KidLearn
machine learning applied to the personalization of didactic sequences

Manuel LOPES, Didier ROY, Benjamin CLÉMENT, Pierre-Yves OUDEYER

flowers.inria.fr

manuel.lopes@inria.fr

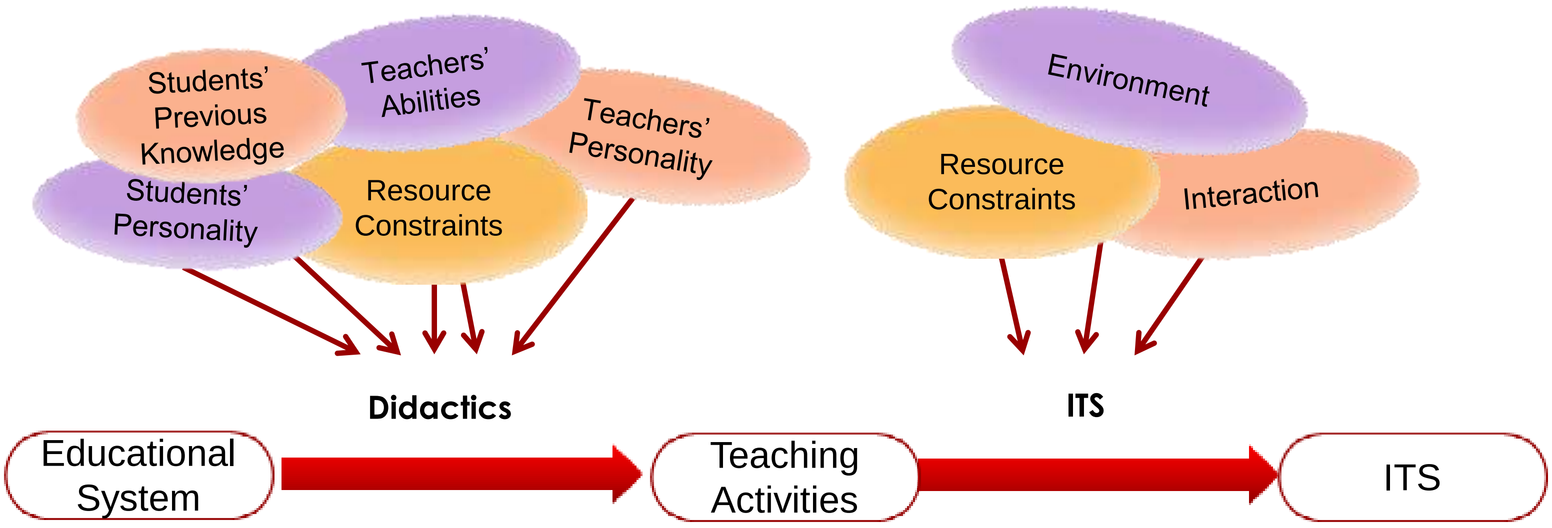
Inria informatiques mathématiques

Topics

1. How to select the best examples to teach at each time step?
   Multi-Armed Bandits for Intelligent Tutoring Systems, Manuel Lopes, Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer. *arXiv:1310.3174 [cs.AI]*, 2013

2. How to generate good examples for teaching?
   Algorithmic and Human Teaching of Sequential Decision Tasks, Maya Cakmak and Manuel Lopes. *AAAI Conf. on Artificial Intelligence (AAAI)*, 2012

Cognitive Model and ITS Design



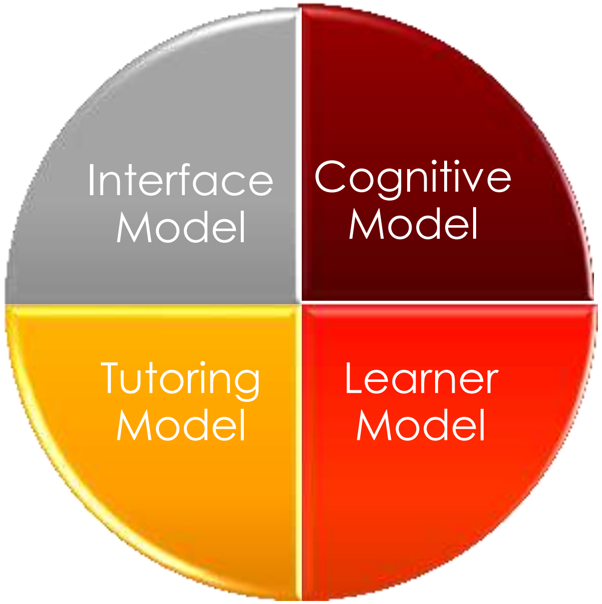
- Both steps may incur problems that do not allow students to acquire the competences aimed by the educational system
- Building an ITS requires a difficult pedagogial study

Objectifs of ITS

- Reduce the conception time of an automated tutoring system
- Provide more personalized teaching
- Adapt to more uncommon situations not accounted for at design time
- Reduce the time to acquire the different competences
- Improve motivation and engagement of learners

Intelligent Tutoring Systems

- A computer system that aims to provide immediate and customized instruction or feedback to learners, usually without intervention from a human teacher.
- Components of an ITS
  - Cognitive Model
  - Learner Model
  - Tutoring Model
  - Interface Model



Cognitive Model

- A set of Knowledge Units (KU)
- A set of activities with different parameters (ai)
- Q-Table with the relation between activities and the required competence level
- For a given exercise, the required competence level is:

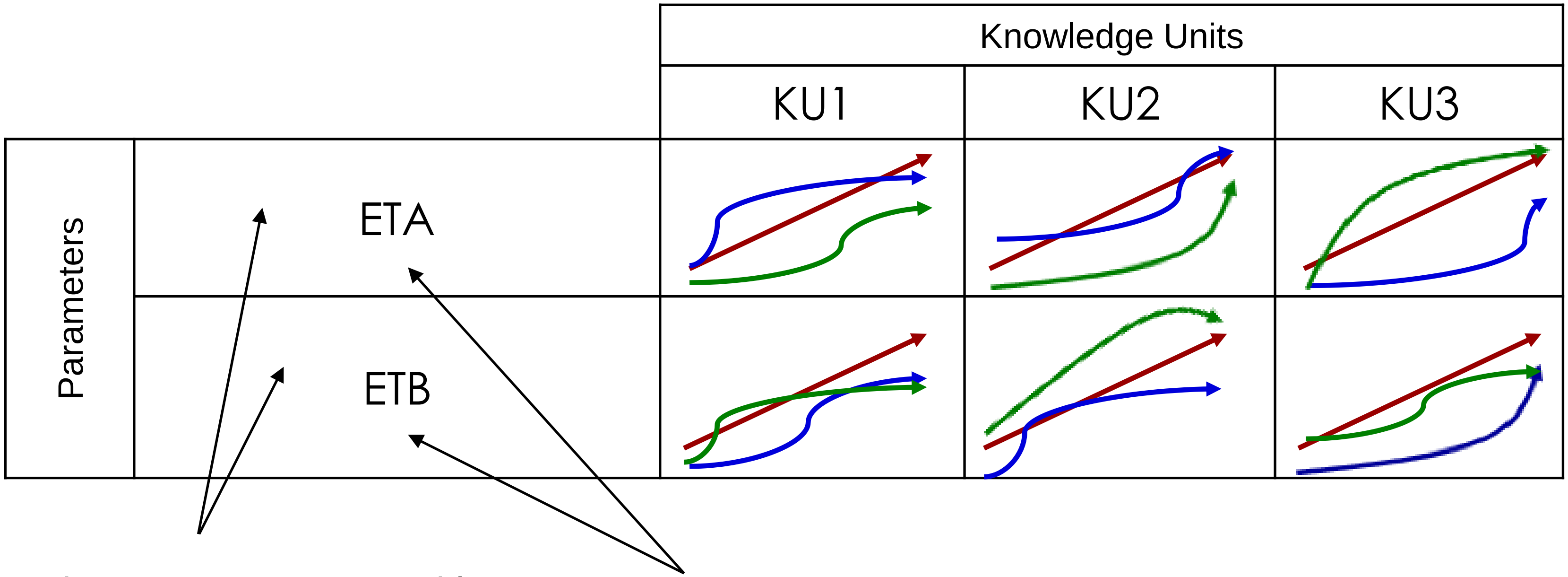
$$q_i(a) = \sqrt[n]{\prod_{j=1}^{n} q_i(a_j)}$$

Knowledge Units: Sum, Subtract, Count, …

		Knowledge Units		
Parameter	Values	KU_1	$\ldots$	KU_n
a_1	1	$q_{1,1,1}$	$\ldots$	$q_{n,1,1}$
a_1	2	$q_{1,1,2}$	$\ldots$	$q_{n,1,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_m	1	$q_{1,m,1}$	$\ldots$	$q_{n,m,1}$
a_m	2	$q_{1,m,2}$	$\ldots$	$q_{n,m,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_m	k_m	q_{1,m,k_m}	$\ldots$	q_{n,m,k_m}

The activities can be very different: interactive exercises, animations/videos, …

What is the best activity?



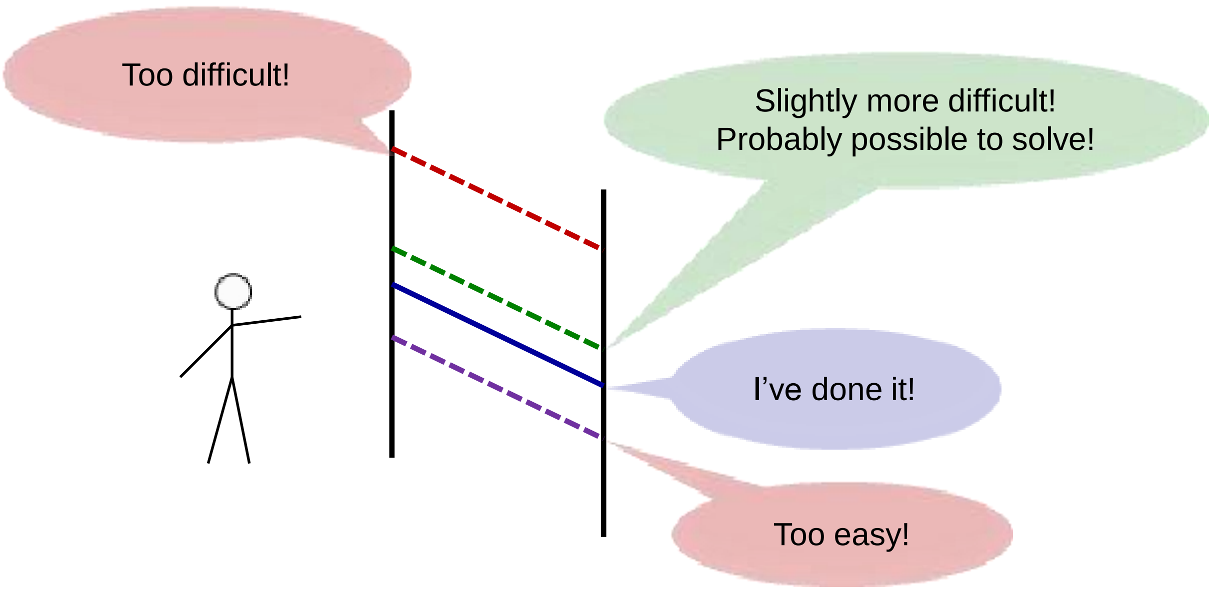
ETA and ETB are parameterazable exercises that can be used to allow students to acquire KU1-KU3.

Parameters: data for exercises, difficulty level, modality of presentation, type of interaction, …

Predefined sequence

Optimal sequence for specific students

Intrinsic Motivation



- **Maximum of motivation** : when the difficulty level is just slightly above the competence level

Cognitive/Student Model

- Q-Table with the relation between activities and the required competence level
- For a given exercise, the required competence level is:

$$q_i(a) = \sqrt[n]{\prod_{j=1}^{n} q_i(a_j)}$$

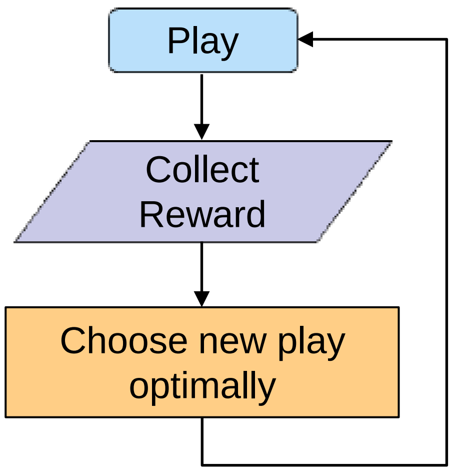
- If exercise correct : $r = q_i(a) - c_i^L$
- Update competence level (c_i^L) : $c_i^L = c_i^L + \alpha r$

- Expected learning progress per parameter: $w_i(a_i) \leftarrow \beta w_i(a_i) + \eta r$
- Exercises are chosen proportionally to w_i.

Multi-Armed Bandits

How to play to optimize the received reward

Expected reward is unknown and different for each machine



- Many algorithms that can simultaneously explore to estimate the return of each machine, and exploit to collect the maximum reward.
- RILRIT propse the activity more adapted to the student

Algorithm

Algorithm 2 Right Activity at Right Time (RiARiT)

Require: Set of n_c competences C
Require: Set of exercise parameters $A = \{A_1, \ldots, A_{n_a}\}$
Require: Set of n_a experts w_i

Initialize $c^L = 0, \ldots, m$
Initialize experts: $w_i(j) = \frac{1}{\#(A_i)}$
while *learning* **do**
  {Generate exercise}
  for $i = 1 \ldots n_a$ **do**
    $\tilde{w}_i = \frac{w_i}{\sum_j w_i(j)}$
    $p_i = \tilde{w}_i \xi_g + \gamma \xi_u$
    Sample a_i proportional to p_i
  end for
  Propose exercise $a = \{a_1, \ldots, a_{n_a}\}$
  Get Student Answer
  $C^L, r \leftarrow$ Update competence level
  {Update greedy expert}
  for $i = 1 \ldots n_a$ **do**
    $w_i(a_i) \leftarrow \beta w_i(a_i) + \eta r$
  end for
end while

How to define the cognitive model

Table 4

Q table that was used in the simulations and the user studies.

		KnowMoney	IntSum	IntDec	DecSum	DecDec	Memory
Exercise Type	1	0,7	0.4	0	0	0	0.5
	2	0,7	0.6	0.3	0	0	0.5
	3	0,7	0.7	0.6	0	0	0.5
	4	1	0.7	0.6	0.5	0.3	0.7
	5	1	0.9	0.7	0.7	0.5	0.7
	6	1	1	1	1	1	1
Price Present.	S	0.9	1	1	1	1	1
	W	1	1	1	1	1	0.6
	S&W	0.8	1	1	1	1	0.2
Cents Not.	x.x€	0.8	1	1	1	1	1
	x€x	0.9	1	1	1	1	1
Money Type	Real	1	-	-	0.9	0.9	1
	Token	0.1	-	-	1	1	1

Pedagogycal Restrictions

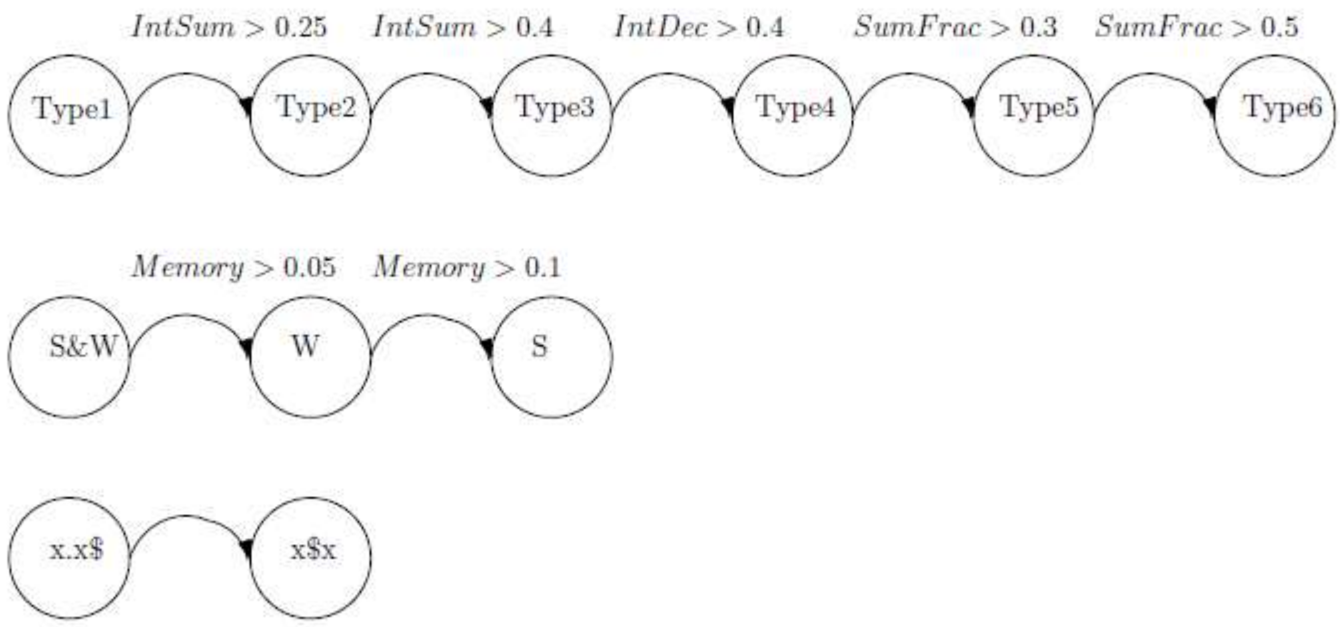


Fig.14. Pre-requisites graph

How to use

List of **competences** to be acquired

↓

Definition of the **activity types** and their **parameterization**

↓

Definition of **Pedagogical Constraints** and **Cues**
Restrictions on the parameters automatically selected.
List of errors and the aid to be given

↓

Creation of the **Cognitive Model**
Relation between the parameter of the activities and the minimum required competences' level

↓

After each activity, **estimate student's competence**.
Propose new activity based on the competence level.

↓

Individual report: detailed results, level of knowledge acquisition, personal difficulties/strengths

The activities can be very different: interactive exercises, animations/videos, …

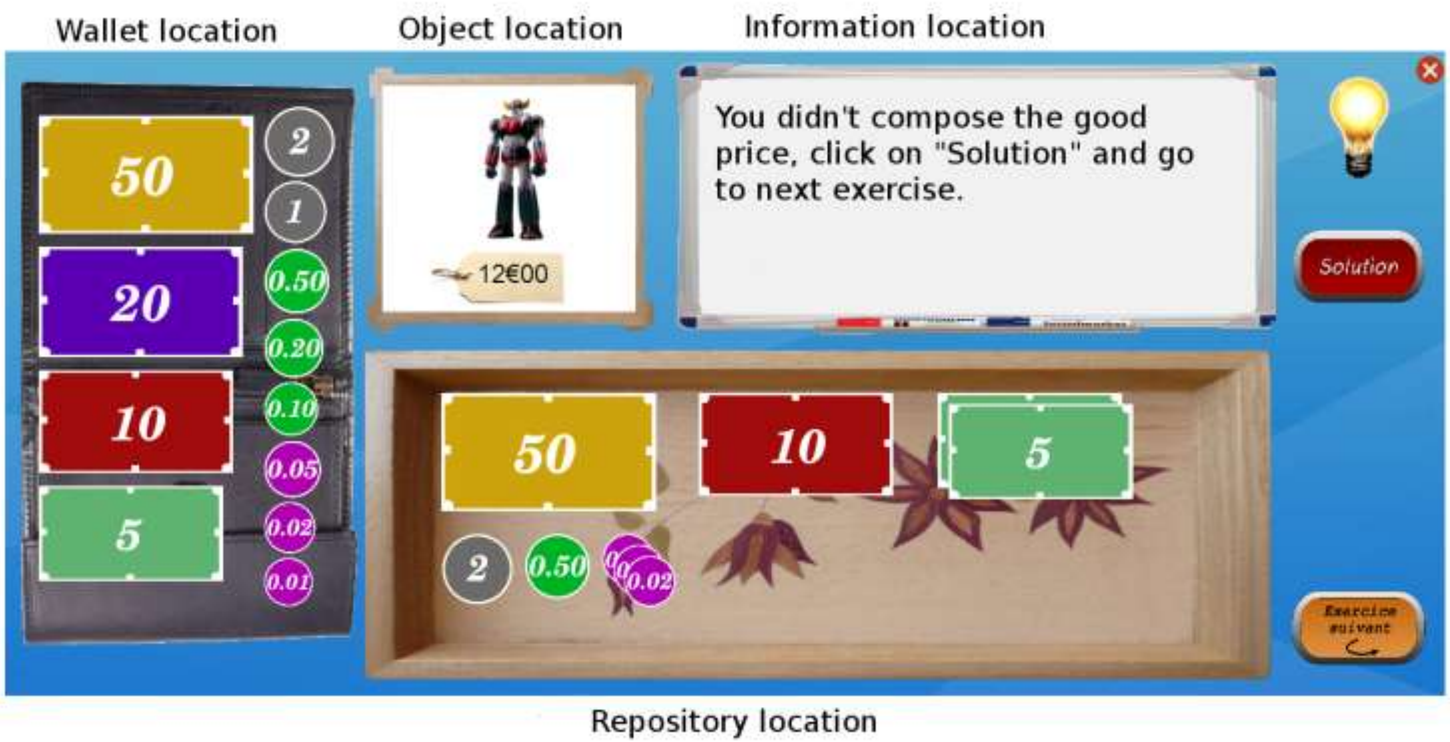
Other Possible Optimizations

- Bootstrap optimization based on pre-knowledge about the student
- Create student profiles to share information among students
- Use biometric information: attention, concentration, …

Money Game

Two experiments

- With simulated students
- With real students (CE1) level in the Bordeaux region



Competences

- Know the money
- Sum/Subtract and decompose integers
- Sum/Subtract and decompose fractional numbers
- Optimal decomposition
- Memory

Parameters

- P1 : Price complexity

ND = {0 ; 1 ; 2 ; 5} ; ND* = {1 ; 2 ; 5} : Valeurs à lecture directe.

NC = {3 ; 4 ; 6 ; 7 ; 8 ; 9} : Valeurs à composer.

Niveaux	Niveau 1	Niveau 2	Niveau 3	Niveau 4	Niveau 5	Niveau 6
Décomposition	$a \in N_{D^*}$ $b \in N_D$ $c = d = 0$	$a \in N_{D^*}$ $b \in N_C$ $c = d = 0$	$a \in N_C$ $b \in N_C$ $c = d = 0$	$a \in N_D$ $b \in N_D$ $c \in N_{D^*}$ $d \in N_D$	$a \in N_C$ $b \in N_C$ $c \in N_{D^*}$ $d \in N_D$	$a \in N_C$ $b \in N_C$ $c \in N_C$ $d \in N_C$
exemple	10	18	39	10.25	39.15	98.97

- P2 : Real and monopoly money
- P3 : Two different representation of decimals
- P4 : Price written or spoken

Séquence prédéfinie

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
lvl	1	2	2	3	3	4	4	4	5	6
O&E	OE	OE	O	OE	O	OE	OE	E	E	E
Pres	x€x	x€x	x€x	x€x	x€x	x€x	x,x€	x,x€	x,x€	x,x€
M&J	M	M	M	M	M	MJ	MJ	MJ	MJ	MJ

Virtual Students

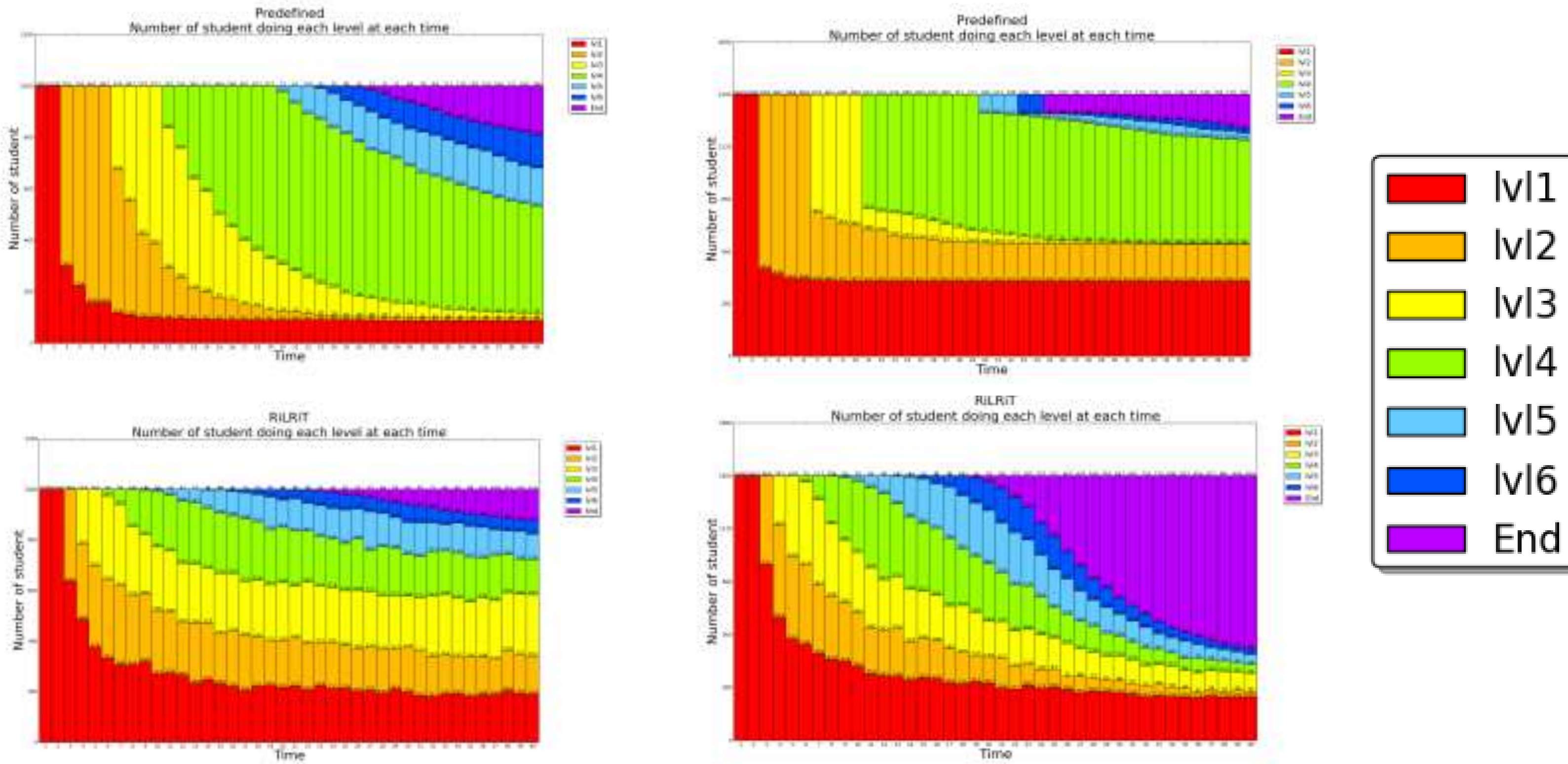
- «Q»
  - Modeled by specific levels of competence per KC
  - Probability of answering right depends on the difference between competence level and required competence
- «P»
  - Modeled by specific comprehension levels of each parameter
  - Probability of answering right depends on the difference between competence level and required competence, and the level of understanding of each parameter

Evolution of complexity of decomposition

Modèle Q | Modèle P

PredSeq

RiLRiT



RilRit propose more difficult exercises earlier on, but keeps proposing simpler exercises longer. This shows an adaptation to the difficulties of particular students.

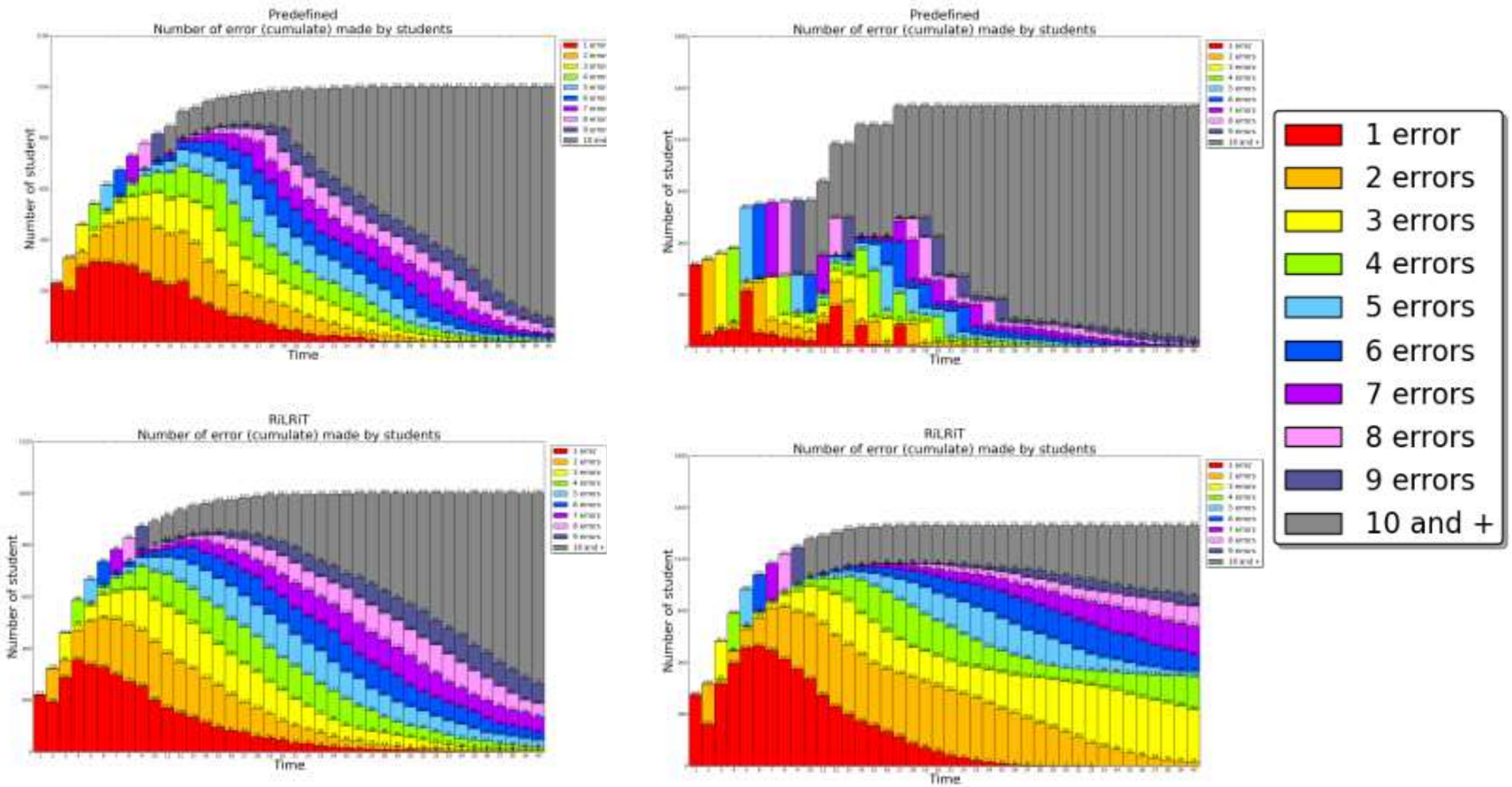
Number of errors

Q

P

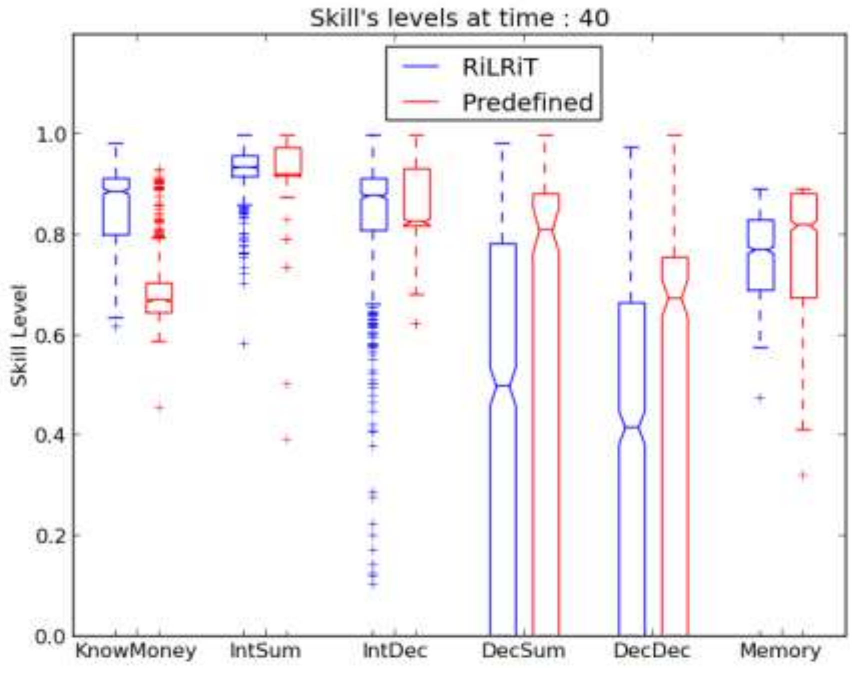
PredSeq

RiLRiT

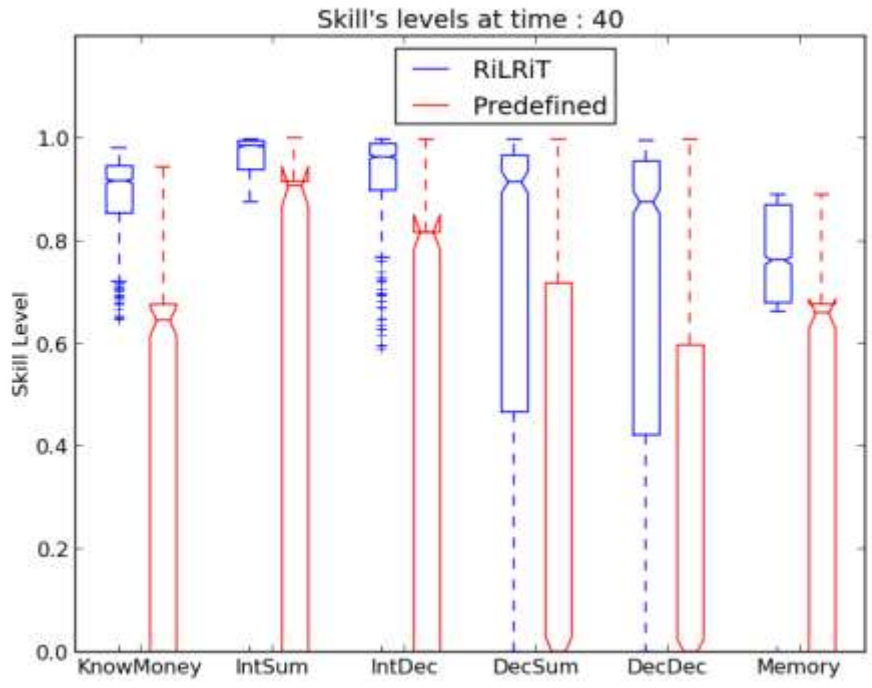


Competences' Level

Q



P

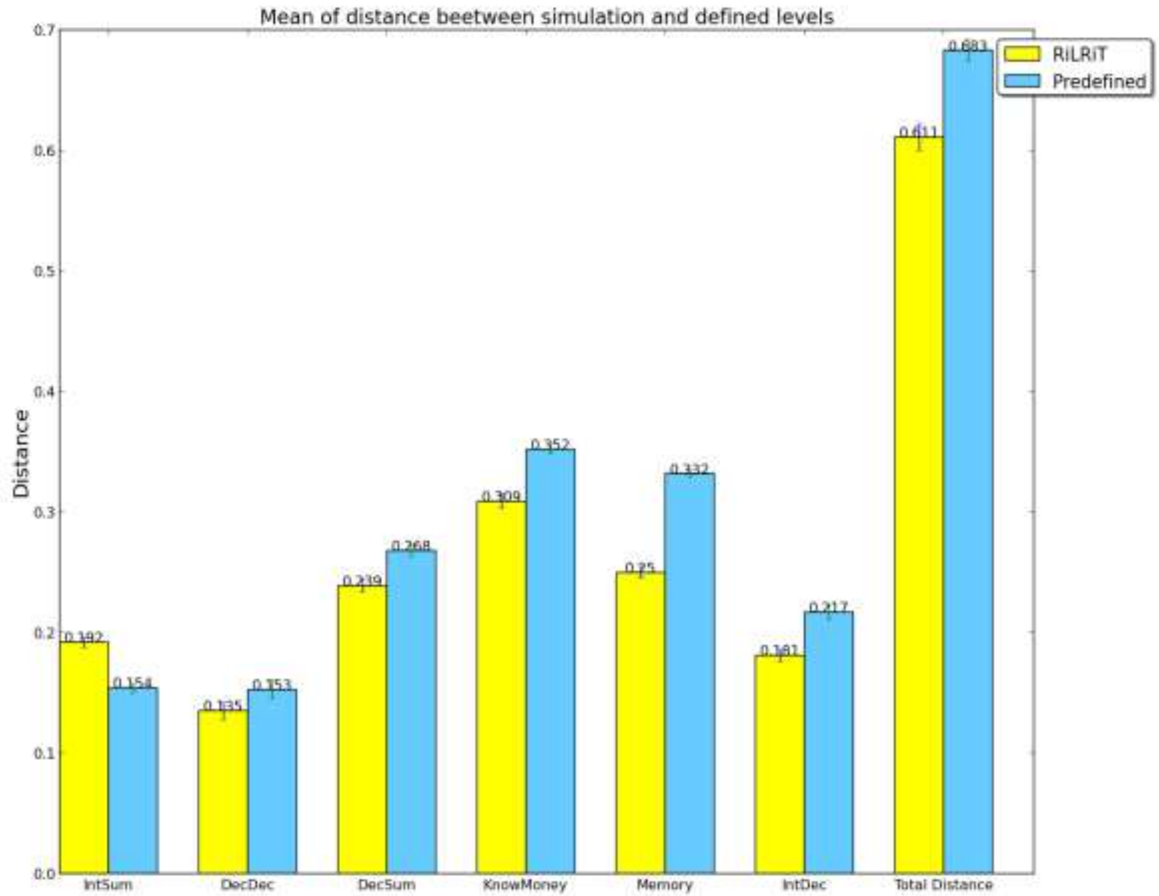


Not significantly different in population Q.

Big difference in population P.

Difference between real and estimated level

- RilRit estimates better the level of students

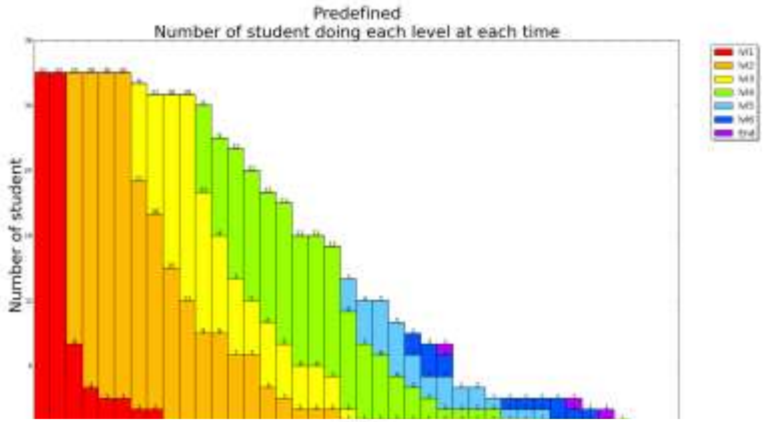


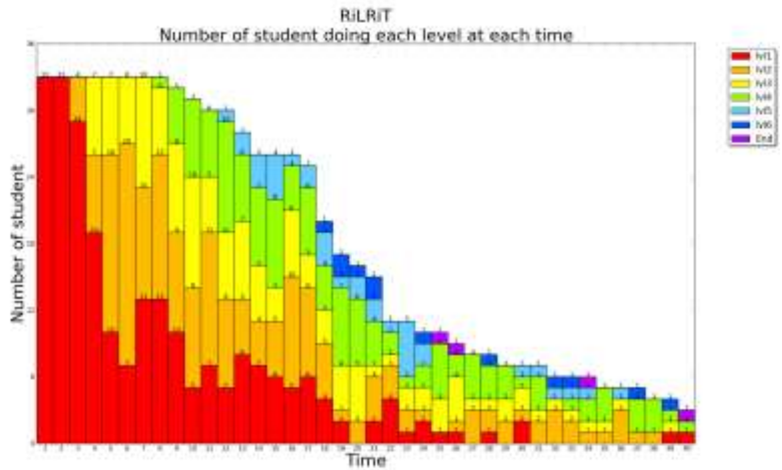
User Studies

- Experiments:
  - 5 different schools, 130 students (CE1)
  - Use of the computers of schools
  - 35 minutes per student
    => each student does a different number of exercises
- Observations:
  - Bad informatic infrastructure in most schools
  - 66 students with reliable data
  - Good participation and engament by the students and the teachers

Results

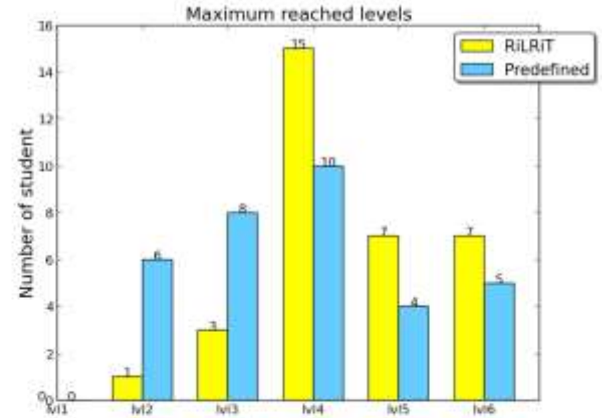
Level

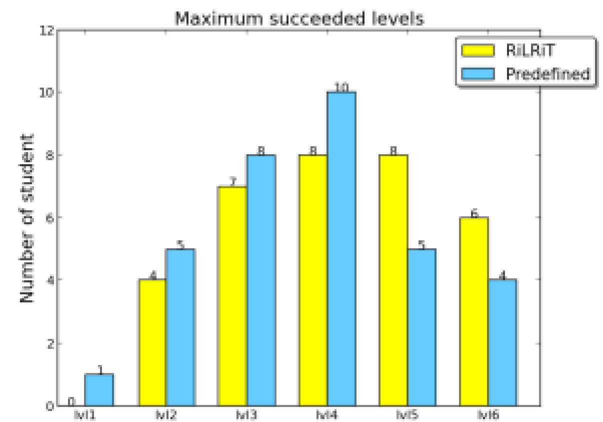




The most difficult exercises are proposed sooner

Maximum level suceeded





More student get and solve most difficult exercises

Conclusions

- In general the optimized sequences are better adapted to each particular student
- Faster learning
- Better estimation of students level
- Easier to develop and distribute than a hand-made sequence, and robust to design errors

- In general students are very motivated to play these games, good reception by the teachers.

Manuel LOPES, Didier ROY, Benjamin CLÉMENT, Pierre-Yves OUDEYER

flowers.inria.fr manuel.lopes @inria.fr