CHAPTER TWO

Work that has Influenced this Project

Models of Melodic Expectation and Cognition

Leonard Meyer

Emotion and Meaning in Music (Meyer, 1956) is the foundation of most modern work in music cognition. Based on gestalt psychology and the idea of musical expectation but also delving into the emotional nature of musical listening, Meyer's work is far-ranging and full of interesting ideas. I present just a few highlights relevant to my work.

Emotion and Meaning

Meyer states that the "central thesis of the psychological theory of emotions" is that "Emotion or affect is aroused when a tendency to respond is arrested or inhibited." (p. 14) That is, when a given stimulus occurs, a person may naturally respond by reacting in a certain way or having certain thoughts in response. In music, a simple example is the stimulus of an ascending scale that stops on the leading tone, just short of the tonic. The ascending scale evokes the mental response of expecting the tonic to occur next. However, when the expected tonic does not occur, the listener will have an emotional response (in this case, frustration). According to Meyer, emotion derives from this sort of response-inhibition.

He discusses how once a listener is accustomed to the conventions of a particular musical culture, tension arises when expectations are not fulfilled; "deviations can be regarded as emotional or affective stimuli." (p. 32)

On meaning, Meyer carefully notes that in many forms of human communication, meaning is about how one stimulus (such as a word of text) can refer to another item (such as a concrete object or an event or action). He terms this *designative* meaning. However, he points out a special situation where the stimulus item can refer to another thing of the very same type as the stimulus itself; his example is that the dim light of dawn foreshadows the arrival of the full sunlight of the day. He calls this case *embodied* meaning, and notes that this type of meaning is especially important in music (especially "absolutist" music). For Meyer, embodied meaning in music is where one musical event causes the expectation of another musical event. "Embodied musical meaning is, in short, a product of expectation" (p. 35). If a stimulus causes an expectation, "then that stimulus has meaning."

Interestingly, Meyer attributes both emotional response and musical meaning to expectation. Expectation comes from two sources according to Meyer: from learned patterns in a particular musical style as well as from innate perceptual processes as described by gestalt psychology. While Meyer is a proponent of these gestalt ideas, he also is careful to say that "any generalized gestalt account of musical perception is out of the question" because of the influence of style.

Gestalt Principles

I list a few of Meyer's applications of gestalt principles to music here, as they are critical in my work. The Law of Prägnanz is an overaching idea about how people naturally form mental representations that are as well-structured and concise as possible. The Law of

Good Continuation describes how musical processes are expected to continue, so it plays a key role in modeling expectation. The Law of Closure, on the other hand, describes how we hear some structures as completed, which is important in modeling grouping.

The Law of Prägnanz

According to the Law of Prägnanz, people strive for concise, simple, symmetric, well-structured mental representations of the things they perceive. I think of this as a sort of Occam's Razor of perception. Occam's Razor states that all other things being equal, we should prefer the simplest explanation that fits an observed situation. The Law of Prägnanz, similarly, says that people naturally desire simple mental representations. This mental striving for simplicity leads to expectations for structures (visual shapes, observed physical motions, musical gestures, and so on) to continue on to form a complete structure that is easy to represent as a whole. The Law of Good Continuation is a natural consequence of this idea.

The Law of Good Continuation

The Law of Good Continuation (p. 92) states that we expect patterns to continue in the same way, once they are started. Meyer defines several terms which were used very frequently later on (especially by Eugene Narmour in his Implication–Realization model), including *process continuation* and *process reversal.* Process continuation (or simply continuation) refers to the normal mode of "continuing in the same way". For instance, a melody ascending through a major scale is an example of process continuation — at any point until reaching the tonic, we may expect the scale to simply continue. Harmonic motion around the circle of fifths is another example. On the other hand, process reversal, or simply reversal, refers to the stopping of a process. For example, a melody that ascends through the scale from one tonic to the next, an octave above, undergoes a process reversal if the high tonic is held, thus stopping the ascending motion. Note that a reversal in direction is not required; it is the ending of the continued process that reversal refers to, even if the melody does not change direction and move downwards. Process reversal is an abstract type of reversal, unrelated to surface-level reversing of direction.

Meyer discusses several types of continuation, including melodic and rhythmic continuation. Some examples of melodic continuation are:

- An ascending or descending major scale.
- Notes ascending or descending through a triad.
- A phrase that ends on a particular note, followed by a phrase that starts on that same note. The third phrase would be expected to start with the final note of the second phrase if this is heard as a process.

Some examples of rhythmic continuation are:

- Perception of equal pulses (*e.g.*, as described by "internal clock" models — the section on Povel and Essens, later).
- Perception of meter (accented hierarchical beat structure).
- Perception of rhythm: "grouping one or more unaccented beats in relation to an accented beat" (p. 103).

Rhythmic perception depends critically on which notes are heard as accented. Meyer points out how *dynamic stresses* (*i.e.*, stresses added to notes by a performed modifying the notes' volumes or articulations) are not necessarily the same as *perceptual accents*, which are mental constructs based on a listener's perception: "Basically anything is accented when it is marked for consciousness in some way. Such mental marking may be the result of differences in intensity, duration, melodic structure, harmonic progression, instrumentation, or any other mode of articulation which can differentiate one stimulus or group of stimuli from others. Even a silence, a rest, may be accented…" (p. 103).

Completion and Closure

The Law of Prägnanz suggests that the mind is "continually striving for completeness and stability and rest." (p. 128) Meyer gives several concrete ideas about how a musical structure might sound completed. These ideas about grouping have influenced many models of music grouping, including that of Musicat, so it is helpful to review them here. I focus on Meyer's discussion of tonal organization and melodic shape; he also mentions rhythmic completeness and harmonic completeness, but has less to say about them. (In brief, harmonic completeness arises from motion to the tonic, as we would expect, whereas rhythmic completeness has to do with the "apprehension of a relationship between accented and unaccented parts of a cohesive group." (p. 143)

Meyer writes: "In any particular musical work certain melodic patterns because of their palpable and cohesive shapes become established in the mind of the listener as given, axiomatic sound terms." What is it about certain shapes that make them sound complete? For Meyer a primary force is the tonal structure in the musical culture of the piece in question. In Western tonal music, the tonic note typically provides closure, especially when approached stepwise. In other cultures, a different melodic structure might provide similar closure. Even the repetition within a piece of a melodic segment that normally would not sound complete can cause a listen to hear the segment as establishing closure. In addition to repetition and tonal structure, Meyer discusses melodic direction, but his focus is on expectations instead of on completeness. (Perhaps this is because he was implying that if a melody moves as expected it sounds complete.) Although he uses different terminology, Meyer essentially says that in pitch height we expect melodic regression to the mean — in other words, once a melody moves to a high or low register (taking into account the melodic range of a particular instrument or voice) we expect a melody to eventually return to the middle of its vertical range. Listeners hear melodies proceeding as expected as moving towards closure, while melodies denying expectation will sound incomplete.

More generally, Meyer goes beyond gestalt principles and suggests that closure is also due to the shape of tension and relaxation in a melody. For example, when a melody descends, it sounds more relaxed and leads to a sense of closure. More generally still, he asserts that "completeness is directly related to our ability to understand the meaning of a particular pattern" (pp. 138–9). A pattern that is not understood in any special way and that causes no sense of "process" will remain incomplete. Completeness is the result, then, of setting up expectations and motion, and then fulfilling the expectations somehow and coming back to a stable state. This general principle applies to Meyer's ideas not only of melody, but also of rhythm and harmony.

Eugene Narmour

Overview

Narmour's Implication-Realization (I–R) model focuses on the implications (*i.e.*, expectations) generated by the movement from a note to another note and the ways in which the implications are realized (*i.e.*, come to pass as expected) or denied. The model is ambitious in scope, encompassing such topics as bottom-up and top-down perception, the effects of intra- and extra-opus style, and the multi-parametric nature of melody. A novel taxonomy is introduced (Narmour's "genetic code") for categorizing the types of implications and realizations in a melody.

Gestalt Principles and the Problem of Style

The I–R model is primarily concerned with implications generated by subconscious, bottom-up processing. However, Narmour goes to great lengths to explain the important top-down influence of style. In broad terms, he states that for a listener the features of the input music result in subconscious expectations *except* when the context of style modifies those expectations. The roles of top-down and bottom-up processing are clearly separate for Narmour, who insists on "the existence of two expectation systems. The top-down one is flexible and variable but controlled; the bottom-up one is rigid, reflexive, and automatic – a computational, syntactic input system." (Narmour 1990, 54) Although Narmour is careful to put top-down and bottom-up processing on an even footing, the majority of the theory, including a system of symbols introduced used to describe any melody in terms of implications and realizations for various musical parameters, focus on bottom-up expectations rooted in gestalt principles. Style is not treated analytically in the way that note-to-note details of melody are, but Narmour does expose many inherent difficulties with incorporating style into a theory of expectation.

Interestingly, the I–R model states that both top-down and bottom-up processing occur at multiple levels of musical hierarchy. The "top" and "bottom" in "top-down" and "bottom-up", then, must not be confused with "surface" and "deep" structures of musical hierarchy. This theme — the application of the same principles at several hierarchical levels — shows up in several models, including Lerdahl and Jackendoff's *Generative Theory of Tonal Music* and Larson's Seek Well model, as well as in Schenkerian analysis. However, the cognitive basis for applying gestalt principles to musical events at deeper levels (*i.e.*, with a longer time span between events) may still need to be tested. Specifically, if low-level processing describes an innate universal mechanism for processing *raw* perceptual input, it remains to be shown that it is cognitively plausible for this same mechanism to be applied to a more *abstract* level of input (*e.g.*, a background structure in a Schenkerian analysis) that has been extracted from the raw input.

In the study of bottom-up expectations, Narmour invokes the gestalt principles of *similarity*, *proximity*, and *common direction* as cognitive universals that apply to individual musical parameters such as the direction of pitch motion and the size of pitch motion (in terms of vertical intervals between pitches). Additionally, he proposes two new hypotheses — *reversal* and *parametric scale* — to supplement those three gestalt principles. All in all, there are five specific aspects of melody used in classification and description in Narmour's system. These aspects are:

1. Registral direction
2. Intervallic difference
3. Registral return
4. Proximity
5. Closure

Five Fundamental Principles

Registral direction

Registral direction refers to the change in pitch from one note to the next in a melody. The direction is either up, down, or lateral (no change). As long as the interval between two pitches is small, this principle states that a listener will subconsciously expect a *continuation* of pitch direction: upwards, downwards, or lateral motion is expected to continue. A large interval, on the other hand, will lead to an expectation of *reversal* of direction.

An interval is defined to be *small* if it is less than a tritone and *large* if greater than a tritone — the tritone interval itself is a boundary case. Narmour sometimes uses this classification of intervals into these distinct categories in his writing, but he makes it clear early in his book that perception of interval size really operates on a continuum; these categories are used sometimes for convenience but must not be taken too seriously (it seems to me that people often forget about these disclaimers that Narmour makes). Thus, the I–R model's prediction of either a continuation or reversal is not as simple as the interval-size metric based on strict categories might suggest. Narmour uses the term "parametric scale" to refer to a continuum (or an ordered set) of possible values that is psychologically innate — perception of the parameter involved on a particualr parametric scale is subconscious and low-level — and automatically generates implications. Narmour writes that interval size and the strength of resulting implications can be considered as operating along a parametric scale. A very large interval on this scale generates a stronger implication of reversal than a moderately large interval; the same holds for very small intervals and implication of continuation. Additionally, even a very large interval may result in a so-called "recessive implication" for continuation. The word "recessive" means that in retrospect, a listener may reinterpret previous parts of a melody as having implied the notes that actually occurred, even though the implication was quite different earlier in the listening process. For example, a leap up of a major fifth might result in a strong implication for process reversal (*e.g.*, for the melody to move down after the large leap). Narmour calls this the "dominant implication". However, if the leap were to be followed by another leap instead — say, up again by a perfect fourth — the listener might retroactively hear this second leap as a realization of a recessive implication for process continuation.

Intervallic difference

While registral direction refers to the direction of motion between pitches, intervallic difference relates to the size of the interval between pitches. (The size of interval was involved in registral direction in determining the strength of the implication, as discussed in the previous section, but the size of an interval itself is also a parameter that can be implied.) The principle states that small intervals imply a continuation with similar-sized intervals. Large intervals, however, imply relatively a continuation with smaller intervals. Large and small intervals were already defined above in reference to registral direction, but here the concept of "similar-sized" intervals is also important. Narmour defines interval similarity as a difference of a minor third or less in interval size.

Registral return

The typical motion of a melody away from a pitch and then back to the original pitch is known as registral return. Exact registral return describes the perfectly symmetrical melodic archetype **aba**, exemplified in typical melodic shapes such motion to and from a neighbor tone (*e.g.*, the melody G–A–G over a G-major chord). Near-registral return (**aba′**) occurs when the final tone is very similar to the original pitch (within two semitones). Patterns become less archetypal as they deviate more from the symmetrical case.

Recognition of registral return can make an otherwise surprising realization more expected. This idea is also used to explain melodic "streaming" in the I–R model. Streaming is the process by which a monophonic melody is heard as consisting of two or more simultaneous melodies, generally separated by register (pitch height). This occurs often, for example, in Bach's Suite for unaccompanied cello — a melody and countermelody are interleaved, with the cello quickly alternating between low and high notes. In this case, every three notes involves near-registral return. According to Narmour, the listener hears such a melody as a series of overlapping **aba′** processes, which aids the listener in splitting the music into multiple input streams in different registers (Narmour, 1992, p. 352).

Proximity

When the gestalt notion of proximity is applied to pitches, the result according to I–R theory is that small intervals are more strongly implied than large intervals. Additionally, the implications generated by larger-sized intervals are said to be stronger than for relatively smaller intervals.

Closure

Closure describes how listeners break up melodies into separate perceived segments. According to the I–R model, closure in the pitch domain occurs in two cases:

1. The melody changes direction.
2. A relatively larger interval is followed by a smaller interval.

Naturally, parameters other than pitch also can give rise to a feeling of closure. Narmour lists the others as:

1. Interruption of an implied pattern.
2. Strong metric emphasis.
3. Dissonance resolving to consonance.
4. Short notes moving to long notes.

I–R Symbols

The I–R model annotates melodies with symbol strings based on the implications and realizations present in terms of the parameters of registral direction and intervallic difference. Narmour's 16 basic symbols (see his book for details) typically apply to groups of three notes: the first two notes set up an implication based on the pitches and the interval between them, while the third note produces an interval with the second note; this interval and the third pitch may realize or deny aspects of the implication of the first two notes. A sample string describing a melody is "ID-IR-VP-P-IP-VR-D-R-M". The I–R theory hypothesizes that its symbol system can be used "to represent the listener's encoding of many of the basic structures of melody" (Narmour, 1990, pp. 6–7).

Tonal Pitch Space and the I–R Model

Narmour's focus is on innate, bottom-up gestalt laws, so it is natural to wonder how the concept of tonal scale step fits in to the theory. For example, although C-G-C♯ and C-G-D are both examples of near-registral return, the former seems less expected if a C-major context has been established. The I–R theory explicitly states that it supplements conventional notions of tonal pitch space and provides another dimension to pitch relation that should be considered. This is a case where a listener's understanding of musical style interacts with bottom-up perception. Rather than defying accepted notions of pitch space, Narmour intends to contribute to a more complete account of pitch perception.

After making a point of separating the function of scale steps in harmonic vs. melodic contexts, Narmour introduces several categories of scale steps in order to introduce a parametric scale (recall the description of this term above) for melodic implication with respect to scale step. Degrees 1, 3, and 5 are called *goal notes* (GN), degrees 2, 4, and 6 are *nongoals* (NG), and the leading tone is a *mobile note* (MN). The I–R theory already states that small intervals imply continuation while large ones imply reversal. Taking the type of scale step into account affects the strength of the generated implication. Within a clear tonal context, the more differentiated the two tones of an interval are with respect to the scale step categories above, the stronger the implication. Thus an interval moving from 7 to 1 (MN to GN) generates a stronger implication than 2 to 4 (NG to NG) or 1 to 3 (GN to GN). Narmour enumerates all nine possibilities: the most open (i.e. most implicative) combination is GN to MN, while MN to GN is the most closed. Within each category, the particular scale degree chosen also affects the generated implication strengths.

Narmour does not place much emphasis on tonal pitch space in his model, and states that because tonal style is learned, it deserves no more preferential treatment than other parameters affected by learning (Narmour, 1992, p. 85). The focus of the I–R model is on bottom-up processes.

Summary

The I–R theory presents a wealth of ideas about low-level music cognition, especially on the note-to-note level. Although some researchers have shown that parts of the theory, particularly those involving symbol strings and their implications, may be simplified without loss of predictive power (Schellenberg, 1997), I think that these sorts of tests have oversimplified the theory, especially since Narmour goes to some length to discuss the complexity and context-dependent nature of many aspects of melody. Other criticism seems well-founded, however. For example, Margulis (2005) provides a useful critique of the theory that points out how the I–R symbols for basic melodic structures provide taxonomic categorization but do not clearly explain the expectations generated by a melody. Similarly, the question of how expectation relates to affect is mostly ignored by the theory. Finally, melodic hierarchy is invoked by Narmour but the focus remains on local note-to-note relations. Despite shortcomings of the theory, however, I find many of Narmour's comments on the interaction between top-down and bottom-up processing to be especially useful.

Adam Ockelford

Repetition of musical structure is also important to Adam Ockelford (Ockelford, 1991). To describe low-level perception, he coins the term *perspects*, a contraction of the phrase "perceived aspects", to describe parameters of a note that have been perceived by a listener, including such qualities as pitch, duration, volume, timbre, attack time, etc. Larger groups of notes are also associated with such perspects as key and meter. Ockelford describes how a listener may perceive the relationship between certain aspects of the perspects attached to two different notes (for example, a note heard as a quarter-note might be followed by a note heard as a half-note; this relationship would be heard as a doubling in the *duration* perspect.) Next, he presents the concept of higher-order relationships perceived between relationships themselves (meta-relationships). Finally, he discusses *zygonic relationships* between perspects, where the temporal order of events is critical and where a given perspect can influence how. The curious word "zygonic" derives from the Greek word for "yoke" and is used by Ockelford to indicate a relationship in which two things are linked in a particular fashion: typically the first (earlier in time) musical object in the relationship is heard as having an effect on how a later object is perceived. (Sometimes these relationships are heard in the other direction — that is, retrospectively — just as was the case with Narmor's "recessive" implications.) Ockelford's stresses the importance in the directionality of the flow of time when discussing these zygonic relationships. Additionally, although Ockelford does not talk specifically about analogy, it seems that zygonic relationships have much in common with the notion of musical analogies described later, in Chapter 4.

Olivier Lartillot

Each of the authors mentioned in the three previous sections — Meyer, Narmour, and Ockelford — wrote in some way about the importance of recognizing repetition in musical structure. Olivier Lartillot's has developed a computer library named "kanthume" (later called "OMkanthus") (Lartillot, 2002, 2004) that looks for musical patterns in a cognitively-motivated manner that, like Musicat, is explicitly concerned with analogy-making and real-time perception:

> "…an analogy is the inference of an identity relation between two entities knowing some *partial* identity relation between them. Our discussion about the phenomenon of time in music leads to the idea that our experience of music is a temporal progression of partial points of view that consists of the *timely local* perception of it. Thus the analogy hypothesis of music understanding means that the global music structure is inferred through induction of hypotheses from local viewpoint. [sic]" (Lartillot, 2002)

(The term "viewpoint" refers to musical parameters that are the focus of attention in a particular context — this seems quite similar to Ockelford's "perspects".) Even though Musicat and kanthume have completely different architectures, the projects have similar guiding philosophies.

The kanthume program takes symbolic musical scores as input. It groups notes in the score together and detects repeated motivic fragments in a piece, forming a motivic network that is implicitly stored in long-term memory (the program shows the network as a collection of rectangles and relationships superimposed on a musical score). The program discovers relationships between groups: it can notice that a group is a transposition of another group, or that a rhythm of a group is equivalent to that of another group where all duration have been cut in half (*i.e.*, it recognizes augmentation and diminution of rhythms). The model finds such relationships by attempting to determine which parameters are the most important in a given context, and then it extracts motifs based on intervallic, contour, and rhythmic relationships as appropriate. The program uses heuristics to prune away irrelevant candidate motifs.

Lartillot's model makes a special effort to model the effect of temporal relationships. On human music cognition, Lartillot writes, "a pattern will be more easily detected if it is presented very clearly first, and then hidden in a complex background than the reverse." The computer model is designed to respect temporal ordering during listening, and, much like Musicat does, holds notes in a simulated short-term memory buffer. Certain links between notes explicitly account for temporal ordering: Lartillot uses the term *syntagmatic relations* to refer to links between notes that form a temporally-directed relationship in which one note of a motif stored in memory can activate the memory of the successive note. (To me, Latrtillot's "syntagmatic relations", when taken in conjunction with his notion of "viewpoints", seem quite similar to Ockelford's "zygonic relationships".) In other words, these relations represent memory items that are linked in a temporal sequence such that the first note of a sequence can cause the second, and then the third, and so forth to be recalled from memory.

The reader may be curious at this point to know how Musicat and kanthume differ, because both projects involve real-time listening and analogy-making. One major difference is that Musicat's highly-stochastic architecture is based on a simulation of various internal forces and pressures (some of them working in concert and others in conflict) that generate various grouping structures and relationships, in what is often a frenzied and chaotic-looking manner, but from which a stable structure gradually emerges. Kanthume, on the other hand, takes a more traditional algorithmic approach, using heuristics with names such as "maximization of specificity" and "factorization of periodicity" to find a satisfactory parsing of a musical fragment in a deterministic way.

Fred Lerdahl and Ray Jackendoff

A Generative Theory of Tonal Music, or GTTM (Lerdahl & Jackendoff, 1983) has arguably been the most influential single work on music cognition in the past three decades. The book was inspired by Leonard Bernstein's lecture series at Harvard (Bernstein, 1973), in which he made an analogy between Noam Chomsky's ideas about a universal grammar in linguistics and a possible universal grammar of music. GTTM was a formal attempt at describing a new system of music analysis based on linguistic ideas such as formal grammars. Rather than attempting to summarize this large and complex work, I will point out several ideas in GTTM that are relevant to my project. Particularly interesting are the chapters on grouping and meter and the general notion of combining two different types of rules (*well-formedness rules* and *preference rules*). GTTM analyses are visually striking due to the two types of grammar-like tree structures created in accordance with the rules of the theory, so I will also touch on *time-span reduction* and *prolongational reduction* trees.

Well-Formedness and Preference Rules

GTTM provides notation for analyzing several aspects of music such as hierarchical grouping and tension–release structure. For any piece, a formal "analysis" of its structure can be generated using the rules and guidelines in GTTM. *Well-formedness* rules prescribe a specific "mathematical" form for any acceptable analysis; any analysis according to the theory must follow these rules precisely. *Preference rules*, on the other hand, suggest how to choose between competing analysis possibilities. While well-formedness rules are analogous to rules in a formal linguistic theory, preference rules are a unique contribution Lerdahl and Jackendoff found necessary for describing *musical* grammar. They note that:

> the interesting musical issues usually concern what is the most coherent or "preferred" way to hear a passage. Musical grammar must be able to express these preferences among interpretations, a function that is largely absent from generative linguistic theory. Generally, we expect the musical grammar to yield clear-cut results where there are clear-cut intuitive judgments and weaker or ambiguous results where intuitions are less clear. (p. 9)

Temperley (see below) produced a concrete computer model for several aspects of musical cognition based on some of the preference rules in GTTM. The operation of some of the "codelets" in my project can also be seen as an alternate mechanism for implementing some of these preference rules.

Grouping and Meter

GTTM begins with a description of hierarchical grouping structure of music. Interestingly, pitch is absent from the initial chapter. Core concepts include the following:

- Musical groups combine in a hierarchical fashion.
- The hierarchy is nearly strict; groups do not overlap except for very short segments where the end of one group can be elided with the start of the successive group.
- The pattern of accents at the musical surface gives rise to an implied metric hierarchy consistent with the pattern.
- Grouping structure and metrical hierarchy interact with each other, but are distinct and should not be confused; they be or may not be in phase with each other.

Grouping structure is constrained by well-formedness rules that formalize how groups form hierarchies. Preference rules that choose between alternate well-formed group structures are derived largely from gestalt psychology ideas. Some of the rules (paraphrased) include:

- Avoid analyses with very small groups.
- Locally-large rhythmic gaps maybe heard as group boundaries.
- Larger rhythmic gaps indicate group boundaries at higher hierarchical levels.
- Prefer analyses where groups are subdivided into two equally-sized pieces.
- Parallel segments of music should form parallel groups (see Chapter 4 for more discussion about the ambiguity of the word "parallel").

Metrical structure is likewise described by both types of rules. The well-formedness rules here state rather obvious things such that each beat at a level must also be a beat at lower hierarchical levels, but also less obvious things such as:

- Strong beats at each level are two or three beats apart.
- Each level must consist of equally-spaced beats.

The preference rules for meter include (paraphrased):

- Prefer structures where strong beats appear early in a group.
- Prefer strong beats to line up with note attacks.
- Prefer strong beats to correspond with stressed notes (stressed notes are those performed with an articulation or change in dynamics that distinguishes them from nearby notes).

These rules are reminiscent of Povel and Essen's rules (1985) for internal clock induction (described in the section on Musical Rhythm below).

Reduction Trees

GTTM analyzes music using tree structures called *reductions*. These trees are a formalization of the analytic notation introduced by Heinrich Schenker (Cadwallader & Gagné, 1998). The basic concept is that trees are drawn in such a way that a straight branch of a tree might have another branch that attaches to the first at an angle, indicating that this second branch is subordinate to the first, straight branch. Trees start at a "root" drawn above everything else, and branches grow downwards until they terminate in concrete musical events (notes or chords). A tree so constructed makes obvious which structures are primary and which are considered elaborations of more-primary structures; this is very much like Schenker's multi-staff reductions. Trees, like grouping and rhythmic structures, are subject to a set of well-formedness and preference rules.

Surprisingly, GTTM describes two different types of reductions, each of which can be applied to the same segment of music, even though they are independent. *Time-span reductions* relate closely to the metrical and grouping structure of a piece, and also are informed by tonal cadences. *Prolongational reductions*, on the other hand, describe the tension–relaxation structure of a piece; in GTTM this is described as "the incessant breathing in and out of music in response to the juxtaposition of pitch and rhythmic factors." Time-span and prolongational reductions may interact (after all, they describe the same underlying music), and the tree structures generated may be more or less *congruent*: "Congruent passages seem relatively straightforward and square; noncongruent passages have a more complex, elastic quality."

The idea of two different reductions applying to the same music is consistent with GTTM's principle of considering the effects of not only well-formedness rules, but also multiple, possibly conflicting preference rules. Even though GTTM is a formal music theory, it maintains surprising flexibility of description. Indeed, as presented in the book it is too flexible to be used for algorithmic analysis, although several people (Hirata, Hamanaka, & Tojo, 2007; Temperley, 2001) have made computer models based on particular aspects of GTTM.

Fred Lerdahl

Lerdahl's model of tonal pitch space (Lerdahl, 2001) is not exclusively a model of melodic expectation, having perhaps more to do with harmony than melody. However, some elements of the model do produce quantitative predictions of melodic motion. In this model, the tones of the scale exist in a hierarchy of alphabets, after an idea of Diana Deutsch and John Feroe (1981). Basic tonal space, by this account, has five levels. In C-major, the levels include the following notes:

1. C
2. C, G
3. C, E, G
4. C, D, E, F, G, A, B
5. C, D♭, D, E♭, E, F, F♯, G, A♭, A, B♭, B

That is, the levels successively add the tonic, fifth, tonic triad, diatonic octave, and chromatic scale.

The theory provides metrics for calculating "distances" between two notes based on the number of hierarchical levels and horizontal steps between the notes in this alphabet hierarchy. Next, formulas are provided that define distances between two chords in different tonal contexts. Later, the theory gives a way to calculate melodic tension; that is, it assigns a specific numeric tension value to each note in a melody (given the context of a key or a chord within a key). Furthermore, the theory gives a way to calculate the amount of tension in a particular chord (with respect to a given key).

Melodic tension (pitch instability) depends on the *anchoring strength* of the two pitches involved and the (squared reciprocal) distance between the pitches in semitones. Anchoring strength derives from the hierarchical level on the pitch in a space similar to the alphabet hierarchy above (in C-major, C has anchoring strength 4; E and G both have 3; D, F, A, and B have 2, and the chromatic pitches have strength 1.) The strength of attraction between two notes is calculated by dividing the anchoring strength of a possible second note by the strength of the first note and dividing by the squared distance between notes. A pitch is thus attracted to nearby pitches with high anchoring strength. This formula for attraction provides a quantitative value for the attraction between any pitch and a possible following pitch. The formula is the analogue in this theory to the expectations generated by Narmour's I–R theory (Lerdahl, 2001, p. 170)

Elizabeth Margulis provides a critique of the formula, pointing out that registral direction is ignored, semitone movement yields unreasonably high attraction values, and pitch repetition is ignored completely by the model (Margulis, 2005). Margulis incorporates an extended version of Lerdahl's model of attraction in her own model. Larson's single-level Seek Well model also includes a similar parameter of attraction – magnetism – but it, too, is augmented with other model components.

DIANA DEUTSCH AND JOHN FEROE

Deutsch and Feroe (1981) developed a cognitively-inspired model of internal pitch representation for pitch sequences. This internal representation is based on hierarchical *alphabets* of pitches and operators that refer to these alphabets to represent melodies in a compact way. Alphabets are ordered lists of pitches, such as "CDEFGAB" in the C-major scale or "CEG" in the C-major triad. Alphabets typically extend cyclically in either direction (*e.g.*, the C-major scale can continue on to the C just above the B). Alphabets can be hierarchical, which refers to two related concepts. First, one alphabet (such as "CEG") may be a subset of another alphabet (such as the C-major scale). Second, melodic lines may use different alphabets at different structural levels: at a higher level a melody may move through an alphabet such as the C-major triad, while at a more surface level the melody may be elaborated using notes from a superset alphabet such as the C-major or the chromatic scale.

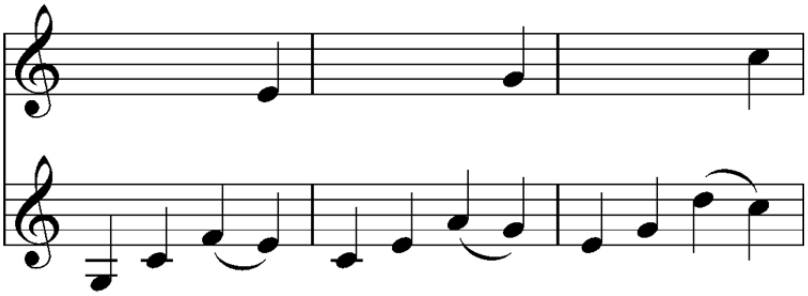

Figure 2.1: Melody described using pitch alphabets.

For example, the melody in Figure 2.1 can be heard as three sequential copies of a four-note motif that ends on a member of the C-major triad, and we can hear the higher-level structure as simply E-G-C. At the surface level, each of those notes is elaborated by a preceding upper neighbor tone from the C-major scale alphabet (indicated by slurs from F, A, and D in the figure). Incidentally, the remaining notes at the surface level are also members of the C-major triad alphabet. Given these two alphabets, we can describe the entire melody quite compactly using simple operators such as alphabet-successor/predecessor and composition. The high-level melody line can be generated by starting with E and applying successor twice in the C-major alphabet. Then the entire melody can be generated by composing this three-note line with a four-note pattern:

predecessor of predecessor in C-major triad – predecessor in C-major triad –
successor in C-major scale – identity

Describing melodies using alphabets and operators results in a compact representation that Deutsch and Feroe argue is cognitively plausible — they suggest that tonal music evolved to use hierarchies that take advantage of human memory structure. For tonal music in particular, different notes in a scale have different functions and degrees of stability. Hierarchical alphabets can model the relative importance of different notes in tonal music quite naturally.

David Temperley

Lerdahl and Jackendoff's GTTM belongs to the world of music theory and analysis; it was cognitively inspired, but it is not specific enough to be implemented as a computer model. Temperley (Temperley, 2001), along with Daniel Sleator, implemented computer models of many components of music cognition, using principles from GTTM as inspiration. More recent work (Hirata et al., 2007) has attempted to implement GTTM itself, but for my purposes, Temperley's book (which I refer to by his acronym TCoBMS — *The Cognition of Basic Musical Structures*) is very interesting because it has goal much like my own in terms of modeling basic processes in music cognition. Of the most interest to my work is his model of melodic phrase structure, which I summarize here.

The models in TCoBMS work by optimizing an analysis of a piece of music according to a set of preference rules. For any proposed analysis, each preference rule can be used to compute a cost that measures the degree to which each musical structure generated by an analysis violates the rule. These costs are summed across all rules and structures to determine a global cost. The preference rules are all relatively local in scope and thus the global cost can be minimized efficiently using the Viterbi algorithm, a dynamic programming technique which often is applicable to music analysis. For the problem of segmenting music into melodic phrases, TCoBMS uses three preference rules, which I paraphrase here (Temperley, 2001):

1. **Gap Rule**. Phrase boundaries should be located where there are large rhythmic gaps, formed by either a phrase ending with a relatively long note or a long rest. (In the formula, actual rests are weighed twice as strongly as long note durations.)
2. **Phrase-Length Rule**. Phrases should have close to 8 notes. Phrases between 6 and 10 notes long have a low penalty, but outside of this range the cost goes up.
3. **Metrical-Parallelism Rule**. Successive groups should start at the same rhythmic position (*e.g.,* if a phrase starts on an upbeat such as beat 3 in 3/4 time, the next phrase should also start on the same upbeat).

Temperley applied these three rules to a subset of 65 songs from the Essen folksong collection, which is notable because it is a digital collection of melodies where the data files have been annotated with phrase boundaries. (Unfortunately, hierarchical grouping structure is not indicated; if it had this feature, it would be extremely useful for studying Musicat's performance. Still, the phrase-level analysis is interesting.) Temperley's model correctly identified 75.5% of the phrase boundaries. In a later chapter I will examine Musicat's performance on the same test set.

Elizabeth Margulis

Margulis developed a model of melodic expectation that included elements of both Narmour's and Lerdahl's models (Margulis, 2005). Both tonal pitch space and innate bottom-up processing are given significant status in the model. The model provides solutions to problems that Margulis had pointed out with each in her critiques. For example, it explicitly describes how expectation connects to the experiences of affect and tension, how to deal with repeated notes, and how hierarchy can be used formally to include more than the two preceding notes in the generation of expectations. The model is composed of five separate components: stability, proximity, direction, mobility, and hierarchy.

Model Components

Stability

Stability of melodic events is calculated based on the tonal context. Rules adapted from Lerdahl (2001) select a chord and current key for each pitch event. Based on the tonal function of each pitch, a stability rating is assigned. These are numerically similar to the anchoring strengths in Lerdahl's theory. However, they are more sophisticated because they are based on the current tonal context. For example, several exceptional cases such as augmented sixths and Neapolitan chords are enumerated in the model to improve the quality of stability predictions.

Proximity

Just as in Narmour's I–R model, the principle of proximity states that listeners have higher expectations for pitches nearby in frequency. This model gives a numerical proximity rating to pitches based on the distance in semitones away from the preceding pitch. Margulis selected the particular numerical values by hand based on results from various studies and on her own intuitions.

Direction

Narmour's I–R model states that small intervals imply continuation of direction but large intervals imply reversal. Margulis incorporates this idea to generate particular expectations (for continuation or reversal) along with the strengths of each expectation. These expectations are based on the interval size measured in half-steps. Instead of prediction strength being a simple linear function of interval size, Margulis uses data from Schellenberg's study on simplifying the I–R model (1997) to suggest a particular nonlinear mapping from interval size to prediction strength.

Mobility

Although many theories ignore the possibility of repeated notes, Margulis includes a mobility parameter that increases the degree to which a note is expected to move to different pitch. Repeated notes lead to a mobility penalty whereby the stability and proximity scores are multiplied by 2/3 to encourage motion. Margulis notes that this parameter was explicitly added to reduce the strong expectations for repetition otherwise produced by the model.

Hierarchy

Without the concept of hierarchy, the amount of expectation for a pitch to follow another pitch is given by the model as **stability × proximity × mobility + direction**. In other words, it's good if the second note is close to the first one in pitch, if the second note is a stable pitch, and if the motion is in the expected direction. Hierarchy is incorporated in the model by generating a time-span reduction of the input and then applying the formula above to each level of the hierarchy. The final expectancy value for each pitch (expectancy is simply the degree to which a pitch is expected) is given by a weighted average of the values at each level. Margulis selected the weights in this formula by hand; the surface level has a weight of 15, other levels of no more than two-second duration have weights of 5, remaining levels with less than a six-second duration have a weight of 2, and no levels are considered with a time span longer than six seconds. In the model, the time-span reduction is generated via an implementation of preference rules from GTTM (Lerdahl & Jackendoff, 1983), augmented with an additional preference rule.

Tension and Affect

This model defines three different types of tension based on calculated expectancy values: surprise-tension, denial-tension, and expectancy-tension. The first two types are calculated for a third event in a series based on the two prior events. For example, the notes A-B in a C-major context may set up an expectation for continued upward motion to the tonic C, but if the pattern continues A-B-F# the third event yields a high value for denial-tension. Expectancy tension, on the other hand, is calculated when looking forward to a potential expected future event. Thus, in our example, the second event, B, may have a high value for expectancy tension if the tonic C is strongly expected. Expectancy tension describes the strength of a future expectation, whereas the other two types describe the amount of surprise or frustration that result from an event that has just occurred.

Surprise-tension describes how unexpected an event was. In other words, its value is inversely proportional to the expectedness of the event. An event that was not expected at all results in high surprise-tension, while an event that was expected to a moderately high degree results in moderately low surprise-tension. In a C-major context, a sudden, unprepared F♯ would result in high surprise-tension. Margulis describes high values of surprise-tension as being associated with an experience of "intensity" and "dynamism".

Denial-tension is proportional to the difference between the expectancy value for a note that occurred at a particular time and the maximum expectancy value over all possible notes that could have occurred at that time. Thus denial-tension is strong when there is a very strongly expected note that does not occur, or when the actual note is quite unexpected relative all possible note options. For example, consider a melody moving up the C-major scale from C to the leading tone, B. If the melody continued by moving down to A instead of moving up to the tonic, C, it would result in high denial-tension, because the tonic is so expected after the leading tone. Margulis writes that high denial-tension results in feelings of "will, intention, and determinedness".

Expectancy-tension is proportional to the strength of the most-expected following note. Its calculation is thus quite similar to that of denial tension except that it is forward-looking. In the example of the C-major scale above, there would be a large amount of expectancy-tension at the moment the scale reached the leading-tone, B, because the tonic C is so much more expected than any other note. Expectancy-tension is associated with feelings of "yearning" and "strain".

Margulis uses the three tension formulas to generate graphs of each type of tension versus time for musical pieces.

David Huron

Huron's book *Sweet Anticipation: Music and the Psychology of Expectation* (2006) gives a broad overview of research relevant to musical expectation. In the book Huron develops his "ITPRA" theory of expectation. Like Margulis's model of expectation, Huron's model describes emotional states that come along with the experience of expectation. ITPRA stands for *imagination*, *tension*, *prediction*, *reaction*, and *appraisal* — all of these are stages in the process of expectation. Notice that ITPRA is a general model, applicable to other domains than music.

Huron's model is fascinating, but expectation is not completely central for Musicat in its current state. Therefore, I will not describe the model in detail here. However, I will point out a few key sections of the book that I found particularly relevant.

Heuristic Listening

In Huron's book, the title of Chapter 6, "Heuristic Listening", is especially interesting to me because I think of Musicat as a heuristic listener. A key element in this chapter is a comparison of statistical features of Western melodies with experimental results of how people actually listen. Huron gives four statistically common musical patterns, but three of the four seem to be represented by imperfect heuristics in listeners' expectations:

1. Pitch proximity (the tendency for the next note in a melody to be close to the previous one) is found both in statistical analysis of melody and in listeners' expectations for the next pitch.

2. Regression to the mean (melodies tend to revert back toward the mean pitch) is approximated by listeners expecting post-skip reversal (a large leap should be followed by a change in pitch direction).
3. Downward steps are the most common melodic motion, but listeners instead expect inertial motion for small intervals (in either direction) to continue in the same direction.
4. Arch phrases are common (melodies tend to begin by ascending and to end by descending) but listeners only expect the second half – descending pitches in the last half of a phrase.

Tonality

Scale Degree Qualia

Huron performed a short survey of listeners' experiences in Western tonal music, asking them to describe the individual qualities of each chromatic scale degree in a musical key (p. 144–147). The survey resulted in quite rich descriptions for each note, and suggest that a model of musical listening needs to somehow incorporate this sort of knowledge. Some responses were unsurprising: the tonic was described as "stable" or "home", but also as being associated with "pleasure" or "contentment". The mediant was described using words such as "bright" and "warmth", but also with the much more emotional-laden words "beauty" and "love". The dominant tone was "strong", "pleasant", and even "muscular". Huron found many types of scale degree qualia that were shared across listeners, and speculates on their origins. These qualia seem crucial to the human experience of listening, but current models (Musicat included) barely incorporate this sort of knowledge at all. This is clearly a vast and complex aspect of musical listening that is important for future work in music cognition.

Cadence and Closure

Huron discusses how points of temporary closure in music are related to tonal pitch patterns (pp. 143–157). The notion of "cadence" is well-explored in music theory: a cadence is a specific pattern of melodic and harmonic elements that signify closure in Western tonal music. Different cadences can signify different amounts of closure. Huron notes that even non-Western music has cadential patterns, and within tonal music, different composers and different genres have different types of cadence.

Huron refers to the discipline of "information theory" to describe cadential patterns in terms of the statistical predictability of certain patterns of notes. According to the Shannon-Weaver equation, the notes that occur just before a point of closure (such as a cadence) have low uncertainty (high predictability). A cadence is often heard as a "reset" point, and notes following a cadence have high uncertainty (they are hard to predict). Huron mentions that Narmour's conception of closure can be summarized in a way that supports this view of cadence as a "reset point": Margulis has said that for Narmour, closure is "an event that suppresses expectancy" (p. 157).

In summary, cadences are learned patterns in tonal music that establish a feeling of closure. Although we can understand parts of music cognition in terms of gestalt theory, cadential patterns must be learned for each particular musical tradition.

Expectation in Time

Huron describes a set of experiments by Caroline Palmer and Carol Krumhansl to study listeners' perceptions of musical meter and metric hierarchy, inspired directly by Krumhansl and Edward Kessler's probe-tone experiments in perception of tonality. Interestingly, Huron makes a very strong association between tonal hierarchy and metric hierarchy:

> But the similarity between scale-degree expectation and metric expectation is not merely metaphorical or informal. Both scale degree and metric position are perceived categorically. Like scale-degree pitches, metric positions provide convenient "bins" for expected stimuli. The metric hierarchy is truly homologous to scale or scale hierarchy. (p. 179)

This analogy between pitch and meter was quite unexpected to me, and I find it quite provocative and exciting. When a melody ascends through a major scale from the tonic up to the tonic an octave higher, it starts on a very stable pitch (say, C, in C-major), moves past a unstable pitch (D), then a somewhat stable pitch (E), a less stable pitch (F), a very strong and rather stable pitch, the dominant (G), a less-stable pitch (A), an extremely unstable pitch, the leading tone (B), and finally it arrives on the tonic, again, a very stable pitch. The pattern of stability alternates quite regularly between stable and unstable (aside from the leading tone), and even the degree of stability fluctuates in a regular pattern: C and G are more stable than E, sandwiched between them and G itself is less stable than the two tonic C's that surround it. This pattern of alternation is quite reminiscent of the pattern of alternating strong and weak beats we find in a metric hierarchy. Both patterns look something like the markings on the edge of a ruler that denote inches, half inches, quarter inches, and so on. If only our default time signature were 7/4, the analogy between pitch and meter would be quite elegant! Still, the analogy suggests that the same (or similar) cognitive mechanisms might be applicable in both the pitch and meter domains.

Binary Default

Huron describes some research (p. 195) supporting the idea that Western listeners expect "binary" structures in musical meter. Even without resorting to experimental data with human subjects, a survey of Western classical music suggests that meters based on two or four beats are much more common than other meters. Specifically, Huron used over 8000 melodies from Barlow and Morgenstern's *Dictionary of Musical Themes*, and discovered that 66% of them used a binary meter of two or four beats per bar. (Incidentally, in a study I conducted, 79% of participants' improvised melodies were in duple or quadruple meter — see Appendix B.) Huron's results are not surprising, but they are important because it helps to justify building in a strong duple-meter bias in computer models such as Musicat.

Bob Snyder

Music and Memory (Snyder, 2000) details how human memory processes influence music cognition in many different ways, across time scales ranging from a few milliseconds to many years. Unlike many other references in this chapter, this book's aim is to synthesize existing work in memory-related aspects of music cognition, rather than present a new theory or computer model; as a result, it examines a surprisingly wide range of complex musical phenomena and cognitive processes. Some highlights relevant to my work include the book's overview of memory processes, closure, melodic schemas, categorical versus parametric aspects of music, rhythmic tension, and hierarchies of musical chunks. (Other sections of the book such as those on gestalt grouping and GTTM have been addressed earlier in this chapter.)

Memory Processes

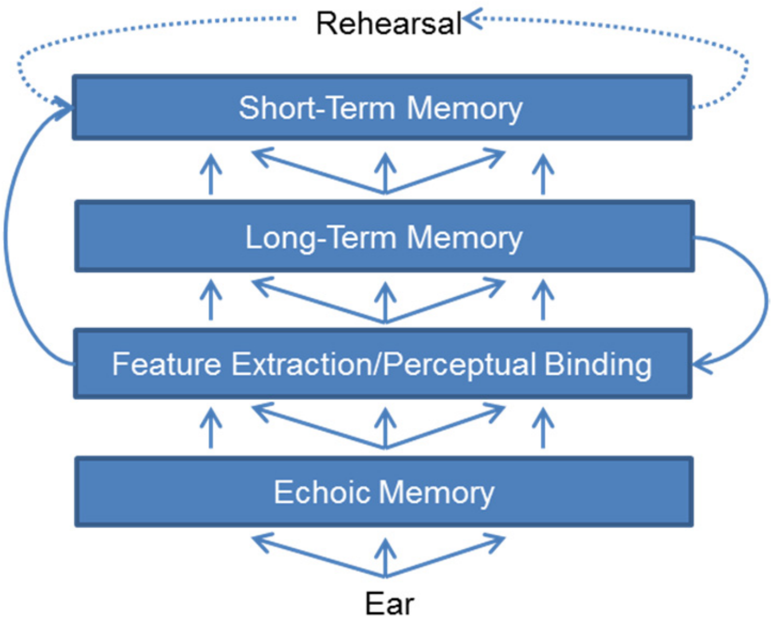

Figure 2.2: Auditory memory processes, adapted from (Snyder 2000).

Music and Memory begins with a chapter describing several important cognitive processes involved with music perception. Three different types of memory appear in a figure on p. 6, shown here in quite simplified form in Figure 2. Sound comes into the ear and is separated into its component frequencies by the clever architecture of the inner ear and some neural circuitry. This information persists for a short time in the *echoic* memory, a sort of short-term buffer where we can remember the past few seconds of raw perception. Traces of sound "echo" here long enough for slightly higher-level perceptual features such as individual notes, percussive events, and timbres to be extracted. These perceptual features can interact with and activate categorical structures in long-term memory (LTM), resulting in perception of music as sequences of larger conceptual structures (such as musical motifs). The stream of low-level perceptual information can also be sent directly to short-term memory and conscious awareness, where it will be perceived in a fleeting way that Snyder describes as "nuance". Nuance refers to the details of an instance of a perceptual category that differentiate the instance from the generalized category.

Most of the processes in the diagram are unconscious. Synder describes a window of conscious awareness in which the structures in short-term memory are available for (completely) conscious examination. The process of rehearsal (the conscious and internal repetition of a fragment of music) helps make it more likely that the current perceived structures will get stored in LTM. This is important because only the most recent three to five seconds of music in short-term memory (STM) are generally available for conscious scrutiny.

The time scales involved with both memory-based processes and perceptual processes interacts with the time scale of musical structures in an interesting way. Snyder divides musical structures into three main groups based on the time scale involved: event fusion (hearing sound waves as musical events), melodic and rhythmic grouping, and form. These correspond to three memory-based or perceptual processes: early processing, STM, and LTM. Early processing occurs for events that are less than approximately 1/20 second long (frequencies above 20 Hz), STM is most involved in processing for events lasting up to about 8 seconds (such as melodic grouping and phrasing), and anything longer than that (longer musical sections, entire movements or works) must have more to do with LTM.

Categorical vs. Parametric Aspects

In a chapter on large-scale form, Snyder is concerned with how boundaries are perceived between sections. He asserts that large-scale boundaries are points of "multiparametric change", where each section internally has "parametric constancy" (of course, these sections can be of various time-scales, so change and constancy are relative terms that take into consideration the time-span in question.) But what are these various parameters which may be perceived as changing at section boundaries? Snyder divides them into *primary* and *secondary* parameters, referencing Meyer and Hopkins. This division is helpful in thinking about modeling various aspects of music.

Primary parameters are those that can be perceived categorically (*i.e.*, parameter values belong to discrete categories), such as notes in a scale or time segments made up of fixed rhythmic units. Microtonal variation in a note (such as vibrato) would not be primary, but hearing a note as having an identity such as "the fifth degree of the major scale" would be primary. Snyder identifies three primary parameters: pitch, harmony, and rhythm.

Secondary parameters are those that vary continuously, such as volume, tempo, or timbre. Even though we very clearly perceive changes in these parameters, they cannot be represented in memory in as rigid a way as the primary parameters.

Tonality and Melodic Schemas

How is the pitch content of a melody organized in memory? In addition to possible *local* representations such as absolute pitch, intervals between pitches, and pitches as elements of an "alphabet", the *larger-scale* structure of pitch in a melody is important. Snyder (like Schenker and others) discusses tonality and melodic shape as organizational forces, but in Snyder's book these are discussed in terms of human memory constraints.

Memory is particularly important for tonal organization because tonality is only perceived in context as musical time moves forward. Some music-informatics approaches to tonality, in contrast, consider collections of pitches as an unordered set (especially those that make use of the Krumhansl–Kessler key profiles (Carol L. Krumhansl, 1990) for key estimation). However, the order of pitches in a melody matters greatly in perceiving tonality (Butler & Brown, 1994). According to Snyder, part of establishing tonality is that the repetition of a pitch causes it to become central. Other musical parameters also help emphasize key pitches: notes occurring on strong metric positions or with long durations are emphasized. Once tonality is established, return to the central pitch can help establish a sense of closure (see below). Tonality establishes expectations and specific goals, and in turn, reaching these goals aids in establishing grouping boundaries.

In addition to perceiving individual pitches in terms of tonal structure, people hear sequences of pitches in terms of melodic schemas. The most well-known such schema is perhaps the *melodic arch* form: many melodic structures gradually rise to a high note and then fall back down rapidly to the starting pitch. The tendency for the climax to be found closer to the end than the middle of the segment is more pronounced for larger-scale structures. (See the previous section on Huron for more on melodic arch and expectation.) Another melodic schema is the *axial melody*, in which a melody will fluctuate about a central pitch — good examples are found in Gregorian chant. Whereas the melodic arch involves rising from and falling back to a stable base (as in Larson's notion of musical gravity), an axial melody significantly involves notes both above and below the central (not base) pitch. A final type is the *gap-fill* schema, in which a melodic leap forms a "gap" and the melody eventually moves in the opposite direction, to "fill in" the missing notes in the gap — the vertical gap between pitches is heard as forming a melodic "vacuum" and the melody notes are "sucked" into the space left between the two pitches involved in the leap.

Melodic schemas offer one way in which we can understand a melody as it progresses in time (as they help suggest goals and expectations). After a melody has reached closure, we may internally represent these larger-scale melodic shapes as belonging to one of these schema types (*i.e.*, the schemas represent broad categories of stereotypical melodic shapes, and we may hear melodies as members of these categories).

Closure

Snyder invents the term “soft closure” to describe the weakest kind of separation in music that causes a grouping boundary. For example, a melodic leap or a relatively long time interval between notes could create soft closure and a very local grouping boundary. He makes a rather bold claim about how the number of parameters establishing closure determines the hierarchal importance of a boundary:

> Melodic or rhythmic grouping boundaries are usually established by changes in one basic parameter, such as pitch contour or duration, whereas phrase boundaries are usually reinforced by changes in *more than one* parameter… (p. 38)

I find this difference between just one and more-than-one parameter to be too simplistic-sounding. However, Synder softens the statement by going on to mention that “the distinction between grouping and phrase is not absolute”. Later, he makes the more general statement that higher levels of structure attain closure by the simultaneous closure of a larger number of musical parameters. The most definite closures are established through culturally-specific musical syntax, such as cadential patterns, but even without the syntax of a particular musical style, changes in multiple parameters can establish a strong sense of closure.

Closure is also strongly tied to expectation. If a musical pattern does not suggest particular expectations for the listener, and then it ends, it will not impart a feeling of closure; it will just sound like starting followed by stopping. Tension and expectation are required before closure can happen. In addition to the things mentioned above, such as cadence and multi-parameter change, which may imply closure, closure can also result from repetition of previously-heard musical material. For example, the ABA form (which can function at many different levels of hierarchy) relies on the contrasting B section to establish some tension, and then the repetition of the initial A section releases the tension and leads to a feeling of closure.

Rhythm

In order to be perceived as such, a rhythm must take place on a short enough time scale that it may be represented in short-term memory — a maximum of five seconds or so. (p. 162) According to Snyder, this is necessary because a rhythm is defined in terms of relative time intervals, and the rhythm needs to be held in memory all at the same time so that the relationships between time intervals can be interpreted correctly. Also, rhythmic perception is categorical — that is, we hear elements of rhythms as belonging to discrete categories such as quarter-note and half-note, not in terms of continuous parameters such as "number of seconds of duration" — as is shown, for example, by the way in which humans can recognize the underlying rhythmic structure of a *rubato* passage despite the distortions it involves). The subtle shifts in tempo and attack times and durations that make for expressive performance are important for listeners, but rhythms still retains their identities during this kind of performance. It seems, then, that short rhythms are encoded in memory in a categorical manner where time intervals are heard in terms of small-integer ratios relative to each other or to a basic, lowest-common-denominator pulse.

It is natural to talk about musical tension in the pitch domain (think of the importance of the highly-tense tritone), but Snyder also defines several useful terms for thinking about tension in rhythm and meter. *Metrical tension* refers to the difference between the accents expected in a particular meter and the accent structure and rhythmic grouping of the actual music. For example, hearing the start of a group on a weak beat adds to metrical tension. *Rhythmic tension*, on the other hand, is unrelated to the meter; it refers to the internal tension created by rhythms using shorter-duration notes and causing a perception of "speeding up" (even when tempo remains constant). For example, a melody in quarter notes that suddenly incorporates several successive sixteenth notes will be heard as having greater rhythmic tension. *Rhythmic contour* refers to the pattern of fast versus slow elements inside a rhythm, and we can imagine a graph displaying rhythmic contour as the rising and falling of note speed (or equivalently, note density) over time.

Hierarchies of Chunks

As is well known in cognitive science, working memory capacity is limited, roughly by the famous "magical number 7±2" study (G. Miller, 1956), and so we use *chunking* to organize information into units that can be held in memory as single objects, thus using up much less working-memory "space". Chunks can form hierarchies so that we can still reason about complex structures, "unpacking" individual chunks or sub-chunks as is necessary when the details are relevant. Naturally, music is heard in terms of these hierarchies of chunks. This isn't to say that an entire piece is represented for a listener as a perfect hierarchical structure: "It is not clear, however, how many levels of hierarchical organization are obvious to musical listeners" (p. 218). We are able to perceive at least certain local portions of the musical hierarchy, and to hear the large-scale structure of a piece as several large sections. Chunks of music are stored in memory and we can most easily recall music from memory by remembering events at the chunk boundaries. This phenomenon naturally is not specific to musical memory, but rather is a general consequence of the nature of human memory and occurs with other types of time sequences in memory.

Snyder gives an example of the way in which children typically learn to speak the English alphabet in a particular rhythmic manner, which suggests the following chunk-based organization (p. 219):

ABCD EFG HIJK LMNOP QRS TUV WXY & Z

Note that each chunk of letters has no more than five letters, less than the “magical number seven”. Snyder suggests that as there are only seven chunks here, the entire alphabet may be represented simply by this sequence of seven small chunks. I suspect there is more structure, however, especially since the rhythm is derived from the “Twinkle, Twinkle, Little Star” melody, which has additional hierarchical structure — see later chapters for more on that melody. For example, “QRS” and “TUV” have the same rhythm (whether spoken or sung) and probably form a mid-sized chunk make up of these two pieces: “QRS TUV”. The following representation shows the additional structure.

{[(ABCD EFG) (HIJK LMNOP)] [(QRS TUV) (WXY & Z)]}

Steve Larson

Three Musical Forces

Steve Larson and FARG developed the creative microdomain Seek Well[2] to study melodic expectation (Larson, 1993b). To reduce the potentially overwhelming complexity of the domain while retaining key features of interest, Seek Well involves melodies in the classical tradition (*i.e.*, Western tonal music) conforming to the following list of restrictions:

- Only one note sounds at once; the sound represented is monophonic.
- All notes have the same duration.

[2] To avoid a possible source of confusion, I should point out that Steve Larson used the term “Seek Well” in two different contexts. As I used the term here, it is the name of a particular microdomain. It can also be used to describe the “Seek Well” project, which refers to a set of computer programs that work in this microdomain (*i.e.*, the “Single-level” and “Multi-level” models described below).

- All notes sound the same except for pitch (dynamics, articulation, etc. are not involved).
- Rests are not allowed (notes come immediately one after another until the melody ends).

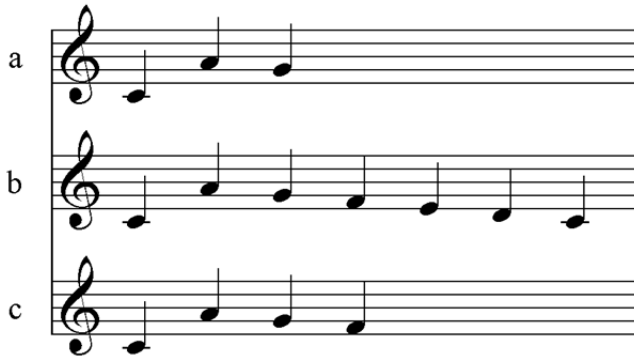

Figure 2.3: Beginnings of typical Seek Well melodies.

Figure 2.3a is a typical melodic beginning in the Seek Well domain. Larson studied typical responses to this cue by asking listeners to sing a completion of any length (Larson 1997). Two common continuations are shown in Figures Figure 2.3b and Figure 2.3c. Note how the response Figure 2.3b implies the listener heard the cue in C-major, while response Figure 2.3c implies F.

Larson's theory of musical forces states that "we tend to hear music as purposeful action within a dynamical field of musical forces" (Larson, 1993a), making an analogy between physical motion through space and the perceived "motion" of a melodic line. The three forces involved are musical gravity, magnetism, and inertia.

Gravity refers to a tendency of notes heard above a stable platform to descend to that platform. For example, given the rising line C-D-E in C-major (Figure 2.4a), gravity would suggest a continuation back down to the stable tonic: C-D-E-D-C (Figure 2.4b). After the alternate beginning C-D-C (Figure 2.4c), however, gravity does not continue to pull the line down past C, because C has been established as a stable base.

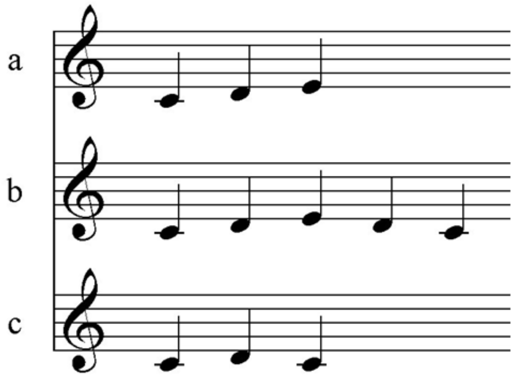

Figure 2.4: Three musical forces.

Magnetism refers to the perceived attraction between notes of unequal stability. For example, in the ascending octave that stops on the leading tone, C-D-E-F-G-A-B (Figure 2.5a), we feel the strong magnetic force of the stable upper octave C “pulling” on the unstable B, leading to the expected completion in Figure 2.5b. As with physical magnetism, the strength of the force is inversely proportional to the distance between two pitches. The B would also be attracted downward to the stable G, but the magnetic pull of the upper C is stronger because it is three times closer, as measured in semitones.

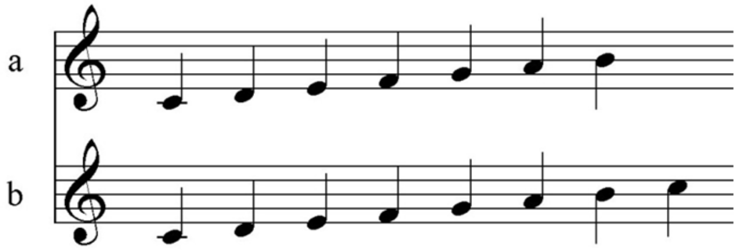

Figure 2.5: Ascending octave.

Inertia is the tendency of a musical pattern to continue “in the same way”. A simple example is the tendency of a moving line to continue moving in the same direction. Given the same beginning as in the gravity example, C-D-E (Figure 2.4a), inertia suggests that instead of falling back down to C, the melody might continue rising through F, G, A, etc., as in Figure 2.5. Musical inertia corresponds to the physical statement “Bodies in motion tend to stay in motion”. Another example of inertia is an Alberti bass, which tends to continue its characteristic pattern once set in motion.

These three forces act continuously on musical lines in a dynamically shifting musical context. A significant part of this context is provided by pairs of *reference alphabets* and *goal alphabets* (Larson, 2004), inspired by Deutsch & Feroe’s (1981) notion of pitch alphabets described above. *Reference alphabets* are tonal pitch sets through which a melody moves, such as the C-major scale in the examples above. *Goal alphabets* are subsets of reference alphabets that serve as stable, goal points for melodic motion, such as the members of the tonic triad in these examples. Thus we can say that musical lines move *through* a reference alphabet *between* notes of a goal alphabet. For instance, the common descending melodic pattern 5-4-3-2-1 in major moves through the major scale reference alphabet (composed of all seven pitch classes of the scale), starting and stopping on the tones of the simple tonic–dominant goal alphabet (consisting simply of 1 and 5).

It is illustrative to reconsider the ascending octave line (Figure 2.5b) in the context of all three forces. The melody starts out at rest on C, the stable base. An external force starts the motion by “pushing” the melody up to D. The external force subsides and the melody begins to be tugged in various directions. Gravity exerts a constant force pulling downwards, back towards C. Magnetic forces are pulling with equal strength up to the E and back down to C. The inertia of the initial push keeps the melody going upwards to E, overcoming the force of gravity. At this point the magnetic pull of G becomes prominent as it is closer than the lower C.[3] Magnetism accelerates the melody up through F towards G, and once it has reached G, it sails past it because of inertia. However, the downward tug of gravity is still noticeable, as is the strong magnetic pull of G, once the melody has ascended to A. Only inertia enables the melody to rise a bit further to B. Once it has reached B, the magnetic force of the upper C is irresistible and overcomes both gravity and the magnetism of G, pulling the melody all the way to C.

The dynamic push and pull of the forces is significant in that it acknowledges some of the complexity of how we listen to music. However, the preceding description may sound overly mechanical — surely the path taken by a melody does not *really* act like an physical object moving through physical fields of force. After all, if this were the case, then an initial C-D melody would always result in the melody traversing the whole octave! Larson's theory avoids this absurd scenario because he claims that many of these descriptions result from *retrospective* hearing (Larson 2004, personal communication). The melody above might well have stopped short of reaching the upper-octave C, but in that case we would have come up with an alternate story describing which forces prevailed at each point in the melody's path. Because the forces interact in a flexible manner, they can provide explanations for alternate melodic continuations. For example, we might imagine that the initiating "shove" was not strong enough to build up inertia to carry the melody past F, A, or even the initial D (Figure 2.6a,b,c below). In each of these cases, the initial inertia-based ascent "ran out of steam", as it was overcome by gravity or magnetism or both. Musical forces can also explain more

[3] In Larson's theory, magnetism is based solely on the distance from a note to the nearest "stable" note. The tonic and dominant scale degrees each induce the same base amount of force. We can imagine an extension to the theory, based on an analogy with physical magnetism, in which the tonic would exert a larger amount of magnetic force than the dominant, as measured at a point equidistant from both. In this system, the attractive force from the leading tone up to the tonic would be much greater than the force pulling the raised fourth degree to the dominant.

extended motion such as that of Figure 2.6d: the initial ascent to A is followed by a descent via gravity and magnetism back down through G, a brief lift back up to F due to the magnetic pull of G, and a final giving-in to gravity to return to the stable platform of C (after briefly dipping below the base due to inertia).

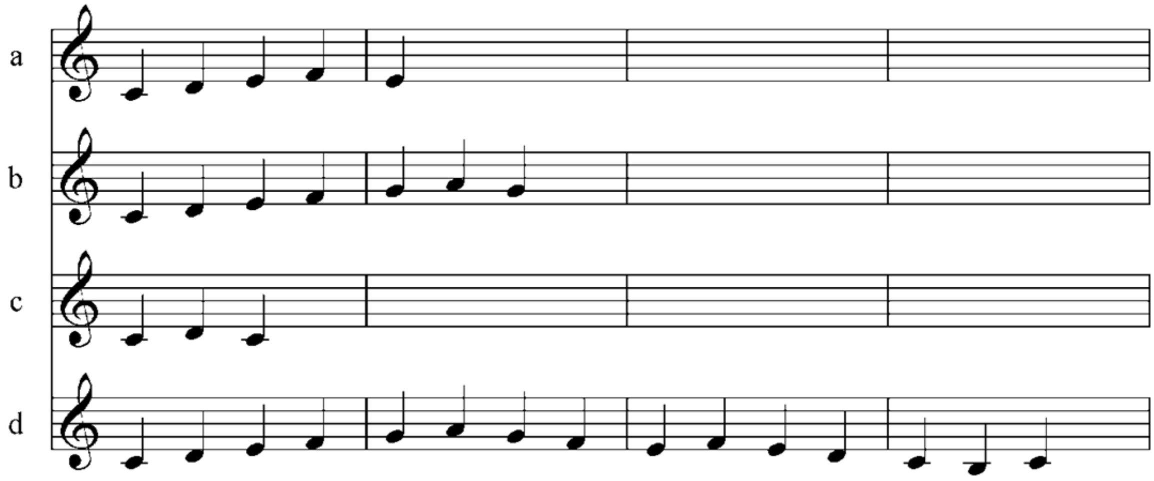

Figure 2.6: Alternate melodic continuations after C-D.

Computer Models Incorporating Musical Forces

Larson's computer models of melodic expectation illustrate his theory by generating explicit predictions of melodic continuations. The *single-level* model is based on the three forces and built-in pairs of reference and goal alphabets. The *multi-level* model additionally uses information provided by an "embellishment hierarchy" (a simplified Schenkerian analysis, as I will explain below).

Single-Level Model

The single-level model takes a melodic beginning as a cue and produces predicted completions of the cue based on gravity, magnetism, and inertia. For each of the three forces, the model analyzes all the pitches in the cue to choose pairs of reference and goal alphabets from a predefined list. Several rules help ensure that the alphabets chosen make musical sense in the context of the force under consideration. For instance, when the model is making a gravity or magnetism prediction, the cue must end on a pitch that is present in the reference alphabet but not in the goal alphabet (that is, the final note of the cue must be heard as unstable so that there is an impetus for continuation). Next, the model examines just the final three notes of the cue and produces predictions based on these notes, the musical forces acting on the notes, and the pairs of alphabets under consideration. These predictions involve motion through the reference alphabet to a stable member of the goal alphabet, providing a sense of completion. The predictions are assigned probabilities according to the strength of the forces as defined by the theory. Thus, the output from the model is a set of different predictions and associated probabilities. Finally, if desired, the resulting predictions can be fed back into the system as new input so that it can predict what will follow each of those possible continuations, generating longer predictions.

Multi-Level Model

The theory of musical forces applies not only to the surface-level notes in a melody, but also to deeper levels of an embellishment hierarchy[4], which is a simplified Schenkerian analysis that can be described in a strict form amenable to computer representation. In Figure 2.7 there are two levels in the hierarchy. Staff A represents the deepest (background) level, while staff B depicts the foreground level, made up of the notes actually heard by the listener. While there are only two levels of hierarchy in this example, there could be several middleground levels in more extended examples. Any discussion in this paper of relations

[4] "Embellishment" is a technical term from Schenkerian analysis. In brief, notes at the musical surface are said to embellish notes that are part of a deeper structure. Larson argues that embellishment is equivalent to the notion of "prolongation" in Schenkerian analysis — see (Larson, 1997b) for discussion.

between the background level and the surface applies equally well when the background or surface is replaced by a middle level.

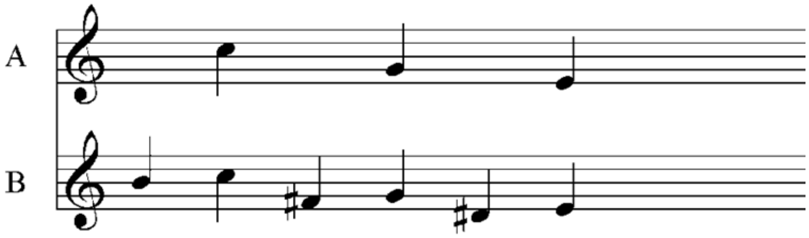

Figure 2.7: A simple melody (below) and an implied background level (above).

The current implementation of the multi-level model requires the user to provide not only the surface notes of the melodic beginning and the key of the melody, but also a specific type of Schenkerian analysis. In this case, both staves A and B would be supplied to the computer, along with a description of the embellishment function of those notes not present in the background level. For instance, the user might describe the initial note B as a "prefix chromatic neighbor" to the following C. This type of analysis differs from a typical Schenkerian analysis in its specificity: the precise embellishment function is explicitly described in the analysis. At the same time, the theory also claims that the analysis must be flexible in describing the embellishment function. For instance, in some musical contexts a description such as "suffix diatonic lower neighbor" might be appropriate, while in other cases the description might be generalized to "suffix diatonic neighbor".

The multi-level model considers each level as an individual melody line and generates a completion of each line. In this example, the *multi*-level model would begin a prediction by applying the *single*-level model to staff A. The single-level model would generate middle C as a likely final note, due to the influence of both inertia and gravity. (The other possibility, discussed below, would be a return to G based on the magnetism between E and G.) Here the reference alphabet in use is the C-major triad, with a goal alphabet made up of only the tonic and dominant. Figure 2.8 demonstrates the partially completed prediction after this step.

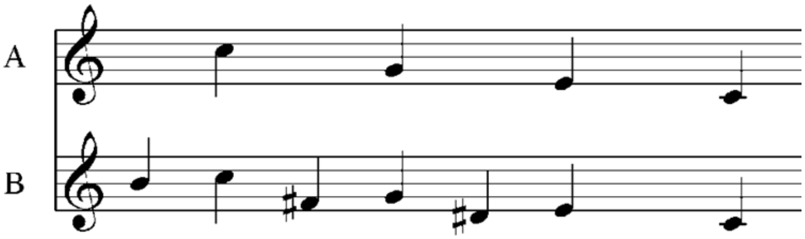

Figure 2.8: Prediction at a higher level.

Having generated a completion for the background level, the multi-level model proceeds to complete the surface level. This step depends on the particular analysis supplied to the system. In the current example, if all the embellishment notes are described as "chromatic lower-neighbor prefixes", the force of inertia indicates that to continue "in the same way" the final C should also have this type of prefix embellishment. The word "chromatic" indicates that the alphabet to be used for the prefix note is the chromatic scale (as opposed to the triadic alphabet used in the background level). Hence, the predicted embellishment note is the note B as shown in the analysis in Figure 2.9. In this diagram, the slurs connect embellishment notes (shown without stems) with notes present at a deeper level in the analysis.

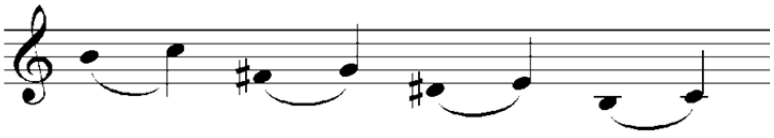

Figure 2.9: Surface-level prediction.

Future Models of Musical Forces

The single- and multi-level models implemented by Larson are only a first approximation to his complete theory, which is described more fully in his book (Larson, 2012). The Seek Well subsection later in this chapter under the FARG heading discusses how the theory of musical forces can be implemented using the ideas in other FARG computer models.

Musical Rhythm

Rhythm plays a particularly primal role in human listening, as well as in the Musicat program. The following sections describe a few pieces of relevant research focused on musical rhythm and cognition. I emphasize that the three sections below just scratch the surface of work on this area. In particular, two important books not described in detail here might be useful in future work on the Musicat project: *The Rhythmic Structure of Music*, by Grosvenor Cooper and Leonard Meyer (1960), and *The Stratification of Musical Rhythm*, by Maury Yeston (Yeston, 1976). Both of these books discuss rhythms with respect to multiple levels of hierarchy, which fits in well with the Musicat's generation of hierarchical grouping structure. Both books discuss relationships between pitch and rhythm — an issue critical to further development of Musicat — and Cooper and Meyer, specifically, discuss many subtleties of rhythm that seem relevant to Musicat.

I suspect that a careful reading of these Cooper and Meyer as well as Yeston would inspire more heuristics and design ideas for future versions of Musicat. The work described below, however, has already proved useful in the development of Musicat.

Dirk-Jan Povel and Peter Essens

Povel and Essens (1985) designed a model of meter perception in a pitch-free domain. The central idea motivating the model is that music causes a sort of ticking "hierarchical internal clock" to be induced in the mind of a listener. Given a series of time points representing note attacks, the model selects the most likely metric structure from a collection of simple meters involving a two-level hierarchy of internal clocks. For example, meters such as 2/4, 3/4, 4/4, and 6/8 can be heard, and these are distinguished based on how accented beats are arranged in a two-level rhythmic hierarchy (*e.g.*, 3/4, a simple triple meter, has a default accent pattern based on a single level with three beats, whereas 6/8 has a different accent structure because it is a compound meter made up of a higher two-beat level combined with a lower three-beat level). Even though there are no differences in the volume or strength of each attack point in the input to the system, it uses heuristics to detect attack points which are perceived to be louder (for instance, attacks at the start or end of a group are perceived to be stronger than those in the middle of a group as a result of perceptual primacy and recency effects). The model selects the best clock simply by comparing each possible clock from a limited set of clocks to the pattern of attacks in the input; the clock that scores the best according to a metric based on shared accents and minimized missing accents is selected as the winner.

Povel and Essens hypothesized that listeners will be able to reproduce a rhythm more accurately when it is amenable to encoding in a simple way with respect to the stresses of the induced internal clock, and verified that this was indeed the case using experiments with human subjects.

EMILIOS CAMBOUROPOULOS

While Povel and Essens give an algorithm for determining musical meter, Combouropoulos (1997) presents a computer model (the Local Boundary Detection Model, or LBDM) that infers, based on local melodic structure, a set of *potential* boundary points. The points discovered by the model might form the starting or ending points of higher-level musical groups, but the output of the model is a list of candidate group boundaries with associated degrees of confidence. Especially relevant to my work is how the model is intended to be used in conjunction with a model involving higher-level structures and relationships such as parallelism:

> "…one has to bear in mind that musically interesting groups can be defined only in conjunction with higher-level grouping analysis (parallelism, symmetry, etc.). Low-level grouping boundaries may be coupled with higher-level theories so as to produce *optimum* segmentations." (p. 279)

LBDM uses two simple rules, based on gestalt proximity, for suggesting boundary points: the Identity-Change Rule (ICR) and the Proximity Rule (PR). ICR considers a set of three successive notes and examines the two parametric intervals formed (the interval between the first two notes and the interval between the second and third notes). Note that these may be time intervals (*e.g.*, durations such as "1 beat" or "2.5 beats"), but the rules are not restricted to the domain of time — they apply to pitch, dynamics, articulations, etc. If the two intervals are different, the rule suggests a boundary point located inside each interval.

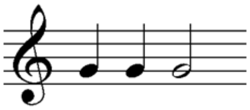

Figure 2.10: No boundaries.

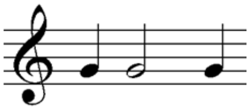

Figure 2.11: Two boundaries.

For example, considering the rules applied to note start times and given the sequence Quarter-Quarter-Half (Figure 2.10), no boundaries are implied because both time intervals are one quarter-note beat long. However, for the sequence Quarter-Half-Quarter (Figure 2.11), the first interval is one beat, while the second interval spans two beats. Two boundary points are indicated; one between beats 1 and 2, and one between beats 3 and 4. Now, the other rule comes into play to adjust the relative strengths of the boundaries: PR gives a stronger preference to the larger interval. In this case, the boundary after the half note (between beats 3 and 4) is stronger. The LBDM model is applied to several different musical parameters and boundary point strengths are summed across these parameters, yielding a strength value for each possible boundary point.

Combouropoulos gives two theoretical applications of LBDM in addition to its use in discovering grouping structure. First, he suggests that the "phenomenal accentuation structure" — that is, the structure of the *perceived* accents in music — can be derived from the model: local maxima in the boundary strengths should correspond to phenomenal (perceived) accents. Note that again here Combouropoulos is careful to point out that there may be ambiguity in the analysis, which should be resolved by using the model in conjunction with other higher-level considerations. Second, given this accentuation structure, various well-formed metrical grids may be superimposed on the structure to find a best match; the best match is a candidate for the true metrical structure of the music.

William Rothstein

In *Phrase Rhythm in Tonal Music,* Rothstein (1989) introduces the concepts of phrase, period, meter, hypermeter, and phrase expansion from a contemporary music theorist's perspective. Several ideas from this discussion are particularly relevant here.

To answer the question "What is a phrase?" Rothstein quotes two theorists: Roger Sessions and Peter Westergaard. Sessions writes that "The phrase is a constant movement toward a goal — the cadence." According to Rothstein, Sessions' concept of the cadence was primarily rhythmic. On the other hand, Westergaard's definition is based on tonal motion. A phrase has these properties:

1. establishes one set of pitches and then
2. moves to a second set of pitches in such a way that
   a. we expect those pitches
   b. we have some sense of when they are about to occur, and
   c. once they have occurred, we know the phrase has gotten where it's going and that no further pitches are needed to complete that phrase.

(Rothstein, 1989, p. 4; Westergaard, 1975, p. 311)

Rothstein's own view (p. 5) is that "a phrase should be understood as, among other things, a directed motion in time from one tonal entity to another… *If there is no tonal motion, there is no phrase*." This is a useful heur*is*tic for ruling out certain musical segments for consideration as phrases. Given a chunk of music *with* tonal motion, however, the notion of "phrase" is still a bit fuzzy. Rothstein notes that some phrases may sound more complete than others, and two phrases may combine to form a larger phrase exhibiting stronger tonal motion.

While Musicat does not try to simulate the theoretical notion of "phrase", these ideas do inform its notion of what constitutes a strong *group*. Rothstein contrasts phrases with other similar-length segments (*e.g.*, 4, 8, or 16 measures, *etc.*) that are still important units but that do not have the tonal motion required in his definition. He calls some of these segments "hypermeasures" if they exhibit rhythmic regularity and an alternation of strong and weak measures. Hypermeasures are a rhythmic, not a tonal, phenomenon. A set of successive four-measure-long hypermeasures contributes to a feeling of a steady pulse analogous to the "1–2–3–4" beats within a measure, but at a higher (slower) level.

The Structure of a Phrase (Motive, Period, Sentence)

Rothstein also discusses the smaller unit of the "motive" (I use the equivalent term "motif" elsewhere but will retain Rothstein's word choice in this section.) He contrasts two different concepts of motive, given by Arnold Schoenberg (1967) and Hugo Riemann. Schoenberg states that "The features of a motive are intervals and rhythms, combined to produce a memorable shape or contour which usually implies an inherent harmony." A motive for Schoenberg can be quite short, and connecting two or more motives can yield a phrase (although Schoenberg's definition of "phrase" does not involve ending with a strong cadence, as required by Rothstein's definition).

Schoenberg then describes two ways in which his phrases can be combined to yield larger structures: the period and the sentence. The difference between the two is not a question of length but of structure. A period is made up of a pair of phrases that are linked: typically, each phrase has a similar beginning, and there is a "question" or tension created by the first phrase that is resolved by the second phrase. The first phrase of such a pair is called the "antecedent" phrase and the related phrase that follows is called a "consequent" phrase. For instance, an eight-measure period might have one of the following forms, where each letter corresponds to two measures:

A B A C or A B A′ C

A sentence, on the other hand, has an initial repetition (or near-repetition) of the first segment, which is followed by a segment of twice the original length that develops the initial motive, such as:

A A′ B

where A and A′ are two measures long but B is a four-measure unit. The segments are in the ratio 1:1:2.

Riemann also describes a process of combining motives into phrases and periods. (Rothstein, 1989, p. 27). However, according to Rothstein, Riemann's motive is primarily rhythmic in nature, where at each level of structure (motive, phrase, and period) we find an unaccented segment followed by an accented segment. That is, an antecedent–consequent relationship is found at every level. This seems extremely strict, but perhaps it captures many common musical structures, and it is interesting that rhythm (especially hypermeasure rhythm) is of such central importance here, as opposed to pitch.

Accent and Meter

I have been asked several times whether I have considered adding accents to Musicat's input; that is, letting it "listen" to something more like a human performance of a score, rather than a plain, uninterpreted score in which all notes have the same loudness. While this would be possible, I would find it less interesting than the current setup that uses unaccented notes, because a human listener should be able to understand metrical structure quite well (at least for simple enough music) even with a very plain performance. Rothstein seems to be in agreement:

> This is perhaps an appropriate point at which to dispel a common misunderstanding concerning the nature of musical meter in general and of hypermeter in particular. It is sometimes thought that metrical patterns are established by means of dynamic accents — by singing or playing certain notes more loudly than others. This is not a necessary condition for the establishment of meter, as an increasing number of theorists have come to realize. In particular, the common view reflects a seriously impoverished conception of musical accent.

Later, Rothstein summarizes Carl Schachter's ideas of meter and accent, and writes: "The beginning of each new span [of time] *receives* an accent in the mind of the listener, just by virtue of its position in time; therefore it need not be *given* an accent by the performer…" Schachter himself writes:

> *…these accents result from the heightened attention attracted by the boundary points of the spans… [E]qual divisions*, once established, can persist in the listener's consciousness without special sensory reinforcement. Indeed, they can persist for a time in the face of strongly contradictory signals…

Successive Downbeats

Typically, we expect measures of music to alternate in metric-hierarchy strength between weak and strong, and this downbeat–upbeat alternation occurs at each hierarchical level (as suggested by Riemann; see above). However, this alternating pattern is sometimes disrupted, and two successive measures may be heard as "downbeats" in terms of the hypermetric structure. Rothstein gives four conditions where this might happen (p. 58):

1. Metrical reinterpretation occurs, in a two-bar hypermeter context;
2. A hypermeter is contracted by dropping a bar (*e.g.*, a four-measure hypermeasure may be shortened to three measures);
3. If a phrase ends on a hypermetrical downbeat, the next phrase might start with another hypermetrical downbeat;
4. If the music consists of melody and accompaniment, then a downbeat might occur in the accompaniment, followed directly by a downbeat in the melody.

How to model the interpretation of hypermetric structures became an important issue in my work on the Musicat program. Musicat, as well as other models of music cognition, could likely benefit from careful consideration of Rothstein's work (the previous four conditions, for example, have not been incorporated into Musicat, but perhaps they should be in the future.)

The previous work described in this chapter was relevant to the aspects of the Musicat program concerned specifically with the domain of music. The remainder of this chapter discusses work not on music specifically, but rather on cognitive modeling in a variety of domains.

FARG

My program draws upon some three decades of work by Douglas Hofstadter and his graduate students and other colleagues in the Fluid Analogies Research Group (FARG). Musicat borrows ideas and architecture components from many previous FARG projects. In the following sections I describe some of the most relevant and influential features of each FARG project instead of giving in-depth descriptions. For detailed summaries of many of them I refer the reader to specific Ph.D. theses or to the book *Fluid Concepts and Creative Analogies* (Hofstadter and FARG, 1995).

There are two major features common to all FARG projects: the use of a very restricted domain and an architecture that supports fluid, creative perception. Projects developed by FARG have made use of *microdomains* to study cognitive mechanisms (Hofstadter 1995). Microdomains are very carefully stripped-down domains involving a much smaller problem space than the typical "real-world" domains seen in fields such as artificial intelligence and machine learning. Problems in these domains, however, are sufficiently rich and deep so that arriving in a human-like fashion to solutions to them requires extremely flexible cognitive behavior. I call the architecture used to model such perception "the FARG architecture" for simplicity, although each FARG project has made its own unique variations on the central theme of simulating parallel subcognitive actions that respond to the conflicting internal pressures of human cognition. In FARG architectures, mental representations are formed, rejected, modified, and eventually settled upon in a flexible and stochastic way. Creative perception in these models is an emergent result of a great deal of seething subcognitive behavior.

I have found the study of many FARG programs helpful in my work on Musicat. Of particular interest are: Copycat, Metacat, Numbo, Tabletop, Letter Spirit, Phaeaco, Seek-Whence, Seqsee, Seek Well, and Capyblanca. The authors of each program are listed below in the headings, but I have omitted Douglas Hofstadter's name because it should be implicit: he was involved in each one as most of these programs were Ph.D. projects of his graduate students.

Copycat (Melanie Mitchell)

Of all the FARG programs, Copycat (Mitchell, 1993) has been probably the most influential for Musicat, because its architecture forms the basis for most of the FARG programs that came later. Copycat involves letter-string analogy problems such as:

"If **abc** goes to **abd**, what does **ijk** go to?"

Most people will answer **ijl**, seeing that the rightmost letter in **abc** was "incremented" and replaced with the next letter in the alphabet, **d**. Applying this rule to **ijk** is trivial. Now consider the following similar-looking problem:

"If **abc** goes to **abd**, what does **xyz** go to?"

The Copycat microdomain is deliberately restricted to a linear (non-wrapped) alphabet, and so the solution involving "wrapping around" from **z** to **a** is disallowed. Subtler analogy-making and greater conceptual fluidity are required to answer this question in a satisfactory manner. Many answers (providing varying degrees of aesthetic satisfaction) are possible besides the forbidden **xya**, such as **xyd**, **xyy**, and the surprising **wyz**. Analogy-making challenges generally do not admit of a single correct response, and it is the process of coming up with an answer, rather than the answer itself, that is the primary object of study.

Copycat's architecture involves three main components: the Slipnet, the Workspace, and the Coderack, as well as a global temperature. I included a Slipnet in the first version of Musicat, but I removed it in the present version (see Chapter 9).

Slipnet

The Slipnet bears some resemblance to a semantic network such as Wordnet (G. A. Miller, 1995), in that it is a network of connected nodes, where nodes represent (the cores of) concepts. Copycat's Slipnet includes approximately 60 nodes: one node for each letter of the alphabet, as well as nodes for concepts such as *letter, successor, predecessor, rightmost, leftmost, sameness,* and *opposite*. Nodes are connected by links, but unlike in traditional semantic networks, each link has a "length" representing the conceptual distance between the nodes. Each node also has an activation level, which varies dynamically as the program runs. The more active a node is, the more it affects the operation of other components of the system. For instance, an active node can cause the program to search in the Workspace for more instances of the concept associated with the node. Active nodes also cause activation to spread through links (as in a connectionist network) to nearby nodes.

Workspace

The Workspace is the locus where perceptual structures are created, modified, and destroyed; I think of it as a model of human working memory. In Copycat, the three strings of the given problem (such as **abc, abd**, and **xyz**) are permanently present in the Workspace. Additional structures are dynamically formed based on concepts found in the Slipnet, such as a bond between the letters **c** and **d**, or groups of adjacent letters, a suggested answer string, and so on. Higher-level structures such as meta-groups made up of groups can emerge as more structures are built in the Workspace.

The activity that takes place in the Workspace is highly fluid. Small bits of computer code called "codelets" (described in the next section) are the source of this activity: many codelets run in a highly parallel fashion, modeling subcognitive activity that is quite unpredictable at the micro scale, as different concepts in memory are activated and perceptual attention rapidly shifts between objects. This activity may look like a seething, chaotic mess: codelets can build and destroy structures in the Workspace, and probabilistic "competitions" between codelets cause structures to flicker in and out of existence rapidly. However, over time, stable structures *emerge* as a result of these interactions. Through the activity of generating structures and destroying rival structures, Copycat performs a sort of heuristic search through possible pathways toward solutions, which is termed the *parallel terraced scan*. Instead of being a brute-force search of the sort familiar in artificial intelligence, or even a heuristic-based search such as *A**, Copycat's search process is a much more cognitively plausible simulation of how human thought might work. For example, Copycat can exhibit phenomena such as getting "stuck" in a problem, or having a "Eureka!" moment when a new, much more satisfying cognitive structure is formed which solves the problem. These are both examples of the flexibility of Copycat's architecture.

Coderack, Codelets, and Pressures

As mentioned above, structures in the Workspace are created by *codelets*, which are small chunks of computer code representing subcognitive activity. Codelets do the work of creating, destroying, modifying, and perceiving structures in the Workspace, and they also affect activations of nodes in the Slipnet. The *Coderack* is a staging area where codelets, after being created, wait until they are selected to run. Some codelets are created and posted to the Coderack by active concepts in the Slipnet, and others are spawned by other codelets.

But why does Copycat use codelets to model perception? Central to Copycat's architecture is the notion of subcognitive *pressures* (*i.e.*, biases, often context-dependent, that affect a host of small decisions made as the program runs), and the codelets on the Coderack are one of the ways in which these pressures are implemented (Slipnet node activations are another of several sources of pressures spread throughout the architecture). Although there are a great many stochastic decisions made during the course of a run and the program's behavior may seem quite chaotic when viewed over a short time scale, various pressures in the program serve to guide the course of a run so that over a longer time scale, the program's perceptual activity has more direction (even though it is still nondeterministic).

Each codelet has an urgency value, so that a high-importance action (such as that recommended by a highly active Slipnet node) can be marked as highly urgent, making it likely to be selected more quickly. The Coderack uses these urgency values as well as information about the current global "happiness" of objects in the Workspace as biases in the stochastic selection of which codelet to run next. (The number of codelets of each type on the Coderack constitutes another source of pressure.) This allows the program to explore multiple different cognitive pathways (the parallel terraced scan mentioned above) at different speeds.

Temperature

The Workspace is in a constant state of flux, with groups and links at many levels of structure being created and destroyed all the time in the search for strong stable perceptual structures. Codelets compute a measure of "happiness" for each perceptual structure, indicating how stable it is and how consistent it is with associated structures. As the overall amount of happiness increases, the system's global temperature drops. The temperature affects how urgencies are used in selecting which codelet to run next, it affects the probabilistic choices that are made by individual codelets, and it also affects the types of codelets added to the Coderack: when temperature is high, more "breaker" codelets are posted to break up existing structures to make room for new ones, whereas when the temperature drops, codelets tend to behave collectively in a way to make the Workspace more stable.

Temperature varies inversely with the overall Workspace happiness, but it also gradually drops over time in order to force the Workspace to stabilize. This controlled descent of temperature is reminiscent of simulated annealing, although it is not deterministic or monotone: a very low level of happiness in the Workspace can cause the Temperature to jump back up to a high level.

METACAT (JAMES MARSHALL)

Metacat, the successor to Copycat, added a degree of self-watching to Copycat's architecture. For example, whereas Copycat might get "stuck" re-discovering the same unsatisfying structures multiple times in a single run, Metacat has some ability to notice the repetition and purposefully avoid it. I haven't used any of these particular new mechanisms from Copycat, but the general idea of self-watching is important and comes into play in one aspect of my program, which was also inspired by Tabletop (see below).

Metacat introduced many new ideas not present in Copycat. In addition to the components present in Copycat such as the Workspace, Coderack, and Slipnet, Metacat's architecture also includes an "Episodic Memory" — which augments Copycat's long term memory, and it also includes a "Temporal Trace", and "Themespace", which provide a meta level that allows self-watching. For more detail please see Marshall's thesis — in this section and in the description of other FARG programs, below, my intent is not to describe the programs in depth, but rather to give credit to particular aspects of these programs that inspired me for one reason or other. In general, I refer the reader to the relevant books or dissertations for more information on each of these programs.

NUMBO (DANIEL DEFAYS)

Numbo is one of the simplest (but most elegant) FARG projects. It makes obvious the difference between raw, bottom-up perception and more directed, top-down perception. As is the case with Metacat, I did not use any of Numbo's elements explicitly, but I found the ideas in it interesting and helpful in clarifying my thinking.

The Numbo program solves puzzles in which a set of positive integers, called "bricks", is provided along with a target positive integer. The goal is to find a mathematical expression that equals the target. Such expressions are built up using the bricks and a small set of arithmetic operators (+, −, ×) and possibly parentheses. For example:

Bricks: $\mathbf{5, 3, 13, 2, 8}$

Target: $\mathbf{42}$

One solution is the expression $\mathbf{8 \times 5 + 2}$, although others are possible[5]. Of course, solving the puzzle would be easy for a brute-force search algorithm, but the point of Numbo is to solve the puzzle this in the same way as humans do. Numbo has preferences to notice certain arithmetic properties of numbers and ways of combining them that are salient for people. In this example, the arithmetic fact $\mathbf{8 \times 5 = 40}$, in the context of the bricks and target above, might be much more salient than the fact $\mathbf{50 - 8 = 42}$. Thus it might be less likely for the program to come up with the alternate solution $\mathbf{(13 - 3) \times 5 - 8.}$

Numbo, like Copycat, uses a memory of permanent knowledge with spreading activation called the "Pnet" (essentially a simpler Slipnet), a Workspace, and a Coderack with codelets. During a run, bricks are combined stochastically in the Workspace to form new blocks. For instance, $\mathbf{8}$ and $\mathbf{5}$ might be combined quite easily using multiplication to form a block with value $\mathbf{40}$, and the proximity of $\mathbf{40}$ to the target $\mathbf{42}$ would strengthen the block. On the other hand, blocks such as $\mathbf{13 \times 8}$ $(= 104)$ with no obvious relationship to the target would be less likely to form. Although it would be possible for such a block to form as a result of the constant stochastic activity of the system, if formed it would likely be quickly broken apart by "dismantler" codelets.

[5] Other solutions are $\mathbf{3 \times 13 + (5 - 2)}$ and $\mathbf{(3 \times 13) + (8 - 5)}$. The solution above, however, was the most obvious to me, because it is less natural to me to multiply by $\mathbf{13}$. However, someone who saw the target, $\mathbf{42}$, as $\mathbf{3 \times 14}$ might realize that it is possible to get close to the target with $\mathbf{3 \times 13}$, leading to these solutions.

The most interesting aspect of Numbo for me is the way that while it has a significant stochastic aspect, allowing blocks to form and disintegrate quite wildly, it also acts in a goal-oriented way, despite the randomness. This is true of all the programs based on the FARG architecture, but I found that this aspect of the architecture even more evident to an observer in Numbo than in other FARG programs. Also, in this domain it seems particularly easy to see how FARG programs differ from programs using classic heuristic search, such as is used in the General Problem Solver (Ernst & Newell, 1969):

> One would be at pains to cast Numbo's method of proceeding in terms of search in a problem space; after all, no central control is involved, and rote knowledge is represented in a task-free way. A goal is just one of many features of a situation. No systematic exploration takes place in Numbo. Taken together, competition between codelets and the element of randomness allow the system to jump from one idea to another, sometimes in a chaotic-seeming (but rather humanlike) way. (Defays, in Hofstadter & FARG, 1995, p. 149)

In Numbo, as other FARG projects, "search" is replaced with stochastic juggling of mental objects, pushed by pressures to examine "interesting" combinations. This mental manipulation often results in good ideas and solutions, but sometimes fails to find a satisfying answer, but sometimes finds a surprising or even a quite aesthetically-pleasing answer — much as happens in human cognition.

Tabletop (Robert French)

The Tabletop program solves problems in a surprising and initially silly-sounding microdomain: touching objects on a coffehouse table. A problem starts with a representation of the two-dimensional surface of a table, with various arrangements of silverware and cups and glasses and such items on either of two sides of the table in particular locations. We imagine that two people (Henry and Eliza) are seated across from each other at the table. The task is simply this: Henry touches an object on the table, and Eliza (the program) is supposed to mirror this action by "doing the same thing". For example, if each side of the table has a cup and a spoon, and if Henry touches the cup on his side of the table, Eliza would probably touch the cup on her side. This seems very simple, but the challenge quickly grows very tricky and subtle when different objects in different arrangements are present on the two sides. The deep problem attacked by the Tabletop project is actually context-dependent analogy-making between groups of objects.

Tabletop's architecture is closely related to that of Copycat, but it introduces many new ideas to handle difficulties that cropped up in the Tabletop domain. The new ideas include: less-destructive competition between structures, a solution to the "urgency explosion" in the Coderack, and a more sophisticated computation of temperature.

Urgency Explosion

While he was developing Tabletop, French noticed that "one codelet of a given urgency can spawn many codelets of the same (average) urgency" (French, 1992), which resulted in a sudden decrease in relative probability of being selected for other codelets on the Coderack. This had not been a problem in Copycat because there, unlike in Tabletop, each codelet could spawn only one follow-up codelet. French's solution was to compute an adjusted urgency that decreased exponentially with the number of generations that had passed since an original ancestor started spawning descendant codelets. In other words, fresh, new codelets representing newly-generated pressures or newly-perceived elements of the scene were favored over stale, old, and potentially out-of-date codelets.

Less-Destructive Competition

In Copycat, structures that lost competitions with other structures were destroyed instantly. This strategy was satisfactory for Copycat and Numbo, where the structures in the Workspace were rather small and could be easily reconstructed stochastically as long as the proper pressures were present. In contrast, Tabletop allowed losing structures to remain in the Workspace, although they "faded out" a little bit. The Tabletop Workspace was thus a collection of many overlapping structures of various strengths, even though many of the overlapping structures in this Workspace would have been mutually exclusive in a Copycat-style Workspace. A new architectural component called the *Worldview* was introduced by French, which maintained a single non-contradictory mapping between objects and groups on the table. Old "faded" structures in the Workspace were allowed to compete to return to the Worldview. This obviated the need to reconstruct large structures from scratch, which would have been extremely unlikely, and provided a more psychologically plausible way for large structures to compete with each other (in a way that is reminiscent of the Necker Cube or Vase/Faces illusions.)

Temperature

Tabletop's calculation of temperature differs from Copycat's in that Tabletop's temperature acts more locally and "indirectly". It takes into account the strengths of structures in the Worldview, but ignores some objects completely, in much the same way that Capyblanca (see below) ignores portions of a chessboard. The factors involved in Tabletop's temperature computation are:

1. quality of the Worldview, composed of the following:
   - sum of the strengths of all correspondences;

    - degree of coherence in how correspondences "bunch together";
    - degree of parallelism between correspondences;
2. average rate of change in Worldview quality;
3. number of changes in the Worldview;
4. number of competitions.

The first factor has the largest effect on temperature. The other three regulate the temperature in a way that provides simulated annealing, and, indirectly a limited kind of self-watching.

Of particular interest is how the notion of "slippage" was handled in Tabletop. Whereas in Copycat only one overall analogy was being formed, Tabletop might have two or more unrelated sets of correspondences, and concept slippages in one set might be unrelated to the other. However, there is only one copy of each Slipnet node, even though in the case of multiple, simultaneously present correspondances, it seems that multiple copies of each node might be necessary. French refers to this problem of the Slipnet's state being applied to wildly different contexts as "the problem of single nodes with multiple activations". A particular Slipnet state might be relevant to one part of the problem (such as one spatially isolated location or one particular level of grouping hierarchy), but not to another part of the problem. He considers possible solutions to this dilemma, and settles on the idea of reconstructing activations of Slipnet nodes each time the program switches from one context to another.

Letter Spirit (Gary McGraw and John Rehling)

The program Letter Spirit aimed to model "the creative act of letter-design" (Hofstadter & Fluid Analogies Research Group, 1995). Although I didn't use any specific features of Letter Spirit in my work, I was influenced (as mentioned in Chapter 1) by the program's focus on modeling creativity. Letter Spirit has separate modules for designing new letters and for examining previously-designed letters, and this separation of creation and perception was appealing to me. Musicat does not create music, but it does perceive music, just as one module of Letter Spirit was designed to perceive letter shapes.

Phaeaco (Harry Foundalis)

The Phaeaco program solves Bongard problems, which are visual puzzles that require the puzzle-solver to find a visual concept that is exhibited by each image in a set of six images, and that is not exhibited by another set of six images that are known to be non-examples of the concept. The most significant idea in Phaeaco that I incorporated in one version of Musicat was the idea of concept formation in long term memory, including the ability to persist the long-term memory across multiple runs of the program. Phaeaco can learn visual concepts and recall them while solving future problems. One version of Muiscat (not the latest version, however) incorporated this notion and could learn musical motifs and then recognize them in future runs.

Seek-Whence (Marsha Meredith) and Seqsee (Abhijit Mahabal)

The program Seek-Whence, and its descendant, Seqsee, were designed to understand sequences of numbers and to make predictions about which numbers would come next in the sequence. For instance, consider the following sequence, from *Fluid Concepts and Creative Analogies* (Hofstadter & Fluid Analogies Research Group, 1995):

1 2 2 3 3 4 4 5 5 6…

This sequence might be understood as consisting of the number "1" followed by a series of pairs of numbers, continually increasing, as so:

1 (2 2) (3 3) (4 4) (5 5) (6 …

However, the initial "1" stands out as a "glitch" in the pattern, so people (or the programs Seek-Whence and Seqsee) might prefer to see the pattern as a series of groups in which each group consists of a number followed by its successor, as follows:

(1 2) (2 3) (3 4) (4 5) (5 6) …

When I was first reading *Fluid Concepts and Creative Analogies*, I read some of the number sequences at the start of the book and imagined hearing them as sequences of musical notes (imagine mapping the number "1" to the note "C", "2" to "D", and so on. Therefore, I was please to read Hofstadter's comment that this second parsing of the number sequence above "sounds like" a Chopin piano prelude: Op. 28, No. 12 (Hofstadter & Fluid Analogies Research Group, 1995, p. 79).

Seek-Whence and Seqsee gave me great confidence that I could write a program that used similar techniques as they did, but in the domain of music, because of the number-to-note mapping mentioned above. The program (Musicat) would require knowledge specific to the music domain (the notes "C" and "G" are related in a special relationship that does not exist between the numbers 1 and 5, for instance). Also, Seek-Whence and Seqsee incorporate the notion of expectation, which was especially important in the early versions of Musicat.

Several specific features of Seqsee were influential in Musicat's development. Seqsee made less use of the idea of a Slipnet than most earlier FARG programs, and in the latest version of Musicat I did away completely with the Slipnet. Seqsee introduced a novel "stream of thought", and although I did not use this idea in Musicat, I found the idea appealing and it subtly influenced \how I thought about the constant progression of musical time. Finally, the user interface of Seqsee, with bar graphs showing the distribution of various codelets on the Coderack, inspired some of the graphs I made during development.

SEEK WELL (STEVE LARSON)

Steve Larson implemented two versions of his Seek Well program — the single-level model and the multi-level model described earlier in this chapter. However, he also gave a set of instructions for implementing his theory of musical forces more completely using a FARG-style architecture. His instructions describe in some detail how the model should generate predictions of melodic completions by automatically generating a hierarchical embellishment analysis including information about the key, mode, and meter, and using this analysis to determine reference alphabets and goal alphabets and to make predictions based on the three musical forces. Larson writes that a complete implementation of his model, based on a FARG-style program that takes all the instructions into account in parallel, "will most likely be quite informative about music cognition" (Larson, 2012).

This parallelism is critical; without it the model is limited. Larson's own implementation of his multi-level model works only in a serial fashion, generating predictions at the deepest structural levels first. However, there are cases where the surface levels can have a more direct impact on predictions. In general, musical forces working at different structural levels can conflict with one another, which shows the need for the model to be able to deal with such conflicts when making melodic predictions. Parallelism would provide a way to handle these conflicts. Additionally, Larson's experimental work shows how other important musical factors such as implied metric structure and implied harmony affect how a human listener hears a melody and generates possible extensions of it; these, too, could be considered in parallel with other factors. Finally, Larson's implementation required as input not only a melody, but also a human-generated embellishment analysis thereof. The model did not provide a way to generate the necessary embellishment analysis automatically, but ideally it would also be generated in parallel with the rest of melodic processing and expectation-generation. Larson writes:

> A complete specification of the theory might describe how hierarchical representations of embellishment structure may be built up and internally represented. (Larson 2004)

My initial version of Musicat (also called "Seek Well") was an attempt to implement a FARG-style model according to Larson's theory of musical forces. However, I ran into obstacles caused by the need to automatically generate the embellishment structure (this is essentially the problem of generating an automatic Schenkerian analysis). Although I believe that a FARG-style architecture would prove useful in generating such an analysis, I removed the embellishment-structure code from my program early during the development of Musicat (see Chapter 9); this important part of Larson's theory of musical forces is out of the scope of my project, which focuses more on grouping structure. Similarly, the initial determination of key and meter, which was a key focus in Larson's Seek Well domain, is not considered in the current version of Musicat, which is given that information "for free". Some of the ideas from Larson's work on Seek Well do appear in Musicat, however, such as the reference and goal alphabets described above, and the general idea of allowing multiple musical interpretations and expectations to compete with each other.

Capyblanca (Alexandre Linhares)

Computer mastery of the game of chess was once a sort of Holy Grail of artificial intelligence research, but Deep Blue's triumph in 1997 over world champion Garry Kasparov using brute-force lookahead search may be seen as a hollow victory for AI: Deep Blue's success in playing chess has nothing to do with playing chess in the way people do. A computer employing this sort of search strategy may play an extraordinary move (such as a wild-looking queen sacrifice), but it would not find the move any more exciting or surprising than any trivial defensive pawn move. Human masters sometimes play spectacular moves for fun when a more modest move would win a game more quickly. Ironically, while people are able to appreciate the aesthetic value of computers' brilliant chess moves, but the computers themselves remain totally oblivious. Today. computer chess feels pretty much like a "solved" problem to the AI community, and interest has turned to the game of Go, in which humans are still far superior to computers (just as was the case for chess through the mid-1980s), although some groups are making rapid progress.

There is still reason to consider chess as an interesting domain of cognitive-science research, however. Alexandre Linhares used a FARG-style architecture to model perception in the game of chess (Linhares, 2008). His program, called Capyblanca (a reference to former world champion José Raúl Capablanca), is unusual for chess computing in that it does not carry out a brute-force lookahead search of the game tree. Instead, it uses ideas about chess intuition (Linhares, 2005) to simulate how a human player might visually scan the chess board, noticing salient aspects of the position, remembering relevant positions from previous games, and forming a long-range plan. Capyblanca not only constitutes an alternative to chess programs such Deep Blue, but it also challenges some traditional thoughts on expert knowledge in chess (Linhares & Freitas, 2010). Note that the current version of Capyblanca does not actually make chess moves; instead, it simply perceives salient aspects of chess positions and forms large-scale plans. Perhaps a future version of Capyblanca will play human-like chess games.

Capyblanca offers a way to model the following features of human chess:

- focusing on salient features in a position;
- investigation of key moves, avoiding brute-force search;
- recognition of strategic themes from past games;
- generation of broad expectations for upcoming game positions;
- conscious ideas emerging from subcognitive processes.

Capyblanca's architecture was inspired by earlier FARG projects such as Copycat and Phaeaco, but it has some unique aspects, including the following:

- **A new way to calculate temperature for a structure**. This involves relevance to current goals, ability to imagine a structure (such as a queen in a position where it isn't on the current board), and the "evidence" for a particular structure's existence. Temperature is locally computed for each structure, as in Tabletop.
- **Starting from an empty Workspace**. The Workspace in Capyblanca starts out completely empty, and perceptual structures describing the relevant features of the chess position are gradually built up at several different levels of description in parallel, based on what the program "looks" at (see next bullet). In most other FARG projects, the Workspace starts out including input elements (in Seek Whence, Seqsee, and Musicat, the Workspace is

initially empty but input elements are automatically added to the Workspace over time).

- **Simulation of eye saccades**. Whereas most chess programs maintain a list of all the pieces on the board, Capyblanca "sees" only certain pieces, depending on random simulated eye movements. Its focus of attention can move randomly and can also be attracted to nearby vaguely-perceived structures or salient parts of the board. This process of shifting attention is quite human-like and constitutes a "visual" top-down pressure to focus on certain aspects of a board rather than on others.