Emergent Global Cueing of Local Activity: Covering in Music

Nick V. Flor
Graduate School of Industrial Admin.
Carnegie Mellon University
Pittsburgh, PA 15213
flor+@cmu.edu

Paul P. Maglio
IBM Almaden Research
650 Harry Road, NWED-B2
San Jose, CA 95120
pmaglio@almaden.ibm.com

Abstract

We explore the function of non-institutionally provided external resources that people working in groups rely on to sequence and synchronize *pre-learned* actions. Through *longitudinal observations* of a music group learning to cover (i.e., play) other groups' songs, and an analysis of the computational requirements for song covering, we show how simultaneous group activity creates — as a *side-effect* — emergent structures that can sequence individual behavior. In short, local activity creates *global cues* that can then direct local action. We argue that this is not specific to music groups, but that global cues emerge in *any* simultaneous group activity in which individuals have access to others' on-going performances. We discuss the importance of our results for distributed cognition research, learning in group contexts, and for the design of technological support for collaborative learning.

Introduction

Between routine performance and problem solving lies a class of activities that require a person to execute a set of pre-learned actions in a specific sequence. A musician playing a piece of sheet music, a person assembling a new bike, or a pilot doing a pre-flight checklist, are all examples of such activities. The individual or group engaged in these kinds of tasks presumably knows how to perform each step, but not necessarily the order of steps. Institution-provided external representations, such as a checklist, music sheet, or recipe are important resources that help people sequence their operations. But as highly adaptive agents, individuals are bound to discover resources and practices which vary from institutionally specified ones. For instance, Kirsh (1995) describes some ways in which cooks spatially layout their ingredients to keep track of what they have and have not done. Hutchins (1995b) and Norman (1992) describe numerous examples of non-institutional resource usage among aviators. Within any given domain, we would expect to find such uses of external resources. But are there *domain-invariant* external resources that individuals can rely on? We sought an answer to this question by studying the activities of a rock band learning to cover[1] popular songs. We found not only a variety of non-institutional resources used but also domain-invariant ones. Specifically, covering songs in a group produces emergent sequence cues; in performing local actions, group members produce as a side-effect sequence cues for other group members. Thus, we have a dialectical process in which global cues emerge from local activity, and in turn feed back to help sequence local activity. This finding has both theoretical implications for distributed cognition research and practical implications for learning in group contexts.

Our report is structured as follows. First, we provide background information, describing relevant cognitive research, as well as a summary of our methods and the music group studied. Our analysis then begins with a description of the general composition of a song. We use this description to analyze the cognitive demands song covering places on musicians who use no external aids. This analysis leads to several conjectures on how processes and representations might be distributed in the musician's environment to simplify the musician's cognitive task, and we show that in fact these conjectures hold true in the performance data we gathered. Finally, we discuss implications of this work.

Background

We position our study against two research areas: music cognition and distributed cognition. Music cognition research has primarily focused on individual musicians and models of individual music learning, improvisation, and performance. From this work we know that information-processing formalisms are sufficient to characterize the melodic, rhythmic, and

[1] Among musicians, the term "covering" refers to the activity of playing another band's song.

harmonic dimensions of music (Simon and Sumner, 1968). We also know that musicians and music styles, such as jazz, have developed practices that minimize cognitive workload (e.g., Johnson-Laird, 1988; Pressing, 1988; Sloboda, 1985). For instance, Flor and Holder (1996) demonstrated how individual guitarists use their instruments to make music *imitation information* visually explicit. However, no cognitive studies of groups performing an already learned music piece exist. We contribute to this literature by studying such a music group as a kind of "larger-than-individual" cognitive system, and showing important distributions of mental and environmental representations and processes.

In distributed cognition research, Hutchins (1990, 1995a) described how the organization of a navigation team allowed the arrival of information in an individual's workspace to drive local activities. For example, the arrival of landmark information (from the "bearing takers") over a loudspeaker prompted the "bearing timer" to write this information into his logbook. Thus, we know that work environments can be organized so that local activity produces information that serially propagates to other workspaces, cueing further local activity. This downstream local activity can be viewed as a global cue for the team members upstream. As such, Hutchins' study provides an example of institutionally-constructed (by the navy) global cueing of local activity. Our study builds on Hutchins' by examining how the environment can cue local activity when team members (musicians) must perform their activities *in parallel* and in an environment that is not pre-organized for global cueing of local activity. We argue that in any such situations, individuals can still use global structures to cue local skills, and we describe the conditions under which this kind of cueing can occur.

Method

Our method combines computational and observational analyses. Following Marr's (1982) notion of levels of analysis, we begin with a computational level characterization of the music-covering task. We then explore several representational-algorithmic level solutions to this problem, contrasting an individual-only distribution of representations and processes during covering with an individual + environment distribution. On the grounds of conserving computational effort, we argue for one particular individual-environment distribution of representations and processes. Our choice presents us with testable hypotheses; we check them by examining an actual situation that we believe serves as an instance (implementation) of our hypothesized approach to covering.

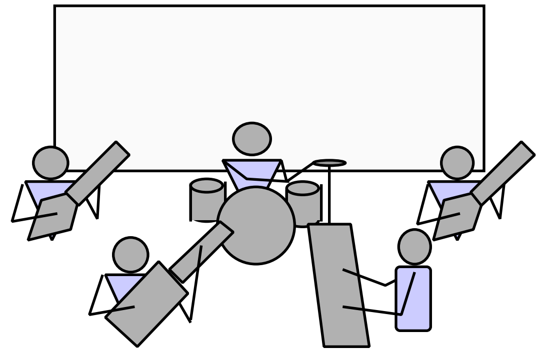

Figure 1. The spatial arrangement of the musicians during "Stray Cat Strut." From left to right, the musicians are: lead guitarist, bass player / singer, drummer, keyboard player, and rhythm guitarist.

The band we observed consisted of two guitarists, two drummers, two bass players, a keyboard player, and two singers. Each member had more than 10 years experience playing his or her instrument and at least five years experience playing in a professional band. The band was studied over a period of three months, from its formation of the group in September, to its first gig in December. During this time, nineteen rock and roll songs were learned (see Table 1). Note that not all members played on every song; for instance, the two bass players never played at the same time, instead they traded off depending on the song played. Practice sessions were held every Friday in the "Herbert A. Simon" lecture hall, and were approximately two hours long.

Set 1:	Set 2:
1. Feelin' Alright	1. Road House Blues
2. Ah Leah	2. Stuck in the Middle
3. The World Has Turned	3. Takin' Care of Business
4. Lightning Crashes	4. Creep
5. Sick of Myself	5. Girlfriend
6. Twisting	6. Santa Monica
7. Driver 8	7. Knockin' on Heavens Door
8. Say it Ain't So	8. Back to Rockville
9. Stray Cat Strut	9. Twist and Shout

Table 1. The Gig Song-list. The gig was played as two sets with a _ hour break in between.

The practice sessions were videotaped and all external representations were collected. A detailed analysis of the formation of the group and their use of these external representations has been prepared (Flor, 1997). Much of that analysis focuses on how the musicians use external representations to learn song *content*. However, this level of analysis is beyond the scope of our paper. Instead, we focus on how musicians *sequence* learned musical content, with an emphasis on understanding the role of external resources in assisting sequencing.

The Problem Structure of Song Covering

The goal of a cover band is to reproduce songs created by other bands. Thus, understanding the task facing a cover band requires understanding the composition of a typical song. A song is a complex sequence of notes and chords (simultaneous note combinations). In Western music, there are 12 basic sounds (tones), known collectively as the chromatic scale. These twelve tones are grouped into octaves, each of which contains the same 12 tones played at a different pitch. Musical instruments usually implement a limited number of octaves. For example, the common piano keyboard contains seven octaves, and a guitar, three. In addition to notes and chords, songs usually contain an atonal beat produced by a drum.

A song can be analyzed along several dimensions. First, there is the *melody* or basic sound sequence. When people hum or whistle a song, they are producing its melody. The role of the other notes in a song is to support or provide *harmony* for the melody. Finally, a song's notes and chords are played at a certain speed and tempo, or *rhythm* (see http://guitarhacker.com/research for audio samples of the differences). Different music styles have *evolved* stable sets of melodic, harmonic, and rhythmic regularities. For example, a common blues-style harmonic regularity is the I-IV-V chord progression (4 measures of the I-chord, 2 measures of the IV-chord, 2 measures of the I-chord, 1 measure of the V, 1 measure of the IV, and 2 measures of the I-chord). The untrained listener may not detect these regularities because of variations in melody, harmony, and rhythm. For example, the basic I-IV-V chord supports different melodies (melodic variation, with different words to the song), the progression can be played fast or with a shuffle feel (rhythmic variation), or the individual chord notes can be arpegiatted, that is, played one after another in sequence, rather than strummed (harmonic variation).

The typical rock song is performed by at least three musicians: a guitarist, bass player, and drummer, each playing a different instrument (i.e., guitar, bass, and set of drums, respectively). This kind of band is known as a "three-piece." The rock band can include additional musicians such as keyboard or horn players, or redundant musicians (with respect to instrument type), such as another guitarist. Any one of the musicians may sing or there is a designated singer, who plays no instrument. If there is a singer, the normative roles are as follows: the guitarist accompanies the singer by playing chords, the bass-player outlines the chord progressions with single notes, all synchronized to the drummer's beat.

The aim of a rock and roll cover band can be understood as the reproduction of a complex sequence of notes and chords played by at least a guitarist and a bass player, along with a beat played by a drummer. Each musician in the cover band need only reproduce the corresponding instrument part in the original song. As such, the group covering task is decomposed into smaller individual covering tasks.

With the above characterization, we can now derive a description of the computational components of song covering:

1. a representation of the song in terms of the notes and chords to play (i.e., the *pieces* to play) as well as their timing, and articulation
2. a *sequencing* process that determines the next note or chord to play, and
3. a *clock* process that produces timing information used by the sequencing process to determine *when* to play the next note or chord.

Symbolically, we represent these components as:

[pieces, sequencer(), clock()]

We tested the sufficiency of this characterization by developing a computer program that used these components to simulate song-covering (see http://guitarhacker.com/ research; note: all the hypothesized internal computational ecologies in this paper were implemented as computer programs, to verify computational sufficiency).

Given the computational structure of song covering, are there representations of these components *within* or *distributed* across individuals and their environments that *minimizes* computational effort while *maximizing* performance reliability?

Reducing Complexity Internally

One possibility is that the musicians represent all the computational components for song covering internally. To determine the feasibility of such an approach, we analyze one of the songs covered in our study: "Stray Cat Strut" (see Appendix A for a transcript of the original song; also refer to http://guitarhacker.com/research/straycat.wav, for a sound sample). For space considerations, we focus only on the guitarist. As mentioned, the role of the guitarist is to create a steady stream of chords that harmonizes with the singer's voice. Throughout the original song there is at least one chord per beat. In this case, the minimum number of chords a guitarist needs to play is 328. We assume that the particular chords have already been internalized, but that their sequence must be learned. Without a strategy to reduce complexity, the distribution of processes and representations — what we will call the *computational ecology* — can be depicted as follows:

Internal[pieces: 328-chords, clock(), sequencer()], External[]

The notation: "pieces: 328-chords," denotes a specific implementation of representation 'pieces' that uses 328-chords. Remembering such a lengthy chord sequence clearly places a heavy burden on the musician, especially considering that this is but one of 18 songs that must be learned. Some means of reducing complexity is needed. Chunking is one obvious means of reducing complexity (Chase & Simon, 1973; Miller, 1957). But how should the musician chunk the song? Where should the chunk boundaries lie? Repeating chord patterns are one candidate for chunking. Examining the chord progressions reveals five chord patterns that repeat at least once (see Figure 2):

1. Cm, Cm, A#, A#, G#, G#, G7, G7
2. Fm, Fm, D#, D#, C#, C#, C7, C7
3. Fm, Fm, Fm, Fm, Fm7, Fm7, Fm7, Fm7
4. Cm, Cm, Cm, Cm, Cm7, Cm7, Cm7, Cm7
5. D7, D7, D7, D7, G7, G7, G7, G7

Figure 2. The five chord patterns

These patterns were determined by a chord pattern analysis program that used an exhaustive algorithm to find repeating patterns of size 8 (http://guitarhacker.com/ research/visualtone.html). If the guitarist chunks these patterns, the length of the sequence that must be remembered reduces an order of magnitude from 328 different chords to 41 chord-chunks (1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 3, 4, 3, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 4, 3, 5, 1, 1, 1, 1).

To see whether there was evidence for these chunks, we examined video of the rhythm guitarist during both practice and actual performance of "Stray Cat Strut" and found a certain pattern of chord fingerings (see Figure 3 below). Simultaneously depicting all the fingerings on the diagram shows an interesting spatial pattern: the choice of chord fingering divides the guitar's fret board into five sections which correspond to the hypothesized five chunks mentioned earlier. We take this as evidence of chunking because in other songs that have similar chords the fingering is different. For instance, the song Creep also uses a Cm chord, but in Creep Cm is fingered as a bar chord starting on the third fret, instead of a bar chord on the eighth, as it is in Stray Cat Strut.

With this kind of chunking, the computational ecology becomes:

Internal[pieces: 41-chunks, clock(), sequencer()]
External []

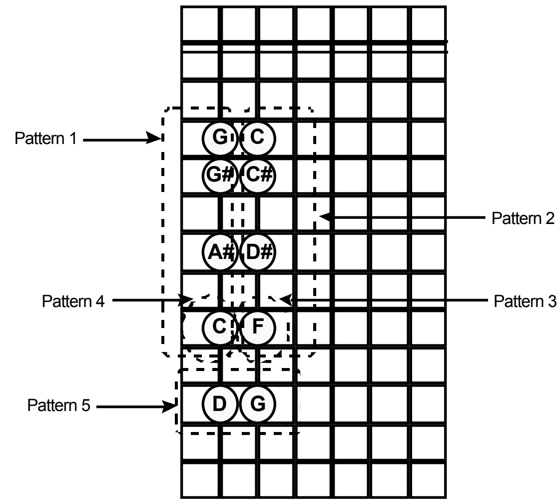

Figure 3. The spatial distribution of bar-chords on the rhythm guitarist's fret board for the song "Stray Cat Strut" (for clarity, only the root note is shown).

Complexity can be further reduced by a sequencer which repeats a chunk a specified number of times before moving onto the next chunk. In our example, the "1"-chunk would be played 8 times, followed by the "2" chunk played 3 times, and so on. The computational ecology would be depicted as follows:

Internal[pieces: 41-chunks, clock(), sequencer(): counter]
External []

The notation "sequencer(): counter" denotes a specific implementation of the process 'sequencer' that uses a counter. Although an order of magnitude less than 328, 41 is still a lengthy sequence especially if covering each of the 18 songs requires as many.

What else can reduce the sequence's complexity? We will show how the use of environmental representations, combined with a specific distribution and transformation of processes can further reduce complexity.

Reducing Complexity Externally

To perform the task without external aids requires mentally keeping track of how many times a pattern has been played, and then playing the next pattern. In effect, the musician must count how many times a chunk (pattern of chords) has been played until a certain number is reached, and then play the next chunk. Moreover, the musician must play his or her chunks at the correct time, otherwise parts will be unsynchronized. This requires that all musicians have a perfect internal clock that ticks at the same interval. Assuming such a clock, the covering process would proceed as follows. Each musicians waits for some globally available signal to start playing. The receipt of this signal starts the musicians' internal clocks. Each musician then plays the appropriate chunk at the appropriate tick, without regard to what the other musicians are playing. We represented this situation symbolically as follows:

Internal[pieces: 41-chunks, clock(), sequencer(): counter],
External []

Obviously, such an internal clock is infeasible. Thus, one way of using the environment is as a clock. This changes the computational ecology as follows:

Internal[pieces: 41-chunks, sequencer(): counter]
External [clock()]

In a band, the drummer is this external clock. In providing a steady beat, the drummer relieves the other musicians from having to internalize a perfect clock. The musicians can synchronize their productions to the drummer. Video of the musicians in the band studied nodding their heads or tapping their feet to the beat of the drums provides evidence of this kind of synchronization.

Thus, one cognitive function that is distributed to the environment is the clock function. However, this is more than just an external distribution of process. One central, external clock takes the place of several internal clocks. Therefore, *both a distribution of process and a reduction in redundancy* takes place. Are there other representations or other processes that can be distributed to the environment? What are some of the ways a guitarist can reduce complexity?

One way to reduce complexity is by having a more detailed set of instructions (i.e., sheet music) about what to play and when to play it.

Internal[sequencer(): visual_routines]
External [pieces, clock()]

Such a distribution requires that the musician maintain visual contact with the instruction sheet, and uses visual routines to determine what the next piece in the sequence is, that is, read the next chord to the right on the instruction sheet. However, despite having chord progression sheets at their disposal, this kind of computational ecology was not observed during practice or performance, suggesting some kind of difficulty with this approach.

Undoubtedly, one could hypothesize other computational ecologies for covering that require musicians to construct various kinds of external aids. But our observational data does not show the musicians using constructed external aids for sequence support. Thus, rather than focusing on structures that musicians could create, we focus instead on uncovering complexity reducing resources which are readily at the musicians' disposal. To find these, we searched the video transcript for structures that could convey task-relevant information for sequencing. More specifically, we looked for environmental structures that reliably predict when a musician should play the next chunk in the sequence. Such a global cue must occur temporally close to when a chunk switch should occur. One candidate is the parts played by the other musicians and the singer. In playing his or her own piece, each musician creates structures which can serve as sequence cues for the others.

If such global cues exist, then the guitarist (and other musicians) can change strategy from one of counting to one of playing a chunk until a specific chord or note is heard (cue) and then playing the next chunk. Let us assume that indeed, the strategy is to repeat one of the five patterns at a particular beat until a cue is heard: 1, 2, 1, 3, 4, 3, 5, 1, 3, 4, 3, 5, 1. This reduces the complexity of what must be recalled 3-fold from 41 to 13. Thus, through chunking *and* changing strategy, the sequence can be reduced from 341 to 13. The corresponding computational ecology is as follows:

Internal[pieces: 13-chunks, sequencer: repeat_until_cue]
External[clock(), cues]

This is not simply a distribution of existing processes and representations. The task is transformed from using a counting strategy, which required chunks, an internal counting program and counter, to a listening strategy which consisted of a smaller set of chunks and a repeat-until-cue sequencer. Externally, the former strategy required a clock and no cues. Whereas the latter requires a clock and cues. The best part is that the global cues come for free!

Do band members use this strategy? It can only happen if there are cues that reliably predict when to change to the next chunk. Examining the transcript reveals that the singer's words reliably indicate when to play the next chunk. For example, the first switch from chunk 1 (Cm, A#, G#, G7) to chunk 2 (Fm, D#, C#, C7) is preceded by the phrase "I got cat class and I got cat style." Similar phrases precede most of the chunk switches. Thus, the repeat-until-cue strategy is at least plausible.

Yet simply because there are cues that reliably indicate when to change to the next chunk, that is not sufficient to prove that the musicians use them. But if the musicians are indeed using the words of the song to cue their repeat-until-cue sequencing strategy, then one would expect them to make mistakes in those portions of the song in which no singing takes occurs, for instance, during the guitar solos.

An Instance of Song Covering

To test this prediction, we analyzed an actual performance of "Stray Cat Strut" by the cover band (see http://guitarhacker.com/research/straycat.avi, starting at 1:14 into the video). The analysis revealed three blatant deviations from the original song. The first deviation occurs at the beginning of the song, when instead of playing chunk 1, 6 times after Guitar Solo 1; the bass player / singer immediately starts singing, "Black and orange..." The next deviation occurs during the second guitar solo, the drummer and bass player stop slightly earlier than they should, but they correct themselves and correctly play on (2:27). However, they forget to stop (go tacet) at the end of the solo (2:38). Despite these deviations, the band plays on. The third deviation happens during the third guitar solo, where the band incorrectly stops (tacet) before the guitarist has begun the solo (3:24). This time the band does not play on. Note, each of these errors occurs during the guitar solos, *when there is no singer* producing the cues which sequence the next chunk. These errors were not specific to the situation reported in this paper. The band made mistakes in at least one of these three sections, up to and including the gig performance.

Further evidence of the use of the repeat-until-cue strategy comes after the third mistake. The following discussion ensues:

3:26	Bass Player / Singer	How's it go?
	Guitarist	(plays solo *beginning*)
3:54	Bass Player / Singer	What's the ending?
	Guitarist	(plays solo *ending*)

Also during this discussion, the drummer complained that he could not hear the guitarist. This complaint is important because it helps explain the bass player's behavior when they retry the song.

The band once again tries the song. Near the end of the second solo, the following signally activity occurs:

6:15	Bass Player / Singer	Turns to drummer, makes an up-down motion with his bass

Then, near the end of the third solo, the bass player once again signals the drummer:

7:11	Bass Player / Singer	Turns to drummer, makes an up-down motion with his bass

We take this as evidence of the repeat-until-cue strategy. The important difference that the bass player is substituting a *visual* cue for the *audio* cue normally used by the drummer.

In summary, converging lines of evidence suggest that musicians in the cover band use the repeat-until-cue strategy. In particular, we found: (a) there are global cues (the singer's words) that predict when to switch chunks; (b) the mistakes occurred in sections of the song where the singer did not sing and thus there were no global cues; (c) the bass player asked how the beginning and ending of the second guitar solo sounded like; and (d) the bass player turned to the drummer to signal when to stop.

Thus, global cues are driving local behavior. And, more importantly, *the global cues are an emergent property of the group performance itself* and are not instrumented in advance. Finally, the role of the environment in delivering the global cues should not be downplayed. The musicians were able to sequence their local activities based on the other musicians' activities because they had audio access to each others' performances. Hutchins (1990) provided an example of the importance of *visual* openness in coordinating self-correction activities among navigators. We extend his work by also emphasizing the importance of *audio* openness in coordinating sequence activities.

Discussion

What do our findings mean for group performance? First, to perform in a group, it makes good sense to practice in a group. This might not seem too surprising, but we have shown that the computational problem of group performance differs from the computational problem of individual performance for musicians in a cover band. In particular, the most effective means for breaking the overall problem faced by the group (i.e., reproducing another band's song) into individual skills was shown to depend critically on the available actors and resources. We argued that a group-level task decomposition (algorithm and representation) was computationally simpler than an individual-level decomposition in this case, and moreover, that skills learned by practicing alone need not correspond to skills needed for group performance. Local skills (chord chunks) need not be activated by global cues (particular song vocals), though the same local skills naturally create global cues as a side effect. Using a simple complexity analysis, we argued that the algorithm for individual play (repeat chord chunk some number of times) is computationally more difficult than the algorithm that relies on cues that emerge from group activity (repeat chord until a certain vocal is heard). Of course, we can only make this claim because we assume that counting chunks requires short-term memory, whereas a vocal cue can be trivially picked by humans' specialized perceptual systems.

The second implication of our results is that task completion can be facilitated when group members work in parallel *and* have access to each others' on-going performances. In studying Navy ship navigation, which was specifically organized as a serial process, Hutchins (1990) found that the way information propagates through established channels serve to cue local actions. We have shown that actors operating in parallel can also rely on cues to sequence local activity. Such cues are not arranged in advance but emerge as a side-effect of parallel activity and proximity. Because the musicians in the band can hear one another, they can use sounds made by others to cue action. This kind of cueing has been observed in other domains. Flor (1994) showed how two programmers working side-by-side used their ability to easily see (access) each others' computer screens to *opportunistically* coordinate a task that would otherwise require a high degree of prior planning.

These findings have several implications for the design of computing technologies intended to support cooperative learning between *remote* participants. First, *such technology needs to make the performance of others globally accessible*. This means more than merely providing a context switch that allows remote participants to access information about each others' performance on demand. Ideally, performance information should be *constantly presented to all participants* since they may not have prior knowledge about which global cues are relevant. The constant presentation of performance information allows participants to opportunistically take advantage of any relevant cues that emerge. This is not a problem when the performance information is represented in an "open" medium, such as auditory information, where co-present participants cannot help but hear each others' performance. However, performance information may be visual, such as when the participants engage in a joint writing task. When information is represented in such a "closed" medium, it is non-trivial to constantly present the information to all participants. Fortunately, computing technology allows information to be re-represented and simultaneously presented in different media. Consider the writing task. As a participant types a word, it can be simultaneously broadcast to a voice synthesizer on the co-participants' computer and each participant can have a different synthesized voice. A less complex approach is for participants to have an editor that shows their co-participants' typing activity as faded-text overlaid in the editor's background using different colors for each participant. Thus, although visual information is not typically open to all participants, technology that either places the information in another medium or overlays information of others in the same medium, provides constant presentation of group performance.

A second design implication of this research is that participants should have *the ability to draw attention to those cues in the group performance that can help individual participants learn their performances*. For instance, in the music covering case, the bass player turned to the drummer and used a visual cue to temporarily substitute for the auditory cues (from the singer and other band members) that the drummer had not yet learned. Performance information was auditory and the temporary learning cue was visual. The multimedia aspect of computing technology makes such cross-media abilities possible. If the performance information is auditory, visual media can be used to simultaneously provide temporary learning cues. Similarly, if the performance information is visual, audio media can be used to provide the temporary learning cues. Technology may also allow the actual performance medium to hold these temporary learning cues. For example, the programming environment described previously, which visually overlays participants' screens, could also be used to overlay temporary visual learning cues.

Both design implications suggest working environments that may be distracting for participants and, thus, negatively effect performance. Empirical will be able to quantify whether the benefits of using global cues to direct local performance outweighs the distractions which result from either the constant presentation of performance information to all participants or the introduction of temporary learning cues.

To summarize, we have shown that the computational problem of song covering lends itself to a more efficient and effective solution in the case of group performance than in the case of individual performance. This follows because by working together in a group, individuals can use cues that emerge to help sequence actions. In the musical group studied, we found that in fact that the group uses the simpler solution that makes use of emergent cues. We believe our findings are not specific to cover bands, but generalize to all group activities in which individuals perform actions in parallel and have access to the on-going activities of others.

References

Chase, W., and Simon, H. (1973). The mind's eye in chess. In W. Chase (Ed.), *Visual Information Processing*. New York: Academic Press.

Flor, N. (1994). *Dynamic organization in distributed cognitive system.* Unpublished Doctoral Dissertation, San Diego, CA: University of California San Diego.

Flor, N., and Holder, B. (1996). Hearing with the eyes: A distributed cognition approach to guitar song imitation. *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, New Jersey: Ablex.

Hutchins, E. (1990). The technology of team navigation. J. Galegher, R. Kraut, C. Egido (Eds.), *Intellectual Teamwork*. NJ: LEA.

Hutchins, E. (1995a). *Cognition in the Wild*. MA: MIT Press.

Hutchins, E. (1995b). How a cockpit remembers its speed. *Cognitive Science, 19,* 265-288.

Johnson-Laird, P. N. (1988). Freedom and constraint in creativity. In R. J. Sternberg (Ed.), *The nature of creativity*. Cambridge, U.K.: Cambridge University Press.

Kirsh, D. (1995). The intelligent use of space. *Artificial Intelligence, 73,* 31-86.

Marr, D. (1982). *Vision*. San Francisco, CA: Freeman.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63,* 81-97.

Norman, D. (1992). *Turn Signals are the Facial Expressions of Automobiles*. MA: Addison-Wesley.

Pressing, J. (1988). Improvisation: Methods and models. In J. A. Sloboda (Ed.), *Generative processes in music: The psychology of performance, improvisation, and composition*. Oxford, U.K.: Clarendon Press.

Simon, H., and Sumner, R. (1968). Pattern in Music. In B. Kleinmuntz (Ed.), *Formal Representation of Human Judgement,* New York, Wiley, 1986, 219-250.

Sloboda, J. A., (1985). *The musical mind: The cognitive psychology of music*. Oxford, U.K.: Clarendon Press.

Appendix: Transcript for "Stray Cat Strut"

As a shorthand, **riff** = Cm Cm A# A# G# G# G7 G7

```
(DRUMS, Bass plays riff once)(Guitar Solo 1)(riff X 2)
Cm A# G# G7 Cm A# G# G7 Cm A# G# G7 Cm A# G# G7
ooooooooooooooooooooooooooooooooooooooooooooooo
Cm        Cm     A#    A#  G#      G#   G7 G7  (riff)
Black and orange stray cat sitting on a fence
Cm    Cm   A#    A#       G#  G#     G7 G7 (riff)
Ain't got enough dough to pay the rent
Cm  Cm   A#    A#  G#     G# G7 G7
I'm flat broke but I don't care, I
Cm    (tacet)
Strut right by with my tail in the air
Fm    Fm  D#    D#    C#  C#   C7   C7
Stray cat strut I'm a ladies cat, I'm a
Fm     Fm  D#    D#  C#      C#   C7  C7
Feline Casanova, Hey man that's that, Get a
Fm   Fm     D#    D#     C#     C# C7 C7
shoe thrown at me from a mean old man, I
(tacet)
Get my dinner from a garbage can
(riff) Meow (riff), Yeah don't cross my path
(guitar solo: riff X 3 during solo)
Fm Fm    Fm   Fm Fm7 Fm7 Fm7 Fm7 Cm Cm Cm Cm Cm Cm Cm
Cm
I don't bother chasing mice around.. I
Fm    Fm       Fm  Fm Fm7    Fm7   Fm7  Fm7
slink down the alley looking for a fight
D7      D7     D7        D7   G7     G7    G7 G7
Howling to the moonlight on a hot summer night
Cm      Cm  A#    A#        G#   G#   G7 G7
Singing the blues while the lady cats cry.
Cm   Cm    A#   A#       G#   G#   G7 G7
Wild stray cat, you're a real gone guy. I
Cm   Cm      A# A# G#         G#    G7 G7
Wish I could be as care free and wild but I
(tacet)
Got cat class and I got cat style.
(riff X 6) (guitar solo: riff X 3 during solo)
Fm Fm    Fm   Fm Fm7 Fm7 Fm7 Fm7 Cm Cm Cm Cm Cm Cm Cm
Cm
I don't bother chasing mice around.. I
Fm    Fm       Fm  Fm Fm7    Fm7   Fm7  Fm7
slink down the alley looking for a fight
D7      D7     D7        D7   G7     G7    G7 G7
Howling to the moonlight on a hot summer night
Cm      Cm  A#    A#        G#   G#   G7 G7
Singing the blues while the lady cats cry.
Cm   Cm    A#   A#       G#   G#   G7 G7
Wild stray cat, you're a real gone guy. I
Cm   Cm      A# A# G#         G#    G7 G7
Wish I could be as care free and wild but I
(tacet)
Got cat class and I got cat style.
(riff) (tacet) (guitar solo) (end)
```