DISSONANCES: BRIEF DESCRIPTION AND ITS COMPUTATIONAL REPRESENTATION IN THE RTCC CALCULUS

Salim Perchy, Gerardo Sarria

DISSONANCES: BRIEF DESCRIPTION AND ITS COMPUTATIONAL REPRESENTATION IN THE RTCC CALCULUS

Salim Perchy
AVISPA Research Group
Universidad Javeriana - Cali
ysperchy@cic.puj.edu.co

Gerardo M. Sarria M.
AVISPA Research Group
Universidad Javeriana - Cali
gsarria@cic.puj.edu.co

ABSTRACT

Dissonances in music have had a long evolution history dating back to days of strictly prohibition to times of enricheness of musical motives and forms. Nowadays, dissonances account for most of the musical expressiveness and contain a full application theory supporting their use making them a frequently adopted resource of composition. This work partially describes their theoretical background as well as their evolution in music and finally proposing a new model for their computational use.

1 THE CONSONANCE AND THE DISSONANCE

1.1 Cognitive Definition

The term *Consonance* in music is considered, from a psychological stance, a sound (i.e. interval in chord or arpeggio) that emits a sensation of ease or relaxation of the ear, something enjoyable. In contrast, *dissonance* is the opposite sensation, something confusing or aggressive to the ear [1].

Although the difference between this two terms may be well-defined with respect to tonal music, their meanings are profoundly attached to and varies depending on the culture, the music genre, and even the spoken language. For this, their cognitive definitions are relative and other factor come to fulfill their descriptions [2].

1.2 Physical Definition

Physically speaking, a *dissonance* may be conceived as the union of the acoustic waves that try to *destroy* themselves, *consonance* being the opposite physical phenomena[3].

[1] This interval was considered dissonant in early counterpoint

SMC 2009, July 23-25, Porto, Portugal

Interval	Ratio
1^{st}	1:1
8^{ve}	2:1
4^{th} [1]	4:3
5^{th}	3:2

Table 1. Perfect consonant intervals and their ratios

Mathematically speaking, the concept of consonance and dissonance is attached vastly to the **ratios** between the sound waves that conform the sound being executed. This ratio, for a consonance consideration, should be with low numbers as observed in table 1

Figure 1. Perfect Consonant Intervals

In tonal music, the intervals considered consonant may be divided in *perfect* (figure 1) and *imperfect* (figure 2), *perfect* ones being simpler ratios than their counterpart. Table 2 shows some imperfect consonant intervals and their ratios. On the other hand, the dissonances are the intervals excluding the ones mentioned before (i.e. Minor 2^{nd} and 7^{th}).

Interval	Ratio
Major 3^{rd}	5:4
Minor 3^{rd}	6:5
Major 6^{th}	5:3
Minor 6^{th}	8:5

Table 2. Imperfect consonant intervals and their ratios

The reader may note that the concrete definition and dif-

Figure 2. Imperfect Consonant Intervals

ferentiation of consonances and dissonances given above are purely tonal and based on western (common practice) music. Throughout time, the concept of consonance has evolved constantly giving way to before-considered dissonances being reclassified as consonances because of their frequent use, as noted in [4].

The choice of this particular vision, as the reader will see in the model proposed, of dissonance is purely convenient because of its strong theory background and documented use.

1.3 Brief historical evolution

The dissonance, as the majority of the compositional tools, has seen an evolution in which it has enriched itself from both theory and practice. Each important age in music history has contributed with its convention of use and theoretical and stylistic meaning, some more than other but always keeping a concrete advance.

Initially, the dissonance was non-existent in the sense of the avoidness of its use because the young age of harmony and its theory concepts targeted to enrich the consonance compendium.

In ancient Greece, the dissonance was not explored as a melodic resource although its meaning was already established, but yet, the Greeks considered the tritone (3 density chord) a disturbing sound, this being the base of the harmony used in the classical period. [5]

Farther ahead, the baroque age would consider intervals like the 3^{rd} and the 6^{th} consonant and extend their use to not just independent entities but also as artistic expressive means. At the dawn of the classical period the first direct dissonance is explored as a form with a well defined objective, this was the V^7 or *seventh dominant* chord.

At the arriving of the romantic and nationalism ages the use of dissonances was widely frequent and have defined purpose and process, primary helping the composer explore new concepts in the harmony theory.

With the establishment of the contemporary music, the dissonances have now the role of directing the harmony in actual composition, musicologists and composers turn to them in a daily basis.[5]

2 THE FUNCTION OF THE DISSONANCE

For an accurate computational definition of the dissonance we have to define its function, use and purpose in music. Also we have to bear in mind that the function is **relative** to each age or musical genre, but is keeps a stable base theory and form of use.

2.1 Harmonic Function

In defining a concrete function in the process of dissonance, there also has to be a definition of an inherent characteristic of musical harmony. The music literature, as normal literature, possesses moments of tension and relaxation, this simple characteristic allows to provide a purpose and at the same time a tool for the description of a given melody.

In the case of the dissonances, their harmonic function is no other than to create tension or confusion to the listener, this objective is more remarked seeing its aggressive character depending on the composer's thoughts, so it is this that sets its goal to the conversion of a melody to a point of obscurity and discomfort to the ear[3].

From the aforementioned, modern music has expanded its use not just to a tension-relaxation characteristic but also to an independent and complete auditive element that might represent an idea on its own[6].

2.2 Usage

The cycle of *tension-relaxation* can be seen as a sequential process that happens over time, each of the parts of this process may be seen as a well-defined component or agent (process with goal). The cycle is generally seen as this flow [7]:

$\boxed{\text{Preparation}} \rightarrow \boxed{\text{Dissonance}} \rightarrow \boxed{\text{Resolution}}$

Each stage contains unique and shared characteristics:

Preparation : Its function, as its name indicates, prepares the listener to the confusion of tension that the dissonance may generate in the melody. Generally, this preparation carries a harmonic line corresponding to its tonality.

Dissonance : In this stage, the dissonance or dissonances are produced, often in weak rhythmic beats and in strong ones depending on its relevance and sonority.

Resolution : Here, the dissonance need to move to a state of resolution or relaxation, it is here that the dissonance is carried to a more pleasing form, often taken to the main tonality on long beats.

The most evident and early example of this above process can be observed in the progression I – IV – V^7 – I (figure 3), in this the preparation consists of the first 2 chords and the dissonance is caused by the *dominant seventh*, lastly resolving in the tonic chord `C`.

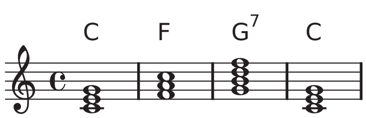

Figure 3. Dissonance using V^7

The usage of dissonances expands when the repertory of dissonant chords grow over time, this was also a crucial aspect of the evolution of this technique, chord like the *augmented sixth* or *Neapolitan sixth* make a wider space of sonority. An example can be seen a figure 4.

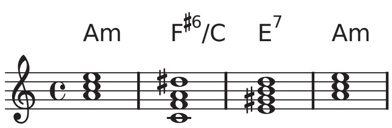

Figure 4. Dissonance enriched with the *German augmented sixth*

The next evolution in the use of dissonances is considering the resolution step as omissible or indefinitely postponed, often leaving many accumulated to an eventual resolution, *Frédérick Chopin* and *Richard Wagner* were amongst the main developers of evolution[5]. The illustration of these can be seen in figure 5.

Figure 5. Dissonance without resolution

Modern music refines the concept of dissonance and transforms it when declaring it a unique entity, self-described and self-functional, making dissonances a complete form without the need of preparation or resolutions. A very recognized composer using this concept was *Claude Debussy*.

Figure 6. *Claude Debussy - "La fille aux cheveux de lin"*, measures 8 and 9. Dissonance as individual entity

3 THE REAL-TIME CONCURRENT CONSTRAINT CALCULUS

Concurrent constraint programming (`ccp` [8]) is a model for specifying concurrent systems in terms of constraints. A *constraint* is a formula representing partial information about the shared variables of the system. Examples of constraints are: $pitch_1 = 60$ or $pitch_2 > pitch_1 + 2$; If variables $pitch_1$ and $pitch_2$ are in the domain of MIDI values these constraints specify that $pitch_1$ must be C and $pitch_2$ must be at least a tone higher than $pitch_1$. The information about the shared variables resides in a *store*, which is, in fact, the conjunction of all the constraints applied to the variables. This store can be accessed by *agents* (processes who interact with the store) with two basic operations: **ask** and **tell**.

The *Real-Time Concurrent Constraint Calculus* (`rtcc` [9, 10]) is an extension of `ccp` developed to specify reactive systems with real-time behaviour. In reactive systems time is conceptually divided into *discrete intervals* (or *time units*). In a time interval, a process receives a stimulus from the environment, it computes (reacts) and responds to the environment. The computational processes of `rtcc` are summarized in table 3.

$$
\begin{array}{lcl}
P, Q, \ldots & ::= & \mathbf{tell}(c) \mid \sum_{i \in I} \mathbf{when}\ c_i\ \mathbf{do}\ P_i \\
& | & P \parallel Q \mid \mathbf{local}\ x\ \mathbf{in}\ P \\
& | & \mathbf{unless}\ c\ \mathbf{next}\ P \\
& | & \mathbf{catch}\ c\ \mathbf{in}\ P\ \mathbf{finally}\ Q \\
& | & \mathbf{next}\ P \mid \mathbf{!}P \mid \star P
\end{array}
$$

Table 3. `rtcc` Processes

Intuitively, the process **tell**(c) adds constraint c to the store within the current time unit. The ask process **when** c **do** P is generalized with a non-deterministic choice of the form $\sum_{i \in I}$ **when** c_i **do** P_i (I is a finite set of indices). This process, in the current time unit, must non-deterministically choose one of the P_j ($j \in I$) whose corresponding guard constraint c_j is entailed by the store, and execute it. The non-chosen processes are precluded. Two processes P and Q acting concurrently are denoted by the process $P \parallel Q$. In one time unit P and Q operate in parallel, communicating through the store by telling and asking information. The process **local** x **in** P declares a variable x private to P (hidden to other processes). This process behaves like P, except that all information about x produced by P can only be seen by P and the information about x produced by other processes is hidden to P. The weak time-out process, **unless** c **next** P, represents the activation of P the next time unit if c cannot be inferred from the store in the current time interval (i.e. $d \not\models c$). Otherwise, P will be discarded. The strong time-out process, **catch** c **in** P **finally** Q, represents the interruption of P in the current time interval when the store can entail c; otherwise, the execution of P continues. When process P is interrupted, process Q is executed. If P finishes, Q is discarded. The execution of a process P can be delayed one time unit with **next** P (P will be activated in the next time interval). The operator "**!**" is used to define infinite behaviour. The process $\mathbf{!}P$ represents $P \parallel \textbf{next}\ P \parallel \textbf{next}(\textbf{next}\ P) \parallel \ldots$, (i.e. $\mathbf{!}P$ executes P in the current time unit and it is replicated in the next time interval). An arbitrary (but finite) delay is represented with the operator "$\star$". The process $\star P$ represents an unbounded but finite $P + \textbf{next}\ P + \textbf{next}(\textbf{next}\ P) + \ldots$, (i.e. it allows to model asynchronous behaviour across the time intervals).

We write $\prod_{i \in I} P_i$, where $I = \{i_1, \ldots, i_n\}$ to denote the parallel composition of all the P_i, that is, $P_{i_1} \parallel \ldots \parallel P_{i_n}$. A bounded replication and asynchrony can be specified using summation and product. $!_I P$ and $\star_I P$ are defined as abbreviations for $\prod_{i \in I} \textbf{next}^i P$ and $\sum_{i \in I} \textbf{next}^i P$, respectively. For example, process $!_{[m,n]} P$ means that P is always active between the next m and $m + n$ time units.

The following simple example illustrates a computational model in `rtcc`:

In the case of changing pace of a song's natural timing such a *ritardando*, this behaviour can be modeled as:

$$
\begin{array}{l}
\mathbf{!}(\textbf{when}\ ritardando = \texttt{true}\ \textbf{do next tell}(bpm = 60)) \\
\quad \parallel \textbf{catch}\ ritardando = \texttt{true}\ \textbf{in}\ \mathbf{!}(\textbf{tell}(bpm = 150))
\end{array}
$$

Intuitively, this process states that the speed of a quarter note (or crotchet) will be 150 (with process $\mathbf{!}(\textbf{tell}(bpm = 150))$) until a *ritardando* signal is given (a presence of constraint $ritardando = \texttt{true}$ in the store). In the case of the signal is given, the process $\mathbf{!}(\textbf{tell}(bpm = 150))$ is interrupted and the speed will change to 60.

4 COMPUTATIONAL REPRESENTATION OF DISSONANCES

Since the dissonance phenomena in music can be seen as an ordered sequence of processes (as shown above), we can express it using concurrent agents that synchronize each other through signals (constraints) that are global to the whole system. Each agent may represent each phase in the dissonance process and also delay its execution until the previous (dependant) phase has been carried out and signals the system to continue the sequence onto the next phase. The model we propose is the following:

$$
\begin{array}{rl}
Conductor_{[n,m]} \stackrel{\text{def}}{=} & Musician \parallel Cycle_{[n,m]} \\
& \parallel \mathbf{!}(\textbf{tell}(go = 1)) \\
& \parallel \star(\mathbf{!}\textbf{tell}(stop = 1)) \\
& \parallel \mathbf{!}(\textbf{unless}\ stop = 1 \\
& \qquad \textbf{next when}\ end = 1 \\
& \qquad\qquad \textbf{do}(Musician \parallel Cycle_{[n,m]}))
\end{array}
$$

$$
\begin{array}{rl}
Cycle_{[n,m]} \stackrel{\text{def}}{=} & \star\,(\textbf{tell}(prep = 1) \\
& \quad \parallel \star_{[1,n]}(\textbf{tell}(diss = 1) \\
& \qquad \parallel \star_{[1,m]}(\textbf{tell}(res = 1))))
\end{array}
$$

The main entry point of the model is the agent *Conductor*, this agent will activate the *Musician* and a process *Cycle* to motivate a dissonance. It also gives a signal to the musician for starting the melody (**!**(**tell** (go = 1)) and eventually (some time in the future) it will give another signal to make the musician stop producing music ($\star$(**!tell** ($stop$ = 1))). Additionally, if the stop signal has not already given and the musician ends a dissonance, *Conductor* will activate the *Musician* and the process *Cycle* again (this could be seen as a loop for the musician to continue playing the melody and eventually to perform a dissonance until the $stop$ signal).

The *Cycle* process posts the signals for each stage of the dissonance. Parameters n and m bound the time to change from one stage to the next.

The agent *Musician* is defined as follows:

$$
\begin{array}{rl}
Musician \stackrel{\text{def}}{=} & \textbf{when}\ go = 1\ \textbf{do} \\
& \quad \textbf{catch}\ prep = 1\ \textbf{in}\ Melody \\
& \qquad \textbf{finally}\ Stage1
\end{array}
$$

$$
\begin{array}{rl}
Stage1 \stackrel{\text{def}}{=} & \textbf{catch}\ diss = 1\ \textbf{in}\ Preparation \\
& \quad \textbf{finally}\ Stage2
\end{array}
$$

$$Stage2 \stackrel{\text{def}}{=} \textbf{catch } res = 1 \textbf{ in } Dissonance$$
$$\textbf{finally } Stage3$$

$$Stage3 \stackrel{\text{def}}{=} Resolution \parallel \textbf{tell}(end = 1)$$

The $Musician$ will start executing the process $Melody$ (supposed to play the main melody of the whole song, may be through `MIDI`) waiting to catch the signal $prep = 0$ during it. When it catches the signal, it interrupts (stops) the $Melody$ and launches the $Stage1$ of the dissonance.

The same philosophy applies to the agents $Stage1$ and $Stage2$ each of them waiting for the signal that tells to carry on the next stage in the dissonance sequence, also assuming that process $Preparation$ plays the preparation and process $Dissonance$ executes the dissonance.

To conclude the sequence, the agent $Stage3$ waits for no signal, instead it launches the process $Resolution$ (also assumed to play a resolution congruent with the dissonance) and post a signal $end = 1$ telling the conductor that the current dissonance is over.

The process $Melody$ is the main harmonic structure the musician has planned for the song and is in charge of *evolving* the melody so to speak. Processes $Preparation$, $Dissonance$ and $Resolution$ will select non-determinalistically a chord to play from a set of chords specifically built to fulfill the process's task. For example, the set of chords from the process $Preparation$ is able to transcend to a dissonance and at this point the process $Dissonance$ will take the lead and the set of chords from where it will choose to play now will be dissonant ones.

These *chords set* may be constructed using a *relative distance* to the current tonality the melody is carrying. Using ranges over these distances a set chords can be discriminated to imply they belong to certain set. The *relative distance* of a certain chord is estimated using its notes' harmonic ratios against the root chord of the tonality and taking the same principle discussed above of deciding the degree of consonance and dissonance. Note that the dissonance concept vary in genre or music so the ranges used to make the chord sets are left for the musician using the model to decide.

5 CONCLUDING REMARKS

In this work, we described the concept of dissonance from various perspectives and also provided its mathematical relation with the consonances, we presented their musical evolution and its main function in the context of composing.

For the appropriate modeling of such problem, it was required that the usage of the dissonances be expressed in a sequential form because music is, as many more view it, a phenomenon occurring over time (melody) and concurrently (instruments or voices). Because of this, we chose the `rtcc` calculus, its concrete and direct way of treating time and how it manages asynchronous behaviour made possible the appropriate modeling of the dissonances as a nondeterministic process over time.

We also proposed a concurrent model that may be expanded or reduced easily to fit the management of the dissonance according to the need of the musician. The reader may see that any of the steps to make the sequence can be easily left out without affecting the integrity of the whole system. For example the musician may avoid the *resolution* step and leave all the dissonances unresolved or postpone it indefinitely using the operator $\star$.

We plan to pursue this work in a more practical direction. We have begun the implementation of an interpreter of `rtcc`. We are convinced that a software helps to better visualize the behaviour of systems, to make possible listening the audio results of the models in real-time, and to prove properties in those models. In the AVISPA research group [2] some interpreters and simulators have been developed for some other extensions of `ccp` such as `ntcc` and `utcc` (see for example [11, 12, 13]). This knowledge has been useful for the development of our interpreter. Initial work on the software has given us encouraging results.

6 REFERENCES

[1] D. Deustch, *The Psychology of Music, Second Edition (Cognition and Perception)*. 2nd Edition, Academic Press, 1998.

[2] R. Jourdain, *Music, The Brain, And Ecstasy: How Music Captures Our Imagination*. Harper Perenial, 1st Edition, 1998.

[3] D.C. Melnick, *Fullness of Dissonance: Modern Fiction and the Aesthetics of Music*. Fairleigh Dickinson Univ. Press, 1994.

[4] T. Harrison, *1910: The Emancipation of Dissonance*. University of California Press, Ilustrated Edition, 1996.

[5] A. Copland, *Cómo escuchar la Música*. Fondo de Cultura Económica, Spanish Edition - Bogotá, 1997.

[2] `http://avispa.puj.edu.co/`

[6] I. Xenakis, *Formalized Music: Thought and Mathematics in Music*. Pendragon Press, Revised Edition, 1992.

[7] A. Schoenberg, *Fundamentals of Musical Composition*. Faber and Faber Limited, 1970.

[8] V.A. Saraswat. *Concurrent Constraint Programming.* ACM Doctoral Dissertation Award. The MIT Press. Cambridge, MA, USA. 1993.

[9] G. Sarria and C. Rueda Real-Time Concurrent Constraint Programming. In *CLEI'2008*, Santa Fe, Argentina, 2008.

[10] G. Sarria. *Formal Models of Timed Musical Processes*. PhD Thesis, Universidad del Valle, Cali-Colombia, 2008.

[11] C. Rueda and G. Assayag and S. Dubnov A Concurrent Constraints Factor Oracle Model for Music Improvisation. In *CLEI'2006*, Santiago de Chile, 2006.

[12] M. Toro-Bermudez, C. Rueda, C. Agon, G. Assayag NTCCRT: A Concurrent Constraint Framework for Real-Time Interaction. In *ICMC'2009*, Montreal, Canada, 2009, To appear.

[13] C. Olarte and C. Rueda. A Declarative Language for Dynamic Multimedia Interaction Systems. *Communications in Computer and Information Science (CCIS)*, vol.38, Springer, 2009. To appear.