A Moodle-based **Learning Analytics Dashboard** for Promoting **Self-Regulating Learning** in Professional Learning Communities

FLAVIO MANGANELLO & **ELEFTHERIOS KOSMAS**

Objectives

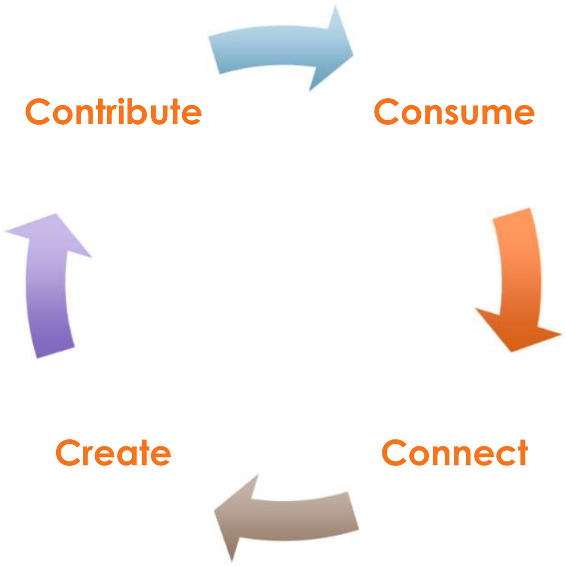
for each **course**:

1. the teacher
   - a) **provides** (mandatory or optional) **tasks** that students have to accomplish
   - b) **rewards** students with (variable) **points** for their achievements
   - c) **has** access to a **visual** aggregate **overview** of the students' accomplishments

2. each student
   - a) **self-regulates** her/his learning procedure by self-monitoring her/his **behaviors**
   - b) **gets rewards** for her/his achievements while interacting within the course
   - c) **has** access to a (personal) **visual overview** of her/his accomplishments
   - d) **has** access to a **visual** aggregate **overview** of the students' accomplishments

Objectives

Some details…



4Cs Framework:

- students **practice four** learning **behaviors**
- **representative** of self-regulated learning in professional learning online communities

Customization:

- the teacher is able to
  a) **assign** each task to **any** of the 4Cs framework's **behaviors**
  b) **designate** each task as either **mandatory** or **optional**
  c) **provide variable** points for each task (accomplishment)

Objectives

Some more details…

Type of tasks:

- the teacher rewards the students for accomplishing tasks of specific **type**

Several visual overviews:

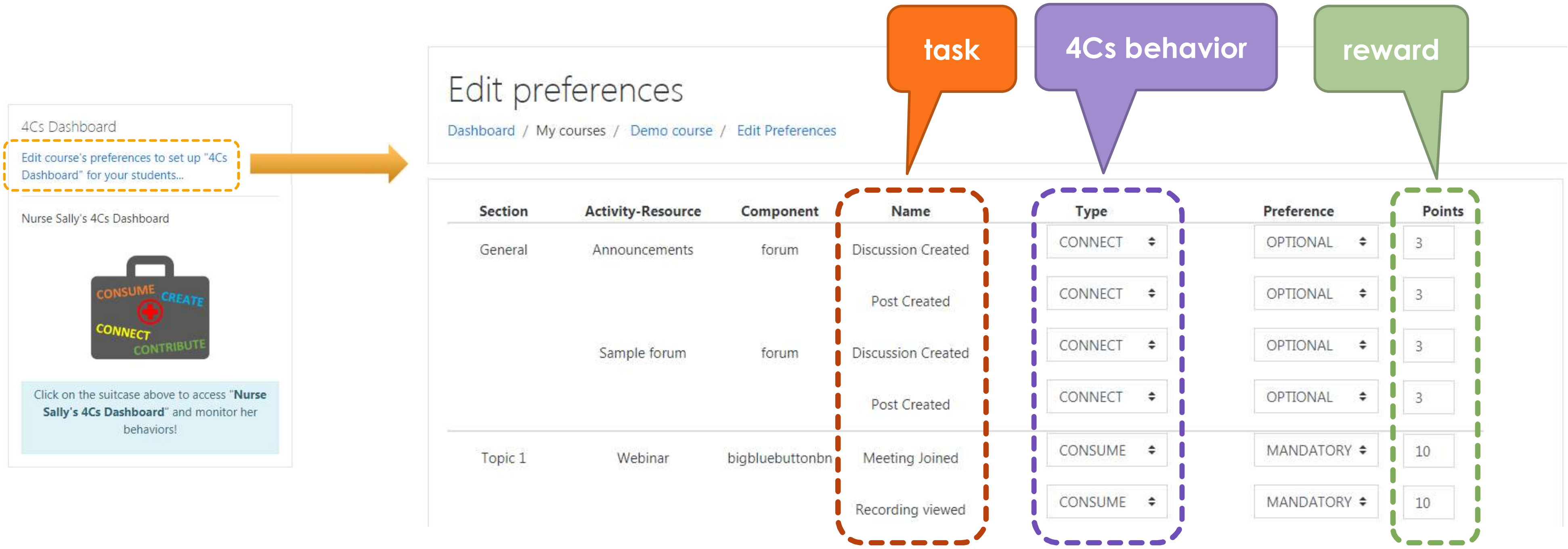
- for both the teacher and the students
  - a) 4Cs framework's **behaviors** and
  - b) type of tasks (for each 4Cs framework's **behaviors**)
- additionally, for the students
  - c) mandatory and optional tasks

Task types
Discussion created in Forum
Post created in Forum
File viewed
Folder viewed
Page viewed
URL viewed
Meeting joined
Recording viewed
Record created
Assignment submission
Quiz submitted
Journal entry created
Journal entry updated

Result

Teacher's side - Customization

- an example of teacher's customization options



4Cs Dashboard

Edit course's preferences to set up "4Cs Dashboard" for your students...

Nurse Sally's 4Cs Dashboard

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

Edit preferences

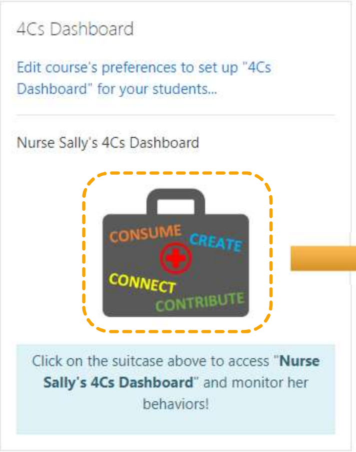
Dashboard / My courses / Demo course / Edit Preferences

Section	Activity-Resource	Component	Name	Type	Preference	Points
General	Announcements	forum	Discussion Created	CONNECT	OPTIONAL	3
			Post Created	CONNECT	OPTIONAL	3
	Sample forum	forum	Discussion Created	CONNECT	OPTIONAL	3
			Post Created	CONNECT	OPTIONAL	3
Topic 1	Webinar	bigbluebuttonbn	Meeting Joined	CONSUME	MANDATORY	10
			Recording viewed	CONSUME	MANDATORY	10

Result

Teacher's side - Aggregate reports

- an example of teacher's aggregate visual overviews

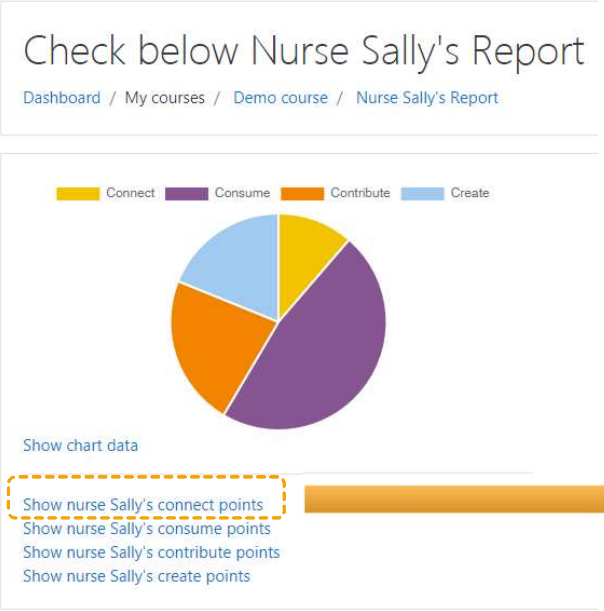


4Cs Dashboard

Edit course's preferences to set up "4Cs Dashboard" for your students...

Nurse Sally's 4Cs Dashboard

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

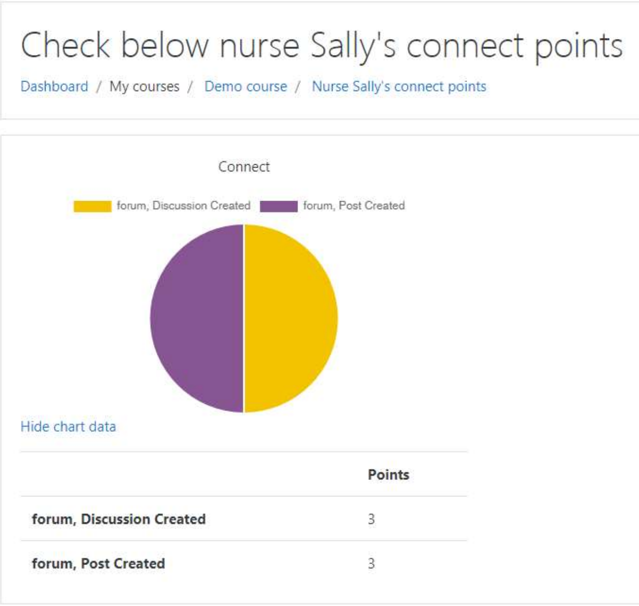


Check below Nurse Sally's Report

Dashboard / My courses / Demo course / Nurse Sally's Report

Show chart data

Show nurse Sally's connect points
Show nurse Sally's consume points
Show nurse Sally's contribute points
Show nurse Sally's create points



Check below nurse Sally's connect points

Dashboard / My courses / Demo course / Nurse Sally's connect points

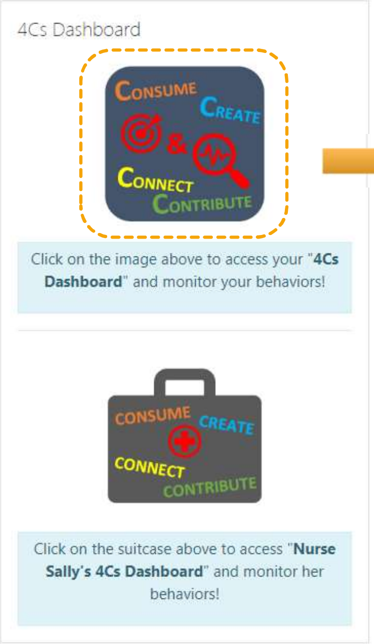
Hide chart data

	Points
forum, Discussion Created	3
forum, Post Created	3

Result

User's side - Personal reports

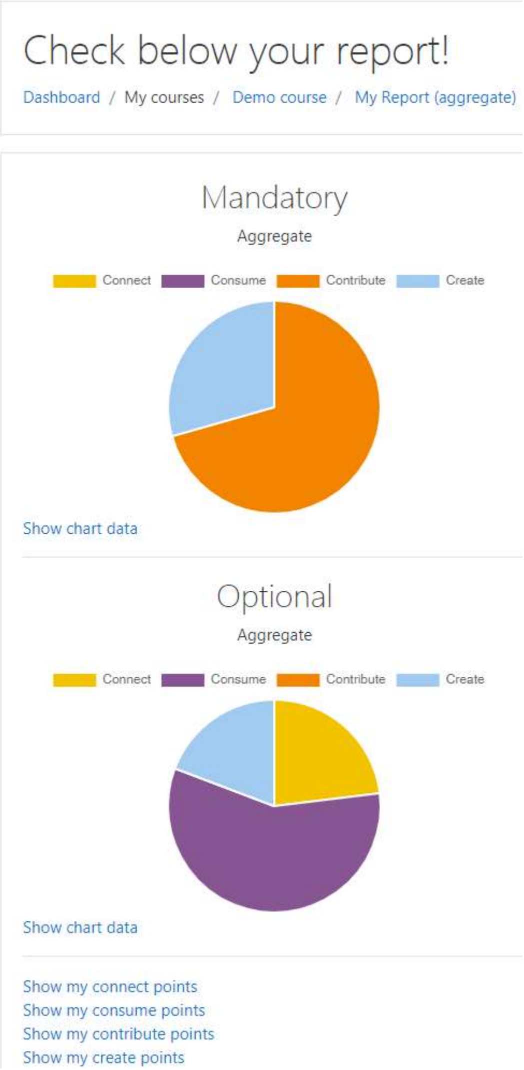
- an example of user's personal visual overviews



4Cs Dashboard

Click on the image above to access your "**4Cs Dashboard**" and monitor your behaviors!

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!



Check below your report!

Dashboard / My courses / Demo course / My Report (aggregate)

Mandatory

Aggregate

Show chart data

Optional

Aggregate

Show chart data

Show my connect points
Show my consume points
Show my contribute points
Show my create points

Tools

- **moodle** - Open-source learning platform

- **HTML** - Hypertext Markup Language
  - *"the standard markup language for **documents** designed to be **displayed** in a **web browser**"* (https://en.wikipedia.org/wiki/HTML)

- **PHP** - Hypertext Preprocessor
  - *"a general-purpose programming language originally designed for **web development**"* (https://en.wikipedia.org/wiki/PHP)
  - *"a server scripting language, and a powerful tool for making **dynamic** and **interactive** Web pages"* (https://www.w3schools.com/php/)

- **MySql** - Open-source relational database management system

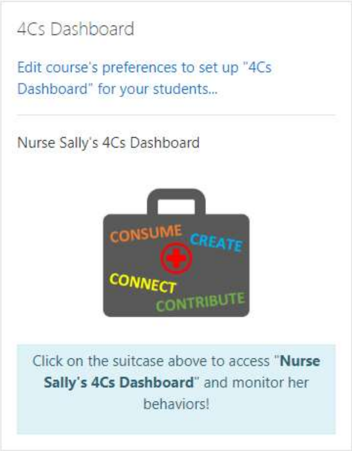
4Cs Dashboard (moodle plugin)

Design & Implementation

Plugin Type

- moodle allows programmers to extend its functionality by implementing **plugins**
  - plugins can be installed into moodle
  - various **types** of plugins are supported (https://docs.moodle.org/dev/Plugin_types)

- we chose to design and implement our functionality (**4Cs Dashboard**) using a **plugin** of type **block**
  - moodle provides a well-written introductory guide for this type of plugin (https://docs.moodle.org/dev/Blocks)
  - as well as a more advanced one (https://docs.moodle.org/dev/Blocks_Advanced)



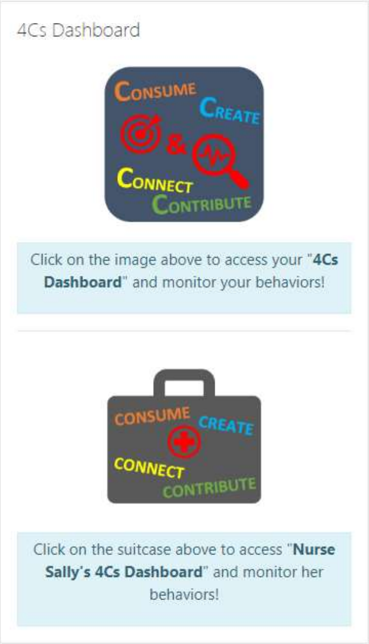
4Cs Dashboard

Edit course's preferences to set up "4Cs Dashboard" for your students...

Nurse Sally's 4Cs Dashboard

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

teacher's view of the 4Cs Dashboard block plugin



4Cs Dashboard

Click on the image above to access your "**4Cs Dashboard**" and monitor your behaviors!

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

student's view of the 4Cs Dashboard block plugin

4Cs Dashboard

Block plugin - How it works

- can be installed **per course**
- **automatically** discovers
  - the **sections** (i.e. topics) of the course and
  - the **activities** (e.g. forum, file, quiz)of each section

  by accessing the appropriate moodle's database tables
  - using functions provided by moodle
- **automatically** tracks **events** performed by students on the activities and accounts them the corresponding rewards
  - each reward is **unique**: if the same **event** happens multiple times → the corresponding reward is accounted only **once**
- **maintains** a database with the required information
  - e.g. points of rewards per student

4Cs Dashboard

Block plugin - Implementation - Requirements for installation

files **required** in order to be able to install a plugin of type block:

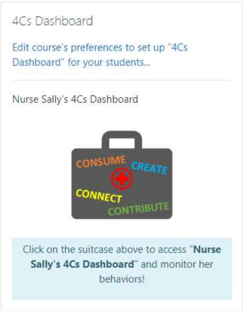
1. the **main file** is `block_4cs_dashboard`.php
   - it includes (and has the same name) with the **main class** of the plugin

     ```
     class block_4cs_dashboard extends block_base { … }
     ```

   - includes the function used to show the dynamic (using PHP) main HTML page of the plugin

     ```
     get_content()
     ```

     - it differentiates the main page between the teacher and the student



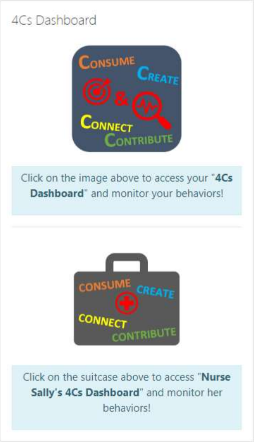
4Cs Dashboard

Edit course's preferences to set up "4Cs Dashboard" for your students...

Nurse Sally's 4Cs Dashboard

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

teacher's main page



4Cs Dashboard

Click on the image above to access your "**4Cs Dashboard**" and monitor your behaviors!

Click on the suitcase above to access "**Nurse Sally's 4Cs Dashboard**" and monitor her behaviors!

student's main page

4Cs Dashboard

Block plugin - Implementation - Requirements for installation (II)

files **required** in order to be able to install a plugin of type block:

2. `access`.php
   - creates the new **capabilities** (features) of the block
   - each capability describes the **permissions** given to each user category
     - e.g. `block/4cs_dashboard/viewpages` → allows any type of user to view pages, except guests
     - e.g. `block/4cs_dashboard/addinstance` → allows teachers to install **4Cs Dashboard per course**
   - some capabilities of the **4Cs Dashboard**:
     a) `block/4cs_dashboard/earnpoints` → allows only students to earn points
     b) `block/4cs_dashboard/editprefs` → allows only teachers to edit the **4Cs Dashboard's** preferences (of some course)
3. `version`.php
   - holds version information for the plugin

important: **when updating the code of a plugin, its version number should be incremented → so that moodle detects this modification and proceed by updating plugin's components**

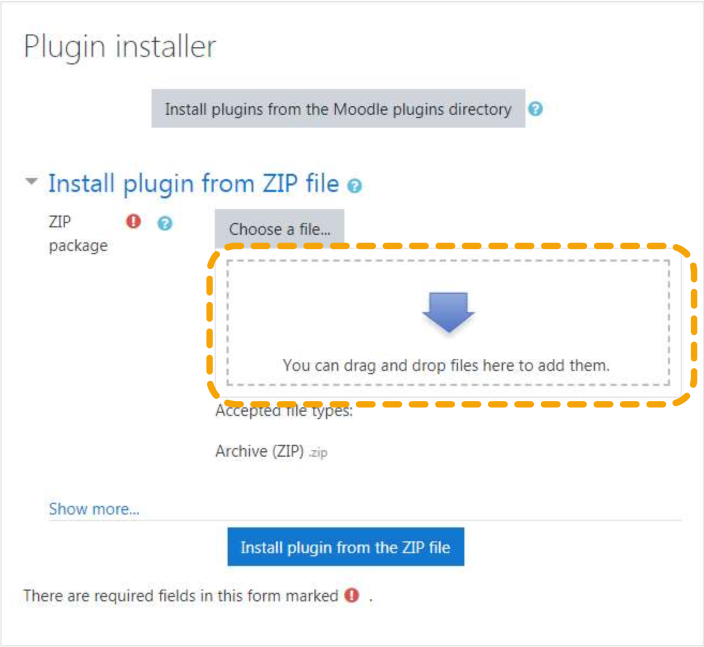
4Cs Dashboard

Block plugin - Installation

- assume that we have already prepared all the files required by the **plugin**
- then, the **plugin** can be **simply** installed to moodle
  - by any user with administration privileges

following the steps below:

1. **create** a **zip file** containing all the produced files, e.g. **4cs_dashboard.zip**
2. **visit** the plugin installer moodle page
   - located in the **administration block**: "**Site administration | Plugins | Install plugins**" (**/moodle/admin/tool/installaddon/index.php**)
3. **install** plugin by simply **drag-and-drop** the **4cs_dashboard.zip** file in the webpage



Plugin installer

Install plugins from the Moodle plugins directory
Install plugin from ZIP file
ZIP package
Choose a file...
Show more...
Install plugin from the ZIP file

You can drag and drop files here to add them.

Accepted file types:

Archive (ZIP) .zip

There are required fields in this form marked .

4Cs Dashboard

Block plugin - Implementation - Automatic retrieval of sections & activities

- moodle provides functions that can dynamically retrieve the sections and/or activities of any course
- the code in **php** looks like the following:

```
$format = course_get_format($PAGE->course);
$sections = $format->get_sections();

$course = $format->get_course();
$activities = get_array_of_activities($course->id);
```

retrieval of course's sections

retrieval of course's activities

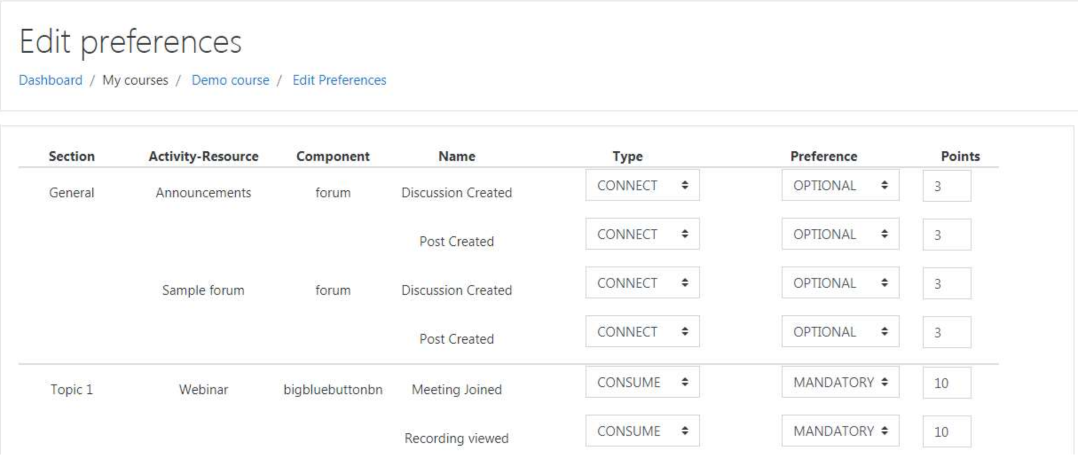
instead of "activity", moodle uses the name "activity module"

- then, by adding some more **php** and **HTML** code …

4Cs Dashboard

Block plugin - Implementation - Edit preferences form

- then, by adding some more **php** and **HTML** code → the edit preferences **form** is implemented as a new webpage
- we **guarantee** that **only** the teacher accesses it by checking the user's capabilities
  - should have the `block/4cs_dashboard/editprefs` capability
- moodle provides **specific** Application Programming Interfaces (**API**s) in order to
  - implement a web **form** (https://docs.moodle.org/dev/Form_API) (https://docs.moodle.org/dev/Blocks_Advanced#Adding_forms)
  - **determine** what the current user is allowed to do (https://docs.moodle.org/dev/Access_API)



Edit preferences

Dashboard / My courses / Demo course / Edit Preferences

Section	Activity-Resource	Component	Name	Type	Preference	Points
General	Announcements	forum	Discussion Created	CONNECT ⬍	OPTIONAL ⬍	3
			Post Created	CONNECT ⬍	OPTIONAL ⬍	3
	Sample forum	forum	Discussion Created	CONNECT ⬍	OPTIONAL ⬍	3
			Post Created	CONNECT ⬍	OPTIONAL ⬍	3
Topic 1	Webinar	bigbluebuttonbn	Meeting Joined	CONSUME ⬍	MANDATORY ⬍	10
			Recording viewed	CONSUME ⬍	MANDATORY ⬍	10

Moodle events

- moodle supports **events** (https://docs.moodle.org/dev/Event_2)
  - pieces of information describing something that happened in Moodle

- when an **event** occurs → the Moodle's events system disseminates this event information to **observers** registered for this **event**
  - **event observers** just receive this information
    - they can not modify event data or interrupt the dispatching of events
    - ☞ it is a **one way** communication channel

- **events** system: a communication backbone throughout the Moodle system

4Cs Dashboard

Block plugin - Implementation - Event tracking & rewards

4Cs Dashboard plugin **automatically**

a) **tracks events** performed by students on the activities

- all the **events** handled by the plugin are included into file: **`events`**`.php`, which includes:
  - the name of the **event**
  - the name of the **observer**, i.e. the corresponding function's name

a) **accounts** the corresponding rewards to students

- this is performed by the **observers**
- using the information received during **event** dispatching
  - **event id**, **activity id**, **user id**, and **course id**
  - other **event**-specific information: e.g. **recording id**
- the code of the **observers** is defined into file: **`observer`**`.php`

4Cs Dashboard

Block plugin - Database - Design

- a set of database tables **maintains** the required information

1. **events** table
   - maintains information about the **tasks** (or **events**) tracked by **4Cs Dashboard** plugin
   - it is initialized upon plugin installation and never changes

id	event name	component	4Cs Dashboard behavior	points
1	Discussion Created	Forum	Connect	3
…	…	…	…	…
3	Course Module Viewed	resource	Consume	5
…	…	…	…	…
9	Record Created	data	Contribute	12
10	Assignment Submitted	assign	Create	10
…	…	…	…	…

this is task "file viewed"

default values of behavior and points

4Cs Dashboard

Block plugin - Database - Design (II)

2. preferences table
   - maintains the preferences of the teacher of each course
   - it is initialized with **default** values
   - each row can be **updated** by the teacher, using edit preferences **form**

Id	course_ id	module_id	event_id	type	4Cs Dashboard behavior	points
1	1	1	3	mandatory	Connect	6
2	1	2	9	optional	Contribute	12
…	…	…	…	…	…	…

may be different from their default values

these ids refer to tables defined by moodle

the corresponding id from the events table

3	Course Module Viewed	data	Consume	5

default values of **event** with id **3**

4Cs Dashboard

Block plugin - Database - Design (II)

3. **points** table
   - maintains the students' rewards
   - new rows can be added by the **event observers**

id	user_ id	course_id	event_id	points	value
1	1	1	3	6	####
2	1	2	9	12	$$$$
…	…	…	…	…	…

these ids refer to tables defined by moodle

the corresponding id from the events table

- **used to uniquely identify each event**
- **so that → reward is accounted only once**

the value:

- **depends on the type of the event**
- **is derived from the event's information dispatched to the observer**

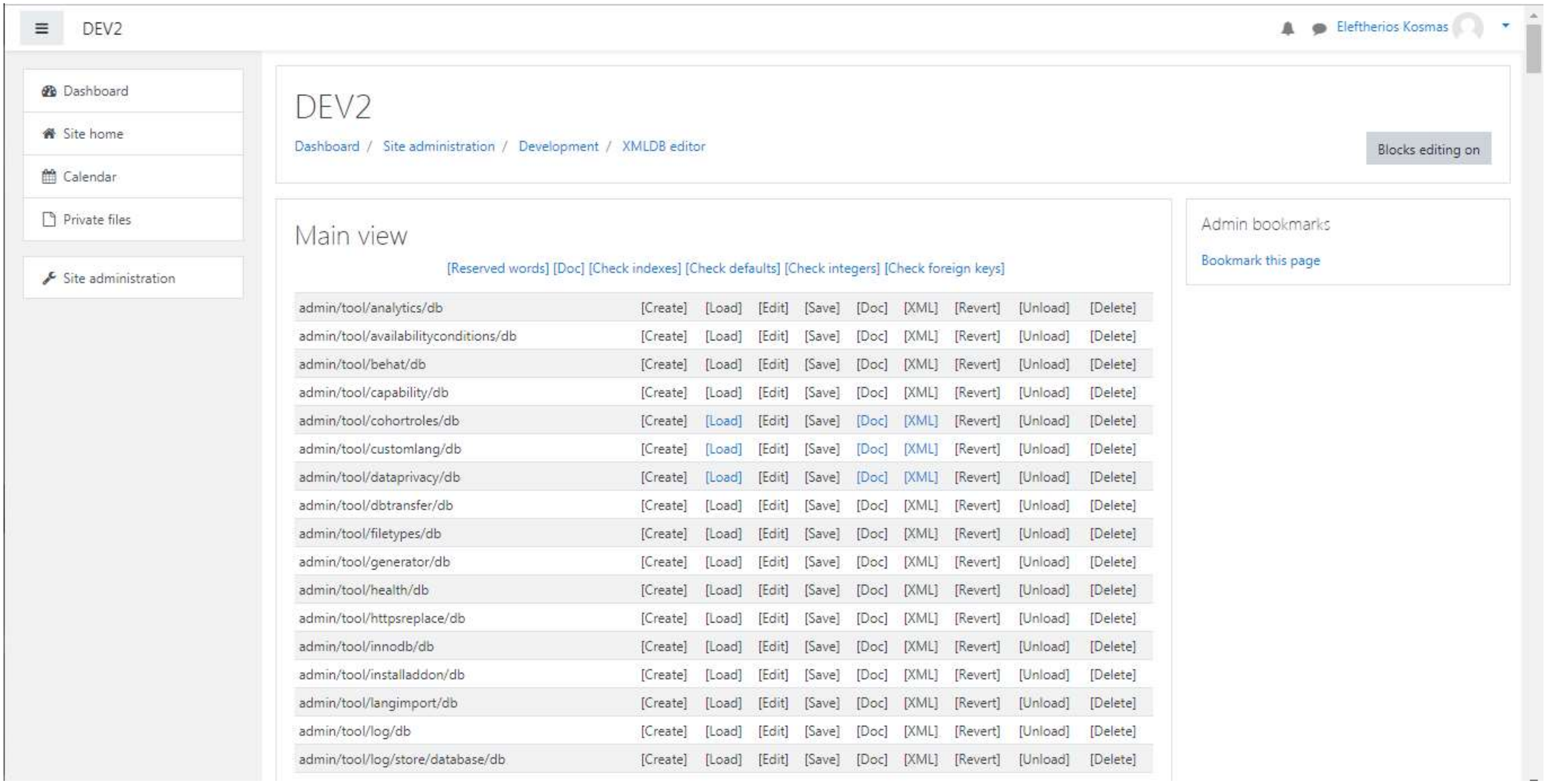
4Cs Dashboard

Block plugin - Database - Installation

- to **install** database tables → moodle **requires** the file **`install`**`.xml`
  - contains the table schemas
- to **create** this file → moodle provides an **editor** (XMLDB)
  (https://docs.moodle.org/dev/XMLDB_editor)
  - located in the **administration block**: "**Site administration | Development | XMLDB editor**"
    (moodle/admin/tool/xmldb/)

- XMLDB **editor**
  - **creates** an **`install`**`.xml` file → which should be loaded during the install
    to create plugin's tables
  - **produces** php **update code** for adding and updating moodle database tables

Moodle XMLBD Editor



DEV2

Dashboard / Site administration / Development / XMLDB editor

Admin bookmarks

Bookmark this page

Main view

[Reserved words] [Doc] [Check indexes] [Check defaults] [Check integers] [Check foreign keys]

admin/tool/analytics/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/availabilityconditions/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/behat/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/capability/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/cohortroles/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/customlang/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/dataprivacy/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/dbtransfer/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/filetypes/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/generator/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/health/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/httpsreplace/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/innodb/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/installaddon/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/langimport/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/log/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]
admin/tool/log/store/database/db	[Create]	[Load]	[Edit]	[Save]	[Doc]	[XML]	[Revert]	[Unload]	[Delete]

4Cs Dashboard

Block plugin - Database - Visual overviews

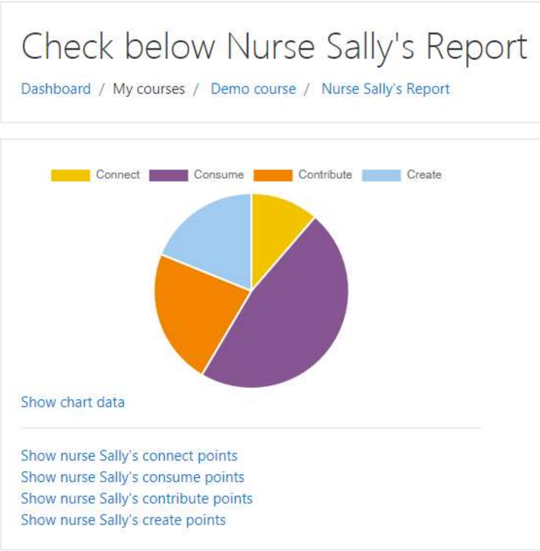
- we use moodle **chart API** in order to provide visual overviews (https://docs.moodle.org/dev/Charts_API)

```
$chart = new \core\chart_pie();

$series = new core\chart_series('Points', [6,25,12,10]);
$labels = array(['Connect',
                 'Consume',
                 'Contribute',
                 'Create']);

$chart->add_series($series);
$chart->set_labels($labels);

echo $OUTPUT->render($chart);
```



Check below Nurse Sally's Report

Dashboard / My courses / Demo course / Nurse Sally's Report

Show chart data

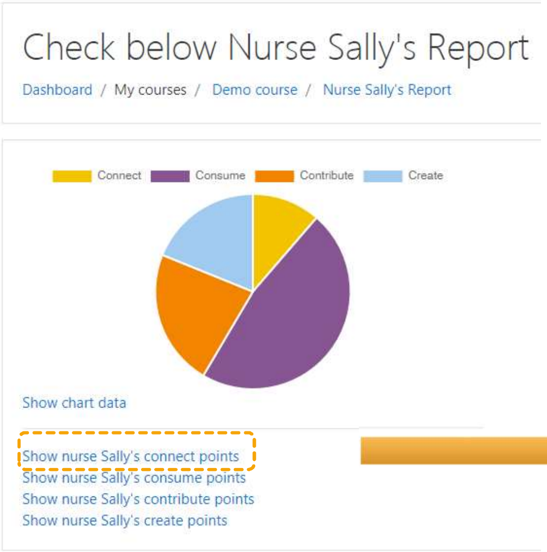
Show nurse Sally's connect points
Show nurse Sally's consume points
Show nurse Sally's contribute points
Show nurse Sally's create points

aggregate visual overview of all students' rewards per 4Cs behavior

4Cs Dashboard

Block plugin - Database - Visual overviews

- we use moodle **chart API** in order to provide visual overviews (https://docs.moodle.org/dev/Charts_API)

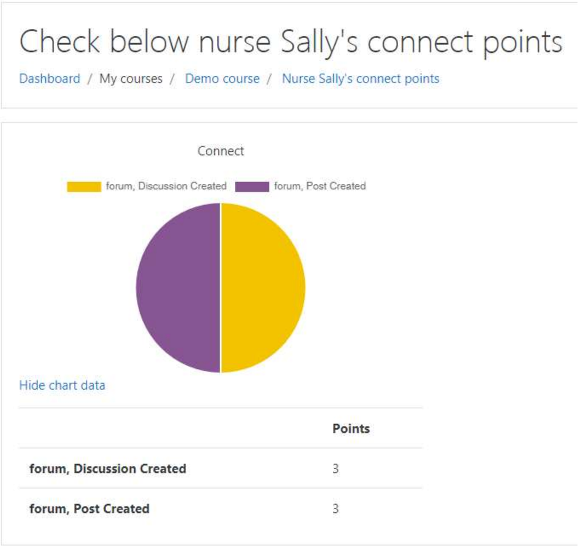


Check below Nurse Sally's Report

Dashboard / My courses / Demo course / Nurse Sally's Report

Show chart data

Show nurse Sally's connect points
Show nurse Sally's consume points
Show nurse Sally's contribute points
Show nurse Sally's create points



Check below nurse Sally's connect points

Dashboard / My courses / Demo course / Nurse Sally's connect points

Hide chart data

	Points
forum, Discussion Created	3
forum, Post Created	3

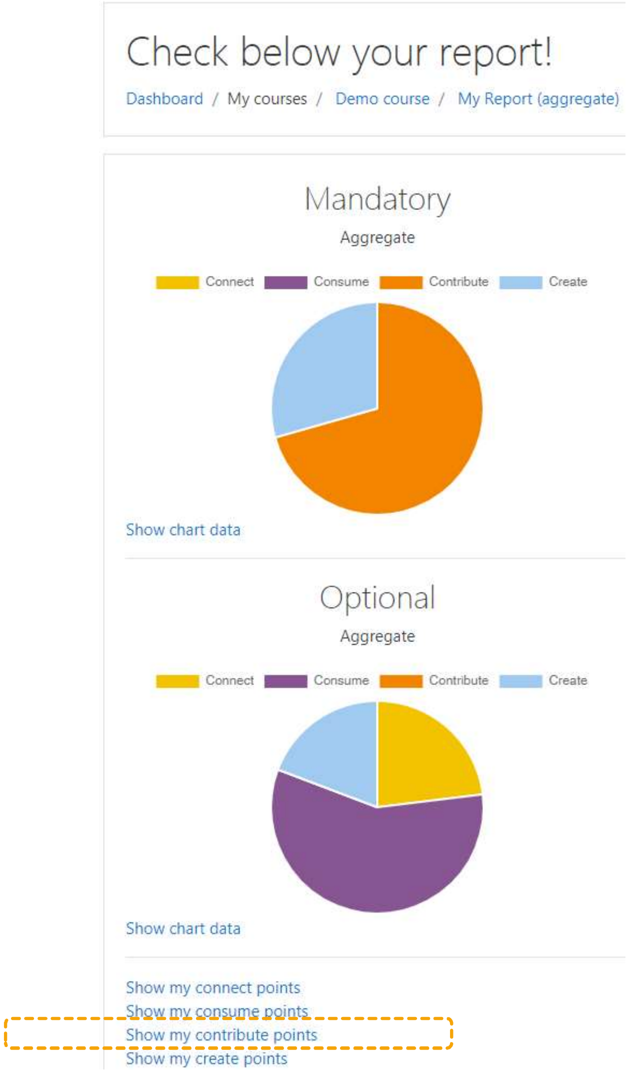
aggregate visual overview of all students' rewards per 4Cs behavior

aggregate visual overview of all students' rewards for Connect behavior per activity

4Cs Dashboard

Block plugin - Database - Visual overviews (II)

- we use moodle **chart API** in order to provide visual overviews (https://docs.moodle.org/dev/Charts_API)



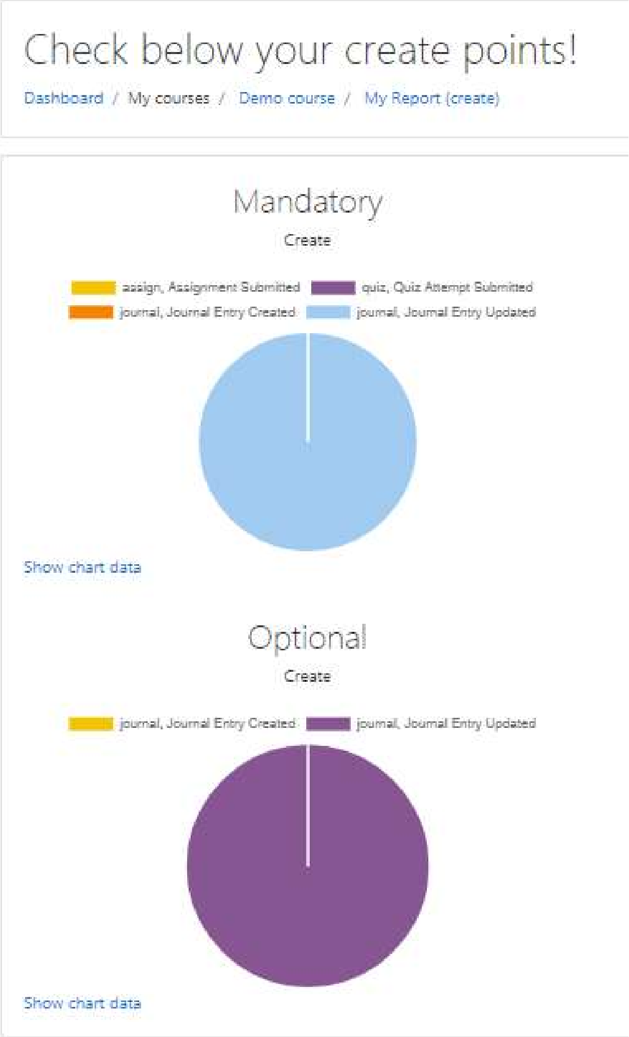
personal visual overview of some student's rewards per 4Cs behavior and type (mandatory/optional)

4Cs Dashboard

Block plugin - Database - Visual overviews ()

- we use moodle **chart API** in order to provide visual overviews (https://docs.moodle.org/dev/Charts_API)

personal visual overview of some student's rewards for Create behavior and type (mandatory/optional)



Check below your create points!

Dashboard / My courses / Demo course / My Report (create)

Mandatory
Create
assign, Assignment Submitted
quiz, Quiz Attempt Submitted
journal, Journal Entry Created
journal, Journal Entry Updated
Show chart data
Optional
Create
journal, Journal Entry Created
journal, Journal Entry Updated
Show chart data

4Cs Dashboard

Overview

- can be installed **per course**

- **automatically** discovers
  - the **sections** (i.e. topics) of the course and
  - the **activities** (e.g. forum, file, quiz)of each section

- **automatically** tracks **events** performed by students on the activities and accounts them the corresponding rewards

- **maintains** a database with the required information

• • •

• • •

- ▶ …

• • •

- …