

Users guide

for

TOTem Offline DAtabase Monitor

Authors:

Maciej Zalewski (maciej.zalewski.mz@gmail.com)

Przemysław Dadel (pdadel@gmail.com)

1. Basic concepts.....	3
1.1. Interval of validity.....	3
1.2. Detector structure.....	3
1.3. Types of data.....	4
2. Totem Offline Database Monitor.....	5
2.1. Database connection management.....	5
2.2. Measurement type management.....	6
2.3. Run entry management.....	7
2.4. Interval type management.....	8
2.5. Intervals management.....	8
3. Configuration.....	10
3.1. Database connection configuration.....	10
3.2. Mapping configuration.....	10
4. Data display.....	11
4.1. Alignment presenter.....	11
4.2. General measurement presenter.....	13
4.3. Structure element measurement presenter.....	14
4.4. Sensor part measurement presenter.....	15
4.5. Sensor part status presenter.....	16

1. Basic concepts

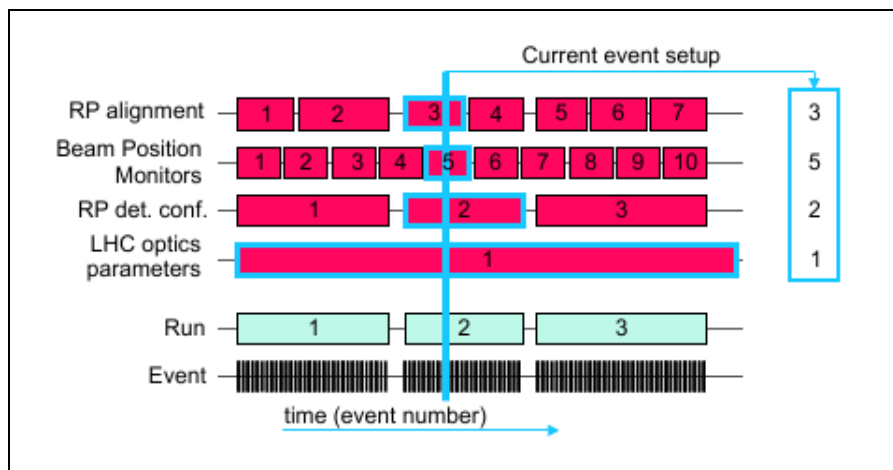
The goal of this documentation is to provide basic information for a User of TOTem Offline Database Monitor (TOTODAM).

The TOTODAM applications' goal is to provide graphical interface for a User to browse the content of the offline database from the point of view of Totem Offline Software. Thus, this chapter will briefly explain basic ideas behind the Totem Offline Software accessing the database.

The application is not intended for changing the data itself – only for previewing the data and modifying their validity Intervals.

1.1. Interval of validity

Totem Offline Software is reconstructing the experiment on the basis of events. In order to reduce the amount data acquisition in the software, there is a mechanism that realizes this functionality. It uses an entity called **Interval of validity**. It represents a range of events when the kept data is valid. The Totem Offline Software retrieves data on the basis of the Interval of validity, therefore the databases' central point is the mechanism that binds any data with proper Intervals of validity.



Picture 1: Interval of validity concept¹

1.2. Detector structure

Totem Offline Database stores the structure of the detector that is generic enough to store any kind of detector structure. They are divided into 2 levels:

- Low level (called Sensor Part) – the lowest level of any detector, defined by parent Structure Element and up to 3 coordinates
- Other levels (called Structure Element) – all the other levels of structure, defined by a type and string identifier, unique within a type.

For instance, in Roman Pot, the structure is kept in such a way (from the highest level):

- Arm: Structure Element (type: RP_ARM, id: 0/1)

- Station: Structure Element (type: RP_STATION, id: 00/02/10/12)
- Unit: Structure Element (type: RP_UNIT, id: 000 - 125)
- Plane: Structure Element (type: RP_SILICON_DETECTOR, id: 0000-1259)
- VFat: Structure Element (type: RP_VFAT, id: 00000-12593)
- Strip: Sensor Part (type: RP_STRIP, parent: RP_VFAT no xxxxx, coordinates (0-255))

Measurements may be combined with any Structure Element or Sensor Part.

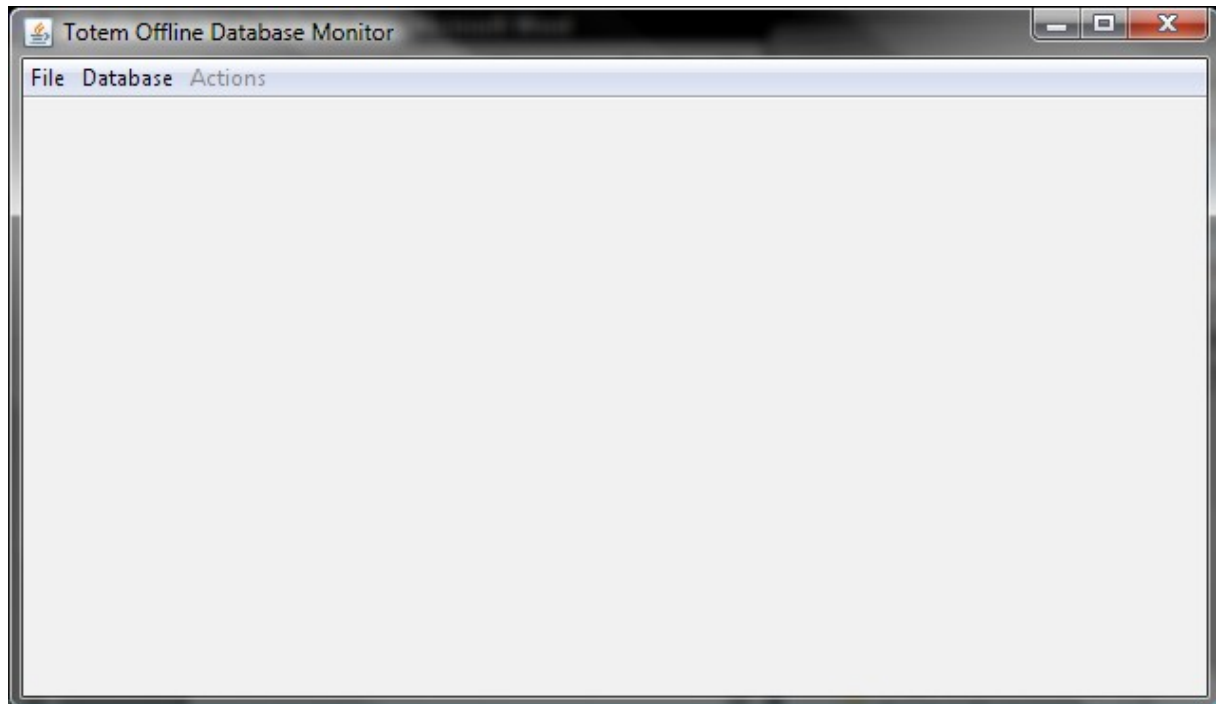
1.3. Types of data

The Totem Offline Database is designed to keep any kind of data that describe the environment during the reconstructed runs. There are multiple possible sources of data. The content of the database may be divided into such groups:

- Group data
This is kind of data for storing numeric data for group of Structure Elements. These data are managed together, they have the same Interval of validity, they are stored and retrieved as a whole.
- General Measurement
This kind of data is designed for storing numeric data for storing global conditions or measurements for detector as a whole.
- Structure Element Measurement
This kind of data is designed for storing numeric data of a chosen type for each of the Structure Elements.
- Sensor Part Measurement
This kind of data is designed for storing numeric data of a chosen type for each of the Sensor Parts.

2. Totem Offline Database Monitor

When the TOTODAM application is started, User can see a general window similar to the one presented below



Picture 2: TOTODAM application - main window (disconnected)

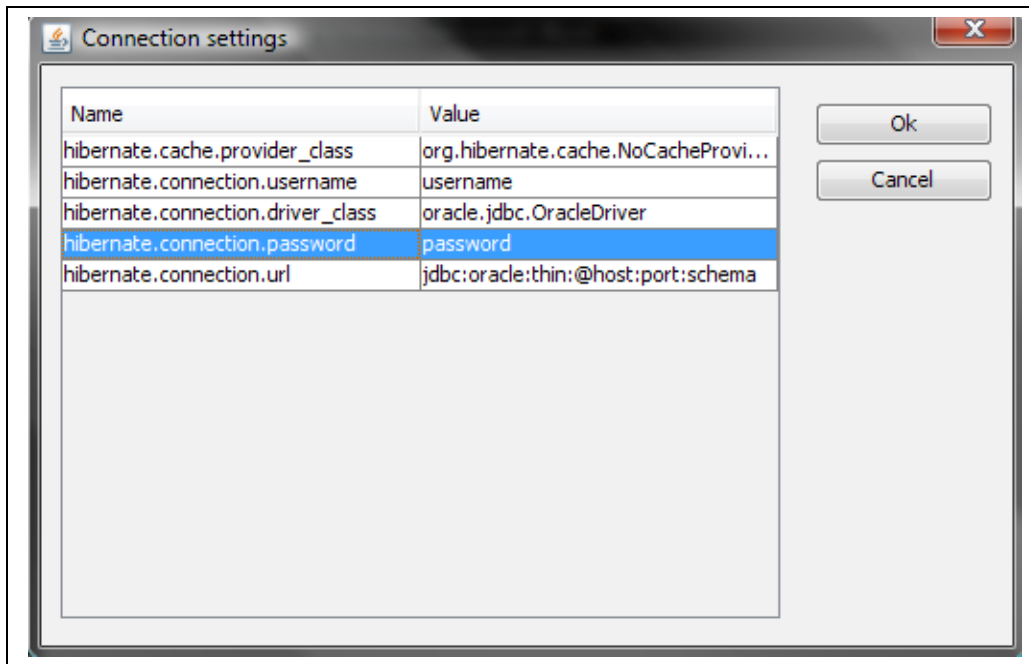
It consists of two parts: a Content Panel and a Menu Bar. When the application is started, the Content Panel remains empty.

The Menu Bar contains three menus:

- 1 **File** menu, which provides the basic application actions
- 2 **Database** menu, which provides functionality of connecting to the database
- 3 **Actions** menu, which provides different actions available for the application

2.1. *Database connection management*

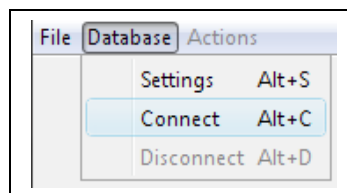
The TOTODAM application connects to the Totem Offline Database in order to present its content and allow the User to interact with it. In order to connect to the database, the User has to provide certain configuration. In order to do this, User must choose **Database\Settings** menu item (or press [Alt]+[S]). This will bring up a dialog box with the database settings.



Picture 3: Database connection settings

The configuration table may be filled with the configuration data provided in the application configuration file. It is described in more detailed way in chapter Database connection configuration.

The settings should be confirmed with **Ok** button. In order to connect to the database, User should then choose **Database/Connect** menu item (or press **[Alt]+[C]**). Application will try to establish connection with the database. If it succeeds, the content of the Content panel will change, otherwise, an error message will appear.



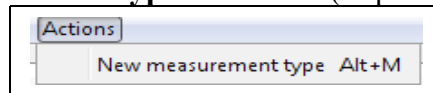
Picture 4: Connect to the database

User can also connect to the database without opening the settings window. However, it will work only if the proper database connection configuration is provided in the configuration file.

After finishing the use of the database, User should disconnect from it by choosing **Database/Disconnect** menu item (or pressing **[Alt]+[D]**). However, terminating application (eg. by pressing **[Alt]+[F4]**) will also close the connection.

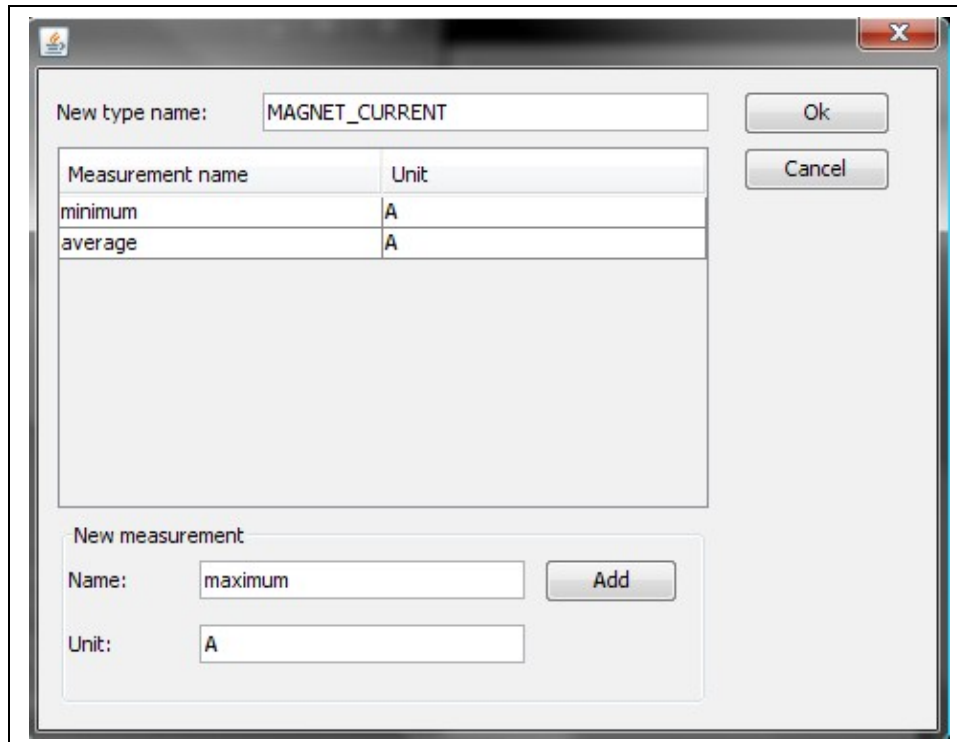
2.2. Measurement type management

After connecting to the database, User can define new measurement types. It is done by choosing **Actions/New measurement type** menu item (or pressing [Alt]+[M]).



Picture 5: New measurement type

The newly inserted measurement will respond to a set of numbers. For each of them, User should define a label and a unit that it will be measured in. Also, User has to provide a unique name for the measurement.



Picture 6: Defining new measurement type

To store the newly created type in the database, it has to be confirmed with **Ok** button.

2.3. Run entry management

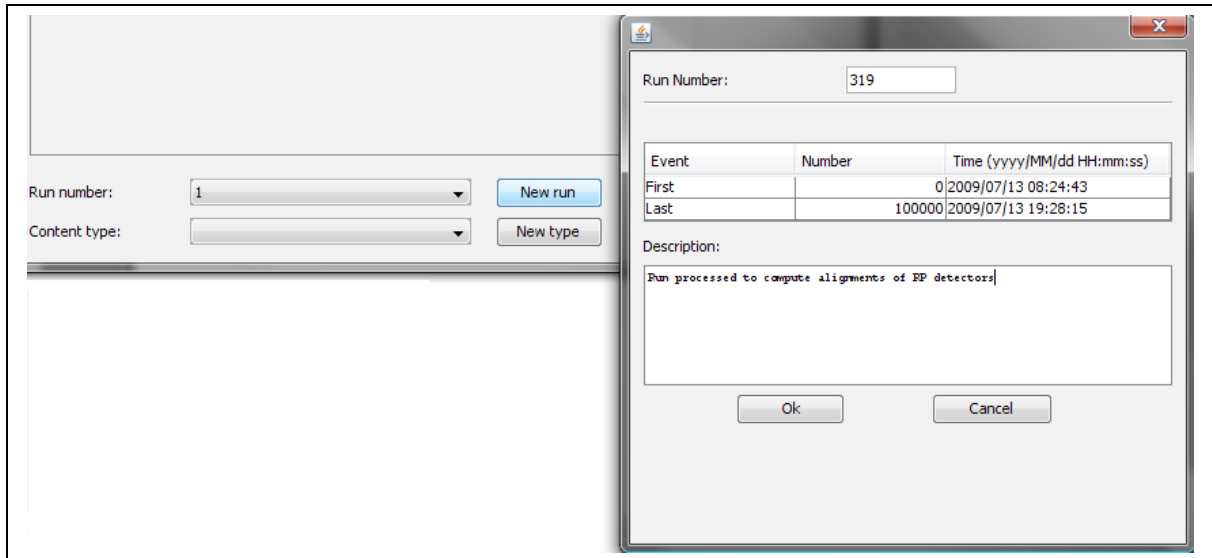
The database stores information about registered runs. List of them is available after connecting to the database, in the lower part of the Content Panel. When User chooses a run, only data corresponding to the chosen run will be displayed.

User can also create a new run by choosing a **New Run** button on the right side of the dialog used to choose a run number.

It is obligatory to provide information about:

- run number
- first event number and time
- last event number and time
- run description

In order for the new run to be stored in the database, it must be confirmed with the **Ok** button.



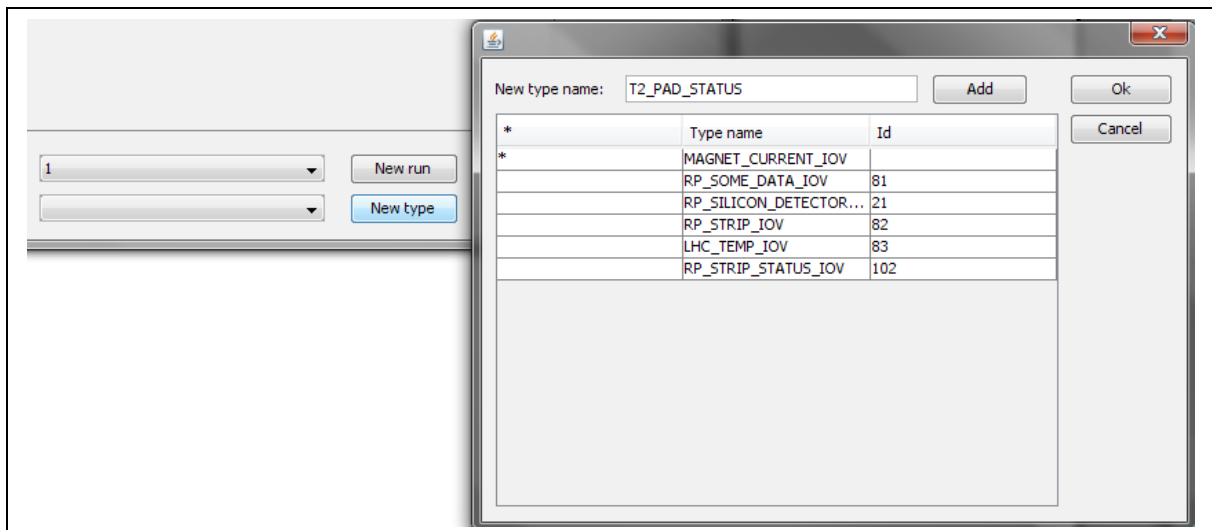
Picture 7: Creating a new run

2.4. Interval type management

Each of the Intervals has its type, which is used to determine which data is combined with the given Interval. After User chooses one of the Interval types, only Intervals of the chosen type will be displayed.

User may also create new Interval types by choosing *New type* button, which resides on the bottom-right side of the Content Panel.

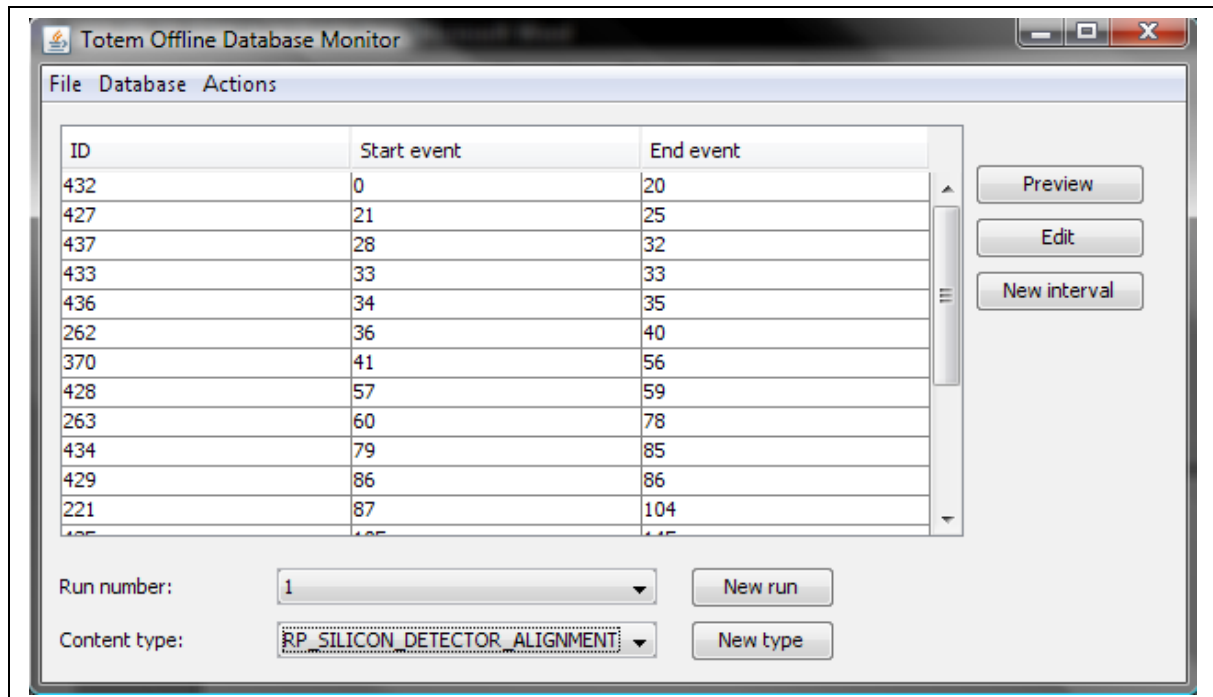
The dialog allows the User to create multiple types at once, their names must be unique. Newly created types will be inserted only after confirming the dialog with *Ok* button.



Picture 8: New Interval type

2.5. Intervals management

The main functionality of the TOTODAM application is to provide the user with ability to see and modify the Intervals of validity for different types of stored measurements or other data.



Picture 9: TOTODAM application - main window (connected)

After connecting to the database, User will see the Content Panel of the TOTODAM application consisting of four important elements:

- **Intervals table** – a table in the middle of the panel, presenting all of the Intervals for the chosen run and of the chosen type. If there is no run or type chosen, the table will remain empty.
- **Run management panel** – placed just below the Intervals table. It consists of:
 - *Run choice box* that allows User to choose a run that will be displayed, and
 - *New run button* that allows User to create new run entries.
- **Interval type management panel** – placed below the Run management panel. It consists of:
 - *Content choice box* that allows User to choose a type of Interval to be displayed, and
 - *New type button* that allows User to create new Interval types
- **Actions panel** – placed on the right side of the Intervals table. Consists of:
 - *Preview button* that allows User to preview the selected Interval
 - *Edit button* that allows User to modify the data combined with the selected Interval
 - *New Interval button* that allows User to create new Intervals of a chosen type

It must be noticed that not all of the buttons on the Actions panel must be available for each type of Interval. It is configured by particular type managers, which are defined in the configuration (see Mapping configuration). If some of the buttons that should be active are disabled, there probably is a problem with mapping configuration.

3. Configuration

The TOTODAM application is configured from an XML file. By default, the application reads the *init.xml* file that resides in its working directory. However, User may specify different configuration file simply by running the application with the filename of different configuration file. However, the other file still has to consist of the same structure of configuration.

3.1. Database connection configuration

The database connection configuration for TOTODAM application resides in an XML file inside the branch **config\database**. It should consist of at least three entries (order does not matter):

- `hibernate.connection.username` – username for the database connection
- `hibernate.connection.password` – password for the given username. It is strongly suggested that this property should be left empty in the configuration file and filled in the settings window (see: Błąd: Nie znaleziono źródła odwołania)
- `hibernate.connection.url` – connection string for the database. Its format is: `<driver_name>@<host>:<port>:<schema>`.

There are also additional settings that may be specified, for the detailed list, see ² or ³

Example configuration of the database connection:

```
<config>
  <database>
    <connection-property name="hibernate.connection.username" value="user" />
    <connection-property name="hibernate.connection.password" value="" />
    <connection-property name="hibernate.connection.url"
value="jdbc:oracle:thin:@oradev10.cern.ch:10520:D10" />
  </database>
</config>
```

3.2. Mapping configuration

In order to make TOTODAM application extendable for new types of data that are possible to present, its architecture was defined in such a way that the information about design elements used to present particular Interval types are stored in configuration file. This configuration should be changed mainly when new Interval type is introduced. For each of Interval types, a presenting design element should be defined. For more extended information on the mapping, refer to Programmers Guide for TOTODAM.

The example below shows sample mapping definition for the application:

```
<config>
  <mapping>
    <mapping-item
name="RP_SILICON_DETECTOR_ALIGNMENT"
```

```

        value="ch.cern.totem.offline.dbmonitor.config.dataconfig.AlignmentDataPresentationConfig"/>
        <mapping-item
        name="RP_SOME_DATA_IOV"

value="ch.cern.totem.offline.dbmonitor.config.dataconfig.StructElementMeasurementDataPresentation
Config"/>
        <mapping-item
        name="RP_STRIP_IOV"

value="ch.cern.totem.offline.dbmonitor.config.dataconfig.SensorPartMeasurementDataPresentationCo
nfig"/>
        <mapping-item
        name="LHC_TEMP_IOV"

value="ch.cern.totem.offline.dbmonitor.config.dataconfig.GeneralMeasurementDataPresentationConf
ig"/>
        <mapping-item
        name="RP_STRIP_STATUS_IOV"

value="ch.cern.totem.offline.dbmonitor.config.dataconfig.SensorPartStatusDataPresentationConfig"/>
    </mapping>
</config>

```

4. Data display

The TOTODAM application has a group of built-in presenters for particular types of data. Those are:

- **Alignment presenter** (class *ch.cern.totem.offline.dbmonitor.config.dataconfig.AlignmentDataPresentationConfig*)
- **General measurement presenter** (class *ch.cern.totem.offline.dbmonitor.config.dataconfig.GeneralMeasurementDataPresentationConfig*)
- **Structure element measurement presenter** (class *ch.cern.totem.offline.dbmonitor.config.dataconfig.StructElementMeasurementDataPresentationConfig*)
- **Sensor part measurement presenter** (class *ch.cern.totem.offline.dbmonitor.config.dataconfig.SensorPartMeasurementDataPresentationConfig*)
- **Sensor part status presenter** (class *ch.cern.totem.offline.dbmonitor.config.dataconfig.SensorPartStatusDataPresentationConfig*)

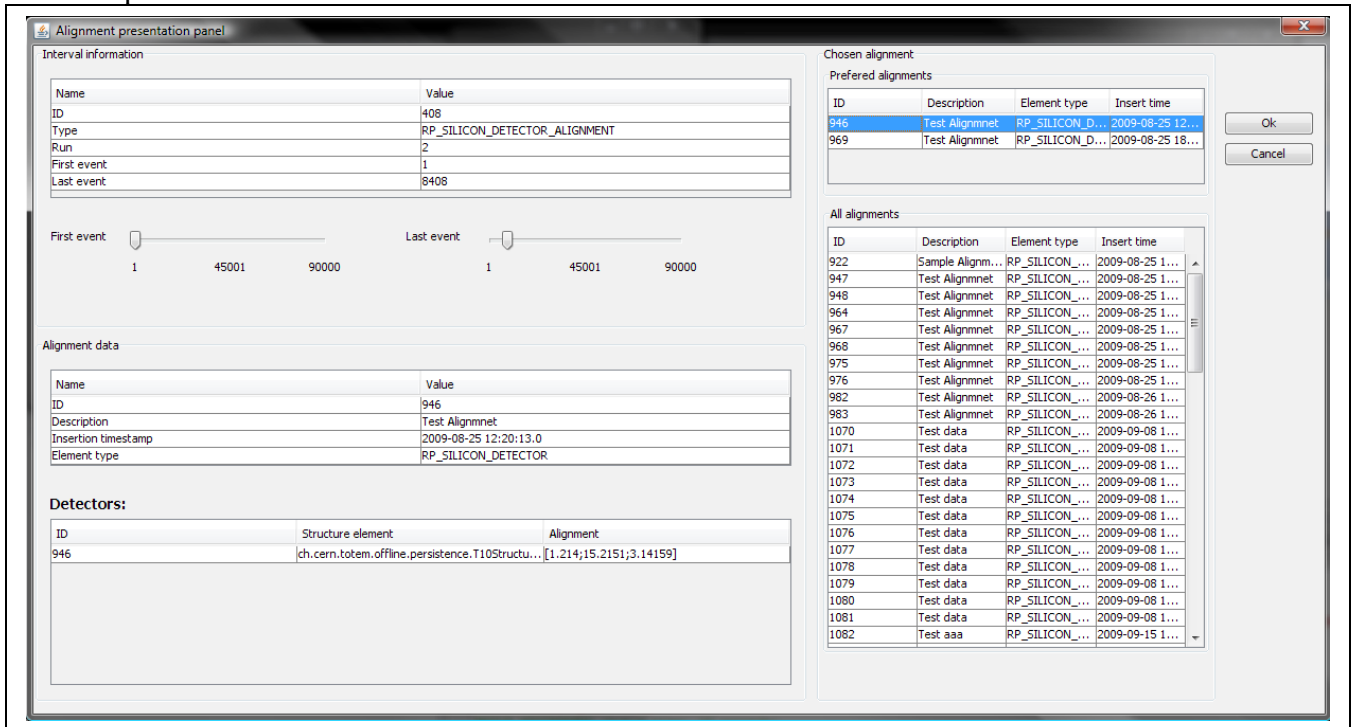
Each of them will be described a bit further in this chapter

4.1. Alignment presenter

Alignment presenter is a presentation item for alignment data.

The alignment data is a group data. It means that although single (so called) **Detector alignment** corresponds to particular Structure element, there is an entity that gathers Detector alignments for all of the structure elements of a given type – as they are all computed

together. This entity is called **Alignment** and it is Alignment that corresponds to each of validity Intervals. The database stores also a record of Alignments that were already used for the particular Interval.



Picture 10: Alignment presenter

The panel consists of three parts:

- **Interval information panel** in the top left corner. It provides basic information about the Interval of validity and also (when in edit mode) enables to change the event range of the Interval
- **Alignment data panel** in the bottom left corner. It provides detailed information about the current default Alignment for the chosen Interval, also in the lower part of the panel there is a list of Detector Alignments
- **Chosen alignment panel** on the right side of the panel. It presents all Alignments that are available in the database. The upper table shows those Alignments that were already used for the chosen Interval. In edit mode, User may choose a different Alignment for the Interval.

Any changes are written to the database only when User confirms them with **Ok** button.

Another important possibility of the Alignment is the possibility to add a new Interval. When editing an existing Interval, User may not set the events so that they overlap. However, when adding a new Interval, User may force the application to accept the overlapping Intervals. In such case, the existing Intervals are changed (and possibly split) in such a way that an Alignment that User specifies for the new Interval is set for the whole range of events that it applies to.

For example, there are three Intervals:

No.	Start	End	Alignment
-----	-------	-----	-----------

1	1	1000	5
2	1001	2000	6
3	2001	3000	6

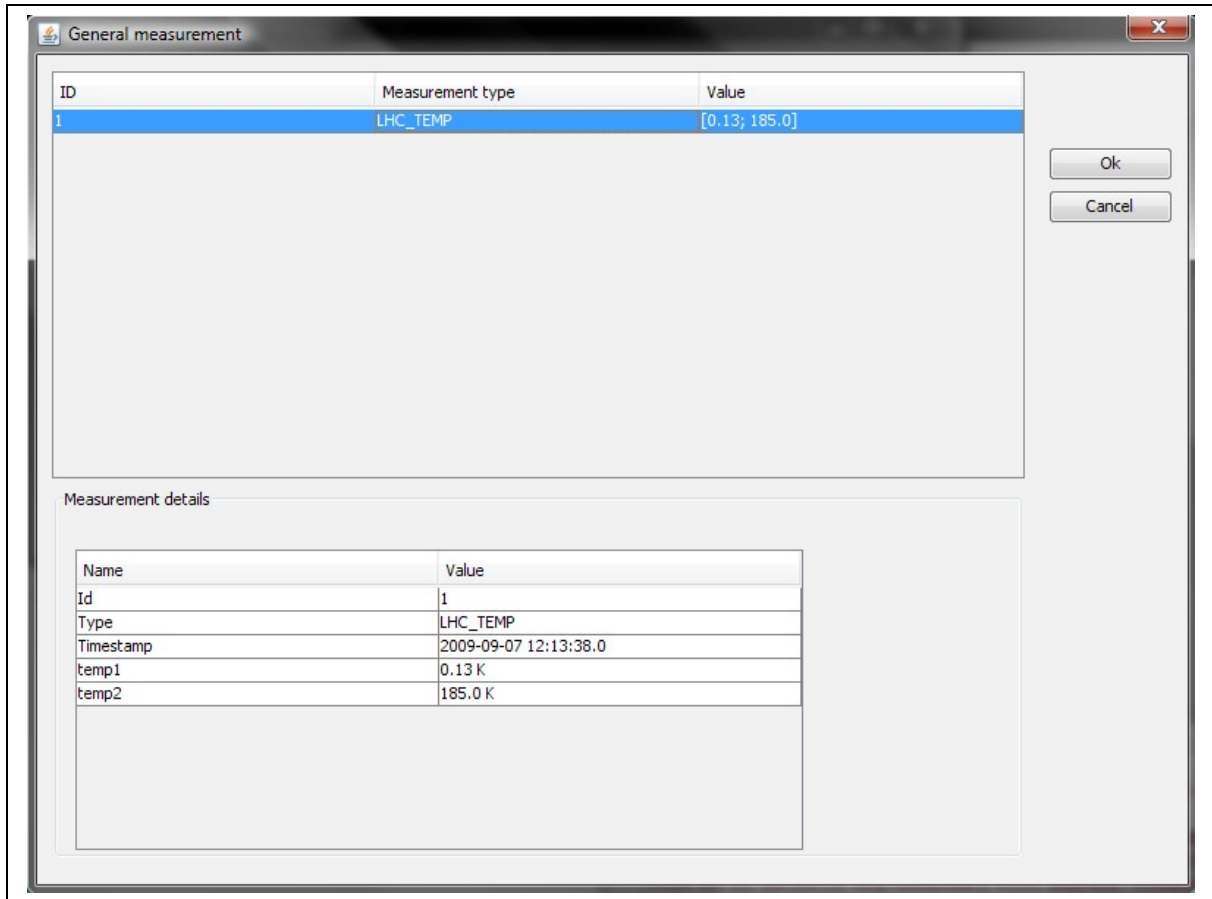
User specifies that a new Interval should be inserted for the events 150 to 2200, with alignment no. 9 set as default, the outcome of the procedure would be like this:

No.	Start	End	Alignment
4*	1	149	5
1	150	1000	9
2	1001	2000	9
3	2001	2200	9
5*	2201	3000	6

It is clearly visible, that for the given Interval, the Alignment has changed to the chosen one. The Intervals marked with '*' are newly added.

4.2. General measurement presenter

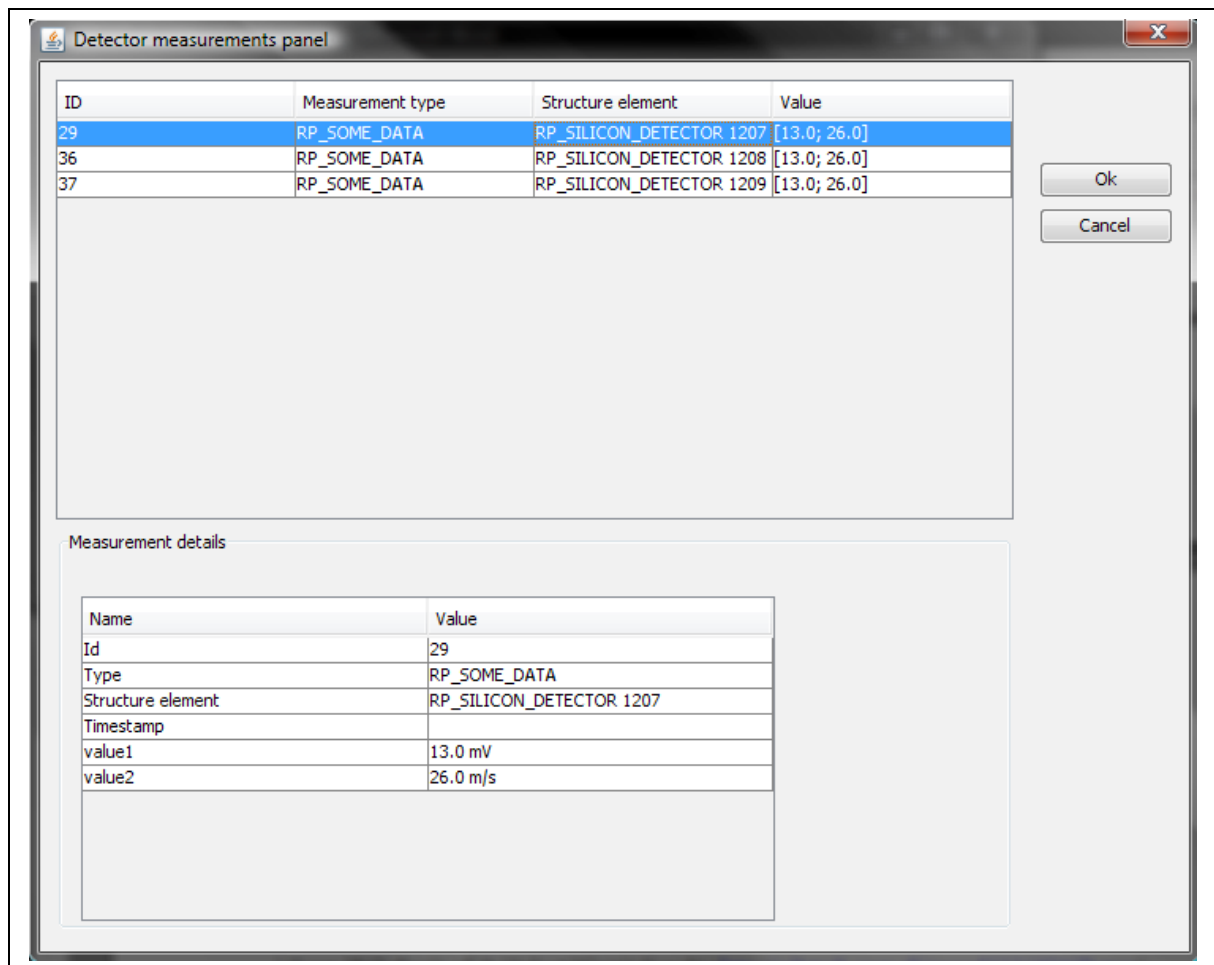
The General measurement presenter enables User to preview (preview only) general measurement stored in the database. As multiple general measurements of a different measurement types may be stored under a single Interval type (eg. Interval type may be BEAM_TEMPERATURE_IOV, and the measurement types may define the part of the beam that the measurement relates to), there is a table that presents a list of measurements under the selected Interval. When User selects any of the measurements, its details are displayed on the Measurement details panel below.



Picture 11: General measurement presenter

4.3. *Structure element measurement presenter*

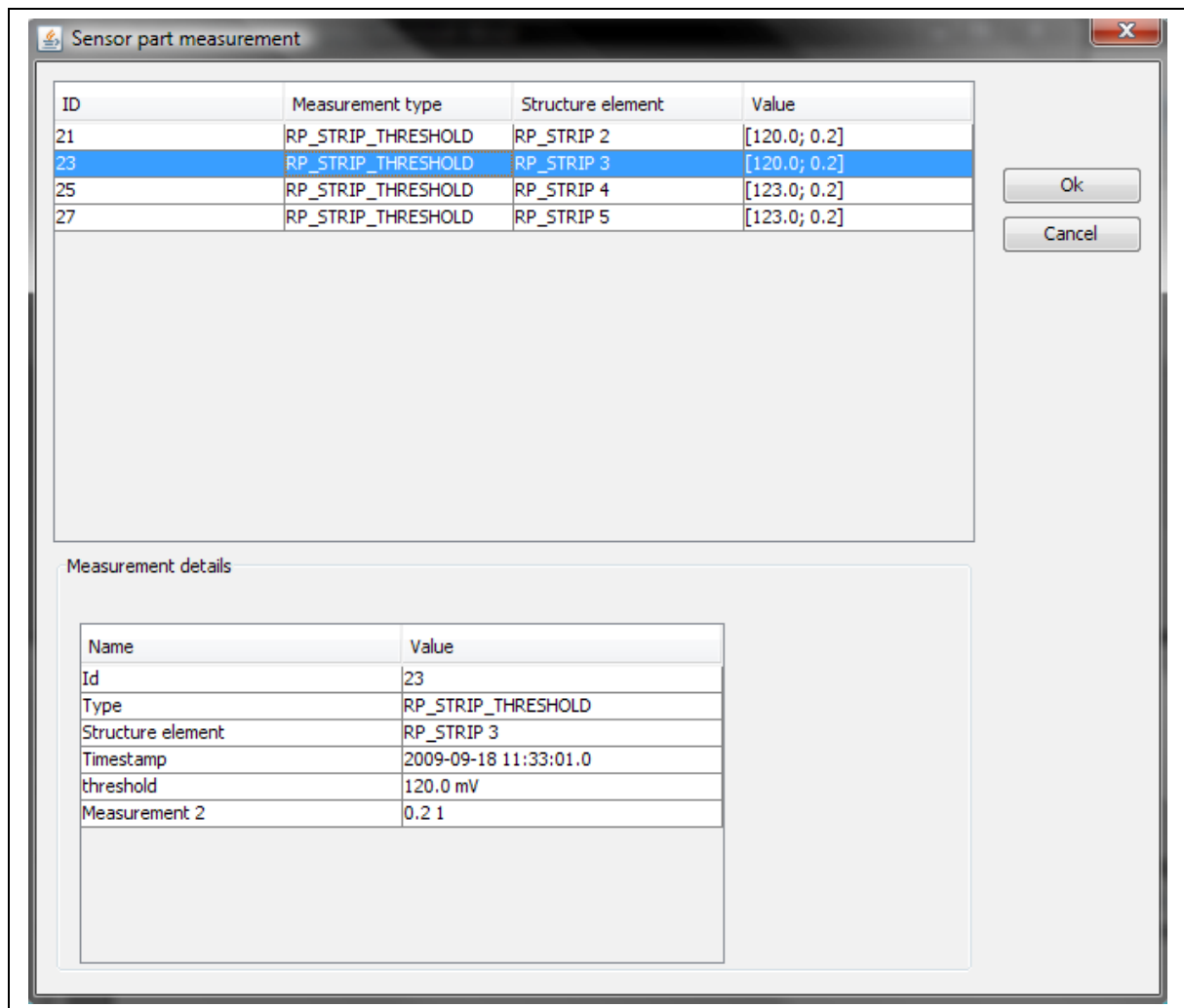
Structure element measurement presenter is build similarly to the General measurement presenter. The main difference is that is also shows information about the structure element that data apply to.



Picture 12: Structure element measurement presenter

4.4. *Sensor part measurement presenter*

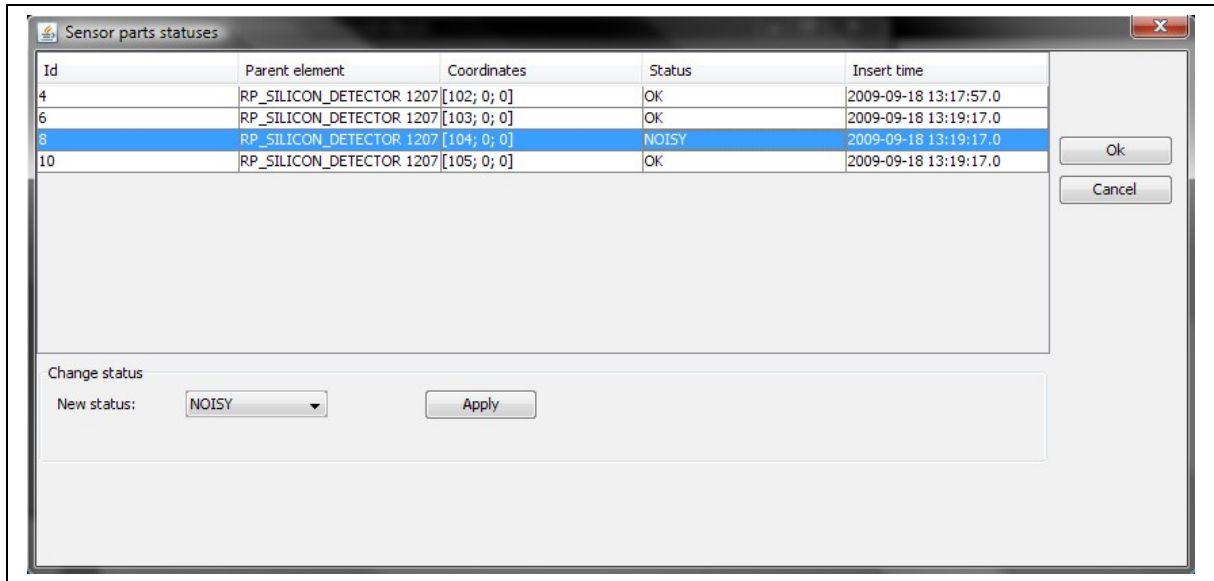
Sensor part measurement presenter provides the functionality of previewing measurements in relation to particular sensor part (eg. threshold of each strip in Roman Pot). The upper table shows the list of measurements valid in the given Interval, and the Measurement details panel below presents detailed information about one of them that is selected.



Picture 13: Sensor part measurement presenter

4.5. *Sensor part status presenter*

Sensor part status presenter provides functionality for previewing and editing the status of all the sensor parts in the given Interval. The upper table lists all the available sensor parts with status for the given Interval, and panel below allows User to change status of selected items.



Picture 14: Sensor part status presenter

¹ From PhD thesis of dr Hubert Niewiadomski (<http://cdsweb.cern.ch/record/1131825?ln=en>)

² <http://www710.univ-lyon1.fr/~ecoquery/files/oracle-jdbc/javadoc/oracle/jdbc/OracleDriver.html>

³ http://www.orindasoft.com/public/Oracle_JDBC_JavaDoc/javadoc920/oracle/jdbc/OracleDriver.html