Generating Earcons from Knowledge Graphs

Enya Nieland
Vrije Universiteit
Amsterdam

Abstract

For this research another modality is implemented to represent Knowledge Graphs (KGs). Currently there are visual tools available to represent the connections of a KG. However, it can be difficult to get an insight into the complexity of a KG by only using a visual representation. We found that other researches have described the use of sound instead of visuals to create an insight into data or processes. Sound can represents the metrics of KGs in the form of an earcon. Earcons are an auditory equivalent to icons, they are known to be a short sequence of notes that differ in pitch, duration and dynamic. The contribution of this research is to create earcons based on mappings between the values of a KG and properties of sound. For this research the earcon consists of three notes that differ in pitch, rhythm or a combination of both. To test whether there is a difference in performance between pitch and rhythm, we created three types of earcons. The first type only implements different pitches, the second only implements different rhythms, whereas the last combines both the pitch and rhythm. We evaluate the different versions through a user study, where we present the user with two earcons of each version. Based on the user study we can conclude that it is possible to create an insight into KGs using sound. But, not all types of earcons worked equally well. The earcon that combined both the rhythm and pitch worked the least, however, this earcon was preferred by most respondents. The reason for this could be that the respondents have to listen to differences in two domains at the same time, pitch and rhythm, which could throw them of. On the other hand, since it implements both different pitches and rhythms this is the most melodic type of earcon, which could be the reason that most people preferred this type.

Keywords Knowledge Graph, Metrics of Knowledge Graphs, Properties of Sound, Earcons, Sound as modality

1 Introduction

The use of Knowledge Graphs (KGs) have become more popular during recent years. KGs provide structure to data by creating links between data. This structure makes it possible for computers to interpret and use the data. For this research we interpret the term KGs as a network of interconnected data. The problem with KGs, however, is that they can become very large and complex. This can make it, for example, difficult to find out how all data is connected or how many links each node has. For this purpose the use of visual tools has already proven to be beneficial in overcoming this problem. An example of such a visual tool is Thinkpedia [11], which visualizes the data of DBPedia. This visual representation tool, called Thinkpedia, allows for a user to zoom in on a specific node and see how the node is connected. Furthermore, it also can show an overview of the entire KG. This insight is created by showing the data and its connections. To determine the diversity and complexity of a KG, values such as the number of nodes of the graph or the degree of nodes can provide some insights. However, when the graph is very large even a visual representation tool can make it difficult to quickly find out what the values of the metrics of KGs are. Because it can be easy to get lost in the web of nodes. Therefore, the purpose of this research is to try to find another modality, than visuals, that can be applied to the metrics of KGs. In this research we investigate the use of sound to this end.

In the case of code debugging it has already been proven that the use of sounds instead of visualizations is can be interpreted as more beneficial. It is discussed that the use of sound can provide an even quicker feedback, as opposed to the use of visual objects [8]. Furthermore, the use of sound for debugging code is said to add value to the debugging process of developing code and, because it is beneficial in finding faults which one cannot always see [20]. The purpose of adding sound to the debugging process was to find out where code gets stuck. For example, hearing that the code gets stuck in a loop can be found out quick because one will hear the same loop of notes over and over again. Furthermore, as described by Brown et al. [4] the use of sound does also create an ease of use for people who are visually impaired. Even though, visually impaired people are not able to see the data presented in, for example, a graph, they are able to hear sounds. This means that they can hear the difference between the different data ponits of a graph. Thus, it can be discussed that the use of sound has multiple benefits when applied to KGs.

To find out if the use of sounds is helpful to figuring out the metrics of KGs, such as the number of nodes or the number of connections, the following research question needs to be answered:

RQ1: How can the use of earcons create an insight into the metrics of KGs?

However, before this question can be answered, we first need to find out what the possibilities are of creating a mapping between the properties of sound and the metrics of KGs.

That is why, the following sub questions are important to answer as well:

RQ2: What are the possible mappings between the properties of sound and metrics of KGs?

RQ3: How can a tool use these mappings to create an insight into the metrics of KGs?

2 Related Work

In this section the literature related to the topic of using sound and KGs is discussed. First, an explanation of KGs is given. Next, currently available tools for KG visualization are explained. The third subsection provides an explanation of the added value of sound and other applications where the use of sound was beneficial. Lastly, an overview will be given of the found properties of sound and metrics of sound.

2.1 Meaning of KGs

The amount of data on the web has increased tremendously in recent years [6]. Most of the data on the web is unstructured, which means that a lot of this data cannot be processed by machines. If the data would be structured this could help to improve search results, or make it possible for machines to derive logic and new insights from it. Therefore, it is necessary to structure the data. KGs are used to represent connections between data. Take for example DBPedia [13], which is a structured version of Wikipedia. Wikipedia contains a human knowledge that is created through collaboration in approximately 300 language variants. Therefore, there could be an article about the exact same topic in each of these languages. However, the content of the article per language could be different, because it is created collaboratively. This could also entail that there is a lot of new data per language, but also a lot of duplicate data. DBPedia creates connections between the data provided in these articles, such as the birthplace and year of van Gogh, that he was an artist and that there is a museum that contains his art. Since the data has been structured by DBPedia it is now available to be used by other applications, such as search applications [15].

For this research, we assume KGs are available in the Resource Description Format (RDF). Here, connections between data entities are made and then stored in an RDF triple to make it readable for computers. The entities of the data are used as nodes and the relations between nodes are always directed and labeled [18]. For example, the relation between a book and its author can be expressed through the triple isWrittenBy. To be able to create such a KG, data needs to be structured. However, most data on the web is very unstructured, contains a lot of duplicates and can contain blank nodes. There are already a lot of knowledge bases (KBs) available in this RDF format, such as DBPedia [13], Freebase [3], Yago2 [12] and WordNet [16]. There is also a tool that cleans structured datasets in this format and republishes it as RDF, namely LOD Laundromat [1]. LOD Laundromat crawls through the Linked Open Data (LOD) cloud and cleans the content, so that it, for example, does not contain any duplicate information or blank nodes anymore. Furthermore, the LOD Laundromat tool also has a SPARQL endpoint that allows for people to query the metadata of a dataset. This SPARQL endpoint will allow us to find properties related to many datasets in one central place, such as indegree, outdegree and many more. Therefore, we will use the cleaned data provided by LOD Laundromat as well as the metadata provided for this data.

2.2 Interaction with KGs

As described by Ehrlinger and Wöß [5] a KG can be described as a tool that acquires information and transforms it into an ontology. The generated ontology can then be used by machines to derive knowledge. These KGs are not only readable for computers, but also for humans which makes it very beneficial. These graphs can, for example, contain connections about a family, mathematical information or even information about an artist and his art. This is similar to the use of the Friend of a Friend vocabulary, which can describe a social network based on social media connections [10]. Depending on the amount of friends one is linked to on the social media page, these networks could be very small or extremely large. Because it cannot only tell who you are linked to, but go even further and find out who your friends are connected to. To see the size of such networks, visual representations could be created. It is already possible to traverse a KG visually. Tools such as introduced by [11] called Thinkbase and Thinkpedia provide an interactive graph-based representation of semantically enriched data on the web. These tools create a graph based on content on the web to create an insight to the connections of this data, which is not always visible when seeing it in a textual form. Furthermore, there is also a tool called Wivi [14] which generates a graph based on wikipedia articles a user has read. It shows the articles one has read as well as the articles that are related to those articles. In this situation it is used as a type of recommender system based on a user's session. The benefits to providing visual representations of knowledge is that it can give new insights into which data is connected. Similarly to the use of Thinkbase and Thinkpedia the use of sound for KGs should create an insight into the data of a KG. However, these tools are all based on visualization of KGs, which still leaves some room for improvement by applying the use of sound instead of visuals.

2.3 Sound as Modality

Hearing is often described as being the secondary sense to seeing [9]. It gives a much quicker feedback, than visual objects [8]. Without the use of sound it could feel as though knowledge is missing [17]. Sound is used in a lot of everyday objects, such as an oven, alarm clock or a teakettle. Even though the sounds of these objects might not seem like much, a lot of thought goes into designing such sounds. For example, an alarm clock has to be loud enough to wake someone up, but not too loud. Furthermore, the sound needs to be repeated until someone has turned off the alarm. As Norman [17] describes, the use of sound should be informative and contain information about its origin. Sound can let people know if errors or successes occur or whether something is still in progress. Also, sound can give information that cannot be seen. However, if it is not applied correctly it can be misleading and even annoying. To create somewhat intuitive sounds that in a way have the same purpose as icons, the use of Earcons can be applied. Earcons are the sound equivalent of icons [2]. An example of an earcon is the sound one hears when sending an email.

Furthermore, the use of sound has already proven to be beneficial when applied to programming. There are different ways in which it can be used, for example as a debugging tool for programming [20] or to represent data from graphs of tables. In contrast to using visual aids to show KGs, which sometimes can be misleading [21], sound can be used to find out where code gets stuck. It is easier to find a fault in code by using sound, since it will play the exact same loop of sound if the code gets stuck in a loop. Furthermore, sound has also been proven useful by using it in combination with a sorting algorithm [4]. With the different types of sorting algorithm it is then possible to listen to how the algorithm works. However, for the research of Brown & Hershberger [4] a visualization tool was used in combination with sound.

For this research we will use sound to represent the metrics of KGs. Similarly to the way in which using sound can add value when trying to debug code or representing sorting algorithms, the sound created for KGs needs to represent the metrics of KGs. To create such sound the principles of Earcons will be applied. This means that a sound will be created where one can instantly understand the metrics of KGs.

2.4 Combining Sound and KGs

To combine sound and KGs, the use of earcons as described in Section 2.3 could be effective. An earcon is an auditory equivalent of an icon, which means that like an icon it would be easy to recognize. This does not mean that it does not take a little time before one recognizes what they represent. However, when a person has learned the meaning of the sound they could be very easily recognized. There are some rules to creating these earcons. When these rules are followed it could make it easier to create such an earcon.

As described by Blattner et al. [2] the use of motives is important to create Earcons. A motive consists of a combination of at most four pitches in a certain rhythm. According to Kerman (1980) the most important part of a melody is the rhythm, because this gets more response from people than any other parameter. On the other hand, it can be discussed that the pitches are equally important. As described by Blattner et al. [2] the starting point for creating a motive consists of choosing a scale, octave and duration. To avoid any musical tension the pitches should chosen based on a major or minor scale. Also, the pitches of a motive should picked from the same octave. The maximum duration used for a motive is four notes, because then the listener can still remember the motive as a whole and not get stuck in a large sequence of notes where the end is not to be seen [7]. Additionally, there are three more parameters related to creating a motive, namely timbre, register and dynamics. The timbre can be defined as the warmth of the sound, this can be changed by using another instrument. The timbre of a piano is very different from that of a saxophone. This means that depending on the implementation of an earcon one could change the timbre to get a different perception of the earcon. The register of a motive is determined based on the pitch of the chosen octave. If the octave is high then the register is high as well. The dynamics of a motive are determined by the volume and softness, which can be varied or the same throughout the motive.

Based on the principles of creating a motive for an Earcon, we can conclude that the properties of sound that are relevant are:

- Pitch: scale (degree, major or minor)
- Register: octave (low, medium or high)
- Rhythm: combination of pitches (max. four)
- Dynamics: volume (crescendo, decrescendo)

These properties are needed to create a single motive. Based on these properties we need to find mappings that entail how the degree, type of scale (major or minor) and octave (high, medium, low) will be chosen. Once these are chosen we know which pitches can be used. Next, a rhythm needs to be created that consists of a maximum of four notes based on the pitches available in the scale and octave chosen previously.

2.4.1 Metrics of KGs

LOD Laundromat is a tool that crawls the Linked Open Data (LOD) cloud and convert the content into a set of triples. The tool than republishes the cleaned version of the data. The LOD Laundromat tool also calculates a lot of meta data for KGs, which we will call KG metrics. The SPARQL endpoint provided by LOD Laundromat allows for a user to find the values for the metrics of KGs indexed by the LOD Laundromat tool. The metrics provided by LOD Laundromat that can be used are shown in Table 1 [19]. Some metrics contain a minimum, maximum, mean, median and standard deviation values, whereas others only contain one value. These values are for example the indegree, outdegree and degree.

3 Method

The method to create an earcon from KGs consists of multiple parts, as is shown in Figure 1. First, the metrics of the KGs are gathered and created. These metrics are also stored to a

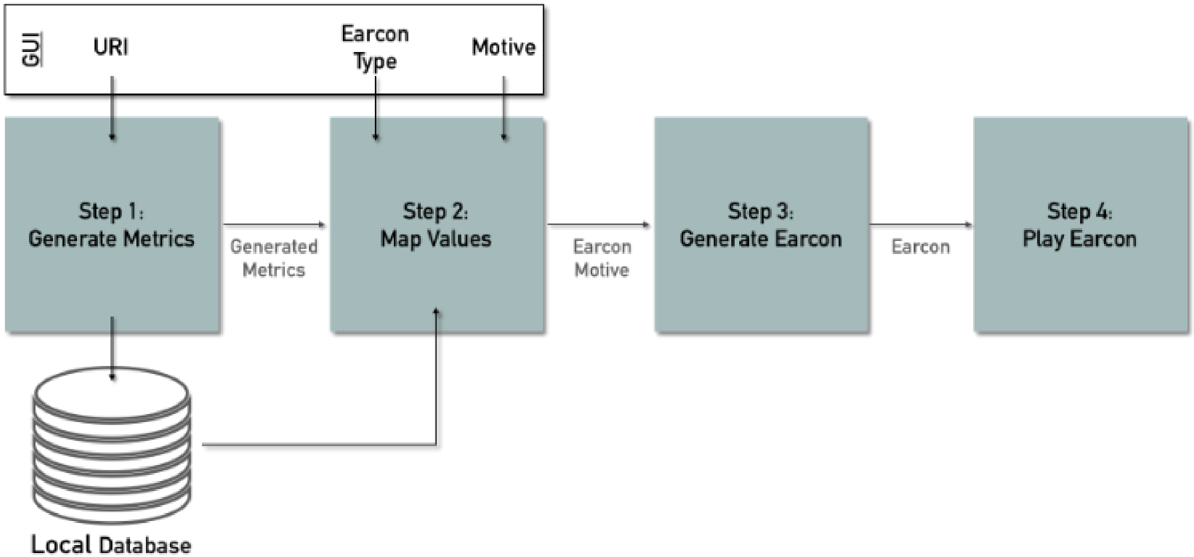

Figure 1. The workflow of creating an earcon consists of four steps. It starts by generating the metrics, then mapping the generated metric to the properties of sound, and then the mapping is used to create an earcon and then play it. The input for step 1 and 2 is provided through the use of a Graphical User Interface (GUI). The output of step 1 is stored to a local database. The data of the local database is then used in step 2.

Table 1. LOD Laundromat KG metrics

Blank nodes	Entities
Classes	In-degree
Data Types	IRIs
Defined Classes	IRI Length
Degree	Languages
Distinct Blank Nodes	Literals
Distinct IRI References	Literal Length
Distinct IRI Reference Objects	Object IRI Length
Distinct IRI Reference Subjects	Out-degree
Defined Properties	Predicted IRI Length
Distinct Literals	Properties
Distinct Objects	Subject IRI Length
Distinct Subjects	Triples

local database to ensure that the data is not lost. Next, these metrics are mapped to properties of sound, such as pitch and rhythm. The properties to which they need to be translated are based on the theory of making an earcon. Based on these mapped values, a MIDI file of the earcon is created and then played.

The sections below will first explain the use of the GUI. Next the steps of the workflow are explained, each in individual sections.

3.1 Graphical User Interface

For this application a GUI was added, which consists of a form. A screenshot of the GUI can be seen in Figure 2. The values that are provided in the form determine the earcon type and the values that the notes of the earcon will represent. The first field is a textfield in which the user can provide the URI for which they want to create an earcon. The second field consists of a dropdown field in which the types of earcons are listed. In this dropdown list the user can select the type of earcon they want to create. This allows for the user to choose the earcon type that they prefer. Lastly, the GUI contains three dropdown fields, which each contain all possible KG metrics. This is where the user selects the values that they want to hear in the earcon. The order in which the KG metrics are selected is also the order in which the notes will be played. Once the user has filled in all these fields they can click on the Generate button to start the process of generating the earcon.

The use of a GUI makes it possible to distribute the application and make its implementation versatile. It adapts to the users needs and therefore has multiple uses. It can, for example, be added to an application such as LOD Laundromat.

3.2 Generating Metrics

The metrics that we use are based on the list of metrics provided by LOD Laundromat as shown in Table 1. However, for this research we started with a smaller set of metrics than provided by LOD Laundromat, namely the number of nodes, the indegree and the outdegree. The smaller set of metrics was chosen so that the focus is on the mapping process and the creation of the earcons. To generate the metrics, the URI input of the GUI is used to generate an RDF Graph. The

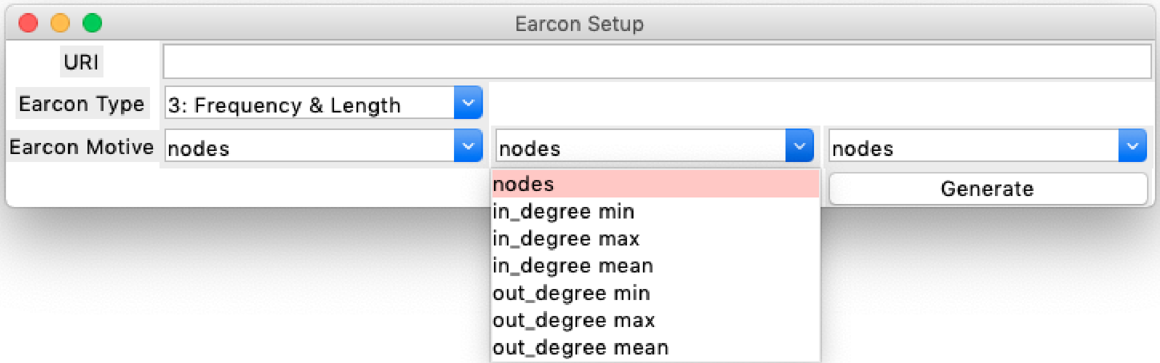

Figure 2. The GUI of the application, which consists of three field. A textfield in which the URI can be given, a dropdown list of the three earcon types, and three dropdown lists containing the KG metrics

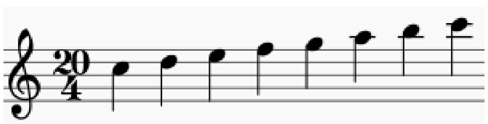

Figure 3. The pitches that are used to create an earcon

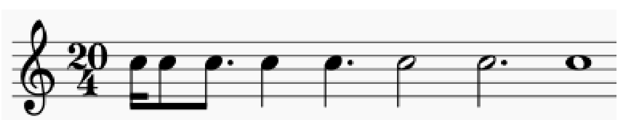

Figure 4. The different note length that are used to create an earcon

graph is generated using the RDFLib[1] package of python. This package provides functions that can get the subject-predicate-object relations, which we use to calculate the number of nodes and the in- and outdegree values. When all the metrics are gathered they are stored into a local database. The database is used in the next step where the metrics are being mapped to the properties of earcons.

3.3 Map Values

Once the metrics for the URI, provided by the user, are generated, the next step is to map these values to a property of sound. The metrics are numeric values. These need to be mapped to a note of a scale and the length of a note. In the next sections an explanation about the properties of sound will be given. Furthermore, the formula used to create these mappings is explained.

3.3.1 Properties of Sound

In Section 2.4 is described that the properties for sound are as follows:

- Pitch: scale (degree, major or minor)
- Register: octave (low, medium or high)
- Rhythm: combination of pitches (max. four)
- Dynamics: volume (crescendo, decrescendo)

As explained these properties are needed to create a single earcon motive. For this research we decided to create three types of earcons that take into account the rhythm and pitch. We decided to not include the dynamics, because the created earcon is a MIDI file. MIDI files do not pick up any dynamics. Furthermore, for the purpose of comparing different earcons we decided to always use the same scale and octave, namely a C major scale with a medium octave. This means that the earcon can contain the following notes: C D E F G A B C. For the length of the notes we decided to have the shortest note be sixteenth note and the longest a whole note. Combination of lengths in between a sixteenth and a whole note is also possible as long as the duration is not shorter than a sixteenth note and not longer than a whole note. The lengths of the notes as well as the pitches that are used are also shown in Figures 4 and 3.

3.3.2 Mapping Process

To create a mapping between the numeric values of the KG metrics and the properties of sound, a linear range conversion is used:

$$\begin{aligned} mappedvalue = ((currentmetric - metricmin)/ \\ (metricmax - metricmin) * (propertymax - propertymin)) \\ +propertymin \end{aligned} \tag{1}$$

[1] https://pypi.org/project/rdflib/

The theory is based on translating an arbitrary scale to another type of scale, e.g. a scale that differs between 1 and 5 to a scale that differs between 0 and 10. For this research we decided to compare a given metric against the same metric in a collection of graphs. Depending on the difference between the metric and the minimal and maximal metric values found in the collection of graphs a new value is generated. Next, this new value is translated to a property of sound, which can be either the pitch or rhythm of a note or both. The value is not directly translated to the property of sound, but to the index of the array that contains all pitches or rhythms. Based on this index we can then translate it back to a certain note or rhythm. The meaning of the values in the formula is as follows:

- Mapped Value: the resulting index value that can be translated to a pitch or rhythm.
- Current Metric: the metric of the current note.
- Metric Min: the minimal found metric in the collection of graphs.
- Metric Max: the maximal found metric in the collection of graphs.
- Property Min: the minimal index of the array of pitches or rhythms. This value is always 0.
- Property Max: the maximal index of the array of pitches or rhythms.

3.4 Generate Earcon

Based on the mapped values for each of the three notes of the motive, an earcon can be created. In this step the mapped values are combined to create an earcon that contains three notes and differs in either pitch, rhythm or both. This collection of three notes is then transformed to a MIDI output, which will then be played.

4 Implementation

The method as described above is implemented through the use of Python. The code uses several python packages:

- SQLite[2]: to create a local database that collects all graphs.
- RDFLib: to create an RDF Graph based on the URI provided by the user and to retrieve the data to calculate the metrics.
- Tkinter[3]: to create the GUI.
- RtMidi[4]: to create and play the MIDI file.

The implementation provides a GUI, which allows the user to decide for themselves which URI they want to use, which type of earcon they want to hear, and which values they want to hear. Once these values have been entered the code generates the metrics based on the URI the user has entered. Then, the generated metric values are stored to a database. The next step is to map these values. To do that the generated metrics are given as input. Also, the code retrieves all graphs and gets all same type of metrics as the current metric. This collection of metrics is used to compare the current metric to the collection and then translate the current value to an index that maps to an array of the property of sound. The last step is to use these mapped values and generate a MIDI output. The code is published on Github [5]. The database that contains the collected graphs and their generated metrics are published to FigShare [6].

5 Evaluation

The effectiveness of the created earcons are evaluated through a user study. The study consists of the different versions of generated earcons, which are evaluated through the use of a questionnaire. The questionnaire was distributed through social media channels such as Facebook and Whatsapp. In total the user study received thirty responses. In the next sections the questions used in the questionnaire are explained. Also, a plan to analyze these results is given.

5.1 Questionnaire

The first part of the questionnaire is used to determine whether the respondents have knowledge about KGs. If answered yes, the respondents are asked to specify if they use KGs in their everyday life and how often. The questions corresponding to this section to determine the respondents knowledge about KGs are shown in Appendix A.1. However, if the respondent answers no, this section is skipped and the questionnaire will be redirected to the next part in which the earcons are tested.

The part where the earcons are tested is divided into multiple sections. Each type of earcon, pitch, length, and pitch and length, is tested in a different section. Each section contains two earcons that each have the same metrics and are the same earcon type per section. To determine if the type of earcon has an effect on the recognition of the KG metrics, each of the before mentioned sections contain the same type of questions. The first type of question determines if the respondent can determine the differences of metrics in a single earcon. There are two of these questions that each is about another earcon. The second type of question lets the respondent compare two different earcons. The types of questions that were asked can be found in Appendix A.2.

After the respondent has answered all questions regarding the effectiveness of each type of earcon. They are asked to choose their own earcon type preference. This is a multiple choice question, where only one answer can be chosen.

[2] https://docs.python.org/2/library/sqlite3.html
[3] https://docs.python.org/2/library/tkinter.html
[4] https://pypi.org/project/python-rtmidi/

[5] https://github.com/EnyaNieland/Earcons
[6] https://doi.org/10.6084/m9.figshare.8847719.v1

5.2 Analysis

To analyze the results, each question of the earcon part of the questionnaire is translated to a value between 1 and 0. Where 1 means that the question is answered completely correct and 0 means that the answer is incorrect. If a question can have multiple answers than the value can be anywhere between 0 and 1. Based on these values several graphs were made. The found results are explained in the next section.

6 Results

The results found based on the questionnaire are shown in Table 2. The following sections analyze these results per earcon type with the help of diagrams. Starting with the results of the earcons that differ in pitch, next the earcons that differ in length and lastly the earcons that differ in pitch as well as length. After these three section there is an additional section that compares the different types of earcons.

6.1 Earcon Type 1: Pitch

When we take a look at the overall results as shown in the first graph in Figure 5 we can see that overall, more questions are answered correct than in correct. For the two questions where the values of a single earcon are compared the results are just above 50%, which means that just a little bit more than half of the respondents were able to answer these questions correct. However, the respondents score higher when having to compare two earcons. The score is highest for the last question where they had to compare each value of the earcons.

In the second and third graph of Figure 5 we divided the results into a group that does have knowledge about KGs and a group that does not have any knowledge about KGs. For this two graphs we can see that for the first individual question, the respondents that do have knowledge score very low. Only a little more than 25% are able to answer this question correct. On the other hand, this percentage is a lot higher for the group that do not have any previous knowledge. For the second individual chapter both groups perform equally, just above 50%. This score gets better for the group that do not have any knowledge about KGs for the third question, the first question that asks to compare two earcons. However, the group that does have knowledge again scores just above 50%. The last question is answered correct the most amount of times by both groups. The last question is the overall best answered for the earcon that differs only in pitch.

6.2 Earcon Type 2: Length

In the first graph in Figure 6 we can see that for most questions the amount of correct answers is close to 75%, which means more people answered correct than incorrect. However, for the third question, which is the first question that asks respondents to compare two earcons, the amount of correct answers is the same as the amount of incorrect answers. For the first two questions this earcon type scores higher than the first type. However, the score is lower for the two question of earcon type 2 that compare two earcons compared to earcon type 1.

When we then compare the second and third graph in Figure 6 we can see that for the first and last question the percentage of correctly answered questions is approximately the same for respondents that do and do not have previous knowledge about KGs. Furthermore, the percentage of correct answers for question 2 is higher for people that do not have previous knowledge. Furthermore, we can see that people that do have knowledge score less than 50% correct for the third question.

6.3 Earcon Type 3: Pitch & Length

For the type of earcon that differs in both pitch and length it can be seen in Figure 7 that the percentage of people that answered question 1 and 2 correct, which is about comparing metrics of a single earcon, is 50% or lower. When comparing the responses for people with or without knowledge of KGs, it can also be seen that the percentage of incorrect answers is relatively high. However, this percentage is lowest for people that do have previous knowledge. When we then take a look at the questions where two earcons are compared, we can see that these questions are answered above average in each diagram. This percentage is highest for the group of people that do have previous knowledge. However, the percentage of correct answers is higher when two different earcons need to be compared. It is even highest for people with previous knowledge.

6.4 Comparing Earcon Types

To compare the different types of earcons we combined the individual questions and the compare questions for each earcon type as can be seen in Figure 8. For the combined individual questions we can see that the first and second earcon type are above 50%, whereas the third earcon type is the only one that scores below 50%. For these questions the second earcon type performed best. For the compare questions all earcon types perform above average, but the first type performs the highest of all. Overall we can conclude that earcon type 3 worked the least well. However, this is the earcon that was preferred by most respondents. The first earcon type was preferred by the least amount of respondents.

7 Conclusion

Based on the results we conclude that the earcon type 3 is the least effective, because it contains the least amount of correctly answered questions. However, this earcon type is chosen as the most preferred. Furthermore, it can be concluded

Table 2. The total amount of correct and incorrect answers divided per question and per earcon type

	Earcon Type 1: Pitch		**Earcon Type 2: Length**		**Earcon Type 3: Pitch & Length**	
	Correct	**Incorrect**	**Correct**	**Incorrect**	**Correct**	**Incorrect**
Question 1	17	13	21	9	15	15
Question 2	17	13	22	8	10	20
Question 3	21	9	15	15	20	10
Question 4	26	4	21	9	19	11
Total	81	39	79	41	64	56

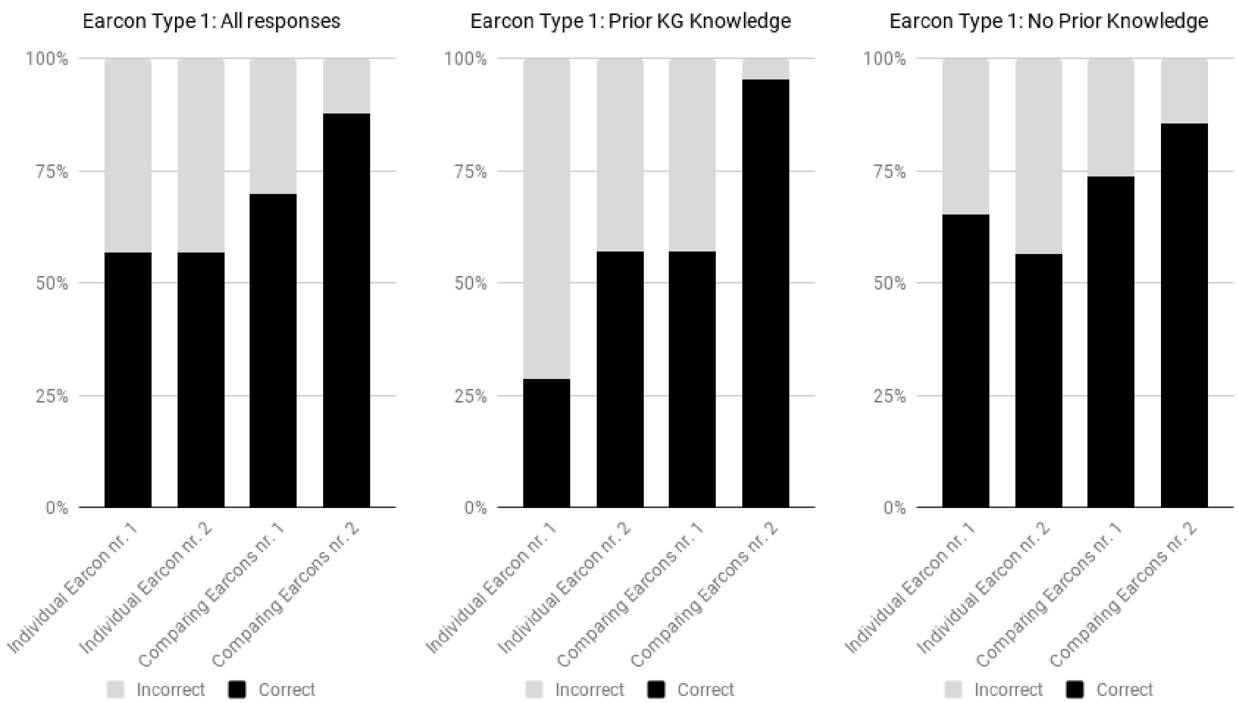

Figure 5. Three graphs showing the percentage of correctly answered questions regarding earcons that differ in pitch. The first two questions regard individual earcons, whereas the last two questions regard the comparison of two earcons. The first graph contains all responses, whereas the second and third graph are split into responses of people that said to have knowledge about KGs or no prior knowledge about KGs.

that both earcon type 1 and 2 are approximately equally effective. The scores of the combined respondent groups range between approximately 50% and 90%. Furthermore, the overall amount of correctly answered questions is higher than the amount of incorrectly answered questions. The results also show that respondents that said that they have knowledge about KGs are more prone to answer incorrect. This could be because of the low amount of respondents that have any knowledge about KGs, which is only 25% of the total amount of respondents. Since each value is mapped the same way and the questionnaire did not show any difference between the different metrics, it can be concluded that the type of mapping does not matter.

Based on these conclusion we can answer the research questions.

RQ1: How can the use of sound create an insight into the metrics of KGs?

We can conclude that using earcons provides an insight into the metrics of KGs, because we found that, overall, the user study contains a larger number of correctly answered questions than incorrect. This can tell us that overall most people can recognize the values that the earcons represent. We can even conclude that using only pitch or length seems to be most effective to provide an insight into the metric of KGs. Unfortunately, the combination of both is less effective.

RQ2: What are the possible mappings between the properties of sound and KGs?

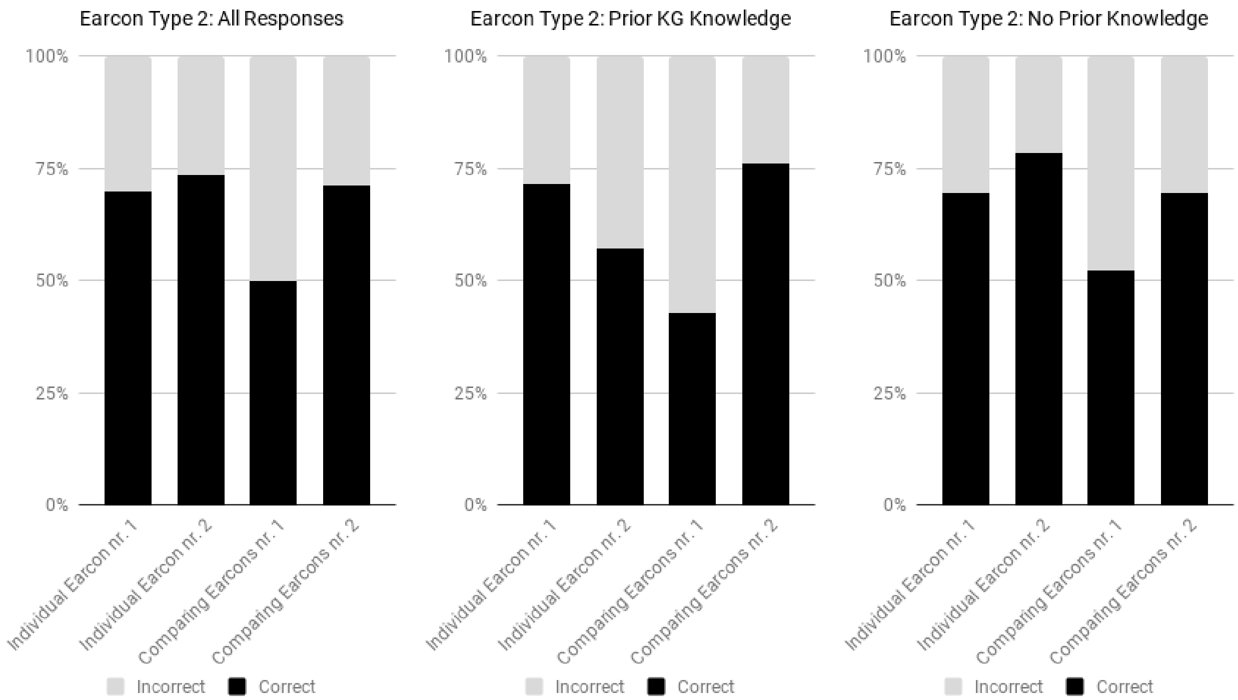

Figure 6. Three graphs showing the percentage of correctly answered questions regarding earcons that differ in rhythm. The first two questions regard individual earcons, whereas the last two questions regard the comparison of two earcons. The first graph contains all responses, whereas the second and third graph are split into responses of people that said to have knowledge about KGs or no prior knowledge about KGs.

We found that the metrics of KGs that could be mapped are the number of nodes, the indegree, or the outdegree. For the indegree the minimal, maximal and mean value could be used for the mapping process. The properties of sound that we were able to map each of the previously mentioned metrics to are pitches of a C major scale and note lengths that differ between a sixteenth and a whole note. However, we found that mapping the metrics to both the pitch and length, was the least effective mapping. Furhtermore, it could be discussed that each of the numeric values that LOD Laundromat provides could work equally well, as the mappings work with each of the generated metrics of this research.

RQ3: How can a tool use these mappings to create an insight into the metrics of KGs?

Using this tool could be an addition to a tool such as LOD Laundromat to provide an instant insight into the complexity of KGs. It could additionally benefit people who are visually impaired and want to get an insight into the complexity of KGs.

8 Discussion

This research has shown that the use of earcons can provide an insight into the metrics of KGs. However, there is still a lot more to research on this topic. For example, implementing the earcons in a different way. For this research we decided to let each note of the earcon represent a comparison of a certain metric to the same type of metric in a collection of graphs. Furthermore, the metrics of the earcon represent the entire graph. Further research could be focused on creating earcons that represent metrics of a single node of a KG. This could be used to getting an insight into a single node and could therefore also take into account the type of relationship that is represented. Another idea could be to find out if the metrics for the earcon can also be only compared to the metrics inside of that graph instead of a whole collection of graphs. These are ideas that change the way the metrics of KGs are implemented. However, we could also look at the properties of earcons that are used. For this research we set the scale degree and type to be a medium C major scale to make it easier to compare the different earcons. However, these variables could also be made variable to create a large range. This larger range could make the difference between metrics with a large range more obvious. For this research the notes of only a single octave could be picked, which is not a very large range. If the metrics of for example the number of nodes can differ between 0 an 1000 and needs to be transformed to a scale of eight notes, the variation is not very large. It would not be clear if the value would differ

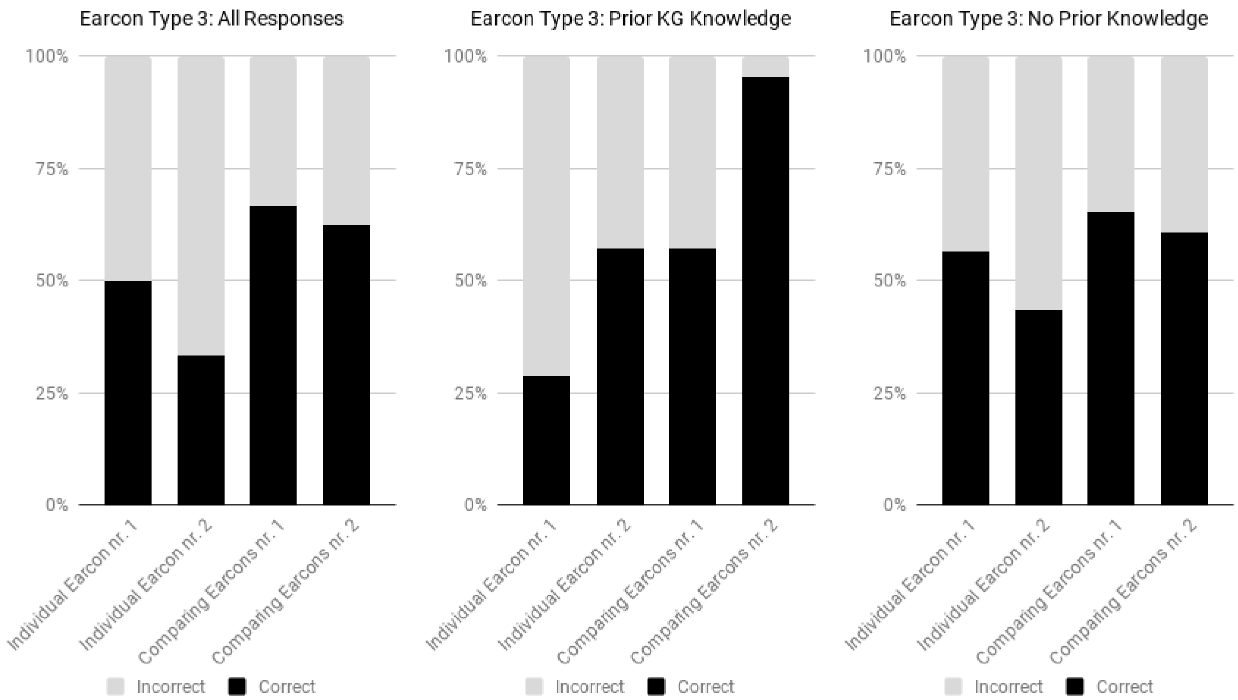

Figure 7. Three graphs showing the percentage of correctly answered questions regarding earcons that differ in pitch and rhythm. The first two questions regard individual earcons, whereas the last two questions regard the comparison of two earcons. The first graph contains all responses, whereas the second and third graph are split into responses of people that said to have knowledge about KGs or no prior knowledge about KGs.

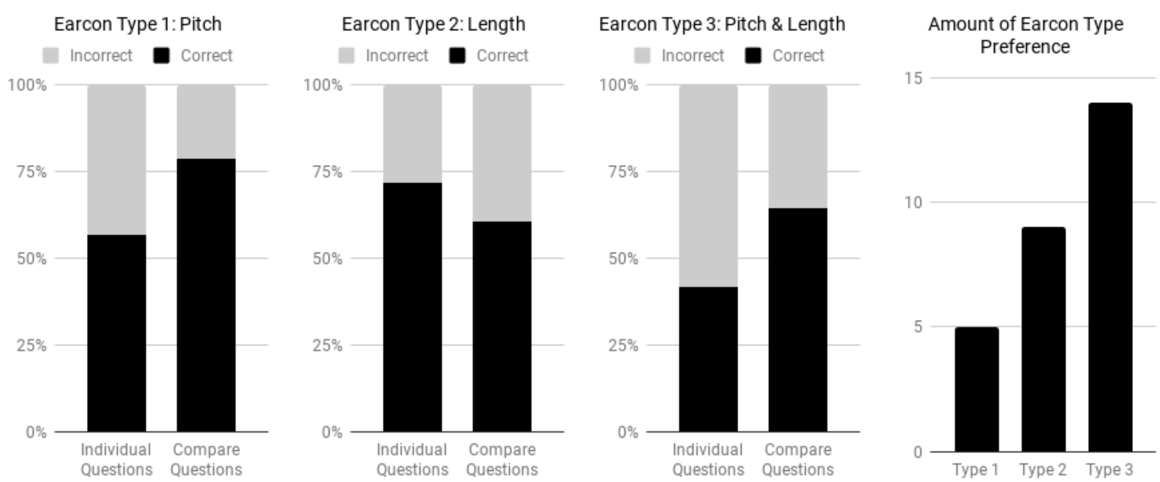

Figure 8. Three column charts that show the amount of correctly answered individual and compared questions. The last chart shows the preferences of the respondents

somewhere between 0 and 100, because all values between 0 and 100 would be translated to the same pitch. Another element could be to implement dynamics, which for now have not been implemented due to the use of a MIDI output.

This implementation can be used for more types of graphs than KGs. In the implementation created in this research the metrics that are used as input consist of numeric values. Therefore, the implementation of other types of network could also be used as long as the values that are used as input are numeric values. This means that the current research is generally applicable to multiple types of graphs. Furthermore, it could also be discussed whether a combination with a visual representation could be beneficial. For example, if the earcon would represent the metrics of a single node of the graph. Then the visual representation could be used to show the node which is used for the earcon.

References

[1] Wouter Beek, Laurens Rietveld, Hamid R. Bazoobandi, Jan Wielemaker, and Stefan Schlobach. 2014. LOD laundromat: A uniform way of publishing other people's dirty data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8796 (2014), 213–228. https://doi.org/10.1007/978-3-319-11964-9

[2] Meera M. Blattner, Denise A. Sumikawa, and Robert M. Greenberg. 1989. Earcons and Icons: Their Structure and Common Design Principles. (1989), 11–44 pages. https://doi.org/10.1207/s15327051hci0401{_}1

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase : A Collaboratively Created Graph Database For Structuring Human Knowledge. *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data (2008)* (2008), 1247–1249.

[4] Marc H. Brown and John Hershberger. 1992. Color and Sound in Algorithm Animation. *Computer* 25, 12 (1992), 52–63. https://doi.org/10.1109/2.179117

[5] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. *CEUR Workshop Proceedings* 1695 (2016).

[6] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y. K. Ng, and R. D. Smith. 1999. Conceptual-model-based data extraction from multiple-record Web pages. *Data and Knowledge Engineering* 31, 3 (1999), 227–251. https://doi.org/10.1016/S0169-023X(99)00027-0

[7] Herman Feshbach, Arthur Kerman, and Steven Koonin. 1980. The statistical theory of multi-step compound and direct reactions. *Annals of Physics* 125, 2 (1980), 429–476. https://doi.org/10.1016/0003-4916(80)90140-2

[8] Dan Gärdenfors. 2001. Auditory Interfaces: A Design Platform. (2001), 17. http://jld.se/dsounds/auditoryinterfaces.pdf

[9] William W. Gaver. 2015. Auditory interfaces. *SpringerBriefs in Computer Science* 9783319221106 (2015), 33–44. https://doi.org/10.1007/978-3-319-22111-3{_}3

[10] Jennifer Golbeck and Matthew Rothstein. 2008. Linking social networks on the web with foaf: A semantic web case study. *Aaai* (2008), 1138–1143. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Linking+social+networks+on+the+web+with+foaf:+A+semantic+web+case+study#0

[11] Christian Hirsch, John Hosking, and John Grundy. 2009. Interactive visualization tools for exploring the semantic graph of large knowledge spaces. *CEUR Workshop Proceedings* 443, May (2009).

[12] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *IJCAI International Joint Conference on Artificial Intelligence* 194 (2013), 3161–3165. https://doi.org/10.1016/j.artint.2012.06.001

[13] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134

[14] Simon Lehmann, Ulrich Schwanecke, and Ralf Dörner. 2010. Interactive visualization for opportunistic exploration of large document collections. *Information Systems* 35, 2 (2010), 260–269. https://doi.org/10.1016/j.is.2009.10.004

[15] Pablo Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Mendes, P., Jakob, M., García-Silva, A., & Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. Retrieved from http://www.citeulike.org/user/lsamper/article/9893296DBpedia Spotlight: Shedding Light on the Web of Documents. (2011). http://www.citeulike.org/user/lsamper/article/9893296

[16] G Miller, R Beckwith, C Fellbaum, D Gross, and K Miller. 1993. Introduction to WordNet: On-line Database. *International journal of lexicography* 3, 4 (1993), 235–244.

[17] Don Norman. 2014. *The design of everyday things*. Vol. 51. 51–5559 pages. https://doi.org/10.5860/choice.51-5559

[18] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. LNCS 8218 - Knowledge Graph Identification. (2013), 1–16.

[19] Laurens Rietveld, Wouter Beek, Rinke Hoekstra, and Stefan Schlobach. 2017. Meta-data for a lot of LOD. *Semantic Web* 8, 6 (2017), 1067–1080. https://doi.org/10.3233/SW-170256

[20] Robert Rubin, Larry Rudolph, and Dror Zernik. 2005. Debugging parallel programs in parallel. *ACM SIGPLAN Notices* 24, 1 (2005), 216–225. https://doi.org/10.1145/69215.69236

[21] Paul Vickers and James L Alty. 2003. Siren Songs and Swan Songs. *Commun. ACM* 46, 7 (2003), 86–92. http://proxy.libraries.smu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=11863464&site=ehost-live&scope=site

A Questionnaire Questions

A.1 Knowledge about KGs

1. Are you familiar with the term "Knowledge Graph"?
2. Have you worked with Knowledge Graphs or other types of Networks in the past?
3. Do you use Knowledge Graphs or other type of Networks in your work?
4. On a scale of 1 to 10 how often do/did you make use of Knowledge Graphs or other types of Networks?

A.2 Effectiveness of Earcons

1. When comparing the lowest and highest indegree of Earcon #1, which value is higher?
2. When comparing the average and highest indegree of Earcon #2, which value is lower?
3. Which Earcon has a higher lowest indegree?
4. Which indegree values differ between Earcon #1 and Earcon #2?