

ІНФОРМАЦІЙНА СИСТЕМА ОЗВУЧЕННЯ УКРАЇНОМОВНОГО ТЕКСТУ НА ОСНОВІ МЕТОДІВ NLP ТА МАШИННОГО НАВЧАННЯ

Ілля Белоусов¹, Любомир Чирун², Софія Чирун¹, Ігор Будз³, Ольга Власенко^{4,5}

¹ Національний університет “Львівська політехніка”, кафедра інформаційних систем та мереж,
вул. С. Бандери, 12, Львів, Україна

² Львівський національний університет імені Івана Франка, кафедра прикладної математики,
вул. Університетська, 1, Львів, Україна

³ Національний університет “Львівська політехніка”, кафедра спеціалізованих комп’ютерних систем,
вул. С. Бандери, 12, Львів, Україна

⁴ Університет Оснабрюка, Інститут науки про освіту, Інститут педагогічної науки,
вул. Хегер-Тор-Волл, 9, Оснабрюк, Німеччина

⁵ Житомирський державний університет імені Івана Франка, кафедра професійно-педагогічної, спеціальної освіти, андрагогіки та управління, вул. Велика Бердичівська, 40, Житомир, Україна

E-mail: illia.bielousov.knm.2018@lpnu.ua, ORCID: 0000-0002-4851-4460

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002-9448-1751

E-mail: sofii.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

E-mail: Ihor.S.Budz@lpnu.ua, ORCID: 0000-0002-5400-0984

E-mail: olha.vlasenko@uni-osnabrueck.de, ORCID: 0000-0001-7258-2108

© Белоусов І. С., Чирун Л. В., Чирун С. Л., Будз І. С., Власенко О. М., 2023

Під час дослідження розроблено інформаційну систему озвучення україномовного тексту на основі методів NLP та машинного навчання. Створена інформаційна система реалізована у вигляді десктоп-додатка, який дає змогу здійснювати озвучення україномовного тексту. Створення системи охоплювало всі стадії розроблення програмного забезпечення: процес проєктування, процес реалізації та процес тестування. Щоб обґрунтувати доцільність створення такої системи, ми проаналізували вже наявні програмні рішення на ринку, їхні переваги та недоліки, які урахували, створюючи нову систему. Під час системного аналізу системи сформовано дерево цілей, дерево рішень та наведено приклади контекстних діаграм із декомпозицією процесів. Одним із етапів є оформлення економічної частини, де проаналізовано бюджет, потрібний для реалізації системи, розраховано усі витрати на податки та адміністративні витрати, також проаналізовано стратегії розвитку та вибрано стратегію розвитку продукту із супутніми рішеннями та стратегію розвитку продукту. Після цього надано оцінку доцільності створення проєктованої системи, її окупності та прибутку. Об’єкт дослідження – процес системи озвучення україномовного тексту на основі методів NLP та машинного навчання. Предмет дослідження – методи та засоби побудови системи озвучення україномовного тексту на основі методів NLP та машинного навчання. Метою дослідження є створення інформаційної системи озвучення україномовного тексту на основі методів NLP та машинного навчання. Результат роботи – готова до реалізації інформаційна система озвучення україномовного тексту на основі методів NLP та машинного навчання. Здійснено аналітичний огляд літературних та онлайн-джерел, що стосуються теми озвучення україномовного тексту на основі методів NLP та машинного навчання, системний аналіз об’єкта дослідження, аналіз та вибір програмних засобів для реалізації системи, практичну реалізацію системи, економічне обґрунтування діяльності впровадження систем.

Ключові слова: інформаційна система; NLP; україномовний текст; машинне навчання; озвучення тексту.

Вступ. Загальна постановка проблеми

Сьогодні людині для роботи, навчання або просто ознайомлення доводиться працювати з текстовими файлами, яким зазвичай є звіти, журнали, книги тощо. Об'єкт дослідження – процес озвучення україномовного тексту. Предмет дослідження – методи та засоби побудови системи озвучення україномовного тексту. Практична цінність роботи полягає у створенні інформаційної системи озвучення, що надає засоби для озвучення україномовного тексту.

Наукова новизна роботи полягає у розробленні загальної типової архітектури інформаційної системи озвучення україномовного тексту на основі методів NLP та машинного навчання, формулювання основних функціональних вимог до таких систем та розроблення загального ІТ-проекту і підходу до реалізації таких типових систем.

Аналіз останніх досліджень та публікацій

Синтез мови – це перетворення заздалегідь не відомої текстової інформації на аудіомову. Мовне виведення інформації – це реалізація мовного інтерфейсу для спрощення використання системи [9]. Фактично, завдяки синтезу мови надається ще один канал передавання даних від комп'ютера, мобільного телефону та інших пристроїв до людини, аналогічно до монітора [3–5]. Звичайно, передати малюнок голосом неможливо, але прослухати електронну пошту або розклад на день інколи доволі зручно, особливо якщо в цей час погляд зайнятий чимось іншим. Наприклад, прийшовши вранці на роботу, готуючись до переговорів, можна поправляти біля дзеркала краватку або зачіску, тоді як комп'ютер читає вголос останні новини, пошту або нагадує важливу інформацію для переговорів. В побуті доводиться стикатися з різними системами синтезу мови. Ось кілька прикладів. Служби, що повідомляють по телефону поточний час, використовують синтезатор мови. Диктор називає цифри й такі слова, як “годин”, “хвилин” та “секунд”, а комп'ютер синтезує повідомлення про поточний час. Технології синтезу мови застосовують у метро для оголошення зупинок. Власники мобільних телефонів можуть спілкуватися з автоматичною сервісною службою для визначення залишку коштів на рахунку, перемикання тарифних планів, підключення або відімкнення послуг тощо. Сервісна служба спілкується голосом із застосуванням технологій синтезу мовлення. Випущено чимало дитячих іграшок, які “розмовляють” людським голосом, у яких застосовано найпростіші синтезатори мовлення. Синтезатори мови використовують у різних голосових системах попередження, що встановлюються в автомобілях та літаках. Такі системи дають змогу привернути увагу людини до виникнення тієї чи іншої критичної ситуації. У наш час актуальною є система оповіщення повітряної тривоги. Також розроблено чимало комп'ютерних програм, що здатні читати вголос вміст текстових файлів або текст, розташований у вікнах додатків. Ці системи можуть виявитися корисними для тих, у кого поганий або повністю відсутній зір [14]. На рис. 1 можна побачити, як комп'ютер перетворює електронний текст на слова, які озвучуються.

Модель компілятивного синтезу передбачає синтез мови, який полягає у складанні записаних зразків окремих звуків, які вимовляє диктор. Використовуючи цю модель, формують базу даних звукових фрагментів, з яких надалі синтезується мова. Створюючи звукові WAV-файли з текстовими повідомленнями, можна озвучити процеси з операційною системою Microsoft Windows та багато її програм, таких як поштові програми, інструментальні засоби розробки тощо [10]. Модель компілятивного синтезу можна використовувати переважно лише у найпростіших випадках, коли синтезатор повинен вимовляти порівняно невеликий і наперед відомий набір фраз. У такому разі забезпечується доволі висока якість мови. Проте це доцільно, якщо для синтезу використовується природна людська мова.

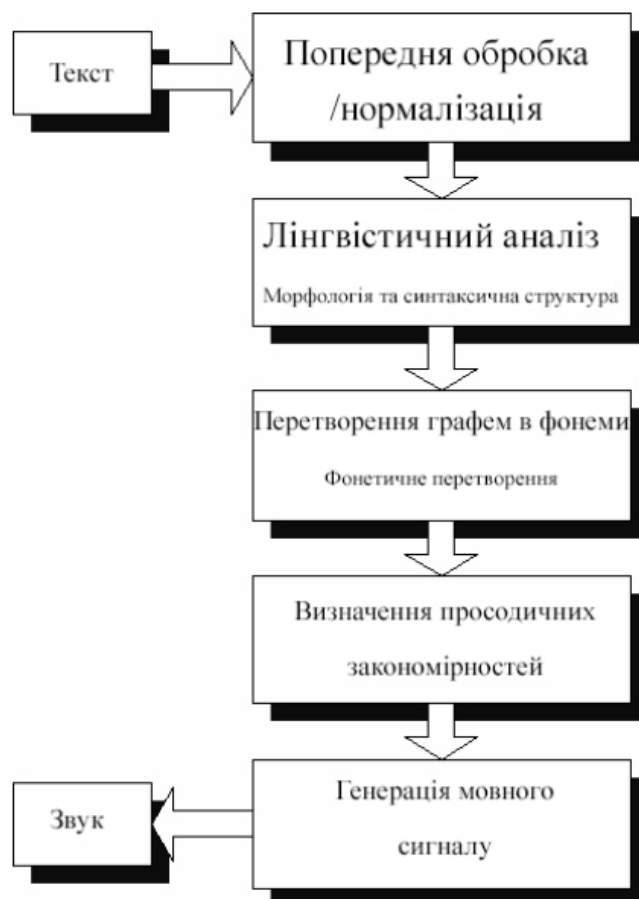


Рис. 1. Принцип синтезу мовлення

Для дослідження процесу озвучення текстової інформації формується форматно-голосова модель, яка полягає у моделюванні мовлення людини. Ця модель може бути реалізована із застосуванням нейронних мереж та допускає самонавчання. На жаль, через складність точного моделювання особливостей мовного тракту, а також урахування інтонаційної модуляції мови форматно-голосова модель забезпечує порівняно низьку точність синтезування звуків мови. Разом з тим, сучасні програми синтезу мови, побудовані з використанням цієї моделі, синтезують цілком розбірливу мову, інколи їх застосовують [11]. Зауважимо, що для систем голосового попередження про виникнення аварійних ситуацій краще використовувати модель компілятивного синтезу, оскільки розбірливість мови в таких системах найважливіша. Що ж до “побутових” синтезаторів мови, то в них можна успішно застосовувати і форматно-голосову модель. Під час формування моделі використано відомості про артикуляційний апарат людини, а також дані фонетики та лінгвістики. Як вихідний сигнал застосовано комбінацію голосового джерела і генератора шуму. Переривник та резонансний пристрій моделює роботу артикуляційного апарату. Внаслідок цього моделювання утворюється випромінюваний звук мови. Для досягнення компромісу між якістю моделі та її складністю вибрано такі основні параметри досліджуваної системи: частота основного тону; частота шуму; кількість формант; центральна частота кожної форманти; внесок кожної форманти; частота основного тону визначає висоту голосу.

Утворення звуку – доволі складний процес, що залежить від багатьох факторів, таких як тиск і швидкість повітряного струменя, геометрична форма повітряного тракту, акустичні властивості матеріалу тощо. Щоб фізично повністю змоделювати звук мовлення, необхідно створити точну модель мовного апарату людини, а це дуже складно. Як альтернативу використовують білий шум, спектр якого розподілений за деяким законом (наприклад, Гаусса) щодо деякої центральної частоти.

Закон розподілу підбирають експериментально, а частотою шуму є згадана вище центральна частота. Кількість активних формант, що беруть участь в утворенні мови, вибирають експериментально, причому орієнтовне значення – чотири. Оскільки форманта є резонансом у мовному тракті, у неї є частота резонансу й огинаюча. Вигляд огинаючої також визначається експериментально, у першому наближенні це гауссовий розподіл. Внесок кожної форманти визначає, наскільки істотно форманта впливає на основний сигнал. Всі наведені вище параметри, крім кількості формант, змінюються під час утворення мови для отримання різних звуків, хоча для якіснішого синтезу мови необхідно формувати детальнішу модель, щоб синтезовані звуки були розбірливі.

Метод, використовуваний для генерації звуків мови на основі моделі артикуляційного апарату людини, називають синтезом артикуляції. Він націлений на імітацію мовного тракту одним або декількома способами та пропонує спосіб отримати уявлення про розвиток мови та вивчити фонетику [10]. Коартикуляція є природним ефектом у такій моделі, і теоретично має бути можливо правильно розглядати властивості голосового джерела, зв'язок артикуляційного апарату з голосовими зв'язками, а також те, як підзв'язувальна система, носовий тракт та порожнини носових пазух впливають на формування людської мови через цю модель. Артикуляційний синтез зазвичай має два окремі компоненти: голосовий тракт, який розділений на кілька субкомпонентів, і відповідні області поперечного перерізу, що використовують параметрично для відображення характеристик голосових зв'язок. В акустичній моделі електрична аналогова лінія передавання апроксимує кожне поле поперечного перерізу. Моделювання мовного тракту схильне до змін, що виникають у функціях області щодо часу. Цільова конфігурація, призначена кожному звуку, визначає темп руху голосового тракту. За правильної конструкції синтезатор артикуляції може відтворювати всі відповідні ефекти під час розроблення фрикативних і вибухових звуків і моделювання переходів коартикуляції, щоб відтворити процеси, залучені до створення реальної мови. У середині 1970-х років у Haskins Laboratories Філіп Рубін, Том Баєр і Пол Мермельштейн створили перший синтезатор артикуляції, зазвичай використовуваний для лабораторних експериментів [13].

Синтез синусоїдального сигналу або синусоїдального голосу – це метод синтезу мови із заміною формантів (видних енергетичних смуг) свистками чистого тону. Філіп Рубін створив перше програмне забезпечення синусоїдального синтезу (SWS) для автоматизованого виробництва стимулів для перцепційних експериментів у Haskins Laboratories у 1970-ті роки [12]. Синусоїдальна мова – це своєрідне явище, за якого деякі особливості мови передаються невеликій кількості синусоїд, зібраних разом, на які вони зовсім не схожі у багатьох аспектах. Високої розбірливості досягають за допомогою трьох синусоїд, що відстежують частоту та амплітуду перших трьох мовних формант. Для дослідження форматно-голосової моделі синтезу мови створено інструмент *Модель синтезу*, в якому ручним завданням параметрів можна синтезувати практично будь-який голосний або шиплячий звук. Алгоритм синтезу мови такий: рівні виходів нейронів ефекторного шару нейромережі за допомогою картки ефекторів перетворюються на значення вибраних параметрів моделі синтезу. Картка ефекторів визначає відповідність між кожним нейроном ефекторного шару та конкретним параметром моделі синтезу, а також граничні значення кожного параметра. Кількість ефекторів і кількість параметрів моделі можуть бути неоднаковими. Якщо параметр не відповідає жодному ефектору, використовується деяке фіксоване значення (за замовчуванням). Далі за поточним станом моделі синтезується сигнал у просторі частот: генерується лінійка частот, що представляють голосове джерело. Цю лінійку частот накладається формантна структура (резонанси). Для синтезу шуму використовується генератор випадкової амплітуди та фази [4]. На останньому етапі виконується зворотне перетворення Фур'є для отримання звуків. Під час навчання системи формувалися нейронні ансамблі кожного звуку з навчальної послідовності А, В, С, D. Потім відбувалося навчання синтезу. У результаті в ефекторному шарі встановилися правильні зв'язки із символьним шаром. Система успішно навчилася синтезу – синтезовані звуки точно відповідають тонам із навчальної вибірки.

Технологія Text-to-Speech (TTS), відома на комп'ютерному ринку вже понад 15 років, зазвичай використовується в додатках, які повинні відтворювати велику кількість різних текстів [10]. Основною рисою, що виділяє TTS із голосових програм, розроблених раніше, є здатність вимовляти слова на основі фонетичних правил та заздалегідь озвученого або синтезованого машиною набору звуків. Наближено процес синтезу мови можна уявити як склеювання за правилами фонетики заздалегідь озвучених фрагментів мови (дифтонгів чи довших фрагментів) у слова і потім – у речення. З цього випливають переваги технології TTS [19–30]:

- можливість озвучення будь-яких слів цієї мови, які або тільки з'явилися в ужитку, або ще ніколи не існували;
- низькі вимоги до оперативної пам'яті комп'ютера, у якій містяться лише озвучені фонемі, а не цілі словники, як це реалізовано за іншими технологіями синтезу мови;
- швидший процес, оскільки синтез мови здійснюється швидше, ніж пошук у величезній основі заздалегідь озвучених слів (ця перевага проявляється особливо тоді, коли необхідно відтворити велику кількість різноманітних слів);
- простота виділення наголосів та інтонацій у синтезованих словах;
- можливість змінити темп читання, не порушуючи тембру голосу.

Синтез мовлення ґрунтується на знанні багатьох наукових дисциплін: психології, фізіології, лінгвістики людини, комп'ютерних технологій [21–27]. Необхідний аналіз структури речення, в результаті якого визначається вимова окремих слів, інтонація та оптимальний ритм синтезованої мови (з урахуванням синтаксису та семантики). Повинні правильно вимовлятися власні імена, телефони, поштові адреси та інші специфічні елементи генерованого тексту, без яких немислимий сучасний інтернет. Не так давно з'явилася нова розробка – Visual TTS, яка полягає у синхронізації генерованої мови з моделюванням міміки обличчя людини, яка говорить. Реалістичний рух губ поліпшує не тільки сприйняття синтезованої мови, а й розбірливість вимовленого тексту.

Технологія Microsoft Text-to-Speech призначена для синтезу мови з комп'ютерних текстових файлів, які містять інформацію, отриману механізмами розпізнавання людської мови. Вихідний сигнал може бути згенерований для двох різних випадків – читання телефоном (частота дискретизації 8 кГц) або відтворення через звукову карту ПК з частотою дискретизації близько 22 кГц. Передбачено також можливість збереження згенерованого мовлення у різноманітних звукових форматах. Корпорація створила програмний інтерфейс для роботи зі звуком – SAPI 4.0 (Speech Application Programming Interface) і доповнює його набором інструментів та утиліт для швидкої побудови мовних програм. У нього входять функції розпізнавання мови ASR (Automated Speech Recognition) та технологія TTS. Триває розроблення нового інтерфейсу SAPI 5.0, який, як оголосила компанія, є повністю оновленою версією. Для розробників мовних програм запропоновано кілька варіантів SAPI SDK. Найповніший з них – SAPI Speech SDK 4.0a Suit – містить докладний опис інтерфейсних бібліотек SAPI, документацію, приклади вихідного коду та додатків, утиліти для тестування, а також мовний інструментарій Microsoft Speech engines. В архіві цей набір займає понад 39 Мб. Але якщо відмовитися від Microsoft Speech engines, отримаємо набір SAPI Speech SDK 4.0a, що займає лише 7,9 Мб і доступний на сайті компанії. Цей пакет позбавляє можливості використовувати мову в додатках, а дає змогу створити програмну оболонку управління нею. У мовній технології TTS від Microsoft можна вибрати три різні типи голосів: “Майк”, “Мері” та “Сем”. Мовний інструментарій підтримує операційні системи Windows 95, 98, NT 4.0 або 2000, потребує звукової карти, процесора не нижче ніж Pentium, ОЗУ від 16 Мб.

Уже створено чимало сервісів, що здатні озвучувати текст [19–30]. Ці сервіси існують у вигляді вебсторінок і додатків на мобільні телефони та комп'ютери з різними платформами та на інші пристрої, здатні синтезувати мову. Однак у всіх сервісів є переваги і недоліки, такі як: зручність інтерфейсу, методи синтезу, ліміт на введення тексту, правильність озвучування, можливість вибору голосу та збереження інформації, що впливають на користування цими сервісами. Проаналізовано такі системи озвучення тексту, як ПЗ Notevibes, яке допомагає пацієнтам із дислексією та людям з іншими порушеннями навчання. Ця програма надає користувачам допомогу в мовленні, читанні та багато іншого. Для тих, кому

потрібна високоякісна система, програма здатна конвертувати текст у природне звучання. Існують безкоштовні та платні версії, якщо потрібні додаткові функції для завершення ваших проєктів. Цей сервіс відомий високим лімітом символів, пропонує понад 5000 символів безкоштовно. Серед особливостей цієї програми можна виділити 24 голоси; забезпечено кілька природних звучань, які не є роботизованими. Для початківців легко конвертувати запис у файли MP3. Але для справжніх шанувальників передбачена опція .WAV для отримання високоякісного звуку. Це ПЗ може швидко перетворити будь-який текст на аудіофайл, залишаючи файл з чітким звуком для подальшого користування. Але є деякі користувачі, які потребують функції комерційного рівня від Notevibes. Заплативши 90 доларів за комерційний пакет, можна отримати понад 1000000 символів та 18 різних мов. Серед переваг цього сервісу можна виділити 18 різних мов для озвучування, налаштування голосу, простий інтерфейс. Недоліки – дорогий комерційний пакет вартістю 90 доларів за місяць використання, проблеми зі швидкістю та неможливість читати файли формату .DOC та .PDF.

Natural Text Reader – програма, яка перетворює текст на мову. Вона пропонує широкий спектр першокласних голосів, кожен з яких звучить, як людський, і не має тієї мінливості, яку багато людей очікують почути у програмі синтезатора мови. Оскільки всі збережені аудіофайли доступні у хмарі, працюючи з Natural Text Reader, також можна без проблем зберегти свою роботу в інтернеті та отримати доступ до неї з іншого місця. Однак, у разі потреби, завжди можна зберегти свою роботу локально у вигляді аудіофайлу. Natural Text Reader надає вибір із більш ніж 50 голосів, і у них буде доступно 18 різних мов, кожна з яких звучить правдоподібно та реалістично. Майже кожною мовою є принаймні два голоси на вибір, як чоловічі, так і жіночі. Вони також мають корисний редактор вимови, який дає змогу змінювати швидкість, акцент, висоту та багато іншого. Natural Text Reader має три доступні плани, від базового до професійного, і кожен з них спеціалізований для особистого або комерційного використання. Перевагами цього сервісу є зручний інтерфейс, реалістичні голоси, можливість працювати локально чи у хмарі. Основним недоліком цього сервісу є обмеження у п'ять озвучень на день для безкоштовних користувачів.

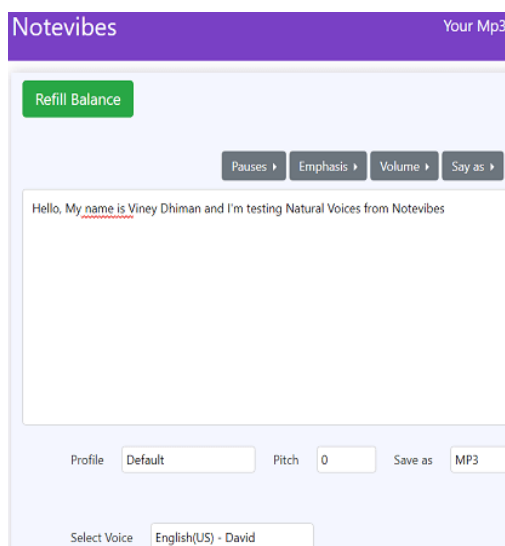


Рис. 2. Головна сторінка Notevibes

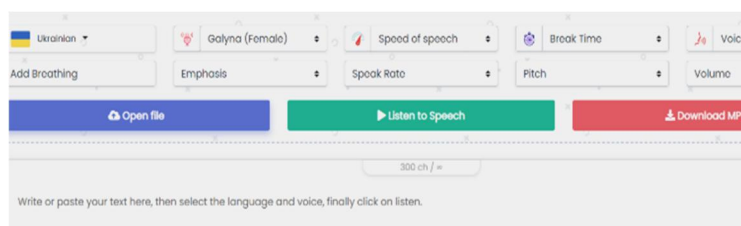


Рис. 3. Головна сторінка Natural Text Reader

Сервіс Vocalware трохи відрізняється від інших сервісів на ринку. Хоча багато служб перетворення тексту на мову орієнтовані на окремих користувачів, Vocalware можуть використовувати компанії, яким потрібні функції перетворення тексту на мову. Vocalware може надати необхідні API для декількох користувачів. Це корисно, коли не один користувач збирається конвертувати текст у мову, а потрібно, щоб клієнти могли це робити паралельно. Серед переваг цього сервісу широкий вибір голосів і мов, два різних види API, безкоштовна пробна версія на 15 днів. Недоліки цього

сервісу – він придатний більше для бізнесу, а не для індивідуальних користувачів, та обмеження тексту в безкоштовній версії.

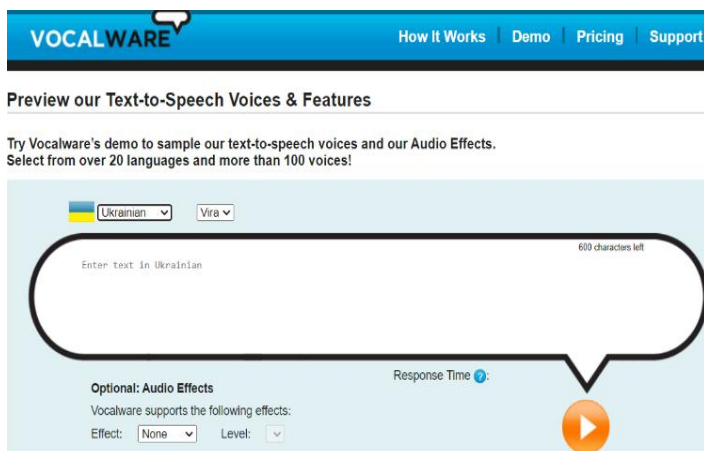


Рис. 4. Головна сторінка Vocalware

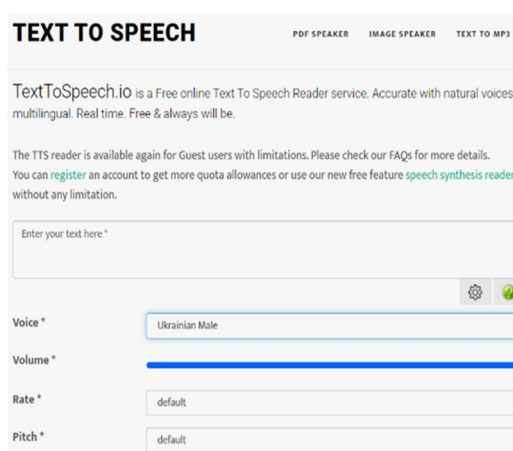


Рис. 5. Головна сторінка TextToSpeech.io

Якщо простота – це те, що шукають у службі перетворення тексту на мову, і є множина функцій, які практично не використовуються, TextToSpeech.io – один з фаворитів. Цей сайт – простий та зрозумілий. У цьому сервісі є широкий спектр голосів, доступних на декількох міжнародних мовах, і все ж запропоновано деякі варіанти, щоб змінити мову на свій смак. Хоча цей сервіс для перетворення тексту на мову порівняно простий, він підтримує мови від англійської до грецької, що зручно для багатьох користувачів. Є також деякі функції, які можуть допомогти змінити відтворення того, що вводиться, і зробити його звучання реалістичнішим. По-перше, можна змінити гучність запису так, щоб він звучав як пошепки, так і на повний голос. Також можна змінити висоту голосу із низького на високий. Перевагами цього сервісу є налаштування для зміни голосу, простота у використанні, безкоштовна версія, недоліком – недопрацьована функція зміни гучності, що не міняє тон.

Сервіс Balabolka призначений для озвучування текстових файлів. Для того, щоб озвучити текст, можна використати будь-які синтезатори мови, встановлені на комп'ютері. Додаток здатний озвучити вміст буфера обміну, показати текст, який міститься в документах, змінити колір і налаштування шрифту, читати вголос текст, набраний на клавіатурі, перевіряти орфографію, в кількох файлів меншого розміру шукати омографи. Щоб уникнути зупинок під час читання слів, додаток здатний видалити всі знаки перенесення у кінці рядків. Перевагами цього додатка є можливість відкриття деяких вкладок паралельно, інструмент “лупа”, що збільшує текст для людей з поганим зором, можливість збереження перетвореного на голос тексту. Серед мінусів можна виділити незручний інтерфейс та мало налаштувань для зручного читання тексту на екрані.

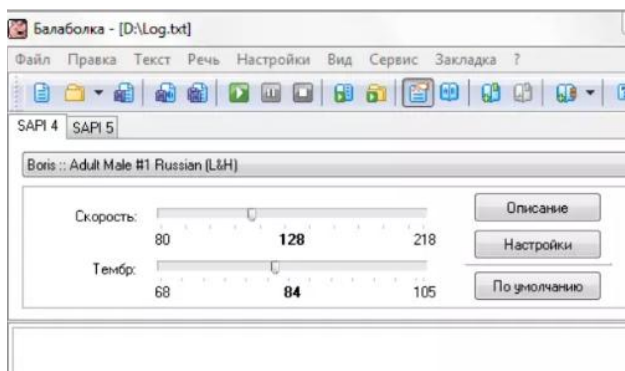


Рис. 6. Головне вікно Balabolka

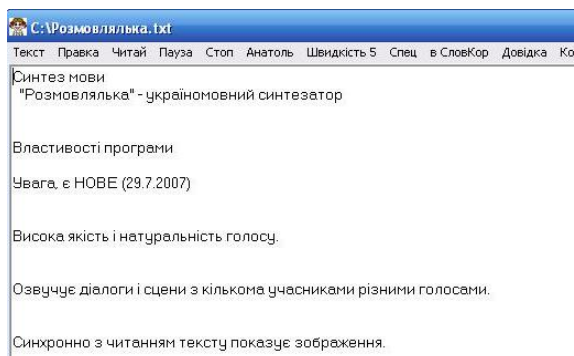


Рис. 7. Головне вікно Розмовлялка

Розмовлялька – безкоштовний комп’ютерний додаток для озвучення тексту українською мовою. Від інших додатків розмовлялька відрізняється підтримкою тільки української мови, можливістю якісно озвучити текст декількома голосами. Перевагами цього додатка є можливість використання різних голосів для озвучування тексту, хороша якість озвучування, великий словник наголосів, меню українською мовою. Серед недоліків можна виділити застарілий інтерфейс та підтримку тільки української мови.

Порівняння переваг та недоліків розглянутих систем наведено у таблиці.

Переваги та недоліки розглянутих систем

Назва сервісу	Підтримка тільки української	Переваги	Недоліки
Notevibes	Ні	18 різних мов для озвучування, налаштування голосу, простий інтерфейс	Дорогий комерційний пакет, проблеми зі швидкістю та неможливість читати файли формату .DOC та .PDF.
Natural Text Reader	Ні	Зручний інтерфейс, реалістичні голоси, можливість працювати локально чи у хмарі	Обмеження – п’ять озвучень на день для безкоштовних користувачів
Vocalware	Ні	Широкий вибір голосів і мов, два різні види API, безкоштовна пробна версія на 15 днів	Підходить більше для бізнесу, а не для індивідуальних користувачів, обмеження тексту в безкоштовній версії
TextToSpeech.io	Ні	Налаштування для зміни голосу, простота у використанні, безкоштовна версія	Недопрацьована функція зміни гучності, що не змінює тон
Balabolka	Ні	Можливість відкриття деяких вкладок паралельно, інструмент “лупа”, що збільшує текст, можливість збереження перетвореного на голос тексту	Незручний інтерфейс та мало налаштувань для зручного читання тексту на екрані
Розмовлялька	Так	Використання різних голосів для озвучування тексту, хороша якість озвучування, великий словник наголосів, меню українською мовою	Застарілий інтерфейс та підтримка тільки української мови

Формулювання цілі статті та постановка задачі

Дерево цілей – це модель, що відображає взаємозв’язок цілей та засобів їх досягнення. На нижньому рівні цілі упорядковуються та сприяють досягненню цілей вищого рівня. Головні та важливі цілі містяться на вершині дерева [15]. Поділ головної цілі на дрібні полегшує її досягнення, тому її поділяють на дрібніші. Щоб побудувати дерево цілей, потрібно завжди починати з верхньої частини дерева, тобто зі стратегічних орієнтирів. Тут потрібно відповісти на запитання: “Що ви повинні отримати в результаті?” Таких високорівневих орієнтирів може бути декілька або лише один – залежно від ваших завдань. Іноді стратегічних цілей може бути декілька, але кожна з них є настільки широкою, що краще зробити окреме дерево для кожної з таких цілей. В результаті виконання цього етапу повинна вийти “верхівка” дерева, в якій закладені основні орієнтири вашого руху. Це місія. Це те, для чого все інше здійснюватиметься. Є декілька принципів, яких потрібно дотримуватися, формуючи дерево цілей:

- Принцип декомпозиції. Суть цього принципу полягає у тому, що спочатку визначають “високорівневі” цілі, лише потім ці великі цілі поділяють на дрібніші, конкретніші та деталізовані. Помилкою спочатку буде зібрати декілька маленьких цілей і спробувати приписати їм щось єдине. Потрібно рухатися зверху – вниз, а не знизу – вгору.

- Конкретність цілей. Потрібно добитись, щоб кожна мета була достатньо конкретною. Це означає, що в неї повинні бути якісь показники, які свідчатимуть про успішність або неспішність

досягнення конкретної мети. Простіше кажучи, потрібно під час опису кожної мети відповісти на запитання: “Як ви зрозумієте, що мети досягнуто?”

- Регулярне оновлення. Сучасний світ змінюється дуже швидко. Щоб стратегія залишалася актуальною, необхідно її оновлювати. Дерево цілей має стати робочим інструментом, на який орієнтуються, і за яким “доглядають” – тобто оновлюють, вносять нові дані, видаляють цілі, які вже досягнуті або не є актуальними.

Дерево цілей інформаційної системи озвучення україномовного тексту на основі методів NLP та машинного навчання подано на рис. 8.

Головною ціллю є “Створення інформаційної системи озвучення тексту на основі методів NLP та машинного навчання”. Для досягнення цієї цілі її розділено на три підцілі: “Збір даних для озвучення”, “Проектування”, “Озвучення тексту”. Кожну із цих цілей необхідно теж розділити на підцілі. Щоб здійснити “Збір даних для озвучення”, потрібно проаналізувати технології та вже відомі рішення, щоб потім їх врахувати під час проектування системи.

Підціль “Проектування” передбачає проектування дизайну та проектування інтерфейсу. Дизайн – перше, на що звертає увагу користувач у сервісі, тому він має бути простим та зручним. Проектування інтерфейсу потрібне, щоб будь-який користувач міг легко знайти необхідні йому інструменти та озвучити текст.



Рис. 8. Дерево цілей інформаційної системи озвучення україномовного тексту

Підціль “Озвучення тексту” містить такі підцілі, як: “Вибрати метод озвучення”, “Вибрати голос”, для коректного та якісного озвучування тексту. Завдяки всім цим даним сформовано загальну картину, що відображає створення інформаційної системи для озвучування україномовного тексту. Як альтернативи вибрано дві інформаційні системи, які класифіковано за використанням інформації: інформаційна система озвучення тексту – надає можливість користувачу озвучити будь-який текст на будь-якій мові; інформаційна система озвучення україномовного тексту – дає змогу озвучити будь-який текст тільки українською мовою.

Виклад основного матеріалу

Проаналізувавши дерево цілей, ми вирішили вибрати інформаційну систему озвучення україномовного тексту, тому що саме озвучення тексту українською мовою є основною метою проектування інформаційної системи. Контекстна діаграма, також названа діаграмою потоку даних рівня 0, є найвищим рівнем діаграми потоку даних. Це інструмент, популярний серед бізнес-аналітиків, які використовують його, щоб зрозуміти деталі та межі системи, яка буде розроблена в

проекті [16]. Контекстна діаграма відображає систему, що розглядається, як єдиний процес високого рівня, а потім показує зв'язки система з іншими зовнішніми об'єктами (системами, організаційними групами, зовнішніми сховищами даних тощо). Інша назва контекстної діаграми – діаграма потоку даних на рівні контексту (DFD) або діаграма потоку даних рівня 0 (IDEF0). Оскільки контекстна діаграма є спеціалізованою версією діаграми потоку даних, потрібно розуміти діаграми потоку даних. Діаграма потоку даних (DFD) – це графічна візуалізація руху даних через інформаційну систему. DFD є одним із трьох основних компонентів методу аналізу та проектування структурованих систем (SSADM). DFD орієнтований на процес і зображає чотири основні компоненти [18]: процеси; зовнішні сутності; сховища даних; потоки даних.

Контекстна діаграма IDEF0 є початковою діаграмою дерева вузлів, що описує модельовану систему в її найповнішій та скороченій формах [17]. Створена контекстна діаграма IDEF0 зображена на рис. 9, в неї входять головний процес “Здійснити озвучення україномовного тексту”, текст, технології TTS, методи синтезу, система озвучення та пристрій для озвучення.

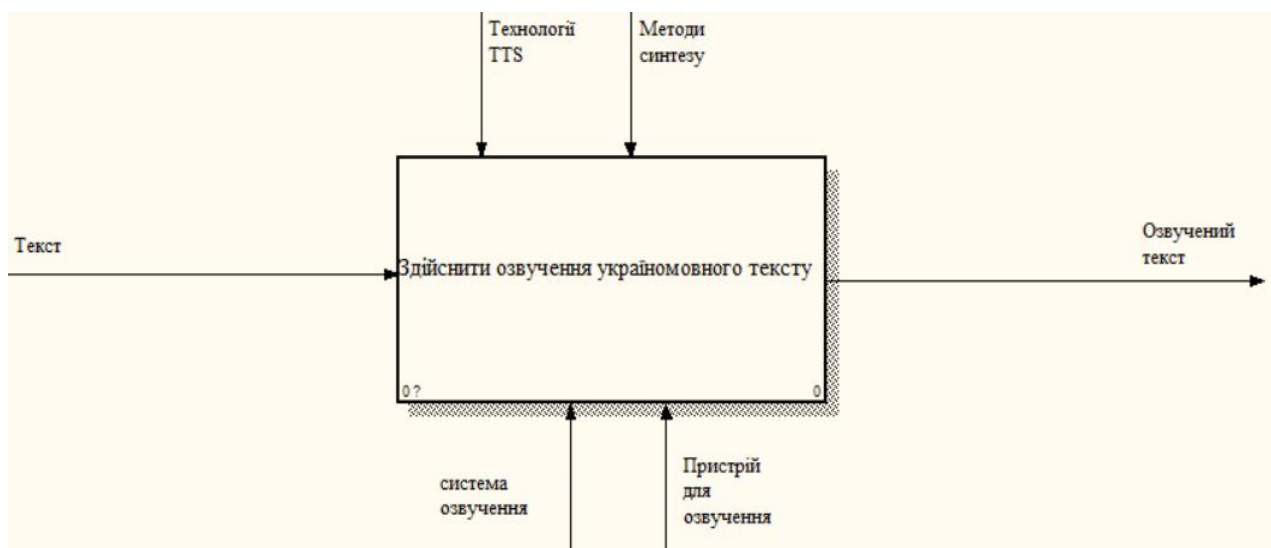


Рис. 9. Контекстна діаграма “Здійснити озвучення україномовного тексту”

Наступним етапом є декомпозиція головного процесу, де функції системи розглянуто детальніше. В результаті побудовано діаграму деталізації першого рівня. Діаграма розділена на чотири процеси: нормалізація, фонетичне перетворення, генерація мовного сигналу, звук. Зобразимо ці функції на діаграмі деталізації, а також визначимо та зобразимо сховища даних і потоки даних між функціями, сховищами та контекстною діаграмою. Результат побудови діаграми деталізації першого рівня подано на рис. 10. Для кожного з процесів: вхідними даними є: текст; керуючі дані: технології TTS та методи синтезу; механізми: спроектована система озвучення та пристрій для озвучення; вихідні дані – озвучений текст.

Процес “Нормалізація” (рис. 11) декомпоновано на дві менші бізнес-функції: аналіз тексту; перетворення усіх небуквених символів. Для кожного із цих двох бізнес-процесів керуючими даними є “Технології TTS” та “Методи синтезу”. Механізми процесів – “Спроектована система озвучення” та “Пристрій для озвучення”; вихідні дані – “Нормалізований текст”.

Процес “Фонетичне перетворення” (рис. 12) декомпоновано на три менші бізнес-функції: проаналізований текст; перетворення графем на фонемі; генерація мови. Для кожного із цих трьох бізнес-процесів керуючими даними є “Технології TTS” та “Методи синтезу”. Механізмами процесів є “Спроектована система озвучення” та “Пристрій для озвучення”. Вихідні дані – “Фонетично перетворений текст”.

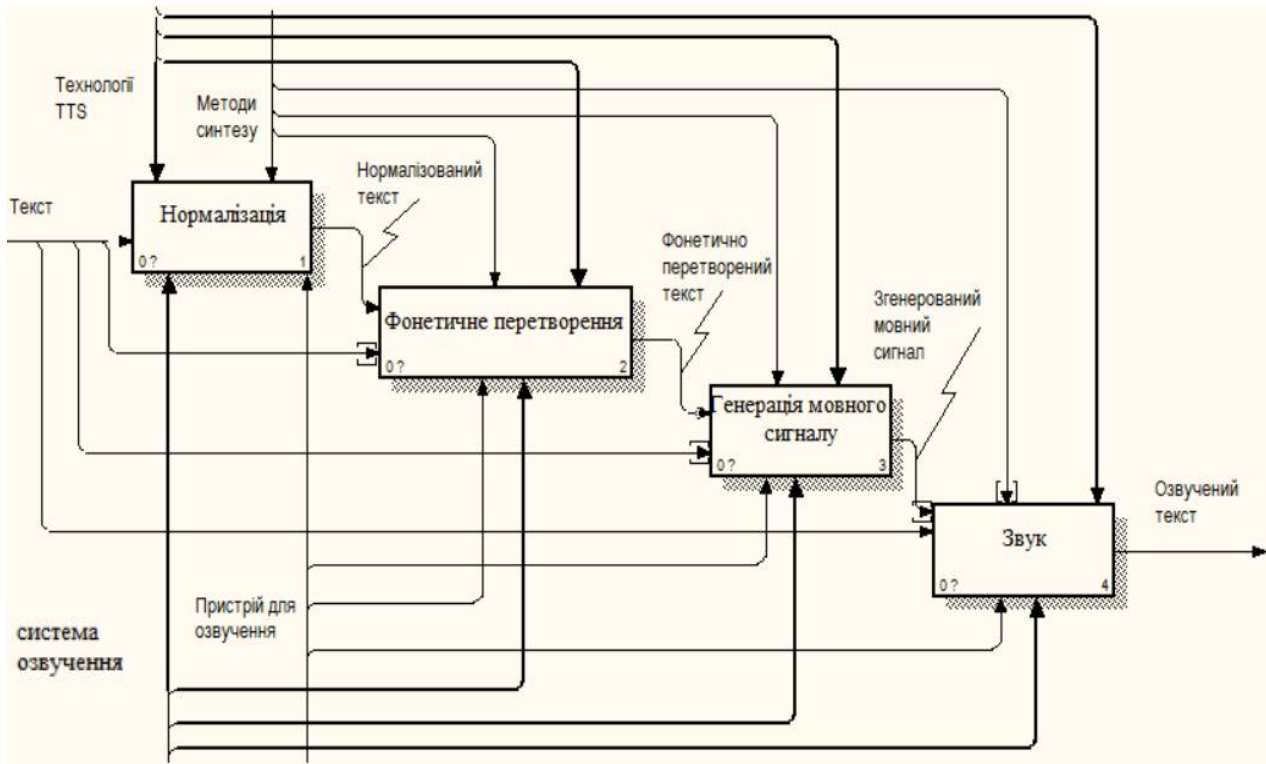


Рис. 10. Діаграма декомпозиції “Здійснити озвучення українського тексту”

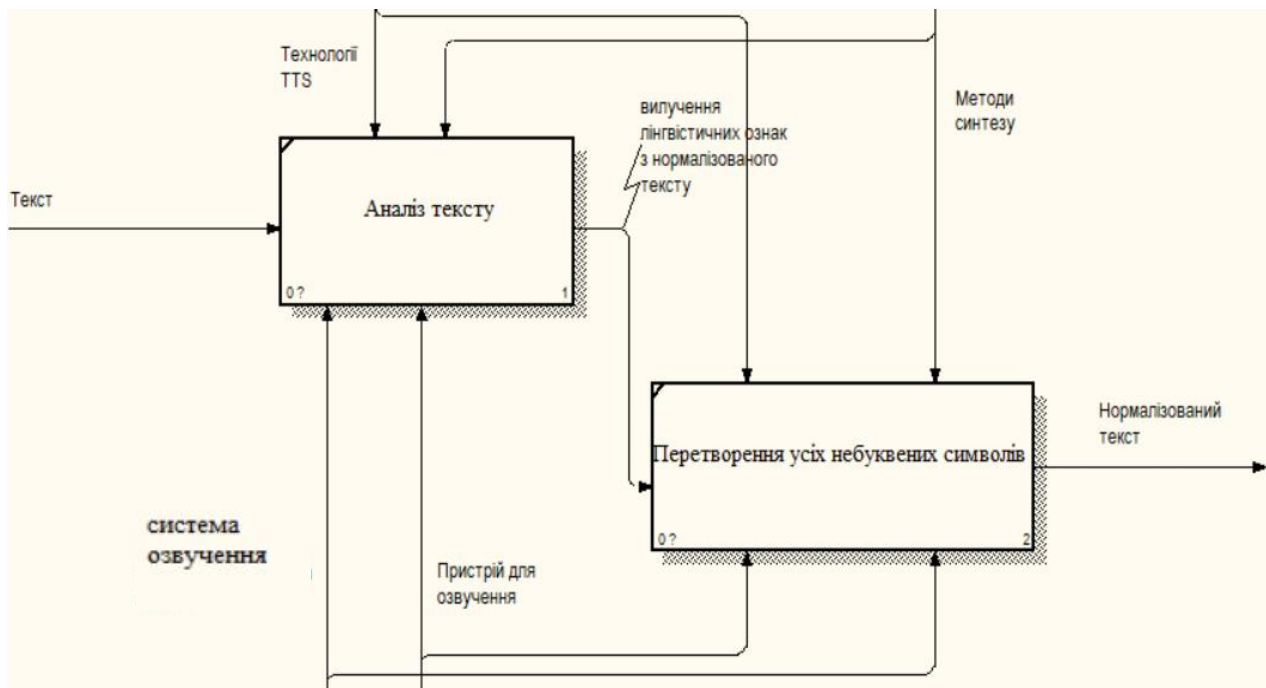


Рис. 11. Діаграма декомпозиції процесу “Нормалізація”

Процес “Генерація мовного сигналу” (рис. 13) декомпоновано на три менші бізнес-функції: опрацювання даних; визначення просодичних закономірностей; озвучений текст. Для бізнес-процесу “Озвучений текст” керуючими даними є “Технології TTS” та “Методи синтезу”. Механізмами усіх трьох процесів є “Спроектвана система озвучення” та “Пристрій для озвучення”, вихідними даними – “Згенерований мовний сигнал”.

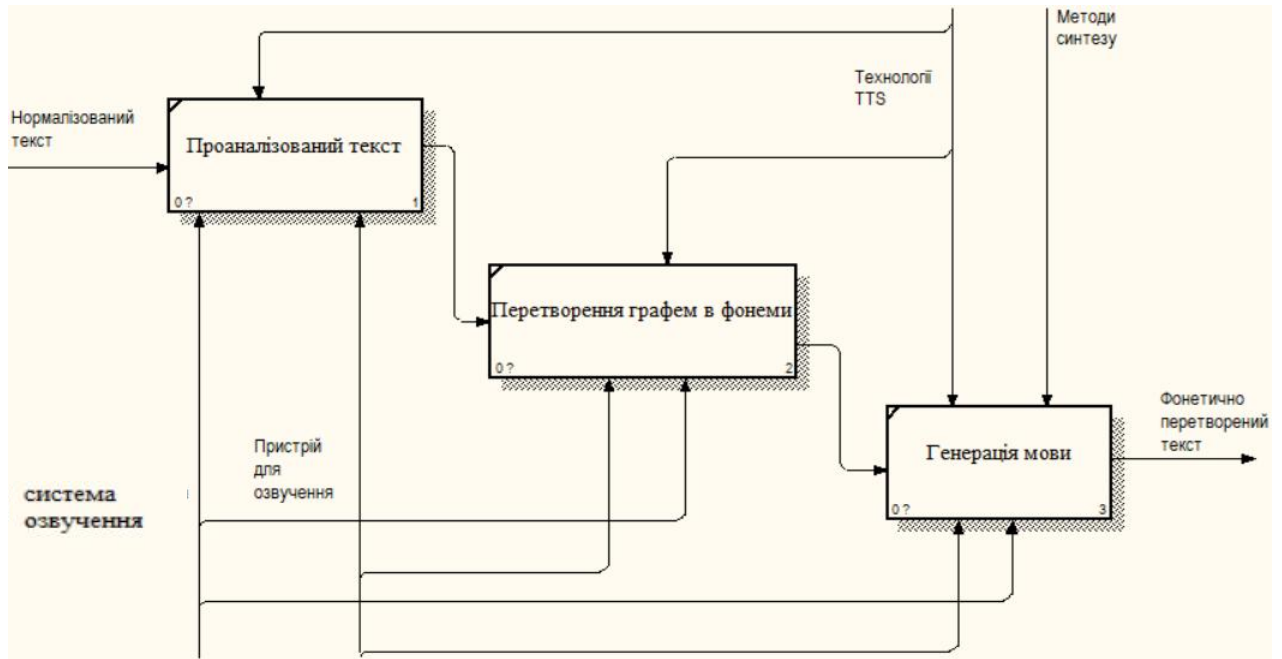


Рис. 12. Діаграма декомпозиції процесу “Фонетичне перетворення”

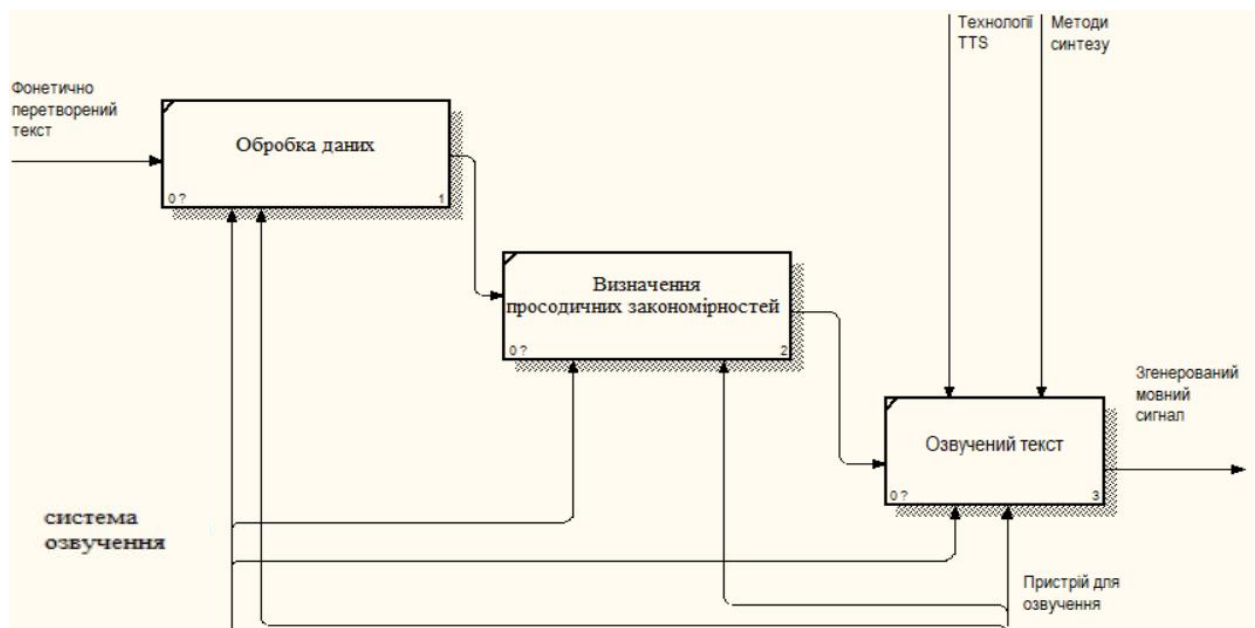


Рис. 13. Діаграма декомпозиції процесу “Генерація мовного сигналу”

Процес “Звук” (рис. 14) декомпоновано на дві менші бізнес-функції: введений текст; синтез мови. Для бізнес-процесу “Синтез мови” керуючими даними є “Технології TTS” та “Методи синтезу”. Механізмами усіх двох процесів є “Спроектвана система озвучення” та “Пристрій для озвучення”. Вихідними даними є “Озвучений текст”.

Діаграма вузлів користується звичним деревом ієрархій, де верхній вузол відповідає контекстній діаграмі, а нижній рівень – декомпозиції нащадків. Діаграма вузлів дерева відображає взаємозв’язок усіх головних діаграм і діаграм-нащадків у вигляді ієрархії блоків моделі, що дає змогу розглядати всю модель повністю. Ієрархія діаграм дерева починається з першого вузла, що також відомий як “коренева сутність”. У результаті з нього виходять так звані “сутності-нащадки”. Діаграма, зображена на рис. 15,

розпочинається з вузла “Здійснити озвучення україномовного тексту”, з гілками, які з’єднуються з вузлами, що показують взаємовиключні рішення або події. Після цього рішення або подія на цих діаграмах з’єднується з іншими вузлами. З цих вторинних вузлів здійснюється додаткове рішення або подія, що приводить до третього рівня вузлів, поки не буде зроблений висновок.

Діаграма взаємозв’язків сутностей (ERD) – це візуальне подання бази даних, що показує, як пов’язані елементи всередині [17]. Діаграма ER складається із двох типів об’єктів – сутностей і відносин. Сутність у цьому контексті – це компонент даних із набору даних, що відображається у вигляді фігури на полотні. Відносини між сутностями подано у вигляді рядків зі спеціальними закінченнями, які називають кардинальностями. Вони описують, як два елементи бази даних взаємодіють між собою. ER діаграму інформаційної системи озвучення україномовного тексту на основі методів NLP та машинного навчання зображено на рис. 16. Метод ієрархічного аналізу завдань полягає у виявленні та структурованій фіксації задач та їх підзадач, необхідних для досягнення цілей користувачів. Подальше опрацювання інтерфейсів відбувається з урахуванням можливості найефективнішого виконання цих завдань. Ієрархію задач системи озвучення україномовного тексту на основі методів NLP та машинного навчання наведено на рис. 15.

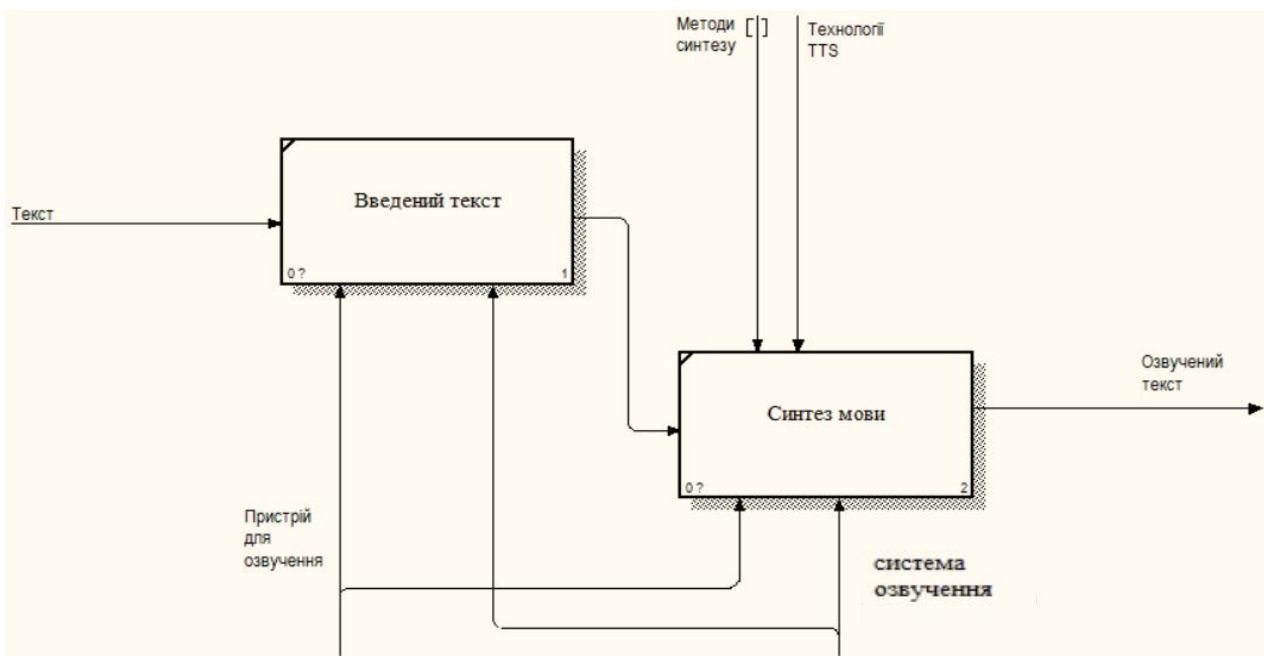


Рис. 14. Діаграма декомпозиції процесу “Звук”



Рис. 15. Діаграма вузлів “Здійснити озвучення україномовного тексту”

Проаналізувавши відомі рішення, ми вирішили виконати інформаційну систему озвучення україномовного тексту у вигляді комп'ютерного додатка для операційної системи Windows. Для реалізації такого програмного забезпечення у вигляді десктоп-дodatка використано такі програмні рішення: Visual Studio Community, C#, .NET6, WPF, Google Text-To-Speech.

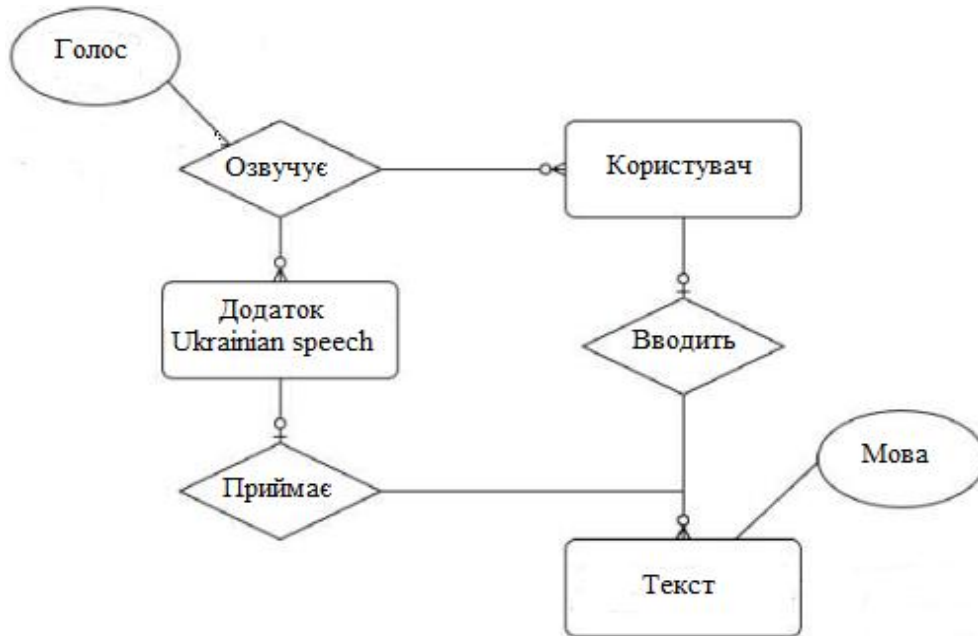


Рис. 16. ER діаграма інформаційної системи озвучення україномовного тексту

Visual Studio, також відома як Microsoft Visual Studio і VS – це середовище розроблення для Microsoft Windows. Це інструмент для написання комп'ютерних програм, вебсайтів, вебдодатків та вебсервісів. Він містить редактор коду, налагоджувач, інструмент розроблення графічного інтерфейсу та конструктор схем бази даних, а також підтримує більшість основних систем контролю редагування. Він доступний як у безкоштовній версії “Community”, так і в платній комерційній версії. Visual Studio підтримує такі мови програмування, як C, C++, C#, Visual Basic .NET, F#, Fossil, M, Python, HTML/XHTML/CSS, JavaScript. C# сьогодні є однією з найпопулярніших мов, яка стрімко розвивається та на яку великий попит у IT-галузі. Нині на ньому проєктують різні програми: від малих програм для десктопу до великих вебпорталів і вебсервісів, які щодня використовують мільйони користувачів. Перша версія мови була випущена разом із релізом Microsoft Visual Studio .NET на початку 2002 р. [1]. Актуальною версією мови є версія C# 10.0, випущена 8 листопада 2021 разом із релізом .NET 6. C# є об'єктно-орієнтованою мовою і в цьому плані схожий на Java та C++. Наприклад, C# містить функції, подібні до Java, такі як: поліморфізм, успадкування, навантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід вирішує завдання із побудови великих, але водночас гнучких, масштабованих і додатків, що розширюються [1]. Також C# продовжує активно розвиватися і з випуском кожної нової версії з'являється все більше нових та цікавих функцій. .NET – це фреймворк від Microsoft, який дає змогу використовувати ті самі простори імен, бібліотеки та API для різних мов. Найчастіше це чотири мови із сім'ї .NET: C#; Visual Basic; Visual C++; F# [6]. Створюючи програму однією з цих мов, спочатку ви підключаєте простір імен System. Якби не .NET, то для кожної з цих мов довелось б створювати окремий System, тобто порушувався б один із головних принципів програмування – DRY (англ. *Don't repeat yourself* – не повторюйся). Також якби не .NET, користувачам довелось б встановлювати середовище виконання програм на

кожній мові. Тобто, щоб запустити програму на Visual Basic, потрібно завантажити середовище виконання для Visual Basic [7]. Якщо програма написана на C#, то доведеться завантажувати середовище і для неї. Це дуже швидко займе все місце на комп'ютері дещо різними копіями тих самих бібліотек. Для програмістів це теж важливо, тому що дає можливість розвивати одне середовище, яке використовується одразу для чотирьох мов. Інакше звичайним розробникам доводилося б чекати, доки вийде нова версія бібліотек для їхньої мови. Менш популярні мови, наприклад F#, отримували оновлення набагато пізніше ніж C#. На рис. 17 зображено схему роботи .NET і C#.

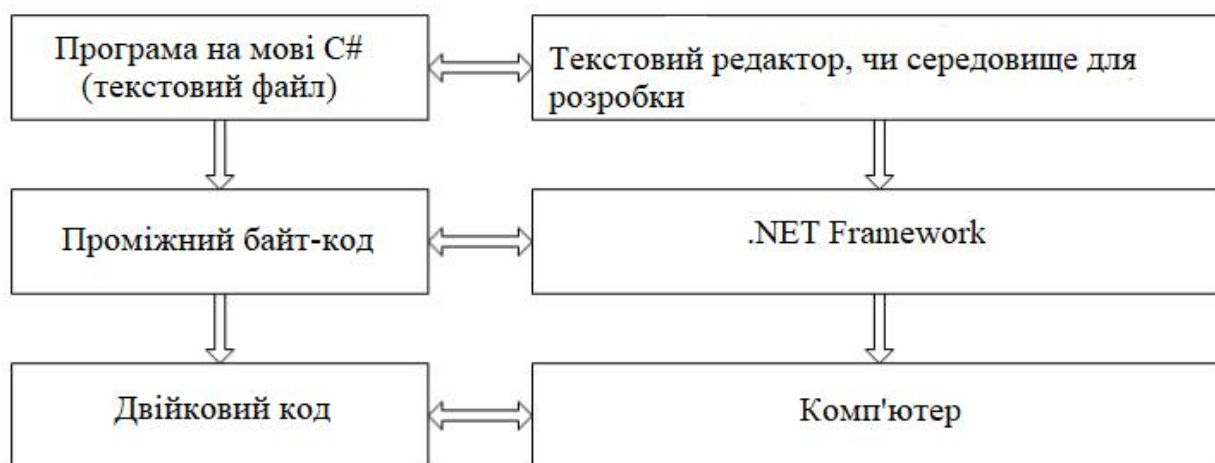


Рис. 17. Схема роботи .NET і C#.

WPF (Windows Presentation Foundation) – це частина фреймворку .NET і підсистема для проектування графічних інтерфейсів [8]. У попередніх фреймворках графічного інтерфейсу немає реального поділу між тим, як виглядає програма і як вона поводить. І графічний інтерфейс, і поведінка створені однією мовою, наприклад C# або VB.Net, що вимагало б від розробника більше зусиль для реалізації інтерфейсу користувача та поведінки, пов'язаної з ним. У WPF елементи інтерфейсу користувача розроблені в XAML, тоді як поведінка може бути реалізована на процедурних мовах, таких як C# і VB.Net. Тому дуже легко відокремити поведінку від коду конструктора. З XAML програмісти можуть працювати паралельно із дизайнерами. Розділення між графічним інтерфейсом користувача та його поведінкою дає змогу легко змінити зовнішній вигляд елемента керування за допомогою стилів і шаблонів. Серед переваг WPF можна виділити:

- Вебподібну модель компонування. Замість того, щоб фіксувати елементи управління на місці з певними координатами, WPF підтримує гнучкий потік, що розміщує елементи управління на основі вмісту. Результат – інтерфейс, який можна адаптувати для відображення високодинамічного вмісту або до різних мов.

- Багату модель малювання. Замість малювання пікселів у WPF ви маєте справу з примітивами – базовими фігурами, блоками тексту та іншими графічними інгредієнтами. Крім того, доступні такі нові засоби, як справді прозорі елементи управління, можливість укладання один на одного множини рівнів з різним ступенем прозорості, а також вбудована підтримка тривимірної графіки.

- Розвинену текстову модель. Після багатьох років нестандартного опрацювання текстів WPF нарешті надає Windows-додаткам можливість відображення розширеного стилізованого тексту в будь-якому місці інтерфейсу користувача. І якщо потрібно відображати значні обсяги тексту, для підвищення читабельності можна скористатися розвиненими засобами відображення документів, такими як перенесення, розбиття на колонки та вирівнювання.

- Анімацію. У WPF не потрібно використовувати таймер для того, щоб змусити форму перемалювати себе. Натомість доступна анімація – невід’ємна частина платформи. Вона визначається декларативними дескрипторами і WPF запускає її автоматично.

- Підтримку аудіо та відео. Колишні інструментальні набори для побудови інтерфейсів, такі як Windows Forms, були дуже обмежені в роботі з мультимедіа. Однак WPF передбачає підтримку відтворення будь-якого аудіо- або відеофайлу, підтримуваного програвачем Windows Media, даючи змогу відтворювати більше від одного медіафайлу одночасно. Що ще більше вражає – WPF надає у ваше розпорядження інструменти для інтеграції відеовмісту в решту інтерфейсу користувача, уможливаючи такі екзотичні трюки, як розміщення відеовікна на поверхні обертового тривимірного куба.

- Стили та шаблони. Стили дають змогу стандартизувати форматування та багаторазово використовувати його у всьому додатку. Шаблони дають можливість змінити спосіб відображення елементів, навіть таких основних, як кнопки. Побудова інтерфейсу з обкладинками ще ніколи не була такою простою.

- Команди. Більшості користувачів відомо, що не має значення, звідки вони ініціюють команду відкриття (Open) – через меню чи панель інструментів; кінцевий результат той самий. Тепер ця абстракція доступна коду – можна визначати команди програми в одному місці та прив’язувати їх до багатьох елементів керування.

- Декларативний інтерфейс користувача. Хоча можна конструювати вікно WPF у кодї, Visual Studio використовує інший підхід. Вміст кожного вікна серіалізується у вигляді XML-дескрипторів у документі XAML. Перевага полягає в тому, що інтерфейс користувача повністю відділяється від коду, і дизайнери графіки можуть використовувати професійні інструменти для редагування файлів XAML, покращуючи зовнішній вигляд всієї програми. (XAML – це скорочення від Extensible Application Markup Language (розширювана мова розмітки додатків).

- Програми на основі сторінок. Використовуючи WPF, можна будувати браузерподібні програми, які дають змогу переміщатися колекцією сторінок, оснащеною кнопками навігації “вперед” і “назад”. WPF автоматично опрацьовує всі складні деталі, такі як хронологія відвідування сторінок. Проект можна розгорнути у вигляді браузерної програми, яка виконується всередині Internet Explorer. Технологія WPF використовує багаторівневу архітектуру. На вершині додаток взаємодіє із високорівневим набором служб, які повністю написані на керованому кодї C#. Робота з трансляції об’єктів .NET у текстури та трикутники Direct3D відбувається “за лаштунками”, з використанням низькорівневого некерованого компонента під назвою milcore.dll. Бібліотека milcore.dll реалізована в некерованому кодї тому, що їй потрібна тісна інтеграція з Direct3D, а також для неї дуже важлива продуктивність. На рис. 18 показано рівні, на яких побудовано роботу програми WPF.

Ключові компоненти архітектури WPF такі:

- PresentationFramework.dll. Містить типи WPF верхнього рівня, такі елементи управління, як вікна, панелі та інші види. Реалізує і високорівневі абстракції, такі як стилі.

- PresentationCore.dll. Містить базові типи, такі як UIElement та Visual, від яких успадковані всі фігури та елементи керування. Якщо вам не потрібен повний рівень абстракції вікон та елементів керування, можете опуститись нижче, на цей рівень, і продовжувати користуватися перевагами механізму візуалізації WPF.

- WindowsBase.dll. Містить базові елементи, які потенційно можуть застосовуватися поза WPF, такі як DispatcherObject і DependencyObject, що підтримують механізм властивостей залежності.

- Milcore.dll. Є ядром системи візуалізації, WPF механізм якого перетворює візуальні елементи на трикутники та текстури, на які очікує Direct3D.

- WindowsCodecs.dll. Є низькорівневим API-інтерфейсом, який підтримує зображення.
- Direct3D. Є низькорівневим API-інтерфейсом, завдяки якому візуалізується вся графіка WPF.
- User32. Визначає, яке місце на екрані якій програмі належить.

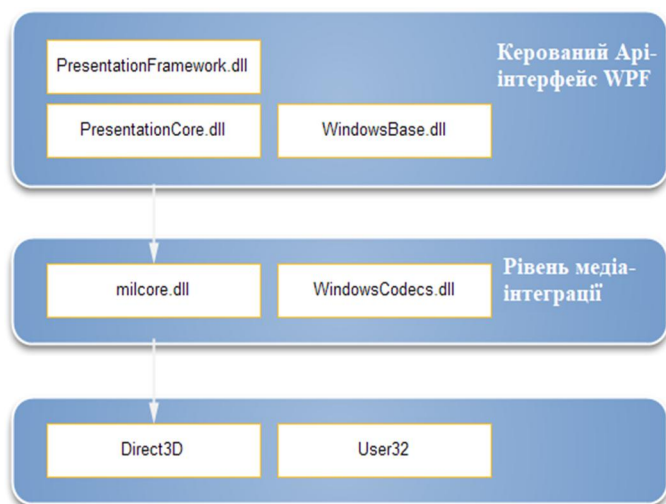


Рис. 18. Архітектура WPF

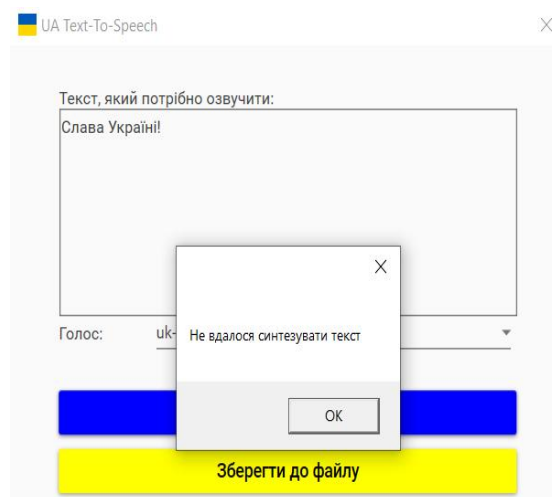


Рис. 19. Повідомлення “Не вдалося синтезувати текст”

Google Text-To-Speech – найпопулярніший API для синтезу мовлення. Його функції використовують такі сервіси, як Google Play Books, Google translate, Google Talkback та інші програми, що виконують синтез мовлення.

Створений додаток названо UA Reader. Озвучення тексту відбувається у вікні самого додатка. Щоб здійснити озвучення, користувач повинен ввести або додати будь-який потрібний йому текст українською мовою. За допомогою пакетів від Google для синтезу мови додаток завжди озвучує текст стабільно та швидко. Функціональним призначенням інформаційної системи озвучення українського тексту на основі методів NLP та машинного навчання є те, що користувач може самотужки та абсолютно безкоштовно озвучити будь-який необхідний йому українськомовний текст. Систему створено у вигляді десктоп-додатка для комп'ютерів з операційною системою Windows. Додаток створено, щоб будь-який користувач без особливих знань зміг здійснити озвучення необхідного йому тексту. Озвучення тексту відбувається завдяки вже наявним пакетам від Google для синтезу мови. Програма складається із головної форми та трьох методів. У головній формі міститься текстове поле, куди користувач вводить необхідний текст для озвучування. Якщо є проблеми з підключенням до мережі або якийсь збій у програмі, на екран видається повідомлення “Не вдалося синтезувати текст” (рис. 19). Якщо програма спрацювала коректно, після натискання на кнопку “Озвучити текст” програма озвучує введений текст.

Дерево рішень – математична модель, яка визначає процес прийняття рішень так, що буде відображене кожне можливе рішення, що передує, і подальші події або інші рішення та наслідки кожного кінцевого рішення. Дерево рішень складається з таких елементів: дуги, вузли рішень, вузли подій та кінцеві вузли (виходи). Метод дерева рішень застосовують у завданнях класифікації та прогнозування, коли рішення доводиться приймати в умовах ризику, невизначеності та коли результат подій залежить від ймовірностей. На кожне рішення впливають якісь певні фактори, і у кожного рішення є наслідки, що є імовірнісними. В умовах процес прийняття рішень є послідовним і метод дерева рішень передбачає визначення того, які дії необхідні на кожній вершині дерева. Дерево

рішень завжди потрібно будувати зліва направо. Точки прийняття рішень є варіантами альтернативних дій, тобто можливими виборами: вирішують, яким шляхом йти. Результати рішень незалежні. Вони залежать від зовнішнього середовища, наприклад, від клієнтів, постачальників чи стану економіки загалом. Як із точок прийняття рішень, так і з точок результатів виходять “гілки” дерева. Якщо існує, наприклад, два можливі варіанти дій, з точки прийняття рішення виходитимуть дві гілки, і якщо є два можливі результати (наприклад, хороший і поганий), то з точки результату теж виходитимуть дві гілки. Оскільки дерево рішень є інструментом для оцінювання різних варіантів дій, всі дерева рішень повинні починатися з точки прийняття рішення, яка графічно подається квадратом. Створену інформаційну систему озвучення україномовного тексту на основі методів NLP та машинного навчання спроектовано так, щоб будь-який користувач міг легко озвучити потрібний йому текст. Для цього йому необхідно мати операційну систему Windows та завантажити і запустити додаток. Мінімальні технічні вимоги для коректної роботи додатка: процесор 1 ГГц; ОЗУ 512 Мб; об’єм жорсткого диска 4,5 Гб. Також у користувача повинна бути встановлена операційна система не нижче ніж Windows XP. Оскільки додаток використовує для озвучування тексту пакети від Google, користувачу необхідне з’єднання з мережею інтернет. Вхідними даними реалізованого додатка є введення необхідного тексту для озвучування. Вихідними даними є озвучений текст. Систему реалізовано у вигляді десктоп-дodatка. Додаток є простим та зрозумілим. Для озвучення тексту необхідно запустити додаток, ввести чи додати потрібний текст, вибрати голос та натиснути на кнопку “Озвучити текст”. Якщо користувач хоче зберегти озвучений текст, потрібно натиснути на кнопку “Зберегти до файлу”.

Створений додаток названо UA Reader. Додаток розроблено із використанням мова програмування C# та пакетів від Google. Саме завдяки цьому додаток працює стабільно та швидко. Дизайн додатка створено за допомогою мови-розмітки XAML. Реалізований додаток вирішує проблему озвучення україномовного тексту. Для того, щоб озвучити текст, користувач повинен завантажити додаток на свій комп’ютер, а після завантаження запустити додаток під назвою UA Reader (рис. 19). Після запуску перед користувачем з’являється головне вікно додатка з доволі простим та зрозумілим інтерфейсом (рис. 20). Для того щоб отримати озвучений текст, користувач повинен ввести або додати необхідний йому текст у поле “Текст, який потрібно озвучити”, вибрати голос озвучення (рис. 21) та натиснути на кнопку “Озвучити текст”. Після озвучення користувач може зберегти файл, натиснувши кнопку “Зберегти до файлу” (рис. 22).

Имя	Дата изменения	Тип	Размер
auth	05.04.2022 19:36	JSON File	3 КБ
grpc_csharp_ext.x64.dll	29.09.2021 13:28	Расширение при...	10 229 КБ
last	10.05.2022 17:34	Файл "MP3"	6 КБ
UA-Reader	06.04.2022 22:18	Приложение	11 998 КБ
UA-Reader.pdb	06.04.2022 22:18	Program Debug D...	16 КБ

Рис. 20. Запуск додатка UA-Reader

Реалізована система є хорошим засобом для вирішення питання озвучення україномовного тексту. Додаток буде популярним серед іноземців, які бажають вивчити українську мову, та людей, які мають певні проблеми із зором.

Щоб перевірити працездатність додатка, спочатку потрібно його запустити. Процес запуску додатка показано на рис. 19. UA-Reader – простий та зрозумілий додаток, тому, щоб озвучити текст,

не потрібні якісь особливі вміння стосовно користування комп'ютером, необхідно лише завантажити та запустити додаток. Тому озвучити текст буде легко та зручно кожному. Після завантаження та запуску додатка користувач може побачити головне вікно додатка UA-Reader із сучасним та одночасно легким та зрозумілим інтерфейсом (рис. 20).

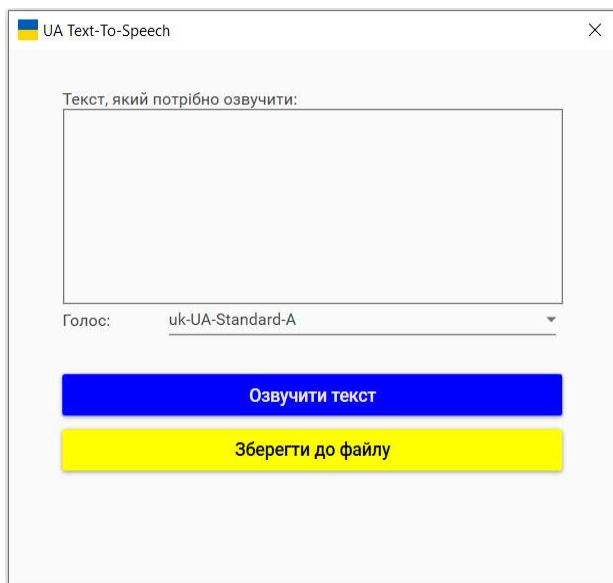


Рис. 21. Головне вікно додатка UA-Reader

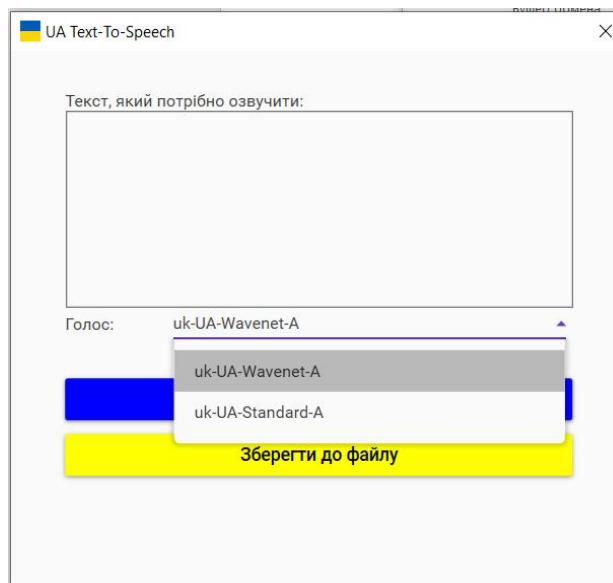


Рис. 22. Вибір голосу

Висновки

У статті проаналізовано відомі програмні рішення. Описано та проаналізовано переваги та недоліки розглянутих програмних рішень та подано порівняльну таблицю переваг та недоліків. Результати аналізу програмних рішень озвучення українського тексту враховано під час реалізації інформаційної системи. Спроектовано дерево цілей, IDEF0 діаграми та декомпозиції цих діаграм, а також ієрархію задач для розробленої інформаційної системи. Визначено головну ціль, за допомогою дерева цілей – “Створення інформаційної системи озвучення українського тексту на основі методів NLP та машинного навчання”, яку розділено на цілі-нащадки, щоб зробити проектування системи зручнішим та легшим. Також за допомогою IDEF0 конкретизовано функціонування системи та всі зв'язки процесів, побудовано ієрархію задач. Розглянуто системні та програмні засоби, які дають можливість виконувати необхідні завдання. Також розглянуто інструменти, технології та мови програмування, які використано під час проектування інформаційної системи. Описано методи та фреймворки, які забезпечують коректну, швидку та стабільну роботу додатка. Реалізовано інформаційну систему озвучення українського тексту на основі методів NLP та машинного навчання. Наведено та охарактеризовано функції реалізованого додатка. Також описано метод дерева цілей, який використано під час проектування інформаційної системи. Надано мінімальні системні вимоги для користування додатком. Створено інструкцію користувача для коректного використання пропонованої системи. Проілюстровано всі процеси користування додатком, проаналізовано контрольний приклад для підтвердження працездатності та функціонування додатка.

References

1. Shieldt, G. C. The complete reference. NY: Osborne McGraw-Hill, 1989.
2. Martin, Robert C. Clean code: a handbook of agile software craftsmanship. Pearson Education, 2009.
3. De Micheli, Giovanni, Rolf Ernst, and Wayne Wolf. Readings in hardware/software co-design. Morgan Kaufmann, 2002.

4. Karan, B., Mahto, K., & Sahu, S. S. (2019). Intelligent Speech Processing in the Time-Frequency Domain. *In Intelligent Speech Signal Processing*, 153–173. Academic Press. <https://doi.org/10.1016/B978-0-12-818130-0.00009-X>
5. Senior, Andrew W., and Anthony Robinson (1995). “Forward-backward retraining of recurrent neural networks”. *Advances in Neural Information Processing Systems*, 8.
6. Richter, Jeffrey. CLR via c#. Vol. 4. Redmond: Microsoft Press, 2006.
7. Nagel, Christian. Professional C# and .NET. John Wiley & Sons, 2021.
8. Matthew McDonald. Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5, 2012.
9. Van Santen, J. P., Sproat, R., Olive, J., & Hirschberg, J. (Eds.). (2013). Progress in speech synthesis. Springer Science & Business Media.
10. Ning, Y., He, S., Wu, Z., Xing, C., & Zhang, L. J. (2019). A review of deep learning based speech synthesis. *Applied Sciences*, 9(19), 4050. <https://doi.org/10.3390/app9194050>
11. Schroeder, M. R. (1993). A brief history of synthetic speech. *Speech communication*, 13(1–2), 231–237. [https://doi.org/10.1016/0167-6393\(93\)90074-U](https://doi.org/10.1016/0167-6393(93)90074-U)
12. Klatt, D. H. (1987). Review of text-to-speech conversion for English. *The Journal of the Acoustical Society of America*, 82(3), 737–793. <https://doi.org/10.1121/1.395275>
13. Flanagan, James L. (2013). Speech analysis synthesis and perception. *Springer Science & Business Media*, Vol. 3.
14. Isewon, I., Oyelade, J., & Oladipupo, O. (2014). Design and implementation of text to speech conversion for visually impaired people. *International Journal of Applied Information Systems*, 7(2), 25–30. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ed030474fe850f1c0ab4e8005af00e0671d14ffa>
15. Whitchurch, G. G., & Constantine, L. L. (1993). Systems theory. In *Sourcebook of family theories and methods: A contextual approach*, 325–355. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-85764-0_14
16. Anderson, Brian DO, and Sumeth Vongpanitlerd. Network analysis and synthesis: a modern systems theory approach. Courier Corporation, 2013.
17. Cronholm, Stefan. Why CASE Tools in Information Systems Development?: An Empirical Study Concerning Motives for Investing in CASE Tools. Linköping University, Department of Computer and Information Science, 1995.
18. Whitman, L., Huff, B., & Presley, A. (1997, December). Structured models and dynamic systems analysis: the integration of the IDEF0/IDEF3 modeling methods and discrete event simulation. In *Proceedings of the 29th conference on Winter simulation*, 518–524. <https://dl.acm.org/doi/pdf/10.1145/268437.268559>
19. Bublyk, M., Kalynii, T., Varava, L., Vysotska, V., Chyrun, L., & Matseliukh, Y. (2022). Decision Support System Design For Low-Voice Emergency Medical Calls At Smart City Based On Chatbot Management In *Social Networks*. *Webology* (ISSN: 1735-188X), 19(2).
20. Dokhnyak, B., & Vysotska, V. (2021). Intelligent Smart Home System Using Amazon Alexa Tools. In *MoMLeT+ DS, CEUR workshop proceedings*, 441–464. <https://ceur-ws.org/Vol-2917/paper33.pdf>
21. Vysotska, V., Holoshchuk, S., & Holoshchuk, R. (2021). A Comparative Analysis for English and Ukrainian Texts Processing Based on Semantics and Syntax Approach. In *COLINS, CEUR workshop proceedings*, 311–356. <https://ceur-ws.org/Vol-2870/paper26.pdf>
22. Aksonov, D., Gozhij, A., Kalinina, I., & Vysotska, V. (2021, September). Question-Answering Systems Development Based on Big Data Analysis. In *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, Vol. 1, 113–118. IEEE. DOI: 10.1109/CSIT52700.2021.9648631
23. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., & Kowalska-Styczen, A. (2018). Computational linguistics and intelligent systems. In *CEUR workshop proceedings*, Vol. 2136. <https://ceur-ws.org/Vol-3171/preface.pdf>
24. Lytvyn, V., Vysotska, V., Mykhailyshyn, V., Peleshchak, I., Peleshchak, R., & Kohut, I. (2019, July). Intelligent system of a smart house. In *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, 282–287. IEEE. DOI: 10.1109/AICT.2019.8847748
25. Voloshyn, S., Vysotska, V., Markiv, O., Dyyak, I., Budz, I., & Schuchmann, V. (2022, November). Sentiment Analysis Technology of English Newspapers Quotes Based on Neural Network as Public Opinion Influences

Identification Tool. In *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, 83–88. IEEE. DOI: 10.1109/CSIT56902.2022.10000627

26. Kravets, P., Burov, Y., Oborska, O., Vysotska, V., Dzyubyk, L., & Lytvyn, V. (2021). Stochastic Game Model of Data Clustering. In *IntelITSIS, CEUR workshop proceedings*, 198–213. <https://ceur-ws.org/Vol-2853/paper19.pdf>

27. Shakhovska, N., Vysotska, V., & Chyrun, L. (2017). Intelligent systems design of distance learning realization for modern youth promotion and involvement in independent scientific researches. In *Advances in Intelligent Systems and Computing: Selected Papers from the International Conference on Computer Science and Information Technologies, CSIT 2016, September 6–10 Lviv, Ukraine*, 175–198. Springer International Publishing. https://doi.org/10.1007/978-3-319-45991-2_12

28. Lytvyn, V., et. al. (2019, September). A smart home system development. In *Conference on Computer Science and Information Technologies*, 804–830. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-33695-0_54

29. Vysotska, V., Markiv, O., Teslia, S., Romanova, Y., & Pihulechko, I. (2022). Correlation Analysis of Text Author Identification Results Based on N-Grams Frequency Distribution in Ukrainian Scientific and Technical Articles. In *CEUR Workshop Proceedings*, Vol. 3171, 277–314. CEUR-WS. <https://ceur-ws.org/Vol-3171/paper25.pdf>

30. Tymoshenko, K., Vysotska, V., Kovtun, O. V., Holoshchuk, R., & Holoshchuk, S. (2021). Real-Time Ukrainian Text Recognition and Voicing. In *COLINS, CEUR workshop proceedings*, 357–387. <https://ceur-ws.org/Vol-2870/paper27.pdf>

INFORMATION SYSTEM FOR UKRAINIAN TEXT VOICEOVER BASED ON NLP AND MACHINE LEARNING METHODS

Illia Bielousov¹, Lyubomyr Chyrun², Sofia Chyrun¹, Ihor Budz¹, Olha Vlasenko^{4,5}

¹ Lviv Polytechnic National University, Information Systems and Networks Department, 12, S. Bandery str., Lviv, Ukraine

² Ivan Franko National University of Lviv, Applied Mathematics Department, 1, University STR., Lviv, Ukraine

³ Lviv Polytechnic National University, Department of Specialized Computer Systems, 12, S. Bandery str., Lviv, Ukraine

⁴ Osnabrück University, Institute of Education Science, 9, Heger-Tor-Wall, Osnabrück, Germany

⁵ Zhytomyr Ivan Franko State University, Professional and Pedagogical, Apecial Education, Andragogy and Management Department, 40, Velyka Berdychivska str., Zhytomyr, Ukraine

E-mail: illia.bielousov.knm.2018@lpnu.ua, ORCID: 0000-0002-4851-4460

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002- 9448-1751

E-mail: sofia.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

E-mail: Ihor.S.Budz@lpnu.ua, ORCID: 0000-0002-5400-0984

E-mail: olha.vlasenko@uni-osnabrueck.de, ORCID: 0000-0001-7258-2108

© Bielousov I., Chyrun L., Chyrun S., Budz I., Vlasenko O., 2023

During the research, an information system for voicing Ukrainian-language text was developed based on NLP and machine learning methods. The created information system is implemented in the form of a desktop application, which allows the process of voicing the Ukrainian-language text. The created system included all stages of software development: the design process, the implementation process, and the testing process. For the feasibility of creating this system, already existing software solutions on the

market were analysed, their advantages and disadvantages were listed, which were subsequently taken into account to create a new system. During the system analysis of the system, a goal tree, a decision tree, and examples of context diagrams with process decomposition are given. One of the stages of the design of the economic part, where the budget that will be spent on the implementation of the system is analysed, all tax and administrative costs are calculated, development strategies are also analysed and the development strategy of the existing product with accompanying solutions and the product development strategy are selected. After that, an assessment was made for the feasibility of creating the designed system, its payback and profit. The object of the research is the process of the voiceover system of the Ukrainian-language text based on NLP and machine learning methods. The subject of the research is the methods and means of the Ukrainian-language text voicing system process based on NLP and machine learning methods. The purpose of the research is to create an information system for voicing Ukrainian-language text based on NLP and machine learning methods. The result of the work is a ready-to-implement information system for voicing Ukrainian-language text based on NLP and machine learning methods, an analytical review of literary and online sources related to the topic of voicing Ukrainian-language text based on NLP and machine learning methods, a systematic analysis of the research object, analysis and selection of software tools for system implementation, practical implementation of the system, economic justification of system implementation activities.

Key words: information system; NLP; Ukrainian-language text; machine learning; text voiceover.