International Workshop on Computer Modeling and Intelligent Systems (CMIS) 2021

"An Approach to Development of Interactive Adaptive Software Tool to Support Data Analysis Activity"

Dmytro Orlovskyi

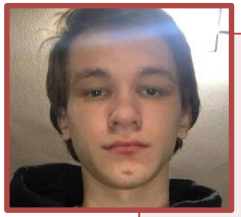
Ph.D. in Information Technology,
Associate Professor

orlovskyi.dm@gmail.com

Andrii Kopp

Ph.D. in Information Technology,
Senior Lecturer

kopp93@gmail.com



Ivan Bilous

Student

ivanbilous2000@gmail.com

Department of Software Engineering and Management Information Technology

Faculty of Computer Science and Software Engineering

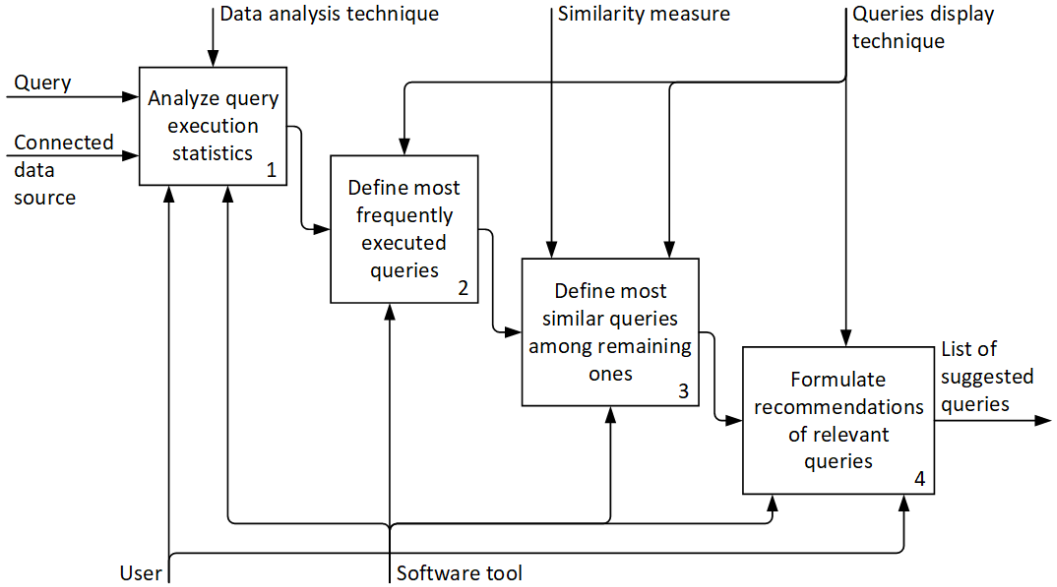
KHARKIV POLYTECHNIC INSTITUTE
1885
NTU

Motivation

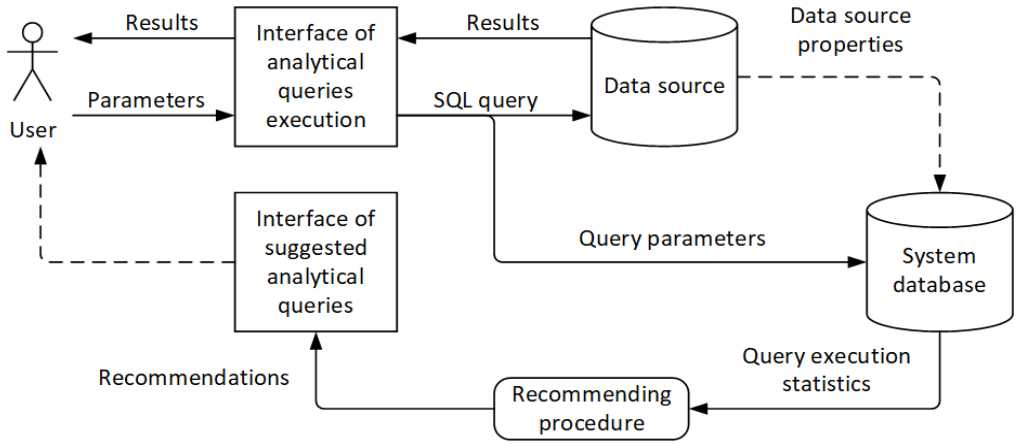
- Nowadays enterprise databases contain extremely large collections of transactional and aggregated analytical data records.
- Without having interactive and flexible querying tools, such data collections are basically useless, since no data could be obtained for analytical processing, visualization, and decision making.
- Since query languages require skills and experience to be applied, and existing “query wizards” are limited in their functionality and mostly are not understandable for end users, the problem of development of a software tool for data querying becomes relevant.
- Therefore, this study proposes an approach to development of interactive adaptive software tool to support data analysis activity.

Problem Statement

- The interactive adaptive software system for data analysis is supposed to be utilized by administrators and data analysts.
- Recommending algorithm should analyze query execution statistics and define frequencies of query execution in order to detect relevant queries.
- Three most frequently used queries should be compared to remaining queries in order to detect two most similar queries for each of the most frequently used ones.



Proposed Approach



- The underlying idea considers capturing statistics of queries execution by users and therefore it becomes possible to formulate the list of frequently executed queries for each user by each data source.
- Most of recommending systems that support content-based filtering approach use keyword matching or vector space model (VSM) with simple TF-IDF weighting.

Content-based Filtering

- According to the proposed recommending procedure, SQL statements are considered as documents, while pairs of parameter names and values of SQL statements that call stored procedures are considered as terms:

$$\underbrace{\text{EXEC GetStudents } \overbrace{\text{@Year } = \text{'2020'}}^{t_{kj}}, \text{@Country } = \text{'Turkey'};}_{d_j}$$

- TF-IDF is the statistical indicator used to evaluate importance of terms within the context of a document that belongs to the document collection.
- Hence, terms that occur in one document, but rarely occur in remaining documents, are more often relevant to the document's topic.
- Cosine similarity measure could be used to calculate similarity between two vectors of normalized TF-IDF measures:

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{n}(w_{ki} \cdot w_{kj})}{\sqrt{\sum_{k=1}^{n} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{n} w_{kj}^2}}, i \neq j, i = \overline{1,m}, j = \overline{1,m}.$$

Example of Content-based Filtering Application

- For example, a stored procedure is called ten times with different parameters.
- Documents d_1 and d_4 could be represented using the following vectors:

$$d_1 = \{0.71,0,0,0,0.71,0,0,0\},$$
$$d_4 = \{0,0.5,0,0,0.87,0,0,0\},$$

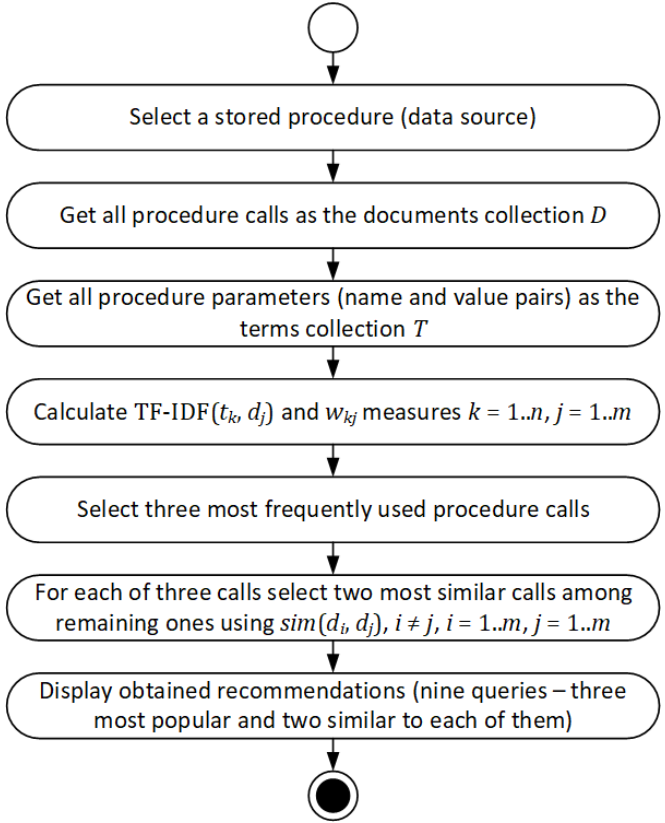
- Similarity between d_1 and d_4 is equal to $sim(d_1, d_4) = 0.61$.
- Another document, which similarity is greater than zero when compare it to the d_1, is d_7 for which $sim(d_1, d_7) = 0.42$.
- Then, there are two queries that could be recommended as relevant:

d_j	Year	Country	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
d_1	2018	Morocco	1	0	0	0	1	0	0	0
d_2	2019	Turkey	0	1	0	0	0	1	0	0
d_3	2019	Turkey	0	1	0	0	0	1	0	0
d_4	2018	Turkey	0	1	0	0	1	0	0	0
d_5	2020	Algeria	0	0	1	0	0	0	1	0
d_6	2017	Turkey	0	1	0	0	0	0	0	1
d_7	2017	Morocco	1	0	0	0	0	0	0	1
d_8	2020	Pakistan	0	0	0	1	0	0	1	0
d_9	2020	Turkey	0	1	0	0	0	0	1	0
d_{10}	2018	Morocco	1	0	0	0	1	0	0	0

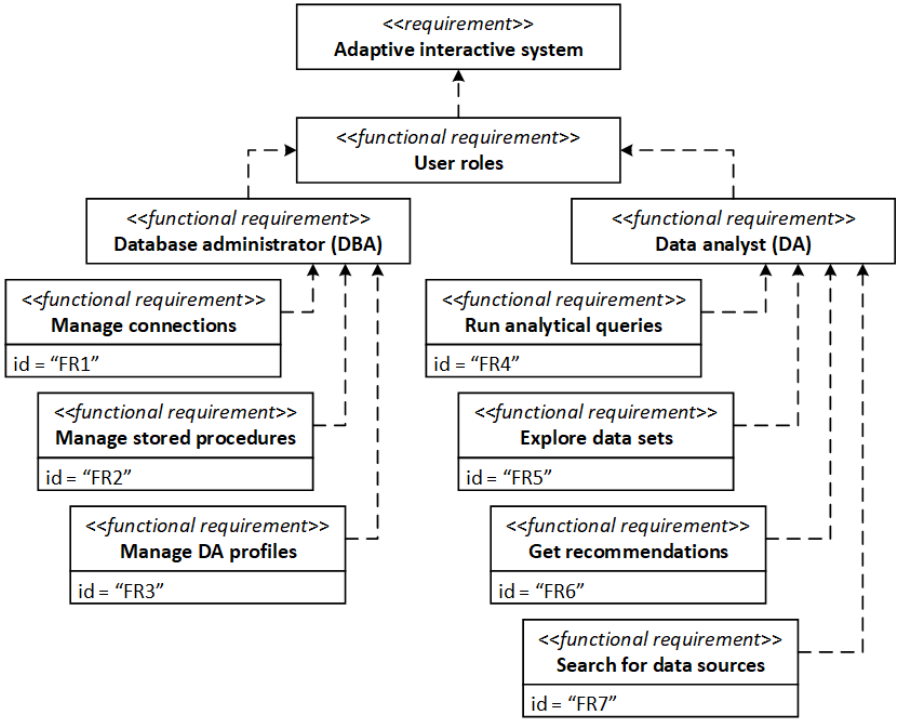
$$\text{EXEC GetStudents @Year} = '2018', \text{@Country} = '\text{Turkey}';$$
$$\text{EXEC GetStudents @Year} = '2017', \text{@Country} = '\text{Morocco}';$$

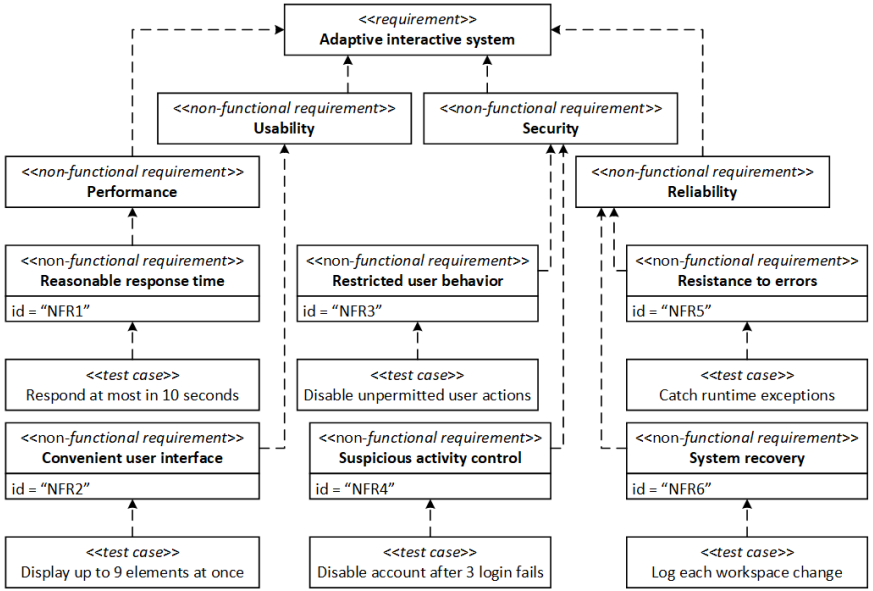
Recommending Procedure

- At first, three most frequent queries should be chosen.
- Then for each of the most frequent queries, selected at first step, should be chosen two most similar to them queries within corresponding data sources.
  Similarity between two queries could be calculated using the Vector Space Model and TF-IDF weighting approach applied to SQL statements of stored procedure calls.
- As the result, the list of nine recommended queries should be formulated:
  - Three most popular queries
  - Six similar queries – by two for each of three most popular queries



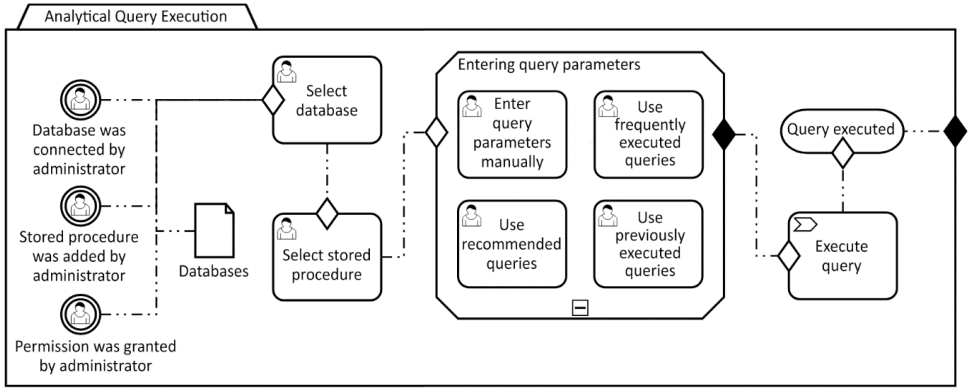
Software Requirements

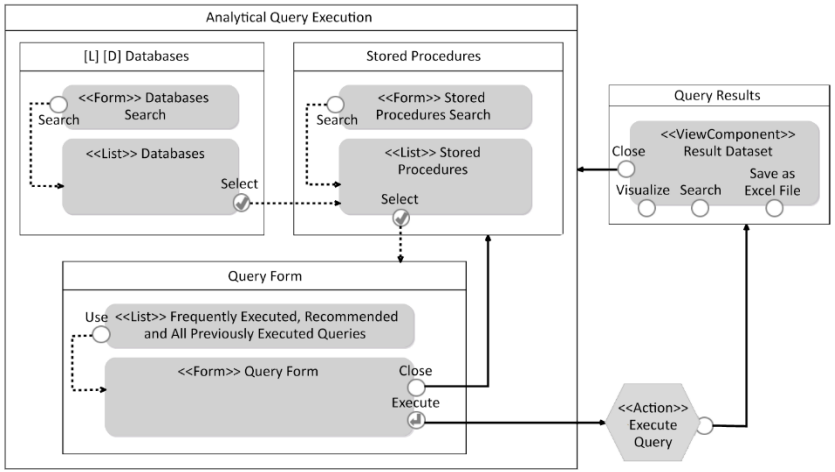




System Design

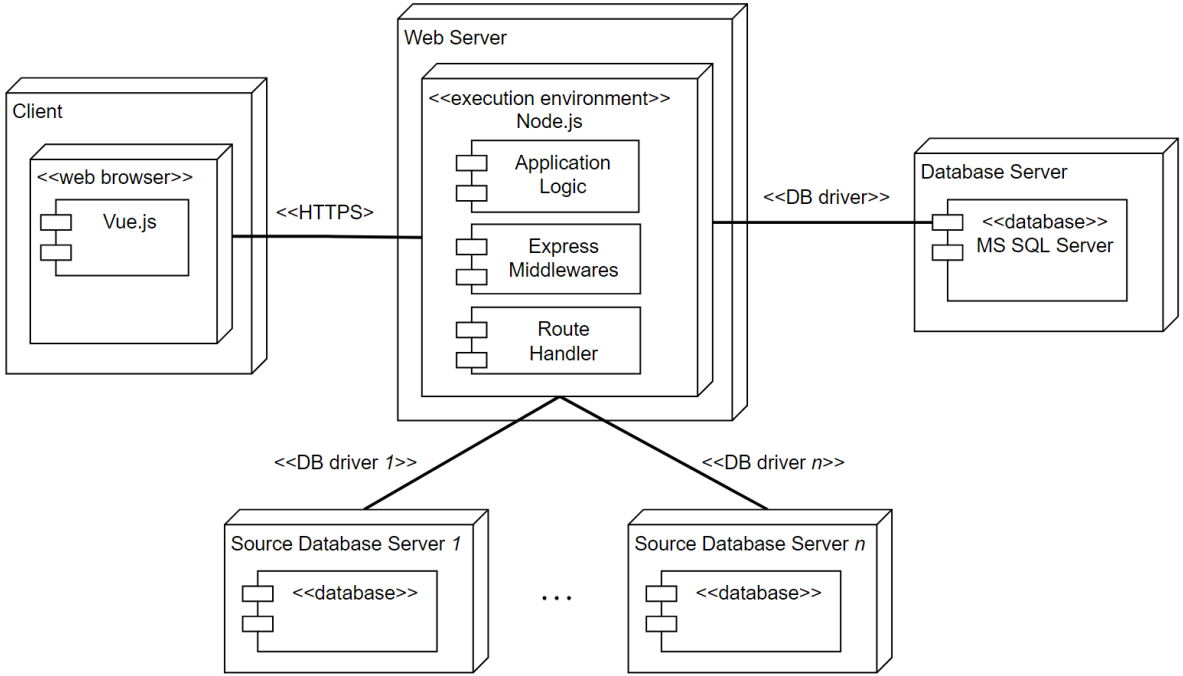
Case management model (in CMMN notation) of the interactive adaptive system for data analytics





Interaction flow model (in IFML notation) of the interactive adaptive system for data analytics

Software Architecture

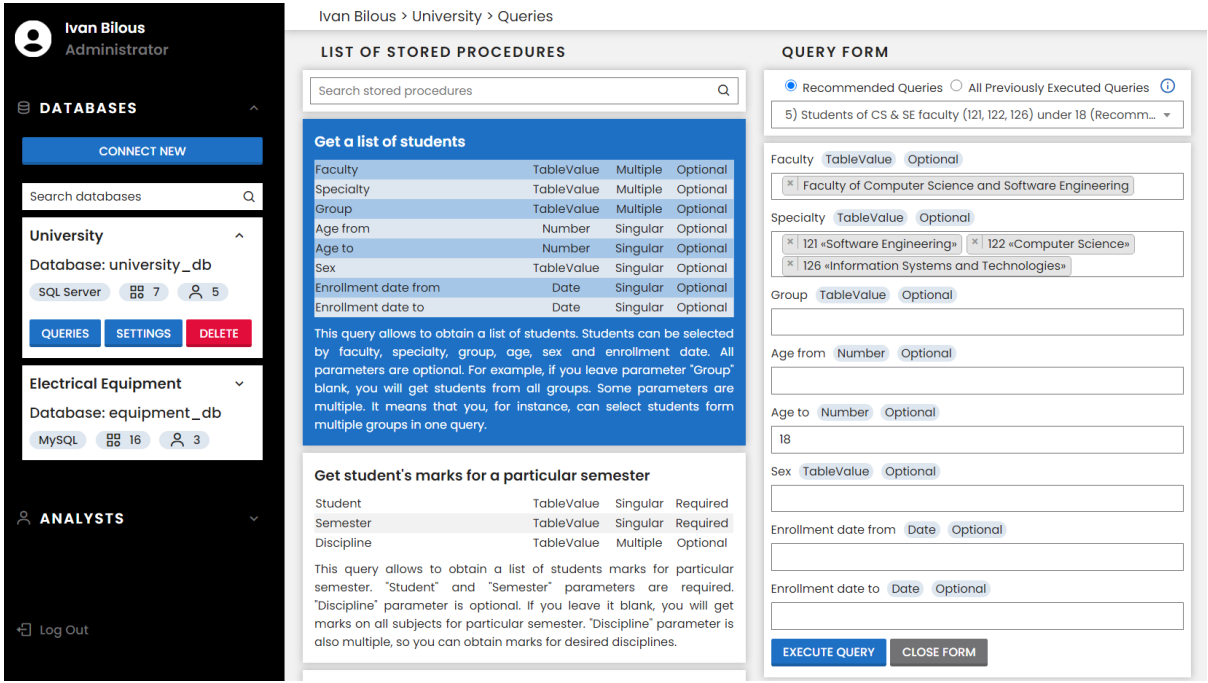


- Client tier contains only the web-browser as a “thin” client that provides user interface.
- Web server tier contains Node.js back-end application that implements business logic.
- Database server contains Microsoft SQL Server database that stores user profiles and granted permissions, data source connection properties, analytical queries, and usage statistics.

Software Prototype

Generic GUI areas are:

- Database connection properties.
- Stored procedures displayed for each of attached databases.
- Recommended queries displayed for each of stored procedures.
- Querying form that could be filled by users manually for a certain stored procedure or could be filled automatically based on suggested relevant queries.



Conclusion and Future Work

- In this study we have proposed an approach to development of interactive adaptive software tool for data analysis based on the content-based filtering.
- According to the proposed recommending procedure, the system works with SQL queries used to access attached data sources (stored procedures, to which parameter values should be passed).
- Proposed recommending procedure captures usage statistics of database queries and suggests the most relevant ones among previously used queries based on usage frequency and similarity criteria.
- The software tool is implemented as a prototype with limited functionality, which allows to connect Microsoft SQL Server databases to access provided data sources, execute SQL queries, and receive recommendations.
- In future the software should be completed and advanced filtering techniques, similarity measures, and thresholds should be considered.

THANK YOU FOR ATTENTION!