Composing by Laypeople: A Broader Perspective Provided by Arithmetic Operation Grammar

Guido Kramann

Fachbereich Technik

Technische Hochschule Brandenburg

Magdeburger Strasse 50

14770 Brandenburg an der Havel, GERMANY

kramann@th-brandenburg.de

Abstract

Many existing approaches to teaching laypersons to compose are based to a certain extent on simply hiding the theoretical background. This is done there, for example, by offering ready-made musical events that can be combined in any way to organize them in time. Here, a different approach is taken: As an alternative for the elaborate classical music theory, the generative composition method "Arithmetic Operation Grammar" (**AOG**), which is much easier to learn than the former, is used. This is done in the conviction that the leaner theory on which it is based, in combination with the compact symbolic representation of entire compositions, can make a significant contribution to bringing forward the "every day creativity" (Keller et al. 2014, pp.5-7 and 29-30) in the field of ubiquitous music. Furthermore, in the field of sonification, **AOG** offers in particular the possibility of sonifying data that do not contain time as an ordering parameter. In order to prove the practical use, **AOG** is combined with a user interface that is more suitable for adults as a target group and another one that is more appropriate for primary school children.

Introduction

> "There will always be great masterworks and great performances by individuals of exceptional knowledge and skill ... But anyone who thinks that such works should or will indefinitely maintain an exclusive dominance over all other musical forms and processes has not been paying much attention to musical history or ethnography, to what is going on musically in this world right now, or to what else-besides masterworks and virtuosi-people really do love about music." (Laurie Spiegel, in Spiegel 1998)

The magic of performing a classical concert lies not least in the illusion of lightness and spontaneity with which a virtuoso on stage interacts with the orchestra. However, all this is painstakingly worked out in years of musical study and even longer intensive practice of the musical instrument. Nowadays, impressive successes have been achieved in giving amateurs and also children the opportunity to create real-time compositions themselves, and thus to have so to speak the exhilarating experience of becoming the creator of a perfect musical moment. Current approaches typically focus on providing special user interfaces that allow to organize pre-produced and pre-selected sound elements in time (Jakobsen et al. 2016, Souza Stolfi et al. 2018, Figueiró et al. 2019). This pre-selection ensures that the provided sound events can be combined arbitrarily. Or they work with post-processing of the user interaction which nowadays happens with the use of artificial intelligence more and more often (Garcia-Valdez et al. 2013, Biles 2007). Thereby, the necessity to provide a theoretical background is avoided. But the disadvantage of this procedure is also obvious: once the given tonal material has been used extensively, it does not open up any further perspectives. Besides a first experience of what composing is, one remains, so to speak, caught in a hermetic world of sound with very limited possibilities. In order to get on further from this point, the only option would be to deal with the traditional theoretical materials as supported for example in (Almeida et al. 2019).

In the approach described here, a different path is taken: An approach to composing will be provided, which also makes it possible to create even complex compositions that satisfy laws similar to those of the classical theory of harmony and counterpoint, but without having to refer directly to them, and even without having to deal with musical theories at all. The method presented here, namely "Arithmetic Operation Grammar" (**AOG**), thus opens up an approach to composing for laypersons that does not require years of study, but does not have to hide anything in terms of the complexity of the mechanisms behind it, since these mechanisms are much simpler than those of classical music theory. However, the price you have to pay is that it's not easy to copy any existing musical style using **AOG**. Rather, the results achieved with AOG have certain qualities that one has to accept.

While **AOG** was presented for the first time at the CMMR 2019 (Kramann 2019), the focus will now be on the extent to which the use of **AOG** in combination with suitable user interfaces can simplify access to composing for laypeople. But what is meant by the statement that with **AOG** it will be easier for amateurs to compose than on the basis of classical music theory? In slightly simplified terms, one can say that the classical theory consists of a collection of prohibitions and commandments. According to Chomsky, **AOG** represents a generative grammar of type 3 (Chomsky 1956, Chomsky 1959). This means that the results are well-formed, which in this case in turn means that, apart from the generative rules of how music is generated from a symbolic representation, no further rules are needed to analyse and correct the result in retrospect. Thus, **AOG** also takes a special position in generative grammars in relation to music: Typical representatives such as Lindenmayer systems or cellular automata provide patterns from a symbolic representation, for which it has yet to be decided what to do with them musically. This means that these are extra-musical procedures (Supper 2001), whereas **AOG** directly provides a musical structure which, apart from a musical interpretation that is still necessary (instrumentation, playing techniques, dynamics, ...) does not need to be corrected further. Thus, **AOG** goes beyond these categories, so to speak.

To categorize **AOG** somehow, it would of course also be conceivable to compare this technique with other approaches that use mathematical methods in some way to perform musical composition. These other approaches usually originate from music theory studies in which mathematical models are created to put any aspect of composing into a more general context. One such an aspect are chord progressions. In corresponding works typically topologies of all chord progressions are created, which comply with a certain musical style. Choosing and traversing paths through these topologies can be seen as a rudimentary form of composing that involves this kind of modeling (see e.g. Hu and Gerhard 2019). **AOG** differs from this type of approach mainly in that the mathematics in **AOG** does not represent a model of an existing type of music, but special properties of a given mathematical object, namely the inner divider structure of the natural numbers are exploited to create music. From the perspective of **AOG** the sequence of natural numbers contains an infinite number of intertwined melodies that are already in rhythmic and harmonic relationships with each other. These melodies are extracted by mathematical operations and made audible by a so called selective division (see detailed explanations below). By doing so, not only a partial aspect of composing is implemented, but polyphonic compositions are generated in which melody, rhythm and harmony are inherent.

If one would like to relate the compositional results achieved with **AOG** to any compositional styles, one is most likely to find correspondences in such compositional directions of the 21st century, which still essentially organize sounds of specific pitches in time, but which allow a broader spectrum of possibilities than is provided for in classical theory, such as the "tintinnabuli" harmony of Arvo Pärt, or in certain forms of free jazz.

Rhythmic correspondences can be found above all in the repetitive elements of minimal music. But to get a first impression of the compositions that can be generated on the basis of **AOG**, refer to the first entry (#1) of a thematically arranged selection of examples created for this article (Kramann 2020b), which includes videos, scores, as well as audio files and which can be accessed under the following link: URL AT MIT PRESS TO COME, CURRENTLY AT http://www.kramann.info/98_AOG.

Arithmetic Operation Grammar

As the basic element mentioned above, in **AOG** the natural numbers are used as time series. More precisely this basic element consists of the non-negative integers including zero – in literature represented symbolically with $\mathbb{N}$ or also with $\mathbb{N}_0$ – or any gapless finite section of it. In order to show something like an inner musical organization of this sequence of numbers, a method will be used in the following to quantify the degree of dissonance of two integers and which goes back to Leonhard Euler, who called it *gradus suivitatis* ($\boldsymbol{g}$) (Busch 1970). To a certain degree, what is determined by this measure also corresponds to our auditory perception when the respective numbers are taken as oscillation periods or frequencies. The limitations of this method in terms of its transferability to human acoustic perception are deliberately accepted in this work in favour of the simplicity of the method and the fact that it works quite well over considerable ranges. It should be mentioned, however, that the two prime numbers 1999 and 2999, for example, provide a quite high $\boldsymbol{g}$ value, but we would hear them as a fifth (2 to 3). In addition, $\boldsymbol{g}$ delivers a much higher value for a minor third than for a major one. For further information see for example (Schneider and Frieler 2008, Kramann 2015). In the following I will use the terms *gradus suivitatis* and degree of dissonance, understood in the musical sense, synonymously, aware that I am not taking a very differentiated approach from a musical point of view, but also not doing anything really out of the ordinary.

If now the *gradus suivitatis* is to be determined for two integers a and b, one first factors out the greatest common divisor of a and b, then for the remaining prime factors p_i, one considers their powers k_i (the number of times the prime number is multiplied in the remaining factorization). According to Euler, the *gradus suivitatis* is then: $\boldsymbol{g} = 1 + \sum_{i=1}^{n} k_i \cdot (p_i - 1)$. For $a = 16 = 4 \cdot 2^2$ and $b = 20 = 4 \cdot 5^1$, the result is $\boldsymbol{g}(16, 20) = 1 + 2 \cdot (2 - 1) + 1 \cdot (5 - 1) = 7$. For $a = 36 = 6 \cdot 2^1 \cdot 3^1$ and $b = 30 = 6 \cdot 5^1$, for example, the result is $\boldsymbol{g}(36, 30) = 1 + 1 \cdot (2 - 1) + 1 \cdot (3 - 1) + 1 \cdot (5 - 1) = 8$. If one now considers the *gradus suivitatis* between a selected natural number and its closest neighbors, it is clear that as the selected number increases, larger and larger values of $\boldsymbol{g}$ will also tend to occur, since the larger the selected number and its neighbors are, the larger (on average) are their prime factors.

If, however, a modified *gradus* $\tilde{\boldsymbol{g}}$ is introduced instead, in which the number of prime factors that are taken into account is limited to the first four $(2, 3, 5, 7)$, for example, and if, in addition, the considered potency of these prime factors is limited, a completely different picture emerges: Along the natural numbers, the resulting mean value between a number and its surrounding neighbors alternates. The larger such a modified averaged $\tilde{\boldsymbol{g}}_{\boldsymbol{mean}}$ value is, the less often it occurs for any number, and in addition, typically large $\tilde{\boldsymbol{g}}_{\boldsymbol{mean}}$ values are surrounded by small ones. They tend to appear as single peaks and not cumulatively.

$\tilde{\boldsymbol{g}}$ can be defined for two natural numbers a and b as follows: If u is the greatest common divisor of both numbers, then follows: $a = u \cdot x$ and $b = u \cdot y$. The prime factors which are not common to both numbers are then $c = x \cdot y = 2^p \cdot 3^q \cdot 5^r \cdot 7^s \cdot REST$, whereas REST refers to the product of all the prime factors greater than 7. In order to calculate $\tilde{\boldsymbol{g}}$, upper limits are defined for p, q, r and s. These parameters, cut off at the top, are marked

$\bar{p}$, $\bar{q}$, $\bar{r}$ and $\bar{s}$. As upper limits for the powers $\bar{p}_{max} = 3$, $\bar{q}_{max} = 2$, $\bar{r}_{max} = 1$ and $\bar{s}_{max} = 1$ were used. For example, if the value of q is 4, then only 2 may be transferred to $\bar{q}$. For the modified *gradus*, we obtain $\tilde{\boldsymbol{g}}(a, b) = 1 + \bar{p} \cdot (2 - 1) + \bar{q} \cdot (3 - 1) + \bar{r} \cdot (5 - 1) + \bar{s} \cdot (7 - 1)$. See Figure 1, in which over the range from 100 to 299 for each number these limited *gradus suivitatis* are calculated with the five left and five right neighbours, and the mean value of all ten values is shown as a bar.

As preliminary motivation one may take for the step to develop a limited *gradus suivitatis*, that for music in so far as it is an organization of sound events of defined pitch in time, rather small prime factors play a formative role, concerning both the frequency ratios of musical intervals and the rhythm. But if one adopts this, with the help of the limited *gradus suivitatis*, humanized perspective on the natural numbers one could say, something noticeable appears: The way in which numbers that are more dissonant with respect to the numbers surrounding them tend to occur as individual peaks and not cumulatively, corresponds quite well with the common musical rules of traditional western composition. For example, in classical choral composition, roughly speaking, any number of consonances may be strung together, but dissonances should always alternate with consonances. This means, they should occur less frequently and well distributed throughout the movement.

To make this tendency even more visible, we will use Figure 2. This time, the points entered in this semi-logarithmic diagram represent all natural numbers in the range between 100 and 5100. The ordinate in this diagram corresponds to the mean value for $\tilde{\boldsymbol{g}}$ between the number in question and the 90 direct left and 90 direct right adjacent numbers. The abscissa is a logarithmic representation of how far away from the number in question the next number happens to be that has at least the same value of $\tilde{\boldsymbol{g}}_{\boldsymbol{mean}}$, or an even greater value. This selection was made in an exemplary manner. It has been shown, however, that the basic structure visible here remains intact, even if changes are made to

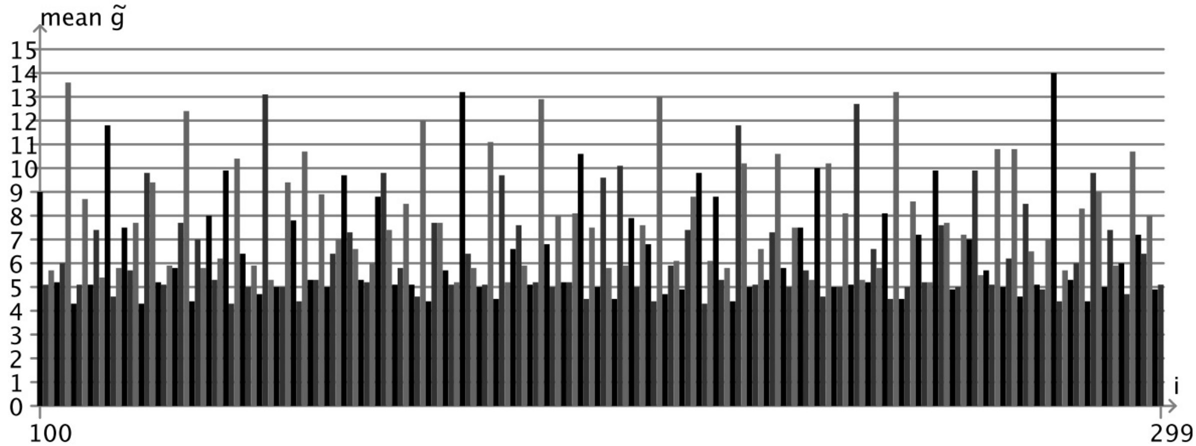

Figure 1. Mean value for $\tilde{g}$ between a number and its 5 nearest smaller and its 5 nearest larger neighbors in the range of natural numbers i from 100 to 299. Larger values tend to occur as individual peaks and not cumulatively.

the range of numbers shown, or the number of neighbours considered. In the diagram, you can now see a band of points lying diagonally with a positive gradient. This can be interpreted to mean that as the value of $\tilde{g}_{mean}$ increases, the distance to a next value with at least the same value of $\tilde{g}_{mean}$ actually increases exponentially. This corresponds to the previous statement that the greater the corresponding $\tilde{g}_{mean}$ is, the more isolated the values are.

Or, seeing $\mathbb{N}$ as a time series and interpreted musically: Tones that lie in a dissonant relationship to other tones nearby in time occur sporadically and increasingly rarely the greater this dissonance value is on average. It does not matter that with AOG the formation of the pitch is won by a division at the end, since the degree of dissonance of two values is preserved if both values are taken reciprocally, as can easily be seen.

Here, then, in the form of this inner structure of natural numbers from the perspective of $\tilde{g}$, we have the basic element mentioned above, which already satisfies musical laws. Remarkably, this is an element that did not have to be invented, but only found, and

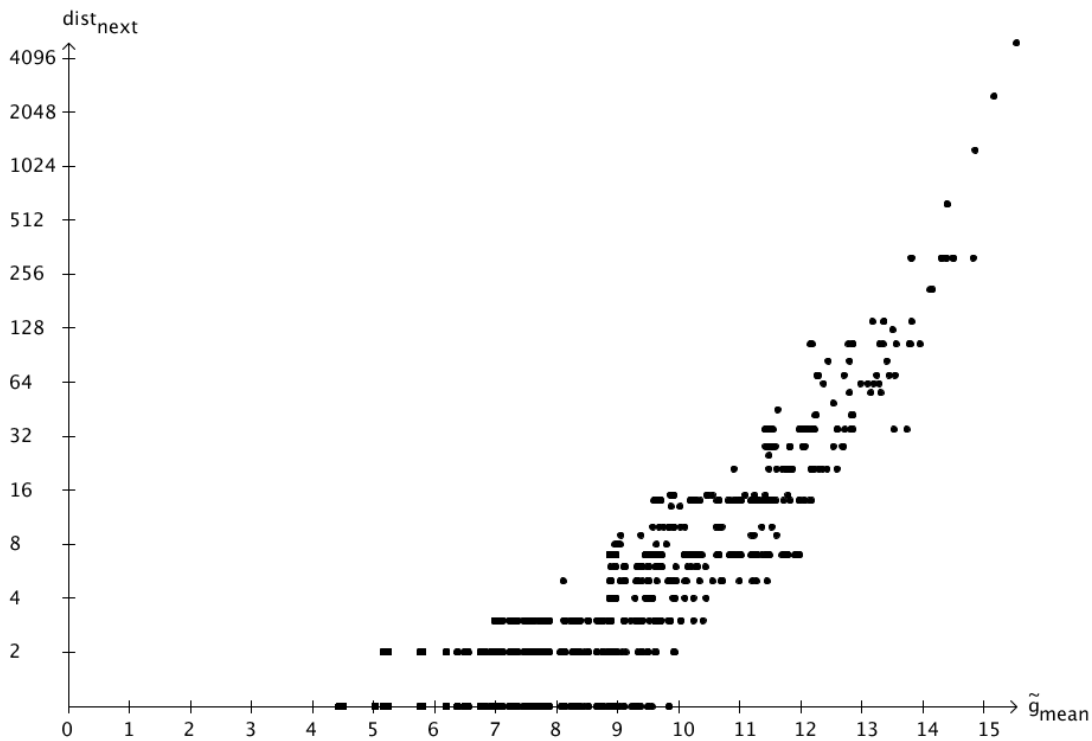

Figure 2. Correlation between $\tilde{\boldsymbol{g}}_{\boldsymbol{mean}}$ and the distance to the next number with in minimum the same $\tilde{\boldsymbol{g}}_{\boldsymbol{mean}}$, see text.

which is infinitely large and diverse. By exercising restraint in constructing or creatively designing a basic element, one has been revealed which is literally ubiquitous and thus corresponds to the ideal of ubiquitous music on an unexpected level.

The next step is to show how this basic element can be brought to sound. The source code in Figure 3 simply goes through the natural numbers and filters the powers of the first four prime numbers out of each number. Like in $\tilde{\boldsymbol{g}}$ the respective potencies that are still considered are also limited upwards here. This is done with a kind of selective division. For this purpose, a number B is formed first, which is made up of the maximum powers of the individual prime numbers 2, 3, 5 and 7 to be considered. In the example here $B = 2^3 \cdot 3^2 \cdot 5^1 \cdot 7^1 = 2520$. Each natural number currently under consideration divides B only with the prime factors 2,3,5 and 7 and this at most only in the potencies 3 for the 2, 2 for the 3, and 1 each for the 5 and the 7. As symbol for this selective division $//$ is introduced. Here are a few concrete examples to avoid misunderstandings: $B//12 = 2 \cdot 3 \cdot 5 \cdot 7$, $B//35 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3$, $B//13 = B$, $B//16 = 3 \cdot 3 \cdot 5 \cdot 7$. In the last example, $16 = 2^4$ exceeds the maximum allowed power of 2, i.e. $2^3 = 8$, so only 8 is effective there. **The result of this selective division is rendered as a note of that frequency** and is played if this frequency is something within the range of the musical instrument (acoustic or electronic) used. The program was implemented in Processing / Java, see example BASIC_Sound_of_N in contributed library "ComposingForEveryone" at https://www.processing.org. To keep it short, no effort was made to achieve a particularly interesting sound. A more elaborate realization of the same thing can be watched as a youtube video: https://youtu.be/e81wd1b3FEE.

Other pieces of music are now created by changing this basic element through the application of arithmetic operations and additionally by using the modulo division and also by performing the above-mentioned selective division for the resulting sequence elements in order to obtain a temporal progression of pitches. In **AOG**, "composing" then

```
/*t: finite section of the natural numbers as a time series*/
for(int t=0;t<1000;t++)
{
  int f = BASENUMBER;
  /*Selective division - Extract primefactors 2,3,5,7 from t*/
  /*and take them away from the BASENUMBER (% is modulo):*/
  while(t>=2 && t%2==0 && f>=2 && f%2==0) {t=t/2;f=f/2;}
  while(t>=3 && t%3==0 && f>=3 && f%3==0) {t=t/3;f=f/3;}
  while(t>=5 && t%5==0 && f>=5 && f%5==0) {t=t/5;f=f/5;}
  while(t>=7 && t%7==0 && f>=7 && f%7==0) {t=t/7;f=f/7;}
  /*Interpretate the result as a frequency and play it:*/
  if(f>=55 && f<=1760) play(f);
  delay(200); /*time delay of e.g. 200 milliseconds*/
}
```

Figure 3. Simple sonification of the natural numbers.

means defining formulas according to which the basic element $id(\mathbb{N})$ is modified. ($id(\mathbb{N}$ means "identity of $\mathbb{N}$": mathematical expression which describes that $\mathbb{N}$ is mapped to itself, i.e. forms a series. In an algorithmic terminology this corresponds to the succession function.) At least for the basic arithmetic operations, the resulting sequence also satisfies the axioms according to Peano and thus retains all the properties that the natural numbers have, and especially also the musical ones (Russell (1920, pp.1-10). In other words: the basic structure visualized in Figure 2 remains similar for these derivates.

But for the division this is only the case if one excludes all the division results that have a remainder. Several formulas can be used to generate several voices. This will be illustrated using a very simple concrete example: The two formulas $f_1 = B//((t+16)mod17)$ and $f_2 = B//(((t*34)mod10)+8)$ with $B = 2520$ and $t = 9000...9022$ provide the two-part musical phrase shown in Figure 4.

The generally occurring harmonic or even contrapuntal interrelationships between two or more voices that are generated by AOG and run polyphonically have not yet been systematically investigated. One could argue: In **AOG** the same sequence t, which is a cutout of $id(\mathbb{N})$ for all voices is the starting point for the following operations. In t all divisors occur with a fixed period, which is equal to their value. Considering this fact, the preservation of a harmonically meaningful relationship can now be shown at least for some special cases, in order to make the existence of a general tendency for this at least plausible: When applying multiplications with numbers consisting only of prime factors greater than 7, the relevant part of the divider-structure of the resulting sequence does not change compared to the original one. Multiplications with small prime factors, like 2 or 3, create a rather consonant relation of elements of both sequences, which are close in time. The same is valid for the addition of numbers, which are rich in the relevant prime factors 2, 3, 5, and 7, because such shifts are widely in phase with the rhythms, in which just these divisors appear.

Figure 5 supplies exactly the miditone pitches from the formulas which correspond to the tones shown in the score. A musical phrase (melody, rhythm, harmony) is created from the few lines of code. Of course, in a program intended for the application, the formulas would not be hard coded, but could be constantly changed by the user via an editor, as for example in a corresponding android app freely available on Google Play (see Figure 6). For the sake of compactness, this editor does not use parentheses, nor does it display the selective division that is always performed. Also the fact that the operations are applied to $t = id(\mathbb{N})$ does not have to be displayed. The example shown here could be displayed in the app just like this: $+16 \equiv 17$ and $\cdot 34 \equiv 10 + 8$, whereby in the editor the symbol $\equiv$ is used for the modulo division.

Having now introduced **AOG** by means of an example, it will now be shown at which points specifications were made which only apply to the specific example but can generally be varied in order to change the characteristic of the generated music. We will also look at the meaning of the operations mentioned from a musical point of view. First of all, it should be noted that despite all the effort that is made here to demonstrate the

*Figure 4. Composing a small phrase on the basis of **AOG** – score.*

```
for(int t=9000;t<=9022;t++) /*Extract from the natural numbers*/
{
 int x = (t+16)%17;       /*formula 1*/
 int y = ((t*34)%10)+8; /*formula 2*/
 int f1 = B;
 /*Selective division 1:*/
 while(f1>=2 && f1%2==0 && x>=2 && x%2==0) {f1=f1/2;x=x/2;}
 while(f1>=3 && f1%3==0 && x>=3 && x%3==0) {f1=f1/3;x=x/3;}
 while(f1>=5 && f1%5==0 && x>=5 && x%5==0) {f1=f1/5;x=x/5;}
 while(f1>=7 && f1%7==0 && x>=7 && x%7==0) {f1=f1/7;x=x/7;}
 int f2 = B;
 /*Selective division 2:*/
 while(f2>=2 && f2%2==0 && y>=2 && y%2==0) {f2=f2/2;y=y/2;}
 while(f2>=3 && f2%3==0 && y>=3 && y%3==0) {f2=f2/3;y=y/3;}
 while(f2>=5 && f2%5==0 && y>=5 && y%5==0) {f2=f2/5;y=y/5;}
 while(f2>=7 && f2%7==0 && y>=7 && y%7==0) {f2=f2/7;y=y/7;}
 /*Mapping the resulting frequencies to midi pitches:*/
 float factor = pow(2.0,69.0/12.0)/440.0;
 int midi1 = (int)round(12.0*log((float)f1*factor)/log(2.0));
 int midi2 = (int)round(12.0*log((float)f2*factor)/log(2.0));
 /*midi==0 means no sound or continue previous tone.*/
 /*Otherwise transpose a half tone upwards to avoid accidentals*/
 if(midi1<52 || midi1>89) midi1=0; else midi1=midi1+1;
 if(midi2<52 || midi2>89) midi2=0; else midi2=midi2+1;
 /*Simple output of the results on the terminal for checking:*/
 println(t+"␣"+midi1+"␣"+midi2);
}
```

*Figure 5. Composing a small phrase on the basis of **AOG**. Transposition up by half a note results as no accidentals are necessary for the representation of the notes. This clarifies the diatonic character of the tone scale resulting from $B = 2520$.*

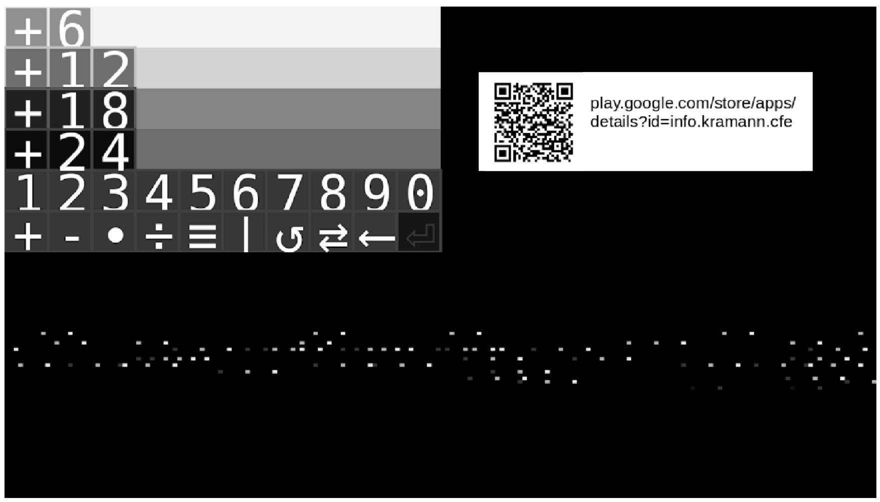

*Figure 6. Android app **CFE** (**C**omposing **F**or **E**veryone) is a direct implementation of **AOG** and can be obtained on google play. An **AOG** formula can be entered for each of four addressable musical instruments using the on-screen keyboard. As direct feedback the resulting piano roll is displayed below. In addition to the basic arithmetic operations and modulo, experimental operations are available which directly manipulate the powers of the prime factors 2,3,5 and 7. Each of the upper four lines of the editor shown controls one of four musical instruments. With the entries to be seen and in the symbolic notation introduced here, the frequencies passed through by these instruments are then calculated as follows:* $f_1 = B//(t+6)$, $f_2 = B//(t+12)$, $f_3 = B//(t+18)$ *and* $f_4 = B//(t+24)$.

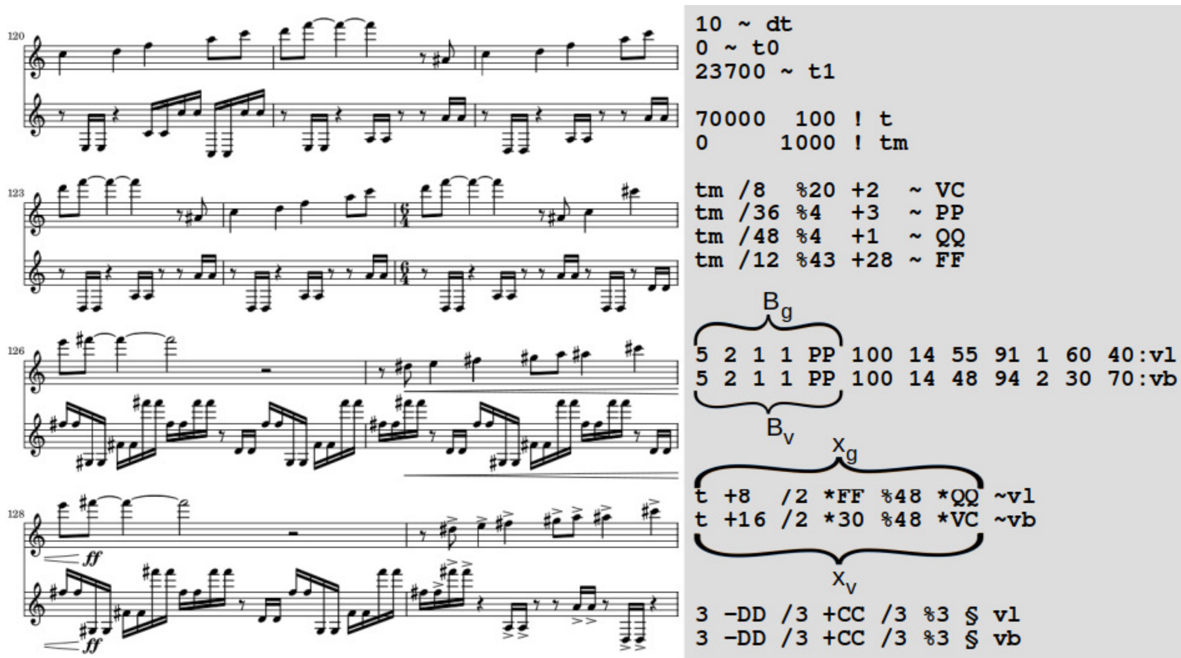

*Figure 7. Excerpt from the score and its **AOG** representation of the composition **AOGdogma#3**. It is a composition for violin and vibraphone of four minutes duration. x_g represents the sequence of operations applied to t for the violin, x_v that for the vibraphone. Thus the frequencies that are passed through on both instruments are calculated as: $f_g = B_g//x_g$, or $f_v = B_v//x_v$. The following applies here: $B_g = B_v = 2^5 \cdot 3^2 \cdot 5^1 \cdot 7^1 \cdot PP$. VC, PP, QQ and FF are drifting parameters, which provide a slow metamorphosis of both the two **AOG** formulas and their base numbers. t is the extract of the natural numbers which are passed through starting with* 70000 *in steps of one with an equidistant time interval of* 100 *milliseconds. tm is a slower counting process for controlling the drifting parameters. By adding a language element for the description of drifting parameters and the possibility to use it at any position of the **AOG** formulas and by providing this together with an editor for real time composition **AOGdogma** gives the possibility to represent whole compositions in a very compact way and to create them interactively (see Kramann 2020a).*

relationship between **AOG** and traditional compositional methods, the method itself is quite simple. The example shown here could easily be reproduced with paper and pencil, apart from the quantization due to the tempered scale. This fact is emphasized here, since the goal was to present a substitute for classical music theory that would be easier to learn than the latter. It should not be misunderstood, however, that the reason why most people will find it easy to learn and understand this method is that they have already learned arithmetic in their school years and have practised the use of equation systems. But even if one would have to learn all this in order to use **AOG**, the gain would still be the incredibly compact formulaic representation of a composition and the possibility to change the character of a composition in its entirety by changing a few symbols. The example given here is very short. However, a piece described by **AOG** is in general potentially infinite long and diverse, because the time sequence of natural numbers forming the input can be continued infinitely and also supplies ever new variants of sequences of successive powers of prime factors. In order not to give rise to the opinion that all pieces created with **AOG** have the minimalist style like the example just introduced, at least one more piece should be presented here as a contrast, which is much more demanding in its musical conception, and which uses more than a pentatonic scale and in which the instruments are playing with many pitch classes. The latter is achieved by the fact that the base number changes in the course of the piece (see Figure 7 and also URL AT MIT PRESS TO COME, CURRENTLY AT https://youtu.be/BYzr9RpfFhc). The character of compositions done with the help of **AOG** is determined by a specification of the following elements:

a) It is specified how many sources of $t = id(\mathbb{N})$ are used.

b) It is specified which operations, in which order, are applied to each t_i.

c) Cases where operations are not applicable can be treated in various ways. For example, a pending division by zero could either result in not playing a note at all on the current tick due to the formula the operation is in, or the operation could simply be skipped. Instead of omitting decimal places after a division, the operation could be skipped when the division occurs, and so on.

d) It is specified according to which operations a sound conversion takes place by applying the selective division: If the result after each operation is used to produce sounds, one obtains exactly as many voices as there are operations. This can be used directly with polyphonic instruments, such as the piano, to control their dynamics:

If the same frequency appears as a result of several operations simultanously these can overlap to form a louder tone. Or, alternatively, you can proceed in such a way that, along a sequence of operations, only the last playable tone that came out is heard. In the latter case, the number of voices is identical to the number of active formulas.

e) The base number is specified. The base number is mainly responsible for which frequencies can be generated at all with the respective **AOG** mechanism. At this point it should be mentioned once more, that the result x, which is a result of applying arithmetic operations to the sequence of natural numbers, for example $x = t/7 + 3$, then appears in the denominator at selective division. The numerator is the base number. To limit the type and count of the considered prime factors, it would have been possible to extract them directly from x and interpret this result as frequency instead of introducing selective division. Symbolically this could then be expressed like this: $f = B * *x$. In fact, both are used in the somewhat more sophisticated composition elegie (see #1 at Kramann 2020b): If $f = B//x$ leads to no playable result, but $f = B * *x$, the latter is used. An advantage of selective division compared to pure selection is that even very small x results in playable frequencies. The harmonic relations of two numbers do not change, no matter if you take them directly or as reciprocal values.

f) The calculated integer frequencies could be made to sound directly, or could be quantized due to the tempered tuning. In the latter case, the frequency range could be compressed or stretched by a certain factor before this quantization, and after that the entire composition could be transposed at will, as e.g. in the example above by a half step upwards. All three elements together, i.e. the base number, a stretch factor and a transposition, can be used to produce a certain desired tone scale. This tone scale could also be microtonal without any problems.

g) All specifications made so far can be changed (slowly compared to the sequence of the time sequence $t = id(\mathbb{N})$ over time.)

h) In the postprocessing layer there should be an algorithm for the musical interpretation of the incoming notes to be played. In concrete, a mechanism has already been implemented which adjusts playing technique (staccato, legato, etc.) and dynamics according to the tension ratio of a new tone's frequency, related to the immediately preceding tones. What is more, commercial physical modeling software was used for the sound generation. All the pieces of music to which reference was made at the beginning and some more are representatives of this approach (see #1, #4, #5, #7 at Kramann 2020b).

For the effect of the individual operations, direct correspondences can in some cases be shown to certain traditional musical forms:

a) An addition ensures that a sound event is brought forward in time. An addition of 3 turns $0, 1, 2, 3, 4, 5$ into $3, 4, 5, 6, 7, 8$. If $+3$ is entered for one voice and $+6$ for the other, this results in a kind of two-part canon of imitation. After these operations the above mentioned selective division, followed by a rendering of the resulting number as a note of that frequency is of course always done. In the introduced formula notation you then have a melody in the simplest case, which results from $f_i = B//(t_i + 3)$ and a second one resulting from $\tilde{f}_i = B//(t_i + 6)$, whereas the latter one is three ticks ahead of the former one. The following points should be understood in the same sense.

b) Divisions. Compared to the unchanged voice, when e.g. a division by 2 is performed the new voice is two times slower. This corresponds to the musical form of an augmentation canon.

c) Modulo or Rest-Division. The result of this operation is always the integer remainder of a division using the same operand. In modulo 6, for example, from $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ the sequence $0, 1, 2, 3, 4, 5, 0, 1, 2, 3$ is obtained. The musical structures that arise here are reminiscent of the repetitive phrases typically found in minimal music.

User Interfaces

As examples, two user interfaces are described below, which are both based on **AOG**: a formula editor, which is more suitable for adults, and a system in which the respective composition is created by decorating balls with colored tape, beads, pompoms and other things. The latter interface is intended to be used by primary school children.

People at a TEDx event tested the formula editor (see #2 at Kramann 2020b). Changing groups of four from the audience could enter **AOG** operations via android tablets, each assigned to a voice. The design of its user interface corresponds to that of the android app already shown in Figure 6. The formulas entered were converted into music by a PC networked with all tablets via W-LAN according to the **AOG** method. Likewise, all currently valid formulas were displayed visibly to all those present with the aid of a projector. The four formulas were displayed in colour-coded form and, based on a previous introduction, the participants should have been familiar with the link between a formula and the musical instrument assigned to it whose individual sound was chosen quite different from any of the other three. Through this feedback a certain exchange of information was established among the participants after some time and a certain learning effect could be observed: The participants started to copy formulas of other participants if they liked the result and then experimented again with variations of this basic form. The functions of the editor and the basic effects of the individual operations on the musical result were explained to the audience in the previous lecture. To what extent the lecture was consciously referred to could not be determined within the twenty minutes in which the audience experiment took place.

The observed casual collaboration by copying and varying was possible with the system because the symbolic representation of the resulting composition is so compact that it could be grasped at a glance and also adopted very quickly.

The promotion of this type of cooperation by copying and varying through an easy to capture form of feedback is also a central feature of the second user interface presented here, "THE FLIPPIN’ POMPOMs" (**TFP**). The role of the formulas, which are visible to all, is played in TFP by colourfully decorated balls, which are kept visible to all, and each ball represents a composition. Some examples are shown in Figure 8. More and also a video of the system in action can be found under #5 at (Kramann 2020b). However, due to the Covid-19 pandemic, the practical experiences with **TFP** are so far limited to small exemple experiments with children. In particular, an already planned use of the system presented here within a STEM project day at a primary school had to be postponed indefinitely.

The balls can be designed according to compositional considerations. For example, there is a ball in which a pompom is pasted with coloured areas. When ever the pompom

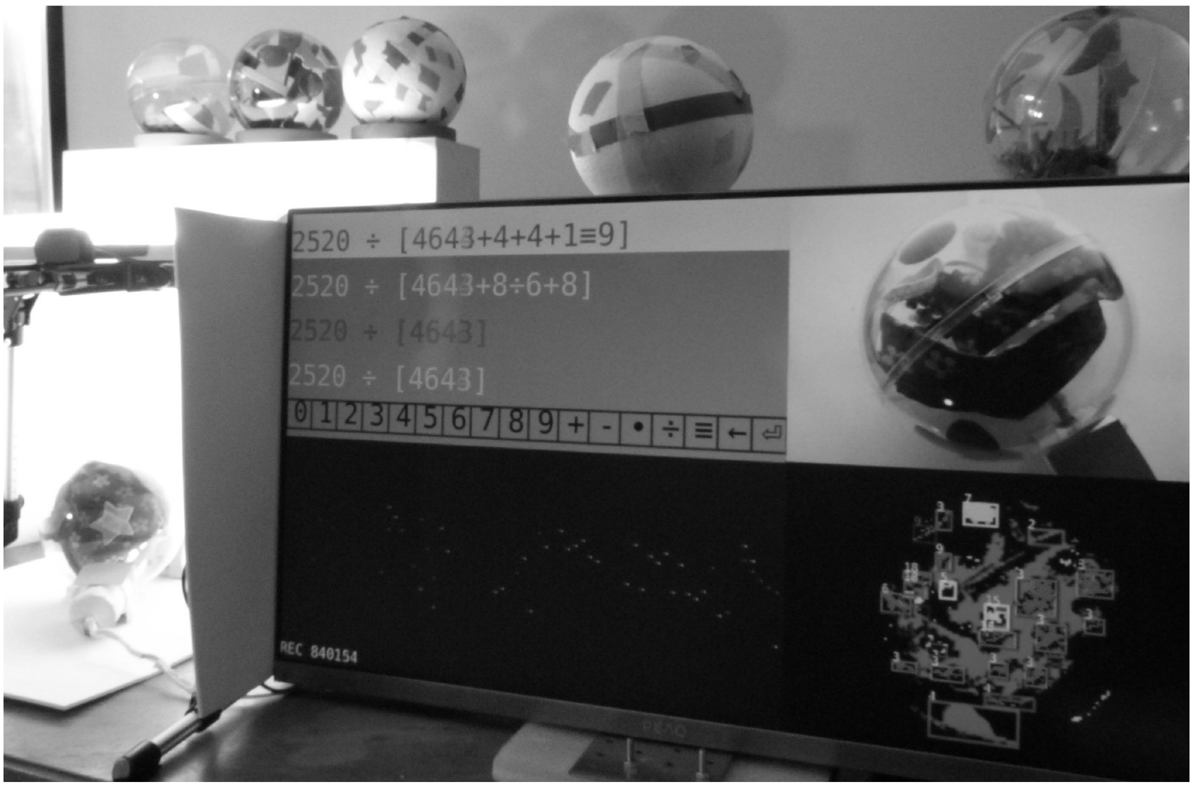

*Figure 8. Setup for THE FLIPPIN' POMPOMs. In the picture above you can see some variations of spheres. On the bottom left of the picture the motorized device for automatic rotation of a ball is shown. On the top right of the screen the image captured by the camera of the ball currently resting on the rotating device is shown. Below, also on the screen, the segmented colored areas of this sphere can be seen. Finally, the **AOG** formulas resulting from this segmentation can be seen on the left side of the screen. These can also be edited directly.*

flips, a new variant of the musical structure appears. The resulting balls are placed on a motorized support that rotates the balls very slowly around two axes, while a webcam continuously records the changing view and passes it on to the software as a pixel image. Since this textual description may give an overview of the functionality, but it is not possible to reproduce every detail exactly, the source code for **TFP**s was provided as an example within the contributed Processing Library "ComposingForEveryone" (see #9 at Kramann 2020b). In this software the image is filtered in such a way that there is only a black background and coloured shapes in red, green, blue and yellow (Figure 9 a). In a next step, single-colored contiguous areas are segmented. Several such areas can be connected again and form a colorful contiguous area. Or they may exist separately without connection to other areas. For areas that are composed of several colored areas, the number of areas in each color is counted. Due to a color coding in which blue represents the two, red the three, green the five and yellow the seven a multiplication of all the prime numbers assigned to the areas with each other is done, and one obtains exactly one number for each individual contiguous colored area (Figure 9 b,c). Now it is examined which arithmetic operation is best suited to get from one number to another. Which of the areas are picked out one after the other is decided on the basis of neighbourhood relations and the absolute size of the areas in pixels. When selecting the "suitable" operations, the system first looks to see whether a modulo operation is possible in which the operand c lies somewhere between the two values a and b. The relation between a, b and c is then: $a \ modulo \ c = b$ and the idea behind this is, that only if $a > c > b$ then b can be a real remainder of the modulooperation $a \ modulo \ c$, like for example at $100 \ modulo \ 30 = 10$. If this fails, it is checked whether multiplication or division is possible. If this also fails, an addition or subtraction is always possible to get from one number to another (Figure 9 d). Because the nearest neighbor is always taken from the largest areas, a path is created from a maximum of four largest areas, which runs over different colored areas. The different paths can also run over the same areas to some extent. This has proven to be very beneficial for the quality of the music, probably because the resulting voices have a lot in common, but also differ slightly. This has not yet been examined in more detail. Each of the sequences of operations found then represents a formula in the sense of **AOG**. The further processing of these formulas up to the conversion into sounds is done from this point on in the same way as with the formula editor, i.e. as described before in the chapter about **AOG** and as it is visualized in Figure 9 e,f,g,h.

In fact, the software for **TFP** is based directly on that of the formula editor. This goes so far that the latter is still displayed in **TFP** and changes to the formulas can also be made there. Thus, a multimodal user interface is available here, which provides possibilities for influencing the formulas on two possible levels, whereby the ball design always represents the entire composition and the currently appearing formula typically represents an actual phrase that varies slightly in the course of time. In particular, however, **TFP** always keeps the AOG-related perspective open and thus offers users the opportunity to theoretically understand the underlying compositional principle and to emancipate themselves from the use of the given tool.

Discussion

In order to open up access to composing with AOG, especially for primary school children, which even does not require knowledge of school arithmetic advantage of the fact is taken that arithmetic operations ultimately help to handle sets. The way back to this basis was gone by letting the children arrange colorful forms into groups. In terms of data sonification, they create a kind of data set. At first the following was noticed: Since the compositions are realized with spheres that are decorated with the colored surfaces, the three-dimensionality in the 2D camera image results in different topologies of the colored groups, depending on the perspective from which the webcam has just viewed the sphere.

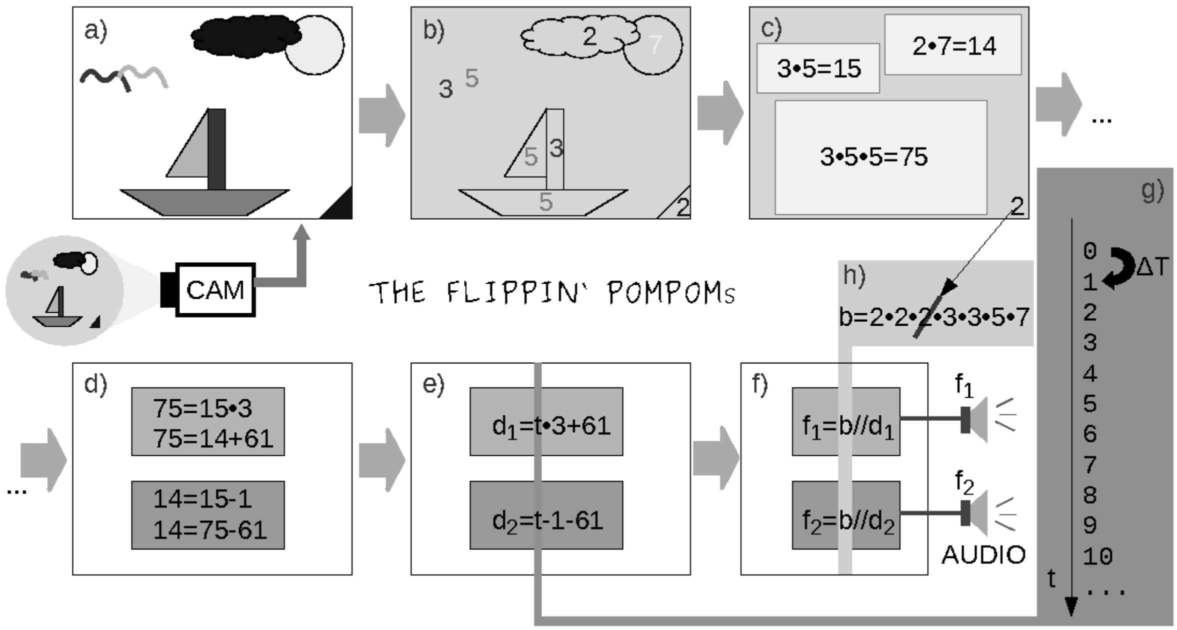

Figure 9. Specific processing scheme of THE FLIPPIN' POMPOMs from image capture to sound generation (see text).

This means that this technique, which does not deal with elements from a musical context, nevertheless adds something that has a lot to do with composing (set of variations). Such an approach would never be directly based on a representation like that in a score. Thus, here is a preparation of the object to be manipulated under a very special point of view, a very special perspective. This is probably the deeper reason for the usefulness of the so-called "facade pattern" as it is used in software development. In the search for descriptive possibilities to characterize meaningful settings for ubiquitous music, design patterns were also considered (Keller et al. 2014, pp.xi-xxiii).

The attempt is now being made to transfer something from the world of design patterns, which elements often characterize software components, to the overall system presented here, which includes not only software but also the user and hardware. The facade pattern represents a structure that seems to correspond to the previously described setting: It simplifies the use of a subsystem consisting of many components by allowing a

Figure 10. Decorating transparent and not-transparent spheres as a compositional method.

client to ensure the completion of common tasks via a single method instead of making a multitude of method calls to different elements of the subsystem to achieve the same goal. The non-standard client still retains the ability to make calls at the subsystem level that are not bundled in the facade (Gemma et al. 1995, pp.185-193). The role of the client is played here by the users of THE FLIPPIN' POMPOMs (**TFP**s). The balls to be decorated represent the facade and the subsystem controlled by them is all the rest, consisting of the formula level down to the sound generation, see Figure 8. It is possible to bypass the facade by directly typing in **AOG** formulas and by getting feedback directly from the subsystem in the form of the perceptible musical event, its visualization as a piano roll and by viewing the **AOG** formulas (see Figure 11). This means that at the beginning of working with **TFP**s, the facade completely hides the subsystem: the children tinker with balls, which they decorate with colored shapes. Little by little, via the feedback channels, the user (client) gains a deeper knowledge of the connections behind the facade (ball). Little by little, certain contexts are recognized, how certain configurations of the balls can influence the sound:

While a transparent sphere is spinning smooth, it can happen that an object inside the sphere suddenly falls from one side to another, if it has more the shape of a cuboid than a sphere. Such a change in the shape of an object leads to sudden changes. For this type of twisting areas can be covered by others and new combinations of coloured areas can emerge. Moving elements within the ball can gradually combine with other elements of the ball surface through the rotation of the ball and thus vary the sound event. The colours determine which instrument dominates. Isolated, single-color areas are used to influence the base number. There may be areas with very complex and less complex patterns etc., see Figure 10. From the perspective of a person who just starts with TFP all this happens at the beginning either by chance or by intentional reference to the visual form rather than the sonic one. In the best case, however, over time these relationships become increasingly clear and exploited by the user. On the level of the facade-explanatory model, this would mean that instead of an initial very simple method of accessing the functionality of the subsystem, more sophisticated methods would gradually emerge that pass more parameters, or else these parameters have always been passed, but gradually the user notices that they are there. TFPs can be used to convey the meaning of what it is like to compose and to make the experience of holding your own composition (as a ball) in your hand and making it heard whenever you want.

Conclusion and Further Work

In this paper we explained in detail what Arithmetic Operation Grammer is and what potential it has in combination with suitable user interfaces to give laypersons the opportunity to compose their own music. But what contribution does this work make in the field of ubiquitous music and sonification? This should be made more explicit in this concluding chapter.

From the perspective of sonification, the method presented here differs from other

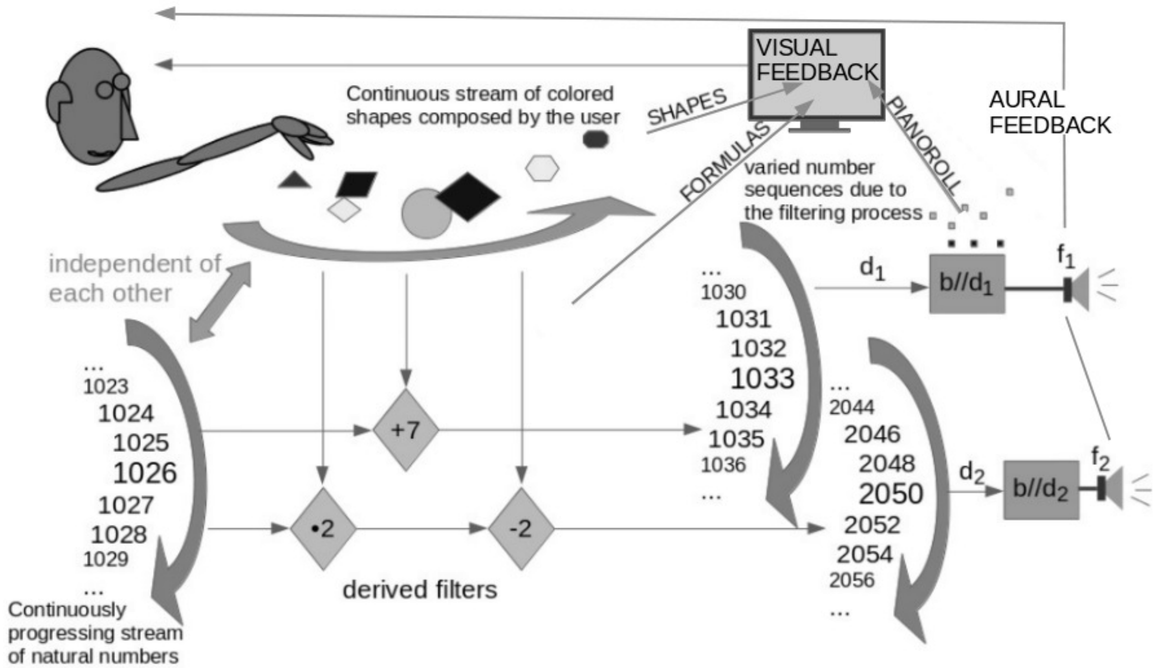

Figure 11. Abstract processing scheme of THE FLIPPIN' POMPOMs from image capture to sound generation (see text).

sonification methods in that time does not have to be present as an order parameter in the input data, how this would be the case with an image. A time parameter – as it would be the case if the input was represented by a video – is in any case a prerequisite for sonifications that do not themselves have a temporal macrostructure (see e.g. Braund and Miranda 2013, Denjean et al. 2019). The microstructure, that is, the actual sound, does not have to be obtained directly from the input data either, as is the case with sonifications that essentially compress or stretch the system time of the source data in order to obtain a frequency range that is audible (like in Holtzman et al. 2013). These independencies predestine the procedure proposed here for sonification of data which itself does not contain time as an order parameter, and none of the other parameters must be interpreted as time here. However, this does not exclude the possibility of sonifying data in which time appears as a parameter. Also because of this independence, the method presented here allows a free choice in the design of the sound and thus the possibility of using high-quality software-based musical instruments. And finally, the resulting macrostructure meets higher musical demands, which is not exactly true of many common sonification methods, see e.g. Matsubara et al. 2019. Overall it can be said that there could be application niches for the procedure presented here in the field of sonification.

Since, as described above, **AOG** in the sense of Chomsky's classification is an example of a 3rd order grammar, it is not necessary to explicitly acquire a theoretical apparatus of rules even if at some point one emancipates oneself from the provided software tool and goes one's own way on the basis of **AOG**. Unlike comparable software tools, complexity is not hidden in the above mentioned way, but the underlying mechanism is per se simple and easy to learn.

In the work of Gil Weinberg (Weinberg 2002), for example, this withholding of a theoretical background is indeed also discussed and then justified by the assumption that laypersons would study differently from professionals, and that the underlying theory would overwhelm the target group and spoil the joy of composing. Finally it is stated there that such first experiences can give reason to deal with music theory at some point. But that claim is formulated as hope rather than as a consequence of the use of the described composition tool.

> "I believe that providing novices with the power to create and phrase a melody by manipulating its contour, regardless of its exact pitches and intervals, offers them a unique creative experience that is usually reserved for experts and that can serve as an entry point for further investigations into more advanced concepts such as harmony and counterpoint." (Gil Weinberg, in Weinberg 2002)

It would of course be possible at this point to put forward a number of counter-arguments to the arguments given against the teaching of a musical theory to laypersons, such as the obvious fact that professionals no longer need to study music theory at all and that lay people are not necessarily associated with a lack of comprehension. On the other hand, one has to give credit to the last quoted approach and comparable ones that they have successfully tried to open up the possibility for novices to also record musical ideas (compare the quote above). This possibility is barred to **AOG** users in its current form. The dominant experimenting with formulas and forms is not compatible with it. However, instead the two user interfaces presented here each provide, a very compact, easy-to-grasp representation layer of the emerging composition. In the one case it is the symbolic **AOG** formula, in the other case it is the colorfully decorated spheres that represent entire compositions. The above-mentioned audience experiment has also shown that it is precisely this compactness of the representation that greatly promotes creative cooperation between the people involved, because on this basis an exchange of ideas between the participants is strongly promoted. This shows a certain affinity to the tagging mataphor (Keller et al. 2014, p. xviii). In connection with this metaphor, it is pointed out that creative activities take place through interaction with material or mental objects and that these are at best designed to form a suitable channel for these interactions.

So one can say on the whole that simply *various* approaches to helping laypersons to compose can be successful in their own way, in the sense that ...

> "Music is inclusive, and musics and their techniques and forms are cumulative, not mutually exclusive." (Laurie Spiegel, in Spiegel 1998)

But with respect to the work presented here, one could object that, for all the universality of the set of natural numbers, $\mathbb{N}$, the special operations introduced and applied to it indirectly result in a pre-selection of the sound material provided, and that, furthermore, the musical examples presented all have a certain characteristic style that one may or may not like. At this point it is up to me now to express my hopes and convictions, which, however, with reference to work that has already been continued but not yet completed, can be presented in conclusion in a well-founded manner:

The limitation to $id(\mathbb{N})$ as the basic element was mainly done because it was possible to prove the properties described as musical. If one drops this restriction and goes over to the use of other types of potentially infinite series, then again completely different perspectives arise with regard to the results that can be achieved with it (as can be seen from the examples here: see #10 at Kramann 2020b). As a preliminary justification for this step it may be said that all these other series can be represented as mathematical mappings of the basic sequence $id(\mathbb{N})$. Whether it will be possible to extend the theory introduced here by such elements while at the same time preserving the current very compact form remains to be seen.

In the meantime, more extensive research has been conducted into the relationship between compositions generated with AOG and traditionally composed music, in which sound events with specific pitches are organized in time. A corresponding experiment can be found at #8 at Kramann 2020b). By means of an optimization algorithm, an attempt was made to optimize a group of four **AOG** formulas, each containing 16 operations, in such a way that for the length of four ticks at a certain starting value t_0 they represent the generally known canon *Frère Jacques*, if all four voices have already begun there. This was successful, but in the ticks before, the musical set obtained with **AOG** converges towards the desired piece and immediately afterwards diverges away from it. This can be well followed in the enclosed sound conversion. As a preliminary result, there is a tendency that a solution of this inverse problem is possible in principle, but that it is by no means trivial. In order to achieve better results here, the next step is to test alternative operations.

It may also be criticized that the user interfaces presented here, which are more oriented towards pictorial design, do not convey the very satisfying experience of experiencing a direct sonic reaction through an action. However, feedback mechanisms, which have the goal of tracing such sound events directly generated by the user back to **AOG** formulas that could have generated them in order to obtain automatic accompaniment, are currently also the subject of further development of the possibilities that arise with AOG (corresponding examples can be called up here: #7 at Kramann 2020b). The advantage of the offline approaches, where a composition is described but not played, is that they always offer the possibility to embrace, distribute and be handled cooperatively with participants who are far away from each other, since latencies do not play a major role. In particular, the automatic and parallel sound conversion can also take place asynchronously at the respective locations. This possibility is also mentioned by Lazzarini (Lazzarini et al. 2014) for the design of Ubiquitous Music Systems.

Finally, and not least of all, **AOG**'s explicitly comprehensible connection between music and mathematics offers welcome starting points for the planned **STEM** project day and all those who may follow it, in order to turn it into a **STEAM** project day (with **A** for **A**rts) in the sense of John Maeda (Maeda 2013), thus breaking the ground for a way of thinking that goes beyond the well-worn cathegories of isolated disciplines.

References

Almeida, I. C., G. Cabral, and G. B. Almeida. 2019. "AMIGO: An Assistive Musical Instrument to Engage, Create and Learn Music." *International Conference on New Interfaces for Musical Expression, Retrieved August 5, 2020, from http://www.nime.org/proceedings/2019/nime2019_paper033.pdf* :168–169.

Biles, J. 2007. "Improvizing with Genetic Algorithms: GenJam." *Evolutionary Computer*

Music :137–169.

Braund, E., and R. Miranda. 2013. "Music with Unconventional Computing: A System for Physarum Polycephalum Sound Synthesis." *Sound, Music and Motion, CMMR 2013, LNCS 8905* :161–174.

Busch, H. 1970. "Leonhard Eulers Beitrag zur Musiktheorie." Regensburg: Gustav Bosse, p. 34.

Chomsky, N. 1956. "Three Models for Description of Language." *Information Theory, IRE Transactions, 2(3), Retrieved April 4, 2019, from https://chomsky.info/wp-content/uploads/195609-.pdf* :113–124.

Chomsky, N. 1959. "On Certain Formal Properties of Grammars." *Information and Control, 2, Retrieved April 4, 2019, from http://twiki.di.uniroma1.it/pub/LC/WebHome/chomsky1959.pdf* :137–167.

Denjean, S., R. Kronland-Martinet, V. Roussarie, and S. Ystad. 2019. "Zero-Emission Vehicles Sonification Strategy Based on Shepard-Risset Glissando." *CMMR 2019, Retrieved Mai 25, 2020, from https://cmmr2019.prism.cnrs.fr/Docs/Proceedings_CMMR2019.pdf* :966–976.

Figueiró, C., G. Soares, and B. Rohde. 2019. "ESMERIL – An interactive audio player and composition system for collaborative experimental music netlabels." *International Conference on New Interfaces for Musical Expression, Retrieved August 5, 2020, from https://www.nime.org/proceedings/2019/nime2019_paper034.pdf* :170–173.

Garcia-Valdez, M., L. Trujillo, d. F. Vega, J. Guervos, and G. Olague. 2013. "EvoSpace-Interactive: A Framework to Develop Distributed Collaborative-Interactive Evolutionary Algorithms for Artistic Design." *Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART 2013)* :121–132.

Gemma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. *Design Patterns*. Boston: Addison-Wesley.

Holtzman, B., J. Candler, M. Turk, and D. Peter. 2013. “Seismic Sound Lab: Sights, Sounds and Perception of the Earth as an Acoustic Space.” *Sound, Music and Motion, CMMR 2013, LNCS 8905* :161–174.

Hu, H., and D. Gerhard. 2019. “Modelling 4-dimensional Tonal Pitch Spaces.” *CMMR 2019, Retrieved Mai 25, 2020, from https://cmmr2019.prism.cnrs.fr/Docs/Proceedings_CMMR2019.pdf* :38–50.

Jakobsen, K., M. G. Petersen, M. K. Rasmussen, J. E. Groenbaek, J. Winge, and J. Stougaard. 2016. “Hitmachine: Collective Musical Expressivity for Novices.” *International Conference on New Interfaces for Musical Expression, Retrieved August 5, 2020, from https://www.nime.org/proceedings/2016/nime2016_paper0048.pdf* :241–246.

Keller, D., V. Lazzarini, and M. Pimenta. 2014. *Ubiquitous Music*. Heidelberg: Springer.

Kramann, G. 2015. “An Overtone Based Algorithm Unifying Counterpoint and Harmonics.” *Music, Mind and Embodiment (CMMR 2015), Retrieved August 20, 2015, from http://cmr.soc.plymouth.ac.uk/cmmr2015/proceedings.pdf* :791–805.

Kramann, G. 2019. “Generative Grammar Based on Arithmetic Operations for Realtime Composition.” *CMMR 2019, Retrieved Mai 25, 2020, from https://cmmr2019.prism.cnrs.fr/Docs/Proceedings_CMMR2019.pdf* :346–360.

Kramann, G. 2020a. “AOGdogma - Real Time Composition Framework on the Base of AOG.” *Retrieved September 14, 2020, from http://www.kramann.info/90_AOGdogma* .

Kramann, G. 2020b. “Arithmetic Operation Grammar (AOG) - Selected and thematically arranged examples.” *Retrieved August 10, 2020, from http://www.kramann.info/98_AOG* .

Lazzarini, V., D. Keller, C. Kuhn, M. Pimenta, and J. Timoney. 2014. "Prototyping of Ubiquitous Music Ecosystems." *Fifth Workshop on Ubiquitous Music (V UbiMus), Retrieved August 12, 2020, from https://seer.ufrgs.br/cadernosdeinformatica/article/view/v8n4pp69-80* .

Maeda, J. 2013. "STEM + Art = STEAM." *The STEAM Journal, Retrieved August 13, 2020, from https://scholarship.claremont.edu/steam/vol1/iss1/34/* 1:1–5.

Matsubara, M., Y. Morimoto, and T. Uchide. 2019. "Auditory Gestalt Formation for Exploring Dynamic Triggering Earthquakes." *CMMR 2019, Retrieved Mai 25, 2020, from https://cmmr2019.prism.cnrs.fr/Docs/Proceedings_CMMR2019.pdf* :983–987.

Russell, B. (1920) 1993. "Introduction to Mathematical Philosophy." *Second edition. New York: Macmillan. (Reprint, Mineola, New York: Dover. Citations refer to the Dover edition.)* .

Schneider, A., and K. Frieler. 2008. "Perception of Harmonic and Inharmonic Sounds: Results from Ear Models." *Computer Music Modeling and Retrieval. LNCS, vol. 5493* :18–44.

Souza Stolfi, A. d., M. Ceriani, L. Turchet, and M. Barthet. 2018. "Playsound.space: Inclusive Free Music Improvisations Using Audio Commons." *International Conference on New Interfaces for Musical Expression, Retrieved August 5, 2020, from https://www.nime.org/proceedings/2018/nime2018_paper0050.pdf* :228–233.

Spiegel, L. 1998. "Should Music-Making Be Reserved for an Elite?" *Computer Music Journal* 22(1):6–7.

Supper, M. 2001. "A Few Remarks on Algorithmic Composition." *Computer Music Journal* 25(1):48–53.

Weinberg, G. 2002. "Playpens, Fireflies and Squeezables: New Musical Instruments for Bridging the Thoughtful and the Joyful." *Leonardo Music Journal* 12:43–51.