IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 31 October 2024, accepted 9 January 2025, date of publication 17 January 2025, date of current version 7 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3531241

RESEARCH ARTICLE

Measurement of Teacher's Orchestration Load: A Framework and a Case Study on Tool Flexibility

CLAUDIO ALVAREZ[1], ISHARI AMARASINGHE[2], GUSTAVO ZURITA[3], DAVINIA HERNANDEZ-LEO[4], LUBNA HAKAMI[4], AND LUIS ROJAS[5]

[1]Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Las Condes, Santiago 7620001, Chile
[2]Department of Artificial Intelligence, Faculty of Social Sciences, Radboud University, 6500 HE Nijmegen, The Netherlands
[3]Faculty of Economics and Business, University of Chile, 8330015 Santiago, Chile
[4]Department of Information and Communication Technologies, Universitat Pompeu Fabra, 08018 Barcelona, Spain
[5]Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Recoleta, Santiago 8420524, Chile

Corresponding author: Claudio Alvarez (calvarez@uandes.cl)

This work was supported in part by Ministerio de Ciencia, Innovación y Universidades (MICIU)/Agencia Estatal de Investigación (AEI)/10.13039/501100011033 under Grant PID2020-112584RB-C33, Grant PID2023-146692OB-C33, and Grant CEX2021-001195-M; in part by Support to Research Groups (SGR) under Grant 00930; and in part by the Annual Internal Competition for Research support funds, ''RESEARCH PROJECTS COMPETITION - FEN 2020'', Faculty of Economics and Business (FEN), University of Chile. The work of Davinia Hernandez-Leo was supported by Institución Catalana de Investigación y Estudios Avanzados (ICREA) Academia.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Internal Committee for Ethical Review of Projects (CIREP) at Universitat Pompeu Fabra, Barcelona, Spain.

ABSTRACT Teacher orchestration in Computer-Supported Collaborative Learning (CSCL) environments demands managing multiple tasks across different social levels, often under tight constraints, leading to an increased orchestration load. This load represents the cognitive and physical effort teachers invest in real-time coordination of learning activities, which remains underexplored, particularly how the flexibility of orchestration tools influences this burden. In response to this gap, we propose a comprehensive framework for tracking and characterizing orchestration load, focusing on the intensity and dynamics of teachers' actions while implementing CSCL scripts. The framework integrates multimodal data sources, including observable orchestration actions, physiological metrics, and self-reported insights, to comprehensively analyze the orchestration load. We illustrate the applicability of the framework through a case study comparing two CSCL orchestration tools: PyramidApp, which offers pre-configured, structured enactment of the Pyramid collaborative learning flow pattern, and EthicApp, a more flexible tool allowing real-time design adjustments for a variety of instructional designs. Our findings reveal that PyramidApp facilitates a streamlined orchestration process with lower intensity and reduced teacher workload. At the same time, EthicApp's high flexibility increases orchestration intensity, requiring more cognitive effort from teachers during real-time phase configuration. These results underscore the importance of balancing flexibility with usability in orchestration tool design, as overly flexible environments may overwhelm teachers, especially during high cognitive load scenarios. This study contributes a methodological framework for evaluating orchestration tools and highlights key design considerations for reducing orchestration load in CSCL environments.

INDEX TERMS Measurement framework, orchestration load, collaborative tools, case study.

I. INTRODUCTION

This notion of orchestration refers to how teachers coordinate and manage classroom activities at various scales, such as individual, group, and whole-class activities, under real-time constraints [1], [2]. Teachers are often compared to orchestra conductors, guiding students to achieve learning outcomes in real-time [3], [4]. Orchestration is a demanding task due to the need to monitor activities across different social planes (individual, group, whole class) and evaluate student work under various constraints, such as time and curriculum [5]. The load teachers experience while managing these tasks is known as the orchestration load [6], [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Ravibabu Mulaveesala.

Measuring orchestration load is essential to assess the pedagogical usability of tools and technologies used in the classroom beyond their technical functionality [5], [6]. To date, methods like ElectroDermal Activity (EDA) and eye-tracking have been used to evaluate teachers' cognitive and physical demands while facilitating collaborative learning [8].

Understanding orchestration load in Computer-Supported Collaborative Learning (CSCL) environments is essential, as these platforms enhance peer interactions through structured scripts [9], promoting socio-cognitive processes [10]. Collaborative Learning Flow Patterns (CLFPs), such as Pyramid and Jigsaw, guide collaboration and student engagement [11]. However, deviations from scripts due to real-world factors, such as student absences or time constraints, increase the orchestration load and call for flexible orchestration tools [3], [12].

While research emphasizes flexibility in orchestration technologies, more empirical evidence is needed on how flexibility affects orchestration load in real-time [13]. Current studies provide conceptual insights but lack data on how flexible tools influence teachers' ability to manage scripted CSCL activities in authentic settings [6], [14]. A multimodal approach incorporating various data sources could help assess both the usability and pedagogical value of orchestration tools, offering valuable insights for technology designers, teachers, and administrators [15].

This study introduces a framework for tracking orchestration load based on multimodal data, incorporating observable actions, physiological measurements (e.g., heart rate and skin conductance), and teachers' self-reports. The framework conceptualizes orchestration load as comprising two distinct constructs: *intensity* and *dynamics*. Intensity refers to the overall effort expended by the teacher in orchestrating classroom activities within a given time unit. In contrast, dynamics pertains to the sequencing and transitional patterns of orchestration actions and physiological states over time.

Using this framework, we investigate how runtime flexibility influences orchestration load during the enactment of instructional designs. Specifically, we compare the orchestration load and usability of two orchestration technologies for implementing the Pyramid CSCL pattern: one tool explicitly designed for this pattern [16], and another that provides teachers with runtime flexibility by allowing sequential specification of CSCL script phases, thereby supporting a broader range of learning designs [17]. The research questions guiding our effort to demonstrate the framework and its applicability for evaluating CSCL orchestration technologies are as follows:

RQ1: Can the proposed framework distinguish differences in the intensity and dynamics of orchestration load generated by different CSCL orchestration tools when implementing a similar learning design?

RQ2: How do flexible mechanisms in CSCL orchestration tools affect teachers' orchestration load in terms of intensity and dynamics?

The subsequent sections detail the theoretical foundation of the study, the proposed orchestration load framework, a case study, the discussion of results and implications, and concluding insights.

II. RESEARCH BACKGROUND

A. MEASUREMENT OF ORCHESTRATION LOAD

Orchestration load refers to the cognitive demands placed on teachers as they manage multiple learning activities in real-time, coordinating tasks across individual, group, and class-wide levels [3], [4]. This concept emphasizes the mental effort required to distribute attention and process classroom information under time constraints, making it a challenging task [7]. Understanding orchestration load is fundamental for designing technologies that can alleviate excessive workloads and improve teacher efficacy and well-being [14].

Prieto et al. [18] were among the first to measure orchestration load in a technology-enhanced classroom using a multi-method approach. They combined eye-tracking data (e.g., pupil diameter, saccade speed) with post-hoc recall interviews where teachers reviewed video snippets and rated their mental effort. The study also coded teacher activities and social planes to identify factors contributing to high or low orchestration load. Later, Prieto et al. [19] refined this method by using principal component analysis (PCA) to estimate an orchestration load score (OLS), combining eye-tracking metrics and process variables, such as teaching activity and attention focus. While the PCA-based OLS provided a robust load measure, it simplified the analysis at the expense of interpretability, potentially obscuring specific causes of orchestration load.

Amarasinghe et al. [8] focused on measuring orchestration load during collaborative learning, using a multi-method approach that combined screen captures, post-session questionnaires, and Epistemic Network Analysis (ENA). ENA revealed distinct patterns in how teachers allocated attention to epistemic and social aspects of learning, mainly when using dashboard support. The study identified three key facets of orchestration: situation evaluation, goal formation, and action-taking.

Situation evaluation involves how a teacher assesses the current learning environment. This typically requires the teacher to interpret and understand data on a dashboard or other platforms. Goal formation is setting goals or objectives based on assessing the learning situation. Once teachers evaluate the current learning situation, they establish objectives to optimize the learning activity. Action-taking is when teachers implement strategies or interventions based on the goals they have formed. After forming these goals, teachers determine the most appropriate actions to achieve them.

Amarasinghe et al.'s [8] facets of orchestration can be linked to orchestration actions defined by Tchounikine [13]. Tchounikine identified three types of orchestration actions: primo-scripting actions, runtime-scripting actions, and monitoring actions. Primo-scripting actions concern planning a learning situation and the design of the corresponding learning scenario. This is prior to the enactment of the learning design in the classroom. However, orchestration actions for goal setting can be embedded within this context, given that the definition of a learning scenario (e.g., learning design or script) has an underpinning pedagogical rationale and educational objectives that must be met [20]. Runtime-scripting actions concern modifying and adapting the initial learning scenario while it is being played. These actions can relate to all three facets defined by Amarasinghe et al. [8]. At runtime, the teacher may perform orchestration actions (i.e., facet of action taking) to adjust a script based on situation evaluation after goals that have been set (i.e., prior goal setting). Monitoring actions concern the assessment of the student work and classroom management (e.g., answering questions or guiding students), which is also linked to three facets of orchestration identified by Amarasinghe et al. [8]. For instance, through the use of dashboards, the teacher can evaluate the current situation, redefine goals, and take pedagogical actions (e.g., remedial or formative) [21], [22]. Realization of the three types of actions defined by [13] posits that orchestration is both digital and physical [23].

Hakami et al. [24] proposed an approach to estimate orchestration load in scripted CSCL activities by triangulating self-reported data (e.g., stress and cognitive demand scores) with objective sensor data (e.g., EDA peaks). Their analysis revealed that orchestration load can be influenced by task type, activity duration, and student numbers.

B. EFFECT OF FLEXIBILITY MECHANISMS ON ORCHESTRATION LOAD

CSCL scripts are designed to structure student interactions and scaffold learning, but a balance is needed between providing enough scaffolding and avoiding over-scripting, which can hinder natural collaboration [20]. Scripts must offer enough flexibility to adapt to the fluid dynamics of classroom interactions. Flexibility can be introduced during different phases of the script's lifecycle [12]. In the editing phase, CLFPs like Jigsaw or Pyramid reduce the cognitive load for designing activities but also limit modifications without altering the script's core pedagogical intent.

In the enactment phase, flexibility is crucial when dealing with unforeseen circumstances, such as student absences or timing issues. Teachers may need to reassign groups, adjust phases, or modify tasks to meet learning objectives. These adaptations, though necessary, increase the orchestration load as teachers manage instructional, logistical, and social aspects simultaneously. Moreover, if too many script elements are left flexible, decision fatigue can set in, further increasing orchestration load and frustration. Thus, CSCL script design must strike a balance between flexibility and minimizing cognitive demands on teachers during real-time orchestration [6].

In the broader context of Technology-Enhanced Learning (TEL) ecosystems, the affordances of ICTs, such as AI and learning analytics, are expected to make teaching more efficient [21], [22]. However, these complex environments often increase teachers' orchestration load, as they must design, manage, and adapt learning scenarios under real-world conditions. A human-centered approach to orchestration technology design is essential, ensuring alignment with teachers' practices and needs. Multimodal data capture methods should maintain ecological validity to assess orchestration load in authentic settings, using structured protocols for systematic analysis and reproducibility [25], [26].

III. RESEARCH CONTRIBUTION AND NOVELTY

This study builds on prior research in CSCL orchestration, which has consistently recognized orchestration load as a complex construct measurable through diverse methods, such as physiological data, observational metrics, and self-reports. While the studies discussed earlier have explored distinct aspects of orchestration load—identifying variables and processes that influence it, as well as developing methods to quantify and interpret it—a key shortcoming of these approaches lies in their limited focus on evaluating how orchestration (or orchestratable) technologies and their design decisions impact teachers' orchestration load.

This study contributes to the field of CSCL orchestration research by introducing a novel framework for conceptualizing and analyzing orchestration load through its dual constructs: intensity and dynamics. By decomposing orchestration load into these two components, the study provides a nuanced perspective on the cognitive and physical demands placed on teachers during classroom orchestration tasks supported by technology, opening new avenues for evaluating the usability and design of orchestration technologies.

In addition to its theoretical contributions, the case study presented here proposes and demonstrates innovative methods for measuring and interpreting orchestration load based on multimodal data, involving timeline heatmap visualizations for load intensity, and stochastic analysis for load dynamics. The analytical tools proposed enhance the depth and reliability of orchestration load measurement as per usability evaluation, compared to existing methods, which have rather focused on aspects such as teacher's coordination and management of the learning activity, along with social and epistemic aspects of teachers' orchestration load [8], [19].

A significant practical contribution of this work is the application of the framework in a real-world case study. The study compares two distinct CSCL orchestration tools—PyramidApp and EthicApp—illustrating how flexible orchestration mechanisms affect teacher orchestration load in terms of intensity and dynamics. By capturing and analyzing these effects, the case study provides actionable insights into the trade-offs between flexibility and cognitive demand in the design of orchestration technologies. This evidence addresses a critical gap in understanding the interplay between tool usability and teacher workload during real-time orchestration.

IV. A FRAMEWORK FOR TRACKING AND CHARACTERIZING ORCHESTRATION LOAD

A. OVERVIEW

In this research, we contend that CSCL orchestration technologies can be evaluated and improved in terms of their technical and pedagogical usability based on analyzing the orchestration load they impose on teachers during the execution of CSCL scripts. For these purposes, we propose the framework illustrated in Fig. 1, which is inspired by the previously reviewed research, as it is based on the conceptualization of orchestration load according to the facets described by [8], and allows for the multimodal processing of diverse concurrent measurement sources [19]. Its results are grounded in the triangulation of these information sources [24].

The framework is aimed at educational technology designers and researchers, and it outlines a series of mandatory and optional activities to track and characterize the orchestration load generated by CSCL tools on teachers. Rather than providing a standardized measurement of orchestration load over time, as other approaches have focused on techniques such as dimensionality reduction [19], the framework aims to offer an explainable and comprehensible characterization of the teacher's orchestration load for human actors. This, in turn, enables the evaluation and refinement of technology to enhance usability for teachers and minimize orchestration load. The framework can be integrated into design-based research processes as an evaluation tool to inform the iterative refinement of educational technology.

B. PRE-REQUISITES

As in traditional usability testing, evaluating a teacher's orchestration load within an authentic operational environment is essential. This environment can be face-to-face or online and should involve active student participation. Conducting interventions in a teacher's regular courses is particularly beneficial as these settings naturally offer higher ecological validity due to real-world teaching conditions. Furthermore, a comprehensive characterization of the orchestration load induced by a CSCL tool necessitates a sample of teachers who are proficient in using the tool and can be systematically monitored. These teachers must be adequately trained in the orchestration technology to ensure they can effectively implement their activities according to the supported instructional design. The sample size does not need to be extensive; following the principles common in usability testing, a group of five to eight teachers is typically sufficient to reach information saturation.

This framework does not make assumptions about the technologies that may be implemented for concurrent, real-time multimodal data collection nor the technological capabilities applicable for real-time data analysis. Instead, the framework provides a methodological guide that facilitates a retrospective and comprehensive analysis of multimodal data and, from these analyses, evaluates the orchestration load generated by the orchestration (or orchestratable) technologies used by teachers.

It is possible to conduct this type of analysis at different levels of technological maturity and adoption, for example, in a fully automated manner, using environments capable of capturing multimodal sources, such as screen recording, eye tracking, EDA sensors, cameras, etc., and through automated teaching analytics techniques [19] for observation, coding and data triangulation, or in a partially automated form. The framework is even applicable under a traditional manual approach to observation, data capture, and information coding.

C. CSCL SCRIPT EXECUTION

The initial activity defined by the framework involves executing a CSCL script, supported by orchestration tools and/or orchestrate tools [13], to analyze their effects on the teacher's orchestration load. Upon execution, the teacher's orchestration effort can be observed by researchers, provided that the observation of script enactment and orchestration is conducted using non-invasive procedures, thus maintaining the ecological validity of the data to be collected.

D. OBSERVATION OF TEACHER'S ORCHESTRATION ACTIVITY

The framework conceptualizes a teacher's activity, particularly orchestration activity, within the lens of socio-cultural activity theory [27], [28] (Activity 3.A). According to this theory, any activity is a dynamic system in which the teacher (the subject) engages with the learning activity (the object), governed by rules and mediated through tools. In orchestration, the teacher acts upon the learning activity by facilitating and mediating students' learning processes. This mediation occurs through specific orchestration actions, carried out using orchestration tools in the environment, such as dashboards [21], and through interaction with the students (the community), communicated via language and physical actions.

Orchestration facets represent the psychological and cognitive processes that underpin these actions, as elaborated by [8]. These facets include processes such as 'situation evaluation' and 'goal formation' (see activities 2.A.1 and 2.A.2 in Fig. 1), which are typically internal to the teacher and remain unobservable unless explicitly externalized through interaction with the community (students and groups), and the environment (i.e., orchestration technologies and other digital tools). These internal facets guide the teacher's orchestration decisions, helping them assess the learning environment and form pedagogical goals.

In contrast, more tangible orchestration facets, such as 'action taking' (i.e., activity 2.A.3 in Fig. 1), become evident in the physical classroom setting. This facet involves

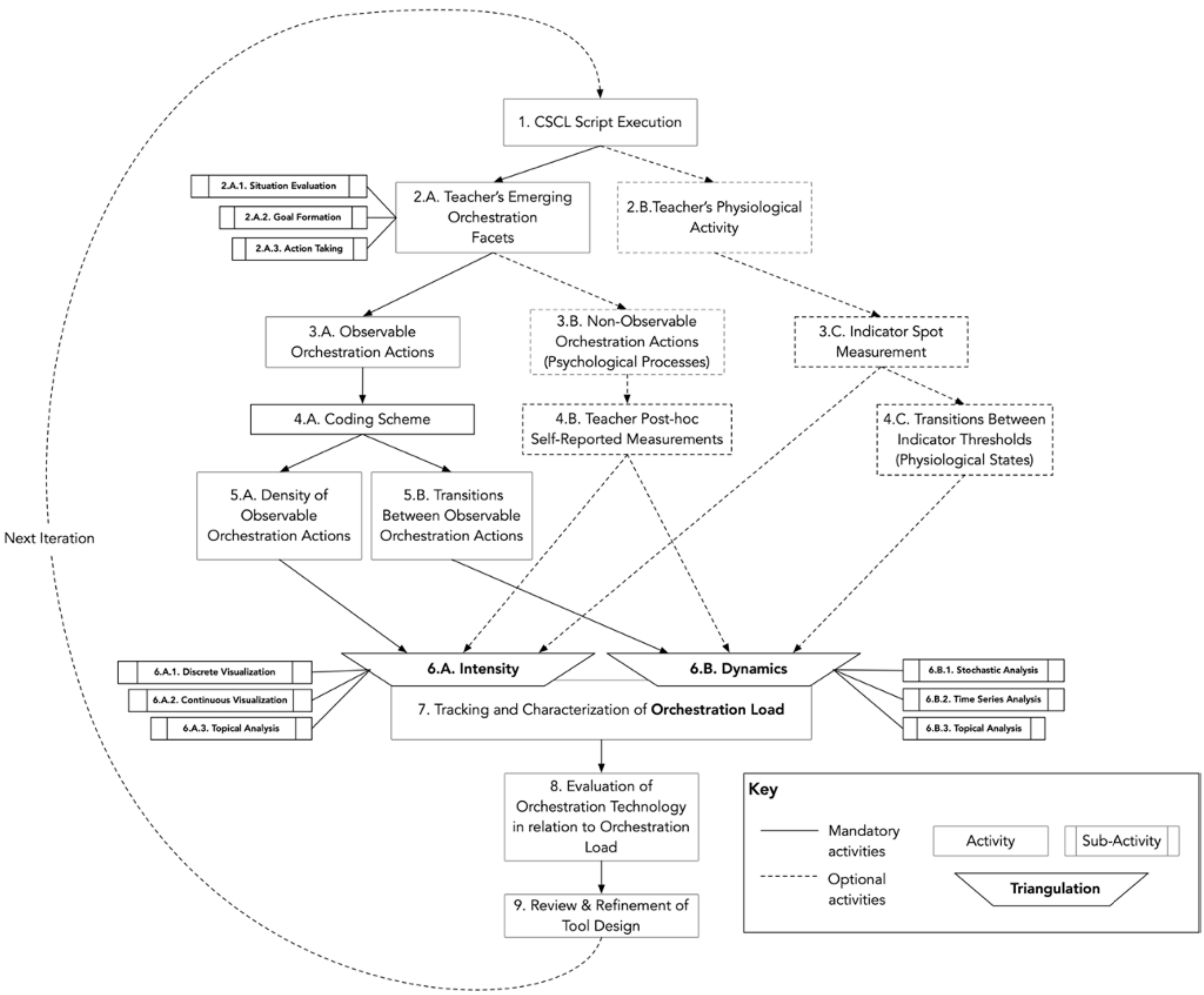

FIGURE 1. Framework process and activities.

the teacher's direct actions, such as guiding, instructing, or adjusting the learning activity in real-time. Moreover, monitoring their physiological responses can complement the teacher's orchestration activity (Activity 2.B. in Fig. 1). Technologies such as heart rate monitors [29], EDA sensors, brain-computer interfaces, and pupil dilation trackers can continuously measure these responses, providing indirect insights into the orchestration load that may not be immediately observable through outward actions alone.

Thus, the interplay between internal facets (cognitive processes) and external orchestration actions forms the foundation of the teacher's orchestration activity, with tools and technologies offering both direct and indirect ways to measure and support this complex process.

E. CODING

Interpreting observable orchestration actions requires the definition of a cogent typology, i.e., a coding scheme (Activity 4.A. in Fig. 1), which must be applied to the audiovisual capture of the teacher's activity, including filming, audio recording, and screen recording of the teacher's computer. Human coders can observe these sources manually or automatically through audio and video capture processing tools. This allows for a fine-grained analysis of how different orchestration efforts by the teacher are distributed over time.

Besides observable actions, physiological measures such as saccade frequency, pupil dilation, and heart rate can be treated continuously or discretized. Within a specific temporal basis (e.g., minutes or seconds), variations in these measures can be coded as being within defined intervals and above or below certain thresholds, which can be adjusted empirically or based on studies conducted in similar settings.

On the other hand, while the teacher's psychological processes account for the non-observable (Activity 3.B. in Fig. 1), they can be investigated post-hoc through surveys or interviews with the teacher (Activity 4.B. in Fig. 1). These can help determine, for example, at which moments of the activity the teacher perceived a greater demand in orchestrating the activity and even if they felt stressed or nervous at times during the activity. It is also possible to administer usability assessment tools to complement these insights.

F. ELICITATION OF ORCHESTRATION INTENSITY AND DYNAMICS

The capture and coding of observable orchestration actions (Activity 4.A. in Fig. 1), along with the collection of post-hoc data regarding the teacher's impressions of the orchestration load during the activity (Activity 4.B. in Fig. 1) and physiological measurement data (Activities 3.C. and 4.C. in Fig. 1) will allow for the tracking and characterization of the orchestration load, based on two empirical, observable aspects of the construct: intensity and dynamics.

We define the *intensity* of the orchestration load as the overall effort that the teacher expends while orchestrating the execution of the CSCL script over a unit of time. To characterize and track the intensity, considering the multimodality of the inputs, it is necessary to triangulate among different data sources (Activity 6.A. in Fig. 1). For this, we can operationalize intensity in its most basic form as the density of observable orchestration actions—that is, the number of teacher's orchestration actions observed during their activity—as a relevant metric for approximating a teacher's overall orchestration effort (Activity 5.A. in Fig. 1). Density can be analyzed using discrete visualizations, for instance, timeline heatmaps (Activity 6.A.1 in Fig. 1) or continuous line visualizations (Activity 6.A.2 in Fig. 1) in the case of physiological measurements. It is essential that these exploratory visualizations be explanatory and allow for straightforward interpretation, presenting both aggregated and disaggregated data for detailed inspection while avoiding clutter and adhering to recommendations such as those proposed by [30].

Additionally, teachers' self-reported post hoc measurements (Activity 4.B. in) can be subject to topic analysis (6.A.3 in Fig. 1), and this can complement the information from continuous and discrete visualizations. Furthermore, the intensity observation in a given time interval can be enriched by triangulating the density of observable orchestration actions with measurements of monitoring physiological indicators (Activity 3.C. in Fig. 1). This would allow for a comparison of different moments in the orchestration of the script and even the intensity of the comparison, considering different orchestration tools.

The *dynamics* of orchestration load refer to the sequencing patterns of orchestration actions (i.e., Activity 5.B. in Fig. 1) and patterns of transitions between physiological indicator thresholds (i.e., Activity 4.C. in Fig. 1). Both sources can be analyzed through stochastic characterization of how orchestration actions and physiological states transition from one to another over time (i.e., Activity 6.B.1. in Fig. 1), represented as Markov chain. A Markov chain is a mathematical model used to describe systems that transition between different states over time, where the probability of moving to the next state depends solely on the current state, not on the sequence of past states. In this context, Markov chains represent the various orchestration actions available to a teacher per the defined coding scheme and the probabilities of transitioning between these actions during the orchestration process (Activity 4.B. in Fig. 1). By analyzing these dynamics, it is possible to identify patterns in teachers' behavior and their use of support tools while orchestrating. Furthermore, time series analysis can be applied to the periods between actions or physiological states, allowing for the determination of, for example, the probability distributions of these times (i.e., Activity 6.B.2. in). The characterization of dynamics can also be informed by the teacher's self-reported measurements (Activity 4.B in Fig. 1), which can be processed using topical analysis (Activity 6.B.3. in Figure 1). Ultimately, the dynamics of orchestration load provide insights into the fluid and evolving nature of how teachers manage and adapt their strategies in response to the unfolding educational context, thus reflecting the probabilistic and sequential dependencies among different orchestration actions.

G. TRACKING AND CHARACTERIZATION OF ORCHESTRATION LOAD

Elaborating the characterization of the intensity and dynamics of orchestration load (Activity 7 in Fig. 1) is a central component of the framework. This process consolidates observations made through various artifacts generated during the triangulation of multimodal inputs (also part of Activity 7 in Fig. 1). For instance, graphical timelines can be constructed to visualize orchestration intensity through heatmaps, displaying minute-by-minute fluctuations. Additionally, the teacher's dwell times in different orchestration actions or states can be analyzed, along with the identification of sequencing patterns through a Markovian approach. The complementary use of physiological measurement devices further enriches the triangulation process, offering a more detailed description and enabling the identification of specific orchestration tool features or instructional design processes that contribute to a higher orchestration load for the teacher.

H. EVALUATION OF ORCHESTRATION TECHNOLOGY

Researchers can conduct descriptive or experimental evaluations of the orchestration technology with the orchestration load (Activity 8 in Fig. 1). They can detect situations of high orchestration load and determine common patterns of orchestration actions. This allows them to identify opportunities for improvement in orchestration technology and aim to enhance it to reduce the teacher's orchestration load.

I. REVIEW AND ITERATIVE REFINEMENT OF TOOL DESIGN

Tool design can be refined and improved, informed by the evaluation based on tracking and characterizing the orchestration load (Activity 9 in Fig. 1). After tools are enhanced, researchers and designers can iterate and repeat the steps

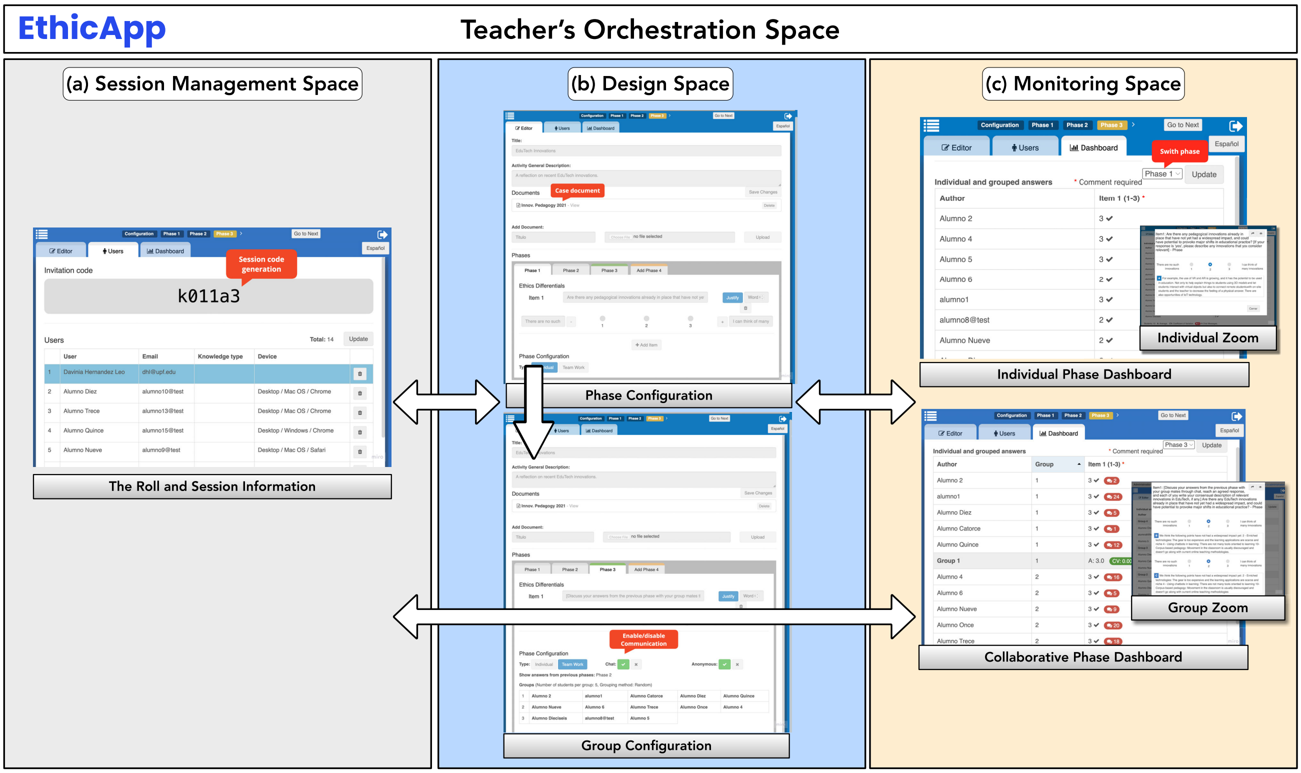

FIGURE 2. EthicApp orchestration space composed of (a) Session Management Space, (b) Design Space, and (c) Monitoring Space.

defined by this framework to validate whether the expected improvements have been achieved.

V. CASE STUDY

Building on the proposed framework for tracking and analyzing the teacher's orchestration load, and leveraging the Pyramid collaborative learning flow pattern to demonstrate its utility, we present a case study designed to explore the two guiding research questions of this work:

1) To what extent can the framework reveal differences in the intensity and dynamics of orchestration load across CSCL tools implementing similar learning designs?
2) How do flexible mechanisms in CSCL tools influence the intensity and dynamics of teachers' orchestration load?

A. CSCL PLATFORMS AND PATTERN UNDER STUDY

This study analyzes two CSCL platforms, PyramidApp [16] and EthicApp [17], using the proposed orchestration load framework to address the research questions. Both platforms are compared based on their impact on teacher orchestration load when implementing instructional designs derived from the Pyramid pattern. These tools were independently developed by two research teams in Chile (EthicApp) and Spain (PyramidApp), both of which include the present authors. EthicApp and PyramidApp represent contrasting approaches to orchestration—flexible (EthicApp) versus structured (PyramidApp)—providing a coherent basis for comparison. Additionally, both tools have been deployed in real-world CSCL environments, further establishing their relevance as case studies. This context offers a practical opportunity for an authentic comparison between two distinct tools within the scope of the proposed framework.

The Pyramid Pattern [11], a Collaborative Flow Learning Pattern (CLFP) [31], involves individual and group work, where participants propose solutions, discuss in groups, and merge their ideas into a final consensus. We chose the Pyramid Pattern because its synchronous implementation highly demands teacher orchestration, even with CSCL tools. Namely, the orchestration load in the Pyramid Pattern depends on class size. Teachers must review individual responses, monitor group work, and provide feedback while forming a mental representation of student progress through stages. This requires significant attention to student contributions and queries throughout the activity.

1) ETHICAPP

We hypothesize that the Pyramid Pattern will generate varying orchestration loads depending on the orchestration tools' features. PyramidApp, specifically designed to implement scripts based on this pattern, simplifies the teacher's task of designing and configuring activities. In contrast, EthicApp is a more flexible tool that allows teachers to adjust learning designs to their specific needs; however, this flexibility comes at the cost of increased interaction with the tool. EthicApp

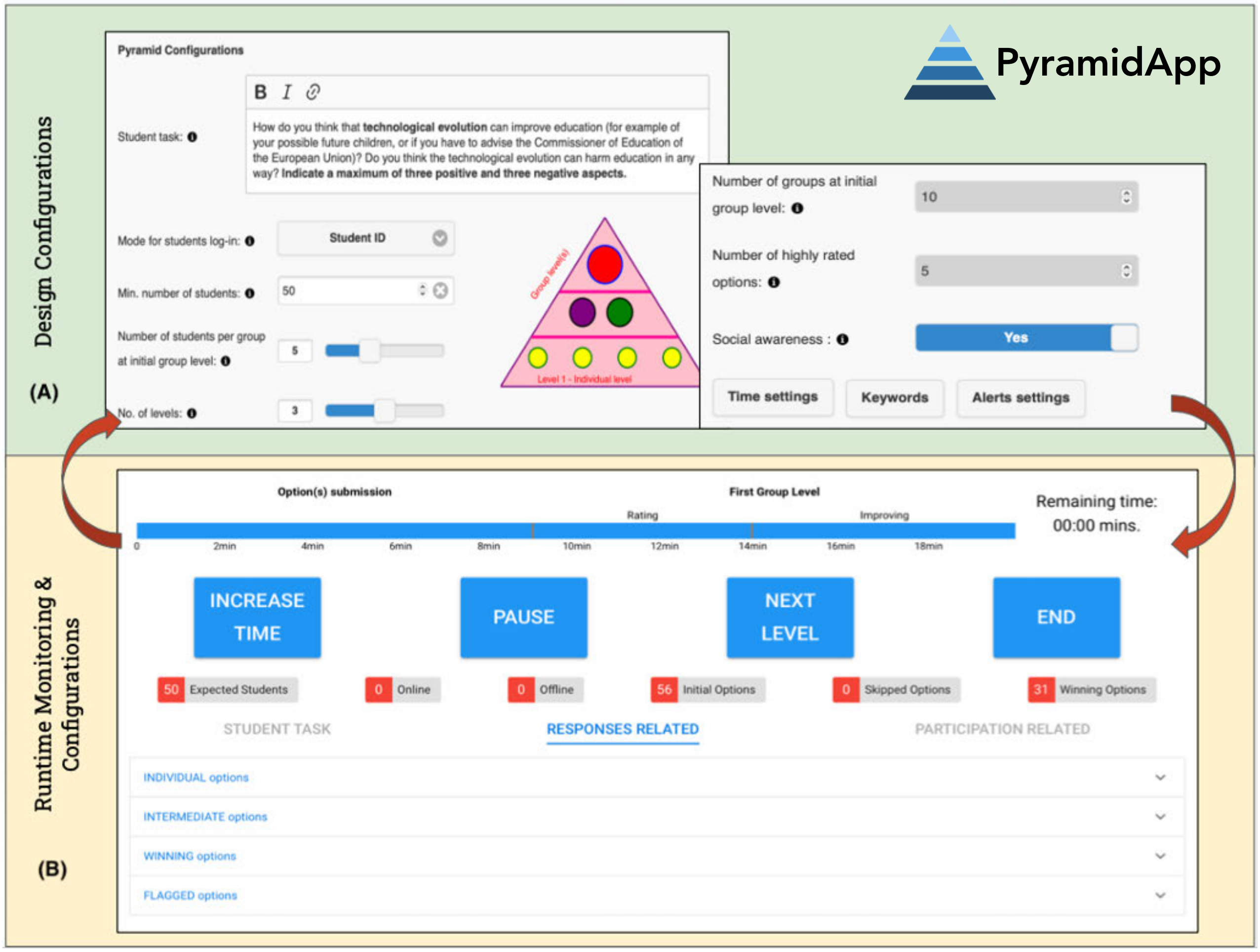

FIGURE 3. **PyramidApp is composed of (A) a Design-Time Editor, and (B) a Runtime Orchestration Space.**

requires the teacher to specify the script on-the-fly, phase by phase. This demands attention and orchestration actions that must be performed alongside routine tasks, such as reviewing student-generated content and interacting with students. Thus, the framework will allow us to thoroughly analyze the impact of different orchestration tool features on the teacher's orchestration load, focusing on intensity and dynamics. The following subsections describe the orchestration features of EthicApp and PyramidApp

EthicApp [17] is a flexible CSCL platform designed for analyzing ethical cases but can be used for other domains. It supports collaborative learning patterns, such as jigsaw, think-pair-share, and pyramid, with phases of individual and collaborative work. Teachers start a session, provide students with an access code, and monitor participation in the Session Management Space (Fig. 2(a)). Then, in the Design Space (Fig. 2(b)), they configure each phase at runtime, entering questions and setting parameters like group formation and response requirements. EthicApp doesn't store full designs, so teachers must design and monitor phases simultaneously, adding to the orchestration load in the Monitoring Space (Fig 2(c)).

Students typically spend 5-6 minutes responding individually if written justification is required, while collaborative phases take 10-15 minutes. A Pyramid activity with individual, group, and whole-class stages can take around 50 minutes.

2) PYRAMIDAPP

PyramidApp [16] is a CSCL platform designed specifically for the Pyramid pattern [11], unlike EthicApp, which allows flexible phase definition. Fig. 3(A) shows PyramidApp's Design Space, where teachers configure the activity before enactment, including task details, student log-in mode, group size, pyramid levels, time settings, and keywords for automatic dashboard highlights. At runtime, teachers can adjust phase duration, pause, skip levels, or end the activity early, as shown in Fig. 3(B).

PyramidApp phases have predefined durations, but teachers can modify them in real-time. Unlike EthicApp, PyramidApp imposes a visible timer for students and does not require chat discussions, making activities shorter. In EthicApp, transitions between phases are manually controlled, and time management is left to the instructor. Students must

TABLE 1. A comparison of Design configurations of PyramidApp and EthicApp.

Design Configurations	EthicApp[a]	PyramidApp[a]
Activity title and description	Yes(M)	Yes(M)
Login mode (student ID, anonymous)	Yes(M)	Yes(M)
Expected total number of students	No[b,c]	Yes(M)
Small group size	Yes[c,d](M)	Yes(M)
Number of group levels (i.e., number of small group levels and large group levels)	No	Yes(M)
Duration of different pyramid levels	No	Yes(M)
Set of questions	Yes[d](M)	NA
Semantic differential scale response per each question	Yes[d](M)	NA
Variable number of values in semantic differential scale	Yes[d](M)	NA
Interaction mode (individual/group)	Yes[d](M)	No
Configuration criteria for grouping	Yes[d](M)	No
Case document	Yes(O)	NA
Anonymity	Yes[d](O)	Yes
Include responses from previous phases	Yes[d](O)	No
Chat communication	Yes[d](O)	No
Enable/disable justification	Yes[d](O)	No
Justification length (in words)	Yes[d](O)	NA
Activity alerts that can inform critical moments of collaboration	No	Yes(O)
Configure keywords teachers would like to detect in students' options	No	Yes(O)

[a]Yes(M) refers to Yes, Mandatory and Yes(O) refers to Yes, Optional. NA refers to Not Applicable
[b]It can be manually controlled by the teacher, as the tool does not offer built-in functionality for this intent.
[c]No restrictions apply to the parameter, i.e., it can take any value
[d]The feature must be configured at runtime

TABLE 2. A comparison of Runtime Configurations of PyramidApp and EthicApp.

Real-time Configurations and Monitoring	EthicApp[a]	PyramidApp[a]
Transition to the next phase	Yes	Yes
Skip script phases	Yes	Yes
Terminate activity	Yes	Yes
Increase the time	No[b]	Yes
Pause the script	No[b]	Yes
Skip script phases	No[b]	Yes
Switch phase view	Yes	No
See students' and groups' progress in the currently selected phase	Yes	No
See table with summary of students' responses in an individual work phase	Yes	No
See detailed student's response in a specific phase	Yes	No
See table with summary of groups' responses in a collaborative phase	Yes	No
See detailed group responses in a specific phase	Yes	No
List groups with their amount of interaction (via chat)	Yes	No
Dashboard indicates groups with least and highest participation	Yes	Yes
Send messages to groups and vote on behalf of students	No	Yes

[a]Yes(M) refers to Yes, Mandatory and Yes(O) refers to Yes, Optional. NA refers to Not Applicable.
[b]Can be manually controlled by the teacher, as the tool does not offer built-in functionality for this intent.
[c]No restrictions apply to the parameter, i.e., it can take any value.

discuss via chat, leading to longer activity durations compared to PyramidApp.

Table 1 presents the script design features in EthicApp and PyramidApp. It is important to note that the teacher must configure most of the design features in EthicApp at runtime, including creating questions and setting response scales. When defining collaborative work phases, the teacher can configure features such as anonymity, the use of chat, and the grouping algorithm. In contrast, PyramidApp does not offer such design options at runtime.

Table 2 presents the features related to the execution and monitoring of scripts at runtime. Both tools are similar in terms of features. However, EthicApp provides the teacher with more possibilities for reviewing individual and group responses, as well as inspecting the discussions that take place in the chat.

B. FRAMEWORK ENACTMENT

1) EDUCATIONAL CONTEXT AND CSCL SCRIPT EXECUTION

We deployed activities using EthicApp and PyramidApp at a research university in Spain over two semesters: the first with PyramidApp and the second with EthicApp. A female teacher conducted the PyramidApp activity with 15 Master's students, and a male teacher conducted the EthicApp activity with the same cohort, both experienced with their respective tools.

For PyramidApp, script parameters (i.e., number of pyramid levels, group size per level, and time per level) were configured before enactment. In the case of EthicApp, the teacher used an external document (Google Docs) with all instructional design details, transcribing them into EthicApp's design interface without improvisation, despite the tool's flexibility allowing for real-time adjustments.

All students provided informed consent, and ethical approval was obtained for the study.

2) INSTRUCTIONAL DESIGN

The Pyramid pattern is executed with EthicApp and PyramidApp according to the following parameters. For EthicApp, the activity consists of three phases: individual submission, a small group discussion with five students, and a large group discussion. The individual submission phase lasts 10 minutes, followed by 15 minutes for small group discussion, and 20 minutes for the large group phase. Similarly, PyramidApp follows three steps: individual submission, a small group

TABLE 3. Coding scheme for observable teacher orchestration actions.

Code	Code Explanation	Subcodes & Further Details
Teacher-Student Interaction	Summarizes the interaction between the teacher and individual students.	Resolving doubts: Responds to specific questions raised by individual students.
Teacher-Class Interaction	Captures the bidirectional interactions between teachers and the students in the whole class.	Giving directions: Provides directions on how to use the CSCL tool. Describing tasks: Describes the task to the class. Debriefing: Discusses the final selection of answers at the end of the class. Praising: Praises the whole class. Surveying: Requests information from the class.
Announcements to Class	Teacher's announcements to the class about time remaining and progress.	Time remaining: Announces the time remaining for the activity. Phase transitions: Announces the phase transitions of the script. Work progress: Announces how many students and groups have finished.
Checking Dashboard	Summarizes teacher inspecting information in the teacher-facing dashboard.	Individual phase: Inspecting individual students' responses on the dashboard. Group phase: Inspecting a group's responses on the dashboard. Manual update: Press button on the dashboard to refresh screen. Scroll up/down: Inspect results grid.
Preparation/design the next phase	Summarizes teacher's actions regarding preparation of the next phases using EthicApp's design interface, or by clicking buttons (Next Level or End buttons) in the PyramidApp Dashboard.	Toggle chat: (enable/disable); Choose phase type; Toggle anonymity: (enable/disable); Select responses from previous phases; Set number of students per group; Choose a grouping algorithm; Generate groups; Start next phase; Click the next level button; Click the end button.
Checking Users	Summarizes the actions conducted by the teacher in the Users' tab. It only applies to EthicApp.	Inspect list of connected users. Manual update. Scroll up/down.
Transition between dashboard views	Summarizes the transition between dashboard views or tabs.	

discussion with three students, and a large group discussion. However, the duration of each phase is shorter: three minutes for individual submission, six minutes for small group discussion, and six minutes for the large group phase.

The Pyramid activities developed with both tools are based on the same question content. The question guiding the students' work through the individual and group work phases is: *"Are there any pedagogical innovations already in place that have not yet had a widespread impact and could potentially provoke major shifts in educational practice?"*.

Behavioral data of teachers was captured using screen recording software and audio recording to track their interaction with computers, the CSCL tools used, and interactions with students. Additionally, log data captured the timestamped actions of both teachers and students.

3) CODING SCHEME DEFINITION AND APPLICATION

A typology of observable orchestration actions and a coding scheme were defined based on previous work by the authors [8]. Categories include 'Teacher-class interaction,' 'Teacher-student interaction,' 'Announcements to class,' 'Checking Dashboard,' and 'Transition between dashboard views' (see Table 3). These actions reflect key orchestration facets such as situation evaluation, goal formation, and action-taking.

One author transcribed the screen-captured data using speech recognition software. Two trained research assistants coded the transcriptions based on the coding scheme. They initially coded 20% of the events, then met to resolve differences and agree on criteria. Coding continued for the remaining events, with further consensus meetings to address discrepancies. Challenges included determining action granularity and accurately timing orchestration actions, especially when teacher interactions with the dashboard were interrupted by student questions, requiring separate time records for subsequent actions.

4) DENSITY OF OBSERVABLE ORCHESTRATION ACTIONS

To analyze the density of orchestration actions, we developed timeline heatmaps based on the orchestration actions observed during the runtime of scripts supported by the two orchestration tools comprised in this research. The timeline heatmaps are two-dimensional; the horizontal axis displays the minutes of elapsed teacher orchestration activity, and the vertical axis shows the types of tracked orchestration actions. For each minute and type of orchestration action on the timeline heatmap visualization, there is a colored cell under an intensity scheme using the Viridis color palette (suitable for colorblind individuals). More intense colors indicate a higher frequency of orchestration actions for the tracked type of orchestration action. The gradation of the color scale is dependent on the maximum observed frequency. The timeline heatmaps presented here were generated using a set of scripts in R language version 4.3, utilizing packages such as ggplot, dplyr, and tidyr.

5) TRANSITIONS BETWEEN OBSERVABLE ORCHESTRATION ACTIONS

To analyze and interpret the dynamics of orchestration load arising from using the two tools, an analysis of transitions

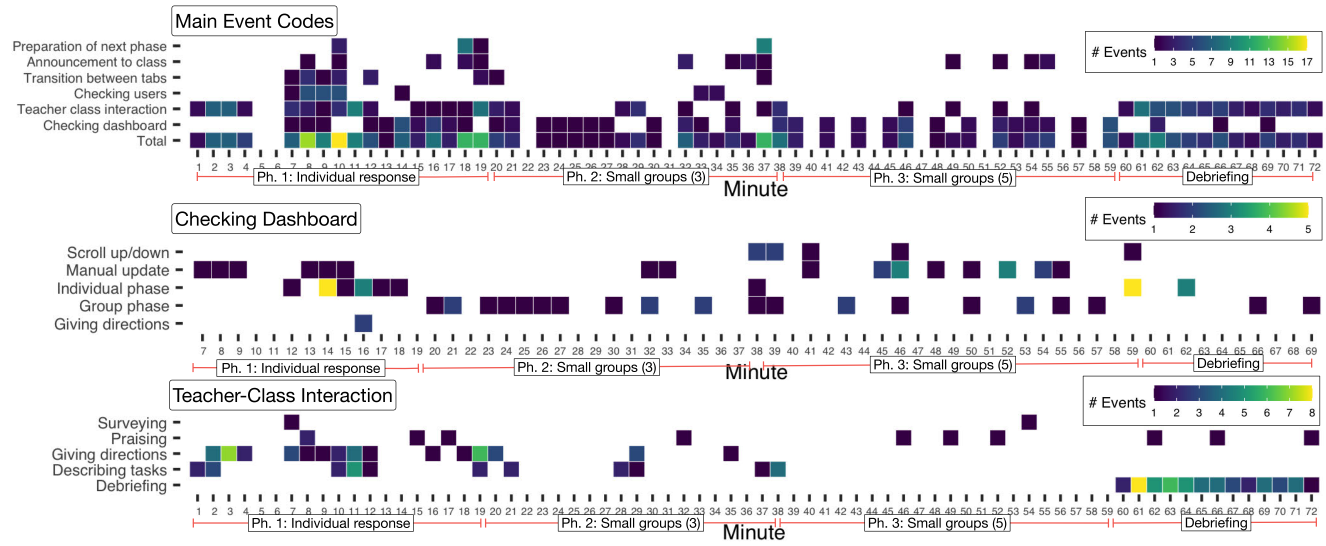

FIGURE 4. Timeline heatmaps of teacher's orchestration actions with EthicApp.

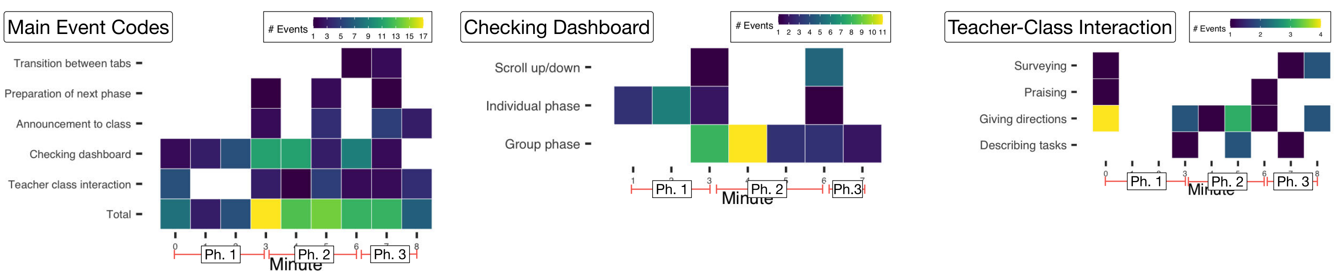

FIGURE 5. Timeline heatmaps depicting teacher's orchestration actions with PyramidApp.

among orchestration actions was conducted. For this, occurrences of orchestration actions with their temporal records were tabulated, and then transition probabilities between them were determined to model a Markov chain. This was accomplished using a script in R with the Markov chain package. As the proposed framework prescribes, Markov chains enable dynamic, stochastic analysis of teachers' orchestration actions by computing actual transition probabilities between orchestration action events according to the coding scheme.

In addition, the distributions of the dwell times in the states of the Markov chain were also determined to gain a clearer understanding of the time dedication of the teacher to the various possible states in their orchestration effort.

C. RESULTS

1) INTENSITY OF ORCHESTRATION LOAD

Interpretation of the intensity of orchestration load is based on the timeline heatmap visualizations in Figures 4 and 5, which are based on the orchestration action categories presented in Table 3, and the coding procedure performed, which registered the start time of each orchestration action.

In Fig. 4, the heatmap visualization for EthicApp shows that the teacher's activity began with preparations, which consisted of giving a briefing of the activity to the students (see minutes 1 to 10 of the "Teacher-Class Interaction" subcode in Fig. 4) and waiting for all students will log into the application (see minutes 7 to 10 of the sub code "Transition between tabs" in Fig. 4). At minute 10, the teacher began the Individual Response phase. The 10th minute had the highest orchestration load in the timeline, with 17 orchestration actions. These included all types of actions except for the Checking Dashboard (CHB). Thus, the teacher's orchestration load was among the highest observed.

During phase 1 of individual response (i.e., from 10:43 to 19:18), the teacher interacted with the dashboard and the students. The teacher interacted with the dashboard to review the incoming responses from the students and continually informed the students how many responses were still to be expected. In addition, the teacher occasionally reminded students of the remaining time. The teacher configured the successive phases of group work (see minutes 18 and 37 in the timeline for subcode "Preparation of the next phase" in Main Codes in Fig. 4), and numerous parameters had to be entered through the interface's design view. In the debriefing phase, the teacher's interaction with the dashboard was sporadic in search of

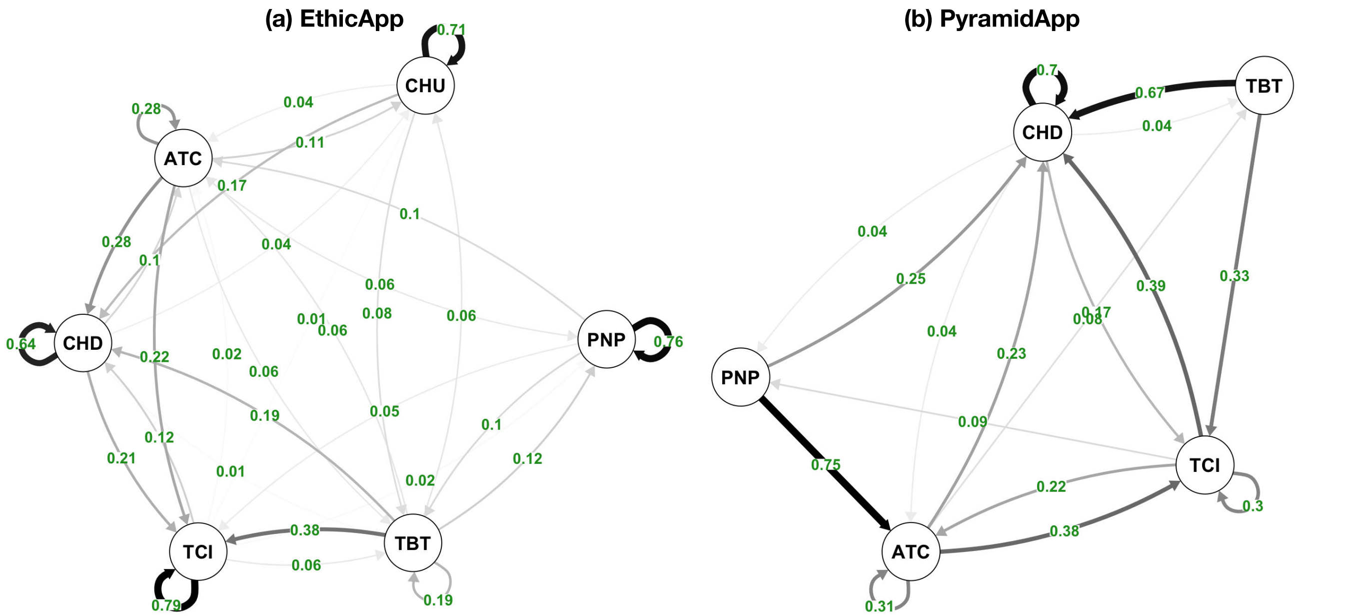

FIGURE 6. Markov chains for teachers' orchestration actions with (a) EthicApp and (b) PyramidApp.

answers to refer to some relevant justifications of the students.

The PyramidApp activity lasted more than eight minutes, including the Individual Response phase and two successive group phases with groups of 3 and 5 students. A high intensity of events can be observed in Fig. 4. Namely, between minutes 3 and 7 inclusive, 68 events are counted.

With PyramidApp, activity progress is comparatively faster, as stages last only three to four minutes. Minutes 3 to 7 have a high intensity, comparable to the highest intensities observed in EthicApp (See Fig. 5). With PyramidApp, students have a set time, controlled by the system, to submit their responses, so the teacher's announcements to the class are solely to introduce the next phase. Students submit short textual responses to the posed question. In contrast, with EthicApp, the teacher handles students' numerical and textual responses. Accessing this information requires navigating through the dashboard and performing "Zoom" actions to view the detailed written responses, both at the individual and group levels (see Fig. 2(c)). Additionally, in the group phases, there are synchronous chat conversations that the teacher also reviews. Students work in EthicApp without automatic time control, so the teacher must continuously announce the remaining time and manage the transition to the next phase by performing design tasks under the Preparation of the Next Phase category.

In contrast, despite the relatively high intensity of orchestration actions with PyramidApp, preparation for the next phase involves a minimal number of orchestration actions for the teacher, so the announcements are sporadic, occurring only at minutes 3, 5, 7, and 8 to signal the beginning and end of phases and the activity. Teacher-student interactions occur at a rate of 1 to 6 per minute. Additionally, the content generated by students in PyramidApp is purely textual and visible to the teacher more directly on the dashboard, without the need for zooming actions as in EthicApp.

2) DYNAMICS OF ORCHESTRATION LOAD

Interpretation of dynamics of orchestration load is based on analysis of Markov chains for EthicApp and PyramidApp presented in Fig. 6, which is complemented with analysis of dwell times of teacher's orchestration actions in both systems, as per Table 4.

To analyze dwell times, the durations of orchestration actions from EthicApp and PyramidApp, as recorded by the coders, were categorized by main action categories, and their distributions were characterized to determine normality using the Shapiro-Wilk test. In most cases, the time distributions were found not to meet the assumptions of normality; for EthicApp, only the distribution of the durations for Transition Between Tabs (TBT) was determined to be normally distributed ($W = .92$, n.s.). On the other hand, for the PyramidApp data, only the distributions of Preparation of the Next Phase (PNP) and Transition Between Tabs (TBT) were found to meet the assumptions of normality ($W = .20$, and $W = 0.07$, respectively, *n.s.*). Consequently, for comparative purposes between the time distributions of EthicApp and PyramidApp, the non-parametric Wilcoxon rank-sum test was chosen (see Table 4).

With EthicApp, the individual actions that demand the most teacher engagement in terms of time are Checking Dashboard (CHD), Announcements to Class (ATC), and Teacher-Class Interaction (TCI). With PyramidApp, CHD and TCI are the two actions with the longest time per event, followed by Preparation of the Next Phase (PNP). In contrast to EthicApp, ATC actions in PyramidApp take considerably

TABLE 4. Descriptive statistics and comparison of orchestration action dwell times between ethicapp and pyramidapp.

Main Code	EthicApp							PyramidApp							Comparison	
	Min	Q1	Median	Q3	Max	*M*	*SD*	Min	Q1	Median	Q3	Max	*M*	*SD*	*W*	*p*-value
Announcements to Class	1.0	4.0	6.0	23.0	78.0	17.7	22.1	2.0	2.0	3.0	3.0	8.0	3.17	1.50	55	<0.05
Checking Dashboard	1.0	4.0	8.5	36.5	150.0	27.3	37.0	1.0	3.0	5.0	6.0	30.0	7.65	6.54	95	<0.01
Preparation of the next phase	1.0	1.0	3.0	6.0	17.0	4.6	4.8	4.0	4.0	5.0	5.0	9.0	5.25	1.94	30	*n.s.*
Teacher-Class Interaction	1.0	5.0	10.0	16.0	174.0	13.5	18.8	1.0	4.0	5.0	7.0	28.0	8.12	6.94	82	<0.01
Transition Between Tabs	1.0	2.0	4.0	5.5	10.0	4.1	2.6	1.0	1.0	1.0	3.0	3.0	1.75	1.04	60	*n.s.*

less time (see medians and means of these events in Table 4). EthicApp's interface contains more details that the teacher must explain to the students when announcing a phase transition. For example, when the group phase starts, students need to activate a conversation interface by pressing a button to anonymously discuss their answers in groups of three while simultaneously being able to see their peers' responses. According to the analyzed transcripts, the teacher announced and explained this to the students.

On the other hand, the transition probability from ATC to TCI is 0.22 (see Fig. 6), while the reverse is only 0.02, indicating that the teacher is more likely to engage in TCI actions only after completing their ATC actions. A different dynamic occurred with PyramidApp. The teacher gave brief announcements to the students and engaged in interactions with them with considerably greater probability (0.38 vs. 0.22). Additionally, the teacher transitioned from TCI back to ATC with higher probability in PyramidApp than in EthicApp (0.22 vs. 0.02).

Transitions from ATC to CHD are similar in transition probabilities, with EthicApp at 0.28 and PyramidApp at 0.23. CHD has high self-transition probabilities in PyramidApp (0.7) and EthicApp (0.64). In both activities, the teacher monitors the dashboard while students work continuously. Once they begin orchestrating actions on the dashboard, they tend to remain in that state, resulting in sporadic transitions to the TCI state. For both EthicApp and PyramidApp, the transition probabilities from CHD to TCI were similar, at 0.21 and 0.17, respectively.

On the other hand, in PyramidApp, the Transition Between Tabs (TBT) state had a null self-transition probability, whereas in EthicApp, the probability was 0.19. This can be explained by the fact that in EthicApp, when a phase is nearing completion, the teacher tends to alternate between the design and dashboard tab to check if all students have responded before starting the next phase. Additionally, at the beginning of the activity, when students are joining the session, the teacher also tends to transition between the tab showing the list of connected users (the 'Checking Users' state) and the design view to configure the first phase of the activity and verify whether the number of connected users is sufficient to begin the first phase.

Finally, the 'Preparation of the Next Phase' (PNP) state is where the most significant differences between EthicApp and PyramidApp are found. In the case of EthicApp, the teacher must perform a series of configurations that take several atomic, sequential steps. With PyramidApp, this is much simpler, as the rules for the next phase are pre-configured, and the teacher only needs to advance. Thus, although there are no statistically significant differences in the durations of orchestration actions for the PNP category (see Table 4), the orchestration dynamics in EthicApp are more complex, as the teacher must execute a sequence of configurations, providing information for multiple design fields, all while students are working and occasionally asking questions (transition probability from PNP to TCI is 0.05).

VI. DISCUSSION

Regarding RQ1, *"Can the proposed framework distinguish differences in the intensity and dynamics of orchestration load generated by different CSCL orchestration tools when implementing a similar learning design?"*, both EthicApp and PyramidApp, based on the Pyramid CLFP, differ in their orchestration demands. PyramidApp facilitates a step-by-step execution with predefined configurations, leading to consistent yet intense orchestration actions. Teachers feel more secure due to fewer interface elements that could cause errors. In contrast, EthicApp extends the activity duration, as teachers must configure each phase in real-time, increasing their need to monitor and adjust the dashboard. The highest orchestration load occurs when configuring new phases, especially due to the absence of an undo function, adding cognitive strain even for experienced users.

The challenge grows as more students participate, increasing the content teachers must manage for feedback and debriefing. In larger cohorts, EthicApp's flexibility demands greater cognitive resources, while PyramidApp's structured design alleviates this burden during phase transitions.

Regarding the dynamics, transition probabilities between states are similar for both tools, except for the "Preparation of the next phase." Scalability could alter the dynamics, with increased teacher-student interactions leading to higher orchestration demands, particularly in EthicApp, where teachers must concentrate on designing the next phase while monitoring the current one.

As for RQ2, *"How do flexible mechanisms in CSCL orchestration tools affect teachers' orchestration load in terms of intensity and dynamics?"*, flexibility significantly impacts orchestration load. EthicApp offers high runtime flexibility, allowing for real-time adjustments, which, while beneficial for tailored instruction, increases cognitive effort. PyramidApp, with predefined configurations, reduces the need for adjustments, lowering the orchestration intensity and allowing teachers to focus more on pedagogical interactions.

Flexibility, particularly during runtime, should be balanced with usability to avoid overwhelming teachers. EthicApp could benefit from features that allow the pre-planning of activities and storing designs for later adjustment, thus reducing the orchestration load. Moreover, scalability challenges could be addressed by incorporating AI-driven tools, such as automated content selection, to help teachers manage large volumes of student-generated content in real-time.

The purpose of class orchestration with technology is to ease the teachers' burden in managing complex pedagogical practices [32]. Despite this, and the orchestration technologies presented in this study have been designed and implemented for everyday learning and teaching practices, the adoption process is slow. The findings of this study highlight the need for a balance between flexibility and usability in CSCL tool design, ensuring that adaptable features do not overwhelm teachers, and supporting effective orchestration while enhancing performance and outcomes.

This is particularly relevant today, given the rise of hybrid learning environments that have emerged since the COVID-19 pandemic. The complexities of using orchestration technologies (and orchestratable technologies) to address mixed cohorts of onsite and remote students introduce novel challenges regarding how to manage and reduce teachers' orchestration load. The proposed framework stands out as a tool not only for evaluating orchestration technologies in synchronous traditional settings but also for hybrid or fully online education scenarios.

While the proposed framework facilitates the evaluation and comparison of orchestration technologies, using the findings as feedback to improve usability and reduce orchestration load, it is equally important to highlight the role of adopting human-centered design (HCD) principles and methods to ensure better design decisions and outcomes [15]. Researchers employing this framework should be able to articulate results through storytelling techniques that resonate with technical and scientific stakeholders, practitioners, and even students, thereby fostering collective efforts to address design, technical, social, and logistical issues that impact orchestration load. Drawing on storytelling methods from fields such as data science and human-computer interaction can facilitate seamless dialogue across diverse roles and disciplines in co-participation or co-design processes.

Teacher awareness has emerged as a relevant concern in orchestration tecnologies [30] and it is commonly implemented through dashboards and alert or feedback systems similar to those presented and evaluated in this study. The proposed framework can thus serve as a valuable tool for improving systems and features aimed at enhancing teacher awareness.

In conclusion, case study presented here demonstrates how the proposed framework, with its dual focus on orchestration load intensity and dynamics, enables an explainable and comprehensive analysis of orchestration demands by integrating multimodal data sources. This holistic approach highlights the distinct challenges posed by flexibility and structure in CSCL environments and provides actionable insights that can inform the design and refinement of orchestration technologies. Future iterations of tools like EthicApp and PyramidApp can benefit from these observations, incorporating features that balance flexibility and cognitive ease, ultimately enhancing teacher efficacy and student learning outcomes.

VII. CONCLUSION AND FUTURE WORK

Designing classroom orchestration technologies requires careful consideration of orchestration load and flexible mechanisms. This study introduces a framework for measuring orchestration load, focusing on two key constructs: intensity, the overall effort teachers expend during orchestration; and dynamics, the sequencing and transitions of orchestration actions.

The case study highlights important considerations for designing real-time orchestration tools. When students generate content that the teacher must assimilate through a dashboard while monitoring individual and group progress, the opportunity to leverage flexibility features is limited. Flexibility realized as real-time parameter specification introduces an interaction overhead that significantly increases orchestration load, making it potentially counterproductive to prioritize high levels of flexibility if the orchestration setting does not allow for its effective use.

The case study compares two CSCL orchestration tools, PyramidApp and EthicApp. While PyramidApp's predefined design allows for quicker activity completion, EthicApp's flexibility in real-time configurations increases orchestration intensity, shifting the teacher's focus from students to tool management. These findings suggest that while flexibility is valuable, it can also be counterproductive in settings where real-time configuration is impractical.

Although this exploratory study was conducted with a small sample size—a factor that could be considered a limitation when interpreting the results—it effectively demonstrated the procedures and analyses of the proposed case study. The small sample size allowed for a clear and simplified examination of the contrasting effects of the different tool designs. The observed differences in orchestration load intensity and dynamics emerged distinctly from the inherent design variations between the tools, underscoring the framework's ability to reveal critical insights even with limited data. Building on these findings, this study opens the door for further research with larger samples, which would enable a more robust statistical analysis and a deeper understanding of the generalizability of the results. Future implementations of the framework should also integrate additional data sources, such as physiological measures and teacher self-reports, to capture a more comprehensive view of orchestration load.

To this end, we are currently investigating the use of sensors to measure heart rate, skin conductance, and temperature to gain deeper insights into the interplay between orchestration intensity and dynamics.

REFERENCES

[1] P. Dillenbourg et al., ''Classroom orchestration: The third circle of usability,'' in *Proc. 9th Int. Conf. Comput. Supported Collaborative Learn. (CSCL)*, Hong Kong, China, vol. 1, 2011, pp. 510–517.
[2] L. P. Prieto, M. H. Dlab, I. Gutié, N. A. Rrez, M. Abdulwahed, and W. Balid, ''Orchestrating technology enhanced learning: A literature review and a conceptual framework,'' *Int. J. Technol. Enhanced Learn.*, vol. 3, no. 6, pp. 583–598, 2011.
[3] P. Dillenbourg and P. Jermann, ''Technology for classroom orchestration,'' in *New Science of Learning: Cognition, Computers and Collaboration in Education*. New York, NY, USA: Springer, 2010, pp. 525–552.
[4] M. Sharples, ''Shared orchestration within and beyond the classroom,'' *Comput. Educ.*, vol. 69, pp. 504–506, Nov. 2013.
[5] S. Cuendet, Q. Bonnard, S. Do-Lenh, and P. Dillenbourg, ''Designing augmented reality for the classroom,'' *Comput. Educ.*, vol. 68, pp. 557–569, Oct. 2013.
[6] A. Van Leeuwen, ''Learning analytics to support teachers during synchronous CSCL: Balancing between overview and overload,'' *J. Learn. Analytics*, vol. 2, no. 2, pp. 138–162, Dec. 2015.
[7] L. P. Prieto, K. Sharma, L. Kidzinski, and P. Dillenbourg, ''Orchestration load indicators and patterns: In-the-Wild studies using mobile eye-tracking,'' *IEEE Trans. Learn. Technol.*, vol. 11, no. 2, pp. 216–229, Apr. 2018.
[8] I. Amarasinghe, D. Hernández-Leo, and H. U. Hoppe, ''Deconstructing orchestration load: Comparing teacher support through mirroring and guiding,'' *Int. J. Comput.-Supported Collaborative Learn.*, vol. 16, no. 3, pp. 307–338, Sep. 2021.
[9] F. Fischer, I. Kollar, H. Mandl, and J. M. Haake, *Scripting Computer-Supported Collaborative Learning: Cognitive, Computational and Educational Perspectives*, vol. 6. Berlin, Germany: Springer, 2007.
[10] F. Vogel, C. Wecker, I. Kollar, and F. Fischer, ''Socio-cognitive scaffolding with computer-supported collaboration scripts: A meta-analysis,'' *Educ. Psychol. Rev.*, vol. 29, no. 3, pp. 477–511, Sep. 2017, doi: 10.1007/s10648-016-9361-7.
[11] D. Hernández-Leo, J. I. Asensio-Pérez, and Y. Dimitriadis, ''Computational representation of collaborative learning flow patterns using IMS learning design,'' *J. Educ. Technol. Soc.*, vol. 8, no. 4, pp. 75–89, Jan. 2005.
[12] P. Dillenbourg and P. Tchounikine, ''Flexibility in macro-scripts for computer-supported collaborative learning,'' *J. Comput. Assist. Learn.*, vol. 23, no. 1, pp. 1–13, Feb. 2007.
[13] P. Tchounikine, ''Clarifying design for orchestration: Orchestration and orchestrable technology, scripting and conducting,'' *Comput. Educ.*, vol. 69, pp. 500–503, Nov. 2013.
[14] I. Amarasinghe, A. van Leeuwen, Y. Dimitriadis, A. Martínez-Monés, D. Hernández-Leo, H. U. Hoppe, K. Wiley, and R. Martinez-Maldonado, ''Teacher orchestration load: What is it and how can we lower the burden?'' in *Rapid Community Report Series, Digital Promise and the International Society of the Learning Sciences*. Hiroshima, Japan, 2022. [Online]. Available: https://repository.isls.org//handle/1/7665
[15] Y. Dimitriadis, R. Martínez-Maldonado, and K. Wiley, ''Human-centered design principles for actionable learning analytics,'' in *Research on E-Learning and ICT in Education*, T. Tsiatsos, S. Demetriadis, A. Mikropoulos, and V. Dagdilelis, Eds. Cham, Switzerland: Springer, 2021, pp. 277–296, doi: 10.1007/978-3-030-64363-8_15.
[16] K. Manathunga and D. Hernández-Leo, ''Authoring and enactment of mobile pyramid-based collaborative learning activities,'' *Brit. J. Educ. Technol.*, vol. 49, no. 2, pp. 262–275, Mar. 2018.
[17] C. Alvarez, G. Zurita, and N. Baloian, ''Applying the concept of implicit HCI to a groupware environment for teaching ethics,'' *Pers. Ubiquitous Comput.*, vol. 26, no. 6, pp. 1373–1391, Dec. 2022, doi: 10.1007/s00779-020-01495-z.
[18] L. P. Prieto, K. Sharma, and P. Dillenbourg, ''Studying teacher orchestration load in technology-enhanced classrooms,'' in *Proc. Eur. Conf. Technol. Enhanced Learn.*, Jan. 2015, pp. 268–281.
[19] L. P. Prieto, K. Sharma, Ł. Kidzinski, M. J. Rodríguez-Triana, and P. Dillenbourg, ''Multimodal teaching analytics: Automated extraction of orchestration graphs from wearable sensor data,'' *J. Comput. Assist. Learn.*, vol. 34, no. 2, pp. 193–203, Apr. 2018.
[20] P. Dillenbourg, ''Over-scripting CSCL: The risks of blending collaborative learning with instructional design,'' *Three Worlds CSCL. Can We Support CSCL*, pp. 61–91, Jan. 2002.
[21] B. A. Schwendimann, M. J. Rodríguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet, and P. Dillenbourg, ''Perceiving learning at a glance: A systematic literature review of learning dashboard research,'' *IEEE Trans. Learn. Technol.*, vol. 10, no. 1, pp. 30–41, Jan. 2017.
[22] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx, ''Learning dashboards: An overview and future research opportunities,'' *Pers. Ubiquitous Comput.*, vol. 18, no. 6, pp. 1499–1514, Nov. 2013.
[23] M. Quignard, P. Tchounikine, and P. Wang, ''Chao: A framework for the development of orchestration technologies for technology-enhanced learning activities using tablets in classrooms,'' *Int. J. Technol. Enhanced Learn.*, vol. 10, nos. 1–2, pp. 1–21, 2018.
[24] L. Hakami, D. Hernández-Leo, I. Amarasinghe, and B. Sayis, ''Investigating teacher orchestration load in scripted CSCL: A multimodal data analysis perspective,'' *Brit. J. Educ. Technol.*, vol. 55, no. 5, pp. 1926–1949, Sep. 2024.
[25] I. Dabbebi, J.-M. Gilliot, and S. Iksal, ''User centered approach for learning analytics dashboard generation,'' in *Proc. 11th Int. Conf. Comput. Supported Educ.*, 2019, vol. 2, no. 1, pp. 260–267.
[26] K. B. Yang, V. Echeverria, Z. Lu, H. Mao, K. Holstein, N. Rummel, and V. Aleven, ''Pair-up: Prototyping human-AI co-orchestration of dynamic transitions between individual and collaborative learning in the classroom,'' in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2023, pp. 1–17.
[27] V. Lachand, C. Michel, and A. Tabard, ''Toccata: Supporting classroom orchestration with activity based computing,'' *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 2, pp. 1–24, Jun. 2019.
[28] M. Schmidt, Y. Earnshaw, A. A. Tawfik, and I. Jahnke, ''Methods of user centered design and evaluation for learning designers,'' *Learner User Exper. Res.*, vol. 5, no. 2, pp. 1–129, 2020.
[29] A. Low and R. McCraty, ''Heart rate variability: New perspectives on assessment of stress and health risk at the workplace,'' *Heart Mind*, vol. 2, no. 1, pp. 16–27, 2018.
[30] V. Echeverria, R. Martinez-Maldonado, S. B. Shum, K. Chiluiza, R. Granda, and C. Conati, ''Exploratory versus explanatory visual learning analytics: Driving teachers' attention through educational data storytelling,'' *J. Learn. Anal.*, vol. 5, no. 3, pp. 73–97, Nov. 2018.
[31] D. Hernández-Leo, E. D. Villasclaras-Fernández, J. I. Asensio-Pérez, Y. Dimitriadis, I. M. J. Abellán, I. R. Requies, and B. R. Avi, ''COLLAGE: A collaborative learning design editor based on patterns,'' *J. Educ. Technol. Soc.*, vol. 9, no. 1, pp. 58–71, May 2006.
[32] Y. Song, ''A review of how class orchestration with technology has been conducted for pedagogical practices,'' *Educ. Technol. Res. Develop.*, vol. 69, no. 3, pp. 1477–1503, Jun. 2021.

CLAUDIO ALVAREZ received the Ph.D. degree in engineering sciences from Pontificia Universidad Católica de Chile. He is currently an Associate Professor of computer science with Universidad de los Andes, Chile. He has published in international journals. His research interests include computer-supported collaborative learning, human–computer interaction, and learning analytics. He is an Associate Editor of IEEE Transactions on Learning Technologies.

ISHARI AMARASINGHE received the Ph.D. degree in information and communication technologies from Universitat Pompeu Fabra, in 2020. She was a Postdoctoral Researcher with the TIDE Group, until 2022. She is currently with the National Education Laboratory AI, Radboud University. Her work emphasizes hybrid human–AI technologies to support teachers and learners, grounded in human-centered design. She has been published in Q1 and JCR-indexed journals, such as IEEE Transactions on Learning Technologies and *Journal of Computer-Assisted Learning*. Her research interests include AI and educational technologies.

GUSTAVO ZURITA received the Ph.D. degree in computer sciences from Pontificia Universidad Católica de Chile. He is currently a Full Professor with the Management Control and Information Systems Department, University of Chile. He has published extensively in journals, such as *Computers and Education* and *Journal of Computer Assisted Learning*. His research interests include collaborative information systems, CSCL, and technology-enhanced learning. He is an Editorial Board Member of the *Journal for STEM Education Research* and IEEE Transactions on Learning Technologies.

DAVINIA HERNANDEZ-LEO is currently a Full Professor of engineering and ICT with Universitat Pompeu Fabra, Barcelona, and leads the TIDE Research Group. Her research interests include human-centered computing, network applications, and learning sciences, particularly learning and collaboration technologies. She has been the Vice-President of the European Association for Technology-Enhanced Learning and an Associate Editor of IEEE Transactions on Learning Technologies. She also serves on the CSCL Committee of the International Society of the Learning Sciences and the Steering Committee of EC-TEL.

LUBNA HAKAMI received the bachelor's degree in computer science from Jazan University, in 2019, and the master's degree in intelligent interactive systems from Universitat Pompeu Fabra (UPF), Barcelona, in 2021, where she is currently pursuing the Ph.D. degree with the TIDE Research Group. Her research interests include educational technologies and learning analytics.

LUIS ROJAS received the Ph.D. degree in computer science from Universidad Autónoma de Madrid, in 2017. He is currently an Assistant Professor with Universidad San Sebastián, Santiago, Chile. His research interests include requirements engineering, empirical software engineering, and teaching technologies.