The Use of Models in Music Theory Pedagogy for Amateur Musicians

Danielle Watson
University of Toronto
75 Lakeway Drive
Ottawa, ON K1L 5A9
1-613-247-1546
dwatson@friars.providence.edu

ABSTRACT

The study of music theory is an essential part of learning to become a musician. However, the majority of student and amateur musicians struggle with music theory, ignoring it to the possible detriment of their musical performance. To find a way around this, a system must be found that allows music students to learn theory at a faster rate. Models help students to keep in mind the essential facts while performing, rather than having a mental separation between the theoretical aspects and the practical aspects of performance practice.

1. INTRODUCTION

The study of music is composed of many different parts, especially if the student is in any way serious about it. To receive a degree in music, courses in music theory and music history are as required as are lessons in music performance. The study of music theory is generally thought to be the most challenging of the three.

Music theory is the study of the mechanisms used by composers. Over the centuries, it grew into a body of rules that, if followed, helped the composer in achieving the wanted affect upon the audience. The main reason for the study of music theory by musicians is to be able to understand the structure of a piece of music, in order to play it in the way the composer intended.

The notation system used for music is, at its base, relatively simple. It consists of a staff of five lines, with each line and each space between lines representing a note. Notes can also be placed above and below the staff to a certain extent. On the staff at the beginning of the piece are various markings representing the type of rhythm of the piece, specifications of what notes are allowed to be played, and the clef, or where on the staff the alphabetical scale of notes begins. Throughout a piece of music, other markings are used at the composer's discretion to change the speed, the volume, and the style in which the piece is played.

Though all of these things exist to help the performer play the piece how the composer intended, the analysis of a piece of music is equally important. For one, in certain situations, certain styles of performance are expected, and so the composer will not explicitly write what he expects, but rather imply it through various ways. A good example of this is the cadence, which gives a sense of completeness to a section. A performer is expected to emphasize a cadence, though the exact way in which this is done depends on the placement of the cadence and the atmosphere of the piece. In most 19th century music the cadence at the end of a section often involves a deceleration (*ritardando)*, for example.

Unfortunately, due to the difficulty that most have with music theory, it is uncommon for amateur musicians to put this into practice. Those that do tend to do so not because they understand music theory, but because their performance teachers tell them to. The main problem with this is that this does not allow them to apply it in other situations. The *ritardando* mentioned above is generally used in places that are easy to identify, such as the end of a piece, but some cadences are not at all easy to find without a formal analysis, so what the composer wanted goes unheard.

Computers and embedded systems are used increasingly in the field of music performance today, the most common (and recent) example of this being the use of an iPad instead of sheet music. There is a great deal of notation software available, and it is very frequently required of music theory students to use this software for composition assignments instead of handwriting them (mainly for intelligibility purposes). However, there is very little available for the student of music theory on a simple level.

The lack of software for students is especially puzzling because of the relationship of music theory to programming. Both disciplines require a similar approach, to the extent where it is not uncommon for students of both to attempt to write a program to analyze a piece of music. The reason this tends to fail is that a certain level of abstraction is required, which most students are incapable of achieving at the point in time at which they are studying both disciplines, leading to a very time-consuming exercise.

It is for this reason that the use of models would be very useful for students and amateur musicians in order for them to better understand the music they play.

2. MUSIC THEORY

2.1 Chords

The simplest version of a chord in four-part harmony is a set of notes played simultaneously. Each note is required to have a certain relationship with each other note. The order in which they are placed upon the staff has relevance only in terms of correct voice leading: each note represents a voice, either soprano, alto, tenor or bass, and the changes between one chord and the next are under a strict set of rules. For example, the tenor and alto voices cannot be more than an octave (8 notes) apart.

One of the most important rules, and perhaps the one most often broken by amateurs, is the rule of parallel fifths. If there are five notes in between two voices in any octave (i.e. the distance between C and G) in one chord, then the next chord cannot have the same distance between those two voices. The reason for the problem with this rule is that the change from one chord to the next is also heavily regulated. There is a set order of chords allowed in order to form a phrase of music, and unfortunately for the beginning theory student, it is impossible to form a proper chord progression without weakening the strength of a chord.

The weakness or strength of a chord depends on which note of the chord is on the bottom, and which note is doubled in the case of a triad. The strongest combination is for the main note of the chord to be both repeated and on the bottom. This is known as root position. The most common chords are known as triads, which consist of a stack of three notes, the root, the third, and the fifth. The next most common are seventh chords, which add a seventh on top of the fifths. There are other chords, but they are generally restricted to jazz music, and only studied by advanced music theory students.

The order in which chords must appear in a progression is determined by the key of the piece. There are 13 notes available in Western music, the letter names A-G and 'accidentals' in between letters. In general, the distance between two letters is known as a tone, and between a letter and an accidental is known as a semitone, though there are two exceptions: between B and C, and between E and F are semitones. A scale is a set pattern of tones and semitones which starts on any given note, in the pattern T-T-S-T-T-T-S (major) or T-S-T-T-S-T-T. The C-major scale, on which all Western tonal music theory is based, is C-D-E-F-G-A-B-C, and the C-minor scale is C-D-Eb-F-G-Ab-Bb-C (the bs represent lowering the note by a semitone). Chords in a piece of music must contain notes appearing in that scale, with some exceptions.

An example of a triad is CEG. This is the C major triad (triads other than major alter the third and fifth). In the key of C major it is known as the I chord, which is the strongest chord in a scale, and can be used at most points in a chord progression, though properly the best points are the first chord and the last chord. However, in the key of G major it is the IV chord, which is much weaker, though still relatively strong, and can only be used in certain types of cadences, or as preparation for a cadence. In the key of D major, it is an illegal chord (except in special circumstances), since D major requires the note C# (one semitone higher than C).

2.2 Types of Chords and Intervals

There are many types of musical intervals, and learning them is another area of early music theory in which a significant amount of time is spent. The two areas intervals can be grouped in are:

Perfect/Augmented/Diminished: perfect intervals are often considered the purest of intervals, and the goal of a chord is to ultimately resolve to a perfect interval. Perfect intervals are also the most dangerous when it comes to voice leading. There are only 3 perfect intervals: 4th, 5th and 8va. Augmented intervals are caused by raising the top note by a semitone, and diminished by lowering it. These two are considered the most dissonant of all intervals, and should be resolved as soon as possible

Major/Minor: major intervals are the most common intervals, and they are generally considered to be consonant and 'good' sounding. It is acceptable to move to a major interval. A minor interval has the top note of a major interval lowered by a semitone, and is treated similarly to major intervals. Major and minor intervals can also become diminished or augmented.

Triads and seventh chords are the building blocks of harmony, and, since they are composed of various intervals, are generally described in terms of those intervals:

Major: a major triad consists of a root, a major third and a minor third on top of it. C-E is major, but E-G is minor, since in the key of E major, G# is used. This is the most stable type of triad.

Minor: a minor triad is an inversion of a major triad: root, minor third, major third.

Dominant Seventh: the dominant seventh is one of the most important chords. It is often used as a substitute for a dominant (V) chord, hence the name. A dominant seventh is technically only a dominant seventh when it is used as a dominant, as otherwise it is known as a major-minor seventh (this is extremely uncommon). It consists of a major triad with a minor seventh.

Many other types of triads and sevenths exist, but these are the most commonly used. There are many methods used to describe chords, including guitar tablature, jazz, figured bass, etc. The most common method for analysis is Roman numerals. The typical major scales involves the chords I-ii-iii-IV-V-vi-vii^{o} (the last being a diminished chord). A minor scale involves the chords i-ii^{o}-III-iv-(V)-VI-VIII(never used). It is generally considered to be against the rules to use a chord from the minor scale in a piece written with the major scale, which is something that a model checker should prevent.

3. EXISTING MODEL METHODS

Keeping all of these rules straight can often lead to a musician becoming so confused that they cannot apply the uses of these rules to the piece of music they are performing. It is for this reason that the use of models and model checking in learning music theory could be quite useful.

3.1 MusicXML

There already exist some methods for doing this. MusicXML is a variation on XML specifically for music notation. Most music notation software is able to convert MusicXML files to sheet music, and vice versa. However, while it is very useful, MusicXML does not possess the level of abstraction required for a student to use. It is faster to write out music by hand than use XML, especially for someone who has little knowledge of programming (see figure 1 for an example of the complexity). MusicXML is ill-adapted for direct usage when it comes to analysis.

Figure 1.1: A section of music with one note (C) written on a clef

```
<measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
```

Figure 1.2: The same section in MusicXML

3.2 Petri Nets

Another method was created by LIM, the Music Informatics Library, using Petri Nets. This method is quite useful in terms of composition and advanced analysis, such as the overarching structure of a piece. Petri Nets allow the user to both look at the piece over time and in terms of many events happening consecutively –in terms of music notation, both horizontally and vertically.

ScoreSynth is an implementation of Music Petri Nets, currently available for download from the Italian LIM website (http://www.lim.dico.unimi.it/loadpage.php?page=download&language=ita&dim=1), though, oddly, not from the English version of the website. It supposedly allows the user to model a musical score using Petri Nets for further analysis. Unfortunately, there is very little documentation available, and what exists is currently in Italian. Though comprehension of either French, Italian or German is generally required for a MMus, the user interface of ScoreSynth can be confusing, and requires an ability to work with Petri Nets.

Most of the work done on music theory software in general tend to either be heavily skewed towards the computer science side, which most musicians are not that knowledgeable about, or tend to be rather basic: quizzes, exercises, etc., and above the price range of the average student. Though there is a certain overlap between musicians and computer scientist in post-secondary institutions, most of those who fall into this overlap are either computer scientists who are passable musicians, composers and people interested in compositional tools, or people interested more in the actual sound rather than the theory behind it. Thus the number of people who are both interested in and can use ScoreSynth is rather small. The same holds true for other available programs.

3.2.1 The Problem with Petri Nets

The Petri Net solution, while quite useful, has a number of problems when applied to music theory pedagogy, especially the problem at hand. The main problem is that it is too good at what it does on several levels. It looks at the structure of the music on a higher level of abstraction than needed, which leads to the assumption that the lower level of analysis was done correctly. This is something even more useful when it comes to music performance, but it is generally taught in 2nd and 3rd year university courses. Another problem is that learning the structure of Petri Nets is akin to learning another form of notation, which would complicate the learning process instead of helping it.

As mentioned above, the major problem with creating simple software for music theory is the level of abstraction required. An experienced theoretician, when looking at a piece of music, is able to automatically see patterns of chords, take into account the key of the piece, and analyze an ambiguous chord using context. A program can accomplish most of these tasks, but most of the music analysis tools available analyze sound files directly, or do not process the information the same way the human mind would; the optimal algorithm for a computer is not necessarily optimal for a human. The available tools, therefore, while useful, are not capable of helping a student to learn the proper techniques; this is not their purpose.

3.3 Why Models Should Be Used

The ultimate problem of music theory for its students is how to abstract relevant information from a large number of data points. Once learned, this process is generally efficient. However, it often takes years to learn how to do this, thus lessening the applicability of music theory to the point where most musicians stop their studies of music theory as soon as possible, or, if possible, do not study music theory at all, preferring to focus on practical performance techniques.

The hastening of this process is therefore to be desired, and the use of models, specifically the FTS, is a useful tool in this. From the modeling perspective, the use of modeling techniques in music theory analysis is an attempt to make the use of models more intuitive for a user who has no programming experience.

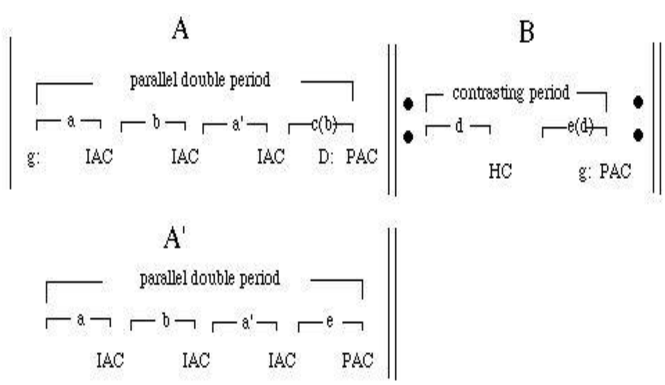


Figure 2: Example of a higher-level form model

4. MODELS FOR MUSIC THEORY

It is for this reason that models are ideal for music theory pedagogy. Musical notation can be said to be a type of model already, albeit one that allows a large amount of information to be placed into a small space, which can be quite confusing. The main difference between a typical model used in software engineering and musical notation is the purpose: notation describes the movement of things through time. Although there are types of models which allow this type of motion, to use the same level of detail there as in notation would be confusing and unwise.

Some models are already in use when teaching theory at a higher level of abstraction, notably of an entire piece. A modeling system that is the most similar to what is already in place makes the most sense, and one that is currently available is FTS, or the featured transition system.

4.1 FTS for Music Theory

A modification of this system is useful for music theory students for several reasons: it slightly resembles the models they will use later if they continue their studies (see fig. 2), it is a relatively intuitive system, and it can be used in both a computer and a pencil-and-paper environment. The structure of FTS also gives the student greater insight into the inner workings of four-part harmony, which is especially useful for chorists. As the vast majority of music theory work is currently done on paper, it shall be assumed that the hypothetical student is working on paper.

4.1.1 Chord Progressions

There are several ways in which FTS can be used for music theory. The simplest of these is in the analysis of a musical phrase. The most common musical phrase is I-IV-V-I, with each roman numeral designating the scale degree the chord in question starts on. There are several variations; V can be replaced with V^7(V with added seventh), or vii°. IV, being the weakest chord, has a number of possible replacements. Even the I chord can be replaced at the beginning, though this is strongly discouraged.

Because of this, it is relatively simple to draw a state chart for a chord progression (see figure 3). The student would then be able to create a model of the piece in question using FTS, and compare it to the base model to discover the phrasing of the piece. Being able to play a phrase properly is an important part of performance,

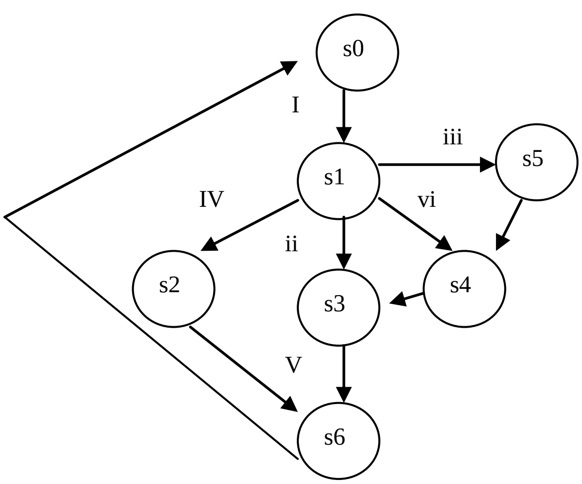


Figure 3: Example of a simple chord progression chart

and one of the problems that younger amateurs have with performing is being unable to find the phrases in a piece of music. Most musicians, after performing a significant amount of music, find themselves able to pick up on phrasing unconsciously, but even an intuitive sense of the structure of the music can often be wrong when compared to a straight analysis. The majority of practice time is spent on technique and musicality, however, the time commitment for analysis can be greater than the perceived benefits.

The one major problem with this is that it assumes that the lower-level analysis is correct –i.e. the chords have been identified correctly. There can be a great deal of ambiguity regarding chordal analysis, which further obscures phrasal analysis. However, the problems with chordal analysis are best left for another discussion.

4.1.2 Voice Leading

Voice leading in four-part harmony is an exercise which all theory students must go through. Though it is not required for performance per se, it is still an essential part of music theory, and helps students to learn analysis tools. The only use in performance is for the piano or choir: the pianist needs to know how the various lines he plays at the same time interact, and the chorist must be able to interact with the only lines being sung. Voice leading is theoretically applicable for string ensembles and other instrumental groups, but in practice, since each player is only furnished with the part for one instrument (as opposed to vocalists, who are given the full score), it is very rarely used. The parts used in instrumental ensembles also tend to have more complexity per part, so it is significantly harder to analyze a single part from an orchestra than it would be for a chorist.

A typical voice leading exercise is for the chords of a phrase to be given, with the student expected to fill them in. On occasion, a single voice is given, sometimes with no chords underneath, and the student must work around the restrictions of that voice.

The FTS model for this situation would bear some similarity to the Music Petri Nets from [1]. Each voice would be represented by a different line, and the distance each moved would be marked in, as well as the various vertical distances (to stop the voices from moving out of range). Though the addition of this step would initially add time to the process, overall it would save editing time.

4.1.3 Common Problems

This covers some of the most common exercises given in music theory classes, but it fails to take into account one important fact: composers often ignore the rules of music theory when it suits them, or, failing that, spend a significant amount of time bending the rules. From a musical perspective, this makes sense: it is extremely difficult to produce appealing or interesting music which follows the rules exactly. It is this fact which so frustrates many students of music theory.

A chord in a piece of music is not necessarily placed vertically. Nor do all voices always begin and end at the same time. In addition, there is the common use of non-chordal tones (NCT), notes which are not within the chord, but are used for artistic effect. The use of NCT is a common addition to voice leading exercises.

4.1.4 Non-Chordal Tones and FTS

The addition of NCT to a 'blank' voice leading exercise (i.e. one without a given voice) is up to the student's discretion, and generally simple. It is when a voice is given that problems arise. The choice between using a given note as a NCT or as part of a chord can cause voice leading problems later in the exercise. A simple solution would seem to be to treat all notes as chord tones, but this is both contrary to the point of such exercises as well as possibly breaking the laws of chord progressions.

The method to use in such a scenario would be to explain the process in a series of steps. The method generally used to do this exercise is to guess at which notes are chords, and then to edit later if the guesses turn out to be wrong. The problem with this method is that it can lead to the entire exercise having to be rewritten. An experienced student can often intuit the right choices, but the goal of using models to solve music theory problems is for the student to be able to fully understand the process, rather than using intuition, which, as mentioned above, can often backfire.

For optimal student understanding, the best process would be to use two models. The first represents the process, and the second, the notes themselves.

4.1.5 Second-Order Harmonic Analysis

The next level of analysis up is the appropriately named second-order harmonic analysis. Most of the techniques mentioned earlier are useful in this analysis. The main difference between second-order analysis and first-order analysis is that first-order analyzes every chord. Second-order analysis, on the other hand, only deals with the important chords, i.e. those that fit into the model of possible chord progression. Any number of chords can be added in between important chords. For example, though the typical four-bar harmony is I-IV-V-I, the reality might look closer to I-V-I-vi-IV-iii-I-V-I.

Since this is a direct analysis technique, rather than application, the process is much simpler, assuming the chords are correct. The first-order analysis is compared to the model of possible chord progression, and those chords which do not fit into the model are discarded, allowing the greater structure of the phrase to be visible.

4.1.6 Cadences

The most important of these to the student musician, however, is probably the identification of the cadence. Using second-order harmonic analysis, a cadence can be identified as the last two chords in a phrase. A cadence is the point at which the music 'rests'; it serves to mark the division of a section, to resolve dissonances created as the phrase moves through time. Though the most common and strongest cadence is the perfect cadence (V-I), other cadences exist which help shape the phrasing of the music, and thus should be noticed.

Other types of cadences, however, are often tricky to identify. What one may mistake as a cadence may in fact be the continuation of a longer-than-average phrase, or indeed any part of a phrase that is not the cadence. The most common example of this occurring is with the plagal cadence (IV-I), the usage of which is comparable to the use of the semicolon. As can be seen in the above example of a first-order analysis, the first section I-V-I-vi-IV could be interpreted as a plagal cadence. Cadences therefore cannot always rely on second-order analysis.

Cadences can generally be identified in that they are given more importance by the score; they may be marked as louder or softer than the surrounding music, they may have less NCT, the relative tempo at which they are played may be different, etc. The way to discover a non-perfect cadence (which includes imperfect, deceptive and plagal) is to use the type of models from voice leading exercises. Though this is the analysis of a phrase rather than the creation, it is actually simpler to use a model.

5. PROBLEMS

5.1 Level of Difficulty

Further levels of abstraction, while useful to the performer in general, are often beyond the ability of a student of this level to keep in mind. When performing, a student musician is often more concerned with practical matters such as intonation and tempo, and so has a limited amount of concentration to spare to keep in mind the greater structure of a piece, and how that informs the musicality of the performance. A professional musician should be expected to keep this in mind, but a student should not be required to do this, especially when performing in an ensemble, where everyone has to pay attention to the music, the conductor, and their section-mates, as well as the usual difficulties in performance.

It is irrational to expect a student to keep in mind a model of chord progressions while performing. However, by doing the in-depth analysis that the creation of the model requires, and realizing the implications of each chord change, the student will eventually know this automatically, especially if it is kept in mind while practicing. The goal of practicing for a musician is to get to know the music, similar to how one would practice a speech. When practicing music, however, the focus is on the notes themselves, so that when it comes time to perform the piece, the focus of the musician is not on the notes, but on how to perform it. Eventually, this becomes a focus on musicality for the more advanced musician.

The human mind enjoys order, and for this reason we have devised shortcuts such as writing and other abstractions, so that we can order our thoughts. The process of learning music theory can to the student seem disordered, with rote memorization often the only way to learn. Rote memorization is not friendly to a complete comprehension of the topic at hand. Allowing the mind to put it into an order by using models means faster learning as well as complete comprehension.

5.2 Problems with the Music Community

There is an odd disconnect between the use of software tools for music inside and outside the pedagogical environment. Though laboratories exist inside university music departments, full of MIDI keyboards and computers with notation software, there is very little attention paid to the use of software in terms of music theory. There seems to be almost a slight hostility towards it, in fact. There do exist some programs which are used by theory teachers, but most of them either for ear training, or meant for composition purposes. An exception lies in the field of post-tonal music theory, which requires a lower level of abstraction, and is more heavily based on mathematics. The ever-growing range of available mobile applications may be putting an end to this hostility on the part of students, but as yet there exists no theory tool which is used to the same extent as notation software such as Finale or Sibelius. Some open-source notation software, such as MuseScore, possess plugins that check for voice leading and similar problems, but these are not common, and often require knowledge of programming that the average musician would not have.

The successful use of models in a pencil-and-paper environment could lead to extensions in software and embedded systems. Requiring a student of music in 3rd or 4th year university course to analyze each chord of a piece separately in order to analyze the entire piece is comparable to asking the student in a calculus course to not use a calculator.

5.3 Problems with the Intellectual Laziness

One unfortunate aspect of using model checking programs rather than by hand is the propensity of some students to take shortcuts. Though it would be useful for students at higher levels of music theory, the temptation to cheat is sadly present when it comes to both the type of exercises and the type of program. In fact, the most common reason university students knowledgeable in both disciplines attempted to write analysis programs was to help with music theory homework (though the attempt consumed much more time than it took to actually do the homework –a matter of weeks rather than hours).

It is for this reason that it is best to start with pencil-and-paper models rather than software. While it does cut into the capabilities of the format, and involve much more time, it is the best choice when it comes to learning the material. In addition, there is the work done on Petri Nets by LIM, for those interested.

6. FURTHER WORK

The user-friendliness of an implementation is an essential part when it comes to pedagogical tools; if learning how to use the tool takes longer than learning the object of the tool, then the pedagogical tool has failed. The use of models for basic music theory is generally pointless, from the perspective of a developer: the use of models with higher levels of music theory is already present. The vast majority of research in the crossover between computer science and music theory is of a high level: common problems being the analysis of a piece of music to find out the composer or era and the generation of computer music in specific styles. The ability to achieve the analysis mentioned in this paper is required for most of these, but it is not looked at since the point of solutions to these problems is to generate a radically different result. While both of these areas remain fascinating, they do not provide much assistance to those who do not have a degree in music theory, but study it either due to it being a required course, or for enjoyment, though the latter is extremely uncommon.

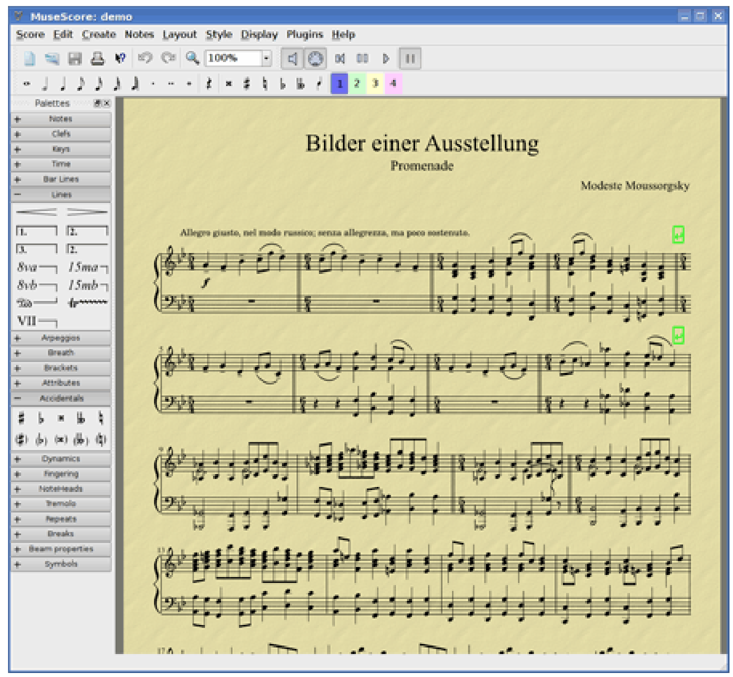
There are two main methods which would be ideal for the further application of models to music theory pedagogy: mobile applications and existing notation software.

6.1 Mobile Applications

The benefit of using a mobile application is that it is easy to disseminate the information, due to the present popularity of devices using mobile applications. There are three possible approaches of varying complexity:

Explicit Models: Due to the versatility of the FTS, a complete mobile application would have several templates to check chords against, as well as the ability to draw models. The main drawback of this format is that it can be awkward to draw on a device which does not have a stylus, once again cutting into the user-friendliness aspect.

Figure 4: MuseScore



1. **Implicit Models with Notation:** Another option would be for a staff to be set, which the app would then analyze without displaying the results. The problems with this approach are the issue mentioned above regarding academic laziness and the problem with notation software in general, which will be mentioned below.

2. **Implicit Models with Photographs:** The most efficient way to accomplish the task at hand would be to physically write out the chord progression, or take printed sheet music, photograph it, and then use the same process as the second option. The problem with this approach is the complexity: a piece of music may contain more information than required to solve the problem, which may lead to inaccurate results

6.2 Notation Software

The other option is a plugin for notation software. This would work similarly to #2 above, except on a computer rather than a mobile device. The benefit of this approach is that the user is not restrained by the confines of a mobile device. By adding it to existing software, it allows for the intelligibility of the result as well as relatively simple access. Notation software, however, has a number of flaws to the point where the benefits from using it do not necessarily outweigh the benefits of writing out the exact same thing by hand. The two things notation software does better than manually writing are intelligibility and playback. Writing music by hand is noticeably faster. Even the use of a MIDI keyboard only helps to a small extent.

6.2.1 The Problem with Notation

The problem with music notation is it requires the writer to work on several levels at the same time: the rhythm of a piece is equally as important as the values of the notes, in both practice and analysis. When it comes to applying this, however, notation software is unable to work on both levels at once. This is one of the main reasons writing out FTS models for music would be more useful for students.

There does exist a plugin for MuseScore which searches for parallel fifths and other voice leading problems in four-part harmony, though it makes the assumption that no NCTs are used, and that the chords change in an orderly rhythm (i.e. each voice in each chord begins and ends at the same time as every other voice in the chord, and this time interval covered is equivalent for every chord).

7. CONCLUSIONS

Due to the generally advanced nature of most of the work done both in the past and at the present time in the field of computer science applications to music theory, there does not exist a tool for student musicians to use in order to increase their learning. The production of such a tool would help save the time spent on small details, so students could instead focus on the aspects of music theory which they are currently learning, instead of being tied down by minutiae.

Unfortunately, the creation of such a tool at the current time is not possible, due to the limitations of those who both have sufficient knowledge in both disciplines and the limitations of the implementations currently available. An automated pedagogical tool must have a similar process to a human mind, especially in a subject which currently relies heavily on intuition.

However, the beginning step in building this is to remove some of the intuitive and automatic aspect of music theory in the students` mind by giving them a more complete comprehension of the problems and solutions. The use of modeling systems such as FTS is something that will allow this. In addition, such models may be used later in the actual software tools, thus acting as a natural continuation for those involved in further theoretical studies.

8. ACKNOWLEDGMENTS

I would like to thank the students and teachers in the Music Programme at Canterbury High School and those majoring in Music and Music Education at Providence for sharing their experiences with me, especially John Pohran, Allen Ridgway, and David Riley.

9. REFERENCES

[1] Baraté, A., Haus, G., Ludovico, L.A. 2005. Music analysis and modeling through petri nets. In *Proceedings of the Third international conference on Computer Music Modeling and Retrieval* 2005. R. Kronland-Martinet, T. Voinier, and S. Ystad, eds. 201-218.

[2] Classen, A. 2010. *Modelling with FTS: a Collection of Illustrated Examples*. Technical Report. University of Namur.

[3] Classen, A., Heymans, P., Legay, A., Schobbens, P. 2011. *Symbolic Model Checking of Software Product Lines.* Technical Report. University of Namur.

[4] Classen, A., Heymans. P., Legay, A., Schobbens, P., Raskin, J. 2010. Model checking lots of systems: efficient verification of temporal properties in software product lines. In *Proceedings* of ICSE 2010: 335-344.

[5] Haus, G., Rodriguez, A.: Formal Music Representation; a Case Study: the Model of Ravel's Bolero by Petri Nets. In: Haus, G. (ed.): Music Processing. Computer Music and Digital Audio Series. A-R Editions, Madison (1993) 165–232

[6] Ludovico, L.A.. 2006. An XML Multi-layer Framework For Music Information Description. Ph.D. diss., Università degli Studi di Milano, Milan.

[7] MusicXML for Exchanging Digital Music. http://www.makemusic.com/musicxml. Updated 2012. Accessed November 30, 2012.

[8] Riley, D.. Class Lectures. Theory III. Providence College, Providence, RI, USA. 2009.