

Interfaces Leading Groups of Learners to make their shared Problem-Solving Organization Explicit

Patrice Moguel, Pierre Tchounikine, and André Tricot

Abstract— In this article, we consider collective problem-solving challenges and a particular structuring objective: lead groups of learners to make their shared problem-solving organization explicit. Such an objective may be considered as a way to lead learners to consider building and maintaining a shared organization, and/or as a way to provide a basis for monitoring learners' organizational process. Work presented in this article highlights the interest of Bardram's model of collective work to inspire design principles of interfaces supporting learners' in making their organization explicit. Of particular importance are the facts that organization is somewhat reified (here, as an array), organization and execution interfaces are similar and connected, and learners are given flexibility in enacting partial organizations or adapting their constructions in context.

Index Terms— Collaborative learning, groupware, problem solving, interfaces.

1 INTRODUCTION

Research in Computer Supported Collaborative Learning (CSCL) has shown that collaboration leads to positive outcomes when learners engage in knowledge-generative interactions such as explanation, argumentation, negotiation, conflict-resolution or mutual regulation [1]-[3]. To improve the likelihood of such interactions, two non-exclusive approaches have developed. One is to monitor learners' process and intervene if interactions are not occurring as expected. This may be based on interaction analysis methods, i.e., analyzing what learners are doing as they communicate and collaborate with one another as a basis for guiding collaborative behavior or launching regulative action [4]-[5]. Another approach is to structure (to "script") the setting by introducing sequences of activities to be achieved and specific support or constraints [2], [3]. This structuring may be implemented by instructions delivered by the teacher or via software (description of the task and subtasks to be achieved, and constraints to be respected) and design decisions related to the categories of technology used (e.g., communication tools or specific editors).

In this article, we consider a particular type of collective setting (collective problem-solving challenges) and a particular structuring objective: leading a group of learners to make their problem-solving organization explicit. A collective problem-solving challenge is a setting within which a group of learners is involved, as a team, in solving a problem that requires learners to pool their work, the problem and the setting being designed to create positive tension (a challenge) which motivates learners [6].

Scripting such a challenge consists in introducing some structure to support learners or make them consider some specific issues. Leading learners to make the organization of their problem-solving explicit is a goal that might be considered for different reasons. First, in relation to the question of support, it leads learners to consider an important, reputedly difficult issue they often overlook: building and maintaining a shared organization [7]-[11]. Second, in relation to intervention, it provides a basis for monitoring learners' organizational process [12].

This article proposes a set of design-principles which may be useful when designing software interfaces in such a way as to prompt learners to make the organization of their problem-solving explicit while communicating online. These principles are based on a theoretical model of collective work [13] and lessons learned from a case study. The data gathered from tests show trends regarding the usefulness and usability of the interfaces.

The rest of the article is organized as follows. In section 2, we introduce the background of this work and the model of collective work we used. In section 3, we present the way this work was carried out. In section 4, we present the proposed general design-principles. In section 5, we describe the case study and the particular implementation of the adopted principles we designed and tested. In section 6, we present lessons learned, in section 7 the discussion, and in section 8 conclusions and perspectives.

2 THEORETICAL BACKGROUND

2.1 Structuring problem-solving challenges

As pedagogical settings, collective challenges present several features of interest: they allow domains (such as mathematics) to be addressed from a new perspective, with a different "error" status from standard exercises; they allow various forms of work (individual, group,

-
- P. Moguel is with the Grenoble Informatics Laboratory, University of Grenoble 1, France. E-mail: Patrice.Moguel@metapatrice.com.
 - P. Tchounikine is with the Grenoble Informatics Laboratory, University of Grenoble 1, France. E-mail: Pierre.Tchounikine@imag.fr.
 - A. Tricot is with the CLLE-LTC laboratory, University of Toulouse II, France. E-mail: andre.tricot@univ-tlse2.fr.

class), and differing personal commitment by learners; the playful side has a positive effect on learners' self-determination, self-regulation and work in the presence of other learners [14]-[16].

A collective¹ challenge can be analyzed in several ways, e.g., problem solving as a cognitive activity involving a meta-cognitive activity or collective activity involving organization. In this article, we focus on organization. Regarding other cognitive activities, we will assume that support offered to learners may be based on the classical six stages problem-solving model [18]. This model includes: (1) Find the problem; (2) Represent the problem; (3) Plan a problem solution; (4) Execute the plan; (5) Check the solution; (6) Reflect to consolidate learning. These stages are not steps to be followed in a linear manner but issues to be considered in cycles. Depending on the settings and goals, support provided to learners may relate to one or more of these issues. At a meta-cognitive level, we assume that a collective problem-solving challenge is a self-regulated learning situation involving meta-cognitive processes: self-observation, self-judgment and self-reactions [19], [20]. As a collective situation, meta-cognitive processes are not only self-directed but also group-directed [20]. In this article, we do not focus specifically on these aspects of collective challenges.

In a collective challenge, learners must control their problem-solving process, which is often difficult for them. This can be supported by providing external control via monitoring and regulation or via scripting [5]. The concept of productive failure suggests that leaving learners to struggle while solving complex ill-structured problems can be a productive exercise in failure [21]. Learners may fail but nevertheless develop interactions forming a rich resource on which teachers can subsequently build during post-activities [22]. Moreover, from a scripting perspective, a known issue is the risk of over-structuring the process and sterilizing collaboration [23]. A core feature is thus to structure and support activity while providing learners with some flexibility [24].

2.2 Supporting problem-solving organization

One of the issues learners encounter in problem-solving settings relates to their tendency to switch directly to basic actions without planning their work [7], [8]. This issue is emphasized in collective problem-solving, which must not be seen as a set of individual sub-problems requiring a solution, and the sharing of the inputs and outputs: individual strategies can act as resources for proposing which tasks the group should consider, but collective solving also requires the additional task of collaboration management [9]. Actors must define goals more precisely

¹ Some CSCL literature dissociates the notions of "cooperation" (work is accomplished by the division of labor among participants) and "collaboration" (learners engage in a coordinated effort to solve the problem together) [17]. As we believe this distinction is often a matter of the degree of granularity, of point of view or concern, we use "collective" as a broad concept (as in "collective challenge"), but quote other authors' original wording when citing specific works or referring to literature in general.

and plan procedures [10]. They must maintain common ground - "mutual understanding, knowledge, beliefs, assumptions, presuppositions, and so on" [11].

Considering how learners may be supported in organizing themselves may be addressed by considering collective problem-solving as a particular instance of a collective work situation, i.e., a setting within which workers are mutually dependent [13], [25], [26]. Such work situations are studied in the field of Computer Supported Collaborative Work. Research has shown that actors engaged in such interdependent processes must address the specific issue of articulating ("dividing, allocating, coordinating, scheduling, meshing, interrelating") their respective activities [25]. This requires a meta-level overhead activity, not focused on producing the target output, but on creating the conditions for producing this output by maintaining a more or less stable pattern of cooperation. We use "organization" to denote this activity and, by extension, the output from this process. With respect to problem-solving, organization relates to the "plan a problem solution" and "execute the plan" steps.

Kobbe & al. [27] propose an overall model for describing CSCL settings in terms of structures (resources, participants, groups, roles, activities) and mechanisms (task distribution, group formation and sequencing). Building on these notions, we may list several organizational issues raised by collective problem-solving settings:

1. Elaborate and maintain a common understanding of the problem and, more generally, common ground (relevant concepts to be used, issues to be considered, possible actions, etc.).
2. Identify and define the tasks the group intends to address and how they will be addressed; plan how work will be divided and articulated.
3. Manage the articulation of work (monitor and regulate the process, reconsider tasks or roles, etc.).

In scripted settings, part of the organization is defined by the structure introduced by the teacher. Another part (called learners' self-organization) is developed by learners themselves, in context, as an inside-group feature which emerges from the way learners enact the setting [26]. Typically, learners may refine the way tasks are divided into sub-tasks or distributed, adopt particular articulation processes (way of monitoring the process or checking results) or develop particular way of using the tools on offer in relation to their effective activity.

2.3 Bardram's model of collective work dynamics

We use Bardram's model of collective work dynamics as a theoretical basis to address organizational issues.

Bardram [13] has proposed a model of collective work which highlights the dynamics of collaboration. It is based on the distinction between three basic notions/levels: *co-construction*, *co-operation* and *co-ordination* (see Fig. 1). Co-construction is the level at which actors focus on conceptualizing or re-conceptualizing their organization in relation to their shared objects. Co-operation is the level at which actors are active in considering the shared goals, relating to one another and mak-

ing corrective adjustments to suit the overall goal. Co-ordination is the level at which actors concentrate on their assigned sub-tasks: work relates to a common goal, but their individual actions are only externally related to one another; they carry out the overall task from the point of view of their individual activity. These levels correspond to analytical distinctions: activity may take place simultaneously at all levels. The model stresses that it is of prime importance for understanding collaboration to consider the *dynamic transitions* which may occur from one level to another and the *breakdowns* which may appear.

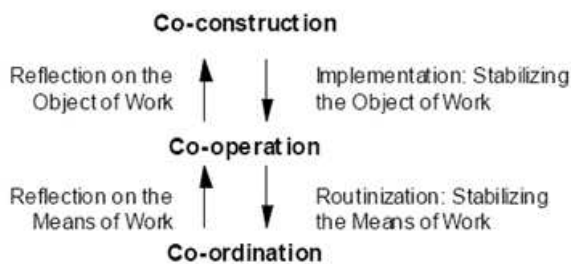


Fig.1. Bardram's model [13].

Various transitions may occur during activity. Bottom-up transitions are related to an analysis of the object or the means of work. They may occur in relation to a breakdown or an explicit shift of focus. This happens, for instance, when a learner fails with a task he/she was supposed to manage, a conflict arises, or learners suddenly realize they have different conceptualizations of the setting or of a particular notion. Top-down transitions are related to the solving of issues and contradictions. They lead to stabilization of the object and the means of work. Breakdowns are natural and important events that, however, are not necessarily discovered or managed. When actors become aware of a breakdown, the group is challenged. Its management should lead to a reflection on the means or the object of the work (bottom-up transition). Solving a breakdown leads to stabilization of the object or means of work, and should end in a top-down transition.

In a learning context, transitions may originate from two sources. First, actors (here, learners) may spontaneously go from one level to another, in relation to a difficulty they encounter or through a voluntary shift of focus. Such transitions correspond to self-organization as defined above. Second, if a tutor monitors learners' processes, he/she may suggest transitions, such as drawing learners' attention to a poorly defined or disputed task or notion, or to the failure to achieve some planned action. Similarly, he/she may highlight a misunderstanding or a learner deviation from what was decided. Transitions may also be motivated by a pedagogical opportunity.

2.4 Leading learners to make their organization explicit

Learners involved in collective work must be guided and supported (see introduction). Within this perspective, considering learners' organization is of specific importance [26]. The absence of any collective organization hinders the very possibility of collective work, in particular in problem-solving contexts, and may lead learners to

lose their commitment (unproductive failure). Learners may also adopt organizations which prompt little or unsatisfactory interactions (quite apart from problem-solving efficiency). For instance, after a period of separate efforts, learners may realize they do not share the same concept of the setting, preventing them from pooling thought or work (common ground issue). A gifted learner may solve the problem while the others look on. Etc.

Supporting learners who must engage in planning and monitoring regulatory processes may be addressed in different ways.

A first approach is to suggest or impose organizational patterns. Such an approach has for instance been explored in inquiry learning [28]. Various studies have shown that supporting learners with interfaces suggesting the consideration of specific inquiry tasks (e.g., "generate hypotheses" or "differentiate variables"), and supporting their articulation and achievement, appears pedagogically fruitful [28]. In the more general case of problem-solving as we consider it, patterns comprising a list of tasks to be considered and how they should be addressed may be elaborated from the canonical model mentioned in section 2.1 and lessons learned related to learners' difficulties.

A less structuring approach is to lead learners to consider the importance of organizing themselves and the generic issues related to collective work, but without suggesting any specific task. Instead of just introducing the problem and leaving learners without support (which is likely to make them start individual basic actions without planning their work), learners are encouraged to get organized and consider (for instance) defining and agreeing on the tasks on which they intend to focus. Such support is generic in that it addresses a meta-level (defining tasks, agreeing on division of work, etc.), as opposed to introducing specific tasks such as "generate hypotheses" in inquiry learning or "represent the problem" in problem-solving.

One way of implementing this less structuring approach is to offer learners interfaces which lead them to make their organization explicit, listing the tasks they intend to consider or how they distribute them. Offering such interfaces draws learners' attention to organizational issues. It is a step towards preventing them from engaging in random actions, while at the same time limiting the risks of over-scripting and diminishing the playful side of the challenge.

When learners list the tasks they intend to consider, how they distribute them or how they intend to address them, the result is a kind of plan. Such a plan is not to be seen as something that will be applied step-by-step by learners. As highlighted by many studies, action is emergent and situated. Such a plan is to be thought of as a resource for action, adaptable in context [29]. Similarly, what can be hoped from suggesting learners to make their organization explicit is not to be considered as perfectly reflecting their thoughts (see issues related to accountability in ethno-methodology studies for instance). Here, "making explicit" refers to the efforts made by an

actor or set of actors to render some construction intelligible to others and, incidentally, improve their own analysis and understanding.

Although we will not address these aspects in this article, we may mention two other merits of prompting learners to make their organization explicit. First, it leads them to engage in communication, argumentation, analysis or reflection related to the problem-solving plan and division of labor, which is entirely consistent with overall goals [26]. Second, when teaching problem-solving, it is of general value to make learners address issues of strategy and methodology. For example, in a cooperative problem-solving task in geometry, Kramarski [30] compared groups receiving metacognitive instruction on strategy (consider which strategies are appropriate) and reflection (monitoring the solution process) with groups not receiving this instruction. The results showed that metacognitive instruction improved performance on problem solving and transfer tasks.

3 METHODOLOGY

Software interfaces meant to suggest and support a certain form of activity must be tested in context and adapted as needed. Initial design may however build on design principles and lessons learned. Within this perspective, the contribution of this article is a set of general design-principles for interfaces which prompt learners to make the organization of their problem-solving explicit.

Our methodological approach involves using models of human activity as resources to build implementable models, which can serve as the basis for a system that learners can use (the purpose is not to make or test hypotheses related to these models). The proposed design principles (described in section 4) are based on Bardram's model and, more generally, the background introduced in section 2. They have been tested (and various lessons have been learned) on a number of research actions carried out on a case study, namely the "race with no winner" challenge. This involved developing a particular implementation showing the interest of Bardram's model for structuring learners' organizational activity (summa-

rized in section 5.3; an initial prototype was presented in [6]); elaborating an analysis grid [31]; conducting an exploratory study to analyze the influence of using or not using these interfaces on learners' organization, motivation and organization salience [32].

The way this work has been conducted is synthesized in Table 1 and described below.

3.1 Elaboration of general design principles

There are various (non-exclusive) ways to elaborate software interfaces designed to suggest a particular form of activity such as iterative design, participative design or continued-in-usage design [33]. In this project, we have used a theory-based approach (designing interfaces which denote the model's layers and notions used as a theoretical background; Bardram's model in this case) coupled with iterative design (testing various versions).

A theory-based approach facilitates traceability, i.e., the capacity to analyze learners' usage of implemented interfaces (and, more generally, learners' enactment of the setting) in the light of the design principles and the underlying model, which favors knowledge capitalization [33]. However, models intended to enhance understanding are not necessarily pertinent for design. Therefore, in addition to an exploratory experiment confirming that Bardram's model helped understanding learners' organizational processes (LL1), we carried out a usability experiment to check if interfaces based on Bardram's notions made sense to learners (LL2, later on confirmed by LL3). These iterations also helped improve ergonomic details. Due to space limitations, the lessons learned, LL1, LL2 and LL3 are compiled in section 6 (not presented step by step). An a posteriori analysis of the process is discussed in section 7.

3.2 Case study and data analysis

The case study we have used is based on a mathematical challenge (not designed by ourselves). It is described in section 5.1. We transposed the original setting into an online setting, requiring learners to solve the problem while only communicating via the available interfaces and chat. We implemented three successive tests. The

	Step#1	Step#2	Step#3	Step#4	Step#5
Input	1. Bardram's model 2. Challenge "race with no winner"	1. Bardram's model 2. Lessons Learned LL1	1. Prototype	1. Bardram's model 2. Input / ergonomic details 3. Lessons learned LL2	1. Implementation #2 2. Coding grid
Actions	Exploratory experiment: two groups of three and four learners solving the problem in front of a common computer; record of data (video)	1. Elaboration of general design principles 2. Elaboration of implementation #1	Usability experiment: one group of two learners + the experimenter (as a peer)	1. Elaboration of implementation #2 2. Elaboration of a coding grid	Experiment: three groups of three learners using the prototype and three groups not using the system; record of data, coding and analysis
Output	1. Confirmation of the relevance of Bardram's model 2. Lessons learned LL1 [6]	Prototype [6]	1. Confirmation of the general design principles 2. Input / ergonomic details 3. Lessons learned LL2 [6]	1. Implementation #2 2. Coding grid [31]	1. Lessons learned LL3 2. Data related to motivation [32] 3. Data related to automatic analyses [32]

Table 1. Methodological approach synthesis

tests were done on learners at a stage (equivalent to 11th Grade, 16-17 years old) at which they master the technical skills (mathematics and physics) required to solve the problem, so they would not run into difficulties related to the field itself. The scenario was identical for all tests.

For the tests in which learners used the supportive interfaces, each learner was connected to a separate computer loaded with software recording the screens (video file). Chat and other tools were logged in XML format. Every action (mouse click) and message was recorded and associated with an author, a time-stamp, duration, the tool being used and additional data (e.g., the typed content) if relevant. This allowed us to build a chronological reconstruction of each collective session as a column table displaying the messages and actions of the learners in the various groups. Data was coded and analyzed according to a coding grid (a set of indicators) allowing analysis of learners' organization in terms of co-construction, co-operation, and co-ordination. This grid is described in [31]. For example, co-operation is analyzed with respect to issues such as "decision-making about the organization" and indicators such as "allocation of tasks", "adoption of a division of labor", "making the organization visible" and "resolution of conflict", as well as the revision of these four sub-criteria in the course of activity.

The experiments and analyses were conducted to discover episodes and general trends reflecting the interfaces' usability and usefulness. Although we analyzed groups using and not using the system, we did not construct these tests in order to use statistical methods. The reason is that organization issues are related to the complexity of the setting. We needed to implement an authentic setting, lasting several hours, in authentic conditions. At this stage, the objective is to show that one can lead learners to consider organizational issues explicitly, and to elicit design principles for organization support that might be included in effective teaching systems. Analyzing statistically the effect of a variable on another (e.g., whether or not learners develop organizational or problem-related skills) is on our research agenda. This, however, is to be conducted in a fixed, consistent setting. The present experiments were not conducted under these conditions, and so items that could be analyzed statistically were thus not measured. It may be noted that this process exemplifies a multilevel, user-centered design approach taking into account problem-solving, organization and usability features.

4 GENERAL DESIGN PRINCIPLES

The goal is to elicit principles for designing interfaces which lead learners to consider organizational issues explicitly and highlight these issues for tutors. Given this goal, it makes sense to use Bardram's model (which is a model for understanding and qualifying organizational issues) as a basis for elaborating interfaces which may act as *conceptual tools* helping learners to reflect on their organization and making the decisions they take explicit. Things would be different if, for example, the concern

was problem-solving efficiency, which is not necessarily related to whether or not organization is explicit.

The proposed approach is thus to lead learners to consider (define, reflect on, use as a resource for activity) organization by offering them interfaces which encourage them to consider issues related to co-construction, co-ordination and co-operation notions, and the aspects introduced in section 2: defining common ground, identifying subtasks, adopting a division of labor or making top-down and bottom-up transitions. Within this perspective, the following design principles may be considered.

1. Provide learners with a transposition of the co-construction notion/level as a *phase* and/or *place* relative to the elaboration (and, in case of difficulty, revision) of the tasks to be considered. It may be a *place* in the sense that learners are offered specific editors to elaborate these constructions and a *phase* in the sense they are led to consider this task before engaging in the actual solving process.
2. Provide learners with a transposition of the co-operation level as a phase and/or a place relative to the definition of how tasks will be fulfilled, e.g., the role of each member or the sequencing.
3. Provide learners with a transposition of the co-ordination level as a phase and/or a place relative to fulfilling tasks (e.g., in our case study, measuring or calculating values).
4. Make the elaborated organization constantly visible to the group (primo-definition and revisions).
5. Allow learners, in context, to come back to a preceding phase and change what had been defined (bottom-up transitions *via* the notion of phase).
6. Allow learners, in context, to change what had been defined at a preceding phase (bottom-up transitions) without going back to this phase as such. For example, they must be allowed to change the division of work while they are fulfilling individual tasks, without skipping to the phase or place when/where division of labor (as such) was initially defined.
7. Keep interfaces and, in particular, interfaces relative to the definition of the plan and its implementation, close to each other.

5. CASE STUDY

In this section, we describe the case study we have used and implementation of the above design principles.

5.1 The "race with no winner" challenge

The setting is based on work from a community of practice dedicated to the use of simulations in mathematics and physics [34]. It is based on a Flash animation.

The animation (see Fig 2) can be used to simulate a race involving three cars (to be chosen out of the 10 shown at the top of the car track). Each car has a different speed and dynamic (some of them stop at one place and restart after a variable number of seconds). However, each car's behavior (speed, where it stops and the time

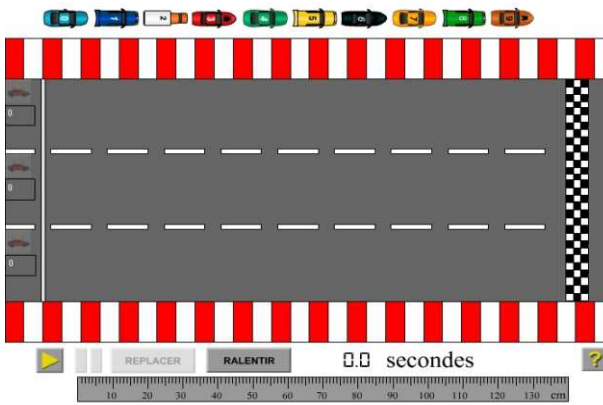


Fig. 2. The animation track [34]

elapsing before it restarts) is identical from one run to the next, and can thus be anticipated by a series of measures.

The challenge, as we transposed it, involves a group of online learners (we used groups of three or four). Each learner is connected to a computer and only communicates with the others via the system interfaces.

The challenge is as follows: the tutor chooses three cars and places one of them (called “the tutor’s car”) somewhere on the track (not necessarily on the starting line); learners have a limited amount of time (e.g., 30 minutes) to calculate where to place the other two designated cars (“the learners’ cars”) on the track in order for the three cars to cross the finishing line at the same time. Once learners have placed their cars, the animation is run to check if they arrive together (“no winner”).

To calculate where they should put their cars, learners are given a set amount of time (two hours; we responded favorably when learners asked for more) to prepare themselves and, in particular, use the animation to collect data and perform the necessary calculations to find the various cars’ speed, stops and duration of stops.

The setting thus consists of three phases. In Phase 1, learners analyze each car’s behavior (as they do not know in advance the three cars that will be selected by the tutor in Phase 2, they have to analyze all 10 cars). They must also prepare Phase 2 (see section 5.2). In Phase 2, the tutor puts the tutor’s car on the track and designates the other two; learners have 30 minutes to calculate where to put these two cars, based on the data calculated in Phase 1. In Phase 3, the animation is run to check the learners’ solution: the tutor hits the “play” button, the three cars start to move, and everybody holds their breath till the cars reach the finishing line.

Technically, the problem requires learners to solve equations in order to determine speed, duration and distance, which involves skills related to variation rate, table of values or Cartesian plan which learners can already manage. They have already done similar individual exercises in their curricula. The challenge is just to put their skills into practice in a coordinated way.

5.2 Organizational aspects of the setting

In section 2.2 we have listed several organizational issues raised by collective problem-solving settings: elaborate and maintain a common understanding of the problem

and, more generally, common ground; identify and define the tasks the group intends to address and how they will be addressed, plan how work will be divided and articulated; manage the actual articulation of work.

With respect to the “elaborate common understanding of the problem and common ground” aspect, learners must identify and agree on the relevant notions (car’s behavior, speed, duration of stop, etc.). As basic examples, learners often conceptualize the setting differently, use different terms or structures for an issue (e.g., the way a car’s various stops are reported) or take different decisions (e.g., what reference is used for measures or units).

With respect to “identify and define the tasks the group intends to address”, “define how tasks will be addressed” and “plan how work will be divided and coordinated” aspects, learners are told that Phase 1 requires a large number of measurements and calculation, and that they should divide work. As different divisions of work are possible, they have to define what tasks they will consider and what role each learner will play. As examples of tasks (and underlying strategies) adopted by learners: each member addresses all measures and calculation related to a subset of cars; each member addresses a given task (e.g., measures speed) for all cars; two members carry out some actions while the third manages their articulation; one member systematically checks the results of the other two. Phase 2 also requires a decision on the tasks to be achieved (e.g., calculation of the race duration), the role of each learner or the overall articulation as learners have to fulfill them in 30 minutes sharp². During Phase 1 they are prompted to consider how they would deal with Phase 2.

With respect to the “manage actual articulation of the work” aspect, learners have to decide how to circulate data, how to make sure the process is progressing smoothly (everybody is active and can manage his/hers tasks) and how to react if not, how individuals should behave when they face an issue, etc. As examples taken from different groups: a difference on a numerical value made learners realize they were not using the same references or notions (grounding issue); one of the learners failed to fulfill their part of the work, or diverged from what was planned, and the organization had to be adapted; a learner was exhausted and became inoperative; a leader emerged, the other learners gained trust in his ability and asked him to take on some additional tasks (the two other remaining active, though).

5.3 An implementation: the Albatros system

Albatros is an implementation of the design principles listed in section 4 for challenges such as the race-with-no-

² In fact, some groups ask for extra time in Phase 1 (preparing data) or Phase 2 (measuring where to put the cars). Allowing them for some limited extra time does not impact the setting and avoids generating frustration. In the final study, all groups using the system felt close to a solution and asked for additional time to finish their calculus, and Phase 2 in fact lasted for an average of 53 minutes (see Table 2). This also exemplifies the playful nature of the setting, and its impact on learners’ engagement.

winner. The system allows a group of online learners to collectively build a solution (Phases 1 & 2 of the challenge) and test it (Phase 3). For this purpose, it integrates a variety of resources including two dedicated shared editors to lead learners build their organization and enact it: the task-editor and the array-editor.

In the following subsections, we introduce the system with respect to the design principles listed in section 4, featuring these specific editors. Learners also have permanent access to an individual and a shared car-race animation (to carry out individual and collective measures and tests), communication tools (a chat and voting tools) and various handy features such as a text zone for individual editing and a calculator.

Co-construction notion/level

Representing an attempt to support learners on the co-construction level (co-construction: conceptualizing or re-conceptualizing the organization in relation to shared objects), the task-editor (see Fig. 3) allows elaboration and revision of the notions used and the tasks considered. With respect to the model, the design decision is to lead learners to consider co-construction issues by making them list and define the notions and the tasks they intend to consider (and, within this process, interact on this issue and take shared decisions).

L I N E	Priority	Involved Notion	DATA NAME	DATA DESCRIPTION	TYPE OF ACTION
1	High	Cars that stop	Distance between two stops		Measure
2	High	All the cars	Speed	Speed of the car	Calculate
3	High	Cars that stop	Duration of the stop	Duration of the stop is important because the speeds are constant	Measure
...

The linetask editing zone and the already edited tasks

Fig. 3. The task-editor (translated)

The editor prompts learners to define the data and actions they will need as a list of tasks (which will be automatically transformed into a structure - an array - within which they can distribute work and enact the plan in the following step). Tasks are defined as a structured line (upper part of the interface) mentioning the involved notion as defined by learners (e.g., “cars that stop”), the name adopted by the group to denote the data (e.g., “distance between two stops”), a textual description (to improve their chances of understanding each other), what is to be carried out in relation to this data (from a list of possible actions provided as a support, e.g., “measure”) and the given priority if useful. The interface is shared and the chat allows synchronous interaction. Each line is to be collectively acknowledged via the voting tool (not presented here) before being moved to the “list of acknowledged tasks” (lower part of the interface). However, a learner may force the vote, avoiding excessive constraints (this requires an explicit action). Learners may return to any line at any time.

The resulting list of tasks forms a kind of general problem-solving plan and, via the fact that they are referred to by the tasks, a list of notions (e.g., “distance between two stops”, “duration of the race” or “speed”).

Learners are first offered to use this task-editor in Phase 1, where it suggests identifying the tasks (and underlying notions) that must be achieved to collect the required data. It is again offered in Phase 2, where it suggests identifying the tasks to calculate where to put learners’ cars according to the tutor’s one.

Co-operation and co-ordination notions/levels

Representing an attempt to support learners on the co-operation (co-operation: actors are active in considering the shared goals; they relate to each other and make corrective adjustments according to the overall goal co-construction) and co-ordination levels (co-ordination: actors concentrate on the subtasks they have been assigned; their work relates to a common goal, but their individual actions are only externally related to each other; they carry out the overall task from the point of view of their individual activity), the array-editor (see Fig. 4) offers two modes allowing (1) distribution of work and (2) enactment of the plan. With respect to the model, the design decision is to lead learners to consider co-operation issues by asking them to distribute and redistribute tasks (and, within this process, interact on this issue and take shared decisions), and to support top-down and bottom-up processing by allowing them to be implemented within quasi-identical (initial division of work) or identical (revisions) interfaces.

The interface is an array whose structure is generated from the list of tasks collectively constructed with the task-editor. For every line (every task) of the general plan, and every object involved in the solving (i.e., each car), a column *per* learner is generated (in Fig 4, the array-editor content has been initialized from another plan than the one featured in Fig. 3, as a way to show tasks and notions defined by different groups). As a result, a line in the array denotes a task (e.g., line 1: data = “arrival time” and action = “measure”) and, for each car, three cells that may be used to denote the fact that learner L#1, L#2 and/or L#3 will address this task. The fact that car/task pairs are associated with one column *per* learner allows them to delegate tasks to just one, two or three of them.

When learners move from the task-editor, the array-editor is tuned in definition-mode. All cells are initialized with “?”, i.e., none of the task is initially attributed. Learners are prompted to declare who will carry out each task by clicking in the corresponding cells. For instance, learner L#i may declare he/she will carry out Task T#j for Car#k by clicking on the relevant cell; the “?” is replaced by an “OK” in the corresponding cell of the L#i column. Learners may interact via the chat to coordinate (as examples of interactions: “Cristina, you make line 1234”; “to better organize ourselves I make the cars 7 8 9, you share the others it will be quicker”). As the interface is shared, each learner is aware of the other learners’ declarations.

Used in this definition-mode, the array-editor leads learners to consider co-operation issues. The result is a more precise plan (list of tasks is refined or confirmed; tasks are distributed).

Once learners decide they are over with the definition-mode (i.e., distributing tasks), they can skip to the execution-mode (here again, this requires a vote). Now, the cells marked as "OK" become editable, i.e., the relevant learner can edit the value. The shared interface allows every learner to know what he/she is supposed to do and what the others are doing. The problem-solving evolution is denoted by the fact that the "OKs" are gradually replaced by values (see Fig. 4).

Used in this execution-mode, the array-editor supports learners at the co-ordination level. As an example, we can see in Fig 4 line 2 that L#2 has achieved the task "measure arrival time" for car#0, #1 and car #2 (values = 5.7, 28.3 and 19.4). L#1 and L#3 have both completed the task "calculate average speed (constant) of cars" for car#0 (values = 0.25 and 0.246) and car#1 (values = 0.05 and 0.049), and plan to address it for car#2.

Learners are first offered to use the array-editor in Phase 1. It suggests distributing (and, then, offers a structure for achieving) the tasks that must be achieved to collect the required data. The editor is also offered in Phase 2, where it suggests distributing (and, then, offers a structure for achieving) the tasks required for calculate where to put learners' cars according to the tutor's one (here, the array is restricted to the three involved cars).

Transversal design principles and features

The other principles listed in section 4 find the following implementations.

The elaborated organization is reified by the array and made visible at all times, including when the plan is executed. In fact, the same representation is used to define and execute the plan.

Learners may skip from the array-editor definition-mode to the execution-mode while some tasks have not yet been attributed. We can see in Fig 4 co-occurrence of non-attributed tasks ("?"), attributed tasks ("OK") and already achieved tasks (numerical values). Learners can also add tasks at any moment. We can see in Fig 4, line 9, an example of a task "Duration of a stop", incompletely defined, which has been added after the other tasks had been already distributed and, for some of them, achieved. In other words, learners may define and enact partial plans, and also modify their plan at any moment. This

feature was inferred from the exploratory experience that showed learners often engaged in problem solving before their plan was complete. It corresponds with the fact that the model's levels correspond to analytical distinctions and activity may take place simultaneously at all levels.

Learners can change decisions by coming back to a preceding interface or mode but, also, in context (not going back to this phase as such). As an example, when using the array-editor, they can change the allocation of a task by coming back to the definition-mode of the array-editor. This denotes an explicit intention of reconsidering what had been previously decided. When they skip back, the change is reported (without re-initializing the overall array, i.e., not losing the already calculated values). However, learners may also change any element (addition, deletion, change or allocation of a task) while remaining in the array-editor in execution-mode, in context, by clicking on the relevant lines and/or cells (this is the case of line 9, which has been directly added in the array-editor). In other words, moving from the organization to the problem-solving activities, and *vice versa*, may consist in a change of phase/place (in the sense introduced previously) or in different uses of the array-editor in execution mode. Here again, allowing in-context changes is in line with the model and corresponds to patterns observed during the experiments.

The interfaces relative to defining and implementing the plan are, in fact, functionally quasi-identical. When skipping from the list of tasks to the attribution of the tasks and, then, to the edition of values, there is little conceptual change and surprise for learners. With respect to the model, the design decision is to implement the fact that as co-construction, co-operation and co-ordination notions are analytical distinctions, they should not necessarily correspond to different interfaces or features. Cooperation and co-ordination are often intertwined. From a user-interface perspective, it thus makes sense to propose very similar interfaces and keep the system as simple as possible. In contrast, co-construction requires a different interface. Moreover, it indeed makes sense to lead learners to consider specifically this task (see Section 2). The system implements this by the fact that (1) it offers a specific interface and (2) the items listed via the interface initialize the array-interface, which is the one that constitutes the center of learners' activity.

6 LESSONS LEARNED

In this section, we compile the lessons learned from the

No.	Task's Priority	Type of Car	DATA	Actions	Car#0			Car#1			Car#2			Car#3
					L.#1	L.#2	L.#3	L.#1	L.#2	L.#3	L.#1	L.#2	L.#3	L.#1
2	1 High	All	Arrival time	Measure	?	5.7	?	?	28.3	?	?	19.4	?	?
3	1 High	All	Average speed (constant) of cars	Calculate	0,25	?	0,246	0,05	?	0,049	OK	?	OK	0,076
4	1 High	With stop	Stop duration	Measure	?	0	0	?	0	0	?	5.3	5.2	?
5	1 High	All	Distance between car and finish line	Measure	OK	?	1,40	OK	?	1,40	OK	?	1,40	OK
6	1 High	With stop	Number of stop	Read	OK	?	?	OK	?	?	2	?	?	1 dep
7	2 Medium	With stop	Cars that stop	Measure	?	OK	OK	?	OK	OK	?	OK	OK	?
8	2 Medium	All	Distance between start line and car	Measure	OK	?	0	OK	?	0	OK	?	0	OK
9	4		Duration of a stop		?	?	?	?	?	?	2,6	?	?	9,2

Fig 4. Dividing work and enacting the plan (translated).

different experiments. The examples and data are taken from the final study (step#5 in Table 1; general figures are summarized in Table 2). The items are structured in 6 subsections: first, the general system's usability; second and third, the two objectives underlying the design: triggering/supporting learners' shared organization and the salience and intelligibility of learners' organization within a perspective of monitoring; fourth and fifth, the two main risks we had identified, over-constraining learners and diminishing their engagement; finally, learners' success (frequency of correct solutions).

6.1 General usability

The tests suggest that the general design does not disorientate learners. During the debriefing, none of the groups using the system mentioned difficulties in accessing the editor, and analysis shows that none of the groups developed their activity without using the available editors, or using them in a way inconsistent with the design intentions. As usual, there is some unattended usage of the system (see following subsections), but none contradict the design decisions and their underlying rationale.

It may be mentioned that, not surprisingly, learners start by playing with the animation. This has the positive impact of making them discover the setting but presents the risk that they in fact start to solve the problem on their own and, consequently, have difficulty in shifting to collective work. This must thus be controlled (e.g., presenting a not-to-the-scale animation at first).

6.2 Triggering/supporting learners' organization

The first and major objective of the design is to trigger and support learners in building a shared organization. With respect to this objective, the tests suggest the following conclusions.

As highlighted by other works ([7], [8]), defining a collectively agreed plan is not something learners naturally engage in but, however, they are capable of doing so, and easily accept to address this issue if suggested. In the exploratory phase, after Phase 1, we asked groups to respond to questions such as "Could you list the various actions you intend to fulfill when the teacher's car is put on the track?" Learners were able to exhibit elements of planning (as an example: "Determine the distance between the selected car and the finishing line; one calculate how much time it will take to get to the finishing line; ..."). When offered the system, all groups used the task-editor and described their plans as a consistent (although not necessarily optimal or complete) list of tasks, see Fig. 3 or Fig. 4.

From a general point of view, learners spend a large amount of the total duration of the challenge organizing themselves. An average of 39% of the total duration and 67% of the messages related to understanding the problem and organizing the problem solving (task-editor and array-editor in definition-mode); 61% of the total duration (33% of the messages) related to execution of the plan and solving the problem.

The task-editor is significantly used and learners interact intensively during this period. Its use corresponded to

an average of 25% of the process and 42% of the messages. When using this editor, learners' activity is as expected: definition of a set of lines denoting the way they intend to act and the involved notions (which is not necessarily what will happen), using the suggested items and/or creating some others, each line being elaborated, discussed and negotiated via the chat. Fig. 3 and Fig. 4 illustrate the editor's use (two different plans created by two different groups). Analysis of the chats suggests that the output (the list of tasks) is considered to be complete (even if it is subsequently revised), which contrasts with the division of labor (see below).

The definition mode of the array-editor is used in various ways. Basically, learners distribute tasks, using both the array-editor and the chat. As an example of interaction: L#1: "we take too many things"; L#2: "let's distribute things cleverly then"; L#1: "we mustn't take too much"; L#3: "two each"; L#1: "here it's OK, isn't it?". However, whereas some groups skip from organization to execution with a well-defined division-of-work, it is very incomplete for some others (see for example Fig. 4). In such cases, voting the decision to skip to the execution mode does not correspond to "everything is settled now" but, rather, "the plan is sufficiently detailed to engage in work, and we will continue later on". As an example of such use, one of the groups used the array-editor five times in definition-mode (allocating tasks) and five times in execution-mode (nine transitions). In this group, we found interactions such as: "can we change mode, I'm done with my part" [the learner proposed to come back to the definition mode as she had finished with the tasks allocated to her and wanted to take some others]; "same for me"; "vote". As a more radical example, one of the groups used the task-editor to co-construct a plan, but did not use the array-editor to allocate tasks. Once what had to be done had been established (co-construction), the group decided that one learner would fill the array while the others would provide support and verification. This role distribution did not need to be represented in the system, and learners thus used the shared array in execution mode only. In other words, as task-allocation was straightforward, they did not need the system's support for the co-operation phase and only used the interfaces related to the co-construction and co-ordination phases. Here, the setting (introduction of various phases, tools making suggestions and hints) succeeded in making learners consider organizational issues while some features were not used as expected, which may be seen as an interesting flexibility of the system.

The suite of tools - task-editor / array-editor (definition-mode) / array-editor (execution-mode) - is used differently in the two phases. In Phase 1 (compiling the data), all groups use the task-editor to list the tasks they intend to consider (a posteriori analysis shows the list is fairly complete and not very different from what is actually achieved by learners), and the array-editor is used to edit and share values, with some variety in the way tasks are allocated (see here before). In Phase 2 the task-editor

is still used. The three groups which were offered the system used the task-editor to build lists of six, five (plus two added via the array-editor) and four tasks. In contrast, the definition-mode of the array-editor was little used (two groups did not use it at all), groups largely using the array-editor only to share data. Here learners' major concern was to start calculating (time was limited), they were already engaged in a way of working together and they tended to skip making the division of labor explicit. In two groups, one learner took the lead implicitly by starting to fill the array (no interaction related to such a decision). In another group, two learners shared efforts while the third was completely lost and could not join the two others' process. From the point of view of system usability, this suggests that when learners voluntarily decide to skip making their organization explicit they still use the offered tools. In other words, these tools are not seen as hindering work on problem-solving. In order to understand such episodes, it would be interesting to study whether there is a kind of transfer of the organization adopted in Phase 1 to Phase 2.

One of the most important aspects of the design seems to be the coherence organization/execution and the fact that the array-editor is initialized with the tasks defined in the task-editor, and is usable to allocate tasks and to share measures and calculated values. This seems to render the interfaces easily understandable and usable by learners to manage (successively and then, in context, simultaneously) organizational and problem-solving issues. Groups that are offered the interfaces systematically used them to transmit and share data. While interacting with the chat, they constantly referred to the data as represented and shared in the editors (whereas, as they were using the chat, they could have transmitted data via the chat, as groups not using the system did). Groups which attached less importance to making their organization explicit (in particular, in Phase 2; see above) still used the editors to share data, which here again suggests that although the interfaces were designed and introduced as supportive to organization, learners also attributed them a value for supporting the solving as such.

Learners really reconsider the list of tasks (bottom-up transition to the co-construction level in Phase 1: G1, one added task; G2, three added tasks and one modification; G3: two added tasks). This seems to relate to the fact that

this phase is conducted rather extensively. However, not surprisingly, when learners are involved in allocating tasks or calculating and some task modifications are required (adding or modifying lines), they do not come back to the task-editor: they achieve the modification while remaining in the array-editor. One can see such a case in Fig 4, line 9. In other words, the task-definition editor appears to be seen as allowing them to engage in the problem-solving process, the primo-plan being naturally adapted, in context, during action.

During actions, learners face different grounding issues (e.g., making a notion or a task more precise; bottom-up transition to the co-construction level). As examples: units of measurement; concept of speed for cars that stop; the way duration of stops should be addressed (e.g., qualifying car's stops in terms of distance from the departure line or in terms of seconds after the start). These issues were usually managed informally, via the chat. As an example of interaction: *"try to determine when car 3 stops for the first time, I find 6.4"*; *"do you mean the stopping time?"*; *"no, when it stops"*. In our design, making notions explicit is only addressed in the task-definition editor, as an open text. This text is not reported in the array to avoid making the interface too complex. However, in keeping with their use of the array-editor as also providing a way of refining their organization, some groups used unemployed cells to make some issues more precise (an unexpected use of the editor). For instance, see Fig 4, end of line 6, learners used the array-editor to share some comments (here they wrote "one departure" to mention this car stops only once).

As another interesting example of unattended usage, a group decided to consider a rehearsal task. They phrased it in a way that could not easily be represented with the offered editor. Although they could have just mentioned it in the chat, they represented it in the editor as a free text edited in a cell. This suggests there is some place for improving the editor but, also, that the offered support seems to lead learners in organizational and explicitation processes that go beyond the basic syntactical support of the editors. Similarly, we found local uses of the array-editor such as to mention an issue learners were anxious to forget or as a support for regulation actions. As an example: L#1: *"we're OK now"*; *"we have almost checked there are no more empty cells"*; *"let's simulate, guys"*; L#3: *"let's launch the shared animation"*.

		G1		G2		G3	
		duration	messages	duration	messages	duration	messages
familiarization with the setting		10 mn	-	12 mn	-	10 mn	-
Phase 1	task-editor	41 mn (30%)	58 (36%)	52 mn (29%)	134 (47%)	50 mn (25%)	84 (50%)
	array-editor (definition mode)	16 mn (12%)	47 (29%)	30.5 mn (17%)	65 (23%)	21 mn (11%)	49 (29%)
	array-editor (execution mode)	78 mn (58%)	57 (35%)	99.5 mn (54%)	87 (30%)	127 mn (64%)	36 (21%)
	Total (100%)	135 mn	162	182 mn	286	198 min	169
Phase 2	task-editor	5 mn (8%)	5 (33%)	17 mn (30%)	21 (32%)	2 mn (5%)	2 (9%)
	array-editor (definition mode)	0 mn (0%)	0 (0%)	14 mn (25%)	4 (6%)	14 mn (38%)	7 (32%)
	array-editor (execution mode)	60 mn (92%)	10 (67%)	25 mn (45%)	40 (62%)	21 mn (57%)	13 (59%)
	Total (100%)	65 mn	15	56 mn	65	37 mn	22

Table 2. System's usage, final study (step#5 in Table 1), general figures

6.3 Salience and intelligibility of learners' organization

The second major objective of the design is to improve the salience and intelligibility of learners' organization within a perspective of monitoring. With respect to this objective, the efforts conducted as researchers to code and analyze learners' processes indicate our interest in the system to facilitate analysis. First, the overall process of the groups using the system is much more structured, which eases coding and understanding [31]. Analyzing groups not using the system is made much more difficult by the fact that (1) organization and resolution of the problem are far more intertwined and (2) learners often change level according to their individual resolution, without noticing or taking into account the opinion or the progress of the others. Second, as the system and the coding tool are based on the same theoretical background, use of the different system features (editors, vote, etc.) provide indicators that may be matched with learners' organization levels and, in particular, transitions. Here, care must be taken that these indicators only allow hypotheses, to be confirmed by the messages (chat) and actions analyzed. However, this does indeed ease the process. As an example, 66% of the transitions may be correctly identified from the system indicators only (i.e., in other words, by an automated analysis) [32]. These preliminary elements are to be confirmed by studying teachers' (*vs.* researchers') analyses.

6.4 Risk #1: over-constraining learners

Tensions and breakdowns are important (and not necessarily unproductive) events. However, issues may be introduced by inadequate system's interfaces, which is the aspect we are interested in here. More precisely, our concern was related to the fact the interfaces may over-constrain learners' work and organization, and render bottom-up transitions difficult.

The adopted design appears not to over-constrain transitions thanks to the fact that the co-operation and co-ordination levels are, in fact, intertwined, and, more precisely, to the following properties: the interfaces are similar (quasi-identical); if something is modified in the organization, the execution interface is automatically adapted; only the items that have been changed in the organization are modified in the execution interface (modifying the organization does not mean starting up again from scratch); learners can define and enact partial organizations; learners can go from the organization to the execution interface and *vice versa* whenever they want; revising the organization is possible whilst remaining in the execution interface.

Various points suggest this conclusion.

First, the chats analysis did not reveal any interactions suggesting they felt particularly constrained by the system (e.g., learners complaining the system imposed -or did not allow- some particular actions), and learners did not make any reference to such an issue during the informal discussions that took place after the experience.

Second, analysis of their processes revealed a variety of usages and constructions. We have already mentioned in section 6.2 different usages of the editors developed by learners. Similarly, the system does not enforce any specific solution. All the groups adopted different approaches (in terms of co-operation and co-ordination). Two of the groups using the system distributed the measurement and calculation tasks among the various individuals, while the third group built common ground (co-construction phase), but decided that a single (brilliant) learner would carry out all the calculations, only using the editor in execution mode (the other learners continued to be active, however). Analysis of the chat and, more generally, of the learners' behavior, shows that the learners did not build a plan and then implement it with the interfaces, they developed in context a usage of the interfaces adapted to the plan they developed. This led them, in some cases, not to use some of the offered features. As an example, one of the groups did not explicitly allocate tasks via the editor and used the array-editor to share results only, the learners continuously communicating by chat to inform each other of what they were doing and, when they finished the task (calculation, measure, etc.) they only then considered planning issues (opportunistic planning).

Third, learners' usage of the system is in line with the notion of "plan as a resource for action". The gathered data highlights the fact that learners use the planning elements they define via the editors and discuss via the chat as a reference. They do not hesitate to revise it, in some cases updating its representation in the system and in some cases not. They also deviate from the plan, here again explicitly (learners being aware and acknowledging the deviation) or not. As examples of in-context adaptations: adding a line/task (see line 9 in Fig. 4); not addressing tasks according to the determined level of priority (this "priority" notion is treated as informative, it does not really impact on their implementation of the plan); stopping their individual actions and reconsidering a value to be used by the various members when they realized their calculations were different (e.g., reconsidering the precise length of the track); attributing more tasks to a learner who seemed particularly efficient while the others moved onto support or verification tasks; etc. As an interesting example of how learners consider new tasks in context: L#3: "I succeeded with a car that stops and another that doesn't, they arrived together I made it"; L#2: "you did it"; L#2: "now test when the car is not on the line"; L#1: "Cris [L#3] you test we complete the array".

The fact that the system leaves open different organizations and does not enforce learners to follow their plan is in line with the design goals. The contrary would be very incoherent with the theoretical background. The fact that learners deviate from the plan, which in any case is but a certain explicitation of their strategy-related thoughts, would be a central issue if the represented plan was used as the input of a workflow that will mechanize the process. In direct contrast, within our structuring

approach, the assistance provided by the system does not consist in a control device, but in suggesting and supporting learners in designing a plan and managing it, and making salient their decisions and actions. What is at stake is not the fact that the plan is efficient or that learners stick to it, but the fact that they engage in the efforts of considering organizational issues. Elaborating a plan and monitoring their process are part of this, as is revising the plan, analyzing (during the solving or post-activities) why and how they deviated from it and did (or did not) realize they were doing so, or why they did (or did not) consider as important the updating of the representation of the plan in the system. However, this means that if using the system leads learners in considering organizational issues, it does not influence towards an efficient organization or a balanced distribution of work. If this aspect is a concern, the tutor must monitor it.

6.5 Risk #2: diminishing learners' engagement

The fact learners engage themselves in the solving is an important aspect of challenges. When designing the interfaces, one of the concerns was that leading learners to consider the over-head activity of organizing themselves might diminish the playful side of the challenge and their engagement.

Various elements suggest that the system does not negatively impact on learners' engagement. First, as reported in section 6.2, learners do use the system (which is offered to them; they can avoid using it). Second, groups using the system engage in much longer sessions (average of 4h vs. 2h50). This is significant because it corresponds to demands for more time to finalize preparations (Phase 1) or final calculations (Phase 2) and succeed. Third, we did not find in the chat any elements suggesting a negative impact.

As another source of evidence, we used the Students' Approaches to Learning questionnaire [35], which is an instrument designed by the Organization for Economic Co-operation and Development to measure 14 factors which assess self-regulated learning strategies, self-beliefs, motivation, and learning preferences. We used the part of the questionnaire relative to motivation and, in particular, the "Interest in Mathematics" factor. As examples of items: "When I do mathematics, I sometimes get totally absorbed"; "Because doing mathematics is fun, I wouldn't want to give it up"; (four-point response scale: disagree, disagree somewhat, agree somewhat, agree). These are well-developed constructs, which are unlikely to be changed by a single event. However, administering this questionnaire to learners after having worked for 2 or 3 hours, the fact that learners using the system obtain rather higher post-test scores [32] suggests that the system is not perceived as negatively impacting learners' engagement and the setting of the playful aspect.

6.6 Frequency of correct solutions

The objective of the designed interface is to engage learners in considering organization. This is different from supporting in solving the problem, and although it may

be thought organization and success are related, the relation is not straightforward.

As a matter of fact, we can mention that using the organizational tools offered by the system does not seem to lead learners to solve the problem with better results (1 success and 2 failures for groups using and not using these interfaces; similar results in the previous experimentations) [32]. However, the qualitative analysis highlights that groups using the system fail because of contingent issues (typically, a calculus mistake), when most groups not using the system engage in Phase 2 facing lack of data, of shared understanding and/or of shared strategy, and have very little chances of success.

7. DISCUSSION

The system's interfaces are an implementation of a certain number of design principles based on the theoretical model adapted to the case study and have been iteratively refined in the light of the lessons learned. We hereafter discuss *a posteriori* this construction.

Bardram's model has three main implications: (1) the conceptualization of organization in terms of co-construction, co-operation and co-ordination; (2) the fact that learners should be led to consider the activities (e.g., grounding or division of labor) related to these notions; (3) the fact that transitions are intrinsic to such settings and should be made easy and natural.

The problem considered influences the design because, at the co-ordination level, the interface must be coherent with the types of tasks to be achieved. The "array" structure is a response to the double objective of (1) making learners distribute tasks (which requires dissociating tasks, here as lines) and (2) allowing learners to share their calculations. This was induced from the way learners compiled the data on their paper sheets during the exploratory experiment. In terms of the scope of the application, this "array" presentation is well suited to problems in which learners must collect or calculate data and share it. The structure of a line (of a task) may be adapted to the considered problem.

Other design decisions do not flow from Bardram's model but, rather, are iteratively improved ideas in line with the problem-solving general model, Bardram's model and its general Activity Theory bases. For instance, the array structure does not only allow division of work and data aggregation. It also allows learners to adopt different strategies such as double or triple calculations, different criteria of allocations (by item, in this case cars; by task), and an easy and intuitive awareness of problem-solving progress. Similarly, allowing the entwining of co-operation and co-ordination is not a straightforward implication of the model: here, we have hypothesized that this general model was pertinent for short periods of time, and verified this through tests.

Tests also highlight unexpected perspectives. For instance, given the fact that learners can modify the organization whilst not coming back to the specific interface, the necessity and value of introducing by prescription or by

the interfaces, three successive phases (defining notions and tasks; allocating tasks; conducting individual work) became an open question. The tests suggested (to be confirmed) that as the array-interface is filled from the tasks learners have previously defined, its structure and functioning appeared easy to understand, which may not be the case if not delivered (learners not using it in the first stage had difficulties afterwards). This supports a decision of the original rationale which was to introduce these three phases as a way to engage learners in considering organizational issues (and in line with the problem-solving general model), and avoid consideration of the array-interface as a way to share calculation results only.

8. CONCLUSIONS AND PERSPECTIVES

Work presented in this article highlights the interest of Bardram's model to inspire design principles of interfaces supporting learners' in making their organization explicit. Of particular importance are the facts that organization is somewhat reified (here, as an array), organization and execution interfaces are similar and connected, and learners are given flexibility in operating transitions, enacting partial organizations or adapting their constructions in context. Although limited, current tests suggest the interfaces are usable, and are used consistently with expectations and goals: learners consider organizational issues as such, make their constructions explicit (common ground, subtasks, division of labor) and achieve top-down and bottom-up transitions in a way that does not appear too artificial or counter-productive for them. It is also interesting that it becomes easier to analyze learners' organizational activity, with respect to envisaging monitoring processes or using this resource in post-activities.

Perspectives are as follows.

First, results so far suggest that learners' engagement is not negatively impacted and, rather, is enhanced. This needs to be explored statistically. If confirmed, it opens up the perspective of exploring such problem-solving mediation as a means to support learners' engagement.

Second, system-based indicators appear to be a fairly liable basis for analyzing learners' activity. This allows considering building analysis-tools for tutors. Such tools can be based on (1) learners' use of the system editors [32] and (2) learners' chat interactions analyzes.

Third, leading learners to consider organization has necessarily an impact on learners' activity. Results so far suggest that the system's principles leads learners to consider organizational issues whilst not imposing a given plan, which was part of the goals, and leads to knowledge-generative interactions related to organization (co-construction of a common ground and strategies; mutual regulation; resolution of conflicts such as breakdowns). Different questions may be investigated: does the fact that learners are led to make their organization explicit change the nature or volume of their knowledge-generative interactions? Does it provide more learning affordances for post-activities? Does it allow organizational productive failures? How does it compare with the

strategy that consists in offering learners with more structuring support such as organizational patterns elicited from specific instructional design?

Finally, another interesting question is the relationship between the facts that (1) learners make their organization explicit, (2) learners solve the problem in an efficient way and (3) learners develop learning outcomes, which is not straightforward. First, it is important to bear in mind that, in CSCL, the collective aspect is often introduced as a way to lead learners to interact, and not as a means to improve problem-solving efficiency. Constraints may be introduced to make solving more difficult but richer in terms of interactions. Efforts to make learners consider organization are thus not to be confused with efforts to support problem-solving, although this may be the argument with which learners are presented. Moreover, part of the learning may occur during the solving phase and part during the debriefing phase. Addressing successfully the challenge may thus be less important than the fact that pertinent interactions occur during the solving, and that the setting enactment provides interesting learning affordances for the debriefing phase. We have also mentioned the potential value of productive failures. However, the fact that learners have the feeling they are going to succeed or anticipate failure does affect engagement. At this level, we may mention that when considering a meta-level skill such as organization and using a problem-solving challenge as a pretext, the fact that learners may fail because of details (e.g., calculus errors) is very negative. A drawback of the case study we used is that "a miss is as good as a mile" (learners are very disappointed when the animation highlights that they failed, whatever the difference on the final line is) while, in fact, they may have developed a fruitful collaboration but made a measure or calculus error. Checking the solution with the animation is fun and participates in the motivation but denotes an aspect of "success" that is impacted by not very important issues. The setting should provide, in addition, feedback related to key features such as the fact that learners have interacted, constructed a shared understanding, developed a rational strategy or adjusted their processes in context.

Acknowledgements. The authors wish to thank the three reviewers for their help in focusing and improving this article.

References

- [1] P. Dillenbourg, "What do you mean by collaborative learning?," *Collaborative-learning, Cognitive and Computational Approaches*, P. Dillenbourg, ed., Oxford: Elsevier, pp. 1-19, 1999.
- [2] L. Kobbe, A. Weinberger, P. Dillenbourg, A. Harrer, R. Hämmäläinen, P. Häkkinen and F. Fischer, "Specifying Computer-Supported Collaboration Scripts," *Int. J. of Computer-Supported Collaborative Learning*, 2(2-3), pp. 211-224, 2007.
- [3] I. Kollar, F. Fischer and F.W. Hesse, "Computer-supported cooperation scripts - a conceptual analysis", *Educational Psychology Review*, 18(2), pp. 159-185, 2006.
- [4] P. Tchounikine, N. Rummel and B.M. McLaren, "Computer Supported Collaborative Learning and Intelligent Tutoring Sys-

- tems," *Advances in Intelligent Tutoring Systems*, R. Nkambou, R. Mizoguchi, J. Bourdeau, eds., SCI 308, Berlin: Springer-Verlag, pp. 447-463, 2010.
- [5] P. Jermann, "Computer Support for Interaction Regulation in Collaborative Problem-Solving," PhD Dissertation, University of Geneva, 2004.
- [6] P. Moguel, P. Tchounikine and A. Tricot, "Supporting Learners' Organization in Collective Challenges", *Proc. of the European Conf. on Technology Enhanced Learning (EC-TEL'2008)*, Maastricht, NL, pp. 290-303, 2008.
- [7] J.H. Larkin and F. Reif, "Understanding and teaching problem solving in physics," *European J. of Science Education*, 1, pp. 191-203, 1979.
- [8] C. Lerch, "Control decisions and personal beliefs: their effect on solving mathematical problems," *J. of Mathematical Behavior*, 23, pp. 21-36, 2004.
- [9] J. Zhang, "A distributed representation approach to group problem solving," *J. of the American Society for Information Science*, 49, pp. 801-809, 1998.
- [10] G.L. Chang and G. Wells, "The literate potential of collaborative talk," *Int. Oracy Conv.*, Norwich, England, 1987.
- [11] M.J. Baker, T. Hansen, R. Joiner, D. Traum, "The role of grounding in collaborative learning tasks," *Collaborative Learning: Cognitive and Computational Approaches*, P. Dillenbourg, ed., Amsterdam: Pergamon / Elsevier Science, pp. 31-63, 1999.
- [12] A. Soller, A. Martínez-Monés, P. Jermann and M. Muehlenbrock, "From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning," *Int. J. of Artificial Intelligence in Education*, 15 (4), pp. 261-290, 2005.
- [13] J. Bardram, "Designing for the Dynamics of Cooperative Work Activities," *Proc. Conf. CSCW'98*, Seattle, pp. 89-98, 1998.
- [14] J.C. Burguillo, "Using Game-Theory and Competition-based Learning to stimulate student motivation and performance," *Computers & Education*, 55 (2), pp. 566-575, 2010.
- [15] C.E. Hmelo-Silver, "Problem-based learning: What and how do students learn?," *Educational Psychology Review*, 16(3), pp. 235-266, 2004.
- [16] E. Barbeau, "A classification of challenges," ICMI, Study Number 16, Challenging Mathematics in and beyond the Classroom, Trondheim, Norway, 2006.
- [17] J. Roschelle and S. Teasley, "The construction of shared knowledge in collaborative problem solving," *Computer Supported Collaborative Learning*, C.E. O'Malley, ed., pp. 69-97. Springer-Verlag, Heidelberg, 1995.
- [18] R. Pea, "Practices of Distributed Intelligence and Designs for Education," *Distributed cognitions: Psychological and educational considerations*, G. Salomon, Ed., Cambridge Univ. Press, 1993.
- [19] B.J. Zimmerman, "Self-regulated learning and academic achievement: An overview," *Educational Psychologist*, 25, pp. 3-17, 1990.
- [20] P. H. Winne and A. F. Hadwin, "Self-regulated learning and sociocognitive theory," *International encyclopedia of education*, P. Peterson, E. Baker and B. McGaw, Eds., vol. 5, pp. 503-508. Amsterdam: Elsevier, 2010.
- [21] M. Kapur and C.K. Kinzer, "Productive failure in CSCL groups", *Int. J. of CSCL*, 4 (1), pp. 21-46, 2009.
- [22] P. Dillenbourg and P. Jermann, "SWISH: A model for designing CSCL scripts," *Scripting Computer-Supported Collaborative Learning - Cognitive, Computational, and Educational Perspectives*, F. Fischer, I. Kollar, H. Mandl and J.M. Haake, eds., New York: Springer, 2007.
- [23] P. Dillenbourg, "Over-scripting CSCL: The risks of blending collaborative learning with instructional design," *Three worlds of CSCL. Can we support CSCL*, P.A. Kirschner, ed, Heerlen, Open Universiteit Nederland, pp. 61-91, 2002.
- [24] P. Dillenbourg and P. Tchounikine, "Flexibility in macro-scripts for CSCL," *J. of Computer Assisted Learning* 23(1), pp. 1-13, 2007.
- [25] K. Schmidt and L. Bannon, "Taking CSCW Seriously: Supporting Articulation Work," *CSCW*, 1(1-2), pp. 7-40, 1992.
- [26] P. Tchounikine, "Operationalizing macro-scripts in CSCL technological settings," *Int. J. of Computer-Supported Collaborative Learning*, 3(2), pp. 193-233, 2008.
- [27] L. Kobbe, A. Weinberger, P. Dillenbourg, A. Harrer, R. Hämäläinen, P. Häkkinen and F. Fischer, "Specifying Computer-Supported Collaboration Scripts," *Int. J. of Computer-Supported Collaborative Learning*, 2(2-3), pp 211-224, 2007.
- [28] T. de Jong, W.R. Van Joolingen and C.A. Grant, "Scientific discovery learning with computer simulations of conceptual domains," *Review of Educational Research*, 68 (2). pp. 179-202, 1998.
- [29] J. Bardram, "Plans as situated action: an activity theory approach to workflow systems," *Proc. of the fifth conf. on European Conf. on Computer-Supported Cooperative Work, ECSCW'97*, Lancaster, UK, pp. 17-32, 1997.
- [30] B. Kramarski, "Making sense of graphs: does metacognitive instruction make a difference on students' mathematical conceptions and alternative conceptions?" *Learning & Instruction*, 14(6), pp. 593-619, 2004.
- [31] P. Moguel, P. Tchounikine and A. Tricot, "A Model-Based Analysis of the Organization of Students Involved in a Computer-Based Pedagogical Challenge," *Proc. of the 8th Int. Conf. on Computer Support Collaborative Learning (CSCL'09)*, Rhodes, GR, pp. 73-77, 2009.
- [32] P. Moguel, P. Tchounikine and A. Tricot, "Supporting Learners' Self-organization: An Exploratory Study," *Int. Conf. on Intelligent Tutoring Systems (ITS'2010, Pittsburg - USA)*, LNCS 6095, Springer, p. 123-133, 2010.
- [33] P. Tchounikine, *Computer Science and Educational Software Design. A resource for multidisciplinary work in Technology Enhanced Learning*, Springer-Verlag, Berlin, Heidelberg.
- [34] www.patrickmoisan.net/copains/course_sans_gagnant.html.
- [35] H.W. Marsh, K. Hau, C. Artelt, J. Baumert and J.L. Peschar, "OECD's brief self-report measure of educational psychology's most useful affective constructs: Cross-cultural, psychometric comparisons across 25 countries," *Int. J. of Testing*, 6(4), pp. 311-360, 2006.

Patrice Moguel, PhD, is a mathematics teacher in Andorra.
Pierre Tchounikine, PhD, is Professor of Computer Science at the University of Grenoble (France). He leads a pluridisciplinary team, dedicated to design and analysis of computer-based learning environments.
André Tricot, PhD, is Professor of Psychology at the University of Toulouse (France). He heads a team named "Learning, Motivation, Metacognition" in the CLLE-LTC laboratory.