

ІДЕНТИФІКАЦІЯ ТА ВИКОНАННЯ ГОЛОСОВИХ КОМАНД ПЕРСОНАЛЬНИМИ МОБІЛЬНИМИ ПОМІЧНИКАМИ ІЗ ВИКОРИСТАННЯМ ПРОДУКЦІЙНОЇ МОДЕЛІ ПРЕДСТАВЛЕННЯ ЗНАНЬ

© Шевчук Р.П., 2013

Побудовано продукційну модель для ідентифікації голосових команд користувачів персональних мобільних пристроїв до операторів стільникового зв'язку України. Запропоновану модель покладено в основу мобільного помічника для абонентів операторів стільникового зв'язку України, реалізованого на основі ОС Android. Проведено порівняльний аналіз часових характеристик голосового та стандартного методу введення команд користувачами мобільних пристроїв, що дало змогу встановити підвищення швидкодії виконання команд до 3,6 разів під час використання запропонованого рішення.

Ключові слова: персональний мобільний помічник, продукційна модель, голосові команди, сервіси операторів стільникового зв'язку, розпізнавання голосу, Андроїд.

In this paper the productive model of identifying the voice commands on personal mobile devices of Ukrainian mobile operators was developed. The personal mobile assistant for users of mobile devices based on OS Android was developed. This assistant takes into account the operation service of Ukrainian mobile network operators. We made a comparative analysis of temporal characteristics of personal mobile users command input. This analysis showed the increase in performance to 3.6 times more than standard method of commands input.

Key words: personal mobile assistant, production system, voice command, service of mobile network operators, voice recognizing, Android.

Вступ

Сьогодні на ринку програмного забезпечення (ПЗ) для персональних мобільних пристроїв (ПМП) спостерігається експансія засобів голосового керування та інтелектуальних мобільних помічників (ІМП), які взаємодіють з користувачами в інтегрованій, розмовній формі, використовуючи природні звороти мови. Як правило, ці засоби використовують зовнішні сервіси та бази даних для оброблення запитів і виконання певних завдань. При цьому в їх основу покладено ідеї персоналізації та передбачення, які дають змогу надавати користувачам інформацію тоді, коли вона їм потрібна.

Основними функціями ІМП є [1–6]:

- голосове управління;
- голосове введення тексту;
- голосовий пошук;
- надання інформації про: погоду, події календаря, автомобільні маршрути, графіки руху транспорту тощо;
- формування відповідей на поставлені запитання.

Під час розробки ІМП застосовують клієнтський і клієнт-серверний підходи. У ПЗ першого типу, наприклад у Speereo Software [1], алгоритм розпізнавання голосу реалізується на самому

пристрої. Перевагою таких ІМП є незалежність від доступу до мережі Інтернет, однак продуктивність їх роботи залежить від потужності ПМП, тому вони використовуються рідко. Під час використання клієнт-серверного підходу голосовий сигнал зчитується ПМП користувача і через Інтернет надсилається серверу, на якому відбувається процес розпізнавання голосу. Для таких систем характерне навчання на основі зразків вже розпізнаних голосових вибірок. Прикладом таких ІМП є Google Now [2], Siri [3], Vlingo [4], Maluuba [5], Sherpa [6] та інші.

Аналіз використання функцій користувачами ІМП, проведений Parks Associates, показав, що найпопулярнішими виявились телефонні функції, а саме – написання смс та здійснення дзвінків [7]. Зважаючи на цей факт, у роботі реалізовано та досліджено ІМП для абонентів операторів стільникового зв'язку України.

1. Постановка завдання

Сьогодні оператори стільникового зв'язку пропонують широкий набір сервісних додатків і послуг для задоволення потреб абонентів. Для підтримки конкурентоспроможності кожен оператор намагається зробити користування такими послугами якомога зручнішим для своїх абонентів. Щоб їх використовувати, абонентам оператора стільникового зв'язку потрібно запам'ятовувати конкретні запити чи абонентські номери, що створює для них істотні незручності. Найчастіше абоненти зберігають необхідні дані у телефонній книзі з певною асоційованою назвою (наприклад, «погода», «стан рахунку», «тарифний план»), за якою згодом здійснюють пошук та виконують запит. Однак багато запитів до операторів є складеними, тобто містять кілька комбінацій цифр, які необхідно ввести абоненту для того, щоб виконати певний запит. Наприклад, запит *124*XXX*380XXXXXXXXX# (сума) (номер абонента) уможливило переказати гроші з одного мобільного рахунку абонента оператора Київстар на інший [8]. На такі запити абоненти витрачають від 5 секунд до кількох хвилин. Крім того, запити для ідентичних послуг різних операторів стільникового зв'язку України різняться між собою.

Отже, актуальним є завдання розроблення мобільного помічника, що враховує особливості функціонування сервісних послуг операторів стільникового зв'язку України. Для реалізації цього завдання пропонується використати продукційну модель представлення знань та Інтернет-сервіс Google Voice Search.

Розроблене ПЗ дасть змогу скоротити трудовитрати, збільшити продуктивність систем самообслуговування, покращити зручність та швидкість використання сервісів та послуг, зменшити навантаження на call-центри операторів стільникового зв'язку.

2. Особливості побудови продукційної моделі представлення знань для розпізнавання голосових команд

Продукційна модель – одна із моделей представлення знань, яка дає змогу представити знання у вигляді речень виду «якщо (умова), то (дія)» [9]. У загальному вигляді продукцією вважають такий вираз [9]:

$$(i);Q;P;A \rightarrow B;N, \quad (1)$$

де i – ім'я (ідентифікатор) продукції, для чіткої ідентифікації серед інших. Це може бути лексема або порядковий номер. Q характеризує галузь (область) застосування продукції. Поділ знань на окремі галузі дає можливість зекономити час під час пошуку розв'язання задачі. Найважливіша частина продукції $A \rightarrow B$ – це її ядро. Ядро продукції читають так: «Якщо (умова), то (дія)». P – умова, за якої застосовується ядро продукції. Переважно P – це логічний вираз. Активація ядра продукції відбувається за умови, що P має значення «істина». В іншому випадку ядро продукції не виконується. Після реалізації B активізується після умови N , тобто здійснюються необхідні дії чи процедури.

Побудова продукційної моделі представлення знань для розпізнавання голосових команд користувачів стільникового зв'язку передбачає наявність двох компонент – бази понять (термів) із множиною значень та продукції (правила).

Продукційна модель ідентифікує такі команди:

- поповнення рахунку;
- перевірка стану рахунку;
- переказ другу;
- дзвінок на номер;
- дзвінок контакту телефонної книги;
- дзвінок оператору.

Для визначення термів та продукцій було проведено анкетування, в якому респондент пропонував зручний варіант голосової команди. Зведені результати анкетування (варіанти голосових команд) наведено у табл. 1.

Таблиця 1

Варіанти голосових команд

Команда	Варіант голосового запиту
поповнення рахунку	«поповнити рахунок», «поповнення балансу», «поповнити», «поповнення»
перевірка стану рахунку	«перевірити рахунок», «баланс», «рахунок», «мій баланс», «мій рахунок», «перевірити баланс», «покажи рахунок», «залишок на рахунку», «залишок»
переказ другу	«переказати», «відправити», «переказати гроші», «відправити гроші»
дзвінок на номер	«подзвонити», «дзвінок», «подзвонити на номер», «дзвінок на номер»
дзвінок контакту телефонної книги	«з'єднати з контактом», «подзвонити контакту», «дзвінок контакту»
дзвінок оператору	«оператор», «дзвінок оператору», «подзвонити оператору», «з'єднати з оператором»

Проаналізувавши табл. 1, для реалізації продукційної моделі були виділені такі терми із множиною значень:

- баланс {рахунок, баланс, залишок, балансу};
- перегляд {перевірити, подивитись, показати, покажи, переглянути, перевірі};
- поповнення {поповнити, поповнення};
- відправка {переказати, передати, відправити};
- гроші {гроші, кошти};
- дзвінок {подзвонити, з'єднати, дзвінок, з'єднай, подзвони};
- оператор {оператор, оператора, оператору, оператором};
- номер {номер};
- контакт {(список генерується автоматично на основі телефонної книги)}.

Під час формування продукційних правил використовуються логічні зв'язки «і» та «або». За формалізації знань зв'язки реалізують операціями логічного множення та додавання відповідно. Для ідентифікації голосових команд виділено такі правила:

1. Якщо «баланс» і «перегляд» або «баланс», то «перевірка стану рахунку».
2. Якщо «поповнення» і «баланс» або «поповнити» і «рахунок», або «поповнити», або «поповнення», то «поповнити рахунок».
3. Якщо «відправка» і «гроші», то «переказ другу».
4. Якщо «дзвінок» і «оператор», або «оператор», то «дзвінок оператору».
5. Якщо «дзвінок» і «номер» або «дзвінок», то «дзвінок на номер».
6. Якщо «дзвінок» і «контакт», то «дзвінок контакту телефонної книги».

Для управління виконанням правил застосовується принцип найдовшої умови, який полягає у виборі продукції, для якої виконалась найдовша умова виконання ядра. Використання цієї стратегії обґрунтовано тим, що потрібно уникнути істинності ядра продукції 1, коли результатом розпізнавання є фрази «поповнити рахунок» або «поповнення балансу». Аналогічна ситуація з

продукцією 5, яка виконається, якщо користувач скаже «дзвінок оператору» або «подзвонити оператору», або «з'єднати з оператором», або «з'єднати з контактом», або «подзвонити контакту», або «дзвінок контакту».

Принцип використання продукційної моделі на прикладі ідентифікації команди «перевірити рахунок» зображено на рис. 1.

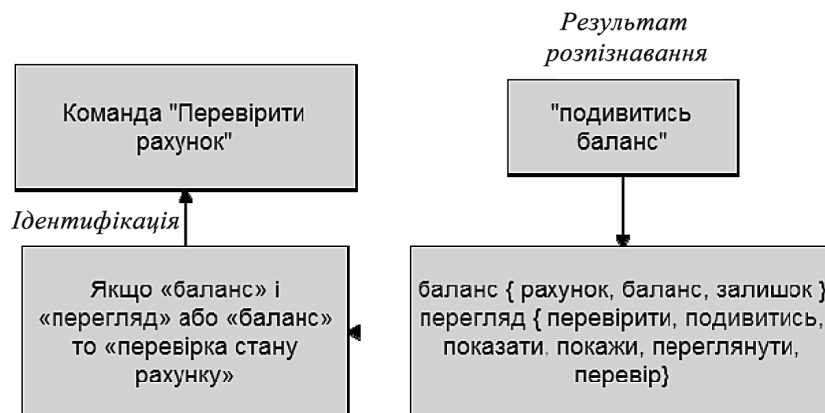


Рис. 1. Ідентифікація команди «Перевірити рахунок»

На вхід продукційна модель отримує фразу «подивитись баланс». Серед бази понять, вибираються терми «баланс» та «перегляд». Далі відбувається пошук правила. Ядро продукції набуло значення «істина», яке ідентифікує команду «перевірити рахунок».

3. Процес ідентифікації USSD для операторів стільникового зв'язку

Для забезпечення доступності сервісних додатків для абонентів оператори стільникового зв'язку використовують так звані USSD (Unstructured Supplementary Service Data) – запити, які уможливають реалізувати інтерактивну взаємодію будь-якого абонента із доступними наборами сервісів [10]. Запити від абонента до сервісного центру передаються в режимі коротких повідомлень.

Принцип використання USSD-запитів ілюструє такий приклад. Абонент хоче дізнатись стан рахунку. Йому потрібно відправити USSD-повідомлення до сервісного центру, щоб викликати відповідний додаток. Сервісний додаток на сервері має деякий унікальний ідентифікатор. Наприклад, для оператора мобільного зв'язку «Київстар» такий ідентифікатор має значення «111». Тоді для абонента, щоб дізнатись стан рахунку, потрібно відправити повідомлення «*111#». Абонент, набравши таке повідомлення на своєму телефоні, передає його у сервісний центр натисненням на кнопку виклику. Коли відповідний USSD-додаток був запущений, він, своєю чергою, виконує запит до бази даних для отримання стану рахунку. Коли дані були отримані, додаток відправляє абоненту відповідь у вигляді USSD-пакета. У абонента такий USSD-пакет відображається на екрані у вигляді тексту.

Символи «*» та «#» вказують на те, що відбувається незвичайний виклик, а саме: USSD-запит.

Такий механізм роботи дає змогу мінімізувати затримку в часі між запитом та відповіддю, чого важко досягти за допомогою SMS.

Також потрібно відмітити, що додатки на USSD-сервері можуть бути параметризовані. Тобто вони можуть приймати від абонента деякі параметри. Наприклад, оператори стільникового зв'язку використовують такі параметри для переказу коштів чи поповнення рахунку. Для переказу коштів іншому абоненту параметрами виступає сума та номер іншого абонента. Для реалізації передачі такого USSD-запиту у мережі «Київстар» достатньо набрати: «*124*сума*номер абонента#».

Для реалізації виконання голосових команд користувача використання USSD-запитів в ОС Android достатньо написати код:

```
Intent intent = new Intent(Intent.ACTION_CALL);
intent.setData(Uri.parse("tel:" + USSD));
startActivity(intent).
```

Виклик здійснюється шляхом використання уніфікованого ідентифікатора (URI) та виклику відповідного Activity з дією Intent.ACTION_CALL. Як уніфікований ідентифікатор використовується змінна USSD, яка може мати значення будь-якого запиту.

Існує проблема реалізації різних запитів, оскільки кожен із мобільних операторів використовує свій унікальний USSD-запит для виклику того чи іншого сервісу.

Для вирішення цієї проблеми були проаналізовані запити мобільних операторів МТС, Life та Djuce, а також розроблений алгоритм, що уможливує виконувати відповідні запити для конкретного оператора стільникового зв'язку.

Серед списку команд USSD-запити використовуються для перевірки стану рахунку, поповнення рахунку та переказу коштів. Для різних операторів можна виділити такі USSD-запити (табл. 2).

Таблиця 2

Класифікація USSD-запитів для операторів МТС, life та Djuce

USSD Оператор	Стан рахунку	Поповнення рахунку	Переказ коштів
МТС	*101#	*100*код ваучера#	*150*номер телефону*сума#
Life	*111#	*100*код ваучера#	*111*номер телефону*сума#
Djuice	*111#	*123*код ваучера#	*124*сума* номер телефону#

Оскільки реалізація ПЗ для платформи Android передбачає збереження усіх ресурсів у XML-файлах каталогу «/res/values/», то значення USSD-запитів можна зберігати у вигляді масивів для відповідного оператора стільникового зв'язку.

Кожен масив реалізується за допомогою відкриваючого та закриваючого тегу <array>. Значення атрибуту name для масивів вказує на мобільного оператора, а для елементів масиву на USSD-запит. Відповідно елементи масиву представляє тег <item>. Своєю чергою, тег <resource> дає змогу системі ідентифікувати, що у цьому файлі містяться саме ресурси для нашого додатка, що в подальшому надає доступ до таких ресурсів із коду.

Для ідентифікації поточного оператора зв'язку використовується об'єкт класу TelephonyManager. Клас дає змогу ідентифікувати MCC- та MNC-код для поточного стільникового оператора. MCC (Mobile Country Code) – дає змогу ідентифікувати код країни мобільного оператора у мережах GSM, UMTS, CDMA, TETRA, iDEN. А код MNC (Mobile Network Code) – це унікальний ідентифікатор оператора мобільного зв'язку. В поєднанні MCC та MNC дають можливість легко ідентифікувати оператора. Оскільки ПЗ розроблялось для стільникових операторів України, то відповідно код MCC для трьох операторів буде однаковим.

Реалізація ідентифікації поточного стільникового оператора відбувається за допомогою виклику методу IdentifyOperator(), який має такий вигляд:

```
public void IdentifyOperator(){
    TelephonyManager tMgr =(TelephonyManager)
    getApplicationContext().getSystemService(Context.TELEPHONY_SERVICE);
    String MCC_MNC_operator_code = tMgr.getSimOperator().toString();
    Resources res = getResources();
    switch(MCC_MNC_operator_code){
        case "25501":
            operator_commands = res.getStringArray(R.array.commands_mts)
            break;
        case "25503":
            operator_commands = res.getStringArray(R.array.commands_djuice);
            break;
        case "25506":
            operator_commands = res.getStringArray(R.array.commands_life);
            break;
    }
}
```

Метод класу `getSimOperator()` повертає поєднання значень MCC та MNC для поточного оператора зв'язку. Наприклад, значення 25501 для оператора «МТС» вказує, що код країни MCC дорівнює «255», а унікальний ідентифікатор (MNC) для оператора – «01».

Такий підхід доволі зручно використовувати для ідентифікації оператора. Для збільшення списку підтримуваних операторів зв'язку достатньо знати його унікальний ідентифікатор, а також список USSD-запитів.

4. Розробка мобільного помічника

Розробка мобільного помічника була розпочата із формулювання функціональних та нефункціональних вимог.

До функціональних вимог зараховано:

- введення голосової команди;
- ідентифікацію голосових команд: «Перевірити рахунок», «Поповнити рахунок», «Переказ другу», «Дзвінок оператору», «Дзвінок на номер», «Дзвінок контакту телефонної книги»;
- виконання голосових команд: «Перевірити рахунок», «Поповнити рахунок», «Переказ другу», «Дзвінок оператору», «Дзвінок на номер», «Дзвінок контакту телефонної книги»;
- можливість повторного введення голосової команди у разі неможливості ідентифікації голосової команди.

Нефункціональні вимоги:

- розробка ПЗ для ОС Android версії 2.2 та вище;
- ідентифікація та виконання голосових команд для операторів стільникового зв'язку України: МТС, life, Djuice;
- використання API функцій сервісу Google Voice Search для розпізнавання голосу;
- особливості ПЗ: мова реалізації Java; передбачає 1 вікно для початку введення голосової команди; інтерфейс вікна складається із елементів: `LinearLayout` та `ImageButton`; збереження даних відбувається за допомогою використання спеціалізованих файлів ресурсів формату XML.

Взаємодія користувача і ПМП передбачає введення голосової команди абонентом, а ПМП, своєю чергою, виконує відповідну дію. ПМП та сервер Google взаємодіють у режимі «запит-відповідь». Запит містить голосові дані абонента, а відповідь від сервера приходить у вигляді текстового повідомлення.

Розроблене ПЗ містить такі модулі (рис. 2):

- користувач;
- голосовий інтерфейс користувача;
- продукційну модель.

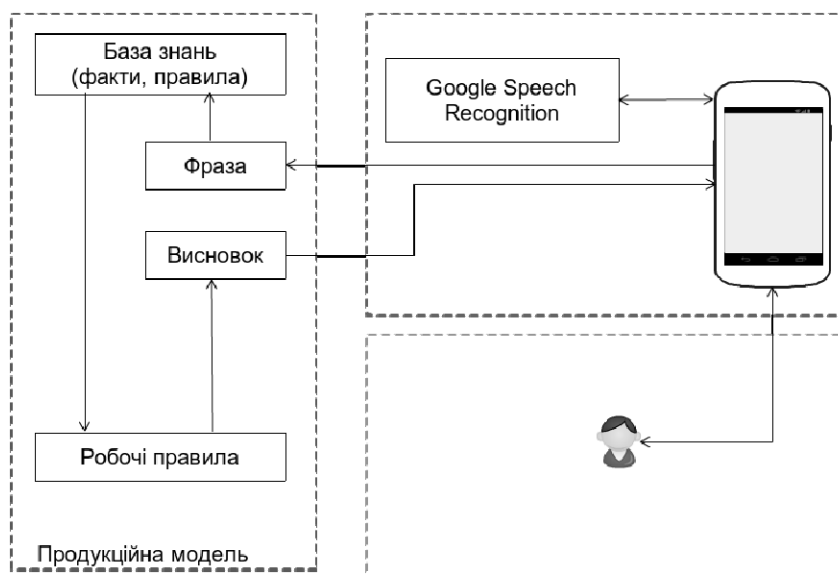


Рис. 2. Архітектура мобільного помічника

Голосовий інтерфейс користувача передбачає функції введення голосової команди розпізнавання та виконання команди. Для введення голосової команди користувач повинен ініціювати виклик системного додатка SpeechRecognizeActivity. Для цього потрібно натиснути на кнопку “Speak Now”.

Після виклику SpeechRecognizeActivity користувач повинен ввести голосове повідомлення. У цей момент системний додаток відправить запит до сервера Google, який, своєю чергою, дасть відповідь із текстовим результатом голосової команди. Якщо неймережа Google не зможе розпізнати голосової команди, то системний додаток SpeechRecognizeActivity не дасть текстового результату та повідомить користувача про помилку і надасть можливість повторного введення команди.

Коли SpeechRecognizeActivity повертає розпізнану голосову команду у вигляді текстового повідомлення до додатка, то цей результат проходить через побудовану продукційну модель для ідентифікації голосової команди.

Продукційна модель на основі отриманої фрази, бази знань та сукупності правил формує певний висновок, що і є результатом процесу ідентифікації команди. Детальніше цей процес можна розділити на такі кроки:

- вхідна фраза порівнюється із деякою кількістю термів із бази знань, кожен з яких містить допустимі значення слів для відповідних команд;
- якщо після порівнянь знайшлися збіги, то результати збігів проходять через набір продукційних правил;
- на основі правил формується висновок (ідентифікована команда).

Якщо продукційна модель не ідентифікує команду, то про це ПЗ повідомить користувача та надасть можливість повторного введення команди. В іншому випадку ПЗ спробує виконати ідентифіковану команду.

Програмна реалізація мобільного помічника виконувалась у середовищі розробки Android Developer Tools версії 22.0.0-675183 мовою програмування Java.

Тестування розробленого ПЗ проводилось за допомогою використання методики функціонального тестування на ПМП Huawei U8815 Ascend G300 із операційною системою Android 4.0.3 (Ice Cream Sandwich).

5. Дослідження часових характеристик мобільного помічника

У роботі проаналізовано часові характеристики розробленого мобільного помічника із використанням моделі GOMS, яка дає змогу розрахувати час виконання голосової команди користувача від моменту виклику розробленого додатка до реалізації виконання на основі додавання усіх часових інтервалів, що необхідні користувачу на виконання послідовних дій для цієї команди [11].

Оцінювання швидкості роботи проведено за допомогою двох підходів:

- використання моделі GOMS;
- використання секундоміра.

Порівняльний аналіз швидкості виконання команд наведено у табл. 3.

Таблиця 3

Порівняльний аналіз швидкості виконання команд

Команда	Підхід	Модель GOMS		Секундомір	
		для стандартного введення, с	для голосового введення, с	для стандартного введення, с	для голосового введення, с
«Перевірка стану рахунку»		5,47	6,37	6	7,1
«Поповнення рахунку»		30,17	8,37	25	8
«Переказ другу»		10,72	7,37	15	7,43
«Дзвінок на номер»		8,72	8,07	8	7,9
«Дзвінок контакту телефонної книги»		12	6,77	20	7
«Дзвінок оператору»		7,52	6,37	9	6,9

Проаналізувавши табл. 3, можна зробити висновок, що розроблений додаток забезпечує підвищення швидкодії виконання команди (у 3,6 раза) під час роботи із складеними запитами. Отриманий результат підтверджує доцільність та актуальність використання розробленого мобільного помічника.

Висновки

Розпізнавання голосу – один з найперспективніших напрямків в області інформаційних технологій. Можливості голосового управління і спілкування природною мовою найпривабливіші для ринку мобільних систем, про що свідчить поява великої кількості засобів голосового керування та ІМП.

У роботі запропоновано розв'язувати задачу ідентифікації та виконання голосових команд за допомогою побудови продукційної моделі та віддаленого розпізнавання голосових команд. Як наочний приклад, побудовано продукційну модель для ідентифікації голосових команд користувача до операторів стільникового зв'язку України. Побудовану модель покладено в основу розробки мобільного помічника для ідентифікації та виконання голосових команд користувачів ПМП на основі ОС Android, що враховує особливості функціонування сервісних послуг мобільних операторів України. Проведено порівняльний аналіз часових характеристик голосового та стандартного методу введення команд користувачами ПМП на основі розробленого мобільного помічника, що дало змогу підвищити швидкодію виконання команд у 3,6 раза.

1. *Speereo Voice Translator*, retrieved from <http://www.speereoovt.com>. 2. *Abi Sehmi*. "Intelligent Personal Assistant: Siri vs Google", retrieved from <http://blog.limetreeonline.com>, July 2013. 3. *Josh Garnier*. "Siri Personal Assistant: A Voice App That Lets You Speak to OpenTable", retrieved from <http://blog.opentable.com>, February 2010. 4. *Banks Courtney*. "A Safer Way to Text on the Road". *The Wall Street Journal*, January 2010. 5. *Howitt Chuck*. "Waterloo startup Maluuba earns funding for voice-activated software", retrieved from <http://www.therecord.com>, October 2012. 6. *Eliane Fiolet*. *Sherpa "Personal Assistant App for Android"*, retrieved from <http://www.ubergizmo.com/>, May 2013. 7. *Over one-half of Apple iPhone 4S users satisfied with Siri and 37% want Siri-style voice command for TV*. Retrieved from <http://www.parksassociates.com>, May 2012. 8. <http://www.kyivstar.ua/> 9. *Brownston, L., Farrell R., Kant E*. "Programming Expert Systems in OPS5 Reading", Massachusetts: Addison-Wesley, 1985. 10. "Unstructured Supplementary Services Data (USSD)". retrieved from www.telecomspace.com, August, 2010. 11. *Schrepp, M., Fischer, P*. "GOMS models to evaluate the efficiency of keyboard navigation in web units". *E Minds-International Journal of Human Computer Interaction*, Vol. I, No. 2, 33-46. – 2010.