# Toccata: Supporting Classroom Orchestration with Activity Based Computing

Valentin Lachand, Christine Michel, Aurélien Tabard

# Toccata: Supporting Classroom Orchestration with Activity Based Computing

VALENTIN LACHAND, Univ Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, France
CHRISTINE MICHEL, Univ Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, France
AURÉLIEN TABARD, Univ Lyon, Université Lyon 1, CNRS, LIRIS, UMR5205, France

We present Toccata, a system that facilitates the management of rich multi-device pedagogical activities. Through interviews with high school teachers, we identified a set of barriers to conducting digital activities in schools: set-up time, network problems, difficulties in following and changing plans as activities unfold. We designed and developed Toccata to support the planning of pedagogical activities (scripting), seamless sharing of content and collaboration across people and devices, live management of activities in the classroom, roaming for situations outside classrooms, resumption across sessions, and resilience to unstable network conditions. We deployed Toccata in three classes, over seven teaching sessions, involving a total of 69 students. Together, these deployments show that Toccata is a generic solution for managing multi-device activities in schools. We reflect on how Activity Based Computing principles support Orchestration in Toccata, and discuss the design opportunities it creates such as better awareness of learners' activity, micro-orchestration techniques for enabling teachers to better control devices in classrooms, or supporting reflective practices of teachers.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; **Empirical studies in HCI**; • **Applied computing → Collaborative learning**;

**ACM Reference Format:**
Valentin Lachand, Christine Michel, and Aurélien Tabard. 2019. Toccata: Supporting Classroom Orchestration with Activity Based Computing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 2, Article 53 (June 2019), 24 pages. https://doi.org/10.1145/3328924


Fig. 1. Toccata used in (a) a school greenhouse; (b) a collaborative Agile activity; (c) a multi-week fact checking course.

## 1 INTRODUCTION

Teachers have a rich ecosystem of devices at their disposal to conduct pedagogical activities, from interactive whiteboards, to computers, tablets, or smartphones. For example, in a vocational school, a horticulture lesson can

Authors' addresses: Valentin Lachand, Univ Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, LYON, F-69621, France, valentin.lachand@liris.cnrs.fr; Christine Michel, Univ Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, LYON, F-69621, France, christine.michel@liris.cnrs.fr; Aurélien Tabard, Univ Lyon, Université Lyon 1, CNRS, LIRIS, UMR5205, LYON, F-69621, France, aurelien.tabard@liris.cnrs.fr.

start in class with a lecture using an interactive whiteboard, continue in the school greenhouse, with individual work on tablets to inventory plants, and finish in the classroom with a presentation of group work. Later, learners can conduct follow-up activities and homework outside of the school on their personal devices.

In practice, planning such an activity is cumbersome, and teachers face numerous challenges in integrating devices into a seamless teaching experience: getting a class ready requires time-consuming set-ups, distributing content to learners is cumbersome, and adjusting plans to the activity as it unfolds is difficult, notwithstanding network and device management issues. Several Ubicomp projects demonstrated the ability to conduct such activities [7, 21, 30]. However, the activities are often hard-coded, leaving little room for teachers to create their own and run them in their situation of choice. While the computer-supported collaborative learning community proposed several systems that let teachers plan, conduct and follow pedagogical activities [10, 11, 28, 40], this often took place in controlled environments, overlooking real-world infrastructure challenges.

With Toccata, we developed a generic approach to offer a seamless teaching and learning experience in multi-device environments. Toccata builds on two conceptual frameworks: Orchestration, an education centric framework with a focus on class management for teachers [10]; and Activity Based Computing (ABC) [1], a set of technical principles grounded in Activity Theory [34] to facilitate individual and collaborative digital activities. We propose an activity centric approach to primo-scripting (before class) and run-time scripting (during class), seamless distribution across people and devices, live management of activities in the classroom, roaming for activities outside classrooms, activity resumption across sessions, and resilience to unstable network conditions.

To evaluate these contributions at a systems level, we deployed Toccata in three classrooms involving 69 students. All the case studies involved scripts that were created by teachers and conducted with their own classes. The first case study showed that Toccata can seamlessly move between a classroom context and a mobile context with unreliable Internet connection. The second case study showed that Toccata supported teachers in tracking and managing group progress. The third case study showed that Toccata supports activities lasting over multiple sessions, through activity resumption. This demonstrates the ability of Toccata to support various learning contexts and technological set-ups. Moreover, it proves that its underlying architecture, tying together orchestration principles and Activity Based Computing, is a promising approach for ubiquitous systems in education.

## 2 RELATED WORK

Computational devices are now ubiquitous in many high-schools of the western world. In France, over the past 15 years, significant funding efforts have led to public schools being equipped, on a widespread basis, with computers, interactive whiteboards, and tablets. However, adoption is still rare due to a lack of dedicated personnel, reliable infrastructure and dedicated tools supporting pedagogical activities [38].

Building upon the increased accessibility of large displays, multi-touch surfaces and mobile devices, several projects explored how fully digital classrooms could foster more collaborative learning activities [35]. Group Scribbles [7], Digital Mysteries [21], or Collaid [31] are a few examples among many. One side benefit of many of these projects is the ability of teachers to get an overview of the classroom activity as it takes place, thanks to the digital traces automatically collected.

These projects focus on *one* type of learning activity tuned to the pedagogical and technological context. While this leads to very rich activities, it makes scaling difficult. It also removes some of the agency of teachers, who have to conform to the learning scenarios imagined when developing the applications. Moreover, most of these projects assume the availability of devices in good numbers, a reliable network and technical infrastructure (good wifi, consistent devices and operating systems, proper management, etc.) Observations and interviews presented in the next section challenge these assumptions.

### 2.1 Planning of Learning Activities

Dore describes teachers' plans and pedagogical strategies as *"techniques and means used to reach an educational goal"* [12]. In educational research, a large number of models have been proposed to help teachers describe their plans. We discuss here narrative, executable and scripting approaches.

The first approaches proposed were mainly document oriented, following a narrative model. They structure learning activities around a document named scenario or narrative [12, 36]. A scenario is usually defined on three levels: courses, activities, and steps. A scenario is a document describing the general elements of a course: area of knowledge (discipline, subject), curriculum, public targets, target age, school level, objective of learning (target skills and educational intentions), and elements more specific to each activity: required skills, materials (for the teacher and for the learner), progress (role, preliminary preparation), phases, assessment.

With the development of computer-mediated learning, executable models emerged. These models are massively used in learning platforms such as MOODLE[1]. Some executable models are more normalized, such as LOM, SCORM, IMS-LD [36], with the goal of facilitating activity sharing, reuse, and remix. They mainly handle documents, and offer very limited support for interactive resources. However, they offer little support for collaborative work and can be complex to configure, thus making them challenging to teachers.

Kollar et al. proposed a scripting approach to deal with the complexities of collaboration in learning environments [26]. *Scripting* aims at describing expected interactions between teachers and learners during collaboration, from a global perspective (macro-scripts) corresponding to the general structure of an activity, and from a local perspective (micro-scripts) for more atomic tasks. Kobbe [24] proposed a script structure composed of participants, groups and roles. Dillenbourg [9] extended it by adding activity representation, time sequencing, activity distribution, and activity type. This specification is particularly suited for designing and monitoring hybrid, complex, collaborative situations. It can help teachers observe how a teaching activity unfolds and move smoothly to an alternative teaching plan [39].

These models are mainly dedicated to structuring and formalizing an activity before running it, in order to control its consistency and quality [32]. However, they are challenging to operationalize.

### 2.2 Orchestration

Going beyond planning, orchestration was proposed in the 2000s to support the creation of **scripts**, their **adaptation** to local constraints, and their **execution** in a given context [13]. Research on orchestration focused on supporting teachers' management of educational activities, by proposing principles to structure the timeline of an activity, i.e., length, curriculum, number of students, etc. [8, 10], and to improve the sequentiality of an activity [10] by controlling factors such as continuity, awareness, or relevance. Orchestration was later expanded to let learners manage their own activity or group activity [14, 42, 43], for example in homework cases [11].

One of the challenges of orchestration centers around preparing learning sessions. This design phase of a "score" [25] is called *primo-scripting* [45]. It involves identifying constraints, pedagogical objectives, and the elaboration of a scenario given available resources in the classroom. Such plans often have to be changed in class as the activity unfolds: this is often referred to as *run-time scripting* [45].

In order to integrate orchestration technologies in classrooms, in a non distracting manner, several systems building on a "modest computing" approach were proposed [10, 11, 28, 40]. Ambient awareness with paper-based computing devices [10, 11] or progression tracking devices [27, 28] are interesting illustrations of this approach.

There are several limitations to orchestration systems. Teachers consider primo-scripting to be heavy, and scripted activities to be not flexible enough. Systems proposing run-time enactment [28] are, for example, not adapted to short activities (classical 1-hour courses). The second limitation of orchestration systems is that, except for recent projects such as FROG [16], orchestration systems are built ad-hoc for specific activities [8, 28]. Finally,

[1]https://moodle.org/

orchestration systems are tested in a controlled environment with reliable network and device set-ups, although software and hardware breakdowns are widespread, thus hindering teachers' adoption of systems unable to cope with unexpected infrastructure problems.

### 2.3 Activity Based Computing

Activity theory [34] has inspired a number of system implementations that aim at tying together the many facets of human activities mediated by digital tools. The general approach consists in making activity a first class entity, in the way that documents and applications are considered today.

Early activity centric systems emerged in the 90s and early 2000s as a proposition to push the personal desktop computing paradigm further. Their main approach was to associate the digital resources required for an activity: documents [15, 20, 33, 46], applications [15], and tasks [20, 33]. Another concern that these systems shared was the ability to suspend and resume digital environments, i.e. support long-lasting activities [41].

With the development of Computer Supported Collaborative Work (CSCW), research on activity centric computing looked into supporting collaboration. Proposed solutions attempted to support communication between users with messaging systems and chats [33, 46]. Although they improved collaboration between users, these proposals did not allow management of the various forms of collaborative activities (synchronous/asynchronous, local or remote activity). Furthermore, user nomadism and roaming were not well supported, and it was not possible to conduct an activity in various places.

Building upon activity centric computing systems, and recognizing the need for supporting collaborative work, nomadism, and multi-device interaction [5], Bardram synthesized principles for activity centric systems under the name of Activity Based Computing (ABC) [3]:

- Activity-centric **resource aggregation** to encapsulate documents, tools, applications, services, and communication mechanisms as resources accessible from within the activity.
- Activity **suspend** and **resume** to support long and fragmented activities that span long time frames.
- Activity **roaming** so that activities can move across devices and maintain their state in the process.
- Activity **adaptation** for available resources of the device, such as screen size or interaction modalities (touch, keyboard/mouse, etc.) [2, 4, 44]. The aim of this adaptation is to provide continuity between devices (computers, tablets, interactive tabletops, etc).
- Activity **sharing**, allowing synchronous and asynchronous collaborative work with an activity involving several users. This principle was used in CSCW systems such as Clinical surfaces [4] or Reticular Spaces [2].

ABC principles were implemented in multiple systems [1, 2, 4, 19, 44], covering various use cases: nomadic work of medical workers, collaborative spaces of development, biology laboratories, etc. These systems are based on multiple architectures: local architectures, where every device has its own activity manager with no possibility of activity roaming and sharing [17, 19]; peer to peer architectures, in which each device has an activity manager with a possibility of activity roaming and sharing [2, 17]; Client-server architectures, where a core server has an activity manager to whom devices connect to reach an activity [1, 17, 44] and, finally, hybrid architectures, combining servers with activity managers and devices with (or not) activity managers [4, 17].

In an educational context, users with very different roles are involved: teachers, learners, administrative staff and parents. Such diversity of goals and possible actions is not well supported by existing ABC systems. Most ABC are collaborative, but do not make role distinctions or support precise tracking of other participants' actions. This distinction in roles is especially relevant for supporting group awareness, better monitoring of classes by teachers, and better self-regulation by learners.

The variety of roles, and the way activities are structured, also make activity sharing and content distribution more complex. Teachers can share activities with students, and students can join groups. However, teachers can also share activities among themselves for reuse, or "fork" activities created by other teachers. Such practices of sharing activity templates are widespread among teachers [37] but not supported in ABC systems.

### 2.4 Commercial Systems

A large number of commercial systems are used in schools today. Moodle is one of the most deployed in Europe. Its main advantage, but also one of its main friction point, is its high configurability. Thanks to plugins and themes, the system enables teachers to create multiple kinds of activities and to distribute them over several kinds of devices. However, the management of plugins is tedious and often constrained by institutions. Google classroom[2] uses a blog-like mechanism for creating and sharing activities. While it is more a collection of resources and applications than scripted activities, a large number of Web applications are integrated with the system. With Apple Classroom[3], the teacher can, in theory, directly create an activity in class, share resources and applications, view students' screens and take control of them. However, to participate, users must be on the same network, which creates problems when running on Eduroam or similar networks.

### 2.5 Comparison of Activity Centric, Orchestration, and Commercial Systems for Education

Table 1 summarizes how research and commercial systems compare in terms of orchestration and activity management. ABC systems were initially developed for collaborative work environments, which overlaps partially with learning environments, by requiring support for activity roaming, adaptation of systems to different kinds of devices, distribution over devices and spaces. Activity resumption, one of the basic principles of ABC, is not supported by orchestration or commercial systems. Adaptation to devices, which is crucial in classrooms with their large variety of devices (tablets, computers, video-projector), runs smoothly in ABC and commercial systems, but less so in orchestration systems where only interface adaptation is supported.

Compared to ABC, orchestration and commercial systems provide pedagogical concepts, such as role management, activity distribution and awareness support, which facilitates the creation and enactment of activities in class. Finally, the workflow of activities, from creation to enactment in class, is not well supported by the

Table 1. Comparison of ABC systems, Orchestration systems, commercial systems and our system, Toccata, in supporting principles of of activity based computing and orchestration for the management of digital pedagogical activities.

| | System | Workflow | Suspend/ resume | Roaming | Adaptation to device | activity sharing | roles | awareness | Network resilience | Distributed over devices AND spaces |
|---|---|---|---|---|---|---|---|---|---|---|
| Activity centric | Reticular space | ✔ | ✔ | ✔ | ✔ | ✔ | X | ✔ | X (NA) | ✔ |
| | Clinical surfaces | X | ✔ | ✔ | ✔ | ✔ | X | ± | X (NA) | ✔ |
| | Bardram | X | ✔ | ✔ | ✔ | ✔ | X | X | X (NA) | ✔ |
| | e-lab bench | X | ✔ | ✔ | ✔ | X | X | X | X (NA) | ✔ |
| Orchestration | Lantern | X | X | X | X | X | ✔ | ✔ | X (NA) | X |
| | Shelf | X | X | X | X | X | ✔ | ✔ | X (NA) | X |
| | Tinker Lamp | X | X | X | X | ± | ✔ | ✔ | X (NA) | X |
| | nQuire | ✔ | ✔ | ✔ | X | ✔ | ✔ | ✔ | ✔ | ✔ |
| | GrouPScribbles | X | X | X | ✔ | ✔ | ✔ | ✔ | X | ✔ |
| | Glue-PS! | ✔ | X | ✔ | ✔ | ✔ | ✔ | X | X | ✔ |
| | Martinez 2015 | ✔ | X | X | ✔ | ✔ | ✔ | ✔ | X | ✔ |
| Commercial systems | Moodle | ✔ | ± | ± | ± | ✔ | ✔ | ± | X | ✔ |
| | G suite for education | X | X | ✔ | ✔ | ✔ | ✔ | ✔ | X | ✔ |
| | apple classroom | X | X | X | X | X | ✔ | ± | X | X |
| Our system | Toccata | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

[2]https://classroom.google.com/
[3]https://itunes.apple.com/us/app/classroom/

three kinds of systems. Each approach supports some mechanisms which, if tied together, could support rich pedagogical workflows.

## 3 FIELD WORK

### 3.1 Interviews and Observations

We interviewed eight teachers (3 women, 5 men; age 26-50; 5 in middle school, 3 in high school) in Paris and Lyon, France, on their practices in planning and enacting digital activities [18]. The teachers specialized in a variety of disciplines such as French literature, Physics, Chemistry, History, English, German, Biology and Computer Science. The interviews lasted one hour. We conducted them in the places where teachers prepared their activities in schools, i.e., office or teachers' room. We focused particularly on identifying breakdowns and bright spots in using digital tools in schools. During the interviews, we asked teachers about their practices in creating and enacting digital activities in class, e.g:

- *Can you describe two or three activities you have created and conducted recently in class?*
- *Did you represent the activity workflow somewhere before conducting it?*
- *How did you manage to conduct the class sessions after the first activity session?*
- *Did you have to change or adjust the prepared activity as it unfolded?*
- *How did you manage this change?*

We audio recorded the interviews, and kept video recordings of teachers' interactions with documents they created during the interviews. We complemented these interviews with observations of the technical set-ups and capabilities of the schools in which we conducted interviews (eight different ones), such as network availability, differences in devices, and facility in using and obtaining these devices for classes, thus contrasting official discourses about widespread equipment availability in schools with actual reality.

We conducted a thematic analysis [6] to analyze the interviews. We extracted 48 stories in total (between 2 and 11 stories by participant) and sorted them according to: (a) the scripting strategy (11 different strategies identified), (b) the tools used to script and conduct the activity, (c) the content of the plans/scripts, (d) the temporality of tool use (before/during/after), (e) places of use and mobility, (f) data transfer strategies, and (g) the breakdowns in enacting the activities.

We complemented this analysis with more quantitative results on the types of tools used and teachers' level of satisfaction. The complete analysis is presented in [18]. We focus below on relevant insights for system design.

### 3.2 Breakdowns in Conducting Digital Activities

We do not seek to provide an encompassing description of teachers' highly varied practices with respect to planning and enactment strategies, but rather to identify a set of breakdowns and opportunities for design. We found that, in the design phase, some teachers created plans that students could enact directly. Others, however, went for lighter approaches and simply defined the content of the activity, or its structure, and sometimes both, with students in class. In practice, few digital plans could be run autonomously by students, and most enactment issues were related to software and hardware breakdowns. In order to reflect on their activity, most teachers used digital tools to keep a trace of their plans and to improve their enactment strategies.

A first class of problems was related to time issues, with severe discrepancies between the expected versus the actual time it took the teacher to run the session. Digital activities were less malleable and more difficult to adjust on the fly.

Software breakdowns were the most recurrent in the participants' stories. Seven out of eight participants encountered software breakdowns that prevented them from conducting their initial plans in the sessions they described. Network and hardware issues were also recurrent, and often created problems with document transfer across different devices. Sharing issues included accessing plans that teachers created at home on the school computer, managing software versions, and distributing content on several devices.

We found that teachers created strategies to work around these problems. For example, they included lightweight versions of the content they wanted to use in the plans they handed out to students, in case students did not have access to the full online version. They also made sure the content was accessible in class, while trying to make it accessible only at the appropriate moment.

From a technological perspective, we observed a wide diversity in the equipment available to teachers and students. Some schools had one or two computer labs/rooms, others had tablet carts, while others had one tablet per student; digital whiteboards were either mobile or attached to static computers. In schools with limited computer or tablet access, i.e., not one per student, conducting digital activities meant extra logistic efforts: booking rooms or devices, making sure software was installed or websites were accessible from the school. These disparities can be accounted for by local policies and by educational stakeholders (teaching or administrative staff) volunteering to manage the technical and administrative aspects of infrastructures and devices.

However, the most striking observation relates to connectivity. While cabled network is reliable, it could be slow when several classes were connected. Also, wireless network coverage was unreliable in all the schools observed, inhibiting the use of tablets in class, even in well funded schools.

### 3.3 Design Directions

Based on the interviews and technical observations, we derived design directions for supporting digital activities in schools. The first set of directions centers around technical aspects:

**D1 - Works on any device.** Due to the disparity of devices available in schools, teachers should be able to run activities on computers, tablets or other devices, with various resolutions and interaction modalities, i.e, accommodating most schools. Set-up time, technical administration and management overhead should be as limited as possible.

**D2 - Supports mobility.** We observed from stories that activities may happen in moving contexts (for example beginning in the classroom and finishing outside the classroom). Thus, teachers and students should be able to move freely if the activity so requires and to change place during an activity.

**D3 - Resilient to network issues.** Due to the variability in school networks, which we observed were generally unreliable, slow or unreliable networks should not disrupt pedagogical activities (for example by reducing tool availability due to network issues).

The second set of directions centers around the way activities are planned and conducted.

**D4 - Support for planning activities.** We observed from interviews that most teachers need to plan activities before the classroom session.

**D5 - Seamless distribution of content and live changes.** To speed up set-up time, which is a breakpoint in classical one-hour sessions, teachers should be able to predefine how content is to be shared with students, and be able to make changes to the activity on the fly, in class.

**D6 - Continuity across sessions.** For activities running for several sessions, to reduce set-up time and resumption load, when they resume an activity after a break, teachers and students should find it in the state they left it.

**D7 - Management of different roles.** Due to their different goals during an activity, teachers and students have to reflect on their roles: teachers should be able to control access to content, whereas students should be able to follow instructions.

**D8 - Support activity reuse.** Teachers should be able to reuse and share activities with themselves and with colleagues (and use other people's activities).

**D9 - Support for note taking and reflection.** Teachers should be able to easily capture how an activity unfolded as it happened, and keep this information within the activity for subsequent reuse.

Building on these design directions, we explored how two strands of research, from Ubiquitous Computing and Computer Supported Collaborative Learning, could be integrated to support pedagogical activities: Activity Based Computing and Orchestration.

From ABC systems we retain the principles of **roaming** and **adaptation**: activities can run and be moved seamlessly across devices, while retaining their interactive properties (**D1**). but also places and participants (**D2**). In ABC, roaming also covers resilience aspects, which are generally addressed through distributed, often peer-to-peer architectures (**D3**). The principles of activity **suspend and resume** are often associated with roaming and mobility: the activity can be started in class, and finished at home or in the library. But is this also relevant when activities unfold over multiple sessions (**D6**).

Drawing from orchestration principles, **scripting** can help finely structure collaborative activities (**D4**). In ABC systems, user roles are not explicitly managed, something that scripting clearly acknowledges. This is particularly relevant for group development **awareness**. Such awareness properties can support class regulation by teachers and self-regulation by learners. Accounting for the **diversity of roles and stakeholders**, e.g. teachers, learners, parents, staff (**D7**) is a factor that is hard to manage in existing ABC systems. Orchestration can help us consider these distinctions and their implications.

In learning contexts, activity **sharing** can assume various forms that go beyond what ABC systems have offered so far. In its simplest form, sharing revolves around teachers pushing activities to learners, in a somewhat similar way to sharing paper documents (**D5**). More sophisticated forms of sharing take place between learners working together: sharing activities or resources, or pulling work together. Looking at longer time frames, teachers can also share activities with colleagues or with their "future selves" for reuse (**D8**). This kind of sharing is already a widespread practice among teachers [37], but is not implemented in ABC or in orchestration systems.

The ability to reflect on learning activities is key to improving teaching practices (**D9**). Orchestration systems acknowledge the importance of reflective processes. Indeed, some implementations try to support it, for instance by showing the difference between what was scripted and what actually happened [29] or by showing information about previous sessions in order to plan future ones [23]. Although this matter has been discussed in the context of ABC [44], it is not supported by current implementations.

## 4 USING TOCCATA

Toccata is a generic platform offering a seamless teaching and learning experience in multi-device environments. It enables teachers to define the structure of pedagogical activities and conduct them in class. Teachers can distribute activities, follow and control students' progress, and modify activities as they unfold. In Toccata, activities can be re-used, either by the same teacher or by others. To this end, teachers can annotate their activity in order to re-structure them later or share activity templates with colleagues. Toccata is Web-based and runs on any device, with or without network connectivity, for a more seamless experience.

From a learner's perspective, Toccata is a Web application containing all their activities. Learners can run it in school, at home or in any other context. Toccata is designed so that all applications are run within it (even external web apps) in order to minimize distractions.

Toccata builds upon the principles of activity-based computing (ABC) [1] and orchestration [8], as unifying concepts. ABC provides a structure in which pedagogical activities can unfold in a collaborative and distributed setting; while the scripting principles of orchestration provide guidance for the planning activities and define user roles.

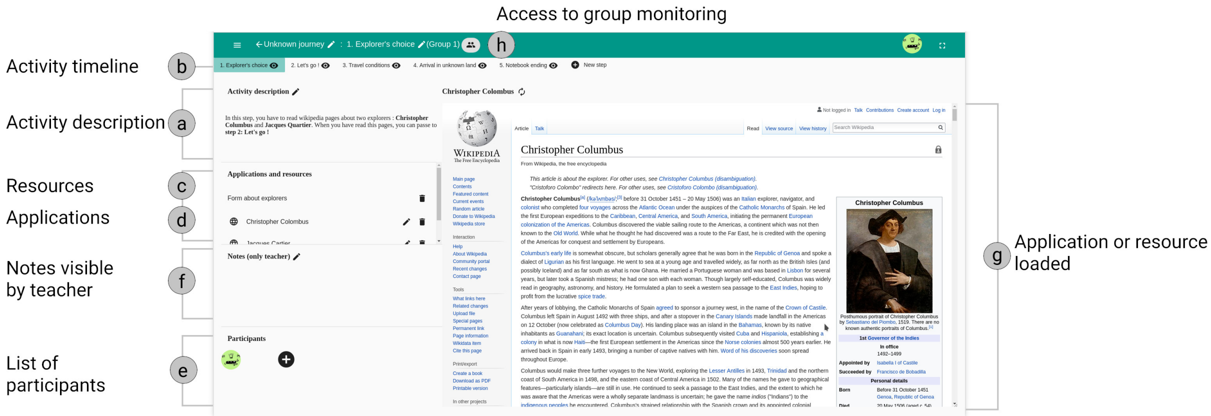


Fig. 2. An activity and its components in Toccata (teachers' view)

## 4.1 Defining the Structure of Activities: Primo-Scripting

The life-cycle of an activity starts with a teacher creating an activity and its associated script: it corresponds to the primo-scripting phase of Orchestration. Teachers can create a sequence of steps (sub-activities) (fig.3). To guide learners, **instructions** can be added to each step (fig.2-a). Teachers can add applications and resources to the activity (fig.4). Resources can be any document readable by a web browser (image, video, audio, Web page, pdf) and can be associated with the whole activity or with a specific step. Teachers can also add applications, i.e. external web applications (such as Web resources) or internal applications developed for Toccata: a text editor, a chronometer, a kanban board and questionnaire (fig.4). As for resources, the teacher can decide whether the resource will be available for the whole activity or for only one step.

Compared to other activity centric systems that mix the definition of activities and their enactment, Toccata places more emphasis on structuring activities first, and running them later (**design direction 4**). Teachers can create steps, rename them and order them freely thanks to a timeline at the top of the screen (fig.3-a to b). They can also decide which steps will be visible during the activity run-time (eyes in fig.3). This enables teachers to create drafts, hiding steps until they are ready. However, it can also be used to store extra activities that can be "unlocked" (i.e. made visible) if learners progress faster than expected. It can also be used to create various activity timelines for different student groups by showing/hiding steps.

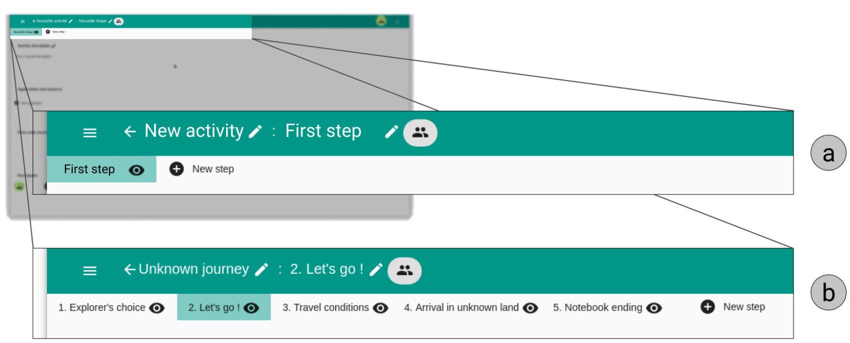


Fig. 3. Teachers can add, edit and change the order of steps directly in the step timeline.

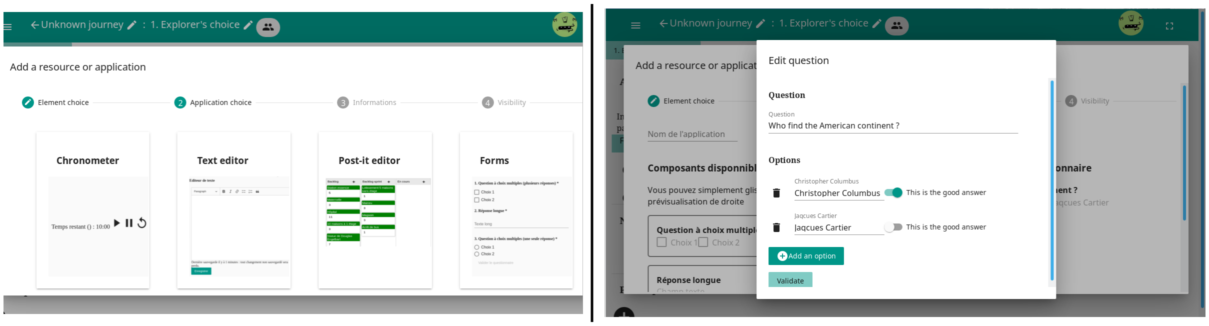


Fig. 4. Four built-in applications are proposed in Toccata (left). Teacher editing a question in the form application (right)

The User Interface (UI) layout is the same for activity scripting (fig. 2) and activity enactment (fig. 5), thus enabling teachers to have a direct preview of what the activity will look like on learners' devices. While editing an activity, teachers simply have extra editing control on elements that students will only be able to use when the activity is enacted.

## 4.2 Activity Sharing and Distribution

Once an activity has been created, it can be shared with the participants. ABC systems typically support activity sharing by inviting participants to an activity. With Toccata, teachers can distribute their activities according to their pedagogical needs:

- inviting all the students for a fully collaborative activity (this is by default in other ABC systems),
- inviting groups of students to multiple instances of the same activity for group work, and
- inviting a student per activity instance for individual activities.

In Toccata, we introduce the notions of "master" activity and activity "instances" which enables finer control of activity distribution. Activity distribution consists in pushing to other users a copy of the activity that is tied to the original. Teachers can either add participants to the master activity, or create instances of the activity and add learners to them. Changes made on the "master" activity will be propagated to other participants and reflected in all instances. On the other hand, changes made in an instance remain confined to it and are propagated only to users involved in it.

Teachers typically plan distribution before the activity is run, as part of the scripting phase. However, they are also free to "hand out" the activity live, in class.

We will discuss further on another form of sharing: template sharing, which consists in sharing copies of the master among teachers.

## 4.3 Activity Enactment and Run-Time Scripting

An activity can be enacted on any device supporting a modern Web browser, and in any situation: at school, at home, in a library, in mobile situations. Activity can also be more or less directed by teachers, e.g., an activity could be a group project on which learners have full scripting, sharing and editing control. We focus here on activities that are conducted in class with teachers.

In the classroom, the Toccata UI for learners is similar to that for teachers: they can see and navigate in the activity steps, instructions, resources, and applications. After tests in classes and exchanges with teachers regarding guidance and focus, we set up Toccata so that learners can only open one resource or application at a time. Learners can switch between resources and applications using the left pane, as they would with tabs (fig. 5).

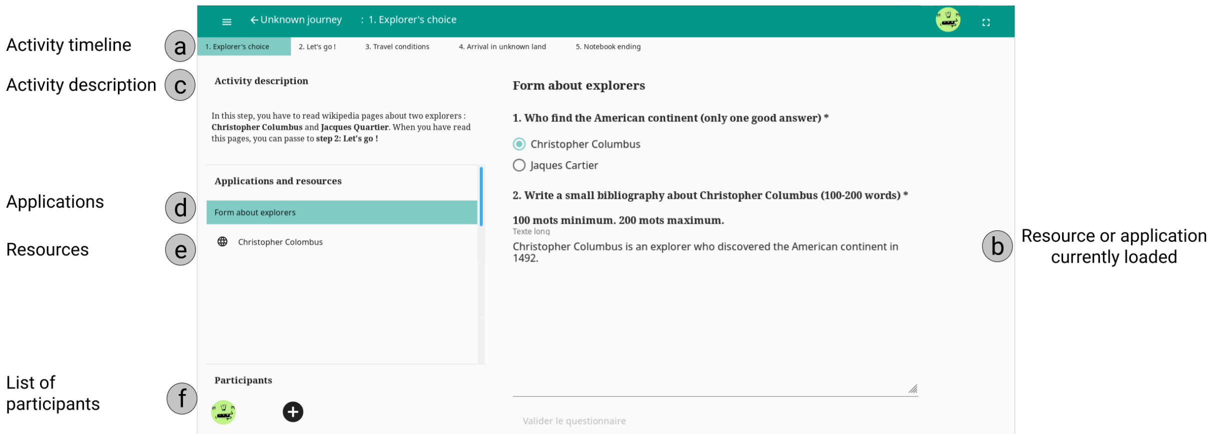


Fig. 5. Learners' view of Toccata: (a) the activity timeline; (b) a questionnaire application opened, (c) the extended left pane with instructions, list of applications, list of resources and participants.

When conducting the activity, teachers can follow students' progress via an overview screen displaying all the students/groups involved in an activity (fig. 6). The monitoring screen conveys information about group progress, time remaining when a step is time-boxed (with the chronometer application), or other application details (e.g. number of post-its in the kanban application). When teachers click on a group tile in the monitoring screen, Toccata mirrors the screen of the student or group. This enables teachers to see what the learners are currently doing.

If needed, teachers can make live changes to the activity script (run-time scripting), e.g. adding a step, modifying instructions, adding or removing resources (**design direction 5**). During the activity, teachers can also decide to hide or show steps, according to learners' progress and needs. Teachers can propagate these changes to the whole class, groups of students or individuals by controlling which instances of the activity are connected to the changes. They can edit their activity in the same way as during primo-scripting. All changes made by the teacher are propagated in real-time to students' activities, and no refresh operation is needed to view changes on the activity. This direct propagation and the ability to choose which users are concerned by modifications allow teachers to adapt the activity as accurately as possible to class progress and students' needs.

When activities are carried out over multiple sessions, they can be paused and resumed, or moved to other devices. Toccata stores the current state of the activity for each participant (current step, resources and applications loaded) and loads the saved state on activity resumption, regardless of the device (**design direction 6**).

Toccata currently supports teacher and student roles (**design direction 7**). By default, teachers have full control of the master activity and its instances. Students can only add resources and applications to their instance.

### 4.4 Reusing and Sharing Activity Templates

Teachers can reuse and share activities with themselves and with colleagues (**design direction 8**). Template sharing consists in sharing a model of an activity without the participants associated with it, but with all its content, sub-activities, resources, applications, etc. When shared, a copy is made, and changes to the template shared, will not be reflected on the original activity. Template sharing is mostly dedicated to copying activities to be deployed in different classes, trying out another activity structure while saving the original, and most importantly, sharing among teachers. When an activity template is shared, the activity appears in the teachers' activity list, and s/he can modify it and create instances of this activity.

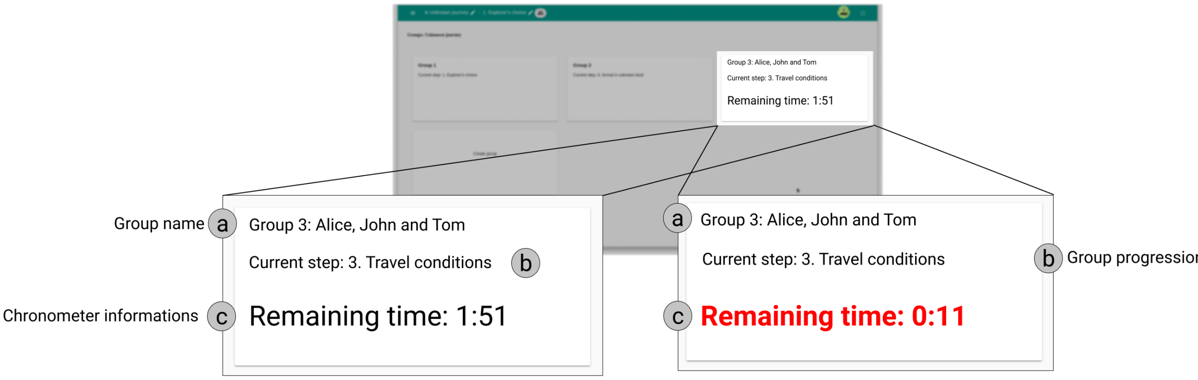


Fig. 6. Teachers can follow group progression: each group is represented by a tile (background) and information about the group is provided on the tile: group name (a), current step (b) and application-related information (c). The chronometer color changes according to the time remaining (c)

### 4.5 Reflection and Re-Design

During the enactment phase or once the activity is over, teachers can take notes on the way the activity unfolded. Notes can be associated with any individual or group activity, e.g., to keep information about group progress. Or they can be associated with the master activity. These notes can be used to reflect on how activities unfolded, offer guidance or reminders for future enactments of the activity, or be available for colleagues **design direction 9**. Building upon reflections on past experiences, these notes can support activity re-design by the author of the activity, or by colleagues wanting to adapt it to their own needs and usage situations.

## 5 TOCCATA ARCHITECTURE AND IMPLEMENTATION

Toccata is a Web application that can run on any device[4]. Its implementation follows a Progressive Web Application approach for handling devices with varying hardware capabilities and offering features normally available to native applications, such as running offline or storing data locally on the device, or device sensors such as camera, sound recording, compass, and other built-in sensors. Toccata's code is freely accessible on Gitlab[5]

Being web-based means that set-up time is reduced on many levels. The application can run on computers, tablets, or other devices with a modern browser (**design direction 1**). There is no need for a complex install process managed by administrators: intra-device communication uses Web ports that are usually open.

### 5.1 Resilience to Network Issues

Based on our observations, we sought to make Toccata resilient to network problems due to poor or unreliable connectivity or to complete lack of Internet access (**design direction 3**). With this in mind, we developed two strategies:

**Local storage:** Once an activity is loaded on a device, it is stored locally and is accessible even without access to Internet, and it will sync when a connection is resumed. This solution works on a per-device basis.

**Local server:** To handle lack of connectivity at class level, we introduced Toccata servers that act as hubs within the school, and syncs with the main Toccata server running outside the school (fig. 7). These local servers are portable (they can run on a Raspberry Pi or an Intel compute stick) and store activities locally.

[4]https://demo.toccata.education
[5]https://gitlab.com/lachand/Toccata

Local devices, i.e., tablets, connect to the local(s) server(s) via a dedicated local wifi network. Changes to the activities are propagated first to the local server so that participants can collaborate. The local server can run without connection to the Internet and the Toccata main server, but once (re)connected to Internet, changes will be propagated to the main server.

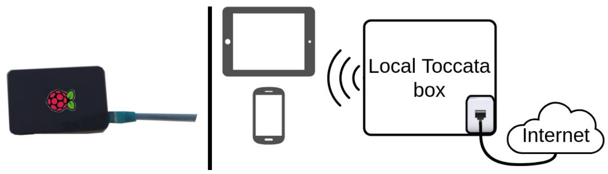


Fig. 7. Example of a server running Toccata on a Raspberry Pi 3 connected to an Ethernet network (left) and the basic scheme of a Toccata server (right)

## 5.2 Mobility

The local server can run on batteries and be connected to a tethered smartphone. This enables Toccata to run in contexts where it would be possible to conduct digital activities normally, e.g. outside the classroom, in the school yard, or on a sports field (**design direction 2**).

Activity roaming across devices is also transparent to users. By logging onto Toccata from any device, the ongoing activity will load in the same state as it was left by the user.

## 5.3 Architecture

Fig. 8 offers an overview of the architecture underlying Toccata. The system is made up of three layers. The first layer consists of Web servers and applications. The second layer is a local server running inside classrooms. The third layer is made up of client devices running Toccata. The second layer is not mandatory, and devices running Toccata can directly communicate with the Web server. When the first layer is not available, the devices synchronize with each other via a local server if they are connected to one. Otherwise they run independently.

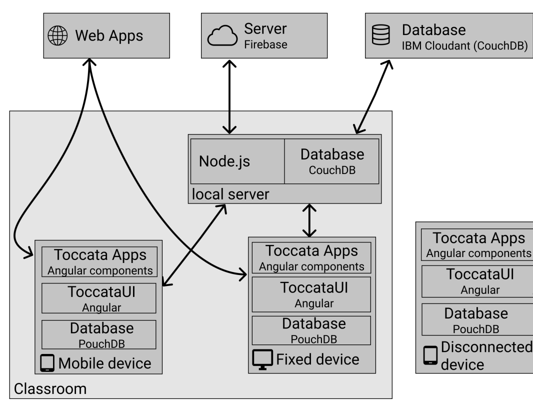


Fig. 8. Architecture of Toccata

This architecture supports three main network situations: (1) at school or in a mobile situation with access to a reliable internet connection. In this situation, the devices running Toccata can directly connect to the remote Web servers. (2) at school or in a mobile context with access to an internet connection but that is not reliable: in this case the local server acts as a proxy and facilitates collaboration across devices, and data storage, and (3) a disconnected context, in which participants have access to devices where the pedagogical activity has already been loaded and then removed from network coverage. The duplication layers are transparent to users. By default, Toccata queries the remote server. If the device is connected to a local hotspot with a local server running on it, the query is captured and redirected locally. Otherwise, it reaches the remote server. If no server is available, Toccata loads its cached version, and loads its local database.

### 5.4 Implementation

Toccata is a Progressive Web Application built with angular and extra synchronization mechanisms. The server delivers a single page application running in the browser. It is hosted on Firebase. Activities are stored on a CouchDB database hosted by IBM Cloudant.

The local server (fig. 7) acts as a WiFi hotspot. It can connect to the school Ethernet network, connect to a tethering smartphone, or run without any Internet connection. The server runs on Node.js with very little logic and, like the main server, only delivers a simple Single Page Application. The activities are stored in a couchDB database that syncs with the remote database whenever possible and if changes were made.

Each device runs Toccata and a PouchDB[6] instance. PouchDB allows synchronization between multiple instances of CouchDB servers, which solves most synchronization issues. As a Progressive Web Application, it offers some native-like features, like home icons on mobile OS, and strong caching mechanisms for data. It also offers an application shell: the webpage can load even when the device is totally offline, and it will synchronize back to the server when it becomes available.

External applications are iframes opened inside Toccata, while local applications are Angular components. This choice of components for local application was driven by their high extensibility, ease of integration, and the future opportunities for incorporating components developed by others.

### 5.5 Model

In addition to what we presented above, our activity model (fig. 9) extends the ABC model [1] with elements required for pedagogical activities derived from our design directions and orchestration principles. For the global structure of an activity, we introduce the notion of master activity, and activity instances. Only a teacher can be involved in a master activity, whereas anybody can be involved in an instance. To manage this, we introduce notions of groups and roles associated with participants. To handle the possibility of script generalization, i.e., a sequence of steps, they must implement a template structure that specifies basic script and sharing properties.

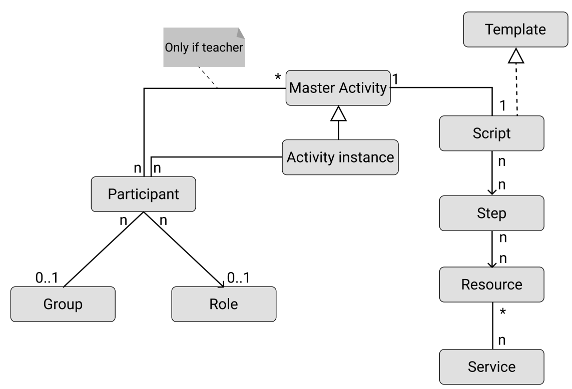


Fig. 9. Activity model used in Toccata

## 6 FIELD DEPLOYMENTS

After multiple design iterations and rounds of discussion and feedback with teachers, we deployed Toccata in two high schools and one university (see table 2). Our goal regarding deployments was to validate Toccata at a system level, i.e., test:

(1) the versatility and genericity of Toccata for supporting scripting and enactment of various activities, e.g., running over one or multiple sessions, in a single environment or in changing contexts, involving purely digital or mixed media resources, and
(2) the robustness of Toccata in real-world school situations, from reliable to no network, on different kinds of devices, with various restrictions on devices, etc.

We interacted with teachers to plan each case study, with activities that either built upon one they were already familiar with (CS2, and CS3), or was completely new (CS1). Depending on the situations, we provided the devices

[6]https://pouchdb.com/

Table 2. Overview of our three deployments

| Activity | Context | Devices | Participants | Age | Activity duration sessions | time |
|---|---|---|---|---|---|---|
| CS1: Sales management | Classroom; Outside | Computers; Tablets | 17 | 15-16 | 1 | 1h30 |
| CS2: Agile workshop | Classroom | Tablets | 41 | 22-23 | 1 | 1h |
| CS3: Fact checking | Classroom | Computers | 11 | 13-14 | 3 | 2h45 |

(CS1, CS2), or used ones from the school (CS1, CS3). This amounted to 72 users, over 5 sessions, and 105 hours of cumulative use across participants. In all three deployments, we collected logs from tablets and computers, videos of groups, screen recordings of tablets if available, and observation notes. At the end of CS1 and CS3, we briefly interviewed the teachers in order to find out what they thought about the use of Toccata for teaching practices and what could be improved in the design.

**The first case study** (CS1) is a **sales management course** for vocational high school students in horticulture (17 students in 10$^{th}$ grade, age 15-16). The instructor expressed the need to use Toccata to organize a 1h30 review session on the major topics of the course towards the end of the year. He wanted to work on very different concepts and vary the type of exercises to keep the students motivated. More specifically, he wanted to combine activities such as analysis of a sales video to work on selling skills, calculation of prices and taxes with a spreadsheet, and design of a sales catalog with texts and photos from the plants in the school greenhouse. Such an activity would not have been possible without Toccata due to lack of connectivity in the classroom and the greenhouse, and the difficulty in conducting three activities in 90 minutes without smooth transitions.

**The second case study** (CS2) is a hands-on **Agile workshop** activity for master level students in Computer-Science (41 students, age 22-23). This activity aims at practicing the SCRUM method. Students had to build a paper mock-up of a city. The activity is based on a well-established activity template used by Agile coaches[7] that the teacher had already practiced in previous years. The main challenge for the teacher was keeping track of time, group progress, and taking action at the appropriate moment.

**The third case study** (CS3) is a course about media literacy and **fact checking** made up of three 55-minute sessions (24 students, 11 using Toccata, age 13-14). Each session corresponded to a step in the activity. The activity consisted in checking information on Vikidia[8], a collaborative encyclopedia for children and teenagers. The activity is built on a template used in the past, but involving paper handouts to keep track of students' work. Use of Toccata enabled an integrated digital environment in which the entire activity took place.

## 6.1 Activity Scripting

For CS1 and CS3, teachers created the scripts on their own devices, and we offered support if needed. For CS2, the teacher, a colleague, gave us the script description which we created in Toccata.

For the sales management activity (CS1), Teacher 1 (T1) structured it into three steps (see fig. 10). In the first step, students had to watch a video, analyze it, and write a document to answer questions with the text editor. They could use a pdf viewer to read their lesson. In the second step, students had to compute taxes and prices to produce an invoice with the collaborative spreadsheet. At the end of steps one and two, the teacher showed his/her activity with the classroom's video-projector and asked a student to come to the whiteboard to correct it. In the third step, students had to prepare the plant catalog by taking pictures of plants cultivated in the school greenhouses, and then organize and describe them in the text editor. At the end of the third step, the teacher

[7]https://www.lego4scrum.com/
[8]https://www.vikidia.org/

asked students to present their work to the class, with each group presenting the pictures taken in the greenhouse on the video projector and discussing with the class how they could be used to create the school catalog.

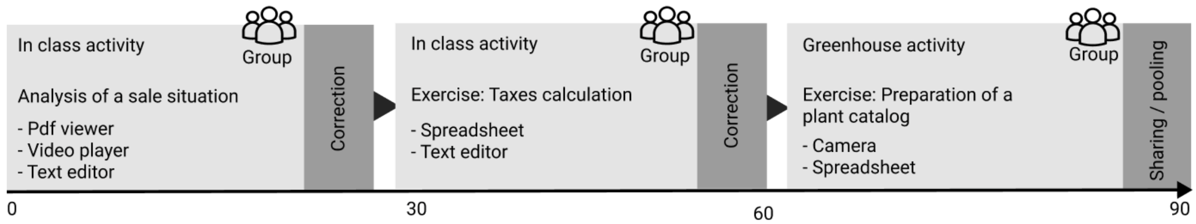


Fig. 10. Script of the sales management activity

For the agile workshop activity (CS2), Teacher 2 (T2) structured it into four sub-activities, corresponding to a preparation phase (creation of a backlog), and three production phases (sprints 1-2-3) (see fig. 11). Each production phase was divided up again into three steps: estimation of the tasks to perform, task completion (construction), and review. The teacher built most of the activity on two applications, letting students control their time and tasks (using a timer and a Kanban board[9]). The duration of each sub-activity was set to 15 minutes, after which students were expected to move to the next stage, and each sub-step was also precisely time-boxed (i.e. 3 minutes' estimation, 7 minutes' construction and 5 minutes' review with the teacher).

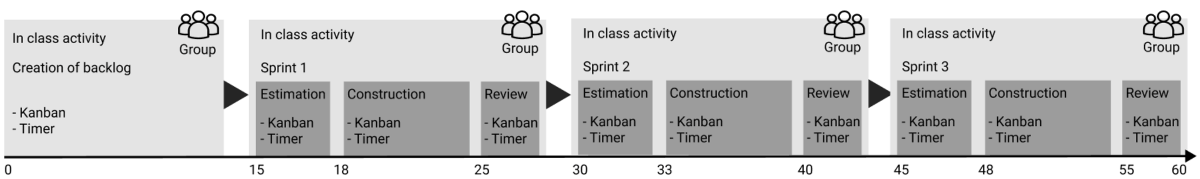


Fig. 11. Script of the agile workshop activity

For the fact checking activity (CS3), Teacher 3 (T3) structured the activity into three steps (see fig. 12). The teacher used the built-in text editor and various Web-based media resources. In the first step, students had to read information about different media, e.g., TV, radio channels, or newspapers, on an online encyclopedia (Vikidia), and check the veracity and freshness of the information. In the second step, they had to edit one encyclopedia article and correct erroneous information. In the last step, they had to add new information about the media to the encyclopedia. They had to explain their analysis and the changes in the text editor. In order to create unique

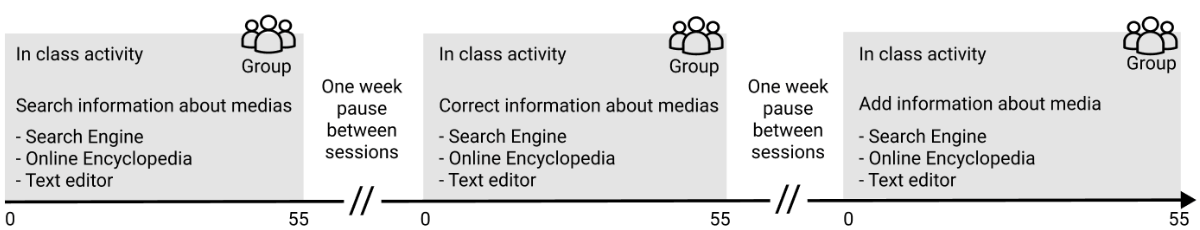


Fig. 12. Script of the fact checking activity

[9] In an Agile context, a Kanban board is a board made up of several columns used to track task progress

activities for each group, T3 created a 5-step activity where only 3 were visible for each group. He duplicated two steps and created variations (one on classical media, the other on digital media).

**Scripting results:** The scripts imagined by the teachers varied in duration, place (inside and outside the classroom) and tasks (analysis, exploration, calculation, writing) (**D4**). The activities leveraged computers, tablets, and whiteboards (**D1**).

From an activity scripting perspective, CS1 and CS2 show that Toccata could support the creation and enactment of complex activities. The integration of multiple resources and applications in Toccata made CS1 possible. T1 could incorporate an external Web-based spreadsheet tool that he already knew how to use, which facilitated the activity. T3 appreciated the ability for learners to move content easily across resources and applications: he realized that having one text editor resource for verifying content and another for correcting content enabled both the students and himself to better track and structure the progress of their work, before moving to real correction of the encyclopedia *"I did not think about it before, but having a verification trace, plus a correction trace allowed them to work on draft" (T3).* T3 also noted that Toccata enabled him to offer more guidance, by adding instructions directly to the resources, in complement to the instruction dedicated space *"what is good is that it allows us to propose instructions with the structure of expected answers" (T3).*

Toccata supported scripting an existing activity with strong temporal constraints (CS2). Our activity model supported the definition and management of scaffolded activities: one main activity, four sub-activities, and three steps in the sub-activities. In CS3, the teacher could script an activity template, and create variations for each group, by individualizing the content (**D5**).

## 6.2 Activity Enactment

*6.2.1 Set-up.* For CS1, the teacher conducted the same session twice, each time with 8 students. Students were split into four groups of two participants. At the beginning of the session, students received one Android tablet to share between two with Toccata loaded, and the teacher used a computer linked to a video-projector (**D1**). Due to the lack of reliable wifi in the school, we also provided a local wifi access point using tethering.

For CS2, the teacher conducted the same session twice, each time with 20 students. Students were organized in four groups of five working simultaneously. At the beginning of the session, the teacher gave each group an Android tablet with Toccata loaded, as well as paper strips and glue. The teacher had an Android tablet for orchestration purposes.

For CS3, the teacher conducted the activity over three sessions, each one week apart, with 24 students, where 11 students used Toccata and 13 students used paper and a Web browser. Students worked either alone or in pairs, based on the number of computers available. We noticed that the groups not using Toccata spent time on non activity-related tasks (some played an online game and others visited a sales website), whereas students with Toccata only used it during the activity. The activity took place in the computer room of the school, on desktop computers, directly connected to the main Toccata server. We did not use a local server. The teacher had a computer connected to a video-projector with Toccata open. We focus only on the students who used Toccata in the analysis.

*6.2.2 Orchestration.* In all the deployments, the teacher distributed the activity to students. In CS1 and CS2, the teacher duplicated the activity for all groups and shared it at the beginning of the session. In CS3, the teacher distributed two variations of the activity: one group received instructions for fact checking encyclopedia entries on digital media, another group for radio or tv channels.

We observed two orchestration moments in CS1. (1) During the transition between group work and class correction: the teacher easily managed this transition by asking students to look at the video-projector where the correction took place. Students then corrected their work on their tablet. (2) When the teacher changed his strategy to follow groups: he began by following progress on Toccata's monitoring screen on his PC. However, due to the PC's position, the teacher decided to show the progress on the video-projector and to directly look at the tablets to follow student progress more precisely. Regarding these two moments, (T1) noted that *"[Toccata] is a good tool for pooling"* and that *"I had a global overview of groups [...] I just saw in which step they were."*

From an orchestration perspective in CS2, although the teacher started by monitoring group progress and keeping time on his tablet with the group overview, he increasingly looked at the students' tablets rather than at his own to monitor group progress. This resonates with previous work on orchestration and awareness [10], suggesting that providing information about group progress and students' needs in a situated manner can be beneficial. The teacher expressed a need for better awareness of the remaining time in the session and for each group, and also for the time he spent with each group. We expected that the teacher would use his tablet to control group activity (**D7**), e.g., suspend and resume the timer, or move tasks around the Kanban board as he accepted or rejected them. However, throughout the activity, T2 preferred to act directly on the students' tablets, rather than on his own (fig. 1.b).

In CS3, the teacher managed the class without using his instance of Toccata. Throughout the sessions, the teacher walked from one group to the next to answer questions, and helped pick another article if the quality was too high and there was little to correct, or when students did not appear to focus on the task.

Before the study, we expected to see more orchestration, such as run-time scripting of the activity to adapt it to student progress. The teacher mostly adjusted the activity verbally either by addressing a group, or the whole class, never using Toccata run-time capabilities, whether for monitoring or for adjusting the activity. With sessions lasting 55 minutes and a lot of student questions, there was little time left to take a step back. The students were also younger (13-14 years old) and less autonomous than in the other deployments, thus requiring more individual attention from the teacher.

To manage the class, teacher 3 thought that it was easier for students with Toccata to run the activity despite its novelty and the need for more precise instructions. *"For [students without Toccata] it is harder because they have to write the verification on their paper sheet or to open a text editor to have a draft [...] I had to be more present with [students with Toccata] for instructions."*

*6.2.3 Roaming and Unreliable Connectivity.* When deploying CS1, the school wifi was fluctuating. While external Web applications, such as the collaborative text editor integrated in Toccata, did not work perfectly, resources hosted on Toccata could be properly accessed and students did not encounter major problems. They managed to move from one step to the next. Due to the fluctuating network, changes made during the correction on the teacher's computer were not automatically synchronized with students' activities, and students had to manually update the activity on their tablet. In the third step, students moved to the greenhouse, with no wifi coverage (fig. 1.a). This step worked smoothly, and students could grab their tablet and continue their activity as expected. In the greenhouse, they moved freely and took pictures of plants they later added to a sales catalog. "[The activity] in greenhouses was much easier and pleasant because straight away we are in an outside class context" (T1) When they came back to the classroom, the activity was updated on the global server, and the teacher could access the pictures taken in the greenhouse.

For CS2, due to an unreliable network, we used a local server and a tethered 4G connection to keep the local server synchronized with the remote one. With this adaptation, the two sessions ran successfully (**D3**).

For CS3, Toccata ran properly in a client/server configuration without any networking issues. Toccata seamlessly integrated external resources from the Web and others stored locally.

**Enactment results:** From a technical perspective, the case studies showed that Toccata worked on a variety of devices (**D1**), in mobility situations (**D2**), with unreliable networks and various network configurations (**D3**). Even with network problems, and the limited access to external applications encountered in CS1, the teachers could conduct the activities they previously conducted on paper (CS2), on mixed media (paper and digital in CS3), or not at all (CS1).

Being a single page application with all the content accessible within it, helped in keeping students focused on the activity, something that T3 appreciated, *"what is interesting with Toccata is that it has everything in the same interface"*. T3 also appreciated the support for continuity across sessions with automatic activity suspend and resume (**D6**).

Teachers managed to distribute content and make live changes to their activity (**D5**). However, we observed few instances of run-time scripting, with teachers changing the activity on the fly inside Toccata (only in CS1). During the course, teachers preferred to stay in contact with students, directly explaining to them the changes in the activity (CS3) or correcting the exercises inside the students' activity (CS2). This suggests that orchestration mechanisms, whether awareness or control mechanisms, should not only be available on the teacher's device, but also on the students' devices so that the teacher can explicitly orchestrate the activity in a way that is visible to all.

## 7 RESULTS AND DISCUSSION

We designed and developed Toccata to support a wide range of pedagogical activities in a reliable manner, regardless of the technological context. Toccata worked in the different situations and in the various network configurations in which we deployed it. Table 3 summarizes the design directions we sought to support. Grey cells represent the times when they could be used in the deployments we conducted.

### 7.1 Linear Scripts Help Define and Manage Activities

Toccata properly supported activity scripting, enabling teachers to adapt existing activities or to create new ones. The ability to integrate external documents and Web applications as resources proved useful to bootstrapping its use without re-developing a complete educational software suite.

Although we initially feared that our chronological and linear approach to scripting could become a limitation, we did not receive negative feedback or requests for more complex structures. In interviews and exchanges we had with teachers outside the deployments, when they envisioned branching activities, they "flattened" them out by hiding/showing the relevant steps. While this linearity facilitates the definition of the activity, teachers also favored this strategy to convey a simpler lesson plan to students.

Table 3. Summary of the design directions. In grey cells, the deployments in which design directions were implemented, and whether they were successfully (✓) or partially (≈) used.

| | | Sales management (CS1) | Agile Workshop (CS2) | Fact checking (CS3) |
|---|---|---|---|---|
| Technical design directions | **(D1)** Works on any device | ✔ | ✔ | ✔ |
| | **(D2)** Supports mobility | ✔ | | |
| | **(D3)** Resilience to network issues | ✔ | ✔ | |
| Activity design directions | **(D4)** Support for planning activities | ✔ | ✔ | ✔ |
| | **(D5)** Seamless distribution and live changes | ✔ \| ≈ | ✔ \| ≈ | ✔ \| ≈ |
| | **(D6)** Continuity across sessions | | | ✔ |
| | **(D7)** Management of different roles | | ≈ | |
| | **(D8)** Support activity reuse | | | |
| | **(D9)** Support for note taking and reflection | | | |

## 7.2 Run-Time Scripting with Verbal and Direct Interventions

While teachers actively used Toccata to script the activity, they did not use it during activity enactment, for run-time scripting, group management or simply to obtain an overview. Teachers carried out run-time scripting by giving verbal instructions or through direct actions on students' tablets. Indeed, the activities were fast-paced, and teachers engaged directly with students or the whole class when something needed to be addressed. Since carrying a tablet to monitor and control the activity was cumbersome, teachers stopped carrying them during the activity. However, using a dedicated computer required going to it, thus often breaking the flow and leading to downtime. In addition, run-time scripting required navigation in Toccata, and the cost involved and the risk of making mistakes was considered much higher than when the script was prepared beforehand in a quiet environment.

## 7.3 Micro-orchestration and Sharing Principles

Instead of run-time scripting, we believe that an interesting future work would be to support micro-orchestration tasks. These tasks are related to actions that teachers perform in order to manage their class around the script they planned, e.g., validating a step, expanding its expected duration, turning off students' screens to draw their attention to the teacher, etc.

Although some commercial systems, such as Classroom by Apple or research projects, support a few of these micro-tasks [8], this is not done in a systematic manner. Field research is needed to better understand the micro-orchestration tasks that teachers already conduct in their classrooms. This could open up opportunities for novel tools and interaction techniques to orchestrate activities on the fly.

In an ABC context, sharing is often equated with adding new participants to an activity. In an educational context, the participants have different roles, and sharing can take many forms. We proposed to distinguish "activity distribution" and "template sharing" to distinguish an activity assignment (from teacher to students), from sharing among teachers. However, these sharing mechanisms are still limited. We did not implement ways of handing-in an activity (from students to teacher) or of sharing with peers (from students to other students).

Moreover, students cannot inform teachers that their work is completed and that they are ready to receive feedback. For example, in the fact checking activity, when editing the online encyclopedia, learners wanted feedback from the teacher before pushing their changes to the world. Toccata neither supports notifications from learners to teachers nor provides the ability for teachers to comment on students' work.

Concerning activity reuse (**D8**), Toccata supports sharing activity templates among teachers. However, we did not explore this with teachers in our one-off deployments. Longer and broader deployments are needed to explore the opportunities related to teacher-to-teacher collaboration with Toccata, such as activities involving several teachers, or sharing activities between teachers in the same field, or reusing an activity from one year to the next.

## 7.4 Tracking and Following Learners' Activity

One of the expected benefits of orchestration technologies is the ability to better capture and follow students' progress. Tracking learners' activities is now widespread, and has opened up avenues in learning analytics research. Yet tracking is always challenging, and many activities are not 100% digital. Also, for digital activities, assuming that everything can be tracked and controlled within Toccata would probably be a mistake: external applications offer limited insights into their inner workings; if several students work together on the same device, identifying individual contributions is difficult; and less tangible elements such as group engagement or mood are hard to capture. A challenge here is understanding how automated tracking can be mixed with more manual and explicit capture, in order to better account for what happens outside of the system, and thereby increase teachers' and students' awareness. This can take several forms: for instance, T1 noted that "It will be interesting to allow students to auto-evaluate their work. For example to know that in exercise 2, half of groups judged to be in difficulty".

To help teachers manage their class, we provided a dashboard with basic information about students' activity, although this was not a challenge identified in our interviews or a design direction. The dashboard proved to be of limited use despite deployments with various set-ups: a fixed device (teacher's computer), on a mobile device (teacher's tablet), on the classroom whiteboard. In a classroom, teachers could gain awareness from what was happening in class first-hand, and the dashboard, although simple, was too challenging to process or did not provide the right information at the right time.

### 7.5 Context Awareness and Wearable Controllers

We only used local servers as a redundancy mechanism for coping with challenging network conditions. However, they could also be used for context awareness purposes, which is useful for adapting activities to the situation. Toccata servers could embed knowledge about their location, teachers' identity, current classes, etc. This could be leveraged to further improve suspend and resume of activities, activity sharing, or to propose minor variants of activities according to where they take place (such as reading specific books when in the library instead of using Wikipedia when in a classroom.)

Toccata architecture supports device adaptation and roaming thanks to our Web-based approach. However, it has limitations compared to native development when it comes to mobile sensing or wearable device support. Web browsers have limited access to sensors such as battery level, advanced camera features, Bluetooth, or Near Field Communication (NFC). Bluetooth or NFC could enable us to detect the close proximity of a teacher to a device and unlock the students' tablet with teacher only features when s/he is around.

Wearables could improve the degree of control and awareness of teachers in the classroom. They have the advantage of keeping users' hands free and of providing information on multiple channels in a discrete manner (vibration, screen, sound). They can also facilitate identification, for instance, IMU worn by teachers could give them extra controls when interacting with students' tablets [22].

### 7.6 From Note Taking to a Reflexive Activity Design Process

Our deployments did not last long enough to observe teachers' reuse activities. Also, the use of Toccata was too unique for teachers to envision repeating the same activity the year after. Yet based on preliminary field work, we believe that there are opportunities for better supporting reflection within Toccata. Martinez Maldonado et al. [29] or Kharrufa et al. [23] proposed to show teachers the difference between what they scripted and what was really done during the enactment phase. While such information would be useful when re-designing an activity, it would imply fine grained tracking of the enactment phase.

With Toccata, teachers can attach free form notes to an activity. More lightweight annotation mechanisms, such as bookmarks or positive/negative flags attached to a step, could support subsequent re-design as well. Another strategy could revolve around more structured notes, for instance by asking teachers closed questions on how their various activities panned out at the end of their day. These strategies would be particularly useful for capturing and reflecting on what happened in class, outside of Toccata, as we discussed above.

## 8 CONCLUSION

We proposed Toccata, a system combining ABC and Orchestration principles for scripting and enacting pedagogical activities in different school contexts. Compared to previous work in ABC, Toccata offers strong scripting/workflow management mechanisms, before and during the activity, it supports role management and sophisticated activity sharing capabilities. Toccata also builds upon modern Web technologies and runs in a browser, which enables us to incorporate existing Web applications as resources. Compared to Orchestration systems, Toccata offers a generic approach to managing pedagogical activities across devices, it can run with limited to no connectivity, and in mobile situations.

We conducted three case studies to test whether our architecture supported different school contexts. Teachers managed to create and run complex activities of their choice, using various devices, applications and multimedia resources.

Teachers could seamlessly launch digital activities in the classroom, move around the school, and conduct activities even with unreliable networks. Students did not face any hurdles in using Toccata, and they could collaborate to conduct their activities. The all-in-one page approach of Toccata limited distractions and encouraged students to stay inside Toccata.

Our deployments revealed that teachers did not rely on Toccata for orchestrating the activity as it took place in class. We offer design directions to improve orchestration such as leveraging students' devices and teachers' mobile devices. Better understanding "orchestration moments" could also lead to more generic orchestration components in Toccata and other orchestration systems.

Finally, although we have not yet investigated this aspect, we believe that Toccata offers many opportunities as a tool to support reflection on pedagogical activities: either by comparing planned scripts to the way they unfolded, or by facilitating a more qualitative and experiential capture of the activity. Such reflection mechanisms could have a direct impact on activity redesign and how activities are shared among colleagues, as well as on teachers' practices.

## ACKNOWLEDGMENTS

This research was carried out within the REPI project funded by the French "FUI AAP21". We would like to thank all teachers and students who participated in the studies.

## REFERENCES


[1] Jakob Bardram, Jonathan Bunde-Pedersen, and Mads Soegaard. 2006. Support for activity-based computing in a personal computing operating system. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*. ACM Press, Montréal, Québec, Canada, 211. https://doi.org/10.1145/1124772.1124805

[2] Jakob Bardram, Sofiane Gueddana, Steven Houben, and Søren Nielsen. 2012. ReticularSpaces: activity-based computing support for physically distributed and collaborative smart spaces. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. ACM Press, Austin, Texas, USA, 2845. https://doi.org/10.1145/2207676.2208689

[3] Jakob E Bardram. 2005. Activity-based computing-lessons learned and open issues. In *ECSCW 2005 workshop, Activity-From a theoretical to a computational construct*. Citeseer.

[4] Jakob E Bardram, Jonathan Bunde-Pedersen, Afsaneh Doryab, and Steffen Sørensen. 2009. Clinical Surfaces – Activity-Based Computing for Distributed Multi-Display Environments in Hospitals. In *IFIP Conference on Human-Computer Interaction*. Springer, 704–717.

[5] Jakob E. Bardram, Morten Esbensen, and Aurélien Tabard. 2016. Activity-Based Collaboration for Interactive Spaces. In *Collaboration Meets Interactive Spaces*, Craig Anslow, Pedro Campos, and Joaquim Jorge (Eds.). Springer International Publishing, Cham, 233–257. https://doi.org/10.1007/978-3-319-45853-3_11

[6] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (Jan. 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[7] Wenli Chen and Chee-Kit Looi. 2011. Active classroom participation in a Group Scribbles primary science classroom: Active classroom participation authors. *British Journal of Educational Technology* 42, 4 (July 2011), 676–686. https://doi.org/10.1111/j.1467-8535.2010.01082.x

[8] Pierre Dillenbourg. 2013. Design for classroom orchestration. *Computers & Education* 69 (Nov. 2013), 485–492. https://doi.org/10.1016/j.compedu.2013.04.013

[9] Pierre Dillenbourg and Fabrice Hong. 2008. The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning* 3, 1 (March 2008), 5–23. https://doi.org/10.1007/s11412-007-9033-1

[10] Pierre Dillenbourg, Guillaume Zufferey, Hamed Seyed Alavi, Patrick Jermann, Lenh Hung Son Do, Quentin Bonnard, Sébastien Cuendet, and Frédéric Kaplan. 2011. Classroom orchestration : The third circle of usability. In *Connecting Computer-Supported Collaborative Learning to Policy and Practice: CSCL2011 Conference Proceedings. Volume I — Long Papers*, Vol. 1. International Society of the Learning Sciences, Hong Kong, China, 510–517.

[11] Yannis Dimitriadis, Luis P. Prieto, and Juan I. Asensio-Pérez. 2013. The role of design and enactment patterns in orchestration: Helping to integrate technology in blended classroom ecosystems. *Computers & Education* 69 (Nov. 2013), 496–499. https://doi.org/10.1016/j.compedu.2013.04.004

[12] Sylvie Doré and Josianne Basque. 2007. Le concept d'environnement d'apprentissage informatisé. *International Journal of E-Learning & Distance Education* 13, 1 (2007), 40–56.

[13] Frank Fischer and Pierre Dillenbourg. 2006. Challenges of orchestrating computer-supported collaborative learning. In *87th AERA*.

[14] Cresencia Fong, Rebecca M Cober, Richard Messina, Eric Jackman, Tom Moher, Julia Murray, Ben Peebles, and James D Slotta. 2015. The 3R Orchestration Cycle: Fostering Multi-Modal Inquiry Discourse in a Scaffolded Inquiry Environment. In *Cscl*. 39–46. files/209/MC-0122-FullPaper-Fong.pdf

[15] George Robertson Mary Czerwinski Brian Meyers Daniel Robbins Greg Smith, Patrick Baudisch. 2003. GroupBar: The TaskBar Evolved. In *Conference for the Computer-Human Interaction Special Interest Group of the Human Factors Society of Australia*. https://www.microsoft.com/en-us/research/publication/groupbar-the-taskbar-evolved/

[16] Stian Haklev, Louis Pierre Faucon, Thanasis Hadzilacos, and Pierre Dillenbourg. 2017. FROG: rapid prototyping of collaborative learning scenarios. In *EC-TEL Demos*. event-place: Tallinn, Estonia.

[17] Steven Houben, Søren Nielsen, Morten Esbensen, and Jakob E. Bardram. 2013. NooSphere: an activity-centric infrastructure for distributed interaction. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia - MUM '13*. ACM Press, Luleå, Sweden, 1–10. https://doi.org/10.1145/2541831.2541856

[18] Ghita Jalal, Valentin Lachand, Aurélien Tabard, and Christine Michel. 2018. How Teachers Prepare for the Unexpected Bright Spots and Breakdowns in Enacting Pedagogical Plans in Class. In *Lifelong Technology-Enhanced Learning (Lecture Notes in Computer Science)*. Springer, 59–73.

[19] Steven Jeuris, Steven Houben, and Jakob Bardram. 2014. Laevo: a temporal desktop interface for integrated knowledge work. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. ACM Press, Honolulu, Hawaii, USA, 679–688. https://doi.org/10.1145/2642918.2647391

[20] Victor Kaptelinin. 2003. UMEA: translating interaction histories into project contexts. In *Proceedings of the conference on Human factors in computing systems - CHI '03*. ACM Press, Ft. Lauderdale, Florida, USA, 353. https://doi.org/10.1145/642611.642673

[21] Ahmed Kharrufa, David Leat, and Patrick Olivier. 2010. Digital mysteries: designing for learning at the tabletop. In *ACM International Conference on Interactive Tabletops and Surfaces - ITS '10*. ACM Press, Saarbrücken, Germany, 197. https://doi.org/10.1145/1936652.1936689

[22] Ahmed Kharrufa, James Nicholson, Paul Dunphy, Steve Hodges, Pam Briggs, and Patrick Olivier. 2015. Using IMUs to Identify Supervisors on Touch Devices. In *Human-Computer Interaction – INTERACT 2015 (Lecture Notes in Computer Science)*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, 565–583.

[23] Ahmed Kharrufa, Sally Rix, Timur Osadchiy, Anne Preston, and Patrick Olivier. 2017. Group Spinner: Recognizing and Visualizing Learning in the Classroom for Reflection, Communication, and Planning. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, Denver, Colorado, USA, 5556–5567. https://doi.org/10.1145/3025453.3025679

[24] Lars Kobbe, Armin Weinberger, Pierre Dillenbourg, Andreas Harrer, Raija Hämäläinen, Päivi Häkkinen, and Frank Fischer. 2007. Specifying computer-supported collaboration scripts. *International Journal of Computer-Supported Collaborative Learning* 2, 2-3 (Sept. 2007), 211–224. https://doi.org/10.1007/s11412-007-9014-4

[25] Ingo Kollar and Frank Fischer. 2013. Orchestration is nothing without conducting – But arranging ties the two together!: A response to Dillenbourg (2011). *Computers & Education* 69 (2013), 507 – 509. https://doi.org/10.1016/j.compedu.2013.04.008

[26] Ingo Kollar, Frank Fischer, and Friedrich W. Hesse. 2006. Collaboration Scripts – A Conceptual Analysis. *Educational Psychology Review* 18, 2 (Nov. 2006), 159–185. https://doi.org/10.1007/s10648-006-9007-2

[27] Stefan Kreitmayer, Yvonne Rogers, Robin Laney, and Stephen Peake. 2013. UniPad: orchestrating collaborative activities through shared tablets and an integrated wall display. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*. ACM Press, Zurich, Switzerland, 801. https://doi.org/10.1145/2493432.2493506

[28] Chee-Kit Looi and Yanjie Song. 2013. Orchestration in a networked classroom: Where the teacher's real-time enactment matters. *Computers & Education* 69 (Nov. 2013), 510–513. https://doi.org/10.1016/j.compedu.2013.04.005

[29] Roberto Martinez Maldonado, Yannis Dimitriadis, Judy Kay, Kalina Yacef, and Marie-Theresa Edbauer. 2012. Orchestrating a multi-tabletop classroom: from activity design to enactment and reflection. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces - ITS '12*. ACM Press, Cambridge, Massachusetts, USA, 119. https://doi.org/10.1145/2396636.2396655

[30] Roberto Martínez, Anthony Collins, Judy Kay, and Kalina Yacef. 2011. Who did what? Who said that?: Collaid: an environment for capturing traces of collaborative learning at the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*. ACM Press, Kobe, Japan, 172. https://doi.org/10.1145/2076354.2076387

[31] Roberto Martínez, Anthony Collins, Judy Kay, and Kalina Yacef. 2011. Who did what? Who said that?: Collaid: an environment for capturing traces of collaborative learning at the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*. ACM Press, Kobe, Japan, 172. https://doi.org/10.1145/2076354.2076387

[32] Philippe Meirieu. 2009. A quoi sert la pédagogie ?

[33] Michael J. Muller, Werner Geyer, Beth Brownholtz, Eric Wilcox, and David R. Millen. 2004. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. ACM Press, Vienna, Austria, 375–382. https://doi.org/10.1145/985692.985740

[34] Bonnie A. Nardi (Ed.). 2001. *Context and consciousness: activity theory and human-computer interaction* (third printing ed.). MIT Press, Cambridge, Mass. OCLC: 249516304.

[35] Heinz Neber, Sarah Sennebogen, Yamin Taisir Subhi, K. Linke Sandra, and E. Vidergor Hava. 2011. Classroom of the Future: Orchestrating Collaborative Spaces. *Gifted and Talented International* 26, 1-2 (March 2011), 165–167. https://doi.org/10.1080/15332276.2011.11673599

[36] Gilbert Paquette, Michel Léonard, and others. 2013. Modèles et métadonnées pour les scénarios pédagogiques. (2013).

[37] Pascal Plantard. 2016. Temps numériques et contretemps pédagogiques en Collège Connecté. *Distances et médiations des savoirs* 16 (Dec. 2016). https://doi.org/10.4000/dms.1660

[38] Gérard Puimatto. 2014. Numérique à l'École–usages, ressources, métiers, industries. *Distances et médiations des savoirs. Distance and Mediation of Knowledge* 2, 5 (2014), 21.

[39] María Jesús Rodríguez-Triana, Alejandra Martínez-Monés, Juan I. Asensio-Pérez, and Yannis Dimitriadis. 2015. Scripting and monitoring meet each other: Aligning learning analytics and learning design to support teachers in orchestrating CSCL situations: Scripting and monitoring meet each other. *British Journal of Educational Technology* 46, 2 (March 2015), 330–343. https://doi.org/10.1111/bjet.12198

[40] Jeremy Roschelle, Yannis Dimitriadis, and Ulrich Hoppe. 2013. Classroom orchestration: Synthesis. *Computers & Education* 69 (Nov. 2013), 523–526. https://doi.org/10.1016/j.compedu.2013.04.010

[41] Adam Rule, Aurélien Tabard, and Jim Hollan. 2017. Using Visual Histories to Reconstruct the Mental Context of Suspended Activities. *Human–Computer Interaction* 32, 5-6 (Nov. 2017), 511–558. https://doi.org/10.1080/07370024.2017.1300063

[42] Mike Sharples. 2013. Shared orchestration within and beyond the classroom. *Computers & Education* 69 (Nov. 2013), 504–506. https://doi.org/10.1016/j.compedu.2013.04.014

[43] M. Sharples, E. Scanlon, M. Paxton, L. Kerawalla, M. Feisst, M. Gaved, M. Wright, T. Collins, S. Anastopoulou, and P. Mulholland. 2012. nQuire: Technological Support for Personal Inquiry Learning. *IEEE Transactions on Learning Technologies* 5, undefined (2012), 157–169. https://doi.org/doi.ieeecomputersociety.org/10.1109/TLT.2011.32

[44] Aurélien Tabard, Juan-David Hincapié-Ramos, Morten Esbensen, and Jakob E. Bardram. 2011. The eLabBench: an interactive tabletop system for the biology laboratory. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*. ACM Press, Kobe, Japan, 202. https://doi.org/10.1145/2076354.2076391

[45] Pierre Tchounikine. 2013. Clarifying design for orchestration: Orchestration and orchestrable technology, scripting and conducting. *Computers & Education* 69 (Nov. 2013), 500–503. https://doi.org/10.1016/j.compedu.2013.04.006

[46] Stephen Voida and Elizabeth D. Mynatt. 2009. It feels better than filing: everyday work experiences in an activity-based computing system. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. ACM Press, Boston, MA, USA, 259. https://doi.org/10.1145/1518701.1518744


Received November 2018; revised February 2019; accepted April 2019