Computers & Education 69 (2013) 485–492

Contents lists available at SciVerse ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

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 education[1] 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.

0360-1315/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compedu.2013.04.013

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' 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.

Download English Version:

https://daneshyari.com/en/article/6835506

Download Persian Version:

https://daneshyari.com/article/6835506

Daneshyari.com