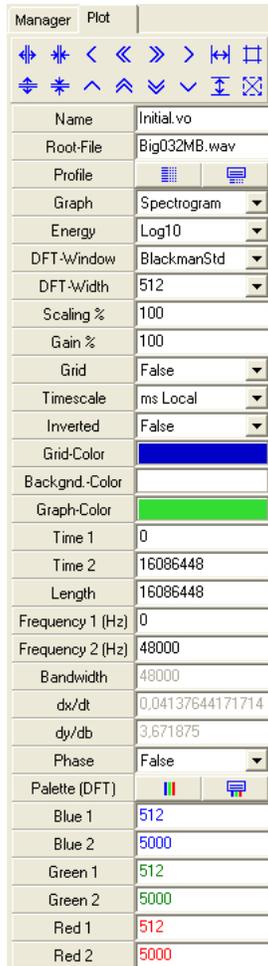


Figure 2.5: Property editor

## Chapter 3

# The FLC Editor

The FLC Editor serves the design, implementation and test of FL classifiers. Currently it supports classifiers based on spectrographic signatures of PCM data. FL classifier decisions may be saved and arranged with the help of an integrated database. Also, they may be converted to standard Prolog files.

### 3.1 Files & Types

#### 3.1.1 Model Files and FLC Files

FL classifiers are stored in files with `.flc` or `.mod` extension. Files with `.flc` extension are finished classifiers. Files with `.mod` extension are classifier models, i.e. classifiers storing both signature and underlying visualization matrix. Please note that `.mod` files are much bigger than `.flc` files. Both types of FLC files are written by default into the 'SL classifiers' folder. It is possible to arrange FLC files into hierarchies via the Windows file system.

#### 3.1.2 First Level Facts

First level facts are stored either in Paradox data tables or in instance **sequence** files (`.isq`). ISQ files store FL Facts in a dense form and can only be interpreted by DSProlog. They can be processed much quicker than database tables. The DSProlog database interface provides functions to convert ISQ files into data tables and the other way round. Both ISQ files and data tables may be converted into standard Prolog files (`.pl`).

#### 3.1.3 Profile Files

Certain parameters of FL classifiers may be stored in profile files. The following two types of FLC profile files may be applied to and driven from FL classifiers:

- Model profiles (`.modp`) for an entire set of FLC parameters.

- Pattern recognition profiles (.prp) for parameters that concern the linked pattern recognition algorithm only.

The two types of VO profiles (see section 2.1) can be used as well.

## 3.2 Management of First Level Classifiers

All FLC and MOD files of a project are shown in a tree view in the object manager located at the left sidebar. Open FLC and MOD files by clicking their icons. Figure 3.1 shows an example with the two different types of FLC objects: A MOD file with a blue red icon and a FLC file with a blue green icon. The object manager provides the following tools:

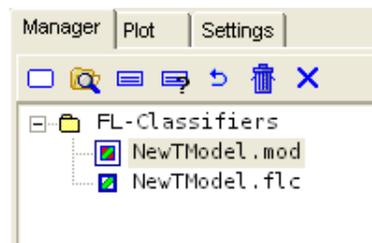


Figure 3.1: The FLC object manager

-  Initialize a new FLC model. Initialization fails if the current VO is either empty or not in spectrogram mode.
-  Open the 'Windows Explorer' in the 'FL-Classifiers' folder of the current project. You may arrange all files inside this folder via the windows file system. Any changes made here are displayed in the tree view of the object manager.
-  Save the current FLC or FLC model.
-  Rename and save the current FLC or FLC model. Select file extension (.flc or .mod) before saving.
-  Undo changes by opening the last saved instance of the current object.
-  Delete all objects and folders that are selected in the tree view of the object manager. Please note that files and folders cannot be restored after deletion!
-  Close the current object and free memory.

 Click to open a FLC. If the check box '1 Click Open' in the IDE settings window is unchecked, double click the icon.

 Click to open a FLC model. If the check box '1 Click Open' in the IDE settings window is unchecked, double click the icon.

## 3.3 Programming First Level Classifiers

### 3.3.1 Initialization

A FL Classifier is initialized by the following procedure:

1. Create a VO in spectrogram mode.
2. Adjust all necessary settings of the VO, so that the phenomenon to be classified by the FLC is displayed properly by the VO.
3. Switch to the FLC editor.
4. Click the 'New Object' button in the FLC object manager.
5. The new FLC model appears in the FLC editor (see 3.2).

### 3.3.2 Interactive Modification

**Inner frame** The coordinates of the area used for automatic signature extraction from the underlying visualization matrix, are defined by the yellow inner frame. The frame may be adjusted by moving its edges or the entire frame with the mouse. Press the left mouse button and drag the edges or the frame to the desired location.

**Search frame** The coordinates of the area used for *fact based* signature extraction and PCM file generation are defined by the orange search frame. The frame may be adjusted by moving its edges with the mouse. Press the left mouse button and drag the edges to the desired location.

**Eraser** Superfluous elements of a signature may be erased by the eraser tool. Click the 'Eraser' button in the toolbar and select a mode in the tool window by pressing a button (see figure 3.3). You can choose from thin, middle and thick erasers. If one of the green buttons is pressed, only members of  $\phi^+$  are erased. If one of the red buttons is pressed, only members of  $\phi^0$  are erased. To erase points in a signature drag over them with the mouse and keep the right mouse button pressed. While dragging a pink line appears, under which the elements of the signature are erased.

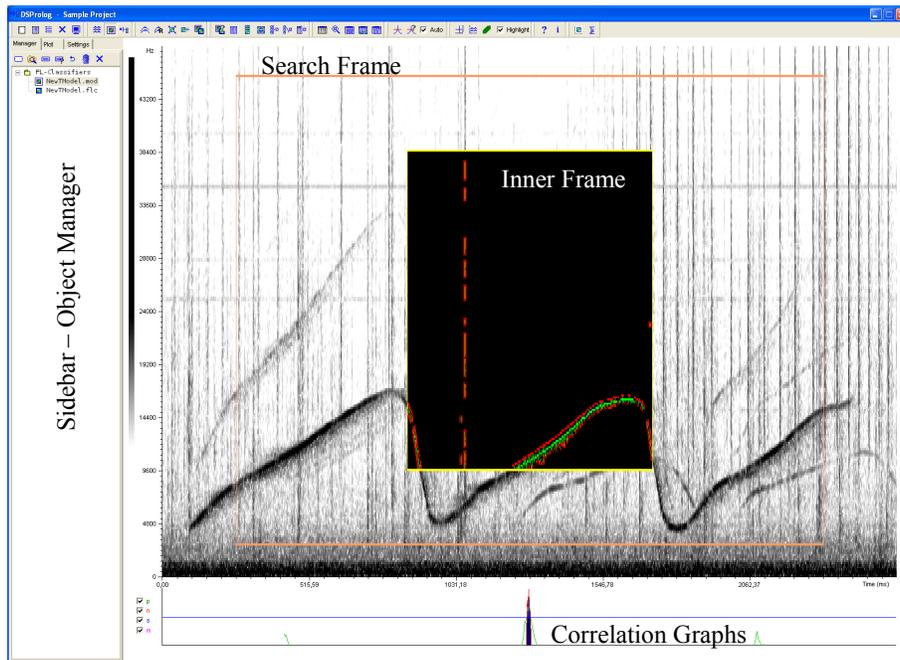


Figure 3.2: FLC model after initialization



Figure 3.3: The eraser tool window

### 3.3.3 Graphical Tools



Export the current FLC as a bitmap (24 Bit). Options may be changed in the IDE settings window under the tab 'Export Graphs'.



Show the mouse position.



Highlight: Signatures may be displayed in normal (see figure 3.5) or in highlight mode (see figure 3.6). In highlight mode the three parts of a signature are drawn in different colors:  $\phi^+$  is green,  $\phi^-$  is red and  $\phi^0$  is blue. Select in the IDE settings window under the 'FL Signatures' tabsheet those parts of the signature that should be visible.

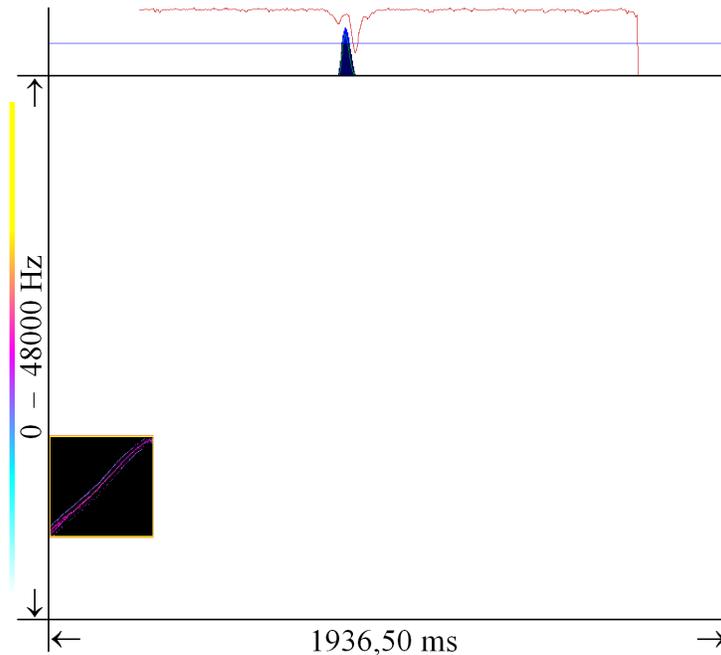


Figure 3.4: Exported FLC visualization with added correlation graph

### 3.3.4 General Purpose Tools



Apply all visual settings of the current FLC to the current VO.



Create the VO which the initial FLC was driven from and apply all visual settings of the current FLC. Fails if the original PCM file is unavailable.

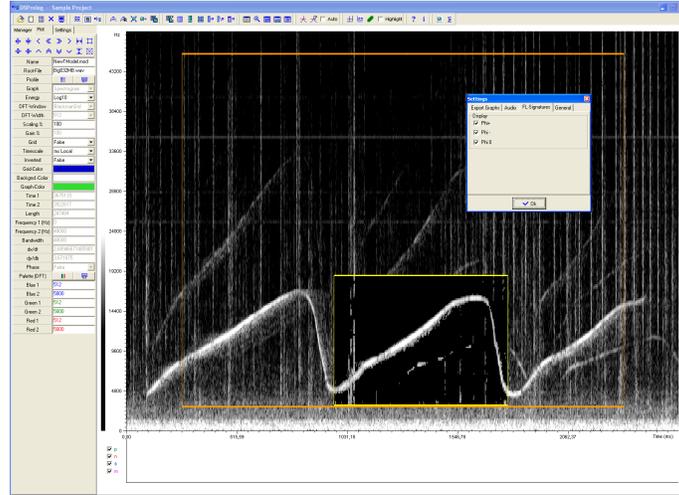


Figure 3.5: Unrefined FLC model. The phenomenon to be classified is the whistle in the middle of the sequence.

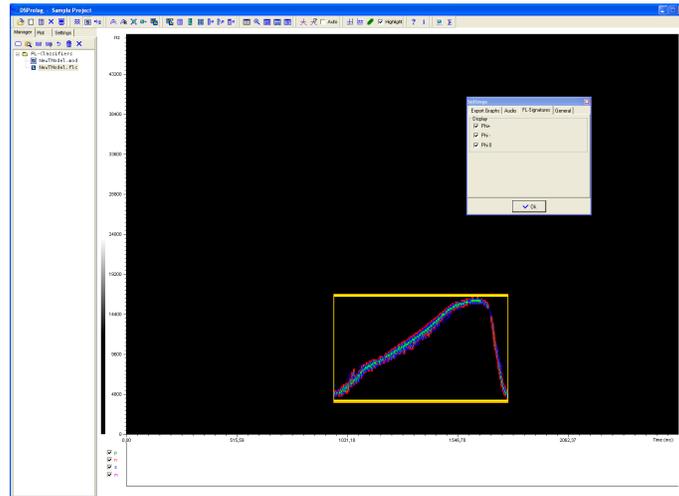


Figure 3.6: Cleaned FLC with signature in highlight mode.



Minimize the size of both the inner frame and the search frame.



Normalize time of current FLC or FLC model.



Convert a FLC model into a FLC file and save the file.

### 3.3.5 Adjusting Properties

Visual properties of FL classifiers may be changed the same way, the properties of visual objects are changed (see section 2.3.5). Please note that some of the VO properties are disabled in the property editor and that visual properties (except the 'Gain' property) do not affect the behavior of a FLC in pattern recognition tasks.

FLC-specific model and pattern recognition properties may be changed in an additional property editor, located in the sidebar under the tab 'Settings' (see figure 3.7).

**Preset:** Apply or save an entire FLC model profile containing all values except the *nom*-parameter.

**Nom:** Number of merged items that formed the signature (*nom*-parameter).

**Energy:** {Linear, Log10}: Applies to all elements of the signature.

**x-Contour:** {True, False}: Select local maxima in rows of the underlying matrix.

**y-Contour:** {True, False}: Select local maxima in columns of the underlying matrix.

**RetColor:** Double click to change the color of the reticule that is used to visualize classifier decisions.

**Horizontal:** Maximum allowed horizontal distance for elements of the underlying matrix from the closest element of  $\phi^+$  to be copied into  $\phi^-$ .

**Vertical:** Maximum allowed vertical distance for elements of the underlying matrix from the closest element of  $\phi^+$  to be copied into  $\phi^-$ .

**x-Density:** Filter density for elements in the rows of the underlying matrix.

**y-Density:** Filter density for elements in the columns of the underlying matrix.

**c:** Constant  $c_1$  in the function  $d^+$ .

**dy:** Constant  $dy_1$  in the function  $d^+$ .

**th>:** Minimum threshold for the elements in the underlying matrix to be copied to  $\phi^+$ .

Manager	Plot	Settings
Preset		 
Nom		1
Energy		Log10
x-Contour		True
y-Contour		True
RetColor		
Horizontal		2
Vertical		1
x-Density		1
y-Density		1
c		100
dy		100
Threshold >		30
n =		657
c		200
dy		100
Threshold <		15
n =		909
PR-Profile		 
Smooth		1
SimilarityThreshold		50
Algorithm		Distance
Vectors		Euclidean Prune
Background		Always

Figure 3.7: Model and pattern recognition property editor

- n:** Number of elements in  $\phi^+$ .
- c:** Constant  $c_2$  in the function  $d^-$ .
- dy:** Constant  $dy_2$  in the function  $d^-$ .
- th<:** Maximum threshold for the elements in the underlying matrix to be copied to  $\phi^-$ .
- n:** Number of elements in  $\phi^-$ .
- PR Profile:** Apply or save a pattern recognition profile containing the values: Smooth, Similarity th, Vectors and Background.
- Smooth:** Smooth factor for the class correlation function.
- Similarity th:** Threshold  $th$  for the overall class correlation function  $s$ .
- Algorithm:**  $\Gamma = \{\text{Distance, Gradient}\}$ . The algorithm that generates the annotations.
- Vectors:**  $C = \{\text{Euclidean, EuclideanPruned, Vector Correlation}\}$ . The basic class correlation function.
- Phi:**  $\{\text{Always, Never, Selective}\}$ . Compute  $d^-$  either always, never or only in those intervals where  $s > th$ .

### 3.3.6 Wizards



Wizard that guides through the process of converting FLC model files into FLC files.



Wizard that guides through the process of selecting and applying profiles to a set of FLC model files or FLC files.



Wizard that guides through the process of cloning FL classifiers.



Wizard that guides through the process of mirroring FL classifiers.



Wizard that guides through the process of similarity based merging of FL classifiers.



Wizard that guides through the process of subtracting a signature from a set of FL classifiers. Please note that all FL classifiers have to be exactly of the same kind (i.e. have identical model parameters) and that the signature to be subtracted has to be in a FLC file named 'Subtrahend.flc' in the same directory.

-  Wizard that guides through the process of creating a merged signature from a set of FL facts. For automatic signature merge the search frames of the FL classifiers are valid.

### 3.3.7 Testing FL Classifiers

-  Scan the current VO and generate correlation graphs.
-  Select a set of FL classifiers and scan the current VO. Does not generate correlation graphs.
-  Auto: Automatically scan the current VO and generate correlation graphs.

The scan procedure tries to find instances of FL classifiers. If a scan is successful it may result in a search record containing a number of FL facts. This search record is active in the background unless it is deleted by clicking the 'Delete Current Search Records' button in the VO editor. The current active search record may be converted into an ISQ file or data table (see below).

Items in the active search record may be visualized in the current VO if they apply to the PCM file section the current VO is showing (see figures 3.8 and 3.9).

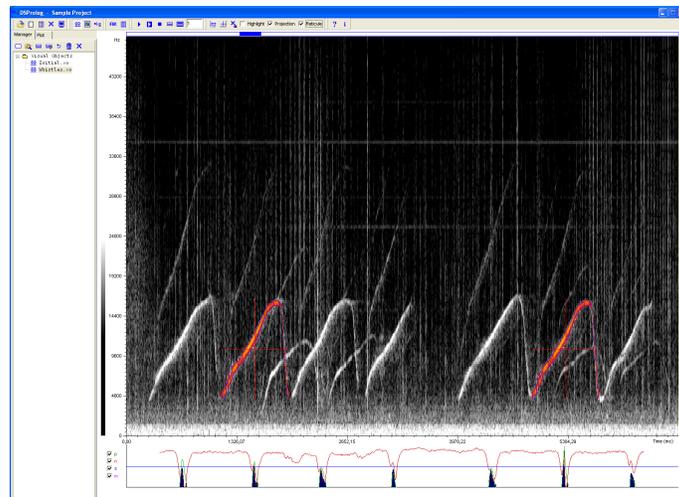


Figure 3.8: Correlation graphs and visualized decisions of the FLC in figure 3.6. The signature of the FLC is mapped onto the VO in contrast colors.

The correlation graphs show the behavior of the class correlation function and its constituents (see figure 3.10).

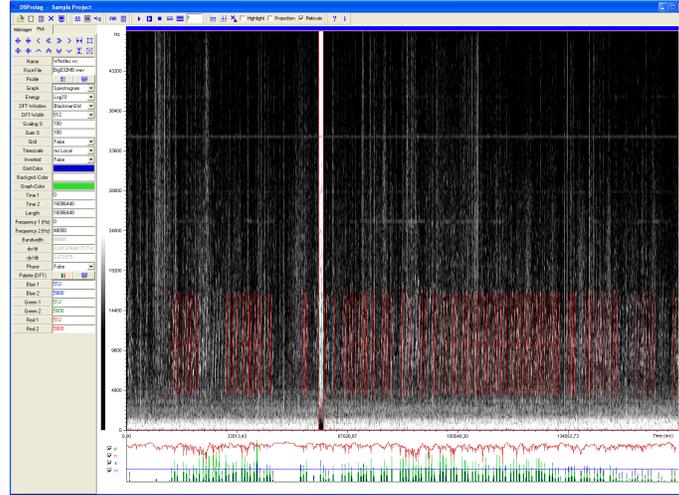


Figure 3.9: Correlation graphs and many decisions of the FLC in figure 3.6. Decisions appear as red lines. Zoom to a decision by marking a selection (inverted) and pressing the key 's'.

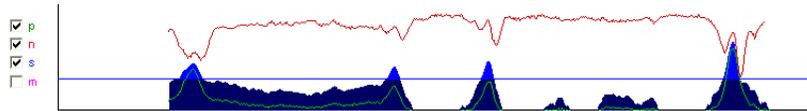


Figure 3.10: Correlation graphs: Green is the graph of  $d^+$ , red is the graph of  $d^-$ , blue is the graph of  $s$ . The blue horizontal line indicates  $th$ . Graphs are visible if their check boxes are checked. The graph  $m$  (unchecked) is the gradient of  $s$ .

## 3.4 The FL Facts Database

DSProlog provides an integrated database that facilitates the handling of FL facts. It supports SQL and provides a number of useful conversion routines. All FL facts may be exported in standard Prolog notation.

### 3.4.1 The FL Facts Database Interface

 Show the FL facts database interface (see figure 3.11).

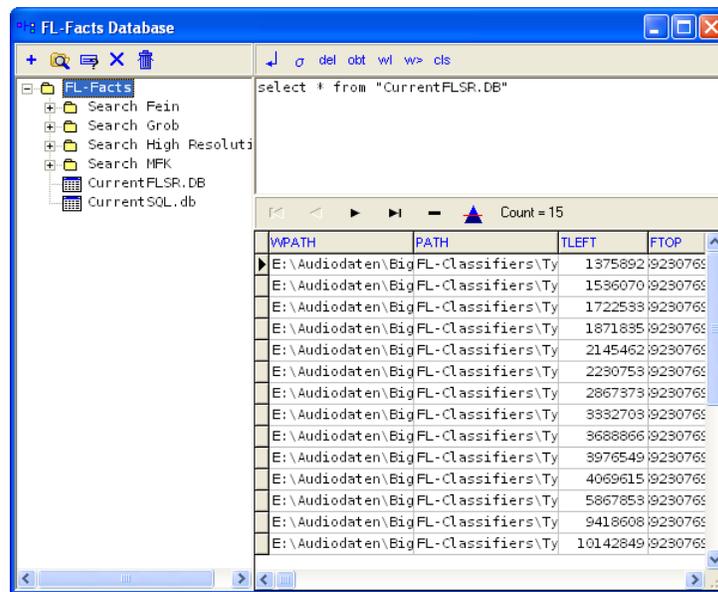


Figure 3.11: The FL facts database interface

#### Database Management Tools

-  Create a table named 'CurrentFLSR.db', which contains all items of the currently active search record.
-  Open the 'Windows Explorer' in the 'FL Facts' folder of the current project. You may arrange all files inside this folder via the windows file system. Any changes made here are displayed automatically in the tree view of the database interface.
-  Rename and save the currently open data table.

 Close the currently open data table.

 Delete all objects and folders that are selected in the tree view of the database interface. Please note that files and folders cannot be restored after deletion!

## SQL

 Execute the current SQL expression.

$\sigma$  Add 'select \* from' to the current SQL expression.

**del** Add 'Delete from' to the current SQL expression.

**obt** Add 'Order by tleft' to the current SQL expression.

**wl** Add 'Where \* like "%\*%"' to the current SQL expression.

**w>** Add 'Where sim > 80' to the current SQL expression.

**cls** Clear the current SQL expression.

## FL facts data tables

There are two types of FL facts data tables: ISQ tables and ordinary data tables. They are marked by two different icons:

 ISQ table. Double click the icon to make the table the current search record.

 Data table. Double click the icon to run the SQL expression 'select \* from *TableName*'. The result appears in the file 'CurrentSQL.db' and is shown in the data grid.

The column headers in the data grid indicate the following properties:

**WPATH** PCM file the fact relates to.

**PATH** Subfolder and name of the classifier.

**TLEFT** Start time of the annotation.

**FTOP** Top frequency of the annotation.

**NS** Value of  $\phi^-$ .

**PS** Value of  $\phi^+$ .

**SIM** Value of  $s$ .

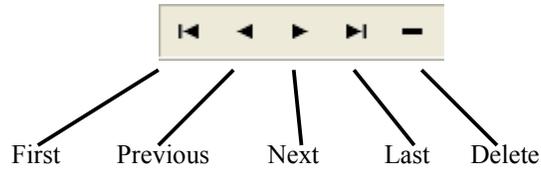


Figure 3.12: The FL facts data table browser

**VMODE** Virtual mode – used only in SL Facts (see next chapter).

You can browse an open data table via the data table browser shown in figure 3.12.

Other options are:



Make the currently open table the active search record.

- Click a field in the column WPATH to show the instance in the VO editor.
- Click a field in the column PATH to open the respective FL classifier in the FLC editor.

### 3.4.2 Database wizards



Wizard that guides through the process of creating new FL facts by searching in a set of PCM files.



Wizard that guides through the process of extracting PCM files from a dataset. For automatic PCM file extraction the search frames of the FLC models are valid.



Wizard that guides through the process of extracting FL classifiers from a dataset. For automatic FL classifier extraction the search frames of the FLC models are valid.



Wizard that guides through the process of converting FL facts into standard Prolog formatted files.

### 3.4.3 FL facts in Prolog

In Prolog notation a FL fact is written in the following way:

```
flf([flc, % ID of flc
     wave, % pcm file
     tleft, % first sample
     ftop, % top frequency
```

```
twidth, % length in samples
trelative % always 0 - SLC only
fheight, % bandwidth
nom, % number of merged
vmode, % always 0 - SLC only
[sim,psim,nsim]). % similarity
```

## Chapter 4

# The SLC Editor

The SLC editor serves the design, implementation and test of SL classifiers. Currently it supports SLC based on decisions of FL classifiers. SLC run on PCM data directly. To guide search in large files it is possible to set up a classifier resolution hierarchy. SL classifier decisions can be saved and arranged with the help of an integrated database. Also, they may be converted to standard Prolog files.

### 4.1 Files & Types

#### 4.1.1 SLC Files

SL classifiers are internally represented as objects and saved in SLC files (.slc). SLC files are written by default into the 'SL Classifiers' folder of the current project. It is possible to arrange SLC files into hierarchies via the Windows file system.

#### 4.1.2 Second Level Facts

Second level facts are stored in Paradox data tables (.db). Data tables may be converted into standard Prolog files (.pl).

#### 4.1.3 Profile Files

Certain parameters of SL classifiers may be stored in SLC profile files (.slcp). SLC profiles may be applied to and driven from SL classifiers. Additionally the two types of VO profiles (see section 2.1) can be used.

## 4.2 Management of Second Level Classifiers

All SLC files of a project are shown in a tree view in the object manager, located at the left sidebar. Open SLC files by clicking their icons. The object manager provides the following tools:

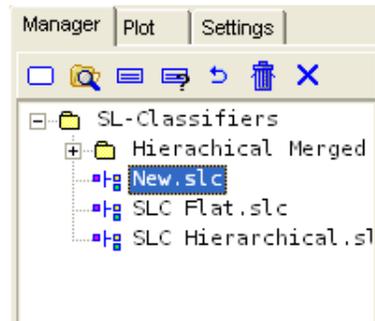


Figure 4.1: The SLC object manager

-  Initialize a new SLC. Initialization fails if the current VO is either empty or not in spectrogram mode.
-  Open the 'Windows Explorer' in the 'SL Classifiers' folder of the current project. You may arrange all files inside this folder via the windows file system. Any changes made here are displayed in the tree view of the object manager.
-  Save the current SLC.
-  Rename and save the current SLC.
-  Undo changes by opening the last saved instance of the current object.
-  Delete all objects and folders that are selected in the tree view of the object manager. Please note that files and folders cannot be restored after deletion!
-  Close the current object and free memory.
-  Click to open a SLC. If the check box '1 Click Open' in the IDE settings window is unchecked, double click the icon.

## 4.3 Programming Second Level Classifiers

### 4.3.1 Initialization

A SL classifier is initialized by the following procedure:

1. Create a VO in spectrogram mode.
2. Adjust all necessary settings of the VO, so that the phenomenon to be classified by the SLC is displayed properly by the VO.
3. Switch to the SLC editor.
4. Click the 'New Object' button in the SLC object manager.
5. The new SLC appears in the SLC editor (see 4.2). Initially its signature is empty.

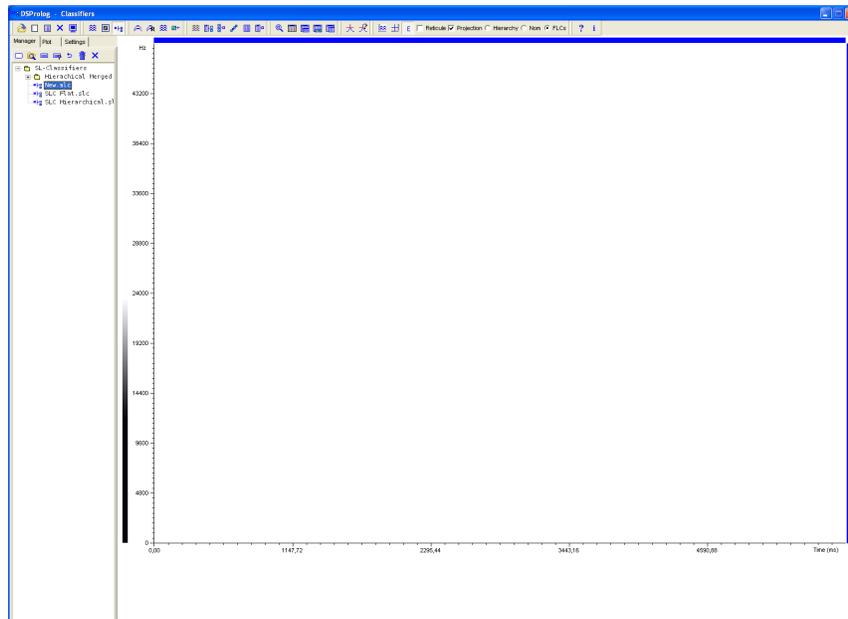


Figure 4.2: SLC after initialization

### 4.3.2 Wizards



Wizard that guides through the process of interpreting a VO by means of a set of FLC. Requires an open VO in the VO editor and an open SLC in the SLC editor. Interpretations are created from either the entire VO or from a selection inside.

-  Wizard that guides through the process of interpreting a set of FL facts in order to create a set of SLC. Runs in two modes: Extraction of sequences or extraction of tuples. Requires specification of neighborhood relation for the elements of a sequence or tuple.
-  Wizard that guides through the process of similarity based merge of symbolic signatures.
-  Wizard that guides through the process of cleaning symbolic signatures from superfluous elements, i.e. elements with low *nom* values.
-  Wizard that guides through the process of selecting and applying profiles to a set of SLC files.
-  Wizard that guides through the process of creating a merged signature from a set of SL facts.

### 4.3.3 Interactive Modification

The SLC editor knows two modes: the edit mode and the zoom-and-select mode. Click the 'Edit-Mode' button in the toolbar to switch between the two modes.

#### Edit mode

In edit mode it is possible to select single symbolic elements of a signature with the mouse pointer by clicking their reticules.

Selecting single elements is necessary for creating parent child relationships between the constituents of a signature: Left mouse click selects an element as parent, right mouse click selects an element as child. It is possible to select one parent and a number of children simultaneously. To accept the constellation click the 'Accept' button (see below) in the SLC property editor. The resulting hierarchy is shown in the tree view of the SLC property editor (see 4.1).

Selected elements may also be erased pressing the 'Del' key. Alternatively, selections may be made directly in the tree view of the SLC property editor.

#### Zoom-and-select mode

All functions and shortcuts for zooming and marking selections known from the VO editor are available in zoom-and-select mode (see section 2.3.5, 2.3.3 and 2.3.7)

### 4.3.4 Graphical Tools

-  Export the current SLC as a bitmap (24 Bit). Options may be changed in the IDE settings window under the tab 'Export Graphs'.

 Show the mouse position.

**E** Switch to edit mode (down) or zoom-and-select mode (up).

 Reticule: Show a reticule indicating the position of each symbolic element and of the entire symbolic signature.

 Projection: Show the FL signature of each visible symbolic element.

 Hierarchy: Color coded highlighting of the hierarchy inside the symbolic signature. The color code is reflected in the color palette (see figure 4.3).

 Nom: Color coded highlighting of the *nom*-value of symbolic elements. The color is reflected in the color palette (see figure 4.4).

 FLCs: Show FL signatures of symbolic elements. The color palette of the first FL signature of the SLC is reflected in the color palette of the SLC (see figure 4.5).

### 4.3.5 General Purpose Tools

 Apply all visual settings of the current SLC to the current VO.

 Create the VO, which the initial SLC was driven from, and apply all visual settings of the current SLC. Fails if the original PCM file is unavailable.

 Apply VO settings to classifier. If in the IDE settings window the check box 'Apply VO Settings to SLC' under the tab 'General' is checked, VO settings are applied automatically to a classifier when opening it.

 Normalize time of current SLC.

### 4.3.6 Adjusting Properties

Visual properties of SL classifiers may be changed the same way, the properties of visual objects are changed (see section 2.3.5). Please note that most of the VO properties are disabled in the property editor and that visual properties do not affect the behavior of a SLC in pattern recognition tasks.

SLC properties may be changed in an additional property editor, located in the sidebar under the tab 'Settings' (see figure 4.6).

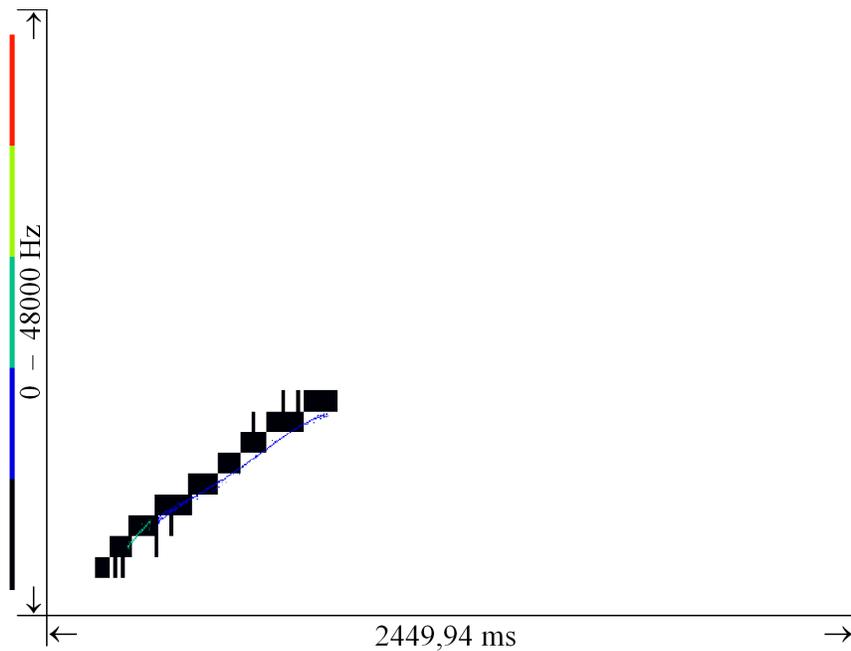


Figure 4.3: SLC with color coded highlighting of resolution hierarchy. The SLC contains three symbolic elements: one for rough, one for better and one for very fine analysis of PCM data. The rough classifier is black, the fine is blue and the finest is green. Search in PCM data may be significantly increased by setting up resolution hierarchies.

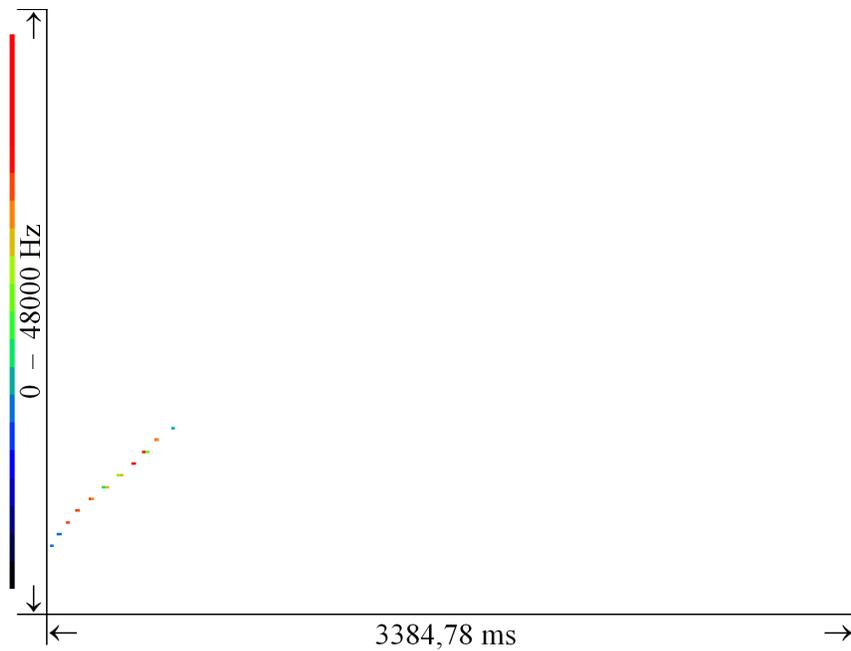


Figure 4.4: Merged SLC with color coded highlighting of *nom*-value. The SLC was created by merging 18 SLC.

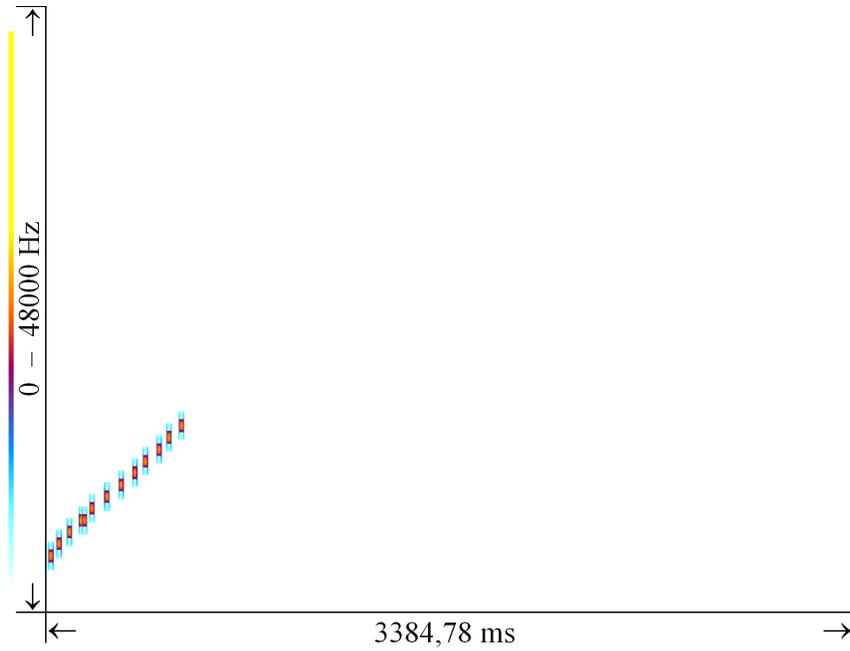


Figure 4.5: Automatically created SLC. The SLC models a sequence of FL facts created by interpreting a whistle.

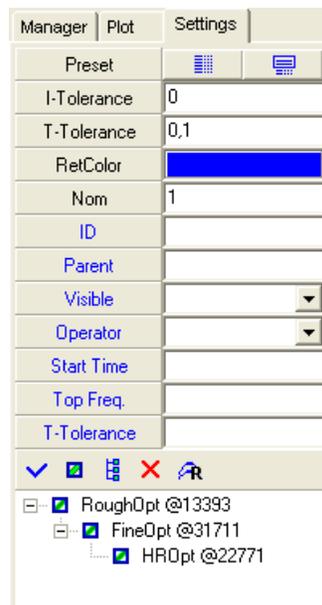


Figure 4.6: SLC property editor

**Preset:** Apply or save a SLC profile. The profile contains the following parameters: I-Tolerance, T-Tolerance and RetColor.

**I-Tolerance:** Number of failed constituent instantiations tolerated when instantiating a whole SLC.

**T-Tolerance:** Overall factor, defining the maximum time deviation tolerated when instantiating a symbolic element.

**RetColor:** Double click to change the color of the reticule that is used to visualize classifier decisions.

**Nom:** Number of merged items that formed the signature (*nom*-parameter).

**ID:** Automatically created identifier of the selected symbolic element.

**Parent:** ID of the parent of the selected symbolic element.

**Visible:** Indicates if the selected symbolic element is visible.

**Operator:** Logic operator {AND, NOT} of the selected symbolic element.

**Start Time:** Start time of the selected symbolic element.

**Top Freq.:** Top frequency of the selected symbolic element.

**T-Tolerance:** Individual factor, defining the maximum time deviation tolerated when instantiating the selected symbolic element.



Accept the selection and create parent child relationships between the selected elements. The resulting hierarchy is shown in the tree view of the SLC property editor (see 4.6).



Unselect all symbolic elements.



Erase all parent child relationships in the signature.



Delete selected symbolic element.



Open the FLC the selected symbolic element relates to.

### 4.3.7 Testing SL Classifiers



Scan the current VO. Does not generate correlation graphs.



Select a set of SL classifiers and scan the current VO.

The scan procedure tries to find instances of SL classifiers. If a scan is successful it may result in a search record containing a number of SL facts. This search record is active in the background unless it is deleted by clicking the 'Delete Current Search Records' button in the VO editor. The current active search record may be converted into a data table (see below).

Items in the active search record may be visualized in the current VO if they apply to the PCM file section the current VO is showing (see figures 4.7 and 4.8).

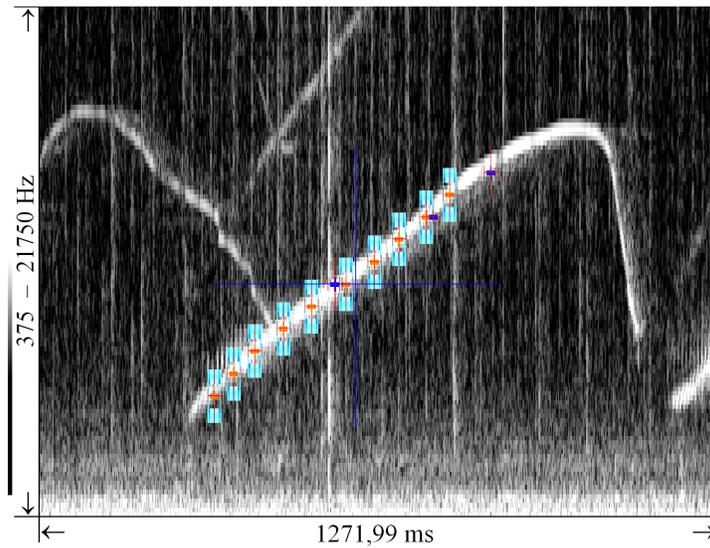


Figure 4.7: Visualized decision of the SLC in figure 4.4. The signature of the SLC is mapped into the VO in contrast colors. Uninstantiated symbolic elements are painted in blue.

## 4.4 The SL Facts Database

DSProlog provides an integrated database that facilitates the handling of SL facts. It supports SQL and provides a number of useful conversion routines. All SL facts may be exported in standard Prolog notation.

### 4.4.1 The SL Facts Database Interface

 Show the SL facts database interface (see figure 4.9).

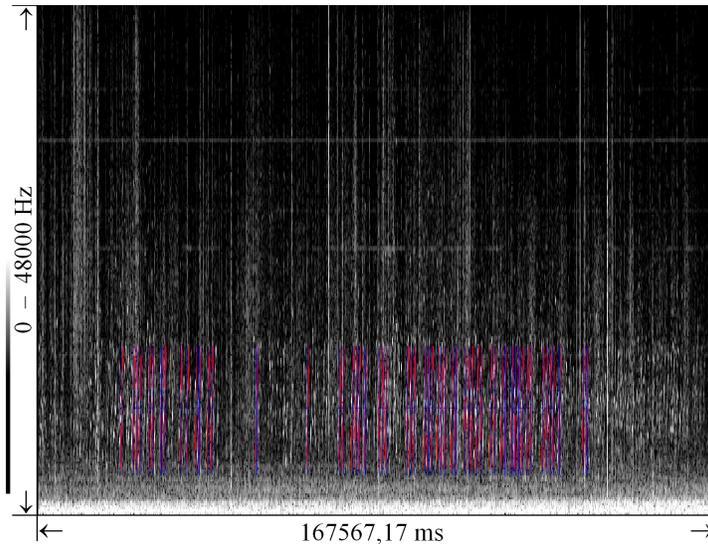


Figure 4.8: Many decisions of the SLC in figure 4.4. Decisions are indicated with blue lines.

### Database Management Tools

- + Create a table named 'CurrentSLSR.db', which contains all items of the currently active search record.
-  Open the 'Windows Explorer' in the 'SL Facts' folder of the current project. You may arrange all files inside this folder via the windows file system. Any changes made here are displayed automatically in the tree view of the database interface.
-  Rename and save the currently open data table.
-  Close the currently open data table.
-  Delete all objects and folders that are selected in the tree view of the database interface. Please note that files and folders cannot be restored after deletion!

### SQL

-  Execute the current SQL expression.
- $\sigma$  Add 'select \* from' to the current SQL expression.

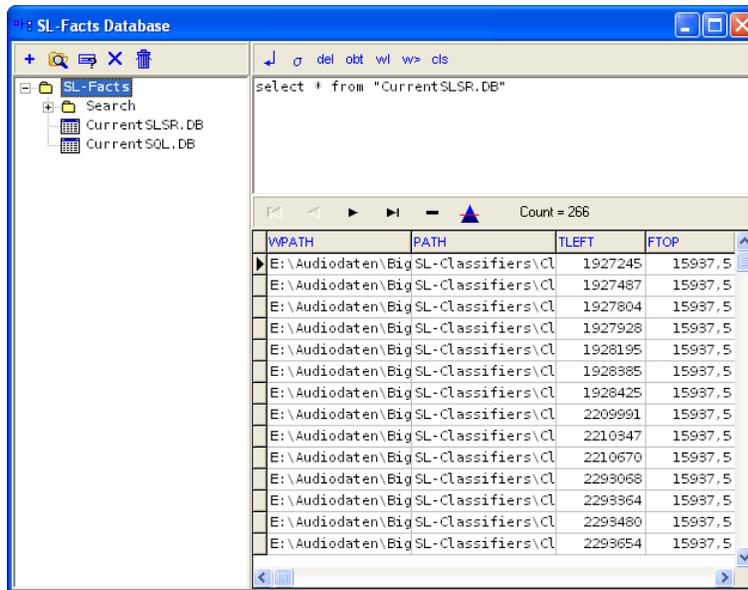


Figure 4.9: The SL facts database interface

- del** Add 'Delete from' to the current SQL expression.
- obt** Add 'Order by tleft' to the current SQL expression.
- wl** Add 'Where \* like "%\*%"' to the current SQL expression.
- w>** Add 'Where sim > 80' to the current SQL expression.
- cls** Clear the current SQL expression.

### SL Facts data tables

SL facts are stored in data tables. The table contains a blob string storing the status (time, frequency, ns, ps, sim and virtual mode) of each symbolic element of the SLC at the time of instantiation.

-  SL facts data table: Double click the icon to run the SQL expression 'select \* from *TableName*'. The result appears in the file 'CurrentSQL.db' and is shown in the data grid.

The column headers in the data grid indicate the following properties:

- WPATH** PCM file the fact relates to.
- PATH** Subfolder and name of the classifier.
- TLEFT** Start time of the annotation.

**FTOP** Top frequency of the annotation.

**NS** Cumulated value of  $\phi^-$ .

**PS** Cumulated value of  $\phi^+$ .

**SIM** Cumulated value of  $s$ .

The virtual mode (vm) indicates how a symbolic element was instantiated:  $vm := 1$  if an element was properly instantiated,  $vm := 2$  if an element could be instantiated only virtually (i.e. true instantiation failed, but this was tolerated) and  $vm := 3$  if a negated element was instantiated virtually as such (i.e the element could not be instantiated, but this is ok as it is connected by the NOT-operator in the symbolic signature). The virtual mode is reflected in the visualized classifier decisions: if  $vm = 1$   $\phi$  is drawn normally, if  $vm = 2$   $\phi^+$  is drawn in blue and if  $vm = 3$   $\phi^+$  is drawn in red.

You can browse an open data table via the data table browser shown in figure 4.10.

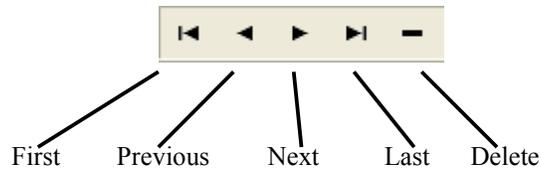


Figure 4.10: The SL Facts data table Browser

Other options are:



Make the currently open table the active search record.

- Click a field in the column WPATH to show the instance in the VO editor.
- Click a field in the column PATH to open the respective SL classifier in the SLC editor.

#### 4.4.2 Database Wizards



Wizard that guides through the process of creating new SL facts by searching in a set of PCM files.



Wizard that guides through the process of extracting PCM files from a dataset.



Wizard that guides through the process of extracting SL classifiers from a dataset.



Wizard that guides through the process of converting SL facts into standard Prolog formatted files.

### 4.4.3 SL facts in Prolog

In Prolog notation a SL fact is written in the following way:

```
slf([slc, % ID of SLC
    wave, % pcm file
    tleft, % first sample
    ftop, % top frequency
    twidth, % length in samples
    fheight, % bandwidth
    nom, % number of merged
    n, % Elements in SLC
    [sim,psim,nsim], % cumulated similarity
    [c_0,...,c_n]]). % list of constituents
```

where each

```
c_i = [flc, % ID of FLC
    wave, % pcm file
    tleft, % first sample
    ftop, % top frequency
    twidth, % length in samples
    trelative, % time relative to leftmost element
    fheight, % bandwidth
    nom, % number of merged
    vm, % virtual mode
    [sim,psim,nsim]] % similarity
```

# Chapter 5

## Appendix

### 5.1 Known Issues and Problems

- Minimum screen resolution is 1280 x 1024.
- DSProlog requires full reading and writing access for the folders you choose.
- Some of the procedures run by DSProlog require 100% of CPU time. Close all other applications before starting the IDE.
- Be aware of memory modules of low quality! DSProlog uses low level DSP functions and maps audio files directly into the RAM of the computer.
- In some cases dual channel memory may cause problems. This problem may be solved by deactivating DCM mode in the BIOS.
- The Windows XP file mapping routines may cause problems when a PCM file is larger than the available physical memory.
- FLC model files and FLC files loaded into the FLC editor are sensitive to the resolution of the screen. Conversion may result in unexpected behavior due to the loss of information that occurs when adjusting them to a lower screen resolution.
- Broken data table files and backups (\*.del, \*.val, \*.mb) may appear in the project tree, when DSProlog is closed improperly, e.g. by shutting down electricity.

### 5.2 Disclaimer

This manual is provided without any warranty for correctness or completeness. The software is provided 'as is' without express or implied warranty to the extent

permitted by applicable law. The software may be used only for noncommercial academic purposes.

©2007 Sebastian V. Huebner, Potsdam