



Design for classroom orchestration



Pierre Dillenbourg

Ecole Polytechnique Fédérale de Lausanne (EPFL), CRAFT Station 20, 1050 Lausanne, Switzerland

Keywords:

CSCL
Classroom
Orchestration
Teacher
HCI
Distributed cognition

1. Introduction

Orchestration refers to how a teacher manages, in real time, multi-layered activities in a multi-constraints context. Many pedagogical scenarios integrate individual activities (e.g. reading), teamwork (e.g. problem solving) and class-wide activities (e.g. lectures). Some of these activities are computer-based, some not; some are face-to-face while others are online. This pedagogical integration is mirrored by the technical integration of different tools (simulations, quizzes, wikis, etc.) distributed over multiple artifacts (laptops, sensors, tablets). These integrated scenarios require forms of management referred to as orchestration.

Orchestration originates with some frustration. Why are technologies under-exploited in schools, despite the fact that, in Western countries, computers and Internet access are ubiquitous, teachers are computer literate and educational software is available? Obviously, teacher training and learning culture are key explanatory factors. But, let me for instance consider the necessity for learners to login into a system: do the 3–5 min lost for login in (e.g. due to forgotten passwords), i.e. up to 10% of a standard lesson, bring a pedagogical added-value around 10%? Instead of blaming teachers and institutions, it makes sense to ask if is there something about the technology we develop that discourages its usage? This paper addresses formal education¹ from an evolutionary hypothesis (technologies could incrementally improve school efficiency), not from a revolutionary hypothesis (technologies could radically change schools).

2. Is orchestration different from instructional design?

2.1. Extrinsic activities

Designing effective learning scenarios is and will remain a priority. However, classroom life is populated by activities or events that are not part of the scenario, such as collecting response sheets. There is indeed a continuum of activities from those intrinsic to the scenario to activities extrinsic to learning. From the center to the periphery, this continuum includes the following:

1. *Core activities (designed as adaptive)*: The activities of the scenario are pre-defined, with certain adaptations to be performed by the system or by the teacher. Adapting the activities to the learner behavior is the first layer of orchestration, but is not novel. What may be novel is that orchestration tools collect data that is simple enough to analyze that the teacher can cope with information about twenty or thirty students. Therefore orchestration tools tend to produce shallow learner modeling such as whether the student is active or attentive.
2. *Emergent activities (designed as contingent)*: Some scenarios include activities whose contents are not predictable because they build upon what learners have produced in earlier phases of the scenario. Examples of 'debriefing' activities are presented hereafter. These activities have been designed before the lesson, but require real time elaboration using students' productions. They are hence demanding in terms of orchestration.

E-mail address: pierre.dillenbourg@epfl.ch.

¹ I do not address informal learning, nor virtual classrooms.

3. *Envelope activities (routinized)*: Some classroom activities do not belong to the pedagogical design but to established school practices. For instance, students often have to copy what the teacher wrote on the blackboard. This time-consuming activity answers to constraints (see hereafter) such as leaving a trace of learning activities for the students or their parents. Another example is multiple testing: the fact that students reach the objectives once does not mean the end of the activities. In schools, the same content is usually covered in multiple sessions, through summaries, homework, revision, synthesis and feedback sessions.
4. *Extraneous events (unavoidable)*: A designed scenario inevitably encounters unexpected events such as: (i) Team members are assigned complementary roles but one team member drops out of the course half way through (ii) During a guided discovery activity, the first child who find the solution is so proud that he shouts the answer aloud for the whole class (iii) Sophie was sick last week and missed the first half of the lesson, (iv) A crane stops in front of the class window... Do technologies enable the teacher to adapt to these events in real time, and hopefully in a positive way (e.g. by using the crane to teach rotation).
5. *Infra activities (necessary)*: Some activities do not constitute a meaningful part of the scenario but are necessary to run it: finding the right document, or moving chairs and tables to set up teams. These logistic details may waste precious time, and, if they fail, they can even spoil the best pedagogical design. If students spend 3 min to log in to a computer, do the functionalities that require the log in compensate for the lost time?

Learning technologies are designed for intrinsic activities (1–2), so how can they be designed for extrinsic ones? I do not suggest neglecting intrinsic activities, but extending design to the external ones.

2.2. Extrinsic constraints

Instructional design has to cope with 3 intrinsic constraints: the contents to be taught (what), the learner's characteristics (who) and how brains build knowledge (how). This is not new. However, beyond these intrinsic constraints, teachers also have to cope with several extrinsic constraints:

1. *Time constraints*: Time is a scarce resource in teaching. For instance, teachers often blame constructivist methods for being too time consuming. In addition, a class is often segmented into slices of 50 min: time management is permanent concern of teachers.
2. *Curriculum relevance*: Don't we often design environments targeting skills that are not in the school curriculum, because they suit our research agenda? Teachers are not free to teach whatever they like.
3. *Discipline constraints*: School directors and parents, expect the teachers to be in control of their students and have a reasonable level of discipline: some animation is expected in classrooms, but no chaos or violence.
4. *Assessment constraints*: Beyond the usefulness of assessment, schools are driven by the need to provide grades. A good pedagogical scenario may be abandoned if this is not the case. For instance, teachers often criticize collaborative projects because it is difficult to give individual grades.
5. *Energy constraints*: The total effort a teacher may invest (preparation work, time to provide feedback, etc) is limited. If we design for heroes, we lose scalability (there are few heroes) and sustainability (heroes get tired).
6. *Space constraints*: Is there enough space in the classroom to set up activities and is the layout compatible with expected social interactions or to the work format (e.g. teams)?

Still more extrinsic constraints exist including finances, teacher's self-esteem, safety, school culture, etc; however, integrating these 6 constraints into technology design would already constitute a major step.

3. Is orchestration different from adaptive instruction?

At an abstract level, orchestration is a regulation process similar to adaptive (individualized) instruction: monitoring the situation, deciding what adaptations are necessary and then performing these adaptations. In adaptive instruction, however, this loop is rather closed. In orchestration, on the other hand, the loop is very open, for two reasons. First, adaptive instruction means only adapting intrinsic activities to intrinsic constraints while orchestration involves adapting both intrinsic and extrinsic activities to both intrinsic and extrinsic constraints. Second, in adaptive instruction,² the variability comes from one single learner while in orchestration it comes from the behavior of more actors (e.g. 25 learners) plus external events. Another difference between adaptation and orchestration is the 'balance of control,' i.e. who among the system or the actors performs the regulation. Design aims to find the optimal allocation of regulation among the system and the actors. The orchestration metaphor ('teacher as a conductor') favors asymmetrical distributions, in which the teacher takes the ultimate decision. In other words, orchestration is a form of educational regulation but with a different scope, which leads to different design choices. I illustrate these choices with four integrated learning environments.

4. Examples

Here, I briefly present four learning environments that I will use to illustrate orchestration features.

1. 'ConceptGrid' is a variation of the JIGSAW script (Dillenbourg & Hong, 2008): students read different papers in which specific concepts are present; they then have to provide a definition of these concepts and assemble them into a grid (a simplified concept map). In the last phase, the teacher asks the students to explain their definitions and justify why they connected certain concepts to each other. He then integrates these concepts with the theoretical framework that students have to acquire.

² I do not mean that individualized instruction is easy to achieve.



Fig. 1. The Lantern (left) and TinkerLamp (right).

2. 'ArgueGraph' is an argumentation script (Dillenbourg & Hong, 2008). First, students answer a questionnaire individually. On the basis of their responses, the system builds a map of students' opinions and forms pairs of students with opposite opinions. Selected pairs have then to answer the same questionnaire together. The script ends with a debriefing session.
3. 'Lantern' (Fig. 1 left) is a tool designed for recitation sessions at the university level (Alavi et al., 2009). Typically a weekly course in physics is composed of 2 h of lecture plus 2 h of exercises. During the exercise session, students work in small groups on a set of problems. When they are stuck, they raise their hands and one teaching assistant (TA) comes when (s)he is available. This orchestration is fairly simple, yet it is far from optimal: while waiting for the TA, students spend 62% of their time visually chasing the TA, instead of working. Another observed orchestration problem concerns unanswered questions (students give up). By rotating the Lantern cover, students indicate which exercise they are working on. The height of the color bar indicates how much time has been spent on the current exercise. To call the TA, they press the Lantern, and it starts blinking. The blinking rate increases slowly indicating the waiting time.
4. 'TinkerLamp' (Fig. 1 right) is an augmented-reality tabletop simulation for training apprentices in logistics (Zufferey, Jermann, Lucchi & Dillenbourg, 2009). These apprentices (16–20 years old) spend four days per week in a company and one day at school. They have to learn how to optimize storage surfaces based on the frequency of product sales, how to manage their stock, etc. Teams build the mock-up of a warehouse by placing small plastic shelves on a table. The system includes a camera that recognizes the visual markers on the shelves and a beamer that overlays information such as shelf content and forklift movements.

5. Orchestration functionalities

These examples illustrate how extrinsic activities/constraints have been integrated into design:

5.1. Enabling the activities

1. *Core activities*: These integrated scenarios can be depicted as a graph in which the nodes are activities and the links are dataflow between individual, group, and class activities. For instance, individual answers to the ArgueGraph questionnaire are used to form groups and are collected by pairs for supporting pair argumentation and for the whole class for supporting the debriefing. A workflow transfers data along these arcs and performs operations such as aggregating data from social plane³ N to $N + 1$ or, conversely, distributing data from N to $N - 1$. When students manipulate physical objects (tangibles, paper sheets): the digital flow mirrors the physical flow, maintaining consistency within a kind of double workflow.
2. *Emergent activities*: In the last phase of *ConceptGrid*, the teacher asks students to explain their definitions and justify why they connected certain concepts. The workflow collects and displays all definitions and grids. The teacher can navigate by grid (each team has produced a grid), by concept across the different grids, etc. Before the session, he may highlight interesting answers (e.g. frequent mistakes) with different colors in order to find and refer to them rapidly during his lecture. In *ArgueGraph*, the debriefing tool displays all elements introduced by individuals and pairs for justifying their answers. The teacher may navigate question-by-question, answer-by-answer. (S) he can ask students to further explain their choice and reformulate their arguments with proper concepts; he can relate them to theories, etc. The system also displays who has changed opinion between the individual and the collaborative phase: asking students what made them change their minds is a good orchestration trick. In *TinkerLamp*, when students save a warehouse model, these models are collected and projected on the 'TinkerBoard' (Fig. 3). This debriefing tool supports comparisons: the teacher may select two models and the system compares their features (surface, performance).
3. *Envelope activities*: Homework and exams are not often integrated into learning technologies but play a major role in student life. When using *TinkerLamp* at school, students save warehouse models, from which the system prints a homework sheet (Fig. 2 left). Students then have to determine in a discussion with their supervisor which of the warehouses is most similar to the one in their company and why. When using *ConceptGrid*, students are informed that their grids will be available during their exam, an effective form of extrinsic motivation.

³ The individual, group and class levels.

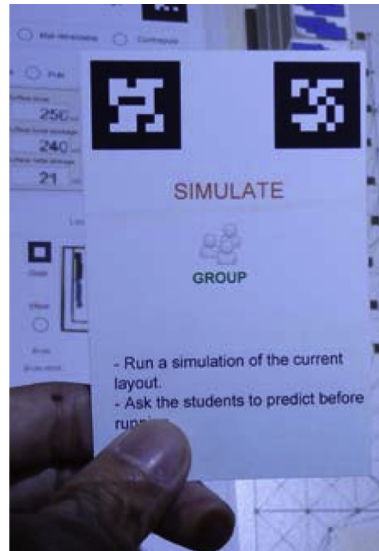


Fig. 4. Paper orchestration cards to control the simulation of a team or the whole class.

definitions produced by the student playing role-X in any other team. In both cases the intrinsic activity is somehow less optimal but at least the scenario can be conducted.

5. *Infra activities.* Instead of logging in to a TinkerLamp and to search for the right exercise, teachers use a binder in which the curriculum has been turned into sheets (Fig. 2, right). The markers on the sheet tell the system which exercise has to be run. Sparing 2 min 5 times per lesson actually saves 20% of the lesson time.

5.2. Monitoring the activities

Monitoring is less ambitious than student modeling: it does not aim to infer about the knowledge state of a learner but rather to provide teachers with simple indicators of activity for all students.

1. In ArgueGraph, the system displays in real time how many questions each student has answered, so that the teacher may 'push' the slower ones. It also displays the number of words per justification, so that the teacher may invite the minimalists to enrich their responses.
2. Lantern is essentially a tool for real time regulation, but it also supports delayed regulation: all user data is sent to the server which produces a summary of the average working time, waiting time and helping time per exercise. With this overview, the teacher may realize that an exercise is too difficult or that more TAs are required.
3. In TinkerLamp, the teacher may glance at the tables and, since the interface is made of physical objects, see how many shelves are placed. Moreover, since teams tend to run the simulations frequently, without reflection, TinkerBoard display a bar that indicates this frequency with a color scale (ranging from yellow to red). In the first example, the interface design makes the behavior visible; in the second one, the designed turned invisible parameters (frequency) into a visible artifact.

These activity-monitoring cockpits do not replace student modeling, but rather add onto it. A good teacher will always need to know what a student does and does not understand, but this diagnosis is only tractable at the individual level while orchestration is at the class level. Orchestration, at the other hand, is at the class level.

5.3. Adapting the activities

Any adaptation developed in 'adaptive instructions' is relevant: adding or skipping exercises or examples, changing the difficulty level, etc. Orchestration requires other forms of adaptation as well, which are illustrated hereafter:

1. In the previous examples of debriefing, the technology facilitates the teacher in the task of adapting the content of his lectures to the student productions, by aggregating data at various levels.
2. The 'joker' option in ConceptGrid and the 'bypass' option in ArgueGraph illustrate how to the teacher can adapt the scenario to missing students.
3. Since TinkerLamp users tend to run simulations without reflection, we developed a *paper orchestration key* (POK- Fig. 4) that inhibits the simulation. Teacher carries the keys in his hands. When apprentices call him to run a simulation, he can ask them to predict if the warehouse performance⁴ will be higher or lower than in the previous run, and why. If he is satisfied, the teacher shows the card to the camera and the team may proceed. The teacher may decide to leave the card with a team that does not need to be regulated. When the

⁴ The warehouse performance is the average time to move a box from the shelves to the truck.



Fig. 5. Two designs of the TinkerLamp, offering different levels of visibility.

teacher needs to give a collective explanation, it is very hard to get the attention of all apprentices. Therefore, another POK allows him to simply switch off (and on) the four lamps displays used in the class.

6. Design principles

The features illustrated so far have not been generated from a theory. They emerged through frequent interactions with teachers while designing activities and from experiments in which these activities were, as often as possible, conducted by the teachers. However, some commonalities appear across these examples that I retroactively extract as design principles.

1. *Control* refers to the fact that the teacher is the driver, in control of what happens in the classroom. In the work of Nussbaum et al. (2011), about 40 children interact with a mouse on a single display. Each student owns a tiny subset of the display area, as small as a phone display. The same activity could be conducted on a PDA, but the teacher would not be able to maintain control of the classroom. By being in the middle, in front of the centralized display, he has more control. In terms of design, control is firstly and foremost related to flexibility: The teacher's decision should supersede any system decision, because the former has contextual information while the latter does not. Secondly, control refers to the awareness of what is happening in the classroom, not only within the designed learning environment but also around it: for instance, are the students reading their email instead of working? The POK that freezes all students displays is a radical example of a control tool. A subtler example is the design of the TinkerLamp hardware (Fig. 5). With apprentices, the right one is better because it allows the teacher to visually scan the whole class, constantly. The left one could be appropriate for primary schools, since its opacity may create desired intimacy in a corner of the classroom.
2. *Visibility* is a factor of control, as illustrated with the shape of TinkerLamps. Visibility can be achieved by reifying – making the invisible visible – and by materializing – replacing digital objects with physical ones. Reification is illustrated by Lantern: waiting times are made visible by the color intensity and height. Materialization⁵ is illustrated by TinkerLamp: when glancing around, the teacher sees what is happening on the table, which would not be the case if the student was using a virtual version on a laptop. Let me stress that visibility is not a one-way process, from the student to the teacher (as in monitoring tools), but rather visibility for all actors in the classroom: Lantern allows students to see that other teams were struggling with the same exercise; TinkerLamp allows students to glance at the models built by peers on the next table⁶. The word 'glancing' stresses that visibility is not about carefully watching what someone is doing, but rather providing actors with awareness of what is happening in the classroom. Design is therefore about providing the simplest visualizations, as in our examples.
3. *Flexibility* refers to the possibility for a teacher to change, on the fly, any decision made either by himself or the designer when preparing the activity. As illustrated with the 'joker' or 'bypass' functionalities, flexibility is not only adapting the scenario to the learners' needs, but also reacting appropriately to extrinsic events and constraints. Time flexibility is another example: does the technology allow students to stop at any point and resume work in the next course? Flexibility is a relative concept: not all design decisions can be modified without losing the pedagogical value of the scenario (Dillenbourg & Tchounikine, 2007). There is a tension between flexibility and workflow, since the latter is about automatizing flows. But, fundamentally, orchestration can only occur if the teacher has the freedom to change activities on the fly.
4. *Physicality*. So far, educational research has ignored the shape of computers. The previous point illustrated the shape of the TinkerLamp affects orchestration. Its color also influenced orchestration. In school experiments, the teacher placed four TinkerLamps in the classroom. Each lamp had a different color head, which enabled the teacher to easily refer to any team. Several configurations of the TinkerLamp positions in the classroom have been explored to optimize the way the teacher could walk from one to another. Even the spatial organization of information matters: we compared the Lantern with a tool providing the same information but in a different way. Instead of using Lantern, students indicated, with clickers, which exercises they were currently working on and if they needed help. This information displayed (the same color codes as on Lantern) was presented on a central display. These two tools provide the same status updates but generate different social processes: the central display induced competition between teams, which was not the case with

⁵ An early example was given by Roschelle and Pea (2002): when a student walks across the classroom to share PDA data with infrared instead of sending it wirelessly, this publicly visible walk makes the teacher and other students aware of the dataflow.

⁶ These inter-team interactions, surprisingly neglected in CSCL, are actually a part of classroom orchestration: should the teacher, for instance, encourage these types of interactions or inhibit them?

Lantern. Physicality is also revealed by our examples through the use of tangibles and paper as an interface. Orchestration is physical: it is about mobility, gaze and distance between actors.

5. *Minimalism* is almost a meta-principle since it applies to all of the previous principles. In Lantern, we could have displayed the exact waiting time, with numbers, rather than a fuzzy color intensity. Our design deliberately reduces the resolution of information. In TinkerLamp, the POKs are very easy for the teacher to manipulate as long as there are between 5 and 10. With more than 10 cards, finding the right card would be time consuming. Our monitoring tools are minimalist compared to the complex visualization developed in the field of educational data mining because these tools are designed to be used in real time. *Less is (sometimes) more*. In learning theories, a lot of focus has been on the cognitive load of the learner, but what about the cognitive load of the teacher (orchestration load)? Educational psychologists aim to improve the usability of instructional software to minimize extrinsic load and foster learners' resources on germane and intrinsic load. Similarly, orchestration tools should minimize extrinsic orchestration load, i.e. any complexity accidentally created by the technology. This is how we reached the notion of classroom usability (see synthesis).

6.1. The case of paper

The use of paper as interface for the TinkerLamp emerged through co-design with teachers. Retroactively, it corresponds the five design principles listed above. The ability to place multiple sheets on the table or to keep several POKs in a pocket allows high *flexibility*, since they can be used in any order and distributed in multiple ways across classroom actors, re-allocating roles in various ways. Paper sheets make the workflow *visible*: their physical manipulation in the classroom makes the actors aware of who, for instance, is giving what to whom. A physical workflow is easier to modify in real time than a virtual workflow: the teacher can just redistribute the paper, rather than modifying options in a system. The homework sheets in Tinker “worked very well”: while teachers told us their apprentices never do their homework, 90% of the students completed it and 82% even discussed their sheet with their supervisor. Thanks to paper, this activity was easy to completed during a short break, less formally than setting a meeting with the supervisor in front of a computer. Curriculum sheets illustrate *physicality* because the curriculum constraints are turned into a concrete object. Teachers can annotate these sheets before the lesson, or after the lesson to remember specific elements for the next year or perhaps copy them to a colleague. From a usability viewpoint, paper is already adapted to the routine of classroom life: distributing, collecting, storing or annotating sheets.

The affordances of paper relate to the work of Hutchins (1995) on paper cards used in aircraft cockpits. How information is distributed over and flows between the different actors and artefacts of cockpits explains the relevance of this type of paper in a cockpit. Despite the differences between a cockpit and a classroom, this analysis also applies to the paper sheets and cards we used in classrooms. Our flow of information is somehow richer since our paper sheets actually overlay 3 layers of information: what has been printed, what is projected and what is handwritten. I do not glorify paper (cockpits are fortunately not entirely made of paper); my point is that sometimes paper has specific affordances within a distributed flow of information.

7. Synthesis

Designing for orchestration implies facilitating the dual flow of information in a classroom, across digital and physical information containers. This workflow integrates learning activities as well as other classroom activities (emergent, envelope, extraneous and infra). It takes into account that, besides learning, teachers face many constraints: time, discipline, curriculum, energy, ... Some design principles seem to meet these requirements: flexibility, control, visibility, minimalism, and physicality.

These elements do not yet constitute a theory, but distributed cognition is a good approach to explain why workflows that satisfy these principles may decrease the teacher's orchestration load. While minimizing cognitive load is the goal of the usability studies at the individual level, minimizing orchestration load constitutes a form of usability at the classroom level, a ‘third circle of usability’ (Dillenbourg et al., 2011). The classroom is viewed as a user in the same way Hutchins (1995) referred to a “cockpit that remembers its speeds”.

I did not use the metaphor of the orchestra conductor on purpose. *I prefer to drop this metaphor because it generates unproductive debates and to use ‘orchestration’ as a concept on its own*. For me, it is a rather technical concept, although it conveys two flavors. The pedagogical flavor is the empowerment of teachers as drivers of classroom activities. This does not mean more lecturing; even constructivist approaches require smart leadership. The technological flavor is minimalism: technologies that are not ‘intelligent’ in the AI sense, that have simple effects such as making visible things that would otherwise be invisible. We may call this ‘modest computing’.

Acknowledgements

This work has been partly funded by STELLAR, a European Network of Excellence on TEL, Dual-T, the Swiss leading house on technologies for vocational training; and RedLink, an NSF project PDFM1—118708. Thanks to P. Jermann, F. Kaplan, G. Zufferey, S. Do Lenh, Q. Bonnard, S. Cuendet.

References

- Alavi, H. S., Kaplan, F., & Dillenbourg, P. (2009). Distributed awareness for class orchestration. In U. Cress, V. Dimitrova, & M. Specht (Eds.), *Proceedings of the 4th European conference on technology enhanced learning, Nice, France* (pp. 211–225). Springer. EC-TEL 2009, Sept. 29th – Oct. 2nd, LNCS 5794.
- Dillenbourg, P., & Hong, F. (2008). The mechanics of CSDL macro scripts. *International Journal of Computer-Supported Collaborative Learning*, 3(1), 5–23.
- Dillenbourg, P., & Tchounikine, P. (2007). Flexibility in macro CSDL scripts. *Journal of Computer Assisted Learning*, 23(1), 1–13.
- Dillenbourg, P., Zufferey, G., Alavi, H., Jermann, P., Do-Lenh, S., Bonnard, Q., et al. (2011). Classroom orchestration: the third circle of usability. In *Proceedings of the 9th international conference on computer supported collaborative learning, Hong Kong, China*, Vol. 1, (pp. 510–517).
- Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science*, 19, 265–288.
- Nussbaum, M. C., Alcolado, A., Tagle, F., Gomez, F., Denardin, H., Susaeta, M., et al. (2011). One mouse per child: interpersonal computer for personal formative assessment. *Journal of Computer Assisted Learning*. <http://dx.doi.org/10.1111/j.1365-2729.2011.00438.x>.

- Roschelle, J., & Pea, R. (2002). A walk on the WILD side: how wireless handhelds may change computer-supported collaborative learning. *International Journal of Cognition and Technology*, 1(2), 145–168.
- Zufferey, G., Jermann, P., Lucchi, A., & Dillenbourg, P. (2009). TinkerSheets: using paper forms to control and visualize tangible simulations. In *Proceedings of the 3rd international conference on tangible and embedded interaction (Cambridge, UK, Feb, 16-18, 2009). TEI '09* (pp. 377–384). New York, NY: ACM.