

GISMO 2.0

(Release version: 20100507200)

Technical Documentation

Version: 1.0

Author: Mauro Nidola

Date: May 7, 2010

Table of Contents

| | |
|--|----|
| 1 Gismo structure..... | 3 |
| 1.1 Introduction..... | 3 |
| 1.2 Requirements..... | 3 |
| 1.3 Back-end (db)..... | 4 |
| 1.3.1 Database schema..... | 4 |
| 1.3.2 Database description..... | 4 |
| 1.3.3 Data exporter..... | 5 |
| 1.4 Front-end (php / javascript)..... | 6 |
| 1.4.1 Third parties libraries..... | 6 |
| 1.4.2 Filesystem structure..... | 6 |
| 1.4.3 Application architecture..... | 7 |
| 2 Developer Information..... | 10 |
| 2.1 Extending Gismo..... | 10 |
| 2.2 Add a new items list..... | 10 |
| 2.2.1 Provide list of items..... | 10 |
| 2.2.2 Left menu class..... | 10 |
| 2.2.3 Gismo javascript class..... | 10 |
| 2.3 Add a new analysis type (new chart)..... | 10 |
| 2.3.1 Read the 'Application architecture' chapter..... | 10 |
| 2.3.2 Collect data..... | 10 |
| 2.3.3 Prepare data..... | 10 |
| 2.3.4 Display chart..... | 10 |

1 Gismo structure

1.1 Introduction

GISMO is a graphical interactive student monitoring and tracking system tool that extracts tracking data from the [Moodle](#) Course Management System, and generates graphical representations that can be explored by course instructors to examine various aspects of distance students.

GISMO uses the students' tracking data from a Moodle, and generates graphical representations that can be explored and manipulated by course instructors to examine social, cognitive, and behavioural aspects of distance students. It enables instructors to identify tendencies in their classes, or to quickly discover individuals who need special attention.

1.2 Requirements

Gismo is primarily developed on the LAMP platform (Linux, Apache, MySQL and PHP), we didn't test it in Windows environments, although it should work also in these environments.

The requirements for Gismo are:

- PHP v. 5.2 (it should work with previous versions of php if you install PECL JSON >=1.2.0, but we didn't test it).
- Mysql 4.0+
- Moodle 1.9+

1.3 Setup

1.3.1 Copy files

First of all, you have to copy GISMO files into your Moodle installation. Please follow the standard procedure to install Moodle plugins:

- copy the gismo folder into the Moodle blocks directory (MOODLE_DOCUMENT_ROOT/blocks)
- logon to Moodle as administrator and visit the Notifications page

1.3.2 Data exporter

Gismo needs to perform some computation on regular basis. The purpose of this computation is to extract data from logs and derive aggregated information. This computation is done daily by a script that, depending on the setting of your Moodle cron script, runs during the night between 02.00 and 04.00 (server time).

When you install the Gismo software the first time, if you don't want to wait the next day to see the result, you can launch this script with your browser by hand at this address:

```
MOODLE_MAIN_URL/blocks/gismo/lib/gismo/server_side/export_data.php
```

1.3.3 Configuration

Gismo doesn't need any particular configuration. Everything runs out of the box.

1.4 Back-end (db)

1.4.1 Database schema

GISMO requires some additional database tables. These tables are automatically added to the Moodle database once the administrator launches the Moodle admin page after installing the files.

The Gismo database schema is reported below.

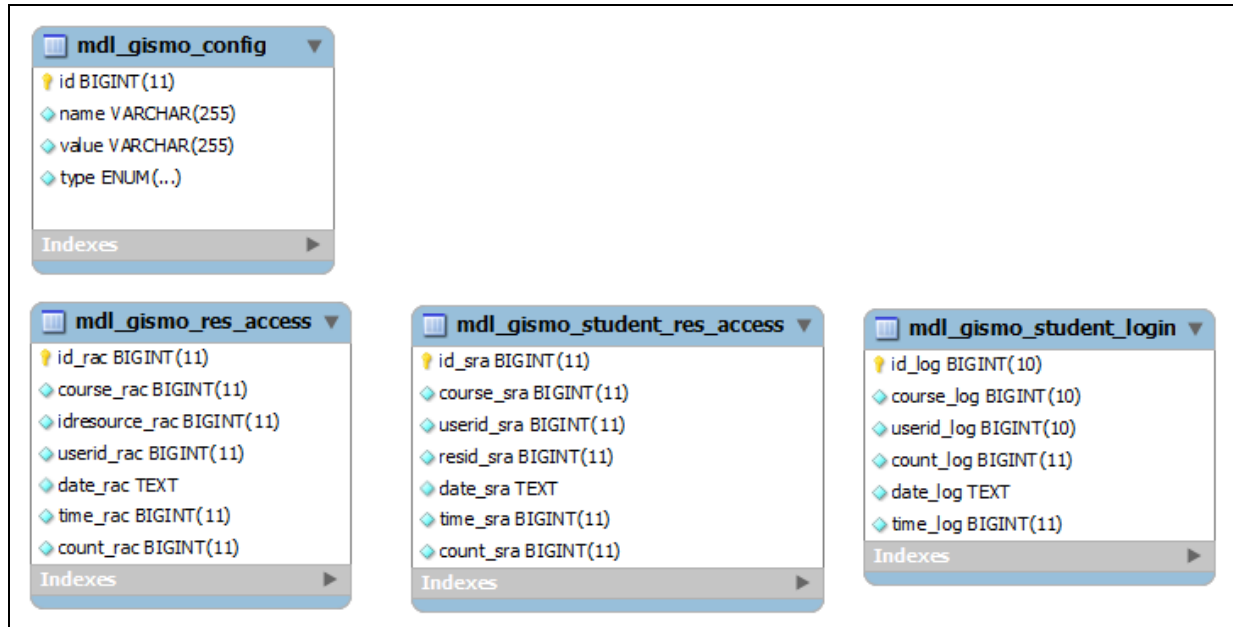


Figure 1: database schema

1.4.2 Database description

Follows the list of Gismo database tables with a short description.

| Table name | Description |
|--------------------------|--|
| gismo_config | Gismo general settings. Values cannot be updated by users. Block instance settings are saved in the moodle <i>block_instance</i> table (field: configdata). |
| gismo_res_access | This table is filled in automatically once a day by the gismo exporter. It contains data extracted from the moodle log table related to accesses on resources. |
| gismo_student_login | This table is filled in automatically once a day by the gismo exporter. It contains data extracted from the moodle log table related to students accesses to the course. |
| gismo_student_res_access | This table is filled in automatically once a day by the gismo exporter. It contains data extracted from the |

| | |
|--|---|
| | moodle log table related to students accesses to resources. |
|--|---|

1.5 Front-end (php / javascript)

1.5.1 Third parties libraries

Follows the list of third parties library used.

| Library name | 0 | Web site url |
|--------------|-------|---|
| jQuery | 1.4.1 | http://jquery.com/ |
| JQuery UI | 1.7.2 | http://jqueryui.com/ |
| jQPlot | 0.9.7 | http://www.jqplot.com/ |

1.5.2 Filesystem structure

The content of the *gismo* folder (main folder) is described in the table below (most interesting files only).

| Name | Type | Description |
|-----------------------|------|--|
| db | dir | Files for the setup (db structure, upgrade scripts) |
| images | dir | Images |
| lang | dir | Applications translations (at this time english only and server side only) |
| lib | dir | Gismo libraries (gismo / third parties and then client side / server side) |
| style | dir | CSS files |
| template | dir | Pages templates |
| ajax.php | file | This is the PHP script called through AJAX to extract data for charts. |
| ajax_config.php | file | This is the PHP script called through AJAX to save application settings. |
| block_lor_adapter.php | file | Main block file. Please visit the Moodle blocks documentation page for details: http://docs.moodle.org/en/Development:Blocks |
| common.php | file | Common things to perform at the begin of almost each php file (includes server side libraries, checks authorization, ...). |
| main.php | file | This page displays list of modules that are going to be exported and keep trace of the export progress (progress |

bar / modules status).

1.5.3 Application architecture

Gismo architecture is quite simple and can be divided in two sections:

a) User interface

Follows a screenshot of the main page

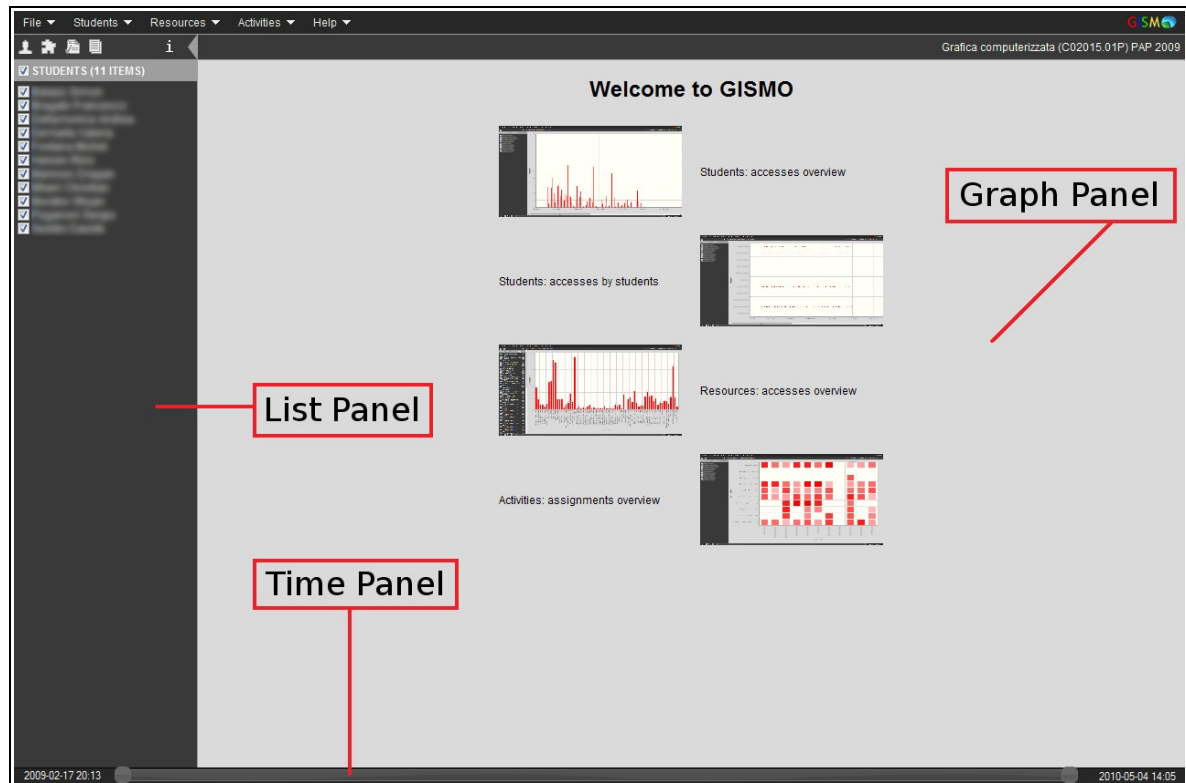


Figure 2: gismo main page

The user interface is managed through four javascript classes: top_menu, left_menu, time_line, gismo.

- top_menu

This class manage the top menu.

- left_menu

This class manage the left menu, it deals with the following tasks:

- build items lists and bind events to items
- show/hide lists
- manage the connection between lists and analysis type

- time_line

This class manage the time line on the placed on the bottom of the page, it deals with the following tasks:

- build the timeline (date picker components, slider)
- update the current analysis chart when start date or end date are updated

- gismo

This class manage analysis types and it is connected to the previous three classes. The connection is achieved through the composite pattern, this means that the gismo class has one instance of each previous class stored in a class field. The previous three classes have a field with the link to the gismo instance.

The gismo class deals with the following tasks:

- charts management (retrieve data from server, prepare data, build the chart)
- manage chart width and height (resize chart)
- manage application options

Class files are stored in the *gismo/lib/gismo/client_side* directory.

- b) Analysis (data retrieval, data management and data representation)

Analysis are managed through the PHP script `ajax.php` and the gismo class.

- gismo class

To perform one of the gismo supported analysis the following methods are involved:

- `analyse(type, options)`

This is the main method, the one to call to perform a new analysis. This method performs the following steps:

- ask the server for data related to the analysis type
- update gismo instance information
- update the left menu (show / hide items lists)
- call the **`create_chart()`** method

- `create_chart()`

This method deals with the creation of the chart. It performs the following steps:

- call the **`prepare_data()`** method
- build the chart according to the analysis type using the jQPlot library

- `prepare_data()`

This method prepare the previously extracted data. This method performs the following steps:

- define the **`prepared_data`** variable to store information
- organize data according to the analysis type and the selected items (users, resources, quizzes, assignments)
- save the prepared data in a gismo class field

- `ajax.php` script

This script is called by the gismo **`analyse(type, options)`** method through an AJAX call. It take the analysis type and options as input, query the database (gismo tables / moodle tables) and returns the data to base the chart on.

The script is located in the gismo main directory.

2 Developer Information

2.1 Extending Gismo

Gismo has been designed for accommodating extensions by third parties. This includes adding new items lists and new charts.

2.2 Add a new items list

To add a new items list please act as follows.

2.2.1 Provide list of items

Open the files *gismo/lib/gismo/server_side/FetchStaticDataMoodle.php* and *gismo/main.php*.

- *FetchStaticDataMoodle.php* defines a class that provide information taken directly from the server. Please add fields and methods useful to extract the list of items from the server.
- Look at lines [40;72] in the *main.php* file (javascript code). As you can see lists are passed to the gismo main class using an associative array, its name is *static_data*. Please add a new entry and provide the list of items through the *FetchStaticDataMoodle* (the instance name is *gismo_static_data*).

2.2.2 Left menu class

Open the file *gismo/lib/gismo/client_side/left_menu.js* and performs as follows:

- Update the *lists* field (please provide an icon and put it to the *gismo/images folder*).
- Update the *set_menu* method (add entries for the variables *all* and *enabled*)

2.2.3 Gismo javascript class

- Open the file *gismo/lib/gismo/client_side/gismo.js* and update the class fields *menu*, *menu_default*, and *menu_details*. Add the new list to the analysis that require it.

2.3 Add a new analysis type (new chart)

To add a new analysis type please act as follows.

- Read chapter *Application architecture* again
- Update the ajax script so that it is able to return data for the new analysis type

2.3.1 Read the 'Application architecture' chapter

You will find useful information in this chapter.

2.3.2 Collect data

Update the *ajax.php* script so that it is able to return data for the new analysis type

2.3.3 Prepare data

Update the *prepare_data()* defined in the gismo class. Think about items that can be selected / unselected in the left menu and be sure that the method is able to include / exclude correspondent data.

2.3.4 Display chart

Update the *create_chart()* method defined in the gismo class. Once you decided the representation type you want to use then check whether it is possible to reuse the code of one of the other already supported. If this is not the case then add a new entry in the switch construct.