



**HAL**  
open science

## Detection and classification of dropout behavior

Timothée Duron, Laurent Gallon, Philippe Aniorte

► **To cite this version:**

Timothée Duron, Laurent Gallon, Philippe Aniorte. Detection and classification of dropout behavior. FECS'22 - The 18th Int'l Conf on Frontiers in Education: Computer Science and Computer Engineering, Jul 2022, Las Vegas, United States. hal-03834009

**HAL Id: hal-03834009**

**<https://hal.science/hal-03834009>**

Submitted on 28 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Detection and classification of dropout behavior

T. Duron<sup>1</sup>, L. Gallon<sup>1</sup>, P. Aniorte<sup>2</sup>

<sup>1</sup> Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Mont-de-Marsan, France

<sup>2</sup> Universite de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France

**Abstract** This article presents the first results of our work aimed at detecting students' risk of dropping out and dropping out during block-based computer programming lessons in the classroom. We use information related to the use of the mouse and the interface of the programming software to detect behaviors characteristic of dropout. The paper describes the methodology chosen to build the rules for detecting these behaviors. The results obtained on a first case study are also given.

These results were obtained within the framework of the PERSEVERONS project<sup>1</sup>, which aims to study the perseverance of students induced by the use of a digital object.

**Keywords:** code.org, block programming, dropout, perseverance, MHI, HCI, Flow theory

## 1 Introduction

Our work aims to characterize school perseverance when students work on a computer, and more precisely, to detect when they drop out. The main idea is to detect these dropouts as soon as possible (weak signals) in order to be able to act and re-engage the student as quickly as possible. Therefore, we are not looking for school perseverance in the academic sense, which is over a long period of time (semester, year, ...), but rather we are looking to qualify perseverance behavior (or dropout) over a very short period of time (one or several minutes). Several works [1][2][3][4] show that emotional and psychological state of the learner have a strong impact on its behavior. So the change of emotional and psychological state can be observed, and measured through the analysis of the use, for example, of the mouse and the keyboard of the computer[5][6][7][8], and more recently [9][10][11].

We make the assumption that the student's behaviors during a computer-assisted educational activity can be classified into several categories : working, risk of a dropout, and dropping out. This assumption is based on numerous works in the psychology of motivation, educational sciences, ... as shown in the figure 2 [12]. Most of them allow us to understand the mechanisms linked to motivation, before, after and during a pedagogical situation. But the most important one's for us is the FLOW theory [13], which links the emotional/psychological state of the learner to the challenge/skills balance of the activity. As the emotional state (and so the behavior of the learner) is

---

<sup>1</sup> <http://perseverons.inspe-bordeaux.fr/>

induced by this balance, and reflects it, we seek to detect and classify these different behaviors during an activity. to characterize the level of dropout of the learner.

In the rest of this paper, we present the methodology we used to build the tools (detection rules) for detecting the dropout level of the learner, and the validation process of these tools. Then we give the results of the first experiments we made, and identify trends regarding the effectiveness of our approach.

## 2 Methodology

The objective of the proposed methodology is to build the rules allowing to classify the learner's dropout behaviors, then to measure the efficiency of these rules in order to modify them if necessary. This methodology is inspired from the De Vicente one [14].

The methodology is decomposed into four major steps, as seen in figure 1 : capture the data, construct the rules, pre-validate rules and finally validate rules.

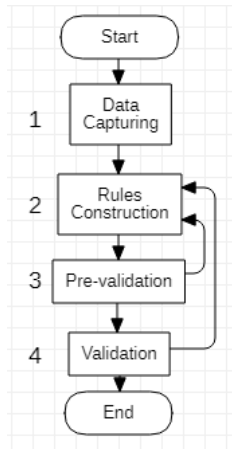


Figure 1: Methodology steps

### 2.1 First step : data capture

The first step is crucial, and concerns data capture. At this stage, we identify the sensors of the hardware and software platform used, thanks to which we will obtain the meaningful informations on the activity of the learner. These sensors can be varied. In our case, we are using a keylogger, which allows us to capture the learner's mouse and keyboard activity. We also use a screen recording, to see the evolution of the learner in the software interface. We differentiate the captured datas into two groups: functional datas, i.e. the data which will be used to build the detection rules (datas from keylogger and screenshot recording), and control datas, which will be used to validate these rules (video recording of the student + screen recording).

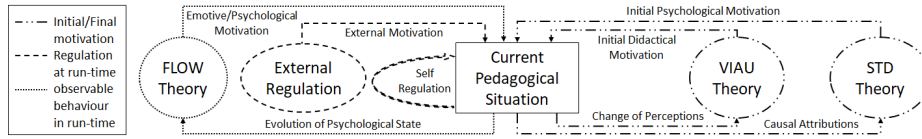


Figure 2: Federative view centered on the pedagogical situation

Basing on the functional datas, different informations can be build (we call them metrics). They are classified into three categories : intensity metrics (quantitative metrics), efficiency metrics (which assess the quality and the efficiency of the actions made by the learner), and semantic metrics (capture the meaning of the actions). Table 1 gives a list of metrics that we can build in our context.

Category	Sensor	Metrics
Intensity	Mouse	#pauses per minute
		pointer speed variation
		#clics per minute
		Dist. run through by the pointer per minute
Intensity	Keyboard	#keystrokes per minute
	Screen	#blocks moved per minute
Efficiency	Mouse	Mean pointer speed
		Drag-drop duration
		Pause duration
		Ratio activity/inactivity
		Sinuosity of the movements of the pointer
	Ratio drag/empty	
Semantic	Mouse	Mean time while pressing a key
		Pointer position in areas of interest
		Time spent by area of interest
		Clics in the void
	Screen	Clics on "start" button
		#blocs created
		#blocs deleted
		#blocs currently isolated

Table 1: Metrics from mouse, keyboard and screen sensors

## 2.2 Second step : rules construction

In step 2, we create a first version of the detection rules; basing on the metrics build in step 1. This construction is done in a completely empirical manner. We ask then some experts, who can be psychologists, experienced teachers, cognitive researchers, etc., to analyze, identify and describe significant behaviors with the recorded control

data. A significant behavior could be, for example, a student who is seen to huff and get impatient, and at the same time make a large number of button clicks, and continue to huff because the program does not respond quickly enough. Once these behaviors are identified, we try to characterize them using logical combinations of metrics from the captured functional data (detection rules). For example, in the case of the impatient student, we can extract from the "mouse" data the metric "number of clicks in a reduced time interval", and consider that we will detect an impatient behavior when this metric exceeds a certain threshold. In our case, we build 11 different detection rules using datas issued from 4 experiments (4 students of the same classroom following individually code.org<sup>2</sup> courses). We give a short description of the rules in the table 2.

M[1]i;ml

Rules	State(s)	Description
R1	Risk of dropout	High number of clicks without load. This rule represents impatient, stressful behaviors, where pup
R2	Risk of dropout	Long load time and strong sinuosity. This rule represents hesitation behaviors, where during the cr
R3	Risk of dropout or Dropout	Reset between two pauses. Here, the student between two long pauses, resets and reruns the simul
R4	Risk of dropout	Deletion of the program. When a student deletes his complete program to start again from scratch,
R5	Work	Low acceleration over a long time. This rule describes a behavior that we observed, where the stud
R6	Work	Rapid change in the number of connected blocks. This rule describes a reorganization of the progr
R7	Dropout	Several times the R4 rule, i.e. several deletions of its complete program, on the same exercise. Thi
R8	Risk of dropout or Dropout	A few movements of the pointer without clicking between two long pauses, reflects either the com
R9	Work	Rapid back and forth between the loading (reserve of block) area and the working area (canvas). H
R10	Dropout	Several tests in a row without modifying the program. Behavior representing a perception of diffic
R11	Dropout	Long time without activity. This rule reflects a dropout, whether due to disinterest or great difficul

Table 2: Rules descriptions

After the construction of the first version of the detection rules, we can pass to a pre-validation phase (step 3) which aims to check if the detection rules define in step 2 detect at the right time the behaviors identified by the experts. To do this, we apply the detection rules on the functional datas captured, and compare the the results (list of dates corresponding to points of interest) with the points of interest identified by the experts in step 2. If some detection rules are not satisfying (i.e. their result does not match that of the experts), they are modified until the pre-validation test is positive (loop with step 2 on figure 1). In our case, the 11 defined rules correctly identified the points of interest initially put forward by the experts.

Then, we give this list of points of interest to a new group of experts, without specifying the type of detection carried out by the rules. We ask them to characterize the type and level of dropout on these points of interest, but also to let us know if in the control data, they identify dropout situations not listed in the list we provided them. The main idea of this part is to quantify the performance of the detection rules.

The results of step 3 of our firsts experiments are detailed in the following of the article. It should be noted that we were largely disturbed by the Covid period, and

<sup>2</sup> <https://code.org/>

that consequently a limited number of experiments could be carried out, as well as a mobilization of a limited number of experts. This explains why we are currently only in phase 3 of our methodology. However, the results obtained seem sufficiently interesting to us to be presented to the scientific community for opinion.

### 2.3 Third step : pre-validation

The result are only in the prevalidate step, due to a lack of new data set. For that we applied our 11 rules our are commons by them constructions with the 6 students. We take all the moment, our rules detect something and give the list of this moment to expert without which rules is activated, and which states the prototype detect.

The expert have to give only one state to each moment in the list (around 15 moments by seance of 45 min by student). We allow them to give a second states, in case of doubt, but to give the states they thinks the more probable first.

In this paper, we talk about raw result, when we only take into account only the first answer they give (we take into account like they have no expert doubt to any points).

### 2.4 Pre-validation result

In this paper, we talk about raw result, when we only take into account only the first answer they give (we take into account like they have no expert doubt to any points).

These results presenting only come from the first iteration of the process<sup>1</sup>, at the pre-validation step (step 3). We applied the 11 rules on 4 pupils included in the 6 pupil used to created the rules. For the 4 pupils, the prototype had detect respectively (16, 25, 12, and 18 points), so 71 points of detection in total.

Only two more point had been suggested by the expert, that mean, that prototype seems to detect properly the important moments.

We begin to evaluate to each point the difference of the 3 opinions in the same time (prototype, expert1 and expert2). We can see, the result in the figure 3. The green part with 37% is when the two expert give the same state that prototype detect. 31% in blue, the prototype concurring with only one expert, and in conclit with the second one. In grey, is the case where nobody is agreed, they is any matching between prototype and one of the expert, and also between experts. These case need to be study further to understand them. And the red represent the case that opinion of expert concurring together, but not with the expert, so this case represent the false detection from prototype.

The cases in green, are validated cases regarding to our purpose. In blue, these cases could be validated because one the two expert has agreed with us. It means that 68% in total of the cases are validated. The non validated cases (in red) need to be improved through the adaptation of the required rules. The other cases (in grey) cannot be validated, or invalidated because nobody is agreed. In 60% of grey case we are between the two experts opinions with "risk of dropout", the other 40% of grey case, prototype said working, instead of expert saying "risk of dropout" and "dropout". This represent respectively 9% and 6% to the total of cases. So 6% seems to be invalidate, because the two experts are more of a begining or a complete dropout, while the prototype detect a working state. The other 9% seems more on our side, and could be validated, because

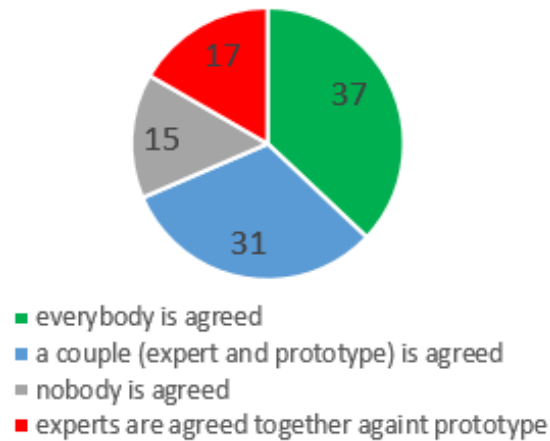


Figure 3: Confrontation 3 together of Raw result to 3-states summary (working, risk of dropout, dropout)

we are the mean of the two experts opinions. Which would bring the total number of validated cases to 77%.

To go further in the evaluation of the prototype, we also look for confrontation between each expert to the prototype, and between experts themselves (table 3).

The raw results for detecting a dropout (or the risk of dropout) are good, with 52-54% of cases where each expert is agreed with the prototype results (see table 3). In 43-44% of case each expert have a slightly difference of opinion (for example prototype said risk of dropout and one expert said work). This kind of case, 1/3 of this case are hesitation between "drop out" and "risk of dropout", the 2/3 are hesitation between "risk of dropout" and "work". It is normal to have this kind of case, because the border between this different states can be defined clearly, the border is more like a blurred zone. However for the moment, they are too high to be only the blurred zone, specially between "work" and "risk of dropout", and the next iteration will permit to improve the rules and decrease these kind of case.

And in 2-6% there is a very large disagreement between an expert and the detection of the prototype, that mean one said "dropout", while the other says "work".

But more interestingly, we can see that the results of the confrontation of expert opinions are surprising. In spite of an equivalent or slightly better percentage on the number of cases where they agree, the number of cases of contradictory opinion is at least twice as high, reaching up to 13%.

## 2.5 Fourth step : validation

Once the pre-validation test is satisfying, we can then move on to a more general validation (step 4), with new experiments and new sets of data captured, in order to see if the detection rules applies to others sessions of the same student or group of students.

	prototype vs expert1	prototype vs expert2	expert1 vs expert2
Agreed	54%	52%	56%
Small difference in opinion (1 state difference)	44%	43%	31%
Big difference in opinion 5 (more than 1 state difference)	2%	6%	13%

Table 3: Confrontation two by two, 3 states version

At this stage, if some issues are identified, we can move back to step 2 to modified some of the rules if necessary. A rule is considered valid if at least it allows to have a good detection rate on other data sets then the one used in step 2.

### 3 Conclusions and perspectives

In this paper, we relate the first experimentation, of the first iteration of our methodology. The result are really encouraging, to continue and improve the prototype.

We are also working of improving the expert review, by working on our protocole and description of the differents states. Because, we can see it clearly, in table 3, that the prototype are more closer to each expert, than the two expert themselfe. It could also permit to reduce the "blurred line" between state.

We also working now of increasing the number of expert, to have particularely for the cases in blue in figure 3, a majority of expert agrees with prototype (and not a 50/50), but the disparity between expert could bring a lot of noise, with to much experts.

We will also continue the methodology to improve the rule with the iterations planned is the process (figure 1).

We also plan to go further and detect more than just severity of the dropout, by including the two major reason of dropout according to flow theory[13]: difficulty and boredom. Some of the rules already take into account the reason. The prototype would then go from 3 possibles states to 5 states (difficulty dropout, risk of difficulty dropout, work, risk of boredom dropout, boredom dropout).

### References

1. E. Deci and R. M. Ryan, *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media, 1985.
2. T. P. Novak and D. L. Hoffman, "Measuring the flow experience among web users," *Interval Research Corporation*, vol. 31, pp. 1–36, 1997.
3. R. Viau, *La motivation en contexte scolaire*. Edition du Renouveau Pédagogique, De boeck, 2009.
4. D. Barbeau, "Pour mieux comprendre la réussite et les échecs scolaires," *Pédagogie collégiale*, vol. 5, no. 1, pp. 17–22, 1991.
5. A. Kołakowska, "A review of emotion recognition methods based on keystroke dynamics and mouse movements," in *2013 6th International Conference on Human System Interactions (HSI)*, 2013, pp. 548–555.



6. D. Ligarreto and D. López De Luise, “Metrics design for keyboard and mouse: Assessing learning levels,” in *2017 IEEE URUCON*, Oct 2017, pp. 1–4.
7. Y. M. Lim, A. Ayesh, and M. Stacey, “Detecting cognitive stress from keyboard and mouse dynamics during mental arithmetic,” in *2014 Science and Information Conference*, Aug 2014, pp. 146–152.
8. I. A. Khan, W.-P. Brinkman, N. Fine, and R. M. Hierons, “Measuring personality from keyboard and mouse use,” in *Proceedings of the 15th European Conference on Cognitive Ergonomics: The Ergonomics of Cool Interaction*, ser. ECCE '08. New York, NY, USA: ACM, 2008, pp. 38:1–38:8. [Online]. Available: <http://doi.acm.org/10.1145/1473018.1473066>
9. R. V. Aranha, L. N. Cordeiro, L. M. Sales, and F. L. S. Nunes, “Engagement and discrete emotions in game scenario: Is there a relation among them?” in *Human-Computer Interaction – INTERACT 2021*, C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen, Eds. Cham: Springer International Publishing, 2021, pp. 22–42.
10. A. Pentel, “Employing think-aloud protocol to connect user emotions and mouse movements,” 2015, cited by: 6.
11. S. Happy, A. Dasgupta, P. Patnaik, and A. Routray, “Automated alertness and emotion detection for empathic feedback during e-learning,” 2013, p. 47 – 50, cited by: 25; All Open Access, Green Open Access.
12. T. Duron, L. Gallon, and P. Aniorté, “Modelling learner’s perseverance in education software,” in *Information Systems Education Conference (ISECON)*, 2019, April 4-6, Galveston, Texas, USA.
13. M. Csikszentmihalyi, *Flow and education, Applications of Flow in Human Development and Education : The Collected Works of Mihaly Csikszentmihalyi*, 2014.
14. A. De Vicente, *Towards Tutoring Systems That Detect Students’ Motivation: An Investigation*, 03 2003.