Original scientific paper

Combining Speech Processing and Text Processing in Conceptual Database Design

Drazen Brdjanin, Goran Banjac, Dejan Keserovic, Nebojsa Babic, and Nenad Golubovic

***Abstract* — The paper presents an approach to automated conceptual database design that combines speech processing and text processing techniques for the automated derivation of conceptual database models from recorded speech. In the first phase, the recorded speech is converted to the corresponding text by applying speech processing techniques. In the second phase, the text is converted to the corresponding conceptual database model by applying text processing techniques. The proposed approach is supported by an online tool named *SpeeD*, which is the first tool enabling automated derivation of conceptual database models from recorded speech, whereby several different natural languages are supported.**

***Keywords* — class diagram, conceptual database model, database design, NLP, speech-driven, SpeeD, TexToData.**

I. Introduction

THE database design process typically undergoes several phases [1], whereby each phase results by the corresponding model, ranging from the *conceptual database model* (CDM), which is platform independent and provides data descriptions on a high level of abstraction, to the physical model that is platform specific and provides implementation details. The related literature is more focused on conceptual design than other phases, considering it to be the most important, since the following phases are usually just straightforward transformations of the CDM. The process of the conceptual design is not straightforward and typically requires a number of iterations before the final CDM is obtained. These are the main reasons for the long-standing interest of researchers in automating the CDM design.

The idea of the automatic CDM design dates back to the 1980s [2]. In the meantime, a number of papers have been published proposing different approaches and presenting different tools enabling (semi-)automatic CDM derivation from different sources, but there is still no tool that enables fully automatic CDM design.

The existing approaches typically take textual specifications [3] or models [4] as the source for the CDM derivation. Although speech is naturally used in human communication, in contrast to text and models that are artificial, still there is no tool aimed at CDM derivation from speech. To fill this research gap, we started a project aimed at developing an online tool for speech-based automated CDM design. In this paper, we present the *SpeeD* tool – the first tool enabling CDM derivation from recorded speech, whereby several different languages are supported.

This paper constitutes an extended version of the paper [5] presented at the *30th Telecommunications Forum* (TELFOR'22), which is extended by the recent improvements of the *SpeeD* tool. Namely, the tool presented in the Conference paper was able to process only English speech, while the improved tool supports several languages.

The paper is structured as follows. After this introduction, the second section presents the related work. The third section presents the approach and the implemented *SpeeD* tool, while the fourth section illustrates its usage. The final section concludes the paper.

II. Related Work

This section presents the related work, whereby the first subsection presents an overview of the (semi-)automatic CDM design, while the second subsection provides an overview of speech recognition.

A. (Semi-)Automatic CDM design

The existing approaches (Fig. 1) to (semi-)automatic CDM design can be classified as: (1) *text-based*, (2) *model-based*, (3) *form-based*, and (4) *speech-based*.

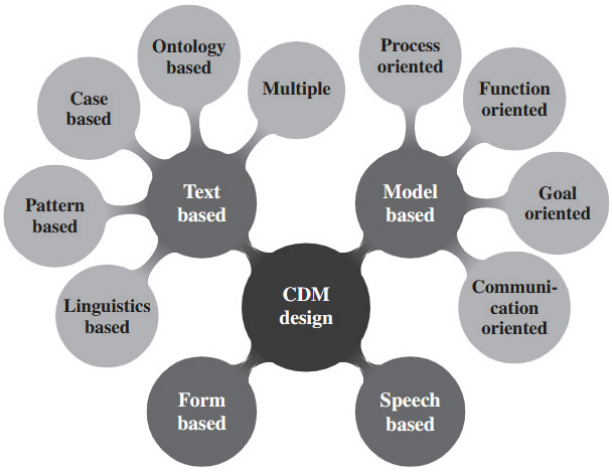

Fig. 1. Taxonomy of approaches to (semi-)automatic CDM design.

Paper received July 07, 2023; revised March 13, 2024; accepted July 30, 2024. Date of publication August 02, 2024. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vlado Delić.

This paper is a revised and expanded version of the paper presented at the 30th Telecommunications Forum TELFOR 2022 [5].

Corresponding author Drazen Brdjanin is with the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina
(phone: 387-51-221851; e-mail: drazen.brdjanin@etf.unibl.org).

Goran Banjac is with the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina (phone: 387-51-221831; e-mail: goran.banjac@etf.unibl.org).

Dejan Keserovic is with Lanaco, Banja Luka, Bosnia and Herzegovina
(phone: 387-66-314076; e-mail: dejankeserovic1@gmail.com).

Nebojsa Babic is with HTEC Group, Banja Luka, Bosnia and Herzegovina
(phone: 387-66-432748; e-mail: nebojsa.baabic@gmail.com).

Nenad Golubovic is with Bravo Systems, Banja Luka, Bosnia and Herzegovina
(phone: 387-65-348342; e-mail: n.golubovic1994@gmail.com).

Text-based approaches. This is the oldest and most dominant category of approaches. These approaches and tools derive CDMs from textual specifications that are typically unstructured and represented in some NL[1]. According to [6], they can be classified as: (1) *linguistics-based*, (2) *pattern-based*, (3) *case-based*, (4) *ontology-based*, and (5) *multiple approaches*.

Most text-based approaches and tools fall into the *linguistics-based* category. These approaches use NLP[2] techniques to convert NL text into CDM. The development of these approaches started with Chen's eleven rules [2] for the translation of English text into the corresponding E-R[3] diagram, which have been further enhanced and extended in [7]–[9].

The most important tools implementing the linguistics-based approach are: *ER-Converter* [8], *CM-Builder* [10], and *LIDA* [7]. The main representatives of other categories are: pattern-based *APSARA* [11], case-based *CABSYDD* [12], ontology-based *OMDDE* [13], and *HBT* [6] belonging to the category of multiple approaches. The existing tools typically support one single source NL (mainly English) and do not provide multilingual support. Only *TexToData* [14] enables automatic CDM derivation from textual specifications in different source NLs, even with very complex morphology, such as Slavic languages. In our speech-based approach, firstly we convert the source speech into the corresponding text, then we apply NLP techniques to process the text and generate the target CDM, whereby text processing and model generation are performed by the *TexToData* services.

Model-based approaches. These approaches emerged as an alternative to the text-based approaches, in order to avoid their shortcomings mainly related to the modest effectiveness for languages with complex morphology. According to [4], even 18 different graphical notations are used in the existing approaches, which can be classified as: (1) *process-oriented* (e.g. BPMN[4]), (2) *function-oriented* (e.g. Data Flow Diagram), (3) *communication-oriented* (e.g. Sequence Diagram), and (4) *goal-oriented* (e.g. TROPOS).

There is still no approach or tool enabling automatic derivation of the complete target CDM from a source model (regardless of the notation). Only a few papers present a set of formal rules for automatic CDM derivation (e.g. [16], [17]), while the majority give only guidelines and informal rules that do not enable automatic CDM derivation. Most of the proposed tools are actually transformation programs (e.g. [18], [19]) specified in some model-to-model transformation language (such as ATL[5]), while only a small number of papers present real CASE[6] tools for automatic model-driven CDM synthesis (e.g. [21], [22]). There is only one single online tool named *AMADEOS* [22], which enables the automatic derivation of an initial CDM from a set of business process models, whereby the most recent release [23] supports a complete model-driven database design process from conceptual model to physical database, by using the standard UML[7] notation. Apart from the functionalities publicly provided to the end-user, *AMADEOS* also exposes services for automatic model-driven CDM derivation, including services for diagram layouting and model export, which are also employed in *TexToData* and *SpeeD*.

Form-based approaches. These approaches take collections of forms as the source for CDM derivation. The most important tools are *EDDS* [25] and *IIS*Case* [26].

Speech-based CDM Derivation. In the existing literature there are some papers considering speech as the source for automatic synthesis of database queries and speech-controlled database manipulation (e.g. [27], [28]), but there are no papers considering the speech-based database design (except the TELFOR paper [5] presenting the *SpeeD* tool).

B. Speech Recognition

Speech recognition is one of the parts of NLP that enables the conversion of spoken language into written text. Since the 1950s, researchers have tried to create such a tool, resulting in the first ASR[8] tool called *Audrey* developed in the Bell Labs, which could identify ten English digits [29]. In the coming decades, many ASR tools have been developed using methods including the HMM[9], the *N-gram model*, and LSTM[10], as well as the deep learning techniques: RNN[11], DNN[12], and CNN[13], which represent the most popular way of achieving ASR of nowadays [30].

Speech analysis begins with the audio signals obtained through a microphone or from audio files. The audio signal is cleaned using *normalization*, *filtering*, and other techniques. The relevant data from the cleaned audio is then obtained through a process called *feature extraction*. Some of the techniques used for feature extraction are LPC[14] [31], MFCC[15] [32], and DTW[16] [33], [34]. The processed audio signal is separated into sets for analysis and forwarded to an *acoustic model*. The acoustic model then calculates probabilities of different linguistic units, and using these probabilities, the *language model* generates text that best matches the audio signal.

The speech recognition process can be executed using different open-source end-to-end *ASR tools*, such as notable examples *Kaldi* [35], *DeepSpeech* [36], *Vosk*, and *LinTO*. Kaldi uses a traditional HTK[17], SGMM[18] and has recently introduced DNN-HMM[19]. In contrast, *DeepSpeech* utilizes DNN systems [37]. The process could also be executed using platforms like *Google Assistant* and Amazon's *Alexa*, providing AI[20] functionalities beyond speech recognition.

[1] Natural Language
[2] Natural Language Processing
[3] Entity-Relationship
[4] Business Process Model and Notation [15]
[5] ATLAS Transformation Language [20]
[6] Computer Aided Software Engineering
[7] Unified Modeling Language [24]
[8] Automatic Speech Recognition
[9] Hidden Markov model
[10] Long Short-Term Memory
[11] Recurrent Neural Network
[12] Deep Neural Network
[13] Convolutional Neural Network
[14] Linear Predictive Coding
[15] Mel-Frequency Cepstral Coefficients
[16] Dynamic Time Warping
[17] Hidden Markov Model Toolkit
[18] Subspace Gaussian Mixture Model
[19] Deep Neural Network – Hidden Markov Model
[20] Artificial Intelligence

In this paper, we focus on the open-source ASR tool called *Vosk*[21], which is based on the *Kaldi* toolkit. *Vosk* offers speech recognition for more than 20 languages, including English, Russian, Chinese, Italian, German, Greek, etc. It supports both small and large models as well as offline and online recognition. Small models are typically a few tens of MBs, while large models, employing AI, offer greater precision and are usually between 1 and 2 GB in size.

ASR tools utilize language models trained on large datasets to assess words based on previous predictions [38]. Training the model requires substantial datasets, especially when dealing with ASR systems based on deep learning, where a considerably larger amount of data is needed to train the model [39]. *Vosk* models incorporate data from acoustic models, language models, and phonetic dictionaries to construct a recognition graph. Users can create their custom *Vosk* models and adapt them for specific purposes.

One of the challenges in ASR is analyzing statements from different speakers due to acoustic differences. Additional challenges arise even in the case of a single speaker, given the variation in speech speed and intensity. Noise poses an additional challenge because models are usually trained on clean speech data, which can lead to potential speech recognition errors [40]. Similar to other ASR tools, achieving optimal recognition quality with *Vosk* depends on having clean audio without background noise. To ensure high speech recognition quality, it is also important to use audio with the sampling rates on which models are typically trained, such as 8kHz or 16kHz.

In recent years, research has focused on various techniques to improve ASR systems, addressing the challenging negative effects of acoustic environments. For example, the REVERB challenge [41] specifically targets reverberation, while CHiME challenges [42] are designed to recognize speech from multiple sources as well as noisy background speech.

III. From Speech to Conceptual Database Model

This section presents the approach and the *SpeeD*[22] tool.

The process of the speech-based CDM synthesis, as shown in Fig. 2, consists of two phases. In the first phase, a recorded speech is converted to the corresponding text by applying speech processing techniques. Then, in the second phase, the extracted text is converted to the corresponding CDM by applying text processing techniques.

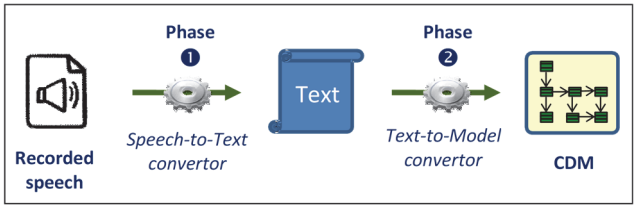

Fig. 2. Two-phase speech-based CDM synthesis.

The aforementioned two-phase approach is implemented by the *SpeeD* tool. *SpeeD* is an online web-based tool aimed at CDM derivation from recorded speech, whereby several different natural languages are supported (English, German, French, Italian, Chinese, Greek, Turkish, etc.) as well as a variety of input formats (*m4a*, *mp3*, *ogg*, *wav*, *wma*, etc.). Fig. 3 shows the tool architecture, while Fig. 4 shows a screenshot of the tool in action.

A. Phase 1: Speech to Text

In the first phase, recorded speech is converted to text, where the *client web application* allows users to upload an audio file with the recorded speech. The recorded speech and selected language are sent to the *SpeechToText* service, which is responsible for the speech-to-text conversion. The *SpeechToText* service plays the role of an adapter that employs another service – currently we employ the *Vosk SpeechToText* service. With this adapter it will be easy to employ other speech-to-text services that we plan to develop in the future.

The *Vosk SpeechToText* service is based on *Vosk*, which is based on the signal database concept – it works by applying audio fingerprinting to the chunks of the audio file and during decoding, the fingerprint hash is looked up in the database. Although speech recognition bindings are implemented for various programming languages, in our case we choose Java implementation. Since *Vosk* requires mono-channel *wav* audio files with 16 k sampling rate, in order to allow the users to upload a variety of audio formats, we use the *FFMPEG*[23] tool, to convert input audio files into the *Vosk*-compatible format.

When the speech-to-text conversion is finished, the *client web application* receives the response (text) from the *SpeechToText* service and populates the corresponding input field. If necessary, users are now able to improve the text.

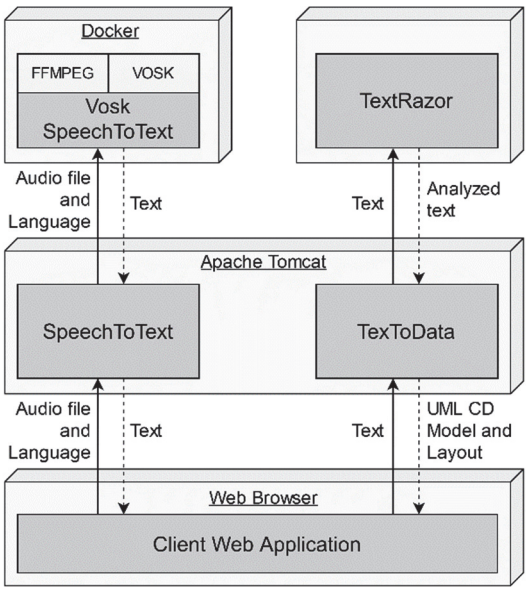

Fig. 3. *SpeeD* architecture.

[21] https://alphacephei.com/vosk
[22] http://m-lab.etf.unibl.org:8080/SpeeD
[23] https://ffmpeg.org

Original scientific paper

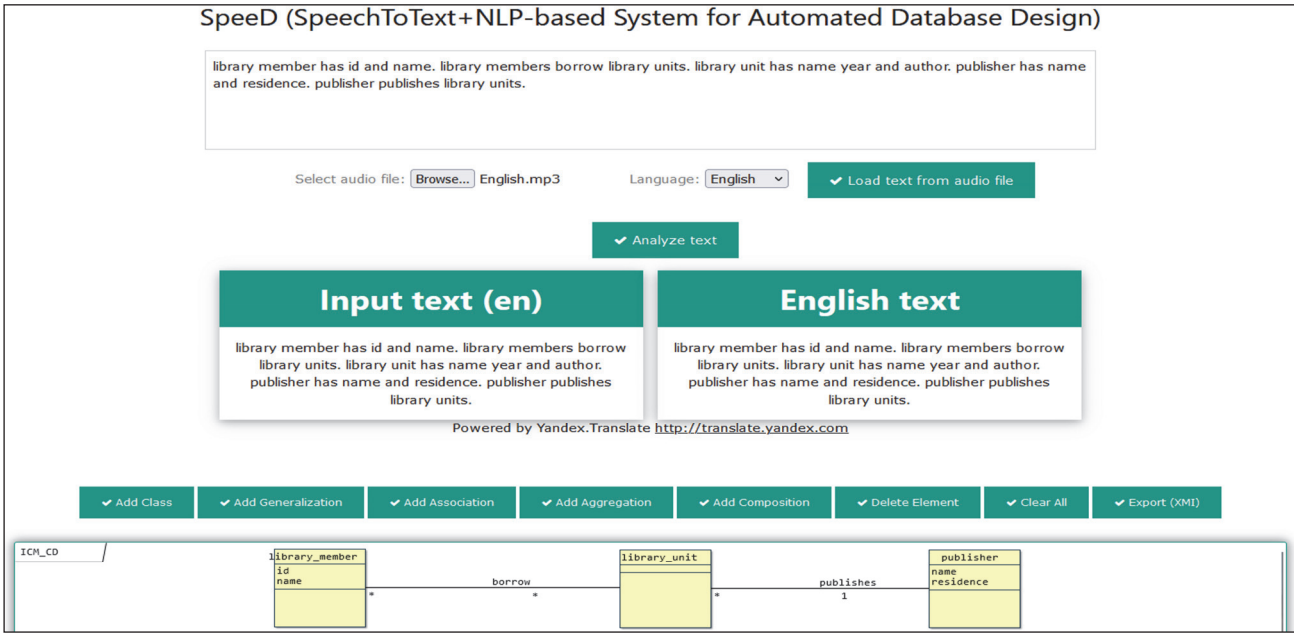

Fig. 4. Screenshot of *SpeeD* in action.

B. Phase 2: Text to CDM

In the second phase, the generated text is sent to the *TexToData*[24] service, which processes the text, generates the corresponding CDM, and returns it back to the *client web application*. The whole process is implemented as an orchestration[25] of internal as well as external services for text translation, text analysis, model generation, model translation, model serialization, diagram layouting, etc.

If the source language is not English, then *TexToData* firstly forwards the source text to the external *Yandex*[26] translation service, which translates the source text and returns the corresponding English text. The English text is further sent to the external *TextRazor*[27] service that performs NLP and returns the analyzed text. Based on these results, *TexToData* applies a set of deterministic rules [14] and generates an internal representation of the CDM. If the source language is not English, then the CDM is translated back into to source language, again by the external *Yandex* translation service. The CDM is further serialized as the corresponding UML class diagram in the XMI[28] format. After the serialization, the model is sent to the *CDlayouter service* (shared *AMADEOS* service), which creates and returns a layout of the class diagram. Finally, the model and the diagram are merged into a single JSON object and returned to the *client web application*.

When the *client web application* receives the JSON[29] response from the *TexToData* service, it visualizes[30] the class diagram in the browser. The visualized diagram is editable so users can additionally improve it. It is also possible to export the model in the XMI format.

IV. Illustrative Examples

In this section we illustrate the use of the implemented tool with two simple examples. The first example illustrates a basic scenario of the CDM derivation from English speech, while the second example illustrates the multilingual support through the CDM derivation from Greek speech.

A. Basic Scenario: Deriving CDM from English Speech

For this illustration, we prepared an audio file (*mp3*) with a recorded English speech of the simple description of the Faculty Library (shown in Fig. 5). After we upload the audio file, the *SpeechToText* service responds with the recognized text, and the *client web application* populates the corresponding text input field with the extracted text (shown in Fig. 6). At this point, the extracted text from the recorded speech can be used as-is, or user can modify the text before sending it to the *TexToData* service. In our example, it is obvious that the recognized text has some grammar flaws – namely, some commas are missing in the extracted text.

If we use the extracted text without modification as the source text for the *TexToData* service, we obtain a class diagram shown in Fig. 7 (left). If we correct grammatical errors in the extracted text (which means that we actually use the text shown in Fig. 5), and use such modified text as the source for the *TexToData* service, then we obtain a class diagram shown in Fig. 7 (right). A comparison of the generated CDMs shows that they slightly differ. When the extracted text is used without modification, attributes (*name*, *year*, and *author*) are missing in the *library_unit* class, which is caused by the already mentioned grammar flaws (missing commas) in the source text. The other two classes (*library_member* and *publisher*), as well as the associations (*borrow* and *publishes*), are correctly generated.

[24] http://m-lab.etf.unibl.org:8080/Textodata
[25] The complete description of the orchestration is provided in [14].
[26] https://translate.yandex.com
[27] https://www.textrazor.com
[28] XML Metadata Interchange [43]
[29] JavaScript Object Notation
[30] The implementation is based on the jsUML2 library (http://www.jrromero.net/tools/jsUML2)

Library member has id and name. Library members borrow library units. Library unit has name, year and author. Publisher has name and residence. Publisher publishes library units.

Fig. 5. Simple textual description of Faculty Library.

library member has id and name. library members borrow library units. library unit has name year and author. publisher has name and residence. publisher publishes library units.

Fig. 6. Text extracted from recorded English speech (without modification).

Fig. 7. CDM derived from recorded speech without text modification (left), and after text modification (right).

B. Multilingual Support

In order to illustrate the multilingual support in the *SpeeD* tool, we prepared an audio file (*mp3*) with a recorded Greek speech of the same sample description of the faculty library. After we upload the audio file, the *SpeechToText* service responds with the recognized text (Fig. 8). After some minor grammar improvements (Fig. 9), we obtain the CDM shown in Fig. 10.

το μέλος της βιβλιοθήκης έχει ταυτότητα και όνομα τα μέλη της βιβλιοθήκης δανείζονται μονάδες βιβλιοθήκης. η μονάδα βιβλιοθήκης έχει όνομα έτος και συγγραφέα. ο εκδότης έχει όνομα και κατοικία. ο εκδότης δημοσιεύει μονάδες βιβλιοθήκης.

Fig. 8. Text extracted from recorded Greek speech (without modification).

το μέλος της βιβλιοθήκης έχει ταυτότητα και όνομα. τα μέλη της βιβλιοθήκης δανείζονται μονάδες βιβλιοθήκης. η μονάδα βιβλιοθήκης έχει όνομα έτος και συγγραφέα. ο εκδότης έχει όνομα και κατοικία. ο εκδότης δημοσιεύει μονάδες βιβλιοθήκης.

Fig. 9. Text extracted from recorded Greek speech (after modification).

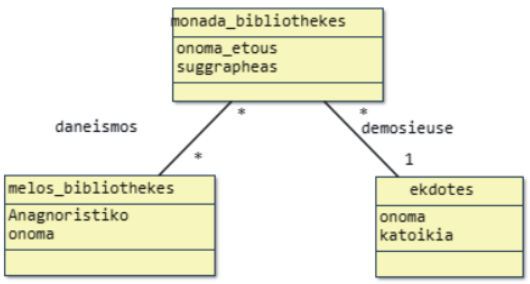

Fig. 10. CDM derived from Greek speech.

C. Discussion

The presented examples show that the implemented *SpeeD* tool has drawbacks compared to the existing text-based tools, since the speech recognition may introduce some noise, making the entire speech-based process very dependent on the speech recognition phase.

However, the presented examples also show that the proposed approach and the implemented tool have great potential for practical use in the future. For instance, *SpeeD* could be used for automatic CDM extraction directly from the recorded user stories, which could be very beneficial in agile software development.

V. Conclusion

In this paper, we presented *SpeeD* – the first online web-based tool that enables conversion of recorded speech into the CDM represented by a UML class diagram, whereby the CDM derivation is executed in two phases. In the first phase, the recorded speech is converted to the corresponding text. In the second phase, the text is converted to the corresponding CDM by an NLP-based tool that uses external services. After the first step, users are able to additionally improve the extracted text and thus influence the generation of the target CDM, which makes the entire process semi-automatic.

The initial results show that the implemented tool has drawbacks compared to the existing tools taking other artifacts (such as text and models) as the source for CDM derivation, but also show that the approach has a huge potential. The presented prototype constitutes a very pioneering achievement in the field of speech-driven database design, so a plethora of open issues should be resolved in the future, which should not be limited only to the drawbacks of the presented prototype, but also many other challenges such as online speech recognition, etc.

References

[1] C. Date, *An Introduction to Database Systems*, 8th ed. Addison-Wesley, 2003.

[2] P. Chen, "English sentence structure and entity-relationship diagrams," *Information Sciences*, vol. 29, no. 2-3, pp. 127–149, 1983.

[3] I. Y. Song, Y. Zhu, H. Ceong, and O. Thonggoom, "Methodologies for semi-automated conceptual data modeling from requirements," in *Proc. of ER 2015*, pp. 18–31.

[4] D. Brdjanin, and S. Maric, "Model-driven techniques for data model synthesis," *Electronics*, vol. 17, no. 2, pp. 130–136, 2013.

[5] D. Brdjanin, G. Banjac, N. Babic, and N. Golubovic, "Towards the speech-driven database design," in *Proc. of TELFOR* 2022, pp. 1-4.

[6] O. Thonggoom, "Semi-automatic conceptual data modelling using entity and relationship instance repositories," PhD Thesis, Drexel University, 2011.

[7] S. P. Overmyer, L. Benoit, and R. Owen, "Conceptual modeling through linguistic analysis using LIDA," in *Proc. of ICSE 2001*, pp. 401–410.

[8] N. Omar, P. Hanna, and P. McKevitt, "Heuristics-based entity-relationship modelling through natural language processing," in *Proc. of AICS 2004*, pp. 302–313.

[9] S. Hartmann, and S. Link, "English sentence structures and EER modeling," in *Proc. of the 4th Asia-Pacific conf. on conceptual modeling*, 2007, pp. 27–35.

[10] H. Harmain, and R. Gaizauskas, "CM-Builder: A natural language-based CASE tool for object-oriented analysis," *Automated Software Engineering*, vol. 10, no. 2, pp. 157–181, 2003.

[11] S. Purao, "APSARA: A tool to automate system design via intelligent pattern retrieval and synthesis," *SIGMIS Database*, vol. 29, no. 4, pp. 45–57, 1998.

[12] J. Choobineh, and A. W. Lo, "CABSYDD: Case-based system for database design," *Journal of Management Information Systems*, vol. 21, no. 3, pp. 281–314, 2004.
[13] V. Sugumaran, and V. C. Storey, "Ontologies for conceptual modeling: their creation, use, and management," *Data & Knowledge Engineering*, vol. 42, no. 3, pp. 251–271, 2002.
[14] D. Brdjanin, M. Grumic, G. Banjac, M. Miscevic, I. Dujlovic, A. Kelec, N. Obradovic, D. Banjac, D. Volas, and S. Maric, "Towards an online multilingual tool for automated conceptual database design," in *Proc. of IDC 2022*, pp. 144–153.
[15] *Business Process Model and Notation (BPMN)*, v2.0, OMG, 2011.
[16] H. B. K. Tan, Y. Yang, and L. Blan, "Systematic transformation of functional analysis model in object oriented design and implementation," *IEEE Transactions on Software Engineering*, vol. 32, no. 2, pp. 111–135, 2006.
[17] D. Brdjanin, and S. Maric, "An approach to automated conceptual database design based on the UML activity diagram," *Computer Science and Information Systems*, vol. 9, no. 1, pp. 249–283, 2012.
[18] A. Rodriguez, I. Garcia-Rodriguez de Guzman, E. Fernandez-Medina, and M. Piattini, "Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach," *Information and Software Technology*, vol. 52, no. 9, pp. 945–971, 2010.
[19] A. Kriouile, N. Addamssiri, and T. Gadi, "An MDA method for automatic transformation of models from CIM to PIM," *American Journal of Software Engineering and Applications*, vol. 4, no. 1, pp. 1–14, 2015.
[20] F. Jouault, F. Allilaire, J. Bezivin, and I. Kurtev, "ATL: A model transformation tool," *Science of Computer Programming*, vol. 72, no. 1-2, pp. 31–39, 2008.
[21] O. Nikiforova, K. Gusarovs, O. Gorbiks, and N. Pavlova, "BrainTool: A tool for generation of the UML class diagrams," in *Proc. of ICSEA 2012*, pp. 60–69.
[22] D. Brdjanin, A. Vukotic, D. Banjac, G. Banjac, and S. Maric, "Automatic derivation of the initial conceptual database model from a set of business process models," *Computer Science and Information Systems*, vol. 19, no. 1, pp. 455–493, 2022.
[23] Z. Spasic, A. Vukotic, D. Brdjanin, D. Banjac, and G. Banjac, "UML-based forward database engineering," in *Proc. INFOTEH* 2023, pp. 1-6.
[24] *Unified Modeling Language (OMG UML)*, v2.5. OMG, 2015.
[25] J. Choobineh, M. Mannino, J. Nunamaker, and B. Konsynsky, "An expert database design system based on analysis of forms," *IEEE Transaction on Software Engineering*, vol. 14, no. 2, pp. 242–253, 1988.
[26] I. Lukovic, P. Mogin, J. Pavicevic, and S. Ristic, "An approach to developing complex database schemas using form types," *Software: Practice & Experience*, vol. 37, no. 15, pp. 1621–1656, 2007.
[27] S. Vraj, L. Side, K. Arun, and S. Lawrence, "SpeakQL: Towards speech-driven multimodal querying of structured data," in *Proc. of SIGMOD 2020*, pp. 2363–2374.
[28] Y. Song, R. Wong, X. Zhao, and D. Jiang, "Speech-to-SQL: Towards speech-driven SQL query generation from natural language question," *ArXiv*, vol. abs/2201.01209, 2022.
[29] J. Meng, J. Zhang, and H. Zhao, "Overview of the speech recognition technology," in *Proc. of the Fourth Int. Conf. on Computational and Information Sciences*, pp. 199-202, 2012.
[30] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," Multimedia *Tools and Applications*, vol. 80, no. 6, pp. 9411–9457, 2021.
[31] B. Millidge, A. K Seth, and C. L Buckley, "Predictive coding: A theoretical and experimental review," *ArXiv*, vol. abs/2107.12979, 2022.
[32] Z. K. Abdul, and A. B. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122136-122158, 2022.
[33] P. Senin, "Dynamic time warping algorithm review," Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA, vol. 855, 2008.
[34] A. Trivedi, N. Pant, P. Shah, S. Sonik, and S. Agrawal, "Speech to text and text to speech recognition systems – A review," *IOSR Journal of Computer Engineering*, vol. 20, no. 2, pp. 36-43, 2018.
[35] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.
[36] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *ArXiv*, vol. abs/1412.5567, 2014.
[37] A. Trabelsia, S. Warichet, Y. Aajaounb, and S. Soussilane, "Evaluation of the efficiency of state-of-the-art speech recognition engines," *Procedia Computer Science*, vol. 207, pp. 2242-2252, 2022.
[38] P. Karmakar, S. Wei Teng, and G. Lu, "Thank you for attention: A survey on attention-based artificial neural networks for automatic speech recognition," ArXiv, vol. abs/2102.07259, 2021.
[39] J. U. Bang, M. Y. Choi, S. H. Kim, and O. W. Kwon, "Automatic construction of a large-scale speech recognition database using multi-genre broadcast data with inaccurate subtitle timestamps," *IEICE Transactions on Information and Systems*, vol. 103-D, no. 2, pp. 406-415, 2020.
[40] H. Aldarmaki, A. Ullah, S. Ram, and N. Zaki, "Unsupervised Automatic Speech Recognition: A review," *Speech Communication*, vol. 139, pp. 76-91, 2022.
[41] K. Kinoshita, M. Delcroix, S. Gannot, E. A. P. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka, "A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 7, pp. 1-19, 2016.
[42] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. of Interspeech 2018*, pp. 1561-1565, 2018.
[43] *XML Metadata Interchange*, v2.5.1, OMG, 2015.