A Brief Tour of the Learning Sciences via a Cognitive Tool for Investigating Melodic Phenomena

Craig Graci

Cognitive Science Program

State University of New York at Oswego

Abstract

The goal of this paper is to contribute fragments of concrete understanding to the on-going search for meaning in ideas associated with the learning sciences (Sawyer, 2006b) by adding to the repertoire of domain specific studies pertaining to this relatively new field. The learning sciences are first characterized as a system of learning principles, fundamental laws about learning that tend to be embraced by theorists and practitioners of the learning sciences. These principles are then examined through the lens of a cognitive tool (Jonassen & Carr, 2000) for studying musical melody. This cognitive tool is a computational system consisting of Clay, a musical knowledge representation language, and MxM, its infrastructural host. MxM and Clay are informally introduced and subsequently applied to problems associated with the representation and analysis of melodic structure. As the presentation unfolds, manifestations of the learning science principles are identified and discussed.

A Brief Tour of the Learning Sciences via a Cognitive Tool for Investigating Melodic Phenomena

In this paper I present elements of a computational system called MxM that can be viewed as a research tool for investigating ideas that are central to the fields of music cognition (music and psychology) and cognitive musicology (music and artificial intelligence). Recently this system has been used to study phenomena related to *grouping structure* (Graci, 2008a, 2008b), as elucidated by Lerdahl and Jackendoff (1983) in their generative theory of tonal music, and *reductional structure* (Graci, 2009), as articulated by Deutsch and Feroe (1981) in their formalism for the internal representation of pitch sequences. But MxM can also be viewed as an *educational microworld* (Papert, 1980), as a *cognitive artifact* (Norman, 1991), or, more generally, as a learning/thinking technology that taps into the potential of the distributed cognition framework (Salomon, 1993). In fact, MxM is a fairly direct descendent of two research programs related to computational learning environments (Graci, Narayan, & Odendahl, 1989, 1992). The emphasis in this paper is on MxM as a cognitive tool (Jonassen & Carr, 2000), a technology that affords learners enhanced opportunities to engage in processes of knowledge construction and reflective thinking.

For reasons well articulated by Kim and Reeves (2007), results achieved by incorporating computers as cognitive tools into classroom activities have been limited, some would say disappointing. Over the past two decades, however, a coalition of powerful ideas has emerged under the heading of "the learning sciences" (Sawyer, 2006b), which might well be leveraged into a renewed, more successful effort to enhance educational experiences with cognitive tools. These learning science ideas revolve around such broad themes as *distributed cognition*, *social/technological scaffolding*, and *mediating artifacts* (Cole & Engestrom, 1993). They have deep roots in sociocultural schools of thought (Daniels, Cole, & Wertsch, 2006), but are being reinterpreted by educational theorists in light of the computer and advances in cognitive science. What, precisely, is meant by the phrase "the learning sciences"? Is the field really a science? The former question is discussed explicitly by Sawyer (2006b), and addressed implicitly throughout this paper. The latter question is raised by Papert (2006), and answered in a politely equivocal fashion which might be paraphrased as "perhaps, in an incipient sort of way".

Challenging metaphysical issues (e.g., defining the scope and defending the nature of the learning sciences) notwithstanding, there are substantive ideas associated with the learning sciences that call not only for appreciation, but for application as well in the form of research and practice (Bransford, Brown, & Cocking, 2000; Sawyer, 2006a). Due perhaps to their philosophical roots, these learning sciences ideas tend to be discussed rather abstractly in much of the literature. Compelling as the ideas are when expressed in narratives crafted by seasoned theorists, their understanding is enriched when examinined in light of meaningful manifestations of the ideas. For example, Sivasubramaniam (2004) and Morgan, Brickell, and Harper (2008) elucidate aspects of distributed cognition by discussing, respectively, the use of graphing calculators in mathematics education and a specialized "copy and paste" function to facilitate learning through enhanced interaction. One of the motivating forces for writing the current paper was to add to the collection of domain specific studies relating to the learning sciences.

The cognitive tool featured in this paper, MxM, is grounded in the field of cognitive musicology. Thus, it affords the learner a range of opportunities to participate in the computational modeling of musical phenomena. MxM has been extensively used in the Cog316 "Cognitive Musicology" course at SUNY Oswego for more than a decade, and variants of Clay, the symbolic programming language embedded within MxM, have been used in courses at SUNY Oswego since the language was first implemented nearly two decades ago. From a practical point of view, use of this cognitive tool will enrich the learner's appreciation of music, as a side effect of engaging in generative processes of analysis and composition. According to Whitehead (2002), "Art is the imposing of a pattern on experience, and our aesthetic enjoyment is recognition of the pattern." The computational modeling framework and the corresponding conceptual modeling methodologies discussed in this paper were designed to enhance the modeler's sensitivity to melodic pattern, and hence their aesthetic enjoyment of music. From a theoretical point of view, this tool provides opportunities to gain a better understand of the cognitive processes associated with music and to sharpen one's insights into generic processes of learning.

Sandwiched between this introduction and a brief conclusion are four main sections. The first section characterizes the learning sciences as a system of principles. A number of principles are named and described. These principles are examined throughout the paper, in keeping with the expressed goal of exploring a selection of learning science principles through the lens of a cognitive tool with a musical orientation. The principles have not been presented in a systematic way, however, for their overlapping nature precludes that possibility in a paper such as this one. The strategy employed is a content driven strategy, the idea being to take advantage of just a few choice occasions for which discussion of a principle seems particularly warranted, striving to give each principle something like equal time. The remaining three sections, rife with computational constructs and musical models, provide opportunities for reflecting upon manifestations of the learning science principles. The second section introduces MxM and Clay in a fairly natural way. The third section discusses the modeling of grouping structure. The fourth section discusses the modeling of reductional structure.

The learning sciences are often discussed in relation to the STEM fields, science, technology, engineering, and mathematics. By discussing elements of the learning sciences from the relatively fresh perspective of music, in particular the analysis of melodic structure, I wish to enrich the discussion. Specifically, by explicitly relating elements of MxM and examples of its use to a selection of ideas from the learning sciences, I intend to contribute fragments of concrete understanding to the on-going search for meaning in these ideas.

The Learning Sciences as a System of Principles

Taking a lead from C. S. Peirce who preferred to view semiotics not as a "science of signs", as some would characterize it, but as a "system of principles" (Danesi, 1998), I find it useful to conceive of the learning sciences as a system of principles. Perhaps this perspective can be justified by observing the fact that L. S. Vygotsky's sociocultural theory of development, which features prominently in the learning sciences, was grounded in the human propensity to create and use signs (Veer, 2007). What are the basic assumptions embraced by learning scientists as a basis for their reasoning about learning and teaching? That is, what are the principles of the learning sciences? Here are my picks for some of the most essential principles.

Distributed Cognition Principle Cognition is something *accomplished* through collaborative interactions involving people and artifacts rather than something *possessed* by individuals in isolation (Pea, 1993).

Constructionist Principle This is an extension of Piaget's constructivism – the theory that learning involves the building of knowledge structures within the individual mind – which adds (1) the idea that "this happens especially felicitously in a context in which the learner is consciously engaged in constructing a public entity" (Papert, 1980), and (2) a "more distributed view of instruction, one where learning and teaching are constructed in interactions between the teacher and students as they are engaging in design and discussion of learning artifacts" (Kafai, 2006).

Deep Learning Principle Education is best accomplished by privileging engagement over explanation, uncoverage over coverage, questioning over answering, reflection over reaction, representation over information, and process over product.

Project-Based Learning Principle Deep learning accrues as a side-effect of engagement in an incremental, holistic process of artifact creation in response to the consideration of a substantial problem of interest to the learner.

Learner-Centered Design Principle Favor bridging the "gulf of expertise" over the "gulf of execution" and the "gulf of evaluation". In other words, place emphasis on *scaffolding* which affords opportunities to enhance understanding by bridging the conceptual distance between a novice and an expert in the domain of interest, rather than on tools or methodologies that merely provide ease for the performance of tasks (Quintana, Shin, Norris, & Soloway, 2006).

Imagery Principle Educators need to search for ways in which the various functions of imagery (e.g., effortless structural interpretation, determinism, perception-action coupling, and pre-interpretation) can be used to support learning, creativity, and reasoning (Schwartz & Heiser, 2006).

Inscription Principle Students learn by doing and by *thinking about what they have done.* Creating external representations of one's thoughts in some sort of inscription system for reflecting upon one's thinking and sharing one's thoughts with others is of central significance to deep learning.

The principles in this list are far from being mutually exclusive. Ideas about learning which are grounded in cognitive science tend to engender interconnectedness. Nor is this list exhaustive. Plenty of other potentially valuable ideas associated with the learning sciences can be found in the literature (Sawyer, 2006a; Salomon, 1993). What I would claim about this particular list of learning science principles is that a commitment to feature some of these principles in a program of learning would assure an educational experience that is grounded in the learning sciences – the more principles featured, the stronger the assurance.

Introduction to Clay and MxM

Clay is a simple symbolic programming language that can be adapted to manipulate different sorts of virtual objects. Clay has, in fact, been adapted to manipulate rectangles, coins and dice, number savvy rabbits (Graci et al., 1989), and musical notes to obtain Mondrian, chance, number theory, and melody "worlds". When used to represent and process musical knowledge Clay is housed in a computational framework called MxM. MxM provides infrastructural support for Clay in the form of sound, graphics, and file processing. It also provides meta-level commands for analyzing melodies represented as Clay programs.

The Note and Basic Clay Primitives

As a music knowledge representation language Clay features *note* objects, each one loaded with more than a dozen properties. The most prominent properties of a note are its alphabet (generalized scale), pitch (degree within the scale), location (register), duration (with respect to one beat), timbre, and volume. By default each note is instantiated so that its alphabet is the C-major scale, its pitch is the first degree, its location is within the vicinity of middle C, its duration is equal to one beat, its timbre that of a piano, and with volume is moderate. Melodies are modeled by playing the note, resting the note, and manipulating the state of the note.

The primitive Clay commands to play and rest the note are simply `P` and `R`. The pitch changing commands are `RP` and `LP`, which raise and lower the pitch of the note by one degree of the scale. The commands `X2`, `X3`, `X5`, and

`X7` expand the duration of the note by factors of 2, 3, 5, and 7. The commands `S2`, `S3`, `S5`, and `S7` (S for "shrink") are inverse to the expand commands. A couple of examples will serve to clarify any unaddressed issues relating to the note's state and behavior. The first focusses on change of pitch: `P LP LP P RP P RP P` ⇒ `C \ A / B / C`. The second illustrates change of duration: `P P X2 P S2 P P X2 P S2` ⇒ `C C C2 C C C2`. The convention used in the body of this text is that a command sequence to the left of a double arrow (⇒) produces the sounds indicated to the right of the arrow. On the right of the arrow the letters correspond to physical pitch classes, the slashes indicate relative direction of movement, and numbers to the right of a note indicate duration relative to one beat. The first step in getting acquainted with Clay is merely to play around a bit with the primitives.

Macro Definitions

Sequences of primitive commands quickly become too cumbersome to work with effectively. The concept of the *macro* serves to remedy this situation. A macro is simply a symbol that denotes a sequence of symbols. Macros are a great convenience because they augment the vocabulary of note manipulating symbols. Macros are distinguished in Clay from *commands* and *reductions*, two alternative types of Clay definition, each with very different semantics. A macro denotation is established by means of a macro definition. In Clay, macro definition takes the form *symbol* >> *sequence*. For example, `PL >> X2 P S2` renders `PL` effectively `X2 P S2`, a "play long" macro. Similarly, `PS >> S2 P X2` renders `PS` effectively `S2 P X2`, a "play short" macro. A slightly more complicated macro would be `PD >> X3 S2 P X2 S3`, a "play dotted" macro, which would play the note for 1.5 times its current durational value. With these macros in effect, `PD PS PL` ⇒ `C1.5 C.5 C2`. A "standard suite of macros" has been specified (which includes `PL`, `PS`, and `PD`) in Clay. The second step of acquaintance with Clay is usually to define the standard macro suite.

The activity of defining the standard suite of macros is just the tip of the constructionist iceberg with respect to modeling melody in Clay. But it is worth noting at this time that the implementation of these macros is at least a nod in the direction of the ***constructionist principle***. This principle is more fully realized when the learner commences to conceive of macros on their own and implement macros suggested by more experienced individuals.

Generative Explorations of the Melodic Surface

A third step in developing Clay modeling skills is to write macros that either play a given melodic fragment or generate a melodic fragment subject to a set of constraints. The melodies and the constraint sets could be provided by a human teacher, or they could be posed by MxM. Both approaches are useful. In this section MxM will pose the problems. What is noteworthy, regardless of the nature of problem posing agent, is that learning is taking place in accordance with the ***constructionist principle***. This principle was introduced earlier by embedding two insightful quotes into a rather long sentence. A more direct way to describe the principle would be: an elaboration of Piaget's constructivism that emphasizes (1) the creation of external artifacts, and (2) social/cultural/technological interaction. The act of writing Clay programs, regardless of the type, produces external artifacts – the programs. These programs are an explicit representation of the learner's thoughts, and can thus be shared with other students. This sharing is a form of *social interaction*, intended both to secure feedback and to suggest possibilities. The programming is sometimes directed towards modeling, and perhaps varying, classic works. In this is regard it possesses an element of *cultural interaction*. And, of course, program development is in some respects the epitome of *technological interaction*. It has been observed that constructionism does not preclude instructionism (Kafai, 2006), but rather endeavors to place the latter in a proper perspective. The fact that a human or machine is involved in posing problems for the learner to solve is significant. A careful sequencing of the problems, or at least a careful selection of problems which is sensitive to the learner's zone of proximal development (Vygotsky, 1978), is an important instructional contribution to a program of learning. True constructionism embodies all of these basic ideas, and myriad subtle variations on each.

There are canned problems available in Clay that either require the learner to model a melodic fragment or ask the learner generate a melodic fragment subject to a set of constraints. For lack of better terms, the former type of problem is called a "puzzle" and the latter type is called a "pattern". The puzzles and patterns described in this section are called "surface" puzzles and patterns, meaning that modeling the various dimensions of structure (e.g., grouping structure and reductional structure) are considered to be irrelevant.

If you type the metacommand `-surfacepuzzle` into the text input field a simple surface puzzle is established. (Metacommands are introduced with a dash.) The puzzle amounts to writing a Clay macro called `solution` that generates a given sequence of notes. By typing `-puzzle` you see the desired sequence of notes. Once you believe that you have a solution to the puzzle, you can ask MxM to evaluate the solution by simply typing `-check`. If you do this, you will get one of three responses. The most favorable is simply "Good!". The least favorable is "Sorry, but you actually don't have a solution." The most interesting is "OK, but a better solution exists." Figure 1 is suggestive of puzzle interaction.

– Insert Figure 1 here –

There are two "levels" of puzzle. Level 1 puzzles are all posed in the key of C-major. The puzzle featured in Figure 1 is a simple level 1 puzzle. (It is the beginning of Robert Schumann's "Melody.") Level 2 puzzles are intended to be solved in the context of generalized scales, or alphabets – to use the terminology of Deutsch and Feroe (1981). Figure Figure 2 presents the pitch alphabet concept by means of a small number of examples.

– Insert Figure 2 here –

The rather more elaborate example shown in Figure 3 illustrates aspects of a simple level 2 puzzle. Three solutions are proposed. The first attempt is sonically correct, but not theoretically correct, as the first `-check` metacommand observes. The second attempt is both sonically and theoretically correct, but it is not conceptually sound, as indicated by the second `-check` metacommand. In this case, the `-check` command determines that a better fit exists by simply comparing the number of primitive pitch changing commands processed in executing the proposed solution with the number processed in a favored internal solution. As a rule, the number of pitch changing commands will be smallest when the melody is modeled in the preferred alphabet. The third attempt hits the mark! By conceiving of the melodic line (the start of the Rodgers and Hart tune "My Favorite Valintine") in C-minor, the solution is "tighter", and more correct.

– Insert Figure 3 here –

If you type `-surfacepattern` into MxM a set of sequences of notes is established, the English specification of which may be observed by typing `-pattern`. The problem is to write a Clay macro called `pattern` that not only adheres to the constraints given in the specification but which is also aesthetically, or perhaps intellectually, pleasing. The problem of producing the pattern is a relatively modest compositional task. But the social interaction that results in a small class of students who play their patterns and share their pattern generating programs with one another can be quite impressive. Students instinctively want to learn the "secrets" of sonic successes. As with the puzzles, there are the two levels of difficulty for patterns. Figure 4 provides a simple Level 1 example.

– Insert Figure 4 here –

A Balinese Gamelan Music Project

During the Spring of 2008 the Cog316 "Cognitive Musicology" course at SUNY Oswego engaged in a modest project relating to Balinese Gamelan Music. About mid-semester someone in the class learned that the SUNY Oswego Music Department had arranged for the Balinese Gamelan Ensemble of the Eastman School of Music to perform in Oswego just a few weeks hence. Since the schedule is pretty loose in this course, we decided to take a few class periods to prepare to enjoy the show. What ensued may be viewed as an expression of the ***project-based learning principle***. In brief, this principle suggests that deep learning tends to result from the construction of artifacts in response to a problem of some interest. Project-based learning happens all the time in classrooms that are void of cognitive tools. But the example that I am about to recount is suggestive of the fact that cognitive tools can play a pivotal role in fostering project based learning, particularly when coupled with other principles of the learning sciences.

Consider the three pitched rhythm patterns depicted in Figure 5. The grids on the left are a visual representation of the Clay code on the right. The timbre (`STEELDRUMS`) and scale (`BLUES`) in which the patterns are realized were chosen for artistic reasons, and are not particularly relevant to the scenario. The central point to be observed is that two simple rhythm patterns (`ONEF` and `ONEM`) can easily be played in parallel in Clay to achieve a more complicated rhythm pattern. The "hat" in Clay is the symbol used to denote parallel execution. This rhythm pattern would typically be played by two people on a pair of pitched percussive surfaces.

– Insert Figure 5 here –

This same idea could be applied to two more simple rhythm patterns (say `TWOF` and `TWOM`) to achieve another more complex pattern (say `TWO`), which would typically be played by two other people. The two resulting rhythm patterns (`ONE` and `TWO`) could then be combined to produce a yet more complex rhythm pattern, one that would typically be played by eight virtual hands. It is surprising how easy it is to *program* mesmerizing rhythm patterns of substantial complexity by carefully combining very simple rhythm patterns. The precise execution of these patterns by *human hands* is, of course, not nearly so easy. But this is precisely what Balinese Gamelan musicians excel at. By judiciously playing simple rhythms on traditional instruments these musicians pound out sounds from which compelling rhythms of striking complexity emerge.

The problem that we formulated for ourselves was simply to learn what we could about Balinese Gamelan music in just three class periods, spread out over roughly three weeks. We approached the problem by engaging in three tasks. Task 1 was an audio/visual presentation task. Working in groups, the eight students prepared three twenty-five minute audio/video presentations on Balinese Gamelan Music. The first presentation focussed on the history and conception of Balinese Gamelan music. The second presentation was an introduction to traditional Balinese instruments. The third presentation featured composers who were influence by Balinese Gamelan music – Claude Debussy, Benjamin Britten, and Steve Reisch were selected. The three presentations can be viewed as artifacts with respect to the overall project. Task 2 involved composing a Balinese Gamelan inspired piece in Clay, which should be recognized as another point of contact with the ***constructionist principle***. The piece consisted of a simple foundation to which various combinations of complex rhythms played at various speeds and volumes were added. The foundation was composed of three components played in parallel, a simple pulse, a repeated four note pattern produced on a selection of gongs, and a repeated eight note pattern consisting of just four pitches drawn from an instrument with a deep, rich timbre. The result was pleasantly surprising, and gave us more insight into the sonic fabric of Balinese Gamelan music than would have been achieved by merely listening to it for an hour. This composition is another artifact with respect to the overall project. Task 3 was especially fun. Each student was assigned a part from the Clay composition and asked to practice it on their own using whatever objects they could find that might reasonable reasonably pass for an instrument. On the class day immediately preceding the ESM Gamelan performance, students brought their "instruments" into the classroom and we spent the period trying to perform our Clay composition. How good was the music? Not very. But it was much better than any of us thought it would be. The performance, however lacking in precision, can also be thought of as an artifact with respect to the overall project. Everyone appeared to enjoy the three activities, and no doubt learned something from them. It was clear that engagement this modest project-based learning experience enhanced our anticipatory excitement for the performance, and heightened our enjoyment of the performance put on by the ESM ensemble.

From another perspective, Clay turned out to be useful as a form of *computational scaffolding* for exploring Gamelan music. In this respect, Clay programming reflected the ***learner-centered design principle***. Recall that this principle places emphasis on scaffolding which affords opportunities to enhance understanding by bridging the conceptual distance between a novice and an expert in some domain of interest, thus allowing the learner to more effectively engage in authentic practice. Neither our hands nor our imaginations permitted us to experiment effectively with the process of combining simple rhythm patterns to produce interesting emergent patterns. Clay enabled us to perform such experiments quite easily, thus enabling us to better engage in the *conception* of Gamelan music.

Modeling Melodies as Grouping Structures

Sloboda (2005) aptly observes that making sense of music has often been equated with the process of discovering and representing its structure. One of the most prominent dimensions of melodic structure is grouping structure (Lerdahl & Jackendoff, 1983). Formally, a grouping structure for a melody is a hierarchy that results from recursively partitioning the sequence of notes that makes up the melody into subsequences of notes. The ability to determine sound grouping structure for tonal melodies is a hallmark of an experienced ear. The goal of the computational modeling methodology described in this section is analogous to that of the Russian Formalist approach to literary criticism, namely to force us into a dramatic awareness of phenomena (e.g., music or language) in order to refresh habitual responses and thus render objects more *perceptible* (Eagleton, 1983). This methodology is intended to slow processing in order to afford enhanced opportunities to look for patterns and underlying processes in a manner consistent with the ***deep learning principle***, which, in short, emphasizes engagement over explanation.

Command Definitions

According to Wiggins and Smaill (2000), structural generality is a dimension along which music knowledge representations may be compared that reflects the amount of information about musical structure which can be encoded explicitly. One of the features of Clay that recommends it as a cognitive tool for learning about music is that it is structurally general with respect to both grouping structure and reductional structure. (Two music knowledge representations that are essentially void of structural generality are the machine-oriented MIDI language and the traditional human-oriented score.) Grouping structure is explicitly represented in Clay by means of command definitions. A Clay command definition takes the form *symbol* = *sequence*. For example, `G1 = 2RP P LP P RP P 2LP P` defines a command such that `G1` ⇒ `{/ E \ D / E \ C}`. Braces are used to indicate that a sequence of notes forms a group. As another example, `G2 = RP P LP P RP PL LP` defines a command such that `G2` ⇒ `{/ D \ C / D2}`. Given these two command definitions, `PH1 = G1 G2` defines a command such that `PH1` ⇒ `{{/ E \ D / E \ C}{/ D \ C / D2}}`. As a rule, a nonprimitive Clay command corresponds to a group. This is illustrated in Figure 6.

– Insert Figure 6 here –

The command `PH1` plays the first phrase of Dmitri Kabalevsky's "Little Tune". Figure 7 displays two very different grouping structures for the complete melody. Each of the two grouping structures corresponds to a different Clay program. MxM can instantly render Clay encoded melodies graphically as a score, and can instantly draw spanning trees corresponding to grouping structures. A score is generated automatically when playing a melody in "graphics mode". The spanning tree corresponding to a particularly Clay encoding of a melody is generated by issuing the `-span` metacommand just after the melody has been played.

– Insert Figure 7 here –

Gestalt Principles, Grouping Preference Rules, and Gamma

What is the difference between the two grouping structures shown in Figure 7? Principally, the one on the top is a very sound model of how the piece would be heard by a "good ear", and the one on the bottom is a rather unsound model of how the melody should be heard. This observation can be understood theoretically in terms of the Gestalt principles, which constitute a set of rules according to which perceptual phenomena are organized (Wertheimer, 1939), and Grouping Preference Rules (GPRs), a manifestation of the Gestalt principles that is oriented towards tonal melody (Lerdahl & Jackendoff, 1983). The Gestalt principles take into consideration *proximity*, as illustrated in Figure 8, *similarity*, as illustrated in Figure 9, symmetry, "regularity", and other phenomena.

– Insert Figure 8 here –

– Insert Figure 9 here –

Lehrdahl and Jackendoff defined seven grouping preference rules, which correspond in rough fashion to the Gestalt principles. GPR2 and GPR3 correspond in a direct manner to proximity and similarity. The remainder of the mapping is not so direct.

MxM provides a metric called “Gamma” (Graci, 2008b) in the form of a metacommand that is intended to measure the quality of a grouping structure on the basis of Lerdahl’s and Jackendoff’s grouping preference rules. Gamma maps a grouping structure to a value between 0 and 1. As a rule, the higher the Gamma value for a particularly melody, the better the grouping structure is for that melody. With respect to the sample grouping structures shown in Figure 7, the gamma value for the sound grouping structure is 0.738, while the gamma value for the unsound grouping structure is 0.590. Figure 10 suggests how scoring, spanning, and the gamma computation are accomplished in MxM, as yet another grouping structure for Little Tune is presented. While this grouping structure correctly captures surface level phrasing, it lacks “depth” – and thus fails to recognize both repeated patterns below the phrase level and essential partitions above the phrase level.

– Insert Figure 10 here –

The “extended” form of the Gamma metric, “Gammax”, which is illustrated in Figure 10, provides information on the individual GPR contributions to the Gamma value. Gamma is defined as a linear combination of GPR factors. That is, $\gamma = \omega_1\gamma_1 + \omega_2\gamma_2 + \omega_3\gamma_3 + \omega_4\gamma_4 + \omega_5\gamma_5 + \omega_6\gamma_6$, where γ_i is the GPR_i factor, a function mapping the structural interpretation of the melody onto a real number between 0 and 1 that indicates the degree to which the interpretation is compatible with GPR_i, and the ω_i are weights, real numbers which sum to 1.0.

Generative Explorations of Grouping Structures

Modeling grouping structure with Gestalt principles in mind can be viewed as one element in a program of learning intended to develop sensitivity to melodic structure. Two distinct types of problem can be posed in this regard, both of which typify the ***constructionist principle***. In the “build” type of problem, a reasonable grouping structure is given for a particular melody in terms of a spanning tree, and the learner is charged with writing a Clay program consistent with the given structure. In the “design” type of problem, a melody is presented and the learner is charged with writing a Clay program to model the melody in a manner that represents a good ear. The first type of problem presents a well-defined puzzle for the learner. The second type of problem opens the door to a variety of solutions, and consequently to debate on the quality of solutions. As with all design problems, stability (e.g., harmonic correctness), function (e.g., memorability), and aesthetics are important considerations for the evaluation of a grouping structure.

For the design problems, Gamma is a useful cognitive tool for helping to inform intuitions about how to model a melody. Gamma serves as a design tool by providing feedback on how well a proposed design conforms to the grouping preference rules. Gamma can be viewed as a metric for assessing the quality of a solution in the design space of all grouping structures for a given melody. As a good grouping structure is sought, the ***distributed cognition principle*** is present in interactions between the learner and MxM. By virtue of the graphics facilities (scoring and spanning) and the Gamma metric, MxM enables people to develop better models of grouping structure than they would be able to do on their own (Graci, 2008b). In other words, MxM serves as a cognitive artifact with respect to modeling the grouping structure of a melody (Norman, 1991).

Grouping Structures as Artifacts

The computer is central to the learning sciences in large part because the computer can represent abstract knowledge in concrete verbal, visual, or sonic form. In essence, the computer is an artifact creation device, transforming, according to programmer specification or user interaction, abstract knowledge into concrete artifacts. Moreover, as Sawyer (2006b) points out, "Computers can allow learners to manipulate and revise their developing knowledge via the user interface, in a complex process of design that supports simultaneous articulation, reflection, and learning." This means that the computer can be regarded as instrumental to educational experiences grounded in the ***deep learning principle***. Engaging in the active, cyclic process of coding a Clay model of grouping structure for a melody and then reflecting on it both visually with `-span` and analytically with `-gamma` and `-gammax` is more likely to produce deep learning than is merely listening to others describe how they would group a melody. In this use of MxM, the program as artifact is central to the deep learning experience.

The computer can facilitate the incremental development of artifacts that reflect learning over time. Artifacts are significant not only because they serve as records of accomplishment, but also because they can be inspected and analyzed by their creators and by others interested in the knowledge they represent and the generative processes by which they were constructed. The ***project-based learning principle*** states that students learn more effectively when they develop artifacts – external representations of their constructed knowledge (Sawyer, 2006b). A Clay program that models the grouping structure of a melody is an artifact. So is the spanning tree corresponding to the Clay program. The former is an easily manipulable verbal artifact in MxM. The latter is an easily rendered visual artifact in MxM. The ***imagery principle*** implies that these spanning trees hold potential as learning objects. This has been shown to be the case with respect to modeling grouping structure. The spanning trees prove useful in both types of problem described above, the "build" type and the "design" type. The juxtaposition of spanning trees with scores taps into the inherent power of visualization, summarized by Schwartz and Heiser (2006) in terms of qualities of perception, including effortless structural interpretation and determinism. It is quite difficult in natural language to describe a grouping structure for even a very short, very simple melody. It isn't so hard in Clay, but perceiving the grouping structure by looking at the Clay code is far from a satisfying experience. Looking at the spanning tree underlying a melody, on the contrary, is easy, and immediately rewarding. This is because, as Schwartz and Heiser (2006) put it "perception packages sensation with little discernable effort, because evolution has conferred specialized abilities that are well matched to recurrent structures in the spatial world." Trying to comprehend a grouping structure in terms of natural language is rather like trying to understand complex directions on how to get to an unknown destination in a foreign city from a native you stop sidewalk. Trying to comprehend a grouping structure by listening to a melody is somewhat like trying to create a mental map of a city by driving around in circles. Comprehending a grouping structure by looking at a spanning tree, on the other hand, is effortless and meaningful.

A spanning tree, being a visual object, is deterministic. Regardless of whether or not it corresponds to a reasonable grouping structure for a melody, you can at least appreciate just which grouping structure it represents. As has been noted, the same cannot be said, in general, for descriptions of grouping structures in a natural language, which tend to be unwieldy, at best, and ambiguous, at worst. It is useful to be able to focus on a particular grouping structure, if only for the purpose of contrasting it with other grouping structures. Interestingly, it is just this sort of contrast that Gibson and Gibson (1955) had in mind when they argued that juxtaposing contrasting cases will increase discernment (pickup) of information, which in turn will improve the ability to perceive information. Using wine tasting as an example, they argued that by comparing cases that are similar in many ways, people can begin to notice what makes the cases distinctive. As Schwartz and Heiser (2006) observe, "It is an important lesson for educators that what they perceive may not be the same thing that their students perceive (Nathan & Koedinger, 2000), and it takes special strategies, like contrasting cases, to help students see what is important."

Modeling Melodies as Reductional Structures

Deutsch and Feroe (1981) commence their seminal paper on the internal representation of pitch sequences in tonal music by recalling a fact that helped to launch the cognitive revolution: "It may generally be stated that we tend to encode and retain information in the form of hierarchies when given the opportunity to do so." Unfortunately, the significance of this observation may be underappreciated due to the fact that the hierarchies are generally hidden from view in the form of implicit, rather than explicit, knowledge. The elegant mathematical formalism proposed by Deutsch and Feroe for representing pitch sequences in tonal melody makes reductional structure explicit. The computational formalism discussed in this section animates the modeling of reductional structure in a manner that is consistent with the conceptual apparatus proposed by Deutsch and Feroe. As a result of embedding a procedural variant of the Deutsch/Feroe formalism in an interpretive framework (Clay/MxM), learners have an opportunity to play with essential conceptual ideas (e.g., *alphabets* and *operators*), and to get acquainted with them in an environment that provides immediate feedback. By rendering this feedback in a focussed visual form the learner can take advantage of the qualities of perception associated with the ***imagery principle*** for an enhanced learning experience.

Reduction Definitions and Reduction Operators

A reductional structure is essentially a hierarchical record of the derivation of a melodic fragment in terms of reductional operators that capture important theoretical and cognitive elements of the melody. Consider Figure 11, a sixteen note pitch sequence taken from Beethoven's Sonata, op. 22. This melodic fragment was used by Deutsch and Feroe to illustrate the representation of reductional structure by means of their notational conventions.

– Insert Figure 11 here –

They observed that four notes, those highlighted in Figure 11, are more significant than the rest with respect to reductional structure. From these four notes the sixteen note sequence can be derived in two simple steps. First, double each note. Second, replace each note by its chromatic predecessor. Remembering the sequence according to this strategy simply requires recall of the D7 chord (highlighted notes) followed by application of the two simple steps of refinement. Figure 12 illustrates the basic idea using notes with colored heads, where a solid red interior is a "level 1" note, a solid blue interior is a "level 2" note, and a solid green interior is a "level three" note [1].

– Insert Figure 12 here –

A more explicit representation of this reductional hierarchy is shown in Figure 13. The relatively elaborate color coding of the note heads indicates, with respect to reductional structure, which notes which are the most significant (green surrounded by blue surrounded by read), which are of secondary significance (green surrounded by blue), and which are the least significant (green).

– Insert Figure 13 here –

Reductional structure is explicitly represented in Clay by means of reduction definitions and reduction operators. A reduction definition takes the form *symbol* -> *sequence*. For example, `RAMP -> 4P+RP 4LP` defines a reduction such that `RAMP ⇒ (C) / (D) / (E) / (F)`, and `STEP -> LP P RP P` defines a reduction such that `STEP ⇒ \ (B) / (C)`. Textually speaking, for any note N, (N) means that note N is played at "level 1".

A reductional operator takes the form S:X->Y, where X is any symbol, and Y is a reduction symbol (i.e., Y corresponds to a reduction definition). This form creates an unnamed reduction definition, the sequence of which is the sequence of S with all occurrences of X replaced by Y. This is fundamentally a Post production operator (Post, 1943). For example `RAMP:P->STEP` would create `π -> 4STEP+RP 4LP` so that `π ⇒ \ (-B-) / (-C-) (-C-) / (-D-) (-D-) / (-E-) (-E-) / (-F-)` and, in effect, `RAMP:P->STEP ⇒ \ (-B-) / (-C-) (-C-) / (-D-) (-D-) / (-E-) (-E-) / (-F-)`. For any note N, (-N-) means that note N is played at "level 2".

---

[1] When reading a black and white version of this paper the different shades of grey to which the colors are mapped will suffice to convey the main points being made.

At this point is is clear how reduction definitions and reduction operators produce note sequences. But how are connections made between levels? This is accomplished by marking *progenitor* notes at different levels of the reductional hierarchy. A progenitor note is a structurally important note, pivotal in the sense that it belongs, conceptually, to two adjacent levels in the reductional hierarchy. Suppose that `STEP' -> LP P RP P!`. The “exclamation mark” is used to indicate that the play command is a progenitor note. Textually, progenitor notes can be spotted in the output stream as the notes wrapped in nested parentheses. A set of parentheses marks a note as belonging to level *i* if the parentheses are *i* characters from the textual representation of the note. Thus `RAMP:P->STEP'` ⇒ `\ (-B-) / ((C)) (-C-) / ((D)) (-D-) / ((E)) (-E-) / ((F))`. In this reductional model of the eight note sequence the even numbered notes (belonging to both level 1 and level 2) are seen to be structurally more important that the odd numbered notes (belonging only to level 2).

As has already been shown, a special font, the “reductional font”, is used in MxM to score melodies that are modeled as reductions in Clay. In this colorful font the interior of each note head indicates the level of the note with respect to the reductional hierarchy. It is noteworthy that the diameters of note interiors differ according to the level of the note. For example, the diameter of the red interior of a level 1 note is larger than the diameter of the blue interior of a level 2 note. This being the case, a level 1 progenitor of a level 2 note is be indicated by an interior consisting of blue dot superimposed on a red dot. Figure 14 illustrates these conventions by rendering the preceding examples graphically. The left image of Figure 14 was generated with the Clay expression `RAMP:P->STEP`. The right image of Figure 14 was generated with the Clay expression `RAMP:P->STEP'` Although details of reductional structure are not explicit in the image on the right, the multilevel salience signifiers (multicolored note heads) nonetheless prove to be helpful the reductional interpretation of a score.

– Insert Figure 14 here –

Sometimes it is theoretically satisfying to indicate progenitor notes all the way back to a single note. For example, suppose that we add `RAMP' -> P! 3RP+P 3LP` and `LINE -> RAMP':P->STEP'` to the set of reductions already defined. Then entering `LINE` into the MxM text input box would produce the score shown in Figure 15. The interpretation is that the red note generates the blue notes and the blue notes each generate two green notes.

– Insert Figure 15 here –

Combining Alphabets with Reductions

As Deutsch and Feroe (1981) made so compellingly clear, coupling alphabets with reduction operators is a very powerful idea. One of the examples presented in their classic paper involves the internal representation of the pitch sequence shown, twice, in Figure 16. Musicians presumably look at the sequence on the left and "see" something like that on the right. How might this reductional structure be modeled in Clay? The following three commands will accomplish the task: `S -> A:P->B` and `A -> C-MAJTRIAD P! 3RP+P 3LP` and `B -> CHROMATIC LP P RP P!`. It is semantically imperative to note that the scope of an alphabetic command in Clay extends from its point of introduction to the end of the command definition in which it appears, except for times during which other alphabetic commands intervene. It is on the basis of this scoping rule that execution flows appropriately through alphabetic environments. For example, running the command `S` will produce: `[ C-MAJTRIAD [ CHROMATIC LP P RP P ] RP [ CHROMATIC LP P RP P ] RP [ CHROMATIC LP P RP P ] RP [ CHROMATIC LP P RP P ] 3LP ]`. The three `RP` commands within the scope of the `C-MAJTRIAD` alphabet result in changes of a major third, a minor third, and a perfect fourth. Each of the other pitch changing commands, which lie within a `CHROMATIC` alphabet, result in a half-step change.

– Insert Figure 16 here –

Proper pitch sequences derive from the essential fact that the behavior of pitch changing commands is sensitive to alphabetic environments. To properly render the notes of a melody, Clay makes use of local alphabetic information stored in the note.

Inscription Languages

Within a distributed cognition framework, no matter what the nature of its constituent agents, ideas must be represented in *inscriptional systems*, that is, in symbolic knowledge representation languages that are external to the individuals. With respect to an individual operating in a distributed learning environment, information transfer both *to* and *from* inscriptional texts can, according to the ***distributed cognition principle***, benefit learning. In the "to" direction, Bell and Winn (2000) state that "Students can be encouraged to make their own thinking visible through inscriptional systems, exchange perspectives through debate-focused participant structures, and work toward a more integrated understanding of an issue through the application of more scientific criteria that are highlighted by the tools being used." In the "from" direction, they claim that "Individuals might also internalize aspects of these artifacts as they make use of them and thereby the use of an artifact can leave some sort of cognitive residue with the individual." While it is widely recognized that inscriptional systems have made important contributions to the development of cultural knowledge, it is also clear that they often pose substantial problems for learners. Why? Pea (1993) puts it this way: "A person has to have been introduced to, and preferably to have participated in, the activities that give meaning to these inscriptions. After such initiations, one may have the sense of directly perceiving the patterns the inscriptional system was designed to make "obvious," but before such initiation, the conventions and uses of the inscriptions are usually obtuse."

The elegant mathematical formalism proposed by Deutsch and Feroe (1981) is an inscription system. So is Clay. In fact, the Deutsch/Feroe work not only motivated the Clay implementation of mechanisms for capturing reductional structure, but it significantly influenced its design. Inscriptional texts in these two inscription languages for modeling the reductional structure of melody look quite different. Table 1 presents examples that Deutsch and Feroe used to illustrate aspects of their formal system pertaining to alphabets and the generation of simple note sequences along side corresponding Clay code.

– Insert Table 1 here –

Table 2 presents Clay code and the corresponding Deutsch/Feroe expressions for refining the higher level melodic line ⟨ C / E / G / C ⟩ into a lower level melodic line ⟨ \ B / C / D# / E / F# / G / B / C ⟩. In this example each pitch in the higher level sequence is refined into the two note pattern consisting of the chromatic predecessor of the pitch followed by the pitch. It is noteworthy that in Clay the refinement is conceptually accomplished by means of a "production" while in Deutsch/Feroe the refinement is conceptually accomplished by means of a "pattern application". The idea in Clay is that "P" is replaced by "B" in "A" (A:P->B, in Table 2). The idea in Deutsch/Feroe is that the pattern denoted by "B" is applied throughout "A" in a systematic fashion based on "*" (A[pr]B, in Table 2).

– Insert Table 2 here –

Each of these inscription systems has its advantages. On the one hand, the Deutsch/Feroe system is particularly elegant, and may well be favored by those who are well versed in music theory and ideas associated with reductional structure, particularly if they are mathematically sophisticated. On the other hand, a number of the learning science principles would collectively recommend the Clay system for students who require some initiation with respect to music theory and ideas associated with reductional structure, particularly if they have little experience with formal modeling methodologies. Pea's observations, quoted above, would appear to bolster this sentiment. Moreover, the ***learner-centered design principle***, which is in large measure a call for the construction of scaffolding to support novices in the performance of activity beyond their level of ability, is evident in the Clay mechanism for modeling reductional structure.

Building reductional structures.

Building sound reductional structures with respect to principles of tonal music theory can be viewed as another element in a program of learning intended to develop sensitivity to melodic structure. As is the case with the study of grouping structure, two distinct types of problem can be posed, both of which typify the ***constructionist principle***. In the “build” type of problem, a reasonable reductional structure is given for a particular melody in terms of a colored score, and the learner is charged with writing a Clay program corresponding to the given structure. The note sequence shown in Figure 13 would form the basis of a more or less challenging reductional structure building problem. The more information provided to the student, the less challenging the problem will be. Providing no information would be reasonable only for students who are music theory savvy. For those students who are still learning music theory, it might be suggested that they focus on the dominent seventh chord (D7). In the “design” type of problem, a melody is presented to the learner who is charged with writing a Clay program to model the melody in a way that represents a good reductional model of the melody. The pitch sequence in Figure 17 could be used as the basis of a reductional design problem.

– Insert Figure 17 here –

The first type of problem presents a well-defined challenge. The second type of problem opens the door to a variety of solutions, and consequently to debate on the quality of solutions.

It is worth noting that both types of problem also illustrate the significance of the ***inscription principle***. The learner will improve their thinking about melodic reduction as they improve their computational models of melodic reduction. More specifically, by writing down reductional descriptions in Clay the learner can improve reductional understanding by running programs, reflecting upon reductional descriptions in light of program execution, and revising models accordingly. Over time, according to this principle, the learner’s thinking about reductional modeling will become stronger, more creative, and more correct.

Themes and Variations

Reduction and refinement are inverse processes. The same Clay mechanisms used to model reductional structure in melody can be used to generate different refinements of melody, or variations on a theme. For instance, consider the theme presented in Figure 18, a theme on which Mozart wrote a number of variations.

– Insert Figure 18 here –

A Clay reduction definition for this melodic line might be `THEME -> P 4RP P RP P LP P LP P LP P LP P LP P`. Figure 19 presents a variation on the theme, one which many of us know from childhood. How might this be accomplished in Clay by means of reduction? Each of the notes except for the last note will be transformed. To reflect this fact one could rewrite the theme as `THEME -> V 4RP V RP V LP V LP V LP V LP V LP P`. Each note is transformed into a pair of notes, so the reduction `PAIR -> P! P` would be an appropriate refinement resource. The variation could then be written in terms of these reductions as `VAR1 >> THEME:V->PAIR`.

– Insert Figure 19 here –

It is an easy matter to generate lots of simple variations on the theme by simply replacing `PAIR` with another reduction definition. Rather more sophisticated variations may be achieved in a similar manner by increasing the number of subvariations within the variation. Figure 20, is, for the most part, a variation penned by Mozart himself. This variation may be achieved by six reductions, a main reduction and five subreductions. The main reduction `VAR2 -> P1 4RP P2 RP P2 LP P2 LP P3 LP P3 LP P4 LP P5` will maintain the integrity of the theme. The first subvariation could be written as `P1 -> PS! 7LP PS 2RP PS 5RP PS`. The remaining four are of about the same degree of complexity.

This engagement in composing variations on a theme can be viewed as a manifestation of the ***learner-centered design principle***. The reductional structure modeling mechanism in Clay helps the learner to focus on the theme and to reflect on the variations. The sonic and visual renderings of inscribed thoughts also help the learner to monitor their thoughts in a manner that informs progress. In short, Clay can be seen as a form of scaffolding that helps the learner to cross the gap from novice theme-and-variation writer to a more capable theme-and-variation writer.

– Insert Figure 20 here –

Conclusion

In discussing the development of thought in human history, Wolf (2007) writes: “As the twentieth-century Russian psychologist Lev Vygotsky said, the act of putting spoken words and unspoken thoughts into written words releases and, in the process, changes the thoughts themselves. As humans learned to use written language more and more precisely to convey their thoughts, their capacity for abstract thought and novel ideas accelerated.” Vygotsky’s understanding of the importance of writing in the processes of learning and development (Vygotsky, 1962) provides a foundation for the ***inscription principle***, which underlies nearly everything discussed in this paper. Inscription systems that are machine executable can facilitate the construction and visualization of artifacts within a distributed cognition framework, and consequently hold potential for leveraging the other learning science principles in the service of education. In this paper I have provided a variety of evidence that demonstrates this from a musical perspective.

References

Bell, P., & Winn, W. (2000). Distributed learning, by nature and by design. In D. Jonassen & S. Land (Eds.), *Theoretical foundations of learning environments*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bransford, J., Brown, A., & Cocking, R. (2000). *How people learn: Brain, mind, experience, and school*. Washington, DC: National Academy Press.

Cole, M., & Engestrom, Y. (1993). A cultural-historical approach to distributed cognition. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations*. New York: Cambridge University Press.

Danesi, M. (1998). *Sign, thought, and culture*. Toronto, Ontario: Canadian Scholars' Press.

Daniels, H., Cole, M., & Wertsch, J. (2006). *The Cambridge companion to Vygotsky*. New York: Cambridge.

Deutsch, D., & Feroe, J. (1981). The internal representation of pitch sequences in tonal music. *Psychological Review*, *88*, 503–522.

Eagleton, T. (1983). *Literary theory*. Minneapolis: University of Minnesota Press.

Gibson, J., & Gibson, E. (1955). Perceptual learning: Differentiation or enrichment. *Psychological Review*, *62*, 32–41.

Graci, C. (2008a). *A genetic programming approach to determining grouping structure in tonal melody*. Paper presented at the ESM/Cornell/UR Music Cognition Symposium Series, Rochester, New York.

Graci, C. (2008b). *A quantitative measure of melodic structure: Computational infrastructure and cognitive implications*. Paper presented at the Empirical Musicology Conference, London.

Graci, C. (2009). *A procedural take on the Deutsch/Feroe formalism: Cognitive motivation and computational realization*. Paper accepted for presentation at the Society for Music Perception and Cognition (SMPC) Conference, Indianapolis, Indiana.

Graci, C., Narayan, J., & Odendahl, R. (1989). Bunny numerics: A number theory microworld. In E. Kaltofen & S. Watt (Eds.), *Computers and mathematics*. New York: Springer-Verlag.

Graci, C., Narayan, J., & Odendahl, R. (1992). Children, chunking, and computing. *Journal of Computing in Childhood Education*, *3*, 247–258.

Jonassen, D., & Carr, C. (2000). Mindtools: Affording multiple knowledge representatins for learning. In S. Lajoie (Ed.), *Computers as cognitive tools*. Mahwah, NJ: Lawrence Erlbaum Associates.

Kafai, Y. (2006). Constructionism. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Kim, B., & Reeves, T. (2007). Reframing research on learning with technology: In search of the meaning of cognitive tools. *Instructional Science*, *35*, 207–256.

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: The MIT Press.

Morgan, M., Brickell, G., & Harper, B. (2008). Applying distributed cognition theory to the redesign of the 'copy and paste' function in order to promote appropriate learning outcomes. *Computers & Education*, *50*, 125–147.

Nathan, M., & Koedinger, K. (2000). An investigation of teachers' beliefs of students' algebra development. *Cognition and Instruction*, *18*, 209–237.

Norman, D. (1991). Cognitive artifacts. In J. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface*. Cambridge: Cambridge University Press.

Papert, S. (1980). *Mindstorms: Children, computing, and powerful ideas*. New York: Basic Books.

Papert, S. (2006). Afterword: After how comes what. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Pea, R. (1993). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations*. New York: Cambridge University Press.

Post, E. (1943). Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, *52*, 264–268.

Quintana, C., Shin, N., Norris, C., & Soloway, E. (2006). Learner-centered design. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Salomon, G. (1993). *Distributed cognitions: Psychological and educational considerations*. New York: Cambridge University Press.

Sawyer, R. (2006a). *The cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Sawyer, R. (2006b). The new science of learning. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Schwartz, D., & Heiser, J. (2006). Spatial representations and imagery in learning. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*. Cambridge: Cambridge University Press.

Sivasubramaniam, P. (2004). Distributed cognition and the use of graphing calculators in the learning of mathematics. In *Proceedings of the 2nd national conference on graphing calculators* (pp. 93–103). Penang, Maylasia: NCGC 2004.

Sloboda, J. (2005). *Exploring the musical mind*. New York: Oxford University Press.

Veer, R. V. D. (2007). Vygotsky in context. In H. D. nd M. Cole & J. Wertsch (Eds.), *The cambridge companion to vygotsky*. Cambridge: Cambridge University Press.

Vygotsky, L. (1962). *Thought and language*. Cambridge, MA: MIT Press.

Vygotsky, L. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.

Wertheimer, M. (1939). Laws of organization in perceptual forms. In W. D. Ellis (Ed.), *A source book of gestalt psychology*. New York: Harcourt, Brace & Company.

Whitehead, A. (2002). *Dialogues of Alfred North Whitehead*. Boston: Godine.

Wiggins, G., & Smaill, A. (2000). Musical knowledge and AI. In E. Miranda (Ed.), *Readings in music and artificial intelligence*. Amsterdam: Harwood.

Wolf, M. (2007). *Proust and the squid*. New York: Harper.

Table 1

Comparison of Clay Code and Deutsch/Feroe Expressions for the Generation of Five Note Ascending Pitch Sequences in Different Alphabetic Contexts

Clay	Deutsch/Feroe	Sequence of Notes
C-MAJOR(1) 5P+RP	{{(*,4n);C}c}	⟨ C / D / E / F / G ⟩
C-MAJTRIAD(1) 5P+RP	{{(*,4n);C_{tr}}c}	⟨ C / E / G / E / C ⟩
C-CHROMATIC(1) 5P+RP	{{(*,4n);Cr}c}	⟨ C / C# / D / D# / E ⟩
C-MAJOR(3) 5P+RP	{{(*,4n);C}e}	⟨ E / F / G / A / B ⟩
C-MAJTRIAD(3) 5P+RP	{{(*,4n);C_{tr}}e}	⟨ E / G / C / E / G ⟩
C-CHROMATIC(3) 5P+RP	{{(*,4n);Cr}e}	⟨ E / E# / F / F# / G ⟩

Table 2

Contrasting the Post Production Mechanism in Clay with Deutsch/Feroe Pattern Application using Inscriptions that Capture the Reduction Illustrated in Figure 16

Clay	Deutsch/Feroe
`S -> C-MAJOR A:P->B`	`S = {A[pr]B;1}C`
`A -> C-MAJTRIAD P! 3RP+P 3LP`	`A = {(*,3n);I}`
`B -> CHROMATIC LP P RP P!`	`B = {(p,*);Cr}`

Figure Caption

Figure 1. Sample MxM frame in which metacommands are issued to to establish a level 1 puzzle (line 1), present the puzzle (line 2), display the solution (line 10), and check the solution (line 15). Also, a solution is input (lines 4 through 7) and displayed (line 8) by the user.

Figure 2. A simple melodic figure is defined (line 1) and then played in the context of a number of different pitch alphabets.

Figure 3. Three solutions to a level 2 puzzle (based on the Rodgers and Hart tune "My Funny Valentine"): a theoretically incorrect solution (line 4); a theoretically correct but conceptually incorrect solution (line 9); and a good solution (line 13).

Figure 4. Example pattern production task (based on Robert Schumann's "Melody"): a rough pattern is established, specified, coded by the learner (presumably, as indicated by the ellipsis), played, and displayed.

Figure 5. Two pitched rhythm patters (top and middle) are combined to produce a more complex rhythm pattern (bottom) in imitation of Balinese Gamelan musicians. Graphical representation of the rhythm patterns appear on the left. Clay coding of the rhythm patterns appear on the right.

Figure 6. Clay command definitions correspond to groups. This is one manifestation of "structural generality" that is manifested in Clay.

Figure 7. Two grouping structures for Dmitri Kabalevsky's "Little Tune". The top grouping structure is very sound ($\gamma = 0.738$). The bottom grouping structure is not so sound ($\gamma = 0.590$).

Figure 8. Illustration of the Gestalt principle of proximity. In the visual realm, notes that are spatially close tend to be grouped together. In melody, notes that are temporally close tend to be grouped together.

Figure 9. Illustration of the Gestalt principle of similarity. In the visual, realm properties such as size and color are the basis for grouping similar shapes. In melody, properties such as pitch and duration serve as the basis for grouping similar notes.

Figure 10. Example interaction with respect to grouping structure in MxM. A program that was written for "Little Tune" is run in line 1. Metacommands are issued to generate the spanning tree (line 2), to display the program (line 3), and to compute gamma in its pure and extended forms (lines 9 and 11).

Figure 11. Four of sixteen notes, the highlighted notes, which constitute a D7 chord, are more important than the remainder of the notes with respect to reductional structure in this pitch sequence adapted from Beethoven's Sonata, op. 22.

Figure 12. Derivation of sixteen notes from four notes in two steps: first, double each note (top to middle); second,

prepend the chromatic predecessor to each note (middle to bottom).

Figure 13. Example illustrating the representation of note salience with respect to reductional structure via Clay's "reduction font". Generally speaking, the more colorful the interior of a note, the greater its reductional salience.

Figure 14. Graphical representation of a two level reduction with (right) and without (left) inclusion of the progenitor note concept.

Figure 15. Full reduction of an eight note sequence: `C` derives `C / D / E / F` which derives `\ B / C C / D D / E E / F`.

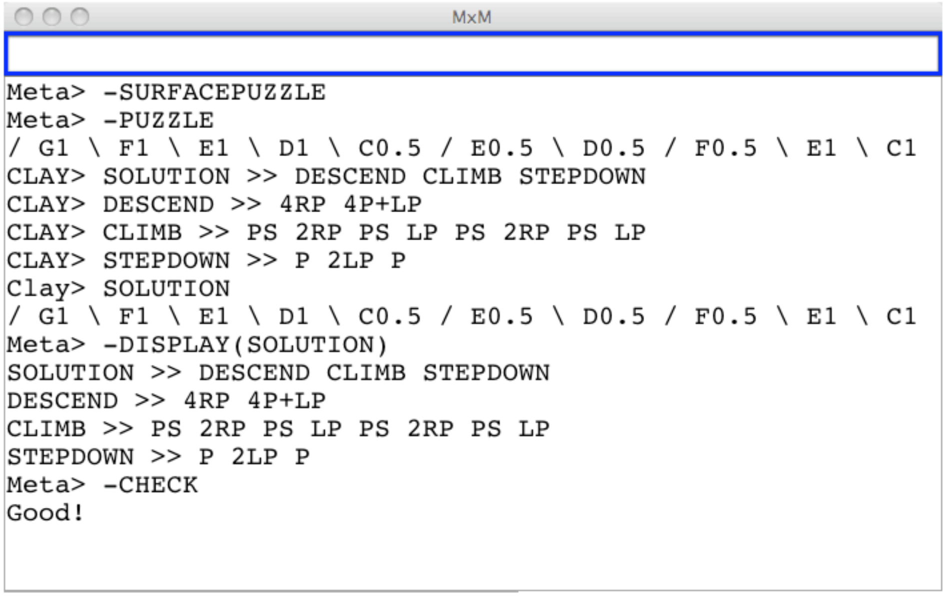
Figure 16. The traditional score of an eight note sequence (left) and a reductional enhancement of the traditional score right).

Figure 17. Example problem for the design of a reductional structure: the learner is tasked with computationally coloring a given melodic line in a manner that reflects a proper structural reduction.

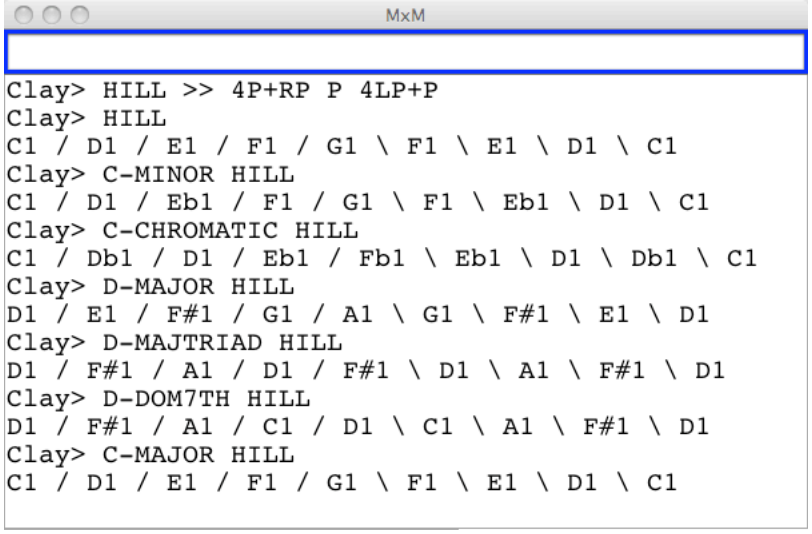
Figure 18. A simple theme (penned by Mozart) is modeled in Clay as a level 1 reduction.

Figure 19. A simple variation on the theme presented in Figure 18 that has become a standard part of "small children's repertoire" in the Western world.

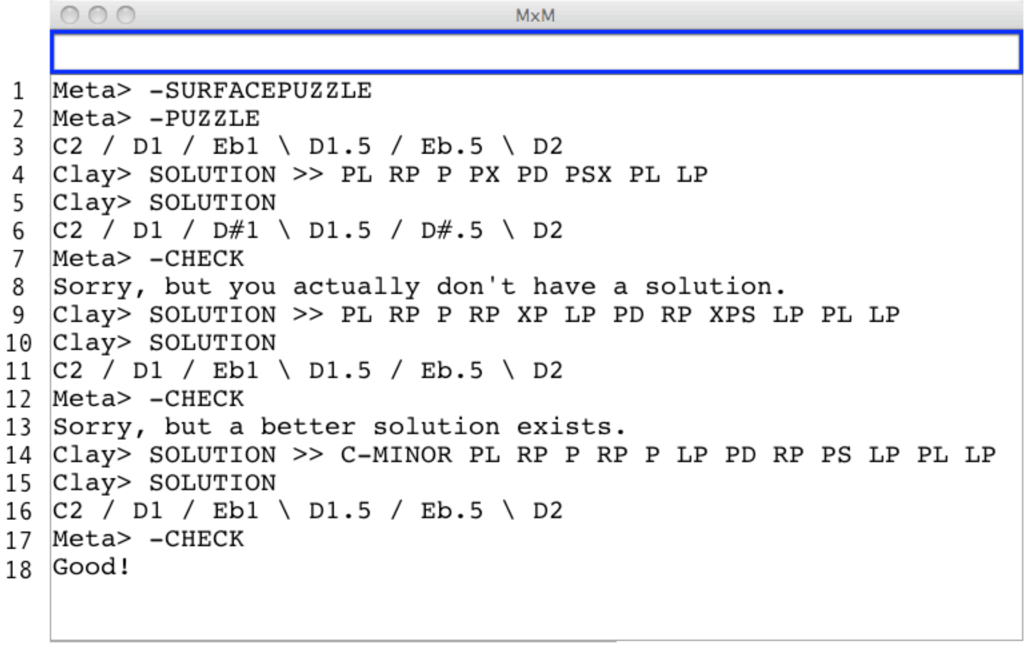
Figure 20. A variation on the theme presented in Figure 18, penned, for the most part, by Mozart himself, which suggests the flexibility of the reductional approach to varying a theme.



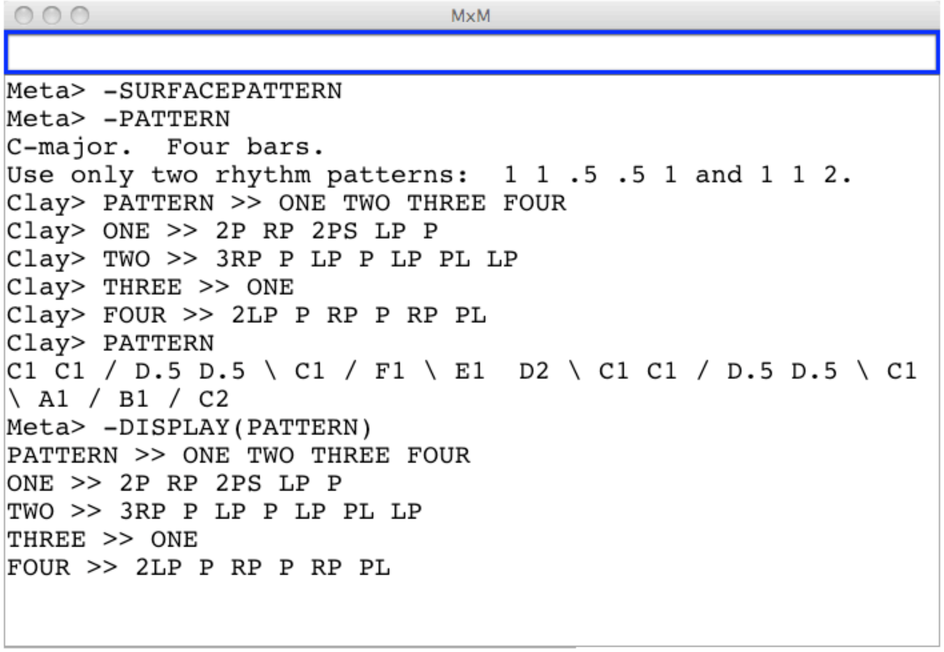
```
Meta> -SURFACEPUZZLE
Meta> -PUZZLE
/ G1 \ F1 \ E1 \ D1 \ C0.5 / E0.5 \ D0.5 / F0.5 \ E1 \ C1
CLAY> SOLUTION >> DESCEND CLIMB STEPDOWN
CLAY> DESCEND >> 4RP 4P+LP
CLAY> CLIMB >> PS 2RP PS LP PS 2RP PS LP
CLAY> STEPDOWN >> P 2LP P
Clay> SOLUTION
/ G1 \ F1 \ E1 \ D1 \ C0.5 / E0.5 \ D0.5 / F0.5 \ E1 \ C1
Meta> -DISPLAY(SOLUTION)
SOLUTION >> DESCEND CLIMB STEPDOWN
DESCEND >> 4RP 4P+LP
CLIMB >> PS 2RP PS LP PS 2RP PS LP
STEPDOWN >> P 2LP P
Meta> -CHECK
Good!
```



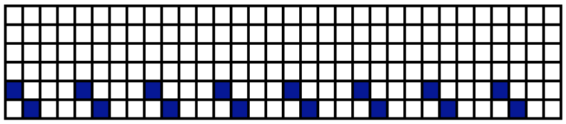
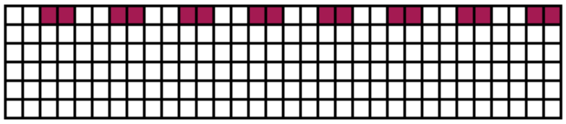
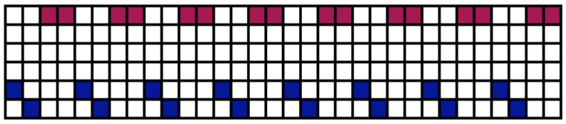
```
Clay> HILL >> 4P+RP P 4LP+P
Clay> HILL
C1 / D1 / E1 / F1 / G1 \ F1 \ E1 \ D1 \ C1
Clay> C-MINOR HILL
C1 / D1 / Eb1 / F1 / G1 \ F1 \ Eb1 \ D1 \ C1
Clay> C-CHROMATIC HILL
C1 / Db1 / D1 / Eb1 / Fb1 \ Eb1 \ D1 \ Db1 \ C1
Clay> D-MAJOR HILL
D1 / E1 / F#1 / G1 / A1 \ G1 \ F#1 \ E1 \ D1
Clay> D-MAJTRIAD HILL
D1 / F#1 / A1 / D1 / F#1 \ D1 \ A1 \ F#1 \ D1
Clay> D-DOM7TH HILL
D1 / F#1 / A1 / C1 / D1 \ C1 \ A1 \ F#1 \ D1
Clay> C-MAJOR HILL
C1 / D1 / E1 / F1 / G1 \ F1 \ E1 \ D1 \ C1
```



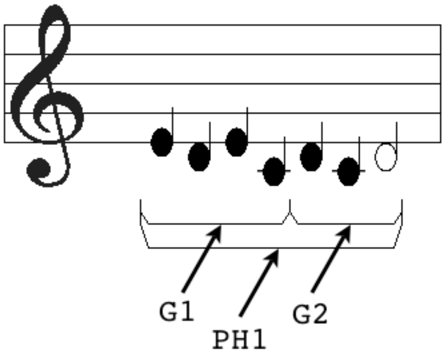
```
Meta> -SURFACEPUZZLE
Meta> -PUZZLE
C2 / D1 / Eb1 \ D1.5 / Eb.5 \ D2
Clay> SOLUTION >> PL RP P PX PD PSX PL LP
Clay> SOLUTION
C2 / D1 / D#1 \ D1.5 / D#.5 \ D2
Meta> -CHECK
Sorry, but you actually don't have a solution.
Clay> SOLUTION >> PL RP P RP XP LP PD RP XPS LP PL LP
Clay> SOLUTION
C2 / D1 / Eb1 \ D1.5 / Eb.5 \ D2
Meta> -CHECK
Sorry, but a better solution exists.
Clay> SOLUTION >> C-MINOR PL RP P RP P LP PD RP PS LP PL LP
Clay> SOLUTION
C2 / D1 / Eb1 \ D1.5 / Eb.5 \ D2
Meta> -CHECK
Good!
```

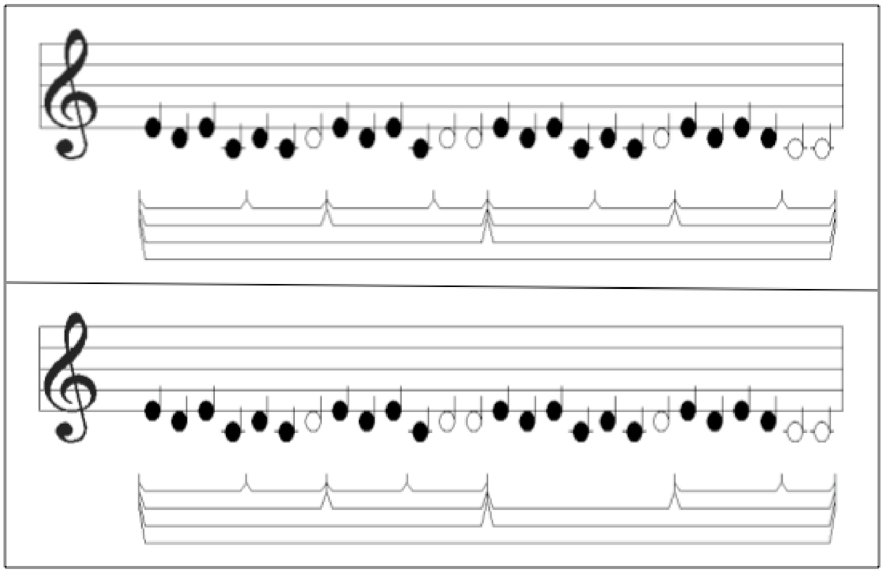


```
Meta> -SURFACEPATTERN
Meta> -PATTERN
C-major.  Four bars.
Use only two rhythm patterns:  1 1 .5 .5 1 and 1 1 2.
Clay> PATTERN >> ONE TWO THREE FOUR
Clay> ONE >> 2P RP 2PS LP P
Clay> TWO >> 3RP P LP P LP PL LP
Clay> THREE >> ONE
Clay> FOUR >> 2LP P RP P RP PL
Clay> PATTERN
C1 C1 / D.5 D.5 \ C1 / F1 \ E1  D2 \ C1 C1 / D.5 D.5 \ C1
\ A1 / B1 / C2
Meta> -DISPLAY(PATTERN)
PATTERN >> ONE TWO THREE FOUR
ONE >> 2P RP 2PS LP P
TWO >> 3RP P LP P LP PL LP
THREE >> ONE
FOUR >> 2LP P RP P RP PL
```

	PAT1F >> RP PSS LP PSS RS ONEF >> STEELDRUMS BLUES 8PAT1F
	PAT1M >> RS RP5 PSS PSS LP5 ONEM >> STEELDRUMS BLUES 8PAT1M
	ONE >> ONEF^ONEM

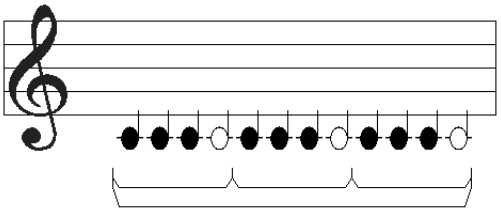
G1 = 2RP P LP P RP P 2LP P
G2 = RP P LP P RP PL LP
PH1 = G1 G2



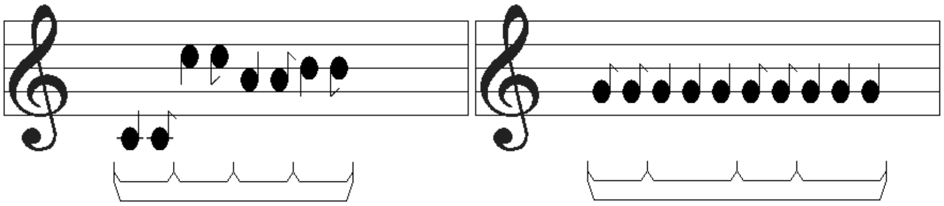
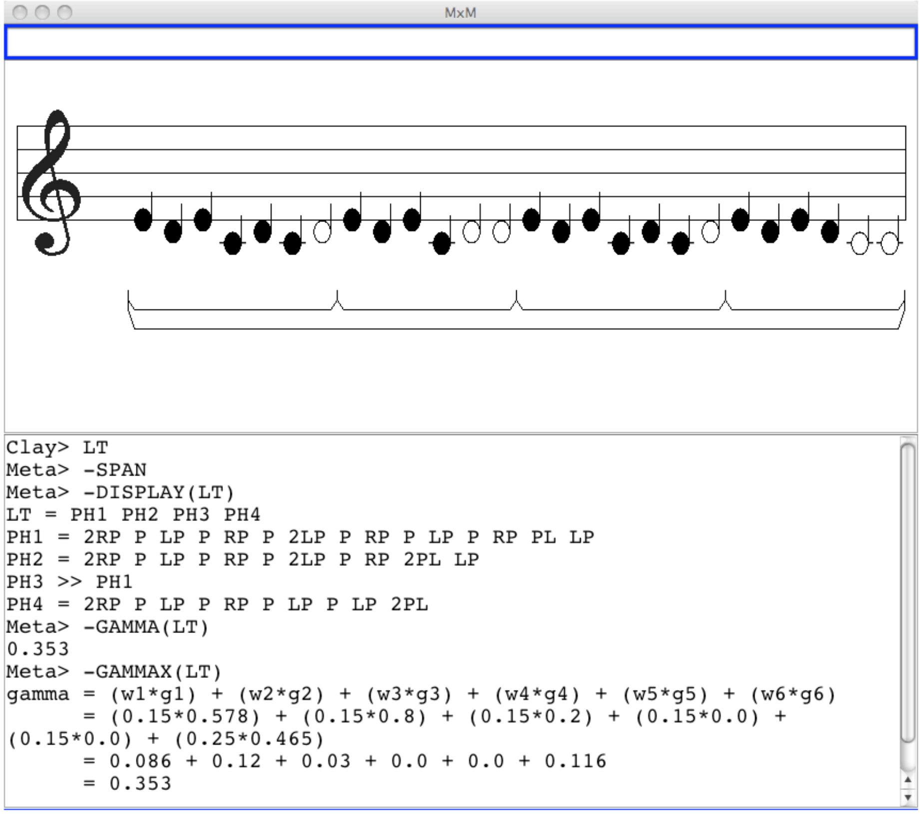


Visual example ...

Musical example ...



Visual example ...

Musical examples ...



```
Clay> LT
Meta> -SPAN
Meta> -DISPLAY(LT)
LT = PH1 PH2 PH3 PH4
PH1 = 2RP P LP P RP P 2LP P RP P LP P RP PL LP
PH2 = 2RP P LP P RP P 2LP P RP 2PL LP
PH3 >> PH1
PH4 = 2RP P LP P RP P LP P LP 2PL
Meta> -GAMMA(LT)
0.353
Meta> -GAMMAX(LT)
gamma = (w1*g1) + (w2*g2) + (w3*g3) + (w4*g4) + (w5*g5) + (w6*g6)
      = (0.15*0.578) + (0.15*0.8) + (0.15*0.2) + (0.15*0.0) +
(0.15*0.0) + (0.25*0.465)
      = 0.086 + 0.12 + 0.03 + 0.0 + 0.0 + 0.116
      = 0.353
```