Flexible CSCL Orchestration Technology: Mechanisms for Elasticity and Dynamism in Pyramid Script Flows

Kalpani Manathunga and Davinia Hernández-Leo
kalpi.nisansala@gmail.com, davinia.hernandez-leo@upf.edu
ICT Department, Universitat Pompeu Fabra, Barcelona

Abstract: Flow patterns (e.g., Pyramid or Snowball) formulate good practices to script collaborative learning scenarios, which have been experimented in small-scale settings widely. Applying flow patterns on large-scale contexts present challenges to educators in terms of orchestration load. Orchestration technology can support educators to manage collaborative activities; yet existing technology do not address flexibility challenges like accommodating growing numbers of students or tolerating dynamic conditions in learning settings. We define elasticity and dynamism as two key elements in the flexibility of a script. Elasticity is related to the capacity of an orchestration technology to incorporate varying participant counts. Dynamism is the capacity to maintain a pedagogically meaningful script progression in presence of different individual behaviors. In this paper we propose flow creation and flow control mechanisms to address elasticity and dynamism in orchestration technology for Pyramid flows. These mechanisms, implemented in the PyramidApp tool, have been evaluated across four scenarios varying from small to large settings. The results show that rules enabling pyramid creation on-demand and the use of timers are useful to achieve elasticity and dynamism in the pyramid formation and progression in an automatic manner.

Introduction

Collaboration is a coordinated process by which individuals construct and maintain shared conceptions where knowledge is co-constructed socially (Roschelle & Teasley, 1995). In collaborative learning, situations are created in which particular forms of interactions among learners are expected to occur, leading to productive learning experiences. Computer-Supported Collaborative Learning (CSCL) contributes mechanisms and technologies supporting creation of such collaborative learning situations (Dillenbourg, Sanna, & Fischer, 2009; Roschelle & Teasley, 1995). Hence, CSCL environments need to be carefully designed and implemented incorporating interaction generation and regulation mechanisms. Moreover such CSCL contexts should scaffold productive interactions and/or to facilitate activity monitoring and intervening when required (Dillenbourg et al., 2009) since free collaboration does not necessarily result in fruitful learning (Dillenbourg & Jermann, 2010).

CSCL scripting means shaping up the way that collaborations are desired to happen with technology-mediation (Dillenbourg & Tchounikine, 2007) triggering specific types of interactions beneficial for learner cognition, while achieving educational objectives. Scripts define the activity sequence, group formation, phase changing, role allocation and rotation, resource distribution, mediating communication and coordination, constrain peer interactions in social and cognitive activities that would otherwise occur rarely or not at all (Dillenbourg & Tchounikine, 2007; Kobbe et al., 2007). In this notion, scripting is possible as *micro-scripts*, emphasizing on individual learner's actions with finer granularity and *macro-scripts*, defining interactions and regulations in coarse-grained activity flows (Dillenbourg & Tchounikine, 2007; Kobbe et al., 2007). Collaborative Learning Flow Patterns (CLFPs) are examples of macro-scripts reflecting best practices to orchestrate collaborative learning which are broadly accepted and repetitively utilized by practitioners (Hernández-Leo et al., 2006). Examples of CLFPs are Jigsaw, Pyramid or Snowball, Think-Pair-Share (TPS) and Thinking Aloud Pair Problem Solving (TAPPS). Each pattern is driven by its governing pedagogy that should not be modified during the design. For example, Pyramid pattern is considered as good practice to structure collaborative learning across multiple epistemic tasks and educational levels, fostering individual accountability and positive interdependence (Davis, 2002; Hernández-Leo et al., 2006). The pedagogy of this pattern is such that individuals study a given problem initially and propose a preliminary solution. Such solutions are discussed and compared to propose a shared solution in groups. This discussing and negotiation will repeat in growing group sizes (e.g. two groups in a level of the pyramid join as a single group in the next level) until the whole group reaches consensus to propose a common solution.

Practitioners are required to invest some effort to understand pattern definitions and types of constraints to design effective, meaningful scripts (Dillenbourg & Tchounikine, 2007; Hernández-Leo et al., 2006). Moreover, in its enactment with students, they need to orchestrate or manage in real time the script mechanics (Dillenbourg & Jermann, 2010). Flexible orchestration allows to adapt CSCL scripts in real-time with a degree of freedom to modify various orchestration aspects like group formation, role allocation and rotation (Dillenbourg & Jermann, 2010). Script modifiability is non-trivial, due to unexpected situations like learners not being present or leaving the activity in the middle spoiling on-going collaborations. As a result, orchestrating activity flows manually can be challenging for practitioners (Dillenbourg & Tchounikine, 2007; Sharples, 2013). Therefore, technology-mediated or semi-automated orchestration services are beneficial to real-time manage on-going activity flows. Previous work in the field had provided extensive knowledge in designing scripted collaborative learning flows effectively (Hernández-Leo et al., 2006; Pérez-Sanagustín, Burgos, Hernández-Leo, & Blat, 2011; Rodríguez-Triana, 2014), yet applicable mostly either at small-scale or co-located learning settings (Manathunga & Hernández-Leo, 2017). Different learning settings scaling, from small face-to-face classrooms to massive online learning communities, present varied script design and orchestration requirements. Diverse scales for collaborative learning orchestration raises a number of challenges, including adaptation challenges, tolerating unexpected conditions like drop-outs or late-joiners, needs for (re)designing scripts on-the-fly or managing the orchestration load (Sharples, 2013).

The main motivation of this research is to seek how already existing pattern-inspired scripts (in our case, Pyramid CLFP) can be enhanced to achieve flexible meaningful orchestration in order to be applied upon various learning scenarios. On the contrary to pre-defined, rigid scripts which can not be modified on-the-fly, flexible scripts allow practitioners to design and adapt in real-time with a freedom for modifications. Such modifications could be embedded during the design of the script or at the execution time as different mechanisms that do not violate the underlying pedagogical definition of the Pyramid pattern. Following sections of the article explain about those mechanisms and models introduced towards flexible CSCL orchestration, experimental settings, an analysis of the proposed mechanisms and a concluding discussion section.

Mechanisms for flexible orchestration

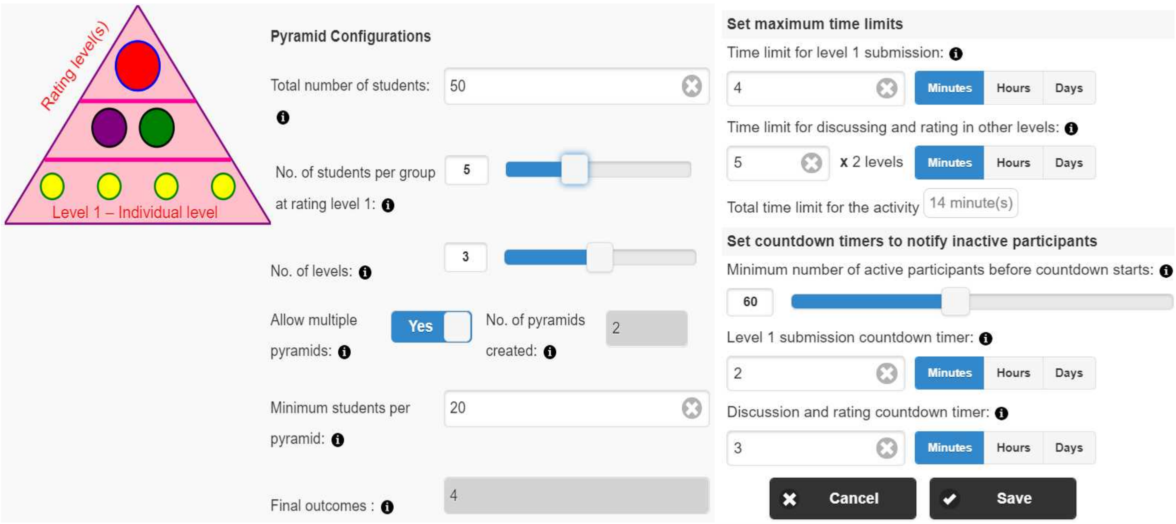
The journey towards orchestration technology that supports flexible scripted collaborative learning flow is approached from two key elements that we define as *elasticity* and *dynamism*. Elasticity is defined as the capacity of an orchestration technology to accommodate growing numbers of learners to collaborative learning activities without violating the underlying pedagogical rationale of the script. Dynamism is defined as the capacity of an orchestration technology to maintain a pedagogically meaningful script progression in presence of different individual behaviors. Hence, unexpected scenarios, e.g. unanticipated activity drop-outs (Dillenbourg & Tchounikine, 2007), would not harm on-going collaborations.

As indicated above, we study the particular case of the Pyramid collaborative learning flow pattern, as an interesting structure for macro-scripts that has a potential to fit well in scenarios with a varying number of participants. Based on an analysis of the pattern structure and the targeted objectives of elasticity and dynamism, we propose a set of mechanisms named as “Flow Creation” and “Flow Control” rules. As the name implies, flow creation mechanisms are suggested at the script initiation stage, when building the flow based on its learning design, whereas flow control rules are inferred during the script execution. To show its feasibility and to evaluate the mechanisms, we have integrated their implementation into the PyramidApp tool.

The case of the Pyramid Flow and the PyramidApp

Macro-scripts, such as those based on Jigsaw, Pyramid, Think-Pair-Share patterns, structure collaborations that potentially lead to fruitful learning (Hernández-Leo et al., 2006; Pérez-Sanagustín, Burgos, Hernández-Leo, & Blat, 2011; Rodríguez-Triana, 2014). Pyramid CLFP is structured in a way that individuals attend a given task and suggest an initial solution. Then they are assigned to small groups to discuss on the initially proposed options and agree upon a common option from the group which will be propagated to the next level(s) where much larger groups are formulated enriching collaborations and consensus reaching. At the global level, all participants agree upon one or few selected options that are reflected with the whole class. Pyramid pattern promotes individual accountability, peer interactions and positive interdependence among peers. The pattern can be applied to any subject matter, educational level and using (or not) different technologies (Hernández-Leo et al., 2006). An implementation of the pattern is reflected in PyramidApp, (Manathunga & Hernández-Leo, 2017), a web-based tool that implements an option submission space for participants to attempt the given task individually, a rating feature helping them to reach consensus with an integrated discussion space for negotiations and collaborations. PyramidApp has authoring features for the educators to create desired Pyramid pattern-inspired collaborative activities. It also provides a monitoring functionality for the educators to monitor on-going activities and to keep track of previous activities. The authoring process of PyramidApp implementation is shown in Figure 1 indicating different parameters, incorporating those related to the implementation of flow creation and control mechanisms. When enacting a PyramidApp activity, there are several phases as option submission, rating and discussion for further clarifications and to reach a consensus.

Next sections explain diverse parameters used for flow creation and control mechanisms and how those are embedded in PyramidApp.



(a) (b)

Figure 1. PyramidApp authoring application with (a) flow creation and (b) flow control mechanisms.

Script flow creation mechanisms

As stated earlier, flow creation mechanisms were introduced to achieve elasticity, i.e. to accommodate growing numbers of learners without affecting the underlying pedagogy. As illustrated in Figure 2, the implemented rationale is creating multiple pyramids on-demand automatically and allocating students to on-going pyramids as pyramids increase sizes till a maximum threshold.

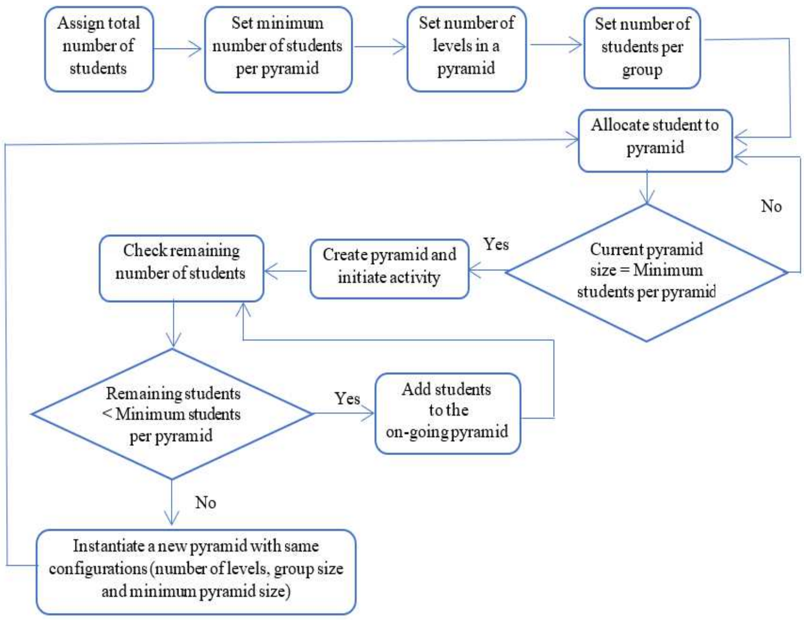


Figure 2. Flow creation mechanisms showing elastic pyramid initiation.

Key aspects introduced here are the pyramid capacity (i.e., both minimum and maximum numbers of students that can be accommodated in a single pyramid) and number of pyramids replicated to allocate further incoming students. Once a pyramid is configured by indicating number of levels in the pyramid, group size and the total number of activity participants, it grows till a maximum volume calculated using the equation: *Max_pyramid_size = {(Min_students_per_pyramid * 2) - 1}*. When the activity is initiated, pyramids are formulated and started using the value for minimum students per pyramid and then filled till the maximum size. During the authoring phase, if a practitioner allows multiple pyramid creation (see Figure 2), PyramidApp replicates the given design automatically to generate several pyramids to occupy the total amount of participants. When the minimum number of students per pyramid is given, the authoring tool will automatically calculate and display the possible group sizes (e.g., 2,3,4,5, etc..) along with the possible number of levels per pyramid. Based on these values and the total class size, the tool calculates the number of ultimate results from the activity after leveling through the pyramids.

Script flow control mechanisms

Once a Pyramid flow is activated, the flow needs to be controlled with more parameters, embedding dynamic behaviors to ensure a flexible progression during the execution time. In order to achieve dynamism with a smooth pyramid progression, we introduced several timers (see Figure 3) for different PyramidApp stages, to avoid the problem of different submission/rating times and drop-outs (inactive participants) causing the pyramid progression to freeze during option submission and rating stages. Submission timer and rating timer are the two main timers that define the maximum allowed time to complete those phases and their values can vary from minutes to days based on the activity being face-to-face or distance. To maintain a fluid dynamic flow, we use a satisfaction percentage (minimum number of active users completing a particular phase). Upon reaching the satisfaction percentage in a group, a countdown timer (*countdown timer < maximum allowed time per phase*) is activated until the maximum time allocated for that phase. The countdown timer notification is displayed in the interface for students to be informed. If all learners complete the task before any timer (either submission/rating timer or countdown timer) expiration, the group is promoted to the next level of the pyramid.

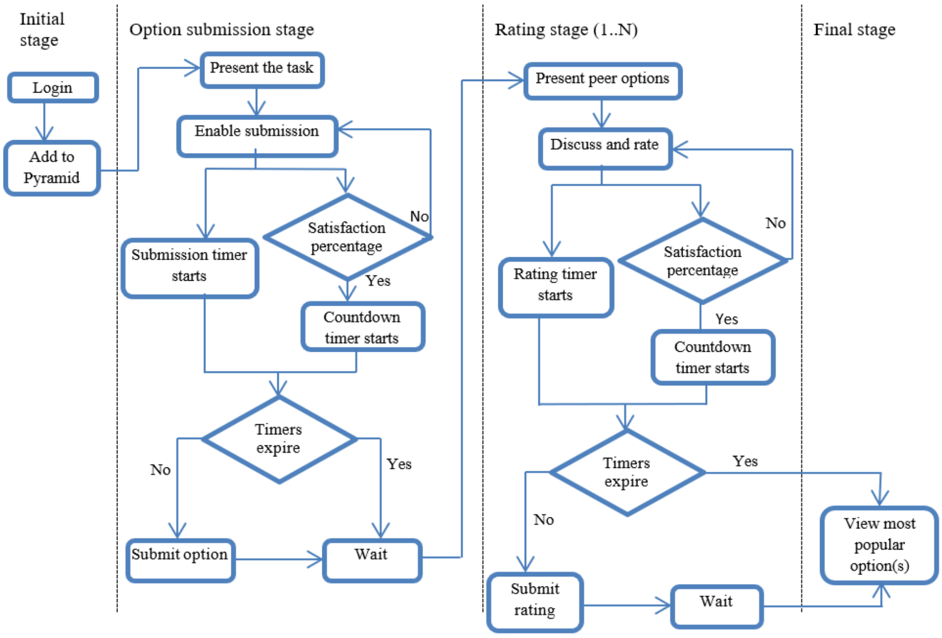


Figure 3. Flow control mechanisms showing dynamic flow progression.

Finally, we have integrated Pyramid flow activity awareness features that trigger information related to the activity status such as current pyramid level details, group members, countdown timer notifications and email notifications providing activity updates. Pyramid participants can see how many peers or groups who have not yet completed the current level during the waiting stage. Once the pyramid is finished, both practitioners and participants can view the highly rated option(s) resulted from the activity, which then could be -for example- further discussed and analyzed by the practitioner with the participants.

Evaluation

Flexibility of the flow orchestration can be construed by means of elasticity and dynamism. Our working hypothesis is that the proposed flow creation and control mechanisms embedded to Pyramid CLFP address flexibility successfully. Hence, the evaluation questions articulated and analyzed in this study are, do proposed mechanisms for flow creation address elasticity and do proposed flow control mechanisms address dynamism with meaningful orchestration? Meaningful orchestration is being pedagogically relevant during the script execution, which means that any novel mechanism introduced to the flow does not violate the essence of the pattern: e.g. pyramid group sizes to be preserved necessarily, every participant should have at least one peer for collaborations irrespective of activity drop-outs, late-comers are combined with on-going pyramids from the next possible activity phase and let them collaborate, propose default field values for the practitioners to create efficient Pyramid flows like possible number of pyramid levels or preferred group sizes and provide activity awareness measures for both participants and practitioners.

Experimental settings and data gathering

Several rounds of experiments were carried out in several sessions from two undergraduate courses (Introduction to Information and Communication Technologies (ITIC) offered in the first year and Network Protocols (NP) offered in the third year) of an Engineering School, taught by the same professor. PyramidApp activity authoring requirements varied based on different epistemic tasks and target groups (see Table 1). Some experiments were enacted during face-to-face classroom sessions whereas others were in distance mode, asynchronously. Some PyramidApp activities were administered in a Massive Open Online Course (MOOC) called Innovative Collaborative Learning with ICT (CLAT), a five-week MOOC ended in summer 2017, launched on Canvas platform. To evaluate the proposed flow creation and control mechanisms, diverse attributes related to the mechanisms were introduced at activity authoring and enactment phases in PyramidApp. Various data sources such as activity log files and questionnaires were used as data collection methods. Log files provided more accurate data to analyze time durations, activity participation and behavior during PyramidApp activities. Questionnaires provided practitioner's viewpoint as well as participants' perspective towards the activity. We used a mixed approach for the analysis triangulating both quantitative and qualitative data gathered from above data sources to answer the research question addressed (Twining, Heller, Nussbaum, & Tsai, 2017). Quantitative figures helped to evaluate how successfully proposed parameters could implement elasticity and dynamism whereas qualitative data provided better interpretation for the results acquired in each scenario.

Table 1: PyramidApp experiment settings

Activity Name	Activity Description	Activity type	Target group	Additional Details
ITIC (face-to-face class)	Three cases were given to read. Initially the class was divided into halves and gave two cases to be discussed using PyramidApp. Then the final case was given to the whole class and enacted one PyramidApp round.	Case study analysis, open-ended question answering and collaborative negotiation	First year undergraduates (N=31)	Students used only one smartphone per small group of two or three students at the first rating level.
ITIC (distance mode)	Watch a video (discussing ethical dilemmas in ICT) and indicate which of the 24 imperatives in the ACM code of ethics and Professional Conduct are related.	Case study analysis and collaborative negotiation	First year undergraduates (N=194)	Activity was enacted over a weekend as a homework before the next session using either smartphones or laptops individually.
NP (face-to-face class)	By observing the given TCP traffic, find some congestion control problems presented and explain your answer.	Problem solving and collaborative negotiation	Third year undergraduates (N=39)	Most students used smartphones individually. Activity was challenging. As expected by the educator, finally selected answer was incorrect.
CLAT MOOC (distance mode)	In your view, what are the main benefits of collaborative learning.	Reflections upon practices and collaborative discourse	Educators from secondary and higher education (N=617)	Heterogenous user groups with diverse expertise levels and experiences used their own devices.

Results and discussion

Scenarios stated in Table 1 have been analyzed indicating PyramidApp authoring configurations composed by the educators over resulted values during the activity enactment. Each case shows how the introduced flow creation and control mechanisms are used for pyramid formulation and how those achieved elasticity and dynamism. In terms of the meaningful orchestration achieved through the mechanisms, it was observed that there was no violation to the rules like maximum pyramid size or collaboration group sizes. Though some participants were dropped without completing all pyramid levels, participants had at least one peer to discuss during the collaboration stages and every group finally witnessed at least one solution irrespective of the number of initial submissions. Some participants joined the activity after the flow was initiated, yet pyramids occupied late-comers without interruptions. PyramidApp authoring features suggested default field values for the practitioners like possible number of levels in a pyramid or preferred group sizes and provided activity awareness measures such as current status of the activity, different groups, their members and responses for both participants and practitioners culminating a meaningful activity flow.

ITIC face-to-face class scenario

In this case, the educator designed three separate Pyramid flow designs (see Table 2) and each design resulted in a single pyramid after the execution also, as the class size was relatively small. Though the three pyramids were initiated with the minimum size, those had grown to occupy more students ensuring the elasticity of the designs. The satisfaction percentage provided for the activity was 60% and the two countdown timers had been activated 24 times overall. Irrespective of the countdown timer expirations students were able to still submit the initial option and the ratings without pyramids being frozen ensuring fluid, dynamic pyramid progression until the submission phase timer expires. All three pyramids had consumed around 10 minutes for the activity completion, as desired by the practitioner.

Table 2: Flow creation and control mechanisms – ITIC in-class activity

PyramidApp authoring parameters and values									
Pyramid ID	**No. of pyramids designed**	**Class size**	**Minimum students per pyramid**	**No. of levels**	**Group size (first level of rating)**	**Submit timer**	**Submit count down timer**	**Rating timer**	**Rating count down timer**
P1	1	15	15	2	3	4 mins	2 mins	4 mins	2 mins
P2	1	7	7	2	3	4 mins	2 mins	4 mins	2 mins
P3	1	7	7	2	3	4 mins	2 mins	4 mins	2 mins
After execution of the PyramidApp activity									
Pyramid ID	**No. of pyramids created**	**No. of logins**	**No. of options**	**Submit count down timer expiration**	**Submit timer expiration**	**No. of students rated in level 1**	**No. of students rated in level 2**	**Rating count down timer expiration (both levels)**	**Rating timer expiration (both levels)**
P1	1	18	15	3	3	16	15	13	1
P2	1	12	6	3	6	10	10	3	2
P3	1	10	4	2	6	8	8	0	2

NP face-to-face class scenario

This is also a face-to-face classroom scenario in which the educator designed only one pyramid, but the flow creation rules replicated the design and created two pyramids on-demand to accommodate all activity participants (see Table 3). Though the educator had expected 40 students for the activity, only 32 were present on that day and that miscalculation had no effect on the PyramidApp enactment due to the elasticity mechanisms applied by the application. Here also the satisfaction percentage was set to 60% and 15 submissions had been done after the countdown timer was initiated and 19 students had rated after seeing the rating countdown timer. The fact that every student not rating both levels had no effect in the pyramid progression due to the dynamism mechanisms proposed which provided flexible orchestration till the final level of pyramids. Both pyramids consumed around 9 minutes to complete all the levels including discussions. The task was authored with more time for submission and rating deliberately by the educator as the task was very challenging and wanted students to fail, to establish the conditions of a motivated and rich discussion in the classroom about why they failed, and which would be the right answer. Students enjoyed the activity irrespective of being failed to answer.

Table 3: Flow creation and control mechanisms –NP in-class activity

PyramidApp authoring parameters and values									
Pyramid ID	**No. of pyramids designed**	**Class size**	**Minimum students per pyramid**	**No. of levels**	**Group size (first level of rating)**	**Submit timer**	**Submit count down timer**	**Rating timer**	**Rating count down timer**
P1	1	40	16	2	6	5 mins	2 mins	3 mins	1 min
After execution of the PyramidApp activity									
Pyramid ID	**No. of pyramids created**	**No. of logins**	**No. of options**	**Submit count down timer expiration**	**Submit timer expiration**	**No. of students rated in level 1**	**No. of students rated in level 2**	**Rating count down timer expiration (both levels)**	**Rating timer expiration (both levels)**
P1	2	16	11	9	3	12	9	11	2
P2		16	10	6	4	12	14	8	2

ITIC distance scenario

This distance mode of PyramidApp activity was enacted as a homework task over a weekend in a relatively large class (n=194). As given in Table 4, the educator designed a distance mode pyramid activity and assigned 16 as the minimum pyramid size. The activity resulted in 6 pyramids that have grown till the maximum volume successfully complying with proposed elasticity mechanisms. Longer timers were assigned for submission and rating since this activity extended over two days in the distance mode. As for the flow control mechanisms, this version of distance mode PyramidApp did not implement the countdown timer based on the satisfaction percentage. Instead, it showed the remaining timer notification for each phase from the beginning of the activity and notified students via email notifications. Hence, the analysis considers only the submission and rating timer expiration. Still, that did not affect the flow of the pyramids. In all pyramids, number of students rated and collaborated in the second level is lesser than the first level. Around 48% could not complete the submission phase (in P2, P3, P4) because they were added to the on-going pyramids as they had accessed the activity late. Yet, these students were given chance to participate and present opinions in the rating stages assuring meaningful orchestration along the Pyramid flow, rather letting them to be idle till the next activity is available. If students login after pyramid creation timestamp (i.e., submission timer expired), they are straightaway added to the next available on-going pyramid, allowing them to collaborate in rating and discussion stages.

Table 4: Flow creation and control mechanisms – ITIC distance activity

PyramidApp authoring parameters and values							
Pyramid ID	**No. of pyramids designed**	**Class size**	**Minimum students per pyramid**	**No. of levels**	**Group size (first level of rating)**	**Submit timer**	**Rating timer**
P1	1	120	16	2	5	18 hrs	12 hrs
After execution of the PyramidApp activity							
Pyramid ID	**No. of pyramids created**	**No. of logins**	**No. of options**	**Submit timer expiration**	**No. of students rated in level 1**	**No. of students rated in level 2**	**Rating timer expiration (both levels)**
P1	6	31	10	21	23	16	7
P2		31	16	15	27	19	1
P3		31	16	15	23	16	4
P4		31	16	15	27	18	2
P5		31	25	6	21	18	4
P6		31	20	11	13	11	14

CLAT MOOC scenario

In the MOOC scenario, the educator designed a Pyramid collaborative activity that extended to three pyramids at the end of the course (see Table 5). Only P1 had the minimum number of participants (four students) whereas P2 and P3 had occupied more, preserving the elasticity properties introduced. Here also the distance version of the PyramidApp including timers only for submission and rating along with the email notifications was used. Here also the timers were quite longer than in a usual face-to-face class. Around 67% from the participants had submitted their options and 60% had rated at least one level of the activity. Though, the activity participation is not equal among all students, the pyramids fluidly finished with no freezing in any branch.

Table 5: Flow creation and control mechanisms – CLAT MOOC distance activity

PyramidApp authoring parameters and values							
Pyramid ID	**No. of pyramids designed**	**Class size**	**Minimum students per pyramid**	**No. of levels**	**Group size (first level of rating)**	**Submit timer**	**Rating timer**
P1	1	4	4	2	2	2 hrs	2 hrs
After execution of the PyramidApp activity							
Pyramid ID	**No. of pyramids created**	**No. of logins**	**No. of options**	**Submit timer expiration**	**No. of students rated in level 1**	**No. of students rated in level 2**	**Rating timer expiration (both levels)**
P1	3	4	3	1	1	2	2
P2		6	3	3	2	3	3
P3		5	4	1	4	1	1

Conclusion

This paper offers a technology-oriented contribution to the orchestration technology research line within CSCL, aiming at facilitating the application of collaborative learning scripts from small to large settings in a flexible way. We have defined script flexibility in terms of *elasticity* and *dynamism* of the collaborative learning activity flow. Moreover, we have proposed mechanisms to address them for the case of scripts structured according to the Pyramid or Snowball collaborative learning flow pattern. A set of flow creation mechanisms (e.g. allocation of participants to pyramid groups considering ranges in desired group sizes) has been defined to enable the elastic incorporation of a varying number of participants to a Pyramid flow in an automatic manner. Flow control mechanisms (including timers and satisfaction parameters) are proposed to maintain a fluid, dynamic pyramid flow execution. These mechanisms have been implemented in the PyramidApp tool. Validation of PyramidApp across different educational settings showed that the flow creation and control mechanisms introduced to the Pyramid flow achieved elasticity and dynamism. Dynamically growing pyramids and replication of pyramids on-demand accommodated late-comers. Dropping out from current activity did not harm the pyramid progression. The mechanisms also led to ensure that orchestration (or real time script management) aspects were meaningful, i.e. in alignment with the pedagogical structural elements of the Pyramid pattern. For example, even if every participant did not submit an initial option (e.g. if they arrive late or leave) for the given task in above cases, the mechanisms ensured every group had at least two participants and one option to discuss in the first rating level and the flow was not interrupted. The same strategy has been used at other rating stages too, assuring that the pyramid can level-up. Future work should study to what extent these mechanisms can be extrapolated to other script families (e.g. those based on the Jigsaw flow pattern).

The results also suggest that human and intelligent agent interventions could further improve the utility of the proposed mechanisms. Usage of the in-built discussion board of the PyramidApp was not satisfactory across experiments. Email notifications used in the distance mode, notifying activity status did not catch sufficient participant attention. Hence, in future discussion prompts, cues or agent technologies like learning companions can be introduced to study how they could aid in promoting higher engagement in discussions. Future improvements like allowing small groups to modify or submit new options after collaborating in the rating levels and enhancing the notification system could be beneficial. Moreover, an extended version of the monitoring dashboard that would enable human intervention by the educator could improve the options for orchestration support, e.g. by facilitating the participation of the educator in discussions when especially needed (e.g. alerted by the system) or to modify timers to regulate the activity progression if required.

References

Davis, W.A. (2002). A comparison of pyramids versus brainstorming in a problem-based learning environment. In *Focusing on the student: Proceedings of 11th Annual Teaching Learning Forum*. Perth: ECU.

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

Dillenbourg, P., Sanna, J., & Fischer, F. (2009). The evolution of research on computer-supported collaborative learning. *Technology-Enhanced Learning*, 3–19. Netherlands: Springer.

Dillenbourg, P., & Jermann, P. (2010). Technology for Classroom Orchestration. In M.S. Khine & I.M. Saleh (Eds.), *New Science of Learning: Cognition, Computers and Collaboration in Education,* 45–52. New York: Springer.

Hernández-Leo, D., Villasclaras-Fernández, E. D., Asensio-Pérez, J. I., Dimitriadis, Y., Jorrín-Abellán, I. M., Ruiz-Requies, I., & Rubia-Avi, B. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology and Society, 9*(1), 58–71.

Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P., & Fischer, F. (2007). Specifying computer-supported collaboration scripts. *International Journal of Computer-Supported Collaborative Learning, 2*(2–3), 211–224.

Manathunga, K., Hernández-Leo, D. (2017). Authoring and enactment of mobile pyramid-based collaborative learning activities. *British Journal of Educational Technology, 49*(2), 262-275.

Pérez-Sanagustín, M., Burgos, J., Hernández-Leo, D., & Blat, J. (2011). CLFP intrinsic constraints-based group management of blended learning situations. *Studies in Computational Intelligence*, *350*, 115–133.

Rodríguez-Triana, M. J. (2014). *Linking scripting and monitoring in blended CSCL Linking scripting & monitoring support in blended CSCL scenarios*. Unpublished Doctoral Dissertation, University of Valladolid, Valladolid, Spain.

Roschelle, J., & Teasley, S. D. (1995). The Construction of Shared Knowledge in Collaborative Problem Solving. In C. O'Malley (Eds.), *Computer Supported Collaborative Learning* (pp. 66-97). Heidelberg: Springer.

Sharples, M. (2013). Shared orchestration within and beyond the classroom. *Computers and Education*, *69*, 504–506.

Twining, P., Heller, R., Nussbaum, M., & Tsai, C. (2017). Some guidance on conducting and reporting qualitative studies. *Computers and Education, 106*, A1–A19.

Acknowledgements

DHL is a Serra Húnter fellow. This work has been partially funded by "la Caixa Foundation" (CoT project, 100010434) and the National Research Agency of the Spanish Ministry of Science, Innovations and Universities MDM-2015-0502, TIN2014-53199-C3-3-R, TIN2017-85179-C3-3-R.