Sound of Databases: Sonification of a Semantic Web Database Engine

Sven Groppe
Institute of Information Systems (IFIS), University of Lübeck
Lübeck, Germany
groppe@ifis.uni-luebeck.de

Rico Klinckenberg
Institute of Information Systems (IFIS), University of Lübeck
Lübeck, Germany
rico.klinckenberg@gmx.net

Benjamin Warnke
Institute of Information Systems (IFIS), University of Lübeck
Lübeck, Germany
benjamin.warnke@uni-luebeck.de

ABSTRACT

Sonifications map data to auditory dimensions and offer a new audible experience to their listeners. We propose a sonification of query processing paired with a corresponding visualization both integrated in a web application. In this demonstration we show that the sonification of different types of relational operators generates different sound patterns, which can be recognized and identified by listeners increasing their understanding of the operators' functionality and supports easy remembering of requirements like merge joins work on sorted input. Furthermore, new ways of analyzing query processing are possible with the sonification approach.

PVLDB Reference Format:
Sven Groppe, Rico Klinckenberg, and Benjamin Warnke. Sound of Databases: Sonification of a Semantic Web Database Engine. PVLDB, 14(12): 2695 - 2698, 2021.
doi:10.14778/3476311.3476322

1 INTRODUCTION

Sonification is typically defined *as the use of nonspeech audio to convey information [...] for the purposes of facilitating communication or interpretation* [6].

Many published contributions to sonification are a hint for the success of sonification approaches [2], but there is still no widespread adoption of sonification due to many open challenges [7].

In a nutshell, the benefits of sonification approaches are typically the following ones: Sonification can help not only humans with visual impairments, but also help to reduce barriers for people with poor education. Audio can present multidimensional data in an easy-to-understand way, so that the most important aspects of the data and its processing can also be understood by people without science education. An additional medium also helps with learning new information. [9]

We claim that these general benefits are of great value to the database community, too, increasing the attractiveness of and interest in database courses, technologies and scientific results in the area of databases, whenever there are sonification approaches for database technologies available. In this demonstration, we propose a sonification approach of processing queries in a Semantic Web database. To the best of our knowledge, this is the first approach to sonification of query processing in the literature.

Besides the above described general benefits of sonification, scientists and developers of database engines may detect anomalies by listening to the sonification of their database engine, which are hints for errors, and repeating sound patterns may be a hint for repeating (at least similar) calculations, which are to be optimized for increasing the performance. Hence sonification enables new ways of analyzing query processing for the purpose of debugging, performance tuning and research.

Our main contributions are:

- Sonification of query processing in a Semantic Web database,
- a flexible approach to map the data processed according to the query execution plan to sound effects, and
- an easy-to-use web application[1] for visualizing the processing of Semantic Web queries along with its sonification.

2 TARGET AUDIENCE

With our sonification demonstration of a database, we address a large spectrum of target audiences and discuss the benefits of our sonification for these audiences in the following paragraphs.

General Public: Sonification is a means to offer a multimedia show to the general public at events attracting more people. By offering sonification applications to be used by any persons, sonification of databases may help scientists to increase the interest in database technologies in a playful way. It may help for an easy understanding of database technologies offering an audio-visual presentation of the data processing.

Students: Computer science students may more easily learn and don't forget the learned facts about database algorithms especially if these can be recognized in the sonification. For example, the sonification of merge joins may result in an ascending scale remembering the student that the input of merge joins must be sorted.

Our purpose is not only to reach students of computer science or related subjects but also to reach students of foreign subjects to give them an access to the basics of databases. So our main focus are associations like: What happens with my query request? How is it processed? How do I request special data? Sonification shall help them to recognize regularities and differences of varying optimizations of requests audio-visually.

To make computer science and its mostly theoretical topics more interesting to students and pupils, multimedia applications are

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476322

[1] Our demonstration with example queries, data and sonification mappings for all described use cases and target groups is available at https://www.ifis.uni-luebeck.de/~groppe/soundofdatabases/.

beneficial. In cases that school students can experiment and create unique and interesting solutions on their own, it drums up interest for usually monotonous topics. Even in computer science there should be possibilities to arrange those topics in an exciting way. With the support of visual and auditive attractions pupils should playfully and perhaps musically learn the basics to work with databases. The main focus is learning with fun, to discover the features of such applications on their own to get access to databases and computer science topics.

Teaching Staff: Our demo is beneficial to teaching staff no matter if they are teaching at universities, schools or provide seminars for employees. Our solution provides an easy access to a complex topic to any level of students and pupils. Therefore different adaptations will be needed depending on the teaching method: presentation of results or experimenting. So in the first case the teacher will be able to record single query requests or create problem depending sound mappings in advance.

Developers of Database Engines: Irregular patterns and anomalies in the sound of the sonification are a hint for errors in the code, such that a sonification of a database engine helps developers to identify these errors. Furthermore, patterns in the sound may help to analyze performance issues and may provide hints which code is worth to be and how this code should be optimized. Especially repeating sound patterns may be a hint for repeating calculations, which might be avoided improving the performance of the overall calculation. Scientists may also get ideas for efficient super operators replacing several simpler operators.

Visually Impaired: Experiments show that visually impaired can recognize the location of nodes in graphical data when using a proper mapping from the 3D location to auditory dimensions for sonification [1]. The graphical data are the operator trees of query execution plans for our proposed sonification. Hence our proposed sonification helps the visually impaired to understand the database technologies, especially query processing and its algorithms, and to make them audibly tangible for these technologies.

Musicians/Artists: All interested persons including musicians and artists are invited to experiment with a new tool to create unique sounds. We offer the option to use different instruments and sounds at the same time stimulating to assemble new rhythms and melodies.

3 THE SEMANTIC WEB DATABASE ENGINE LUPOSDATE3000

The Semantic Web database engine LUPOSDATE3000[2] [10] is the successor of LUPOSDATE[3] [3]. The focus of LUPOSDATE3000 is a modern architecture using latest database technologies (see Section 3.1) while going to support multiple platforms. We introduce the vision of a hybrid multi-model multi-platform (HM3P) database in [5], which spans over different platforms in operation and supports different data models in one single database, and its semantic variant - SHM3P database - in [4]. In future work we plan to develop LUPOSDATE3000 further to a SHM3P database. Currently, LUPOSDATE3000 is a semantic multi-platform database management system.

[2]Source code of LUPOSDATE3000: https://github.com/luposdate3000/luposdate3000
[3]Source code of LUPOSDATE: https://github.com/luposdate/luposdate

3.1 Internals and Sonification Web App

We develop LUPOSDATE3000 in the Kotlin[4] programming language in order to support different targets like the JVM, JavaScript and in the near future[5] native binaries for desktop, server, web and mobile environments. In this way it is possible to develop a common code basis for efficient database servers, distributed databases for cloud and IoT, mobile databases for operating on phones and tablets, and web applications for demonstrating and teaching purposes running completely in the browser.

LUPOSDATE3000 is optimized for modern multi-core cpus by introducing an RDF-3X variant enabling a partitioned input for efficient parallel processing [10]. Our next steps in future work include the full-fledged support of Internet-of-Things (IoT) environments.

LUPOSDATE3000 uses a dictionary to map all values to integer ids. Dictionaries are decreasing the memory footprint during query processing as well as reducing the space used for indices. For the sonification, the integer ids can be directly used for manipulating auditory dimensions like the pitch or duration of the sound.

These ids are then stored in an index similar to RDF-3X [8] using all 6 collation orders for maximizing the use of the fast merge joins on pre-sorted data retrieved from the index. The sonification of processing pre-sorted data generates regular sound patterns with a distinctive sound experience. Operators, which do not generate their output in a sorted way any more (like duplicate elimination using hash tables), also do not generate an ascending scale, which is immediately recognized by users of the sonification. Hence each operator generates a unique sound pattern during query processing resulting in a complex sound ensemble for complex queries.

During query evaluation, LUPOSDATE3000 uses both column and row iterators, preferring the column iterators where applicable. The sonification of a single column value generates a single tone, whereas the generation of a complete row can include the sonification of multiple column values. Hence the difference between the usage of column and row iterators would be audible, too, but the sonification in LUPOSDATE3000 currently only supports the sonification of a single column by implementing column iterators.

The *sonification software* offers a single page application in the web, which provides access to a LUPOSDATE3000 database server, but also fully integrates the LUPOSDATE3000 web version, such that the sonification web application can run stand-alone in the browser. We use Tone.js[6] as simple-to-use library for generating tones with the add-on tonejs-instruments.js[7] for using instruments, and vis.js[8] and vis-network.js[9] for visualizing operator trees and animating query processing.

4 SONIFICATION OF QUERY PROCESSING

Our proposed sonification of query processing generates a tone for each exchange of intermediate solutions[10] between the operators in the operator tree of the query execution plan (see Figure 1).

[4]https://kotlinlang.org/
[5]After the introduction of a new memory manager including a new garbage collector for Kotlin/Native, such that current restrictions are lifted (see https://youtrack.jetbrains.com/issue/KT-42296).
[6]https://tonejs.github.io/
[7]https://github.com/nbrosowsky/tonejs-instruments
[8]https://visjs.org/
[9]https://github.com/visjs/vis-network
[10]To be more precise, for each of its column values.

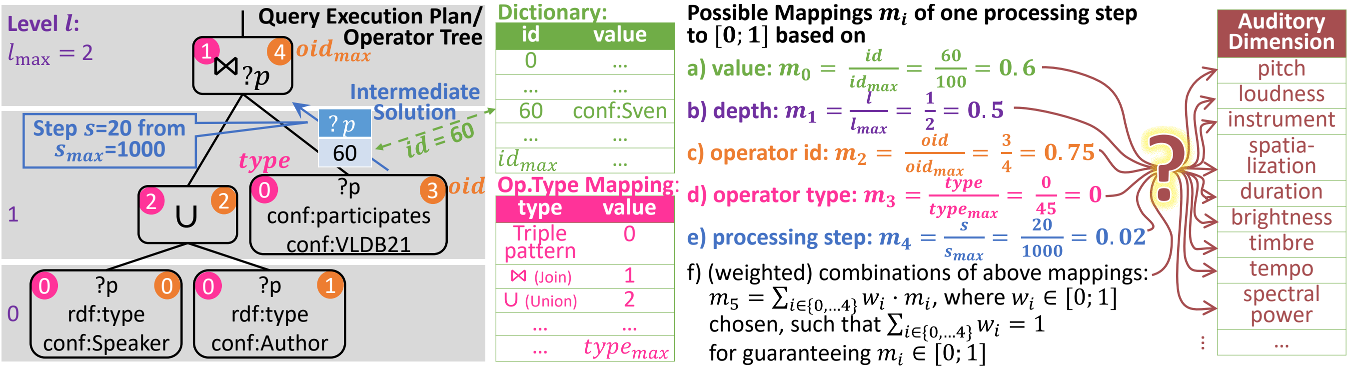

Figure 1: Query execution plan of a simple query with an intermediate solution between two operators along with possible mappings of this processing step

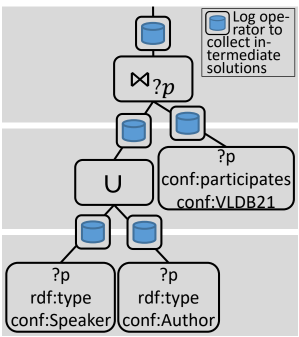

Figure 2: Operator tree to collect intermediate solutions of each query processing step

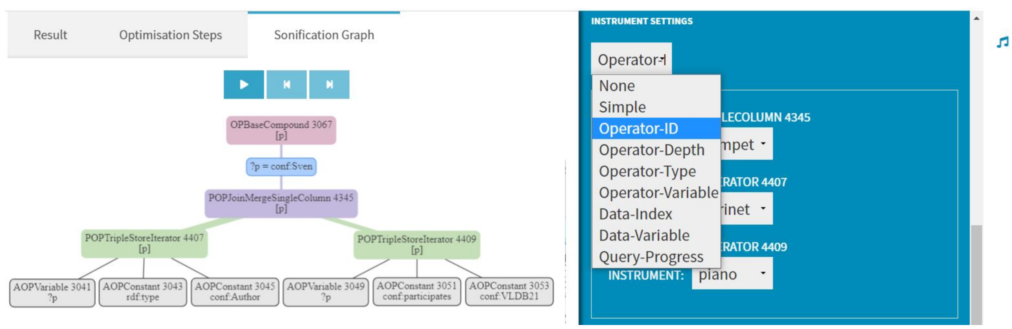

Figure 3: Sonification web app with flexible configuration possibilities: The auditory dimensions pitch, instrument, loudness, spatialization (left to right speaker), duration, melody (e.g., dur cadence and hit formula), chord and octave can be configured in a flexible way to be mapped from operator id, type, depth (in operator tree), variable ("Data-Variable"), variable combination ("Operator-Variable"), bound value ("Data-Index") and query progress. The merge join operator just determined the intermediate solution "?p = conf:Sven".

The sequence of intermediate solutions determines the sequence of generated tones. For the purpose of collecting when which intermediate solutions are exchanged between which operators, we introduce a new kind of operator, which we call *log operator*. We insert log operators between each pair of operators connected by an edge in the operator tree (see Figure 2). In this way we log each step of processing queries in LUPOSDATE3000 for sonification and also for debugging purposes. Please note that during normal operation, the log operators are left out for maximal performance.

Most sonification approaches use the following auditory dimensions (out of 30) [2]: pitch, loudness, spatialization, duration, brightness, timbre, tempo and spectral power. These auditory dimensions are very useful for a mapping of query processing, too. Hence we need quantitative numbers for the mapping of query processing to auditory dimensions during sonification. We first normalize these quantitative numbers to the [0; 1] domain and afterwards scale it to rational minimum and maximum values of the auditory dimension to be mapped to.

During query processing, the most relevant information are the sequence of intermediate solutions consisting of bindings of values to variables, which are the output of certain relational operators serving as input of the succeeding operator in the query execution plan (see Figure 1). Because LUPOSDATE3000 uses a dictionary, the values of the intermediate solution are already represented as integer numbers for which a maximum value can be determined. In this way we have a scale for the mapping to the quantitative auditory dimensions.

Overall, when considering query execution plans and query processing, the following information provides quantitative numbers during query processing (see Figure 1):

- The integer id representations of the values in intermediate solutions,
- the ids of the operators, their depths in the operator tree of the query execution plan or their types having the intermediate solution as input or output, and
- the temporal aspect, i.e., at what time relative to the whole query processing the intermediate solution has been generated.

While the mapping to the auditory dimensions can be configured in a flexible way, according to our experiences the most useful ones seem to be:

- for understanding the operator algorithm (or at least for recognizing characteristics of its in- and output): a mapping from the intermediate solution values to the pitch and the operators - between which the intermediate solution is exchanged - determines the used instrument, such that operators can be easily distinguished from the other (see Figure 3), and
- for generating inspiring melodies and sounds (not only for musicians/artists): a mapping from the operator depth to pitch and different operators may use different instruments.

Dependent on the query and the data, other mappings may generate convincing sound effects, too, such that users are encouraged to extensively experiment with the mapping configuration.

5 DEMONSTRATION

We describe the different used queries for our sonification demonstration[1]. The used queries are presented with an increasing complexity and we motivate their use in the demonstration by discussing their purposes like learning effects of students.

5.1 Sonification for Guessing Single Operators

Experienced database developers and scientists may be asked for their guess what kind of relational operator is introduced in the sonification demonstration for the following queries. Novices to database technologies may be asked about their guess of the functionality of the considered query.

One Triple Pattern: The first query for the sonification demonstration is a simple query consisting only of one triple pattern. The listeners of the sonification hear an ascending scale getting to know that the solutions of triple patterns can be determined by accessing an index of pre-sorted data.

Filter: The second query adds a filter. Listeners of the sonification recognize that not all intermediate results are passing the filter operator playing the previous tone generated by the index scan with another instrument.

Merge Join: This query consists of a binary operator having two inputs and one output. Two index scan operators as well as the output of the merge join play with different instruments. Listeners notice that only if the two inputs of the index scan operators are the same, an output is generated.

Optional-Clause/Left Outer Join: This query contains an optional-clause, such that a left outer join occurs in the operator tree of the query execution plan. In comparison to the merge join, the input of the left index scan passes always the left outer join - even if it is not combined with the input of the right index scan.

Hash Join: This query consists of a merge join of two triple patterns and a succeeding hash join combining the result of the previous merge join with a triple pattern. The hash join is implemented as pipeline breaker by generating the output after reading the complete input of one operator.

5.2 Sonification of Complex Queries

In this section we describe the sonification of more complex queries allowing further analysis and generating a more complex ensemble of sound effects.

Recognizing Single Operators in Complex Queries: Listeners of sonifications of very complex queries may recognize single operators in the complex ensemble of different operators and their generated sounds during query processing.

Common Subexpressions: Queries consisting of the same subexpressions several times generate a repeating sound during sonification, such that listeners may notice these common subexpressions.

Similar Queries: The sonification of similar queries generates similar sound with some few differences (like passing other solutions in different filters), which might be recognized by the listeners.

Queries generating inspiring Sounds: Some sonifications of complex queries may generate interesting regular sound patterns, which may inspire musicians and artists for new musical compositions.

6 SUMMARY AND CONCLUSIONS

In this demonstration we deal with the sonification of query processing. We showcase the sonification of numerous queries by proposing a sophisticated web application[1] supporting a simple way of a complex configuration of the mapping from query processing steps to auditory dimensions. With the help of the sonification, we offer a unique audible experience and new way of experiencing query processing. The listeners of our sonification are encouraged to guess the functionality of the different operators in the queries and determine the types of processed relational operators. Furthermore, we show that auditive analysis are possible by detecting common subexpressions and similar queries.

ACKNOWLEDGMENTS

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 422053062.

REFERENCES

[1] Louis Commère, Grond Florian, François Côté, and Jean Rouat. 2018. Sonification of 3D point clouds for substitution of vision by audition for blind users. In *24ème conference internationale Auditory Display (ICAD 2018).*

[2] Gaël Dubus and Roberto Bresin. 2013. A systematic review of mapping strategies for the sonification of physical quantities. *PloS one* 8, 12 (2013), e82491.

[3] Sven Groppe. 2011. *Data Management and Query Processing in Semantic Web Databases.* Springer. https://doi.org/10.1007/978-3-642-19357-6

[4] Sven Groppe. 2021. Semantic Hybrid Multi-Model Multi-Platform (SHM3P) Databases. In *Proceedings of the International Semantic Intelligence Conference (ISIC), New Delhi, India.* http://ceur-ws.org/Vol-2786/Paper2.pdf

[5] Sven Groppe and Jinghua Groppe. 2020. Hybrid Multi-Model Multi-Platform (HM3P) Databases. In *Proceedings of the 9th International Conference on Data Science, Technology and Applications (DATA), virtual due to Covid-19 pandemic.* https://doi.org/10.5220/0009802401770184

[6] Gregory Kramer, BN Walker, Terri Bonebright, Perry Cook, J Flowers, Nadine Miner, John Neuhoff, R Bargar, S Barrass, J Berger, et al. 1999. The sonification report: Status of the field and research agenda. report prepared for the national science foundation by members of the international community for auditory display. *International Community for Auditory Display (ICAD), Santa Fe, NM* (1999).

[7] J. G. Neuhoff. 2019. Is Sonification Doomed to Fail. In *The 25th International Conference on Auditory Display (ICAD), Newcastle-upon-Tyne, UK.*

[8] Thomas Neumann and Gerhard Weikum. 2010. The RDF-3X engine for scalable management of RDF data. *VLDB J.* 19, 1 (2010), 91–113. https://doi.org/10.1007/s00778-009-0165-y

[9] Nik Sawe, Chris Chafe, and Jeffrey Treviño. 2020. Using Data Sonification to Overcome Science Literacy, Numeracy, and Visualization Barriers in Science Communication. *Frontiers in Communication* 5 (2020), 46.

[10] Benjamin Warnke, Muhammad Waqas Rehan, Stefan Fischer, and Sven Groppe. 2021. Flexible data partitioning schemes for parallel merge joins in semantic web queries. In *Datenbanksysteme für Business, Technologie und Web (BTW), Dresden, Germany (LNI).*