SpeeD: An Online Multilingual Speech-based Database Design Tool

Dejan Keserovic
Muehlbauer d.o.o. Banja Luka
Banja Luka, Bosnia and Herzegovina *dejan.keserovic@muehlbauer.de*

Drazen Brdjanin
Faculty of Electrical Engineering – University of Banja Luka
Banja Luka, Bosnia and Herzegovina *drazen.brdjanin@etf.unibl.org*

Goran Banjac
Faculty of Electrical Engineering – University of Banja Luka
Banja Luka, Bosnia and Herzegovina *goran.banjac@etf.unibl.org*

Danijela Banjac
Faculty of Electrical Engineering – University of Banja Luka
Banja Luka, Bosnia and Herzegovina *danijela.banjac@etf.unibl.org*

Abstract

The paper presents SpeeD – the first online speech-based tool for automated database design. SpeeD enables automatic derivation of conceptual database models from offline (previously recorded) speech, as well as from online (real-time recorded) speech, whereby several different natural languages are supported. Once upon conceptual design is finished, SpeeD enables automated subsequent steps of forward database engineering for several contemporary database management systems.

Keywords: Automated Database Design, NLP, Speech Recognition, SpeeD, UML, VOSK.

1. Introduction

Database design has long been recognized as a significant and compelling area of research. The common design process typically begins with a conceptual design phase, which results in a CDM (*Conceptual Database Model*) that provides data descriptions on a high level of abstraction. This phase is inherently complex, and multiple iterations are often necessary to finalize the target CDM. Unlike the subsequent design steps that are usually just straightforward automated transformations of the CDM, conceptual design is typically performed manually, and its automation is very desirable in order to make it more efficient and effective.

The notion of automating CDM design dates back to the early 1980s [10]. The existing approaches primarily rely on *textual specifications* [21] or *models* [5] as input for automation. Although speech is the most natural mode of human communication, unlike text and models that are artificial, there currently exists no tool capable of automatic database design through spoken input. To address this research gap, we initiated a project [7, 8] aimed at developing a tool for fully automated, speech-based database design. This effort resulted in the creation of SpeeD – the first web-based tool that enables automatic database design using either *offline* (pre-recorded) or *online* (real-time) speech input, with support for multiple natural languages. In this paper we present the entire approach and the implemented tool.

The paper is structured as follows. After this introductory section, the second section presents the related work. The third section presents the approach and the implemented tool, while the fourth section illustrates its usage. Some evaluation results of the implemented approach are presented in the fifth section. The final section concludes the paper.

2. Related Work

Our approach combines speech recognition and text processing techniques to produce an initial CDM based on input speech, and further applies automated transformations of the CDM to obtain the target database schema. This section provides an overview of the related work, firstly presenting speech recognition, followed by an overview of automated database design.

Speech Recognition. Speech recognition is a part of NLP (*Natural Language Processing*) that deals with converting spoken language into written text. The development of ASR (*Automatic Speech Recognition*) tools began in the 1950s. The first major result was Audrey (developed at Bell Labs in 1952), which was able to recognize ten numbers in English [17]. In the following decades, ASR has advanced significantly, and various methods have been developed [16]. Some earlier approaches included HMM (*Hidden Markov Model*) [4] and N-gram [12] models, while the development of deep learning brought more advanced methods such as DNN (*Deep Neural Network*) [2] and CNN (*Convolutional Neural Network*) [14].

Nowadays, a number of ASR tools are available, including open-source solutions (e.g. Kaldi[1] and CMU Sphinx[2]), as well as commercial solutions (e.g. Google Assistant [20] and Amazon Alexa [13]). Our approach employs an open-source tool named VOSK[3]. VOSK is based on Kaldi, using its acoustic models, but optimized for simpler integration and more efficient real-time operation. VOSK supports over 20 languages, as well as small and large acoustic models – small models use traditional techniques and require fewer resources than large models that most often use more effective AI (*Artificial Intelligence*) methods.

ASR has numerous applications across various domains [26], including automotive, human-computer interaction, and education. There are also several papers demonstrating database-related ASR applications, such as speech-based synthesis of database queries and speech-controlled database manipulation (e.g., [25], [22]), but no tools allow speech-based database design, except the SpeeD tool. There is also a framework proposal for *Voice-driven Modeling* [3] that could assist in CDM design, but the speech input is not a system specification (like in our approach) but rather consists of commands (e.g., *Add a class called Person*). The same *voice-driven* approach is also applied in the ModelByVoice tool [15] to aid visually impaired individuals in modeling activities.

Automated Database Design. The existing approaches mainly enable automated database design based on *homogeneous sources* (i.e., sources of the same type). Examples include text documents (*text-based* approaches), collections of models (*model-based* approaches), or collections of forms (*form-based* approaches). There are also some papers (e.g. [1]) that consider database design based on *heterogeneous sources*, but only the DBomnia[4] tool is able to automatically derive CDMs from business process models and textual specifications.

The text-based approaches constitute the oldest and most important category [21].[5] These approaches derive CDMs from (typically unstructured) textual specifications represented in some NL (*natural language*). Most text-based approaches are *linguistics-based* [24]. They apply NLP techniques to derive CDMs from NL text. The existing text-based tools (such as ER-Converter [18], CM-Builder [11], and LIDA [19]) typically support one single source NL and do not provide multilingual support. Only TexToData [9] enables automatic database design based on textual specifications in different source NLs. In our speech-based approach, we first convert the source speech into the corresponding text, then we apply NLP techniques to process the text and generate the corresponding CDM, whereby text processing and model generation are performed by the TexToData services.

[1] https://kaldi-asr.org/doc/index.html
[2] https://cmusphinx.github.io
[3] https://alphacephei.com/vosk
[4] http://m-lab.etf.unibl.org:8080/dbomnia
[5] For a more detailed overview of other approaches, we refer the readers to [6].

3. Speech-based Database Design

This section gives a more detailed description of the speech-based approach to database design (the entire process is shown in Fig. 1) implemented by the SpeeD tool[6] (the system architecture is shown in Fig. 2). The pre-existing version of SpeeD [8] allowed the CDM synthesis only, while the subsequent forward database engineering steps were not supported at all.

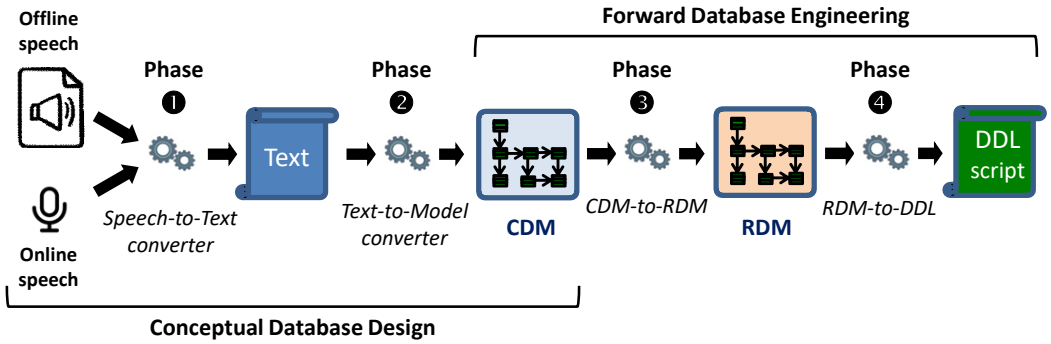

Fig. 1. Speech-based database design process

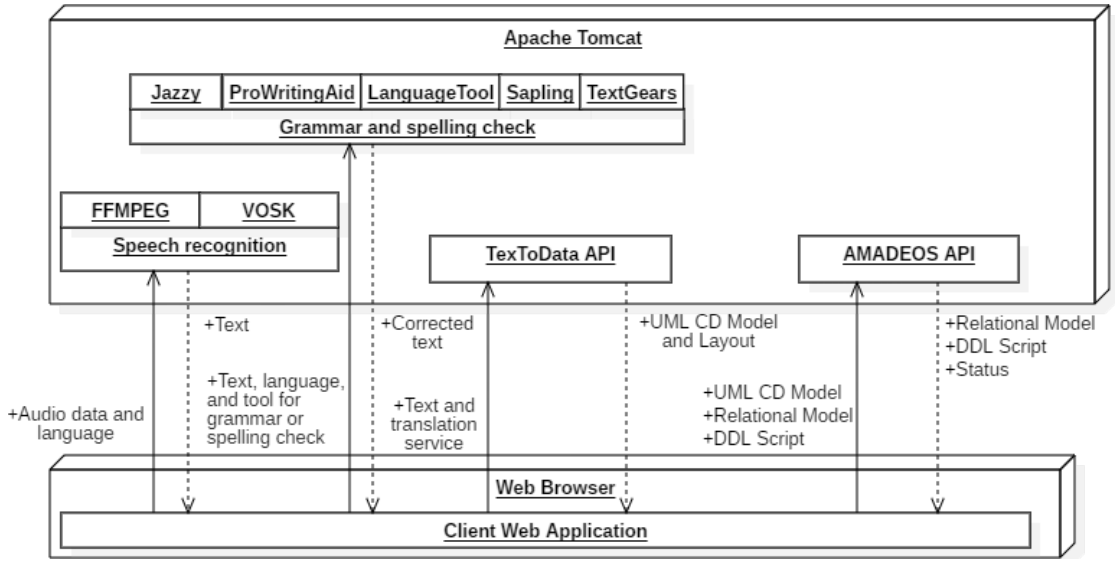

Fig. 2. SpeeD system architecture

Speech-based CDM Synthesis. The speech-based CDM synthesis consists of two phases: (1) *speech-to-text conversion*, and (2) *text-to-model conversion*.

The **first phase** involves the conversion of speech into the corresponding text. The improved SpeeD tool enables automatic recognition of *offline* (previously recorded) speech, as well as *online* (real-time recorded) speech. Here, SpeeD firstly employs the FFMPEG library to convert input audio data into the suitable format (*wav*) and modifies other characteristics of audio data (sampling rate, number of channels, and volume) suitable for ASR. After the preprocessing of input audio data, SpeeD employs the VOSK tool for ASR. In order to obtain real-time speech recognition results, the client-server communication is realized through web sockets – after the sentence is recognized, the server returns it to the client displaying it in a browser.

The initial evaluation [8] of the pre-existing SpeeD tool showed that the ASR process may result in some misrecognized words and missing punctuation in the extracted text that further

[6]SpeeD is publicly available at: *http://m-lab.etf.unibl.org:8080/SpeeD/*

affect the quality of the generated CDM. So the improved SpeeD provides an additional functionality – grammar and spelling check in the extracted text. This functionality is implemented by the GrammarCheck component employing several services (Jazzy, ProWritingAid, LanguageTool, Sapling, and TextGears) that differ in accuracy and applied grammar check method.

In the **second phase**, the resulting text specification is transformed into the corresponding CDM. Here, SpeeD employs the TexToData services to produce the target CDM. TexToData performs the text-based CDM synthesis through an orchestration[7] of several external online NLP services for translation and POS (*Part-of-Speech*) analysis, as well as internal services for CDM construction and diagram layouting.

Forward Database Engineering. Once a CDM is designed, a user proceeds with the forward engineering process. SpeeD implements this process in the same way as described in [23].

Firstly (**phase 3** in Fig. 1), a platform-independent CDM should be transformed into the corresponding RDM (*Relational Database Model*), which is platform-dependent and provides platform-specific details, such as primitive datatypes supported by the target DBMS (*Database Management System*). SpeeD represents both models (CDM and RDM) by the standard UML (*Unified Modeling Language*) class diagrams, as described in [23]. Each DBMS[8] has its own specific data types. SpeeD implements default datatype mapping for primitive types from CDM into the corresponding platform-specific datatypes in RDM, but a user is able to additionally configure this mapping together with the target DBMS selection.

Then (**phase 4** in Fig. 1), the corresponding DDL (*Data Definition Language*) script, which contains commands for the creation of the target relational database schema, is to be generated based on the RDM. This M2T (*Model To Text*) transformation is performed by an Acceleo[9] transformation program tailored for the given DBMS.

Finally (not shown in Fig. 1), the DDL script should be executed in a particular DBMS.

4. Illustrative Example of Automated Speech-based Database Design

This section presents an illustrative example of the entire automated speech-based database design process in the SpeeD tool.

Speech-based CDM Synthesis. In Fig. 3, the initial page (aimed at speech and text analysis) of the SpeeD tool is shown, along with an example of attaching an audio file and displaying the recognized text (page bottom).

In the given example, the speech recognition component successfully recognized the text, except for the end of the sentence in two places where periods are missing, along with a few missing commas. As shown in Fig. 4, these issues were detected by the TextGears tool (after clicking the *Check Text* button), and by selecting the suggested corrections, the underlined text is replaced with the chosen suggestion.

In the next step, the CDM is generated (by clicking the *Analyze Text* button) based on the corrected text. In our example, the generated CDM is shown in Fig. 5. By default, each strong entity type has an *id* attribute as the primary key represented by the corresponding *PK* operation.

Finally, the automatically generated CDM could be manually improved. In our example, the generated CDM contains all the necessary elements based on the input speech, except for the *title_description* attribute in the *course* class, which should be split into two separate attributes: *title* and *description*.

[7] For more details about the entire orchestration, we refer the readers to [9].

[8] The following DBMSs are supported: MySQL, PostgreSQL, Microsoft SQL Server, Oracle, and IBM DB2.

[9] http://www.eclipse.org/acceleo

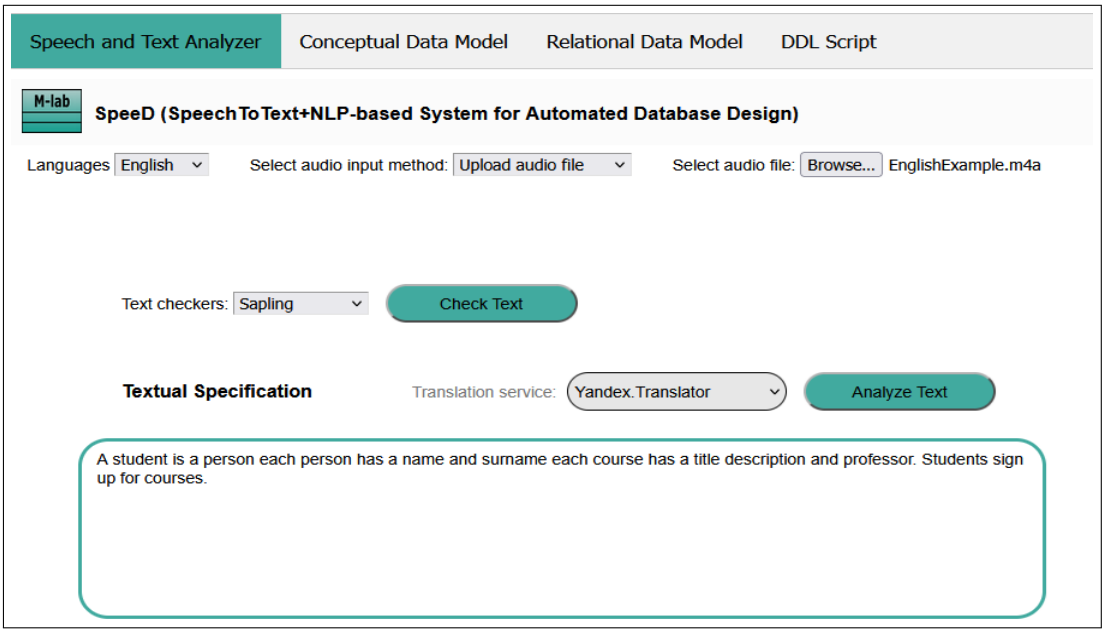

Fig. 3. Screenshot of UI form aimed at speech and text analysis

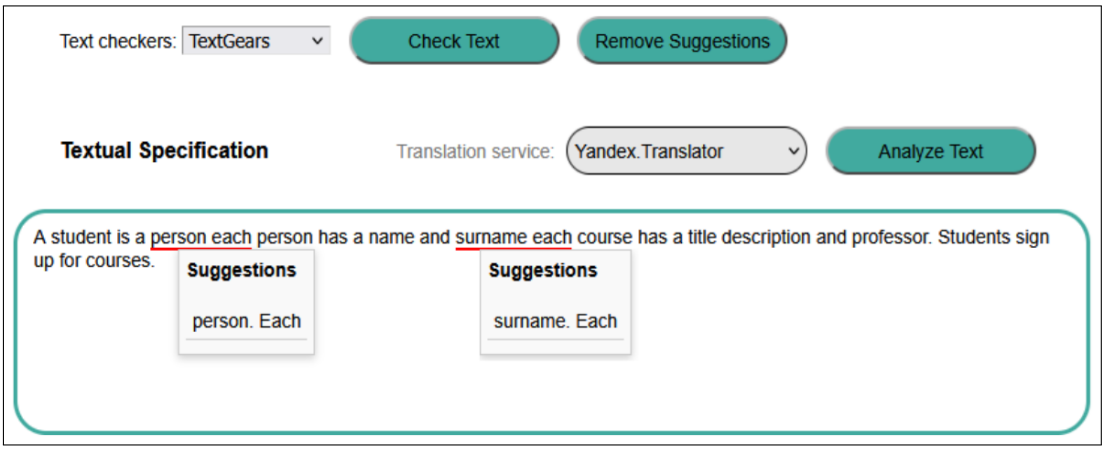

Fig. 4. Text analysis using the TextGears tool

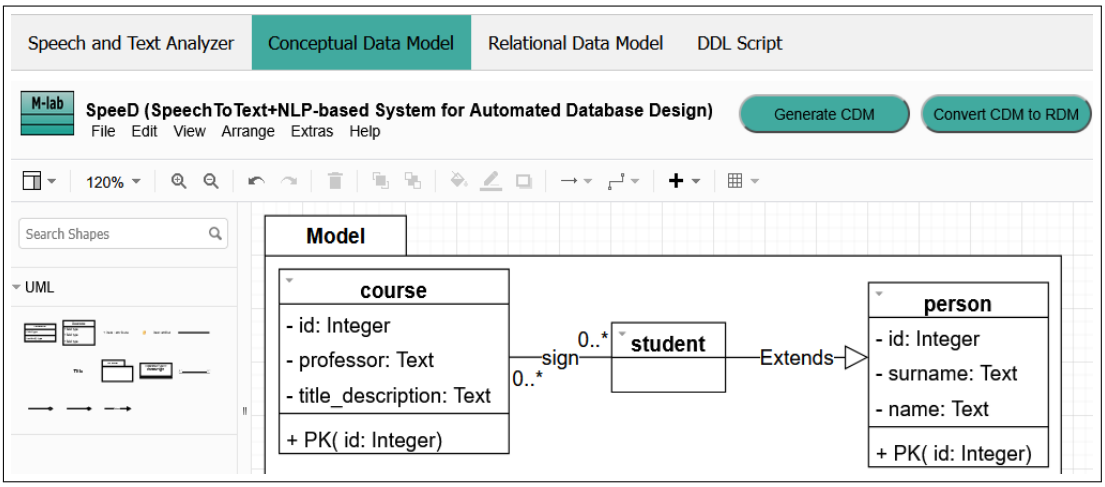

Fig. 5. Screenshot of UI form containing automatically generated CDM

Forward Database Engineering. Upon the creation, the CDM is to be transformed into the corresponding RDM. After clicking the *Convert CDM to RDM* button, it is necessary to select the target DBMS, as well as the default data type mappings. In our case (MySQL, default data mappings, automatic indices creation), we obtained the RDM shown in Fig. 6.

The automatically generated RDM could be manually improved. The typical improvements include datatype changes, since each occurrence of the same primitive datatype in CDM is mapped to the same corresponding platform-specific datatype. Such mapping is not always suitable and should be changed to satisfy specific requirements (such as format and data range). For example, if *Integer* (CDM) to *int* (RDM) is specified, then all corresponding attributes in RDM will be of the *int* type, although *smallint* is more suitable in some cases.

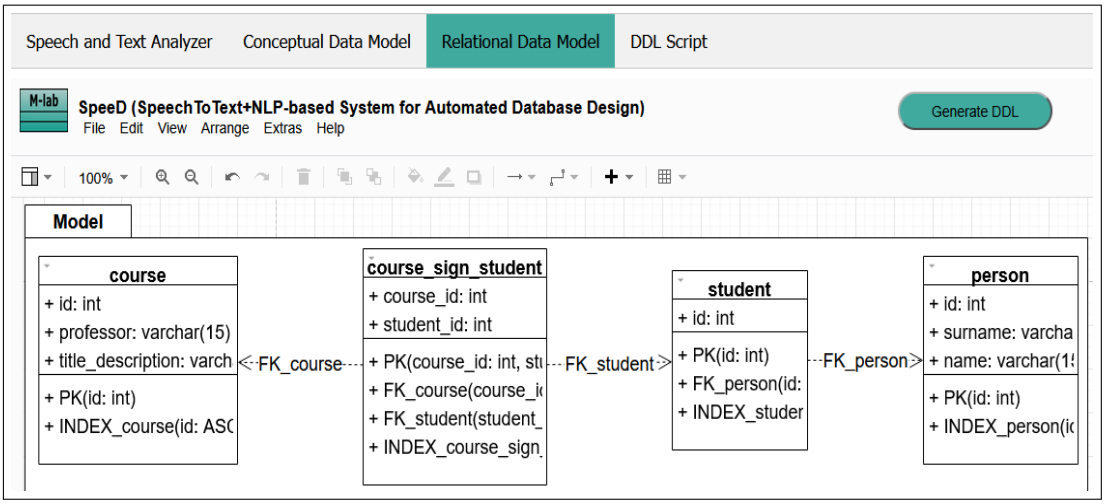

Fig. 6. Screenshot of UI form containing automatically generated RDM

Transforming the RDM into the corresponding DDL script (after clicking the *Generate DDL* button), as shown in Fig. 7, results in the DDL script.

Finally, the DDL script should be executed in a particular DBMS in order to obtain the target physical database schema. Here, SpeeD enables the user to establish a connection to the target DBMS (after clicking the *Generate Physical Database* button and entering the necessary connection parameters and execute the script. Alternatively, the user may copy the DDL script and execute it in the target DBMS, independently of the SpeeD tool.

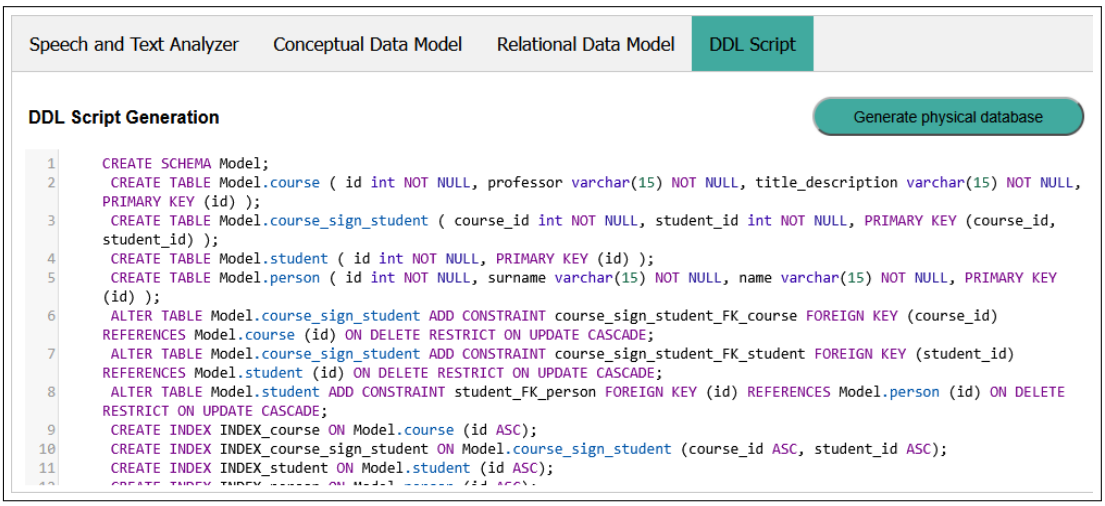

Fig. 7. Screenshot of UI form containing automatically generated DDL script

5. Evaluation

This section presents the most significant evaluation results of the presented speech-based approach to automated database design, focusing on the speech-based CDM synthesis.

The presented approach was evaluated through two (limited) case studies: *Organizational structure* and *Network infrastructure*. For each domain, textual specifications (about ten sentences) were prepared in both English and German, based on which the corresponding audio files (MP3 format) were created. For each specification, the corresponding reference CDM was manually designed and later compared with the CDMs generated by the SpeeD tool based on the recorded speech.

The main focus was on the effectiveness of the CDM synthesis with respect to the application of the small/large language models and with/without grammar correction in the recognized speech. Figure 8 shows the average *effectiveness*[10] (for four main types of concepts – classes, attributes, associations, and inheritances) of the CDM synthesis for both languages. Although the reliability of the results is limited, since the specifications were relatively short and the speech was recorded by only one speaker, the results are still very promising. The overall average effectiveness (calculated for all CDM concepts) for English is 49% (small language model), 57% (large language model), and 77% after grammar correction. The average effectiveness for German is 70% (small language model), 74% (large language model), with grammar corrections having no impact on the CDM synthesis effectiveness.

The results largely align with expectations, confirming that larger models recognize speech better than smaller ones, leading to more accurate CDMs. Additionally, grammar checking positively impacts the generated CDM, which is especially noticeable in attribute recognition. In the case of German, the results remain the same with or without grammar correction, as grammar tools did not improve text accuracy, and consequently, the CDM.

The performed qualitative and quantitative analyses confirm that the proposed approach, together with the implemented tool, enables the automated speech-based database design, achieving a (relatively) high level of correctness and completeness of the generated models.

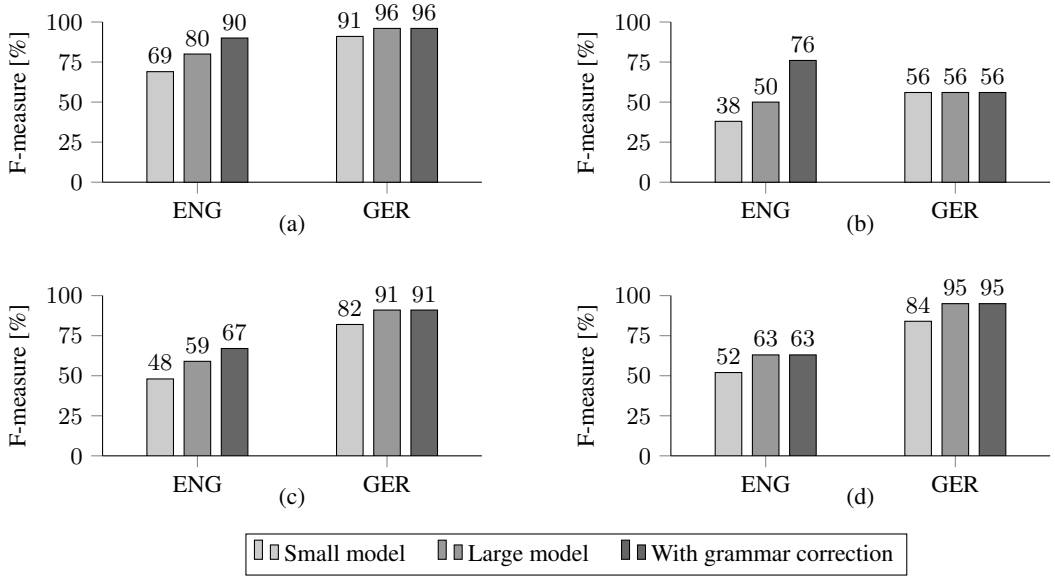

Fig. 8. Average effectiveness of automated speech-based CDM synthesis for: (a) classes, (b) attributes, (c) associations, and (d) inheritances

[10]The effectiveness, named *F-measure*, is defined as the harmonic mean of *precision* (the percentage of correctly generated concepts in the generated CDM) and *recall* (the percentage of the target CDM that is automatically generated).

6. Conclusion

In this paper, we introduced SpeeD, the first tool for automated database design based on speech. SpeeD performs speech recognition in both *online* and *offline* modes, and provides an option for grammar-checking of the recognized text, whereby multiple languages are supported. After successful speech recognition, SpeeD enables the automatic synthesis of the conceptual database model and the subsequent forward database engineering steps resulting in the physical database schema for the selected DBMS.

The experimental results demonstrate significant potential for further development of SpeeD. Relatively high accuracy of the generated conceptual database models is achieved, especially when using large VOSK models. It has been shown that grammar-checking tools positively impact the correction of recognized speech, which in turn enhances the accuracy of conceptual model generation and, consequently, the target database schema.

Although the presented illustrative example is intentionally simple to illustrate the core concepts, and the evaluation was performed on two limited case studies, our experience shows that the approach can be extended to more realistic scenarios and complex spoken input. However, with the increasing complexity of spoken input, the system will need more advanced mechanisms for grammar and context interpretation to ensure the generation of high-quality CDMs. This is partly due to the current limitation that users must formulate their input using relatively simple and well-structured sentences. These issues point to the need for further improvements of the services for text-to-model conversion. Since the results are highly dependent on the recognized speech, we also intend to research the potential usage of other ASR tools and AI-based online services as alternatives to the existing implementation.

References

[1] Banjac, G., Brdjanin, D., Banjac, D.: Automatic Conceptual Database Design based on Heterogeneous Source Artifacts. Computer Science and Information Systems 21(4), pp. 1913–1961 (2024)

[2] Bengio, Y., Bottou, L., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8), pp. 1798–1828 (2013)

[3] Black, D., Rapos, E.J., Stephan, M.: Voice-driven modeling: Software modeling using automated speech recognition. In: Proc. of MODELS-C 2019. pp. 252–258. IEEE (2019)

[4] Bosch, A.: Hidden Markov Models, pp. 493–495. Springer US, Boston, MA (2010)

[5] Brdjanin, D., Maric, S.: Model-driven Techniques for Data Model Synthesis. Electronics 17(2), pp. 130–136 (2013)

[6] Brdjanin, D., Vukotic, A., Banjac, D., Banjac, G., Maric, S.: Automatic derivation of the initial conceptual database model from a set of business process models. Computer Science and Information Systems 19(1), pp. 455–493 (2022)

[7] Brdjanin, D., Banjac, G., Babic, N., Golubovic, N.: Towards the speech-driven database design. In: Proc. of TELFOR 2022. pp. 1–4. IEEE (2022)

[8] Brdjanin, D., Banjac, G., Keserovic, D., Babic, N., Golubovic, N.: Combining speech processing and text processing in conceptual database design. Telfor Journal 16(1), pp. 8–13 (2024)

[9] Brdjanin, D., Grumic, M., Banjac, G., Miscevic, M., Dujlovic, I., Kelec, A., Obradovic, N., Banjac, D., Volas, D., Maric, S.: Towards an online multilingual tool for automated

conceptual database design. In: Braubach, L., et al. (eds.) Intelligent Distributed Computing XV. pp. 144–153. Springer (2023)

[10] Chen, P.: English sentence structure and entity-relationship diagrams. Information Sciences 29(2-3), pp. 127–149 (1983)

[11] Harmain, H., Gaizauskas, R.: CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis. Automated Software Eng. 10(2), pp. 157–181 (2003)

[12] Jurafsky, D., James, H.M.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall (2000)

[13] Kepuska, B., Bohouta, G.: Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). IEEE (2018)

[14] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86(11), pp. 2278–2324 (1998)

[15] Lopes, J., Cambeiro, J., Amaral, V.: Modelbyvoice - towards a general purpose model editor for blind people. In: Proc. of MODELS 2018 Workshops. pp. 762–769 (2018)

[16] Malik, M., Malik, M. K., Mehmood, K., Makhdoom, I.: Automatic speech recognition: A survey. Multimedia Tools and Applications 80(6), pp. 9411–9457 (2021)

[17] Meng, J., Zhang, J., Zhao, H.: Overview of the speech recognition technology. In: Proc. of the Fourth Int. Conf. on Computational and Information Sciences. pp. 199–202 (2012)

[18] Omar, N., Hanna, P., McKevitt, P.: Heuristics-based entity-relationship modelling through natural language processing. In: Proc. of AICS 2004. pp. 302–313 (2004)

[19] Overmyer, S.P., Benoit, L., Owen, R.: Conceptual modeling through linguistic analysis using LIDA. In: Proc. of ICSE 2001. pp. 401–410. IEEE (2001)

[20] Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M., Strope, B.: "your word is my command": Google search by voice: A case study. In: Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics. pp. 61–90. Springer (2010)

[21] Song, I.Y., Zhu, Y., Ceong, H., Thonggoom, O.: Methodologies for Semi-automated Conceptual Data Modeling from Requirements. In: Johannesson, P., et al. (eds.) ER 2015, pp. 18–31. Springer (2015)

[22] Song, Y., Wong, R., Zhao, X., Jiang, D.: Speech-to-sql: Towards speech-driven sql query generation from natural language question. ArXiv abs/2201.01209 (2022)

[23] Spasic, Z., Vukotic, A., Brdjanin, D., Banjac, D., Banjac, G.: UML-based forward database engineering. In: Proc. of INFOTEH 2023. pp. 1–6. IEEE (2023)

[24] Thonggoom, O.: Semi-automatic conceptual data modelling using entity and relationship instance repositories. PhD Thesis, Drexel University (2011)

[25] Vraj, S., Side, L., Arun, K., Lawrence, S.: Speakql: Towards speech-driven multimodal querying of structured data. In: Proc. of SIGMOD 2020. pp. 2363–2374. ACM (2020)

[26] Yu, Y.: Research on speech recognition technology and its application. In: Int. Conf. on Computer Science and Electronics Engineering. pp. 306–309. IEEE (2012)