Toward Agile BI By Using In-Memory Analytics

Mihaela MUNTEAN
Academy of Economic Studies, Bucharest, Romania
mihaela.muntean@ie.ase.ro,

This paper explores one of the newer technologies related to the field of Business Intelligence: in-memory technology. The new class of in-memory BI tools turns a BI solution into an agile BI solution. Also, the paper focuses on the main data models used by in-memory BI technologies and tries to answer following questions: Which are the main characteristics of an agile data model? And, which is the best data model that can be used for enabling an agile BI solution?
Keywords: *In-Memory Analytics, "Associative" Data Model, Interactive Visualization, Associative Search, In-Memory Technology*

1 Introduction

In the last years, emerging technologies such as interactive visualization, in-memory analytics and associative search marginalized IT role in building BI solutions. Figure 1 shows the trend in use of these technologies (as search terms) using Google Trends. We see that the interest for these technologies has increased in the last years.

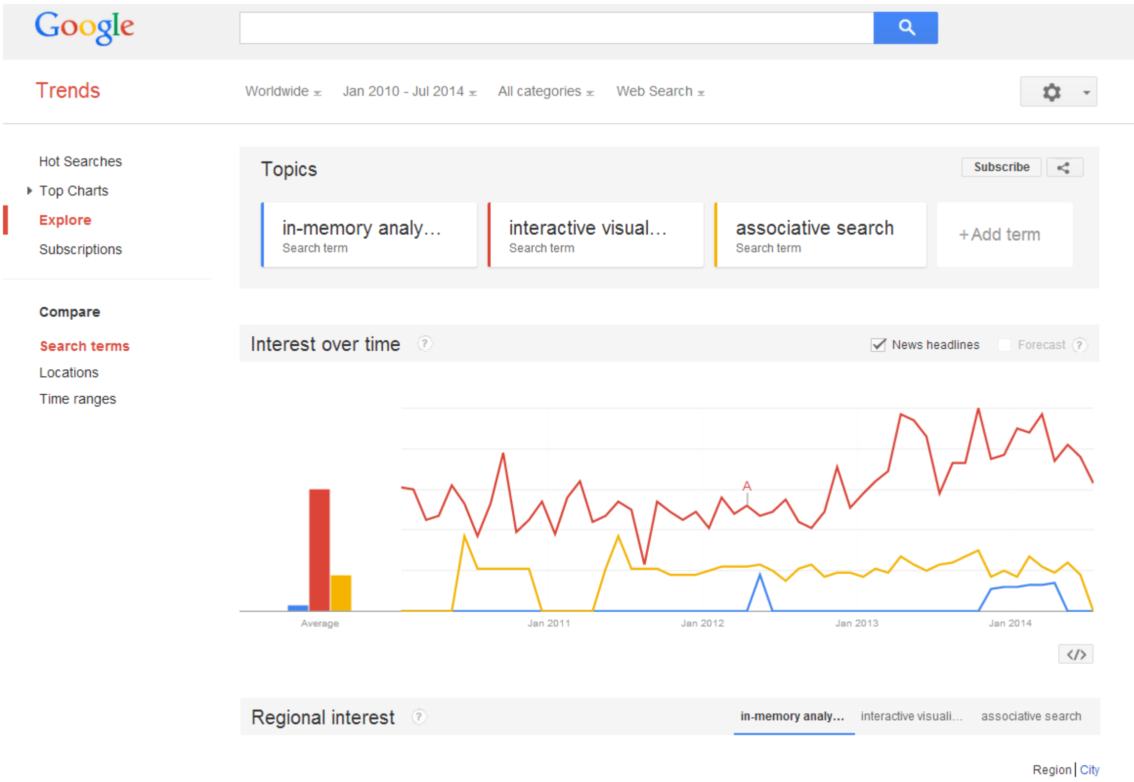

Fig. 1. In-memory analytics versus interactive visualization versus associative search. Interest over time

Also, Figure 2 shows how these technologies affect businesses. These technologies allow business people to do basic exploration of larger data sets and to find better answers to business problems. In-memory technology has the potential to help BI systems to become more agile, more flexible and more responsive to changing business requirements. This section takes a look at the pros and cons of in-memory BI. The primary goal of the in-

DOI: 10.12948/issn14531305/18.3.2014.03

memory BI technology is to replace traditional disk-based BI solutions. The important differences between them are: speed, volume, persistence and price [1].
For decades BI solutions have been plagued by slow response times, but speed is very important in analysis and in-memory BI technologies are faster than disk-based BI technologies. In-memory BI technologies load the entire dataset into RAM before a query can be executed by users. Also, most of them can save significant development time by eliminating the need for aggregates and designing of cubes and star schemas.

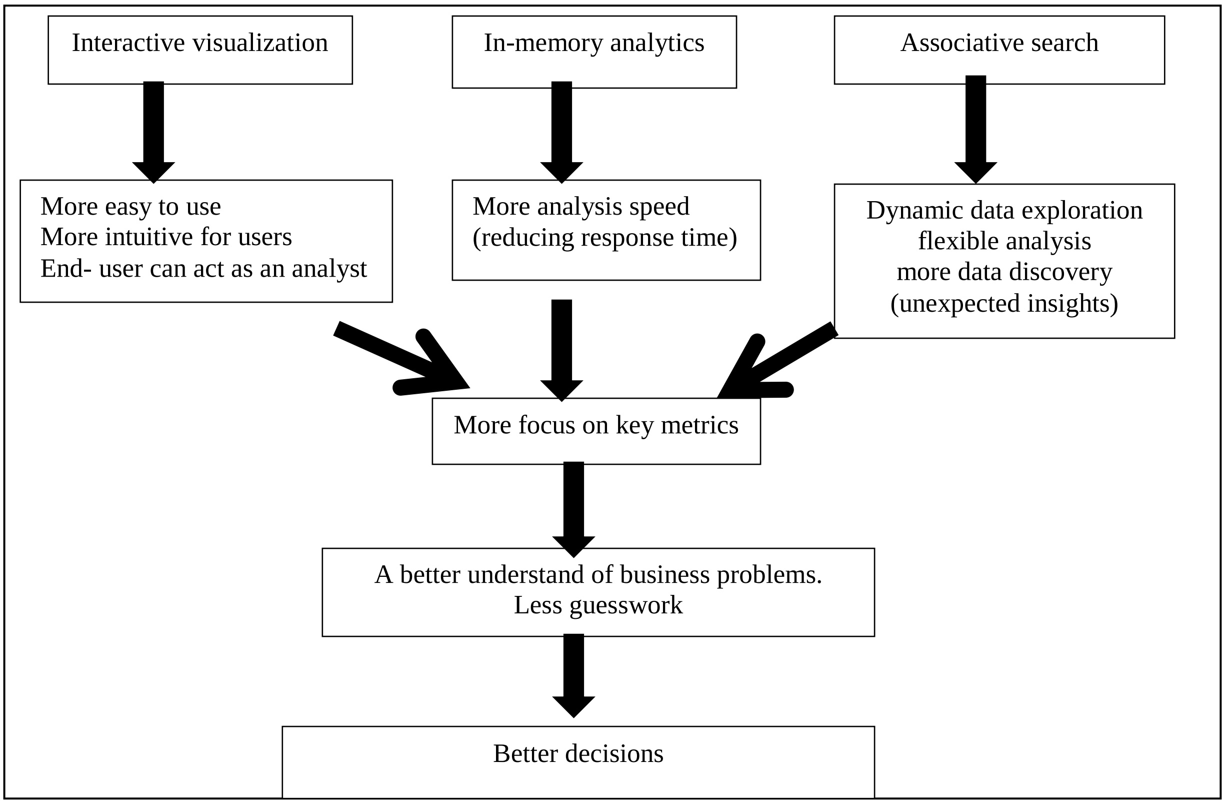

Fig. 2. How in-memory analytics, interactive visualization and associative search affect businesses

The speed of in-memory technology makes possible more analytics iterations within a given time. Ken Campbell, director of PwC Consulting Services company notes: "*Having a big data set in one location gives you more flexibility. T-Mobile, one of SAP's customers for HANA, claims that reports that previously took hours to generate now take seconds. HANA did require extensive tuning for this purpose*."[2].
But RAM is expensive compared to disk. In-memory technologies use compression techniques to represent more data in RAM. Also, most of in-memory technologies use columnar compression to improve compression efficiency.
The traditional disk-based BI solutions use query-based architectures such as: ROLAP, MOLAP and HOLAP. ROLAP uses SQL or another query language to extract detail data, to calculate aggregates and store them in aggregate tables. Detail data are stored in data warehouses or data marts (disk-based persistence) and are used when necessary. MOLAP pre-aggregates data using MDX or another multidimensional query language. HOLAP (hybrid OLAP) is a combination of the two above architectures. But these query-based solutions don't maintain the relationships among queries. Some of in-memory BI technologies can maintain the relationships among queries.
Today, one of challenges of BI is to allow users to become less dependent on IT. BI so-

DOI: 10.12948/issn14531305/18.3.2014.03

lutions must be easier to be used by all BI users. Traditional BI solutions don't provide a dynamic data exploration and interactive visualization**.** The in-memory BI tools like Qlikview, Tableau, Tibco Spotfire can simplify a larger number of tasks in an analytics workflow. The director of Visual Analysis at Tableau Software, Jock Mackinlay says "*Inside Tableau, we use Tableau everywhere, from the receptionist who's keeping track of conference room utilization to the salespeople who are monitoring their pipelines*" [2]. Tableau Software, a leader in Magic Quadrant for Business Intelligence and analytics platforms/Garter (2014) is an example of how these BI tools change the businesses.
This class of BI tools has the following characteristics: interactive visualization, self-service, in memory processing, speed of analysis, rapid prototyping and more flexibility. Using a self-service BI tool, the end-user can act as an analyst. Also, use of mobile devices and social networking inside the company promote to adopt this class of tools. For example, TIBCO Spotfire for iPad 4.0 integrates with Microsoft SharePoint and Tibbr, a social tool [www.tibbr.com/] [3]. Also, QlikView 11 integrates with Microsoft SharePoint and is based on HTML5 [4]. There are many and different in-memory BI solutions. Table 1 presents a comparative analysis using the following criteria: 1) the main characteristics; 2) query language; 3) data model [5].

Table 1. In-memory BI solutions

Solution	**Characteristics**	**Example**	**Query language**	**Data model**
In-memory OLAP	MOLAP cube and data are all in memory	IBM Cognos-Applix(TM1) Actuate BIRT Dynamic Cubes - Cognos BI version 10.1	MDX or another multidimensional query language	hypercube
In-memory ROLAP	only ROLAP metadata loaded in memory although MicroStrategy can build complete cubes from the subset of data held entirely in memory	MicroStrategy	SQL	dimensional model hypercube
in-memory columnar database with data compressions techniques	load and store data in a columnar database	Tableau Software	VizQL, a declarative language	relational/ multidimensional database less modeling required than an OLAP based solution
In memory spreadsheet	spreadsheet loaded into memory	Microsoft Power Pivot VertiPaq is the internal column-based database engine	DAX (Data Analysis Expression).	no data modeling required

DOI: 10.12948/issn14531305/18.3.2014.03

		used by Power Pivot		
In memory "associative" data model Column based storage with compression techniques (with compression ratio near 10:1)	loads and store all data in an "associative" data model that runs in memory; all joins and calculations are made in real time; less modeling required than an OLAP based solution;	QlikView includes Expressor (ETL tool)	script language is required to load the data and to transform the data; AQL technology (Associative Logic); don't use query language or definition language;	without aggregations, hierarchies, cubes; can access star scheme / snowflake / cubes;
Hybrid approach/dual format approach with data compression techniques	Relational database +columnar database; Both formats are simultaneously active;	Oracle Database In-memory a pure in-memory columnar technology Oracle Exalytics In-memory machine includes OBIEE, Oracle Essbase, Oracle Endeca Information Discovery and in-memory Oracle TimesTen database [6] SAP HANA store data in both rows and columns.	SQL	Dimensional model hypercube
Hybrid storage solution (disk + RAM)	Multidimensional model (traditional OLAP Cube) organizes summary data into multidimensional structures; Aggregations are stored in the multidimensional structure; Tabular model	SQL Server 2012 with compression algorithms and multi-threaded query processing, the Xvelocity engine delivers fast access to tabular model	MDX for multidimensional; DAX for tabular;	star schema for MOLAP; Tabular solutions use relational modeling;

DOI: 10.12948/issn14531305/18.3.2014.03

	(In-Memory Cube)	objects and data through reporting client applications such as Microsoft Excel and Microsoft Power View		

Figure 3 presents a disk-based BI solution versus an in-memory BI solution (e.g. a Qlikview BI solution).

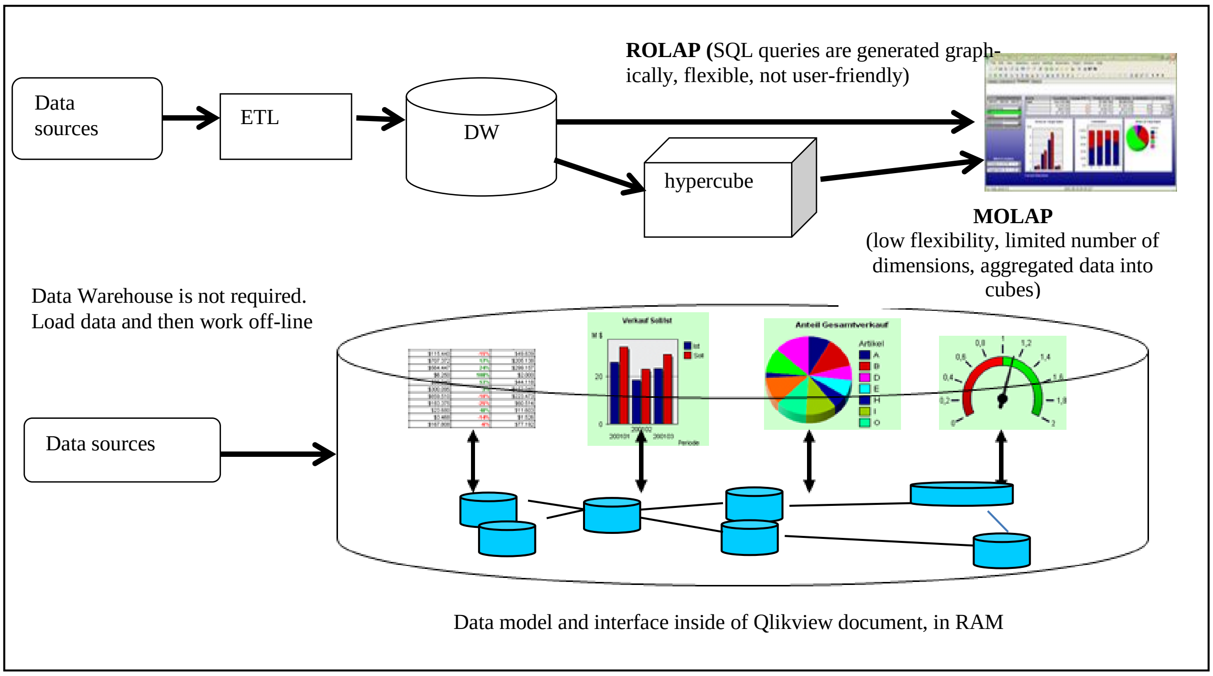

Fig. 3. A disk –based BI solution versus an in-memory BI solution

According to [5], an agile BI solution requires: 1) an agile development methodology; 2) agile BA; and 3) an agile information infrastructure. An agile information infrastructure must be able to extract and combine data from any data sources, internal and external sources including relational, semi-structured XML, multidimensional and "*Big Data*. According with these requirements, the main characteristics of a data model for agile BI are:

- adaptable to rapid business changes;
- agile design;
- high flexibility to analysis;
- excellent speed of analysis;
- easy and universal access to any data sources.

Most of in-memory BI solutions use the following data models: dimensional model (star schema, snowflake or combinations), hypercube and "associative" data model. Which of them are agile? The next section tries to answer to above question. Also, the next section briefly presents a comparative analysis of these data models using the following criteria: 1) basic concepts; 2) modeling approach; 3) flexibility.

2 Measures and Dimensions versus Free Dimensional Analysis

In dimensional model and hypercube we distinguish between measures and dimensions. The main concepts of dimensional model are: facts, dimensions, granularity and hierarchies. The main characteristics of dimensional model are:

DOI: 10.12948/issn14531305/18.3.2014.03

- the information can be classified into: facts (data elements that are analyzed) and dimensions that provide descriptive information about each fact;
- a fact is a numeric attribute of fact table. Values of facts changes continue;
- the granulation of a fact refers to the level at which the information is stored in the dimensional model;
- usually the dimensions contain static data and are not normalized;
- most dimensions have hierarchies;
- dimensions are essential for data analysis;
- the associations between fact tables and dimensional tables are defined explicitly with foreign keys;
- SQL queries a subset of tables from dimensional model. Query result sets are independent of each other.

The hypercube is a set of variables/measures, which use the same dimensions for identification. The main concepts of hypercube are: dimensions, hierarchies, hypercube cell, measures, and sparsity. In a hypercube, a dimension is represented by an axis and can have one or more members. Usually, a member can have only one parent. A member with no parent is called „root”. A node with no child is called „leaf”. Most dimensions have hierarchies. Hypercube cells contain basic measures and/or derived measures. Hypercube is implemented by multidimensional databases. The dimensional model and hypercube use predefined hierarchies for accessing and exploring data.

“Associative” model free users from the paradigm of dimensions versus measures (Figure 4). The model is implemented by Qlikview tool (Qlik Tech company, a leader in Magic quadrant for Business Intelligence and analytics platforms/Garter, 2014) [7]. *“Associative”* model makes no distinction between attributes that are facts and attributes that are dimensions. The word “*associative*” puts emphasis on understanding how datasets relate to one another. This model is built around the concept of datasets with related logic tables. The datasets are loaded in memory, in a compressed and fully normalized format, via the Load script.

The main characteristics of “*associative*” model are:

- is based on the heterogeneous sources (databases, spread sheets, Web pages and Big data). This model is persistent and reacts as a whole to user “queries”. A selection affects the entire schema. You can select any value for any attribute and all the related data from the entire data model will be displaying (associative search);
- eliminates the need to develop hierarchies, hyper-cubes and pre-aggregation of data;
- you don’t have to use a data query language;
- you don’t have to use data definition language;
- each load or select statements generate a logical table during the data load process. The associations between logical tables are generated automatically during the data load process based on matching column names across logical tables. Any fields with the same name in two or more tables will associate. The relationships among logical tables usually don’t reflect foreign key relations. The associations between logical tables are similar to full outer joins. If there is more than one field with the same name a synthetic key is created. A synthetic key contains all possible combinations of common attributes among tables. It is resource intensive and makes data model difficult to understand;
- the aggregations can be done both in the load script (pre-defined) and at the user interface development stage. This enables a user to interact with a broader range of data than will ever be possible in SQL;
- adaptable to rapid business changes and flexibility in analysis. Any value of any attribute can be the starting point of analysis;
- faster model design. The common problems of the model are: synthetic keys and circular references. There are many ways to resolve a synthetic key such as: concatenation of logic tables, using link logic tables, etc. [8], [9].

DOI: 10.12948/issn14531305/18.3.2014.03

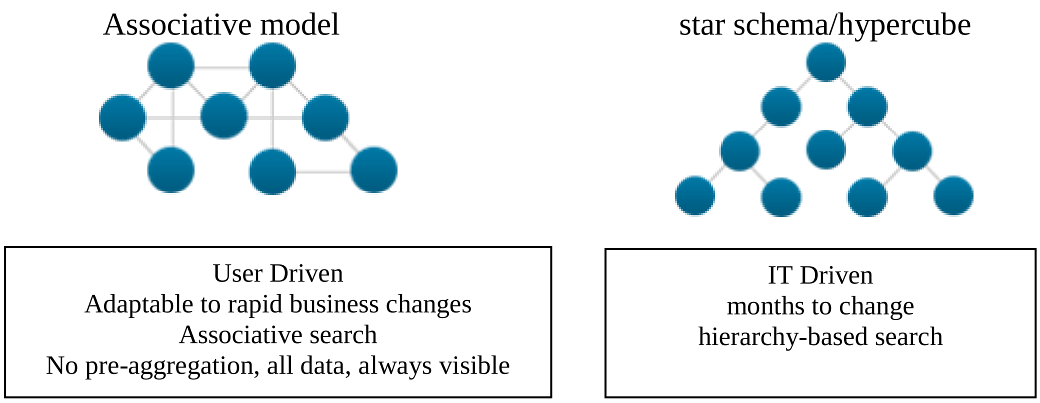

Fig. 4. Associative model versus star schema

3 Data Modeling

Dimensional model and hypercube force the user to pre-model the data before the analysis is performed (e.g. classification of data into measures and dimensions, defining the hierarchies). The dimensional model/Kimball model is a top-down model because it begins with the identification of the important processes from the company (where the data are collected). Dimensional modeling uses a waterfall approach with the following steps:

- identifying the business process/processes which will be modeled. For each identified process the will be created one or more fact tables;
- setting the granulation for each fact table;
- setting the dimensional tables for each fact table. The granulation for each dimension will also be determined;
- setting the basic and derived measures;
- setting the dimensional attributes and their description;
- how different changes in dimensions are managed (slowly changing dimensions)?
- storage of the pre-calculated aggregates in the aggregate fact tables.

Hypercube modeling is divided into the main steps:

- identifying the measures;
- identifying the dimensions and the hierarchies;
- defining the hypercube or multi-cube;
- refining the hypercube or multi-cube (e.g., defining the aggregation formulas). A data warehouse is usually built before designing the hypercube. Associations between dimensions are not computed.

"Associative" model is a bottom-up model and it is developed by each department and then adopted by the company. The model and the user interface are developed together using an agile development approach (e.g. SCRUM). This approach changes the focus from data driven to the decision driven. This approach is divided into the following phases:

- identifying the initial business requirements and the data requirements;
- prioritizing the business requirements (SCRUM product backlog) and defining the data staging requirements (QVD files for larger deployments);
- iterative execution phase (many sprints) that includes: data loading (configuring of the connections, development of the initial load script), data modeling, data provisioning, user interface development (use of the data for analysis), testing, user review and refining;
- deployment [10] (Figure 5).

DOI: 10.12948/issn14531305/18.3.2014.03

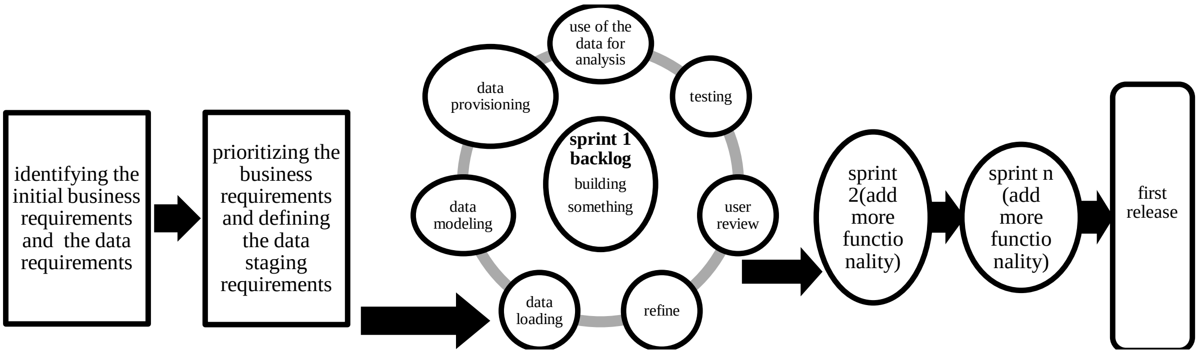

Fig. 5. An agile development approach for the "associative" data model

4 Some considerations about Advanced Business Intelligence Queries

SQL ranking and windowing aggregate functions combined with nested queries enable you to answer complex BI queries, but it is difficult for end users to write these queries. Also, MDX language, used for querying the multidimensional data stored in hypercube, is difficult for end users. For example, we have the following BI query: *"Finds the top two sellers for each city that contributes more than 5% of the sales within its region*". The data sources are *Vanzari.xls*, *Agenti.xls and Judete.csv*. We will use a partial snowflake schema with a fact table: *Vanzari* and four dimension tables: *Agenti*, *Judete*, *Articole* and *Calendar* (Time dimension). The schema was implemented in Oracle DBMS. The hierarchies are: *agenti→oras→judet→regiune*, *artId→categorieid* and *timpid→perioada→ziua→luna→an*. The excel file *Vanzari* has attribute *Data*. The attributes of *Calendar* table were defined using data functions. Data was imported from files (excel and csv). The Figure 6 shows the structure of tables.

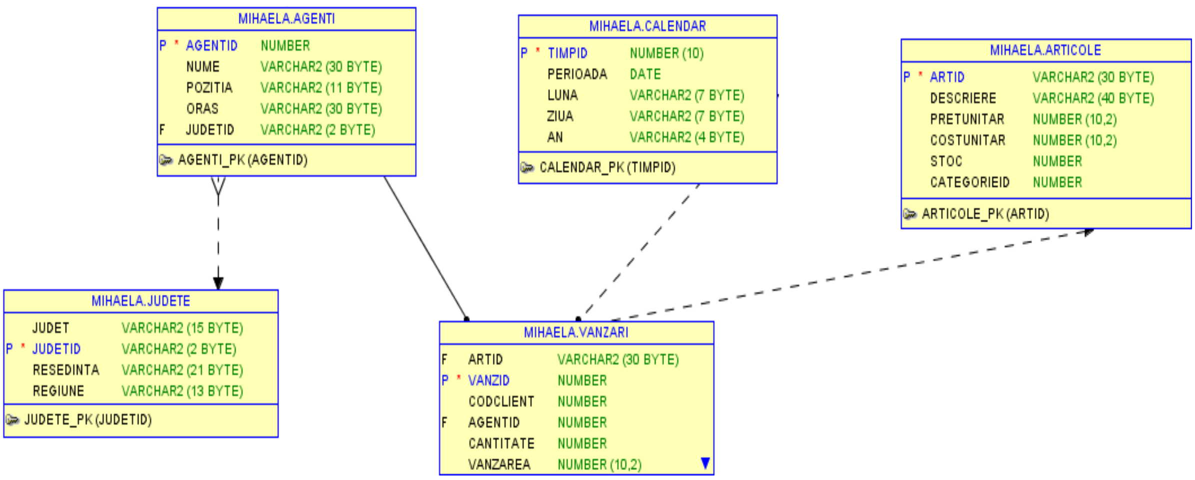

Fig. 6. The structure of tables

The definition of BI query (DBMS Oracle) is:

```
Select Regiune, Oras, Nume, Ag_Vanzari, regiune_vanzari, rang
From (Select Regiune, Oras, Nume, Sum(Vanzarea) As Ag_Vanzari, Sum(Sum(Vanzarea))
Over (Partition By Regiune) As Regiune_Vanzari,
Dense_Rank() Over (Partition By Oras Order By Sum(Vanzarea) Desc Nulls Last) As
Rang
From Vanzari V, Agenti A, Judete J, Calendar
Where V.Agentid=A.Agentid And A.Judetid=J.Judetid And V.Timpid=Calendar.Timpid
Group By Regiune, Oras, Nume Order By Regiune, Oras)
Where Ag_Vanzari >0.05* Regiune_Vanzari And Rang<=2
```

The query result is:

DOI: 10.12948/issn14531305/18.3.2014.03

Regiune	***Oras***	***Nume***	***Ag_vanzari***	***Regiune_vanzari***	***rang***
Banat	*Timisoara*	*Goaga Ion*	*113579.45*	*207446.1*	*1*
Bucovina	*Suceava*	*Mihailescu Ana*	*9983.98*	*9983.98*	*1*
……					
Transilvania	*Brasov*	*Gaman Radu*	*67348.55*	*501470.63*	*1*
Transilvania	*Brasov*	*Basarab Bogdan*	*40314.39*	*501470.63*	*2*
Transilvania	*Cluj*	*Durnea Nicoleta*	*32182.67*	*501470.63*	*1*
Transilvania	*Cluj*	*Ionescu Dan*	*27526.73*	*501470.63*	*2*
Transilvania	*Deva*	*Laszlo Toma*	*32058.95*	*501470.63*	*1*
Transilvania	*Sibiu*	*Iancu Liviu*	*25594.34*	*501470.63*	*2*
Transilvania	*Sibiu*	*Predoiu Victor*	*180418.07*	*501470.63*	*1*

This query uses:

- the hierarchy: *agenti→oras→judet→regiune*;
- dense_rank () function with logical partitions;
- nested queries.

Time dimension is not used in this query. If end-user wants to use Time dimension, then he needs to rewrite the SQL query. The query result set is independent of the previous query result set.

A large variety of powerful analytics are available with *"associative"* model such as: aggregations on-the-fly, set analysis, comparative analysis, conditional analysis, calculated dimensions, and so on. Data sources are loaded in memory, via the Load script. Database is not required. The data model and the associations between data sources are generated automatically during the data load process. The end users don't need to use a definition language. Also, the end users don't need to use a query language. They only create a pivot table with three "dimensions"/fields (*Regiune, Oras* and a "calculated dimension") and one expression: *sum (Vanzarea).* The definition of the "calculated dimension" (*Top 2 Agenti*) is:

```
=AGGR (IF (Rank (sum (Vanzarea), 4) <=2
and sum (Vanzarea)>0.05*sum (Total
Vanzarea), NumeAgent), Oras, NumeAgent)
```

Also, two list boxes were created: *An* and *Regiune*. We must select only the values of *Regiune* list (such as, *Transilvania*). The "calculated dimension" will return top 2 sellers for each city that contributes more than 5% of the sales within its region (such as, *Transilvania*), and a null value for all others (Figure 7a) [9] [11]. But, we can make every data selection (any combination of year and region). For example, Figure 7b shows top 2 sellers for each city in 2012, only for *Transilvania* region and *Muntenia* region. In conclusion, more information, high flexibility, easier for end users to make Top N analysis than with SQL/MDX.

You can select any value for any attribute and all the related data from the entire data model will be displaying (associative search). For example, when we select 2013 in the *An* list box, the screen automatically updates to show the associated data in the *Regiune* list box. The *Bucovina* region is shown with a gray background to indicate that is not associated (we have no sales in *Bucovina*, in 2013). Selection is green, unrelated data is gray and associated data is white (Figure 7c).

DOI: 10.12948/issn14531305/18.3.2014.03

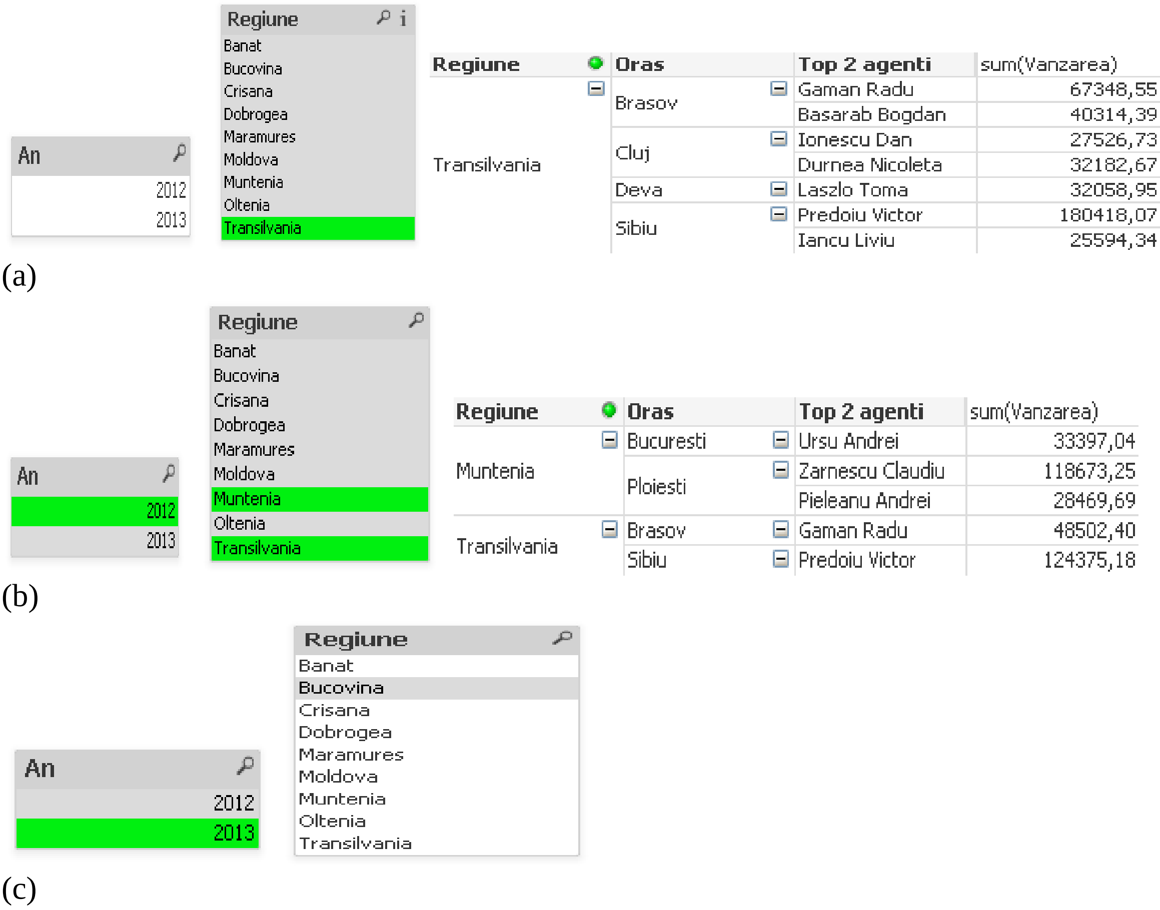

Fig. 7. An advanced BI query with "associative" data model

Also, we can select data based on associated values. For example, we want to see regions where we sold the product "*Capac protectie*". The search term will not only be checked against the *Regiune* list box, but also against the content of the entire data model (Figure 8).

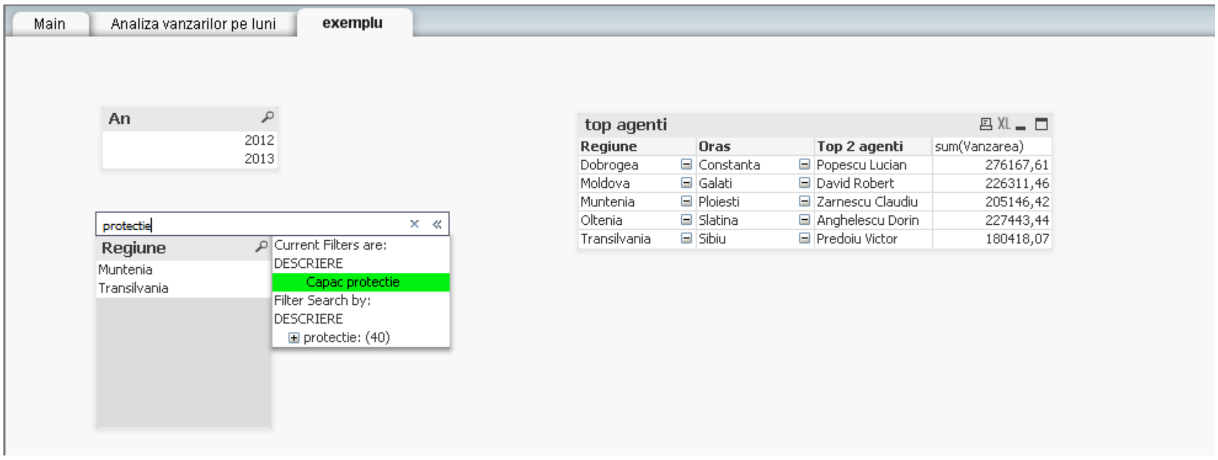

Fig. 8. Associative search

Time is very important in BI. Comparative performance metrics over a period is a fundamental task from any BI solution. Users want to easily compare different performance indicators in a period–over–period basic such as: current year-to-date indicators versus the same period last year, current month versus same month last year, current month versus

DOI: 10.12948/issn14531305/18.3.2014.03

previous month, current quarter versus previous quarter, current quarter versus same quarter last year, etc. The following script creates *Time* dimension, in associative model using data functions:

```
Let varMinDate = Num(Peek('Data',0,'Vanzari'));
Let varMaxDate = Num(Peek('Data',-1,'Vanzari'));
TempTimp:
Load
date($(varMinDate) + rowno()-1) As TempData
Autogenerate
$(varMaxDate)-$(varMinDate) + 1;
Timp:
LOAD *,
Floor(Data) as DataID,
autonumber(An&Luna, 'LunaID') as LunaID,
autonumber(An&Trimestru, 'TrimestruID') as TrimestruID;
Load
TempData As Data,
Week(TempData) As Saptamana,
Month(TempData) As Luna,
Year(TempData) As An,
Week (TempData) & '-'& Year(TempData)as SaptamanasiAn,
WeekDay(TempData) as denumire_zi,
WeekStart(TempData, 0, 0) AS prima_zi_sapt,
WeekEnd(TempData, 0, 0) AS ultima_zi_sapt,
Weekyear(TempData) as anul_pt_sapt,
Month(TempData) & '-' &Year(TempData) AS LunasiAn,
MonthStart(TempData) as prima_zi_luna,
MonthEnd(TempData) as ultima_zi_luna,
'Trim' &Ceil(Month(TempData)/3) AS Trimestru,
QuarterStart(TempData) as prima_zi_trim,
QuarterEnd(TempData) as ultima_zi_trim,
YearEnd(TempData) as ultima_zi_an,
DayNumberOfYear(TempData) as numartotalzile
resident TempTimp
ORDER BY TempData ASC;
DROP TABLE TempTimp;
```

The Figure 9 shows the associative model with *Time* dimension. We see that this model looks like snowflake schema.

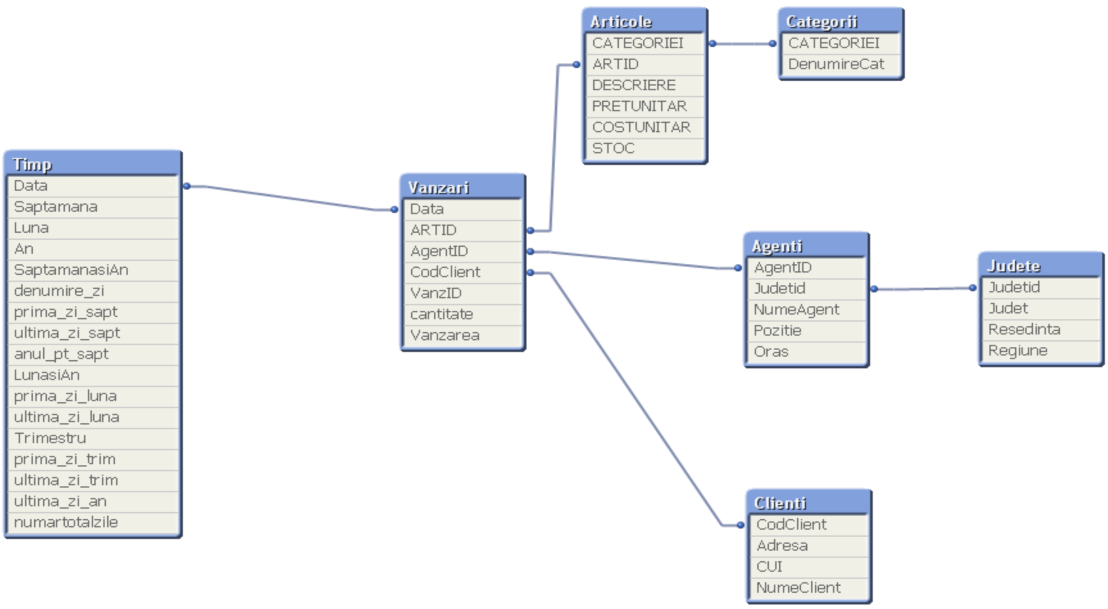

Fig. 9. Example of data model defined in QlikView

DOI: 10.12948/issn14531305/18.3.2014.03

QlikView Set Analysis is a powerful tool for comparative analysis, for example, current year versus previous year, ordered products versus products not ordered, or selected data versus the unselected data. Figure 10 shows a comparative analysis using different time periods: current month versus previous month versus two months earlier. We can compare results for three different time periods in one single view based on the same selection state. The comparisons are dynamic and based on the user's selections.

Oras	luna curenta	luna anterioara	doua luni anterioare
Arad	0,00	6.899,69	1.210,00
Baia-Mare	2.494,00	5.077,93	4.064,98
Brasov	49.436,17	9.519,54	28.241,70
Bucuresti	12.470,72	10.004,25	2.071,24
Cluj	15.190,36	8.916,03	14.936,38
Constanta	6.905,24	123.175,43	25.913,40
Deva	11.693,90	1.285,19	626,40
Focsani	0,00	1.030,32	1.397,28
Galati	73.252,56	31.413,67	0,00
Giurgiu	4.592,70	14.483,74	3.667,17
Iasi	37.072,69	3.803,45	50,30

Fig. 10. Set analysis tool-comparative analysis using different periods

Also, the Figure 11 shows a comparative analysis using alternate states between period A (february - march 2012) and period B (february-march 2013) for *Cluj* city. We can select any city, any year and any month.

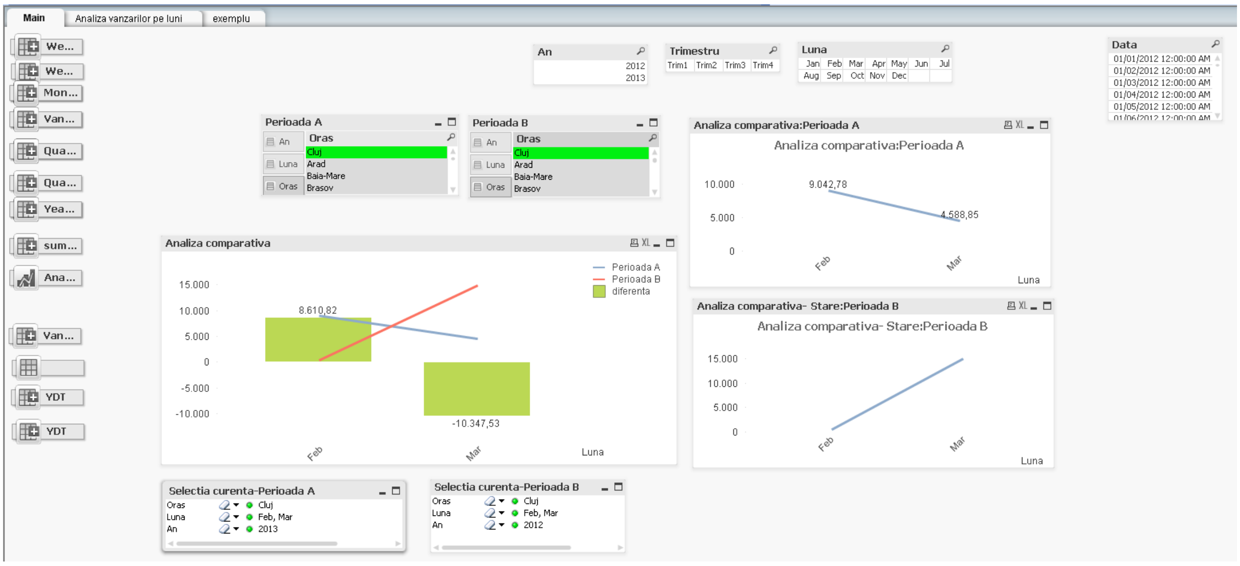

Fig. 11. Comparative analysis between different periods using alternate states

In conclusion, all three emerging technologies are implemented by Qlikview: interactive visualization, in-memory analytics and associative search.

At the end of this paper, Table 2 presents a comparative analysis between the data models and tries to show which of the models is agile. We see that the *"associative"* model has almost all characteristics of an agile data model.

DOI: 10.12948/issn14531305/18.3.2014.03

Table 2. Hypercube versus Star schema versus associative model

Criteria	**Hypercube**	**Dimensional model/ (only star schema)**	***"associative"* model**
Data modeling	Top-down	Top-down and waterfall approach Normalized fact tables and de-normalized dimensional tables	Bottom-up and **agile approach** De-normalized /normalized structures; Associative Query Logic technology infers data relationships. Exploring the associations in data, search across all data-directly and indirectly Some problems with synthetic keys and circular references
Data volume	Medium, large only aggregate date	huge detail and aggregate date	Large, only detail date, „on-fly" aggregation, no pre-aggregation, no hierarchies, no cube building Loads all data into memory (millions of rows)
Speed of analysis	Excellent for small and medium DB – **partial agile!**	Acceptable for medium and large databases	**Excellent –Agile!** Faster and more efficient than pre-packaged OLAP –based BI solution
Easy and universal access to any data sources	limited	Very good **Agile!**	A highly intuitive easy-to-use user interface Very good (Big data support) **almost Agile!**
Data integration	Very good	Very good	Very good (using script) and ETL tool
Dimensionality	Multidimensional models with 5-10 dimensions	Complex data models with many dimensions	No dimensions, no hierarchies, no cube building
Volatility	low	high	high
Adaptable to rapid business changes	No (new dimensions require hypercube rebuilding). **Not agile**	No (changes in that data model cause massive cascades of changes throughout the solution, months to change.) **Not agile**	**Yes. Agile!!!!**
Time to design	Moderate (months)	Biggest(months-years)	Low (weeks-months)-**Agile!!!**

DOI: 10.12948/issn14531305/18.3.2014.03

Query language	MDX or another multidimensional query languages	SQL or another query languages	Not available Enables real-time associative search and analysis

5 Conclusion

In conclusion this paper has identified the main characteristics of an agile data model. Considering these characteristics, the paper made a comparative analysis of the data models used in-memory BI technology: dimensional model, hypercube and "associative" model. The "associative" model has almost all characteristics of an agile data model. Also, three emerging technologies (interactive visualization, in-memory analytics and associative search) are implemented by Qliview. Deploying of these technologies in a BI solution results in an agile BI.

References

[1] M. WhiteHorn (2010), In-memory analytical systems: perspective, trade-offs and implementation, Available: http://spotfire.tibco.com/~/media/content-center/whitepapers/in-memory.pdf, accessed February, 2014

[2] A. Morrison (2012), The art and science of new analytics technology, Pwc Technology Forecast, issue 1, Available: http://download.pwc.com/ ie/ pubs/ 2012_technology_forecast_issue_1.pdf, accessed July 2014

[3] C. Kanaracus (2011), "Tibco ties Spotfire business intelligence to SharePoint, Tibbr social network, "*InfoWorld*, Available: http://www.infoworld.com/d/business-intelligence/tibco-ties-spotfire-business-intelligence-sharepoint-tibbr-social-network-178907, accessed July 2014.

[4] E. Driver (2011), "QlikView Supports Multiple Approaches to Social BI,"*QlikCommunity*, Available: http://community.qlikview.com/ blogs/theqlikviewblog/ 2011/06/24/ with-qlikview-you-can-take-various-approaches-to-social-bi, accessed January 2014

[5] M. Muntean, T. Surcel, "Agile BI – The Future of BI", *revista Informatica Economică* vol. 17, no. 3/2013, Available http://revistaie.ase.ro/content/67/10%20-%20Muntean,%20Surcel.pdf

[6] Oracle white *paper* (2011), *Oracle Exalytics In-Memory Machine: A Brief Introductio*n, Available: http://www.oracle.com/us/solutions/ent-performance-bi/business-intelligence/ exalytics-bi-machine/overview/exalytics-introduction-1372418.pdf, accessed July 2014

[7] R. L. Sallam, J. Tapadinhas,J. Parenteau, D. Yuen, B. Hostmann (2014), "Magic Quadrant for Business Intelligence and Analytics Platforms", ID:G00257740, Available http://www.ioz.pwr.wroc.pl/Pracownicy/lubicz/DSS/HOMEWORK/Magic-Quadrant-for-Business-Intelligence-and-Analytics-Platforms.pdf, accessed June 2014

[8] A Qlikview Technology White Paper (2010), The associative experience: Qlikview's overwhelming advantage, Available:http://www.litcosys.ca/images/WP-The-Associative-Experience-EN.pdf, accessed January 2014

[9] QlikTech International (2010), Qlikview Reference Manual, vers. 10, Internet: http://www.qlik.com/, accessed January 2014

[10] B. Sterry and J. Nieuwenhuijs (2011), Qlikview project methodology handbook v.1.0,Available:http://www.slideshare.net /AntoninoBarbaro/qlik-view-project-methodology-handbook-v-10-docx, accessed January 2014

[11] S. Redmond, "*Qlikview for developer cookbook*", Birmingham: [Packt] Publishing Enterprise, 2013, pp.106-108, 123-126, 153-157 and 369-400

DOI: 10.12948/issn14531305/18.3.2014.03

Mihaela MUNTEAN is associate professor in Economic Informatics Department, Faculty of Economic Cybernetics, Statistics and Informatics, Academy of Economic Studies of Bucharest. She received her doctoral degree in Economics in 2003. Since 1997 she is teaching in Academy of Economic Studies, in Economic Informatics Department. She is interested in Databases, Information &Communication, OLAP technology, Business Intelligence Systems and Economic Information Systems Design. She published over 50 articles in journals, over 30 scientific papers presented at national and international conferences, symposiums and workshops and she was member over nine research projects. She is the author of two books and she is coauthor of seven books.

DOI: 10.12948/issn14531305/18.3.2014.03