



UNIVERSITÉ
LAVAL

Markus: New marking scheme

Functional & Technical analysis

presented to

The MarkUs development team

from

Team 01 — Laval's trio

<i>matricule</i>	<i>name</i>	<i>email</i>
906 266 951	Mélanie Gaudet	melanie.gaudet.1@ulaval.ca
906 309 769	Simon Giroux-Lavigne	simon.lavigne-giroux.1@ulaval.ca
907 148 364	Gabriel Roy-Lortie	gabriel.roy-lortie.1@ulaval.ca

Université Laval
october 14, 2009

Version history

<i>version</i>	<i>date</i>	<i>description</i>
	October 7, 2009	Creation of the document
1.0	October 14, 2009	Delivery 1: Functional & Technical analysis
1.1	October 16, 2009	Corrected the english problems (orthographic and syntactic)
1.2	October 26, 2009	Final proposition

Contents

List of Figures	iii
1 Introduction	1
2 Description	2
3 Functional	3
3.1 User Stories	3
3.1.1 Create / edit assignments	3
3.1.1.1 Create an assignment	3
3.1.1.2 Change the marking scheme	3
3.1.2 Flexible marking scheme criterion edition	4
3.1.2.1 Create a new criterion	4
3.1.2.2 Re-order criteria	4
3.1.2.3 Edit a criterion	5
3.1.3 Grade submissions with the flexible marking scheme view	5
3.1.4 Consult results with the new marking scheme view	6
3.2 Prototype	6
3.2.1 Marking scheme selection	6
3.2.2 Flexible criteria entry	6
3.2.2.1 Criteria position	8
3.2.3 Flexible criteria grading	8
3.2.4 Flexible criteria grading summary	8
4 Technical	10
4.1 Modifications to the Data Base schema	10
4.2 Class diagram	12
4.3 Tickets	12
4.3.1 First marks table migration	12
4.3.2 Second marks table migration	12
4.3.3 Assignment table migration	12
4.3.4 Refactor the model	13
4.3.5 Rename the tab “Rubrics” for “Marking Scheme”	13

4.3.6	Assignment's property view	13
4.3.7	The (newly named) "Marking Scheme" tab has to support the correct format depending on the marking scheme used for the current assignment	13
4.3.8	Refactor the Submission's grading view	13
4.3.9	Refactor Students' view	14
A	Grep	15
A.1	Grep on rubric_criterion_id	15

List of Figures

- 3.1 Marking scheme selection 7
- 3.2 Flexible criteria entry 7
- 3.3 Flexible criteria position 8
- 3.4 Flexible criteria grading 8
- 3.5 Flexible criteria grading summary 9

- 4.1 Database schema 11
- 4.2 Class Diagram 12

Chapter 1

Introduction

This document contains functional and technical information about the new flexible marking scheme. First, all the user stories associated with it are presented. Then, prototypes show how we plan to integrate the new marking scheme into MarkUs. In the technical part of the document, we discuss the database and class models. Finally, we divide the user stories into tickets.

Chapter 2

Description

From its birth to now, MarkUs offered, to instructors and TAs, tools to grade assignments upon criteria based on the *Rubric* marking system. But now, the need has been expressed to be given a more flexible marking opportunity.

It has been determined that a more flexible marking scheme should have the following characteristics:

1. The flexible marking scheme is not compatible with the existing rubric system, so they cannot be used together on the same assignment;
2. The flexible marking scheme has a variable number of criteria where each criterion has:
 - (a) a name;
 - (b) a description that may be formatted (user would really like to have lists);
 - (c) a score that ranges from 0 to a specified maximum value;
 - (d) The flexible marking scheme will also include the ability to add bonus marks or deductions.
3. It will be referred to as the *Flexible* marking scheme until we get some pedagogical input on its final name.

Chapter 3

Functional

3.1 User Stories

Here are the new user stories related to the flexible marking scheme.

3.1.1 Create / edit assignments

Feature: Create or edit assignments

As an instructor

I want to be given the choice of which marking scheme an assignment should be graded with.

3.1.1.1 Create an assignment

Scenario: Create an assignment

Given I need to create an assignment

And I'm in the property view

Then I should be able to choose which marking scheme to use.

Acceptance criteria:

1. The view should feature a pair of radio buttons allowing to choose which marking scheme to use (*Rubric* or *Flexible*);
2. The created assignment should use the selected marking scheme.

3.1.1.2 Change the marking scheme

Scenario: Change the marking scheme

Given I need to edit an assignment

And I'm in the property view

Then I should be able to change the marking scheme in place.

Acceptance criteria:

1. If criteria have already been defined, I should be warned that criteria information is going to be lost;
2. If no criteria have been defined, I should be able to “save changes”.

3.1.2 Flexible marking scheme criterion edition

Feature: Editing “flexible” criteria

As an instructor

I want to edit the criteria upon which an assignment is graded.

3.1.2.1 Create a new criterion

Scenario: Create a new criterion

Given I’m editing an assignment using the flexible marking scheme

And I’m in the “Marking Scheme” tab

And I click “Add criterion”

Then I should be prompted for the name of the new criterion.

Acceptance criteria:

1. I should be able to cancel the operation;
2. I should not be able to create an un-named criterion;
3. I should not be able to specify an already used name;
4. I should be able to create a new criterion by entering a name and clicking “OK”;
5. The newly created criterion should appear at the end of the criteria list
6. The newly created criterion should have been filled with default initial values (maximum value is 0 and description is empty).

Note: That last item has been introduced in order to mimic the actual rubric criterion’s creation work flow.

3.1.2.2 Re-order criteria

Scenario: Re-order criteria

Given I’m editing an assignment using the flexible marking scheme

And I’m in the “Marking Scheme” tab

And I drag a criterion to a new location in the list

Then I should see the criteria order change.

Acceptance criteria:

1. I should be able to drag any criterion to any location in the list (above the first, between two others or after the last);
2. Criteria’s position attribute should be updated;
3. This feature should work the same as re-ordering rubric criteria.

3.1.2.3 Edit a criterion

Scenario: Edit a criterion

Given I'm editing an assignment using the flexible marking scheme
And I'm in the "Marking Scheme" tab
And I click a criterion's name in criteria list
Then I should see the criterion's properties appear on the right-hand panel.

Acceptance criteria:

1. I should be able to see/edit the criterion's name;
2. I should be able to see/edit the criterion's maximum value;
3. I should be able to see/edit the criterion's description;
4. I should be able to save changes;
5. I should not be able to save a criterion without a name;
6. I should not be able to save a criterion without a valid numeric maximum value;
7. I should be able to cancel changes;
8. I should be able to remove the criterion by clicking a "Delete" link.

3.1.3 Grade submissions with the flexible marking scheme view

Feature : Grade submissions with the flexible marking scheme view

As an Instructor or TA

I want to grade submissions with the flexible marking scheme view.

Scenario : Grade a submission with the flexible marking scheme view

Given I have an assignment using the flexible marking scheme to evaluate
And I am grading a submission for this assignment
Then the criteria should be viewed in the flexible marking scheme format.

Acceptance criteria :

1. The view should have the same features offered in the rubric view;
2. I should be able to expand the criteria and see the descriptions;
3. I should be able to collapse the criteria;
4. I should be able to expand the unmarked criteria;
5. I should see the final mark in the summary;
6. The final mark is calculated and not saved.

3.1.4 Consult results with the new marking scheme view

Feature : Consult results with the flexible marking scheme view

As a student

I want to see my results in the flexible marking scheme view.

Scenario : Consult my results with the flexible marking scheme view

Given the flexible marking scheme is used for the assignment

And the results are released

I want to see my results for each criterion and my final mark

Acceptance criteria :

1. The view should have the same features as those in the rubric view;
2. I should see the description of each criteria;
3. I should be able to see my results for each criterion;
4. I should see my final mark in the summary.

3.2 Prototype

3.2.1 Marking scheme selection

The Instructor is allowed to choose the marking scheme on the assignment creation. Only one marking scheme can be selected for each assignment. If the user chooses *rubric* marking scheme, the application will behave as it used to apart the fact that the *Rubric* tab is now named *Marking Scheme* (but features the same content). If the Instructor chooses *flexible* marking scheme, the *Marking Scheme* tab will feature the new interface for flexible criteria edition.

If the Instructor decides to switch to another marking scheme, he can do it at any moment. If grading is started, a warning message will be displayed, and the modification will be applied if he really wants to continue. The grading done with the other marking scheme will be lost.

Refer to figure 3.1 to see the Marking Scheme selection screen layout.

3.2.2 Flexible criteria entry

The Instructor can add criteria through the *Marking Scheme* tab. He can choose to add each criteria manually or he can upload them as a CSV file. The criteria description is mandatory and the default maximum value is 0.

Refer to figure 3.2 to see the *Marking Scheme* tab layout.

Assignment Properties

Short Identifier: TW1

Name: TeamWork 1

Message: [Empty text area]

Due Date: [Empty date field] Format: YYYY-MM-DD HH:MM - Example: 2009-01-13 17:44

Marking scheme

Flexible

Rubrics

...

Submit

Figure 3.1 – Marking scheme selection

Flexible Management: TW1 [Add Criterion](#) | [Upload](#) | [Download](#)

Criterion	Maximum value (14.5)
Language Quality	5.0
Code Quality	9.5

Maximum value total

Code quality [Save changes](#)

Criteria: Code quality [Delete](#)

Maximum value: 9.5

Description: Code quality description...

Figure 3.2 – Flexible criteria entry



Figure 3.3 – Flexible criteria position

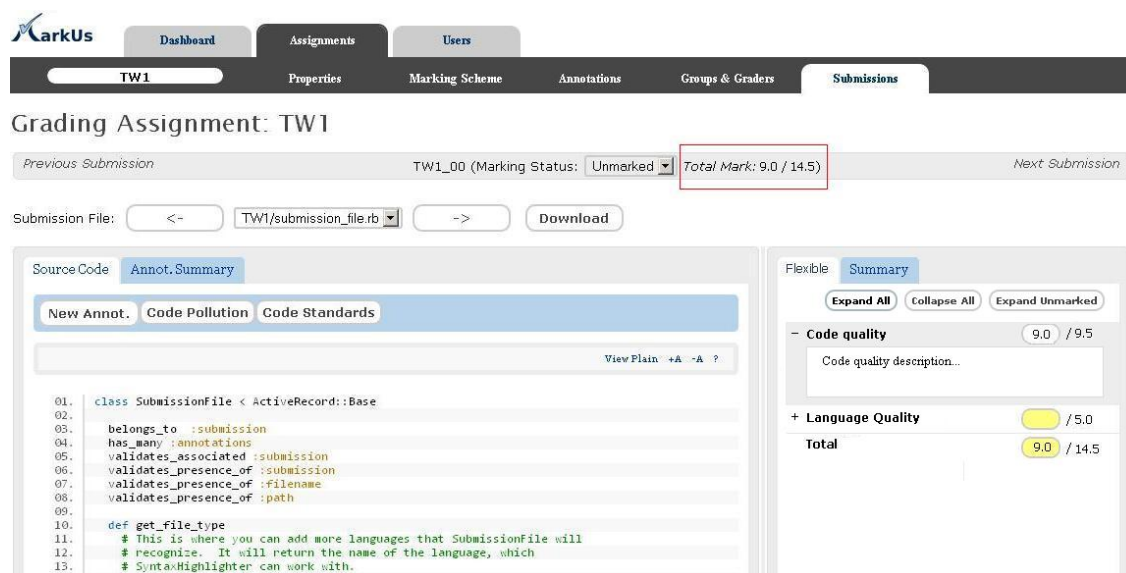


Figure 3.4 – Flexible criteria grading

3.2.2.1 Criteria position

Instructor can reorder the criteria as it is possible to do with rubric criteria. Refer to figure 3.3.

3.2.3 Flexible criteria grading

Instructors and TAs can do grading, Students can only view. The grading is done, by Instructor or by TA, manually for each criterion. The application verifies that it is impossible to insert a grade higher than the maximum value. Yellow boxes indicates that the grading is started, but only for some criteria.

Refer to figure 3.4 to see the *Flexible Grading* screen layout.

3.2.4 Flexible criteria grading summary

The grading detail is available in the *Summary* tab. Instructor, TA and Student have access to this screen.

Refer to figure 3.5 to see the *Flexible Grading Summary* screen layout.

Flexible		Summary	
Title	Maximum value	x Mark	= Grade
Code quality	9.5	1	9 / 9.5
Language Quality	5.0	0	0 / 5.0
Subtotal:			9.0 / 14.5
Bonus/Deductions			
Comment	Mark		
Add Extra Mark			
Total Extra Marks:			0.0
Comment	Percentage		
Total Extra Percentage:			0.0
Translated to Points:			0.0
Final Mark			9 / 14.5

Figure 3.5 – Flexible criteria grading summary

Chapter 4

Technical

4.1 Modifications to the Data Base schema

This section presents the adopted database schema. The schema illustration comes with a description of the modifications required and a list of aspect related to the choosen database schema.

After many discussions everybody agree to implement scenario #4. Figure 4.1 illustrates the required database modifications.

Description of the database modifications:

1. New field `assignment.marking_scheme_type`, to save the marking scheme selection;
2. New table `flexible_criteria`, which contains the fields required for a *Flexible Marking Scheme*;
3. Rename field `marks.rubric_criterion_id` for `marks.criterion_id`;
4. Modify the field `marks.mark` format from integer to decimal.
5. Add the field `marks.criterion_type` to use Rails' polymorphic relations.

Some aspects related to the choosen database schema:

1. Database space is properly used:
 - (a) Each record created for a flexible criteria is composed of exactly and only the fields required by a flexible criteria;
 - (b) Each record created for a rubric criteria is composed of exactly and only the fields required by a rubric criteria;
 - (c) Polymorphic associations is available through `mark.criterion_id` and `mark.criterion_type`.
2. Renaming `mark.rubric_criterion_id` for `mark.criterion_id`, implies some modifications in the existing programs. Every references to `mark.rubric_criterion_id` have to be changed for `mark.criterion_id` (See A.1);

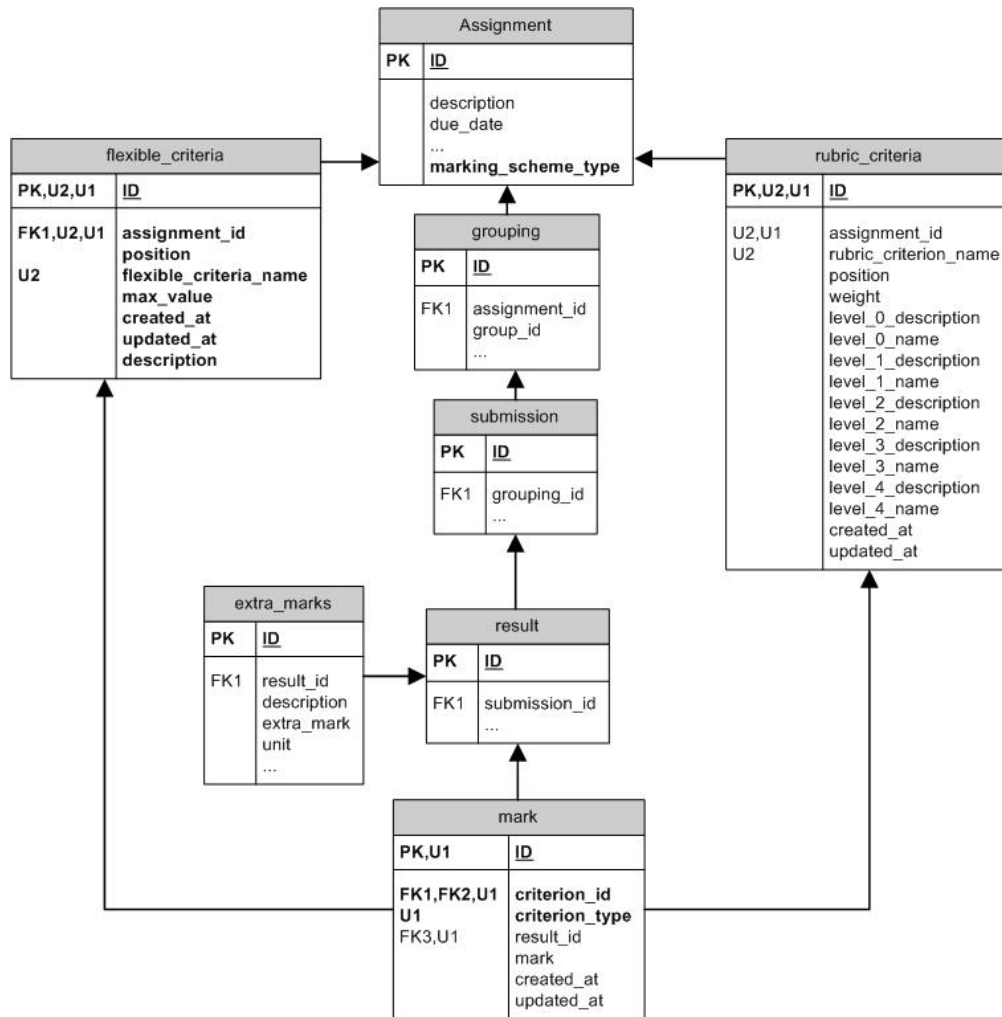


Figure 4.1 – Database schema

3. If marking scheme type is modified, purge all the data that could have been created with another marking scheme.

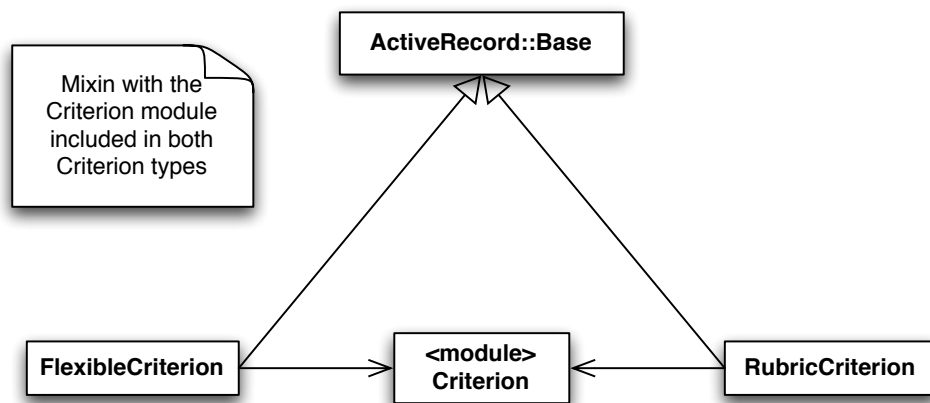


Figure 4.2 – Class Diagram

4.2 Class diagram

Figure 4.2 shows the classes we need to add the new marking scheme. FlexibleCriterion and RubricCriterion inherit from ActiveRecord::Base. The Criterion module is created to contain all the common methods for the types of criterion. Both Criterion types will include this module to do a "mixin".

4.3 Tickets

Here are the (high level) tickets that ought to be created in order for us to fit the job in the remaining iterations.

4.3.1 First marks table migration

1. Produce a DB migration to rename the field `marks.rubric_criterion_id` for `marks.criterion_id` and to add the field `marks.criterion_type` to support polymorphic relations;
2. Modify the classes using `marks.rubric_criterion_id`.

4.3.2 Second marks table migration

1. Produce a DB migration to modify the field `marks.mark` format from integer to decimal.

4.3.3 Assignment table migration

1. Produce a DB migration to add the field `assignment.marking_scheme_type` with default to "rubric".

4.3.4 Refactor the model

1. The `FlexibleCriterion` class has to be created. If necessary, the `Criterion` module has to be created and included in both criterion types. It will contain the methods common to a criterion;
2. The `Assignment` object needs a `marking_scheme_type` property;
3. The verification in `Mark` has to be done by `RubricCriterion` or `FlexibleCriterion` Class in order to allow any positive (≥ 0) decimal instead of the actual (> -1 and < 5 validation) for marks of `FlexibleCriterion`.

4.3.5 Rename the tab “Rubrics” for “Marking Scheme”

4.3.6 Assignment’s property view

1. Refactor instructors’ Assignment marking scheme view;
2. The Assignment’s property view has to offer to choose a marking scheme;
3. Display a warning message if the user changes the marking scheme type (All marks done with another marking scheme will be lost);
4. Purge all data related to an old marking scheme type if the user changes the marking scheme type.

4.3.7 The (newly named) “Marking Scheme” tab has to support the correct format depending on the marking scheme used for the current assignment

Note: The left column could stay precisely the same since it offers all the needed functionality for either marking scheme. The right-part would have to be refactored to provide necessary additional functionalities for the flexible marking scheme’s criteria.

Needed fields: "Criterion" (name), "Description" and "Maximum" mark.

4.3.8 Refactor the Submission’s grading view

This change impacts Instructors’ and TAs’ view.

1. When grading a submission for an assignment using the flexible marking scheme, the appropriate UI should show.

Guidelines: Keep the Expand/collapse logic. When a criterion is "expanded", you get an edit field allowing to type a result. Once a result is validated, the corresponding criterion is collapsed and the entered value is shown in plain text. The user should re-expand a criteria to change the value if it is needed.

4.3.9 Refactor Students' view

If all preceding tickets are completed, the following should be trivial:

1. Assert that the Student's result view shows appropriately the marks given to each criterion and that the total calculates correctly (shouldn't be a problem since we intend to leave that logic in the base `Criterion` module);

Appendix A

Grep

A.1 Grep on rubric_criterion_id

The following files contains references to the word `rubric_criterion_id` and needs to be analysed to figure out the impact.

```
markus-0.5/trunk/test/unit/mark_test.rb
markus-0.5/trunk/test/unit/.svn/text-base/mark_test.rb.svn-base
markus-0.5/trunk/test/fixtures/marks.yml
markus-0.5/trunk/app/helpers/submissions_helper.rb
markus-0.5/trunk/app/models/mark.rb
markus-0.5/trunk/app/controllers/results_controller.rb
markus-0.5/trunk/app/controllers/rubrics_controller.rb
markus-0.5/trunk/app/controllers/submissions_controller.rb
markus-0.5/trunk/app/views/results/common/_annotations.js.erb
markus-0.5/trunk/app/views/results/common/.svn/text-base/_annotations.js.erb.svn-base
markus-0.5/trunk/app/views/submissions/submissions_table_row/_filter_table_row.html.erb
markus-0.5/trunk/app/views/submissions/submissions_table_row/.svn/text-base/
  _filter_table_row.html.erb.svn-base
markus-0.5/trunk/log/development.log
markus-0.5/trunk/log/test.log
markus-0.5/trunk/db/schema.rb
markus-0.5/trunk/db/db_schema_postgresql.sql
markus-0.5/trunk/db/development_structure.sql
markus-0.5/trunk/lib/benchmarking/submission_files/submissions_controller.rb
markus-0.5/trunk/lib/benchmarking/submission_files/results_controller.rb
markus-0.5/trunk/lib/benchmarking/submission_files/rubrics_controller.rb
markus-0.5/trunk/coverage/app-controllers-results_controller_rb.html
markus-0.5/trunk/coverage/app-controllers-rubrics_controller_rb.html
markus-0.5/trunk/coverage/app-controllers-submissions_controller_rb.html
markus-0.5/trunk/coverage/app-helpers-submissions_helper_rb.html
```

markus-0.5/trunk/coverage/app-models-mark_rb.html