This document is a personal draft of the paper published as Tchounikine, P. (2013) Clarifying Design for Orchestration: Orchestration and Orchestrable Technology, Scripting and Conducting. Computers & Education. DOI: 10.1016/j.compedu.2013.04.006. It is enhanced with a Figure of the SPAIRD model.

Clarifying Design for Orchestration: Orchestration and Orchestrable Technology, Scripting and Conducting

Pierre Tchounikine
LIG, Université de Grenoble 1, France

1. Introduction

The orchestration metaphor usually refers to "what the person in the middle of the orchestra (transposition: classroom; on-line setting) is doing with the help of the baton and gestures (transposition: with instructions, hints or by tuning the computer-based system's parameters)".

This metaphor is potentially confusing as, according to some definitions, orchestration seems to be essentially, if not only, about writing music for an orchestra (i.e., deciding how some music should be played by a set of instruments), directing performance (i.e., directing the way the musicians perform the orchestration) being referred to as "conducting".

Dillenbourg proposes a definition: "Orchestration refers to how a teacher manages in real-time multi-layered activities in a multi-constraints context" (Dillenbourg 2012). This clarifies the issue. This issue is not new, and many topics related to the notion of orchestration have been studied, with different perspectives and/or in different contexts, in previous works (although they were not presented in this way). However, it is beneficial to make orchestration a key issue of the field once again, and give it a definition that is in-line with current uses of technology in education.

Dillenbourg highlights the fact that settings now involve different modes (individual and collective, in class and on-line, etc.). I would add the fact that ICT has changed the context in different ways, due to the fact that players (e.g. teachers, students) come with their technology and their technological skills and habits, and are not dependent (and do not want to be dependent) on technologists. Educators asking professional developers to build software specific to the targeted setting, or students limiting themselves to the use of software offered by the institution, are the exception rather than the norm. Players tend to use whatever tool they will contextually find available and convenient - the applications running on their smart phones, software they have installed on their laptop, available web services, etc. - which may vary from player to players and from session to session. Within such an approach, in some sense, software is a contextual construction. Moreover, although in some cases the setting involves specifically designed software, users' usages and expectations are influenced by this general evolution (Tchounikine 2011).

Orchestration as introduced by Dillenbourg is mainly about empowering human teachers. The orchestration concept is not limited to the context of classrooms and orchestration by teachers. Nevertheless, the way technology is used in classrooms and the specificity of this context gives importance to this issue.

Considering how orchestration may involve and/or be supported by computer-based systems requires, however, differentiating different notions.

2. Orchestration technology and orchestrable technology

In order to make clearer what "designing for orchestration" means, I propose to distinguish *orchestration technology* from *orchestrable technology*:

- Orchestration technology is technology that achieves or supports the activity of orchestrating.
- Orchestrable technology is technology which use can be decided or adapted (before the session and/or at run-time) by the players in charge of the orchestration (the teacher, a system) while

orchestrating the setting, in the same way that other parameters of the setting (the timing, the groups, the task, the physical space, the teaching objectives, etc.) may be adapted.

Orchestration technology may correspond to different realities such as:

- Technology that provides teachers with some support for managing the setting. Examples: technology that renders salient some dimensions of the setting that is of interest for orchestration (e.g. tangible interfaces rendering learners' workflow salient (Dillenbourg 2012)); technology that provides teachers with monitoring or intervention means.
- Technology that attempts to manage the learning setting as attempted ITSs (e.g., automatically managing the flow of activities). The objective may be to manage orchestration in a way that is completely automated or, rather, to deal with part of the orchestration, as a way to allow teachers to concentrate on the core issues.

Orchestrable technology may correspond to different realities such as:

- *Flexible technology*, i.e., technology reifying in some way or another some given pedagogical intentions (e.g., a workflow structuring learners' collective activity) while allowing some tuning and run time adaptation by teachers (or the system itself, or the learners).
- *Creating-affordance technology*, i.e., technology whose usage is likely to create pedagogically rich events. Here, there is an explicit decision not to design and consider technology according to a single precise targeted usage. Rather, the artifact is meant to allow different usages, and give the player in charge of the orchestration some latitude with respect to how students will be prompted or allowed to use it.

3. (Primo)Scripting, conducting and (run-time)Scripting

In classrooms or in on-line settings, managing unexpected events or taking opportunities requires real-time management (Dillenbourg 2012, Dillenbourg & Tchounikine 2007). However, real-time management does not mean there is no pre-session management.

In order to help clarifying the orchestration notion one might come back to the music metaphor[1] and distinguish what is to be played, how it should be played, and the management of the performance[2].

However, in order to identify issues and research questions to be considered for developing orchestrable and orchestration technologies, an interesting perspective may be to consider these distinctions in terms of teachers' processes and requested tasks, and not only in terms of "before" and "during" the session:

- There is a process that consists in defining what is to be played (the music, the learning scenario general principles). Given the fact we focus here on the orchestration notion, I will not consider this further.
- There is a process that consists in deciding how what is to be played will be played in the current setting. This is the process within which one analyzes the way different means (registers, instruments, dynamics, etc.) may be used in order to obtain a targeted enactment (the musicians' performance), and one makes design decisions with respect to this objective.
- There is a process that consists in, using the output of the precedent process as a resource, analyzing the setting enactment and using different means to influence what happens, particularly what performers do.

Within this perspective, orchestration of a learning session may be analyzed by distinguishing *scripting* and *conducting*:

- *Scripting* is about envisaging how a set of means should be used in order to address teaching objectives. It is an analysis, design (from scratch or by adaption) and taking-decisions task, which result in a resource for action. As an example, scripting a CSCL setting may include:

[1] As I am a not a musician, the way I use this metaphor may not be more pertinent than the way others use it. Therefore, readers should stick to the view introduced here, although some others may exist and be more coherent with the musical context. This contradicts the point of metaphors but, anyway, metaphors are not good for science.

[2] If sticking to the definition according to which orchestration is about writing music for an orchestra only, the different processes may be seen as: writing the music, orchestrating the music and conducting the orchestra.

  - Analyzing the way means such as task and sub-task definitions, task distribution, roles, scheduling, data-flow, the technological framework, and the scaffolding or physical space provided may be used to structure and support students' activities.
  - Making design decisions.
  - Attempting to anticipate some real-time issues (as much as possible) by introducing some flexibility (Dillenbourg & Tchounikine 2007): how the teacher and/or software may react or be adapted in response to the absence of a member of the group, a timing issue, a dispute, a technological failure, etc.
  - Representing decisions in a way that enables their implementation by students, teachers and/or platforms. The output is what is usually called "the script[3]".
- *Conducting* is about contextually communicating directions to performers and adapting the setting components or their articulation. Here, the main performers are the students (there may be other players such as tutors) and the setting is defined by the technology, the physical space, the timing, etc. As an example, conducting a CSCL setting may include:
  - Analyzing the script unfolding and the students' individual and collective activities.
  - Providing students with directions, support, hints, etc.
  - Tuning the technology using the system's flexibility, changing the physical space, etc.

Scripting and conducting, as I introduce them here, do not correspond to separate period of times, but to analytic distinctions of the activities involved in orchestration: if conducting occurs during the session, scripting takes place *before* the learning session and, when requested, with different modalities, *during* the learning session.

For instance, an orchestrated CSCL session may be viewed (from the teacher's perspective) as:

1. Identifying and/or analyzing the pedagogical objectives, the general principles of the adopted scenario and the features of the particular setting (students, timing, space, etc.).
2. *Primo-scripting*, i.e., defining the initial script and the associated technological decisions.
3. *Conducting* the script, which consists of:
   - Analyzing the enactment (the performance).
   - Providing students with support and hints, tuning some parameters of the script or the technology.
   - When needed, engaging in *runtime-scripting*, i.e., reconsidering objectives and/or means.

In this description, I dissociate tuning some parameters from what I call *runtime-scripting*. This distinction is not related to the impact of the modification, but to the process the teacher engages in.

In some cases, probably in most cases, conducting just requires considering punctual questions such as "should I be smart enough to please the students and give them a few minutes more", "does it really matter if this student quits the group?" or "if I let them use this technology that I did not know about but that they apparently really want to use, will the rest of the script be more difficult to implement?" Modifying the deadlines or changing a student from a group to another may have a huge influence on the students' enactment of the script. But, from the point of view of what tasks the teacher must consider (and how to support this task), this is significantly different from scripting, which I see as a problem-solving task (analyzing complex and interrelated issues and designing a solution). The solution (the script) has already been constructed, and is not reconsidered as such.

In some cases, however, the script enactment may require that the teacher reconsiders the script as such, facing again the complexity of scripting. For instance, if it appears that the students will only go through three out of the five scripted phases and that it questions the script rationale, the teacher must engage in a process that is not about considering a punctual detail. There is a need to reconsider the problem and the solution in a way that had not been envisaged in the primo-script, at run-time, in relation to the current script effective enactment and in the light of run-time input (the current-script's unfolding). This is the same activity as scripting (i.e., envisaging how available means should be used in order to address current objectives), which is why I propose to refer to this as runtime-scripting.

[3] In this article, "script" refers to pedagogical scenarios in general, and to CSCL scenarios in particular.

A script being a complex structure, and a trade-off between often contradictory-constrains, its design rationale may be questioned by an array of events. The fact that only part of the planned phase will be enacted will likely require re-scripting, but this may also occur for many other reasons. As another example, a jigsaw script, which rationale is to provide students with different knowledge and then pair them in a way they will have to explain this knowledge one to another, may be completely ruined by unattended previous knowledge (or lack of knowledge) of the participants. What could have been a detail in one script (e.g., changing a student from a group to another) may require reconsidering the overall construction in another.

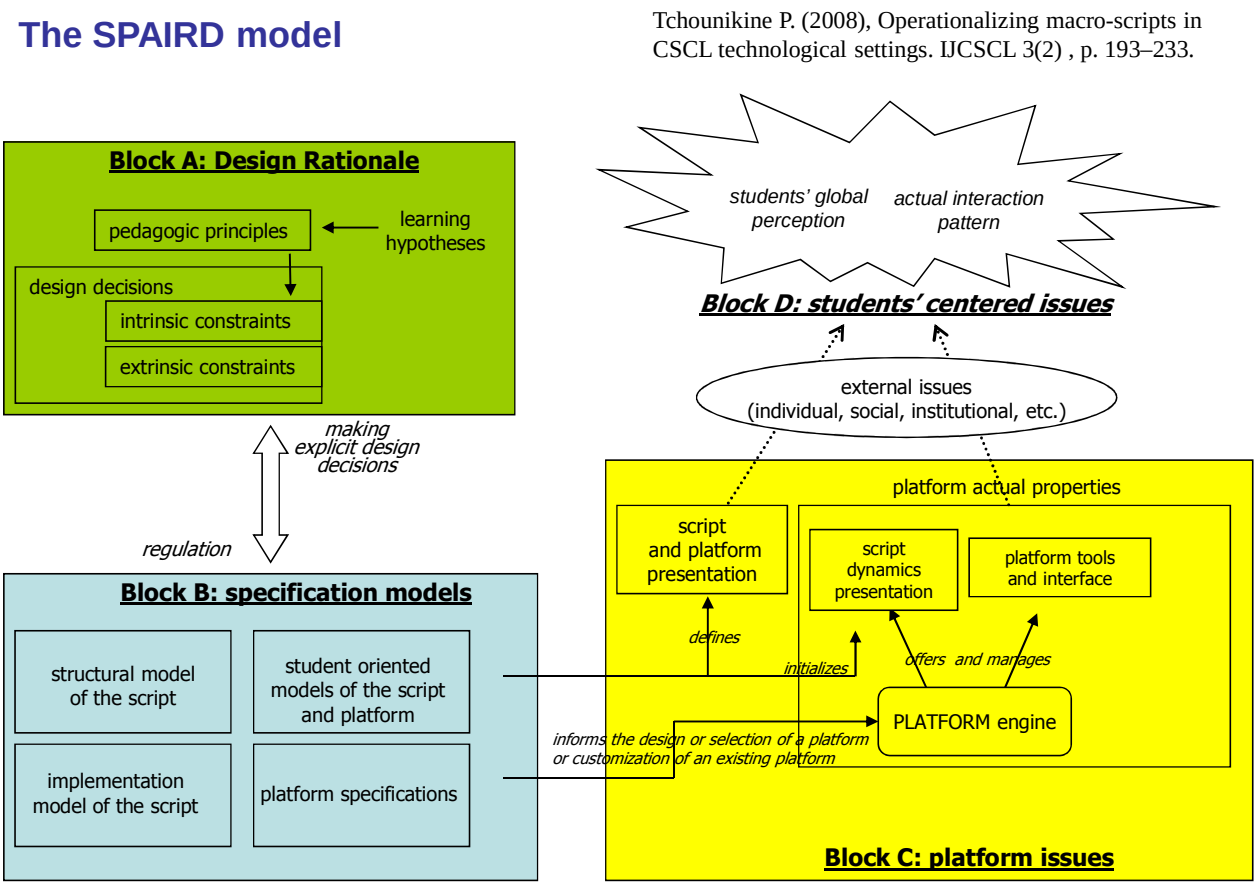
Of course, there is a continuum between tuning a parameter and run-time-scripting. However, disentangling these perspectives raises an important issue, that of allowing and supporting teachers when they need to deeply reconsider the script at run-time. Would it be acceptable for a teacher not to be able to do so? Supporting teachers in facing such issues is, to my opinion, at the very core of the orchestration perspective, and an incentive for teachers to use scripts and associated technology.

Run-time scripting may result in minimal actions (including locally over-ruling the script, i.e., at the end, nothing more than changing a parameter if one considers the result and not the process). It may also lead to important modifications: change the way some means such as breakdown of work or distribution of roles were used; abandon some technology or use an additional one; change the objectives (abandon the targeted objective for another more in line with the setting unfolding in order for the session not to be a total failure; build on an unexpected episode; consider new dimensions such as preserving or developing social relationships; etc.); change the didactic envelope (e.g., add a post-activity that will build on what effectively happened); etc.

Whatever the output is, supporting teachers in considering run-time scripting issues is very different from allowing them to change a parameter.

4. An example: scripting and conducting CSCL settings

As a way to see how these different notions relate one to another, we may take the example of CSCL scripts and the general model of CSCL script operationalization proposed in (Tchounikine 2008).



Primo-scripting may include (items are not in strict order):

- Identifying the script's intrinsic constraints (bound to the script's core mechanisms) and the extrinsic constraints (bound to contextual factors), given the pedagogical objectives and the adopted *pedagogical principles* and *learning hypotheses*. Extrinsic constraints define the space within which a script is modifiable because the related decisions result from arbitrary or practical choices, while intrinsic constraints set up its purpose and the limits of flexibility (Dillenbourg & Tchounikine 2007).
- Defining (as an explicit or implicit decision) the *structural model*, i.e., what conceptual notions (what "language") will be used to address the setting (e.g.: phase, group, physical space, group's ambiance, etc.). This may be determined by the teachers' education, the fact that the teacher reuses some material (e.g., a known script), uses a methodology or a platform that introduces certain notions, etc.
- Elaborating the *implementation model*, i.e., using the different notions to describe the script: the group formation policy and dynamics; the task sequencing and articulation; the dataflow/workflow determining access to individual and collective data and/or to functionalities/tools; the physical space issues if any; etc.
- Elaborating the *platform specification* (definition of the technological platform requested properties, and the selection, adaptation or implementation of a platform).
- Elaborating what students will be prompted with: the *script presentation* (description of what students are supposed to do) and the *platform presentation* (description of the technological means).

Conducting[4] may include (items are not in strict order):

- Monitoring and analyzing learners' performance and, more generally, the setting unfolding.
- Recalling directions, answering questions, providing hints, refining how students are prompted, adapting dynamically the *script presentation*, providing students with feedback, etc.
- Relaxing constraints or adapting directions within the scope of the extrinsic constraints. This may be a teacher's (or system's) initiative, or a response to a student's demand. It may consist in modifying groups or schedules, tuning the platform, etc.

Run-time scripting may include (items are not in strict order):

- Reconsidering the *intrinsic constraints*, as a way to adapt to the actual performance and, for instance, avoid a breakdown or take an opportunity.
- Reconsidering the *structural model*. For instance, it may be necessary to use new "conceptual tools" to analyze the setting, considering dimensions that had not been previously identified as important (e.g., ambiance, stress, emotions, saving data or maintaining leadership).
- Reconsidering the *implementation model*, i.e., the way means are used. Many things may be open for the human-based dimensions (although not everything is open: a teacher cannot go against some decisions or facts). What is related to a given platform is constrained by the platform's tailorability[5]. Changing for another technology is an option, which may be easy or painful (e.g., if it leads some works to be lost). One may also change what is supported by technology and what is not.
- Reconsidering teaching objectives (e.g., to be less ambitious or take an opportunity).
- Reconsidering the *teaching principles* underlying the orchestration efforts (or, rather, adapting to the actual unfolding, the way the complex and often-contradictory decisions to be taken in pedagogical settings are balanced).

5. Conclusions

Orchestration is an interesting notion to be kept to the fore and be re-interpreted in the light of current technology and trends (which does not remove the interest of older views). In order to define contributions to

[4] Conducting actions are related to the actual script, i.e., the primo-script as initially designed or the result of some run-time scripting.

[5] I refer here to tailorability as what can be adapted at run-time by the software users -here, the teacher or the students- without "going into the code" (the platform implementation can rarely be changed by an engineer during the session).

this issue, concepts that further specify matters of concern must be prepared. Within this context, I propose to distinguish orchestration technology and orchestrable technology, and define orchestration as a combination of scripting and conducting. Moreover, I propose to differentiate primo- and run-time scripting. These notions, and their articulations, are interesting topics for research.

6. References

Dillenbourg, P., (2012), *Design for orchestration*. Position paper.

Dillenbourg, P., Tchounikine P. (2007), *Flexibility in macro-scripts for CSCL*. Journal of Computer Assisted Learning, 23 (1), 1-13.

Tchounikine P. (2008), *Operationalizing macro-scripts in CSCL technological settings*. In: International Journal of Computer-Supported Collaborative Learning, 3(2), 193–233.

Tchounikine P. (2011), *Computer Science and Educational Software Design - A Resource for Multidisciplinary Work in Technology Enhanced Learning*. Springer. DOI 10.1007/978-3-642-20003-8_6.