

Computational Challenges of Co-Creation in Collaborative Music Live Coding: An Outline

Anna Xambó^{1,2}, Gerard Roma^{2,1}, Pratik Shah³, Jason Freeman¹ and Brian Magerko²

¹Center for Music Technology, Georgia Institute of Technology

²Digital Media Program, Georgia Institute of Technology

³School of Interactive Computing, Georgia Institute of Technology

North Ave NW, Atlanta, Georgia, USA, 30332

anna.xambo,gerard.roma,pratikshah,jason.freeman,magerko@gatech.edu

Abstract

Co-creation between a human agent (HA) and a virtual agent (VA) is an approach to collaboration that has been explored in different creative domains, particularly in computer music. With a few exceptions, there is little research on the use of virtual agents in collaborative music live coding (CMLC), a network music improvisational practice. This paper considers the benefits of CMLC both in education and in performance involving human agents, with or without virtual agents. We reflect on our previous work on, and lessons learned from, two studies of collaboration and live coding using EarSketch, an educational online platform for learning music via code based on audio clips. We speculate future scenarios, in particular, we envision a virtual agent that can help students to improve their programming and musical skills, and can help musicians to exploit computational creativity applied to music.

Introduction

Music live coding is an improvisational music practice that consists of a programmer/musician, also known as a live coder, who writes and manipulates code in real time (Collins et al. 2007). Collaborative music live coding (CMLC) typically involves a group of at least two networked live coders, who can be either co-located in the same space, distributed in different spaces, or both (Barbosa 2006). CMLC is a promising approach to both music and computer science education, as well as music performance, because it can promote peer-learning in the former, and an egalitarian approach to collaborative improvisation in the latter.

Live coding and collaboration is an emerging field of research, as shown in the Dagstuhl Seminar on *Collaboration and Learning through Live Coding* held in 2013 (Blackwell et al. 2014). However, there are still a number of open questions on how to computationally best support this practice and how to theoretically and methodologically understand the nature of collaboration between the live coders. We discuss these questions, which can potentially inform the design of new tools.

We have investigated CMLC using EarSketch (Magerko et al. 2016), a browser-based programming environment for making music that is based on audio clips. In particular, we have explored two strategies for editing a shared script:

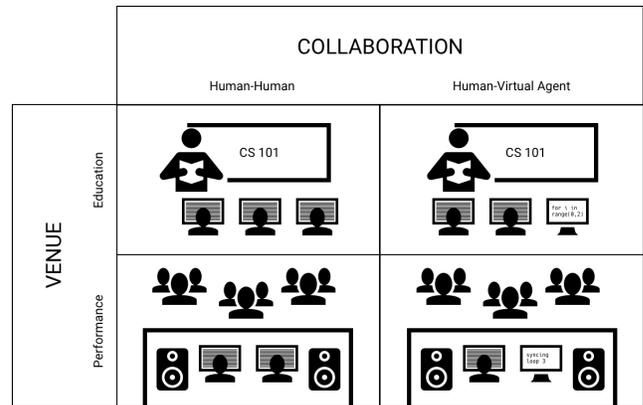


Figure 1: Matrix of CMLC with *human agent vs. human agent (HA-HA) in music education, human agent vs. virtual agent (HA-VA) in music education, HA-HA in music performance, and HA-VA in music performance.*

(1) simultaneous editing, as in Google Docs (Xambó et al. 2016), and (2) turn-taking, as in pair programming, in combination with the use of a chat window for mutual communication (Xambó et al. 2017). In these studies, we have used a social approach to understand the nature of collaboration, i.e., analysis of the behavior from the text in the chat window or the text in the code editor, analysis of informal interviews with teachers about their observations in class, analysis of music performances by live coders, analysis of screencasts of live coders, and so on.

In this position paper, we explore potential synergies and identify novel insights through the lens of co-creativity as applied to CMLC. This field looks into cognitive aspects of collaborative content creation, such as *participatory sense-making* (De Jaegher and Di Paolo 2007). CMLC can be seen as a conversation between at least two people, thus participatory sense-making can be informative about the collaboration from a conversational perspective, e.g., what drives the live coders to take decisions or what are the kind of problems that the live coders solve during their collaboration. From this angle, we expect to understand how to computationally best support collaborative music live coding. Inspired by our previous work on CMLC, we map this emergent space into

four cases focusing on the EarSketch environment. The four cases are based on two scenarios, music / computer science (CS) education and music performance, and on two types of groups, a group with human live coders only, and a mixed group with a virtual agent (see Figure 1). Expanding the notion of collaboration between not only human agents is helpful for identifying new possibilities of collaboration in both music education and performance. We speculate that a virtual agent can be useful in CMLC because it can help the student to improve programming and musical skills in education, as well as provide the live coder with computational creative solutions in rehearsal and performance.

Background

Machine musicianship applies artificial intelligence concepts and techniques to computer music systems (Rowe 2001). Some notable examples are the Continuator (Pachet 2003) and Shimon (Hoffman, Guy and Weinberg 2010). In particular, we are interested in research on using bots in real-time music collaboration. Bots have been used in collaborative improvisation using laptops. For example, LOLbot is a virtual agent that plays with human performers (Subramanian, Freeman, and McCoid 2012), and Autocode is an autonomous live-coding virtual agent in *ixi lang* (Magnusson 2011).

Typically, the bot in collaborative improvisation using laptops behaves like a ‘follower’ or ‘learner’ of the human actions, e.g., music generation from what performers are doing based on pattern matching, where the human performer can define the level of contrast with a slider (Subramanian, Freeman, and McCoid 2012). There exist also instant messaging bots that support collaboration (Chan, Hill, and Yardi 2005). In this paper, we are interested in the combination of these two roles: the live coder bot and the instant messaging bot.

With respect to human-only CMLC, we investigated pair programming using EarSketch thinking on a music performance scenario, and found out that turn-taking pair programming should be combined with simultaneous editing by both live coders to make the performance more fast-paced, and that trio live coding was a more dynamic and interesting configuration (e.g., more roles were explored) than duo live coding (Xambó et al. 2017).

Regarding educational settings, there is a significant amount of research on the pedagogical benefits of live coding in education, as evidenced in the special issue on “Live Coding for Music Education”, published in the *Journal of Music, Technology & Education* (Brown 2016). This issue included an analysis of the challenges and opportunities of teaching live coding in the classroom using EarSketch (Freeman and Magerko 2016). Our work on CMLC in education is a follow up of this literature, particularly looking at different configurations of human-centered CMLC in the classroom, from pair programming to multiple live coding (Xambó et al. 2016).

Designing and evaluating CMLC systems that involve VAs is an open question that can benefit from existing theoretical frameworks. In cognitive science, cognitive theories have been used to design and evaluate co-creative systems,

for example *enaction*, *co-creativity* and *participatory sense-making* (Davis 2015; Davis et al. 2015). Participatory sense-making combines social cognition and enaction theories for understanding the interaction between individuals during a collaborative activity (De Jaegher and Di Paolo 2007; Fuchs and De Jaegher 2009). A successful instance of co-creation between a human agent and a virtual agent is applied to the domain of creative drawing (Davis et al. 2016).

HA-HA in Education

Pair programming is a common practice in CS education borrowed from industry. In pair programming, two programmers frequently alternate between the roles of a *driver* (i.e., writing the code) and a *navigator* (i.e., giving advice) when solving a problem together. Pair programming in software development settings (e.g., helping each other to debug, reviewing code, or writing code to solve a clearly defined problem) contrasts with pair programming in musical settings (i.e., a musical discourse is needed about what the goal or aesthetics actually are, and group discussion to resolve differences of opinion about the direction the music might take (Dobson and Littleton 2016)). In pair programming applied to CMLC, students can explore different tasks related to either code or music while working together. For example, the navigator(s) focuses on understanding an error, discovering sounds, or planning new parts of the song, while the driver writes and executes the code (Xambó et al. 2017).

In the classroom, it is an open question as to what are the implications in learning and how the system should be designed to effectively support both coding and music composition from a collaborative angle. We here focus on live coding, which combines both activities in real-time collaboration: as already discussed in our prior papers, problem solving can be addressed socially, and the platform should support this social conversation (i.e., communication tools, script sharing, attribution and ownership mechanisms, and so on). In designing these digital social tools, and observing their use, we can understand better the nature of collaboration in each of the two domains, CS and music.

A shared script that is displayed in a projection screen, with multiple users simultaneously editing, can be an interesting interactive experience for the students. However, if there is no regulation of turns, it can also become chaotic. The option of division of labor into smaller teams seems convenient if working within a large team. The use of individual spaces vs. a shared space seems useful, yet these spaces should be defined. For example, each live coder can have an individual space to work on smaller modules like functions or selection of sounds before using them in the shared space. However, in order to offer a truly mutual collaboration, perhaps everyone should be able to access all the individual modules and modify their content, ideally at any time, but with some level of control as well.

HA-VA in Education

Adding a virtual agent in CMLC applied to education seems to add pedagogical value to the live coding experience in addition to the available tools. Potentially, the VA should

get “activated” or “directed” towards tasks. For example, an instant messaging bot can recommend sounds to the learner if in the mode of pair programming. If the student stops exploring the computational possibilities of the platform (Helms et al. 2016), the bot can recommend new code examples, e.g., design patterns or more complex structures. Live coder bots can also be helpful as code generators that help when the student wants to avoid mechanical and repetitive code writing. Overall, a virtual agent can help the learner to develop both musical and computational skills. For example, machine learning algorithms can be used to learn from the same student’s and other peers’ code.

It is critical to avoid designing a virtual agent that interferes or disrupts the flow of collaboration rather than supporting it. Similarly, a chatbot style agent should be different from intelligent components of the larger UI: a sound recommendation system can be useful for initial queries or context-aware recommendations, but then a chatbot can recommend from a musical human-like perspective, e.g., adding contrast to the overall outcome, or a little bit of randomization and surprise based on real-time conversations with the student. In addition, the humanization of the intelligence is important to work more closely with the virtual agent, e.g., provide more simultaneous mutual feedback.

HA-HA in Performance

A turn-taking model can sometimes be too time consuming for a performance setting. A multiple-editor format can be faster yet chaotic and misleading for the audience, unless the interface projected in the screen is self-explanatory (e.g., one color for each user, similar to Google Docs-like collaborative editing). Keeping a balance between the clarity but slow-pace of pair programming, and the lively but unorganized simultaneous editing, seems desirable. As already discussed in HA-HA in education, adding individual spaces vs. a shared space seems useful, but individual spaces need to be defined, i.e., to what extent individual spaces should be private between performers, the audience, or both, during shared-script collaboration. There is room for research on finding the right balance and defining what individual and share spaces mean. There is also the complexity of finding appropriate “compile points”, where simultaneous contributions are all in syntax-ready state.

HA-VA in Performance

Adding a virtual agent in CMLC applied to music performance can also improve the live coding experience. Working with a virtual agent can help humans to improve their music coding skills as well as their group improvisational skills (Subramanian, Freeman, and McCoid 2012). Beyond the performance space, a virtual agent seems also helpful during rehearsal time, e.g., when other live coders are not available, or to train a machine learning algorithm with examples from a human live coder. In performance research, a VA can be used as a tool to understand how human live coders improvise together by modeling their behaviors (Subramanian, Freeman, and McCoid 2012).

A VA live coder can provide new computational approaches to music making, e.g., pattern generation (Subramanian, Freeman, and McCoid 2012). More broadly, the use of VAs in music performance has been shown to provide new forms of human-computer interaction, e.g., musical robots (Bretan and Weinberg 2016) or musical avatars (Collins 2011). The VA can be designed to contribute to the performance by having a certain ‘unique’ character as opposed to using intelligent but non-humanoid UI tools, such as just a real-time recommender system.

The benefits of using a VA musician in CMLC are numerous. Live coding can be time consuming in performance situations, thus a computational-led companion can speed up the process (Collins 2011). A ‘follower’ agent model can be helpful: it learns from the coder and types faster. However, like in the case of education, we should consider how to make it more unpredictable. An option is to combine the call-response metaphor with simultaneous interactions (similar to Google Docs-like collaboration), where the live coder bot has its own style besides learning from the human live coder. The use of a VA also raises potential *machine ethics* (Collins 2011) that should be considered when designing future generations of VAs for CMLC.

Discussion

How to design and evaluate these systems? This section discusses tradeoffs and future directions based on the presented four use cases. In addition, we discuss the emerging use case of VA-VA interaction.

Design and Evaluation

An initial step for evaluating the above four use cases is using an existing theoretical framework applied to similar situations. A suitable model is the *computer colleagues* paradigm, proposed in Davis et al. (2015). This paradigm is described as a highly demanding level of collaboration because it requires *perception-action feedback loop* with the environment. The scalability of this model is still an open question, in particular whether and how it can be applied to multiple individuals.

In live coding, screens are supposed to be shown,¹ and so the “algorithmic thinking” of the live coders is exposed to the audience. In this sense, CMLC provides a unique opportunity to expose participatory sense-making. The notion *co-regulation* of interaction between agents has been highlighted in the literature as a feature of co-creation (Davis et al. 2016). The turn-taking timings in CMLC can provide cues to model this co-regulation. The study of simultaneous actions seems more challenging.

Human-human interaction can inform human-robot interaction, e.g., Thomaz and Chao (2011) define a turn-taking framework and identify that information flow is essential for a successful collaboration. Our existing observations of HA-HA interactions will inform a first iteration of HA-VA interaction in EarSketch, with applications to both education and performance. Computational creativity is a broad

¹http://toplap.org/wiki/Read_me_paper (accessed June 8, 2017)

topic. A general definition of creativity is “the ability to generate novel, and valuable, ideas” (Boden 2009, p. 24). What is creativity and what is the role expected from a computer as a partner in order to be creative has been discussed by Lubart (2005). However, assessing whether a computer is creative is in itself a philosophical inquiry (Boden 2009). Open questions include how to measure the contribution of the agent, and how would a novel idea be. Approaches from computational novelty literature (Tsai 2010) can be useful for this purpose. As an example, distance metrics between digital media files have been used to measure human creativity in online databases (Roma et al. 2012).

VA-VA interaction

An interesting topic that stems from the four use cases presented in this paper is to also include VA-VA collaboration applied to both education and performance. A number of research questions emerge from multiple agent collaboration, such as whether multiple agents can collaborate among themselves, how feasible it is, what would be the computational cost, to what extent there should be supervision and how often, and whether it would be interesting for education and performance. Collaboration between live-coding virtual agents can be seen as a particular case of multi-agent systems for music composition and performance, which has been widely researched (Miranda 2011).

Conclusion

In this paper, we have mapped the space of CMLC so that it includes virtual agents in addition to humans. We presented the challenges and benefits in both educational and performance settings. We reflected on previous work with EarSketch and live coding. We plan to continue this research by exploring a virtual agent companion that learns from human live coders using machine learning algorithms and that goes beyond the approach of following live coder actions (also known as the call-response strategy). In order to embody the humanoid metaphor, we envision that the virtual agent should be able to act both as a live coder and a chatting peer. Further investigation on the suitability of the co-regulation model applied to live coding seems promising. From this workshop, we expect to discuss the above ideas with experts in the field of co-creativity and identify more clearly the next steps of this research.

Acknowledgments

The EarSketch project receives funding from the National Science Foundation (CNS #1138469, DRL #1417835, DUE #1504293, and DRL #1612644), the Scott Hudgens Family Foundation, the Arthur M. Blank Family Foundation, and the Google Inc. Fund of Tides Foundation.

References

Barbosa, Á. 2006. *Displaced Soundscapes: CSCW for Music Applications*. Ph.D. Dissertation, Universitat Pompeu Fabra.

Blackwell, A.; McLean, A.; Noble, J.; and Rohrhuber, J. 2014. Collaboration and Learning through Live Coding (Dagstuhl Seminar 13382). *Dagstuhl Reports* 3(9).

Boden, M. A. 2009. Computer Models of Creativity. *AI Magazine* 30(3):23.

Bretan, M., and Weinberg, G. 2016. A Survey of Robotic Musicianship. *Communications of the ACM* 59(5):100–109.

Brown, A. R. 2016. Editorial. *Journal of Music, Technology and Education* 9(1):3–4.

Chan, S.; Hill, B.; and Yardi, S. 2005. Instant Messaging Bots: Accountability and Peripheral Participation for Textual User Interfaces. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP '05)*, 113–115.

Collins, N.; McLean, A.; Rohrhuber, J.; and Ward, A. 2007. Live Coding Techniques for Laptop Performance. *Organised Sound* 8(3):321–330.

Collins, N. 2011. Trading Faures: Virtual Musicians and Machine Ethics. *Leonardo Music Journal* 21:35–39.

Davis, N.; Hsiao, C.-P.; Popova, Y.; and Magerko, B. 2015. An Enactive Model of Creativity for Computational Collaboration and Co-Creation. In *Creativity in the Digital Age*. Springer. 109–133.

Davis, N.; Hsiao, C.-P.; Yashraj Singh, K.; Li, L.; and Magerko, B. 2016. Empirically Studying Participatory Sense-Making in Abstract Drawing with a Co-Creative Cognitive Agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16)*, 196–207.

Davis, N. 2015. An Enactive Approach to Facilitate Interactive Machine Learning for Co-Creative Agents. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (C&C '15)*, 345–346.

De Jaegher, H., and Di Paolo, E. 2007. Participatory Sense-Making. *Phenomenology and the Cognitive Sciences* 6(4):485–507.

Dobson, E., and Littleton, K. 2016. Digital Technologies and the Mediation of Undergraduate Students' Collaborative Music Compositional Practices. *Learning, Media and Technology* 41(2):330–350.

Freeman, J., and Magerko, B. 2016. Iterative Composition, Coding and Pedagogy: A Case Study in Live Coding with EarSketch. *Journal of Music Teacher Education* 9(1):37–54.

Fuchs, T., and De Jaegher, H. 2009. Enactive Intersubjectivity: Participatory Sense-Making and Mutual Incorporation. *Phenomenology and the Cognitive Sciences* 8(4):465–486.

Helms, M.; Moore, R.; Edwards, D.; and Freeman, J. 2016. STEAM-Based Interventions: Why Student Engagement is Only Part of the Story. In *Proceedings of IEEE Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT 2016)*.

Hoffman, Guy and Weinberg, G. 2010. Shimon: An Interactive Improvisational Robotic Marimba Player. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, 3097–3102.

Lubart, T. 2005. How Can Computers Be Partners in the Creative Process: Classification and Commentary on the Special Issue. *International Journal of Human-Computer Studies* 63(4):365–369.

- Magerko, B.; Freeman, J.; McKlin, T.; Reilly, M.; Livingston, E.; McCoid, S.; and Crews-Brown, A. 2016. EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education. *ACM Transactions on Computing Education* 16(4):14:1–14:25.
- Magnusson, T. 2011. ixi lang: a Supercollider Parasite for Live Coding. In *Proceedings of International Computer Music Conference 2011 (ICMC '11)*, 503–506.
- Miranda, E. R. 2011. *A-Life for Music: Music and Computer Models of Living Systems*. Middleton, WI: A-R Editions, Inc.
- Pachet, F. 2003. The Continuator: Musical Interaction with Style. *Journal of New Music Research* 32(3):333–341.
- Roma, G.; Herrera, P.; Zanin, M.; Toral, S. L.; Font, F.; and Serra, X. 2012. Small World Networks and Creativity in Audio Clip Sharing. *International Journal of Social Network Mining* 1(1):112–127.
- Rowe, R. 2001. *Machine Musicianship*. Cambridge, MA: MIT Press.
- Subramanian, S.; Freeman, J.; and McCoid, S. 2012. LOLbot: Machine Musicianship in Laptop Ensembles. In *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 421–424.
- Thomaz, A. L., and Chao, C. 2011. Turn-Taking Based on Information Flow for Fluent Human-Robot Interaction. *AI Magazine* 32(4):53–63.
- Tsai, F. S. 2010. Review of Techniques for Intelligent Novelty Mining. *Information Technology Journal* 9(6):1255–1261.
- Xambó, A.; Freeman, J.; Magerko, B.; and Shah, P. 2016. Challenges and New Directions for Collaborative Live Coding in the Classroom. In *International Conference of Live Interfaces (ICLI 2016)*.
- Xambó, A.; Shah, P.; Roma, G.; Freeman, J.; and Magerko, B. 2017. Turn-Taking and Chatting in Collaborative Music Live Coding. In *Proceedings of the 12th Audio Mostly Conference (AM '17)*.