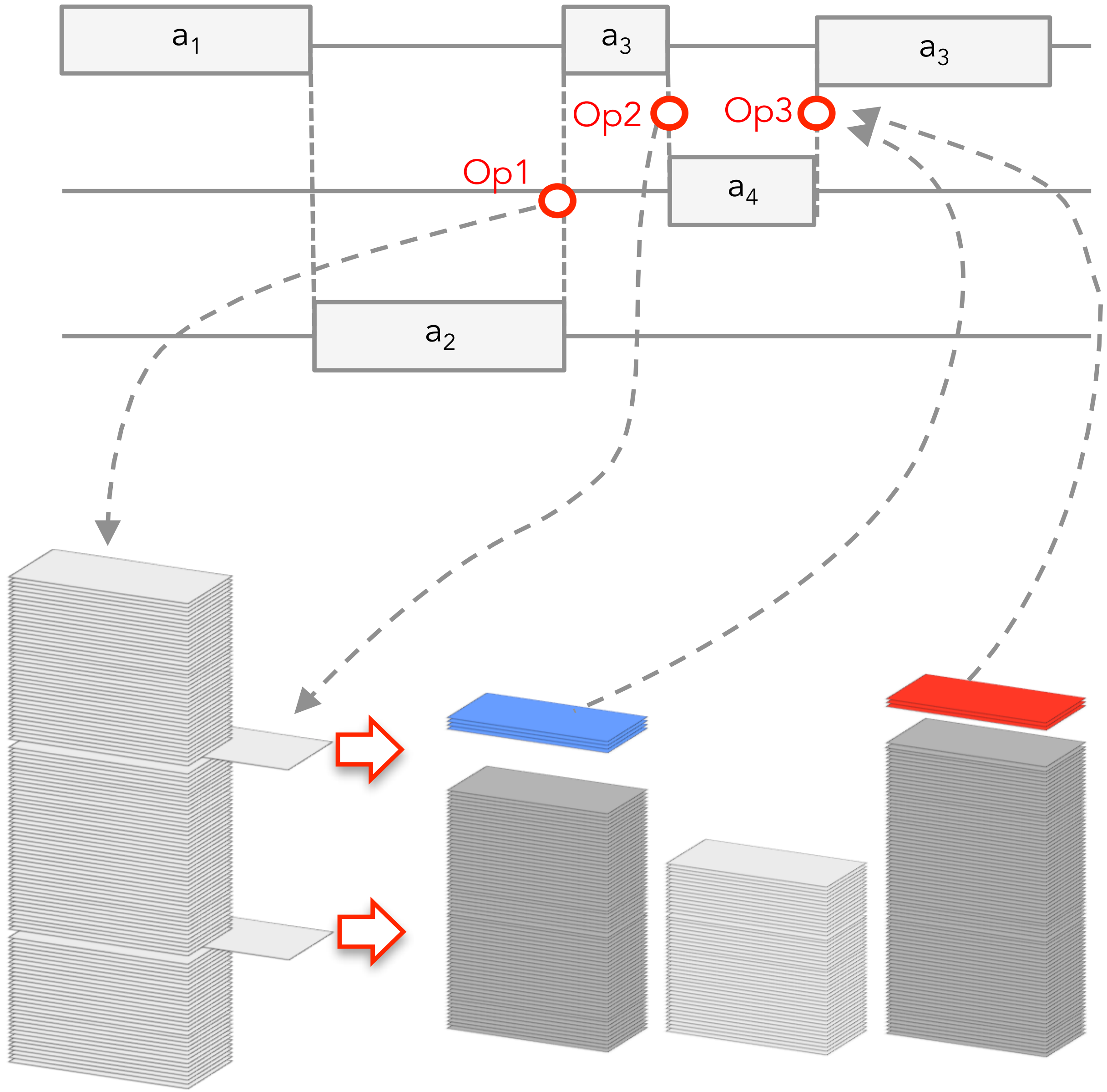
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-411 : Digital Education & Learning Analytics

# Chapter 10: The operator library

# (Educational Workflows)

Pierre Dillenbourg and Patrick Jermann , Łukasz Kidziński



$a_1$
$a_3$
$a_3$
Op2
Op3
$a_4$
Op1
$a_2$


Workflows have been designed for automating bureaucratic processes such as processing insurance claims.

Applied to education, they support scaling up rich pedagogical scenario

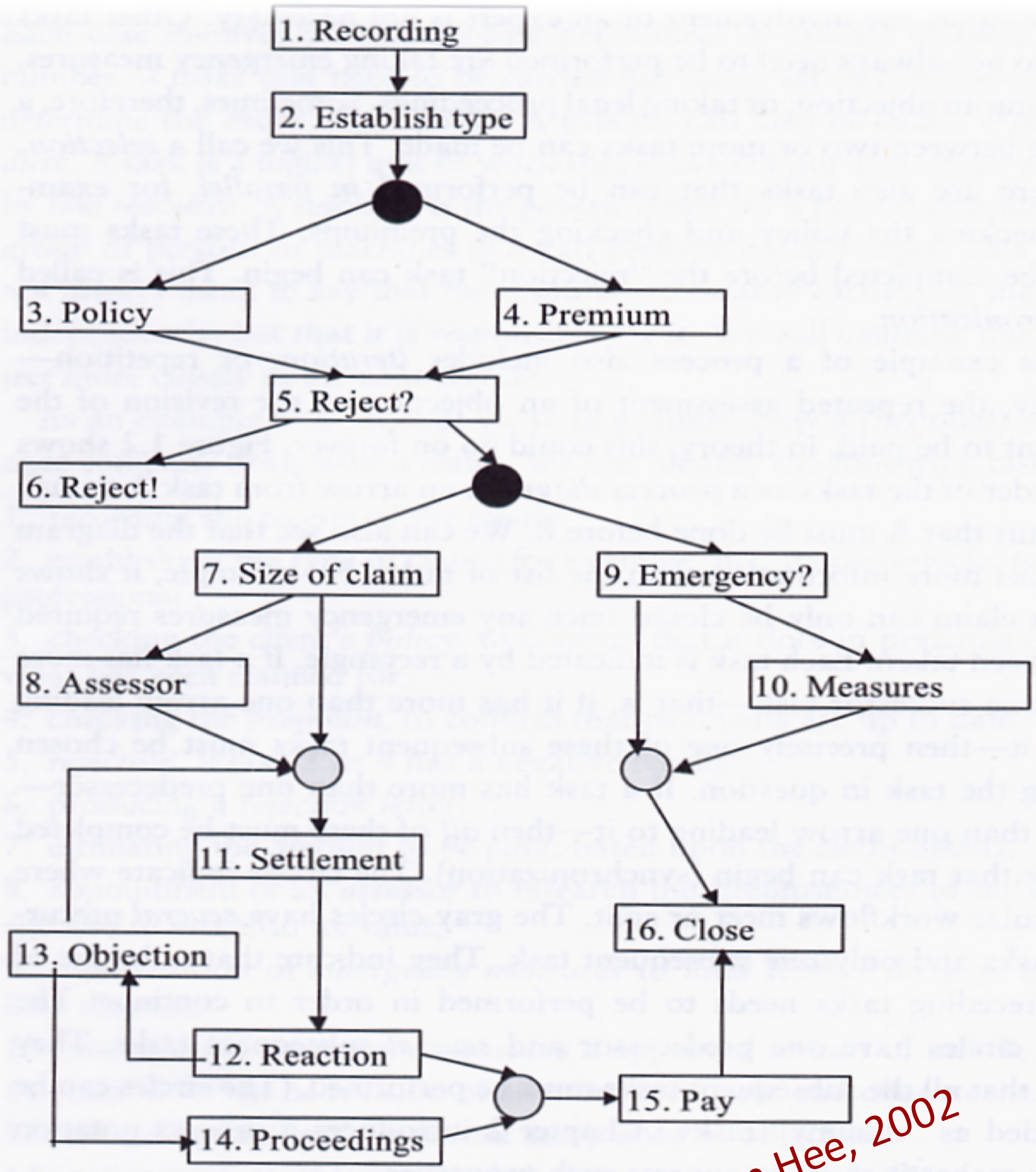


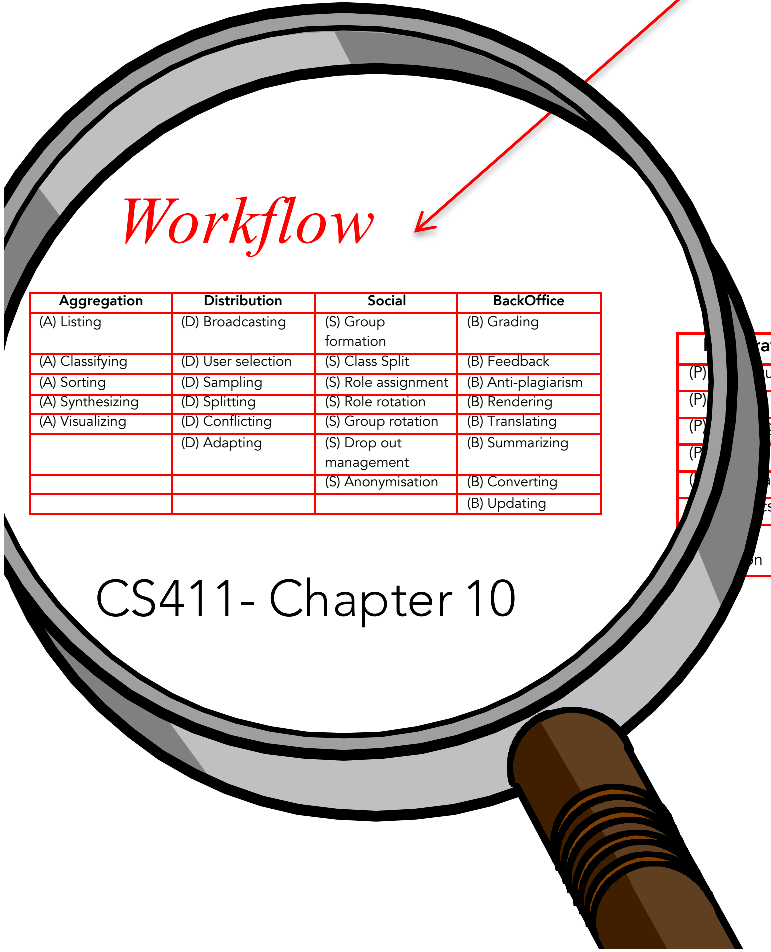

**Figure 1.2**
Insurance claim process

Van der Aalst & van Hee, 2002

An orchestration graph $G = (V, E)$ where $E = V \times V$

$V = \{a_i\} \mid a_i: t^s, t^e, \pi$, object, product, $\{c\}$, traces, $\{metadata\}$

$E = \{e_{ij}\} \mid e_{ij}: (a_i, a_j,$ {**operators**}, {controls}, label, weight, elasticity)



*Workflow*

| Aggregation | Distribution | Social | BackOffice |
|---|---|---|---|
| (A) Listing | (D) Broadcasting | (S) Group formation | (B) Grading |
| (A) Classifying | (D) User selection | (S) Class Split | (B) Feedback |
| (A) Sorting | (D) Sampling | (S) Role assignment | (B) Anti-plagiarism |
| (A) Synthesizing | (D) Splitting | (S) Role rotation | (B) Rendering |
| (A) Visualizing | (D) Conflicting | (S) Group rotation | (B) Translating |
| | (D) Adapting | (S) Drop out management | (B) Summarizing |
| | | (S) Anonymisation | (B) Converting |
| | | | (B) Updating |

CS411- Chapter 10

*Pedagogical idea*

| [illegible]ration | Set | Translation | Generalization |
|---|---|---|---|
| (P) [illegible]uisite | (S+) Aggregation | (T) Proceduralisation | (G+) Induction |
| (P) [illegible] | (S+) Expansion | (T) Elicitation | (G+) Deduction |
| (P[illegible]ganizer | (S-) Decomposition | (T) Alternate | (G+) Extraction |
| (P[illegible]on | (S-) Selection | (T) Re-Frame | (G+) Synthesis |
| ([illegible]tion | (S=) Juxtaposition | (T) Reverse | (G=) Analogy |
| [illegible]cs | (S=) Contrast | (T) Repair | (G=) Transfer |
| [illegible]n | (S=) Identity | (T) Teach | (G-) Restriction |

CS411- Chapter 9

*Stochastic model*

CS411- Chapter 11

CS411- Chapter 12

# Library of Edge Operators

## How data collected in $a_i$ are processed for $a_j$ ?

Aggregation operators gather data for subsequent activities, generally located on a higher plane

Distribution operators split data for subsequent activities, generally located on a lower plane

Social operators modify the social structure of activities. They rely on social distance criteria

Back-office operators enrich data with external information, including information manually provided by human actors

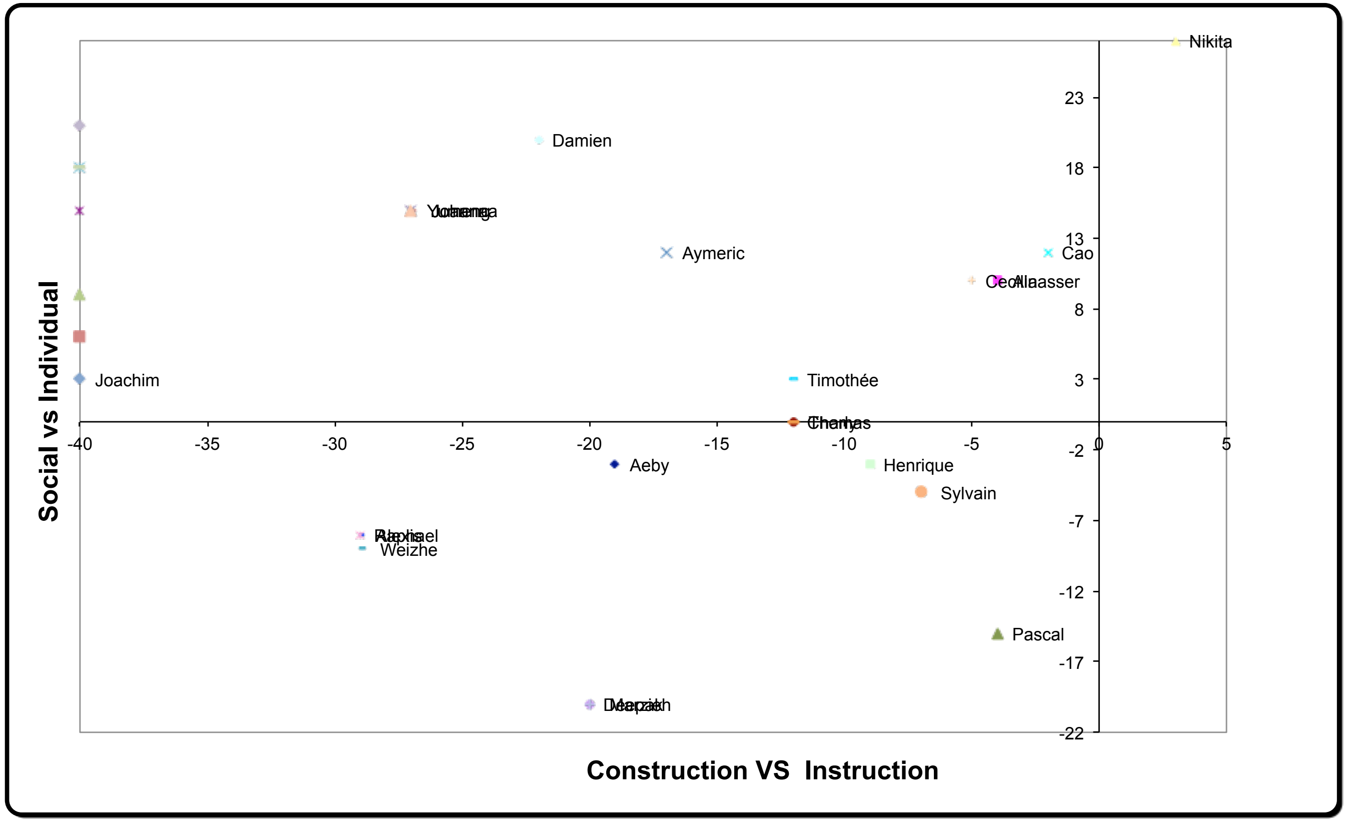
# Library of Graph Operators

| Aggregation | Distribution | Social | BackOffice |
|---|---|---|---|
| (A) Listing | (D) Broadcasting | (S) Group formation | (B) Grading |
| (A) Classifying | (D) User selection | (S) Class Split | (B) Feedback |
| (A) Sorting | (D) Sampling | (S) Role assignment | (B) Anti-plagiarism |
| (A) Synthesizing | (D) Splitting | (S) Role rotation | (B) Rendering |
| (A) Visualizing | (D) Conflicting | (S) Group rotation | (B) Translating |
| | (D) Adapting | (S) Drop out management | (B) Summarizing |
| | | (S) Anonymisation | (B) Converting |
| | | | (B) Updating |

# Operators

## Aggregation

- Listing
- Classifying
- Sorting
- Synthesizing
- Visualizing

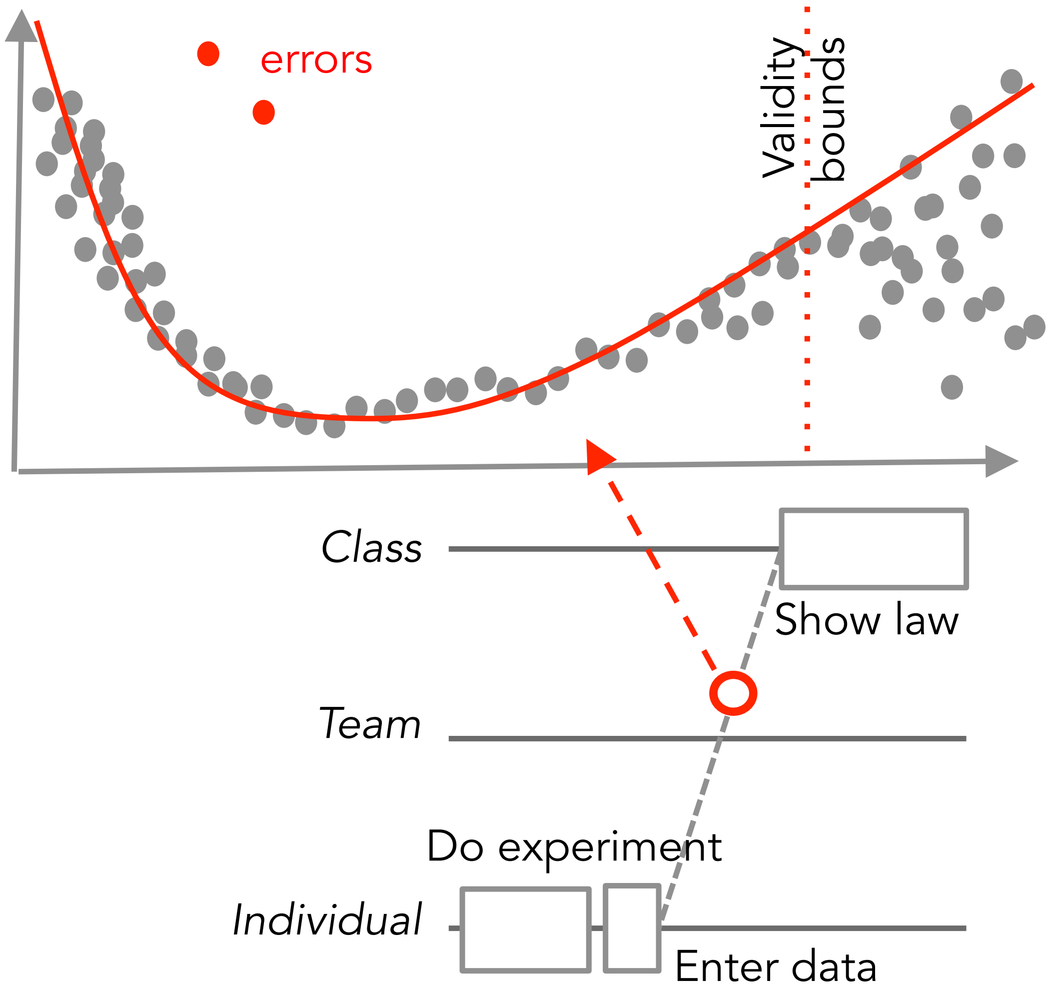



Agumentation Scenario: Opinions collected in $a_1$ are aggregated and visualized as an opinion map to be used in $a_2$.

# Operators

**Aggregation**

- Listing
- Classifying
- Sorting
- Synthesizing
- Visualizing




In a physics MOOC, students have to take an egg, weigh it, and drop it from an altitude of between 100 and 200 centimeters. When the egg lands, they measure the distance between the splashes that are the furthest away from each other. Each student enters the values of the weight, altitude, and distance after impact. The system produces graphs where every experiment appears as a dot. The curve shows the behavior predicted by the theory. The teacher points out which data are measurement errors (red dots) and those poorly explained by the scientific model (on the right of the dotted red line).

**Aggregation**

- Listing
- Classifying
- Sorting
- Synthesizing
- Visualizing

# Design Recommendations

(1) The features of the visualization influence what information students have to process in the next activity, what they will comment on, discuss, or discover, as well as what the teacher will be able to point out in a subsequent debriefing lecture.

The visualization has to be designed with this didactic purpose in mind, that is, how to pedagogically exploit the graphical representation in the next activity, not just for the sake of producing fancy visualizations.

# Operators

**Aggregation**

- Listing
- Classifying
- Sorting
- Synthesizing
- Visualizing

# Design Recommendations

(2) Students are especially engaged when their own data are visualized. These can be the products/traces they produced in previous activities: "*my*" answers, "*my*" comments, "*my*" products, and so on.

It would be politically correct to suggest making data (semi-) anonymous here but this kill the effect. Solutions : replacing a student's name with a pseudo , designing the interface so that the student can see his own name, but not the name of his peers…

**Aggregation**

- Listing
- Classifying
- Sorting
- Synthesizing
- Visualizing

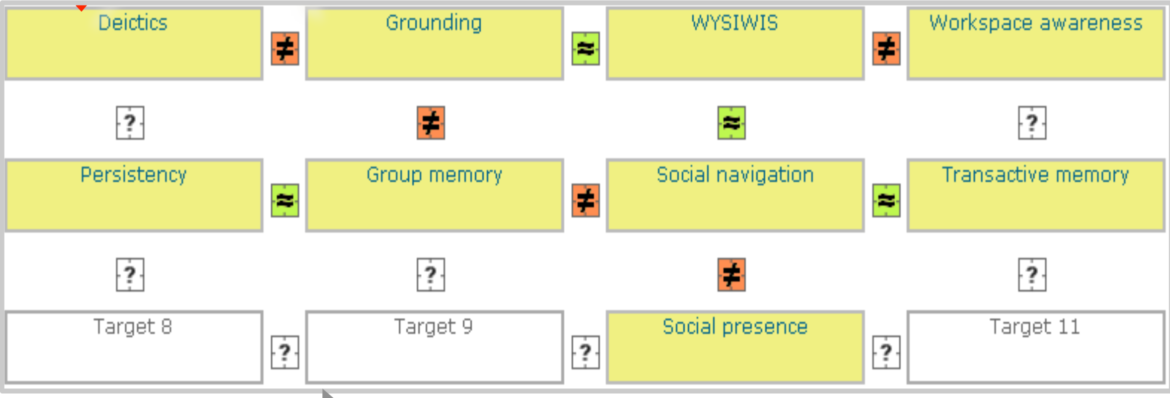
# Design Recommendations

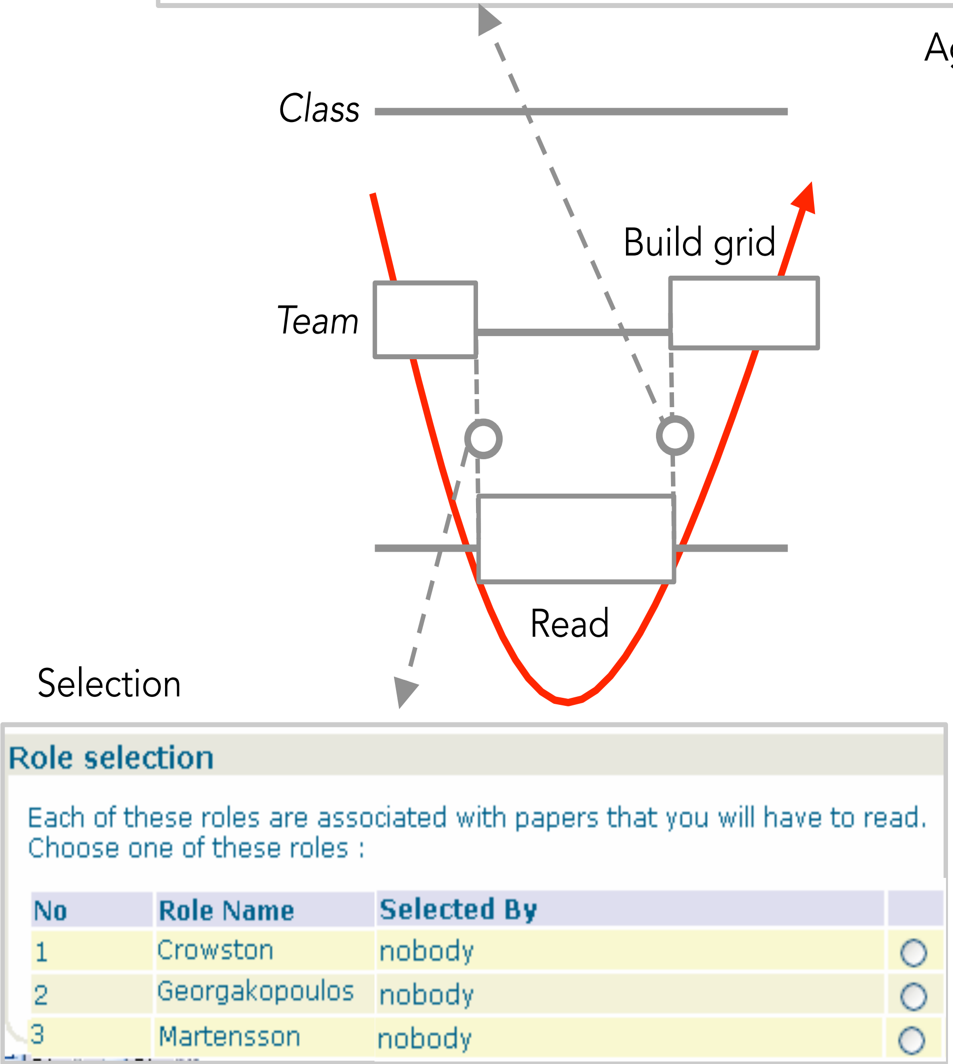
(3) An aggregation operator enables powerful activities when a differentiation operator is used in the previous activity (to be developed hereafter)

# Operators

**Distribution**

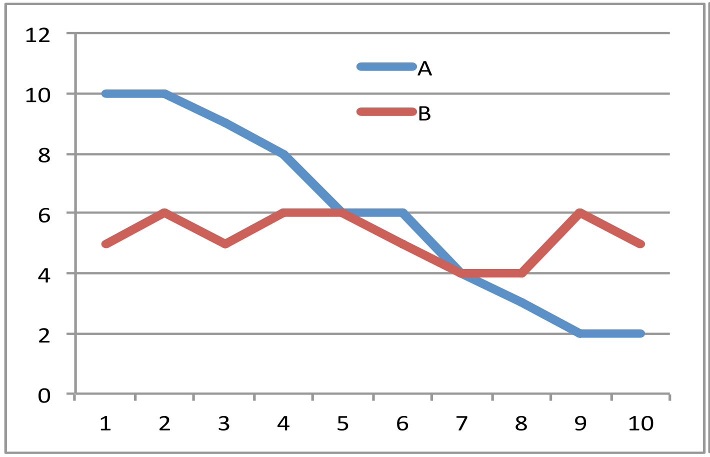
- Broadcasting → delivers the same data to all learners performing $a_j$.
- User selection → users choose which a subset of data for $a_j$
- Sampling → assigns a different subset to individuals / teams for $a_j$
- Splitting → assigns a different subset of data to each individual within a team for $a_j$ *(so called "jigsaw" graph)*
- Conflicting → assigns conflicting subsets of data to individuals within a team for $a_j$
- Adapting → chooses the most relevant material for an individual or a team in $a_j$.



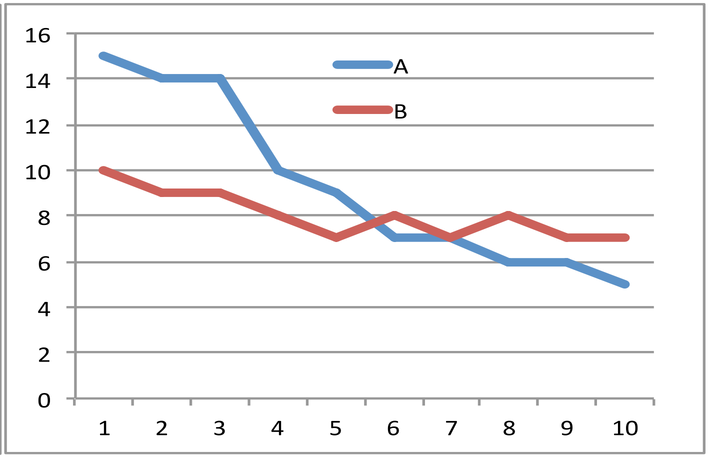




# A pattern:
# Distribute+Aggregate

The "ConceptGrid" graph. Each team has to build a concept grid—a sort of concept map. Each team is composed of several roles (the number of roles can be determined by the teacher) and each role necessitates reading several papers (the number of papers can be determined by the teacher) that correspond to the selected role. Typically, a student will play the role "Piaget" by reading papers from Piaget. Each student selects a role that has not yet been selected by another team member, and the system simply distributes readings assigned to each role. Then, when each student has learned about a subset of concepts, the team has to build a grid in such a way that students can define (text entry) the relationship between two grid neighbor concepts. The way in which concepts are distributed among team members will determine who explains which concepts to whom in the grid construction activity.

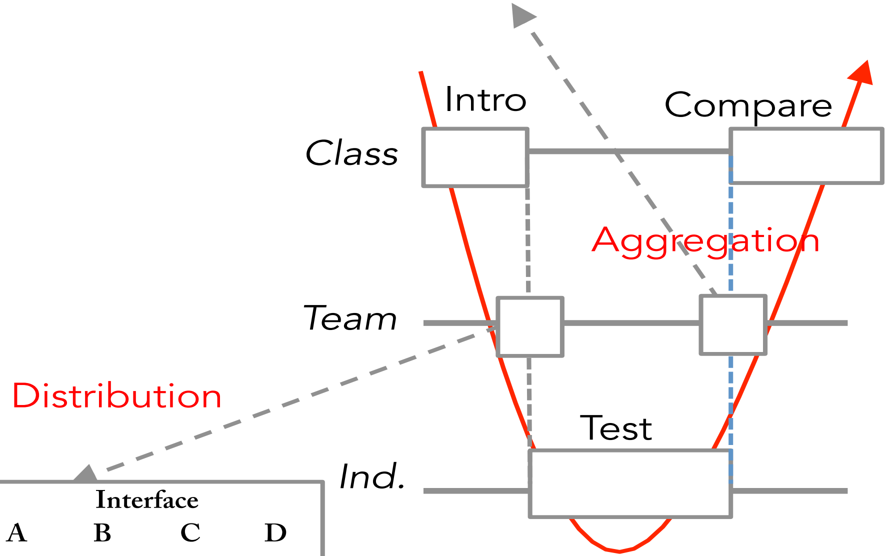



Errors




Speed

# A pattern: Distribute+ Aggregate

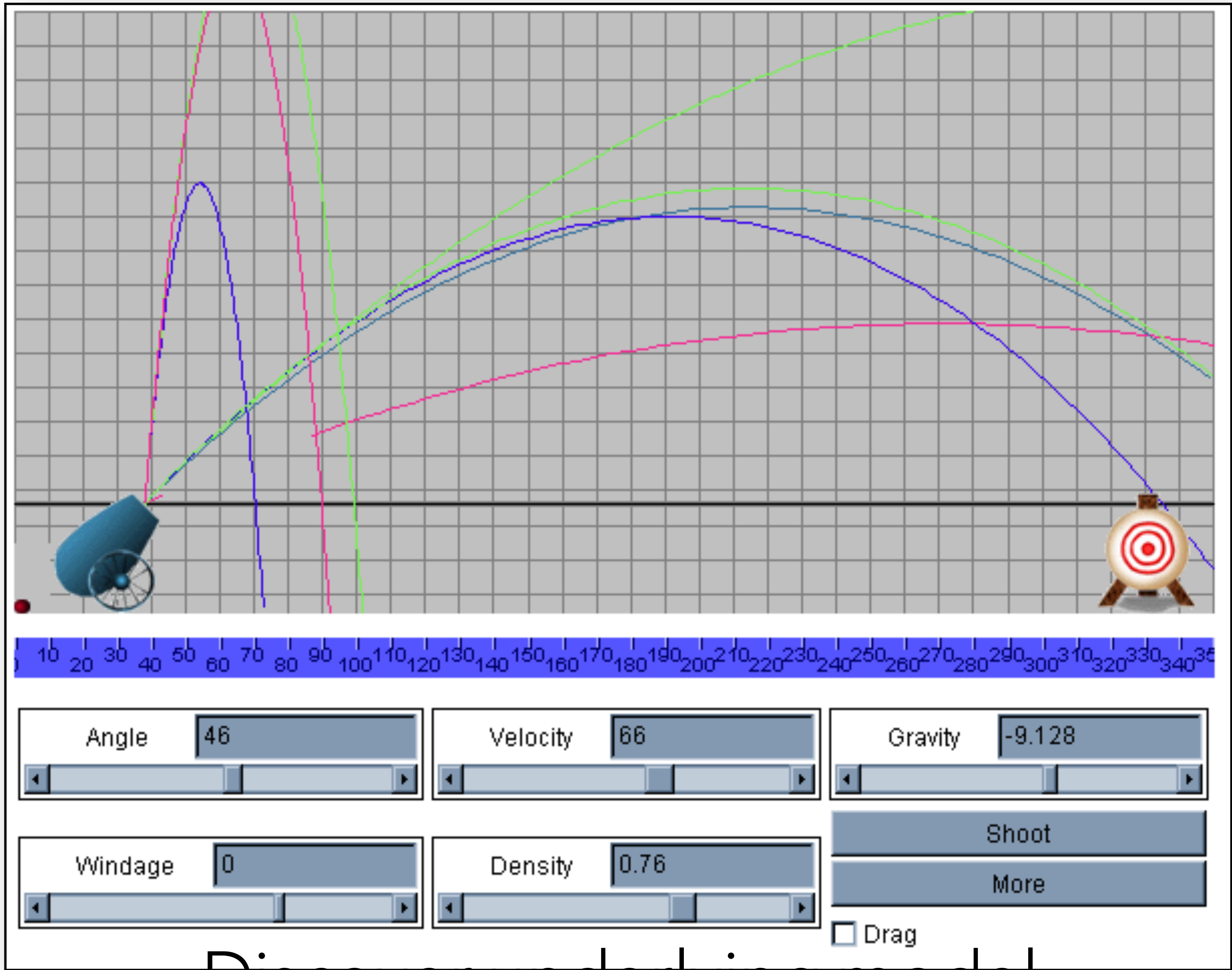



| | Interface | | | |
|---|---|---|---|---|
| **Students** | **A** | **B** | **C** | **D** |
| **1** | 1 | 1 | 0 | 0 |
| **2** | 0 | 1 | 1 | 0 |
| **3** | 0 | 0 | 1 | 1 |
| **4** | 1 | 0 | 1 | 0 |
| **5** | 1 | 0 | 0 | 1 |
| **6** | 0 | 1 | 0 | 1 |
| **7** | 1 | 1 | 0 | 0 |
| **8** | 0 | 1 | 1 | 0 |
| **9** | 0 | 0 | 1 | 1 |
| **10** | 1 | 0 | 1 | 0 |
| **11** | 1 | 0 | 0 | 1 |
| **12** | 0 | 1 | 0 | 1 |
| | *6* | *6* | *6* | *6* |

An HCI-Course Scenario. The teacher proposes 4 versions of a website in which users order train tickets. Each of the 10,000 students has to order 5 fake tickets with two of the four versions of the website and then fill in a usability questionnaire. The system distributes interfaces to students in such a way that (1) all interfaces are tested by the same number of students, and (2) 50% of the students test *A* before *B* and 50% the other way around. The aggregation operator produces a comparison of the task completion time and the number of errors on each interface. It creates contrasted graphs, where we can see that interface *B* generates fewer mistakes at the beginning than *A*, but that the error rate decreases faster with *A*.

From Chapter 5

# Learning from Simulations




Discover underlying model

From Chapter 5

1. (Raise a question)
2. Generate an hypothesis
3. Design an experiment
4. Run/simulate the experiment
5. Interpret results

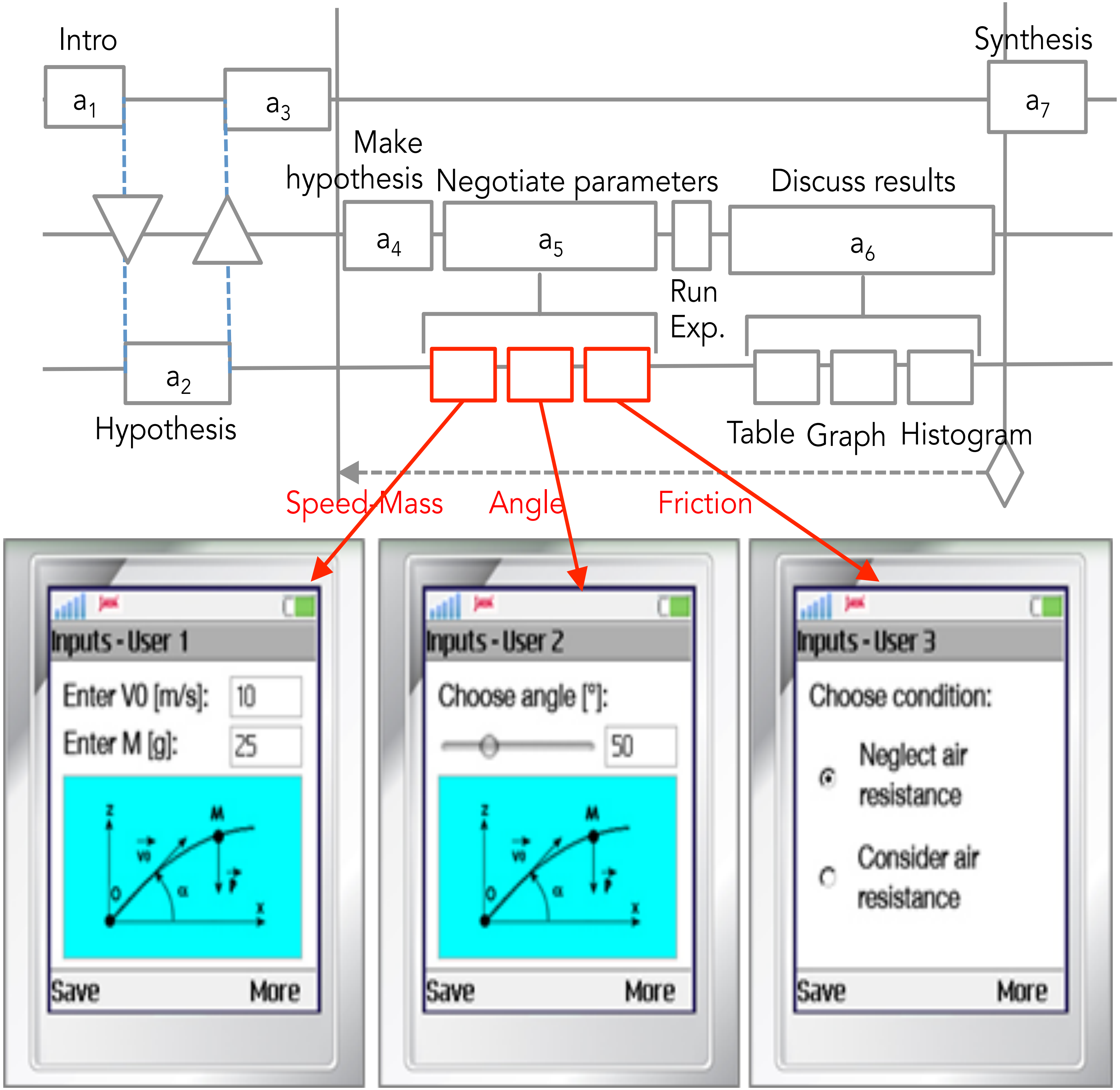
# Hypothetico-deductive reasoning

From Chapter 5

# But…

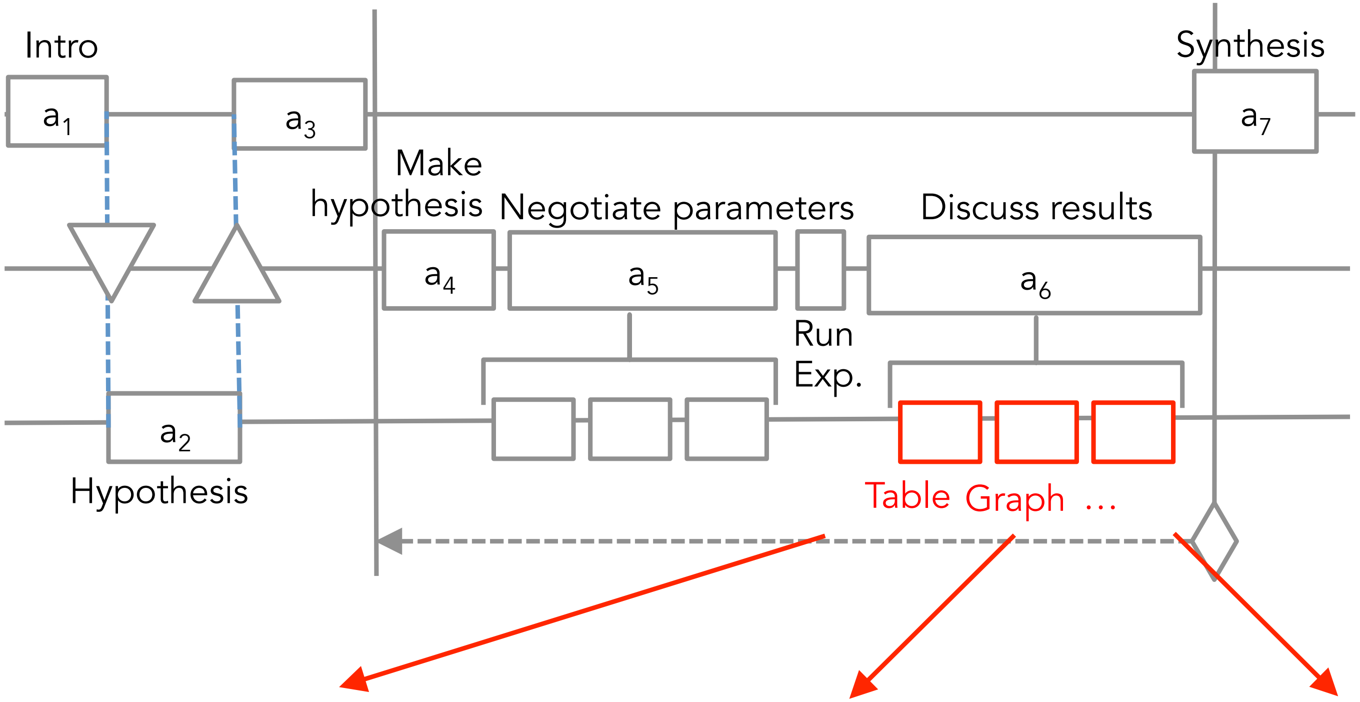
1. Question
2. Hypothesis
3. Design
4. Run
5. Interpret

- No clear hypothesis is formulated or badly formulated (42% ), i.e. no reationship between variables
- Design unconclusive experiments,  students vary several parameters at at time
- Confirmation bias: to design experience that confirm the hypothesis
- 35% to 63% errors in data interpretration and graphics readings

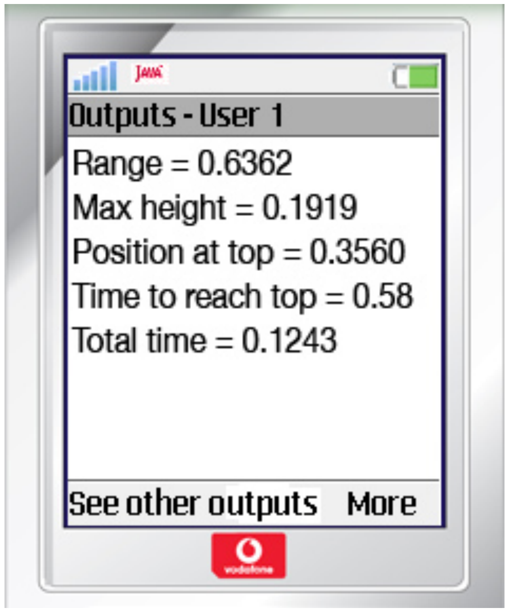


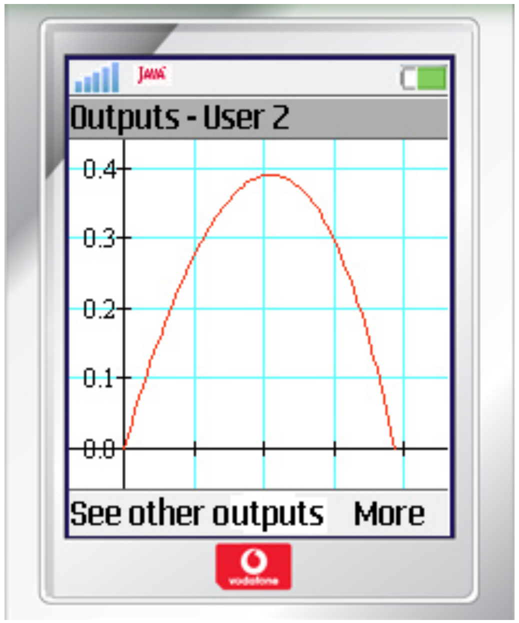

# Split Where Interaction Should Happen

The differences created among team members determine how they will interact in a collaborative task in order to reach a shared solution despite their differences



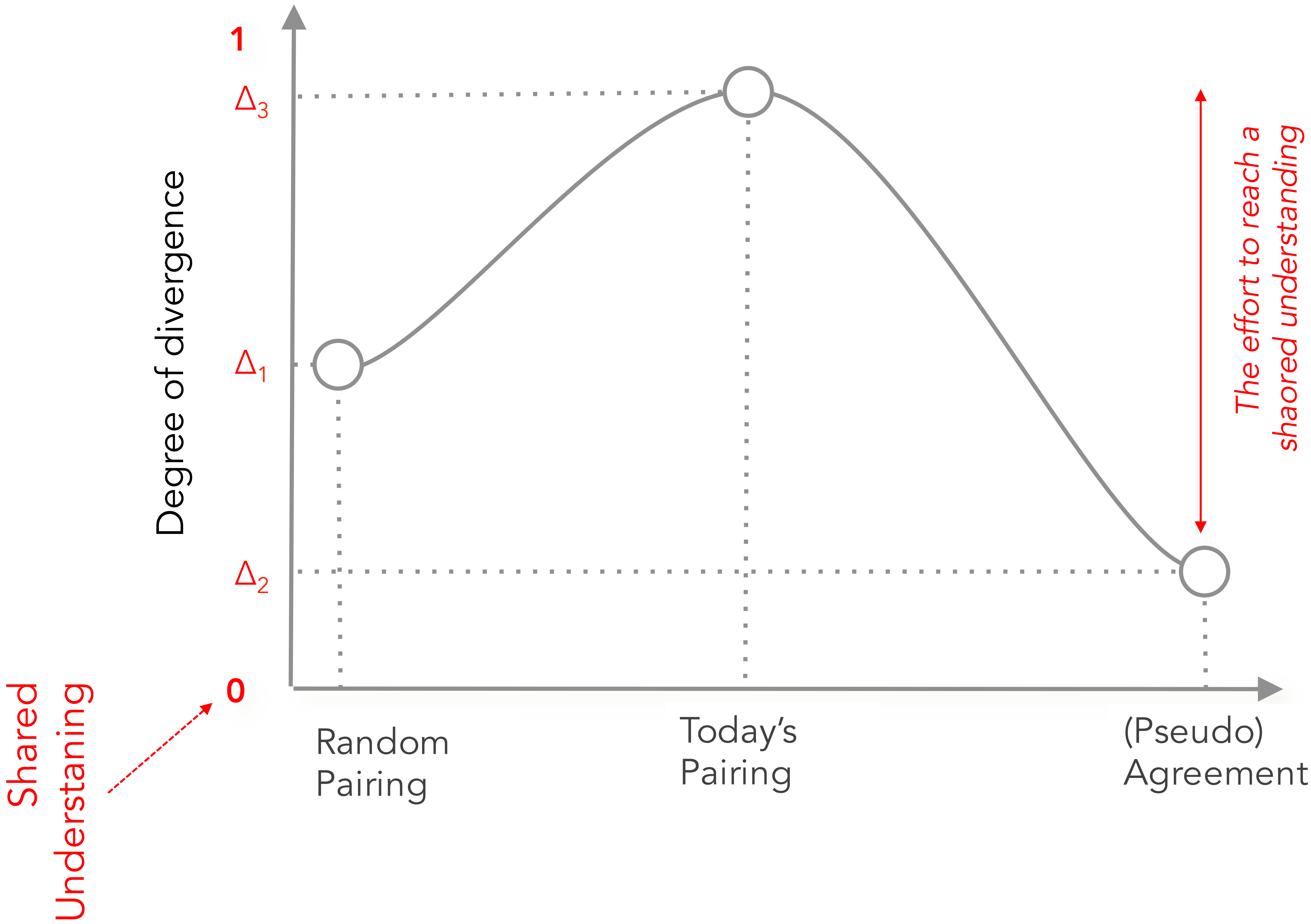

# Split Where Interaction Should Happen




The differences created among team members determine how they will interact in a collaborative task in order to reach a shared solution despite their differences

# Split Where Interaction Should Happen

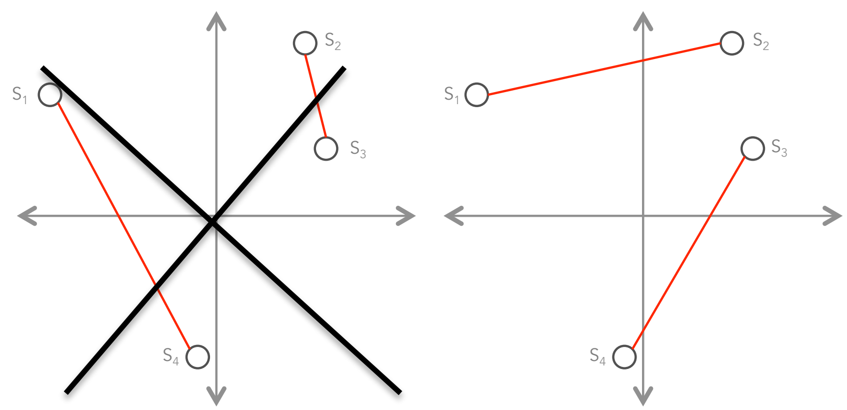


# Operators

## Social Operators

- Group formation → (group size, distance-criterion, min/max)
- Role assignement
- Role rotation
- Group rotatiom
- Class split
- DropOut Mgt
- Anonymisation

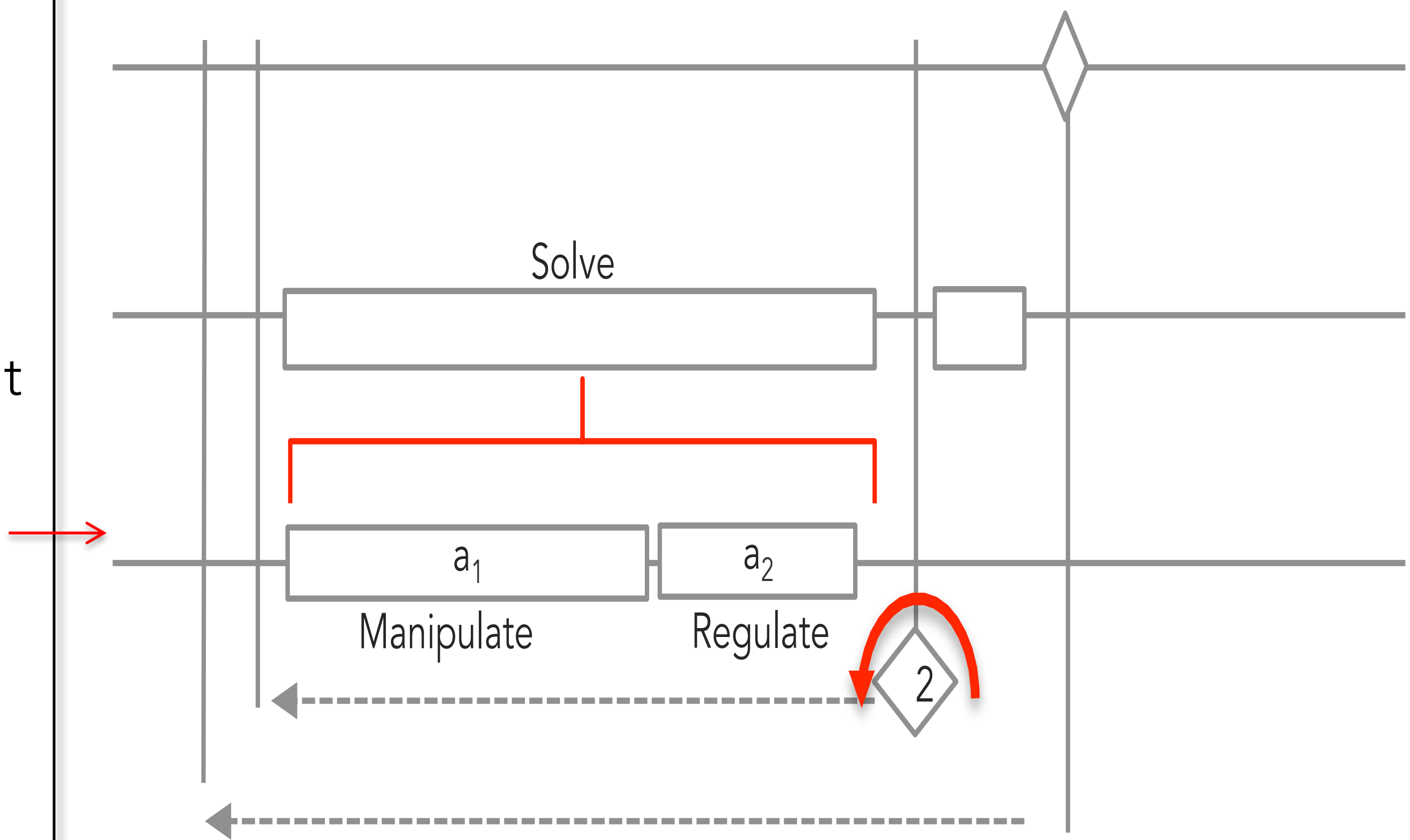
- Level (e.g score at pre-test)
- Knowledge type (e.gh quantitative /qualitative)
- Background (e.g. CS / Education)
- Optinion (as we did a few weeks ago)
- Geography (e.g. Urban vs country)
- TimeZone
- Friendship



# Operators

## Social Operators

- Group formation
- Role assignement
- Role rotation
- Group rotatiom
- Class split
- DropOut Mgt
- Anonymisation



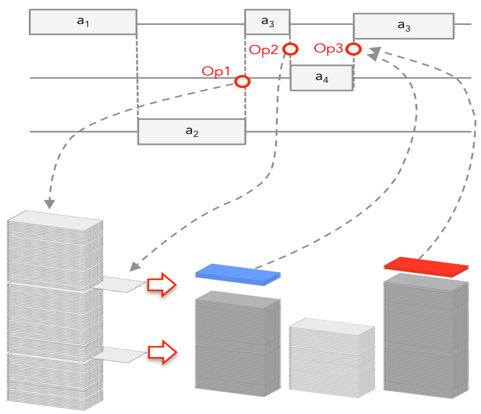

The reciprocal tutoring graph illustrates the mutual regulation pattern, which is relevant for problem-solving tasks that require heuristic knowledge. In this graph, learner $s_1$ reads a paragraph aloud, after which, learner $s_2$ asks him comprehension questions. These two roles are switched at each paragraph. The goal is the acquisition of comprehension monitoring skills.

# Operators

**« Back Office »**

- Grading
- Feedback
- Anti-plagiarims
- Translating
- Updating
- Converting
- Summarizing
- Rendering

See PDF




# Orchestration Graphs

1. Home-made model, not an established theory
2. Modeling rich pedagogical scenarios in order to bring them at scale by using operators
3. Pedagogy is hidden inside technology, e.g. changing an operator changes the pedagogical idea
4. A model is a simplification of the reality; this model does not capture the affective side of learning
5. The do not only apply to learning technologies, but to any situation