INFORMATICA, 2015, Vol. 26, No. 2, 283–312

© 2015 *Vilnius University*
DOI: http://dx.doi.org/10.15388/Informatica.2015.49

Multiple-Criteria Approach to Optimisation of Multidimensional Data Models

Igor KORELIČ[1]*, Violeta MIRCHEVSKA[3], Vladislav RAJKOVIČ[2], Mirjana KLJAJIĆ BORŠTNAR[2], Matjaž GAMS[3]

[1]*Result, Računalniški sistemi d.o.o., Slovenia*
[2]*Faculty of Organisational Sciences, University of Maribor, Slovenia*
[3]*Jožef Stefan Institute, Slovenia*
e-mail: igor.korelic@result.si

Received: June 2013; accepted: March 2014

Abstract. This paper presents a novel approach to the adaptation of multidimensional data models to user-specific needs. The multidimensional data models used in contemporary business-intelligence systems are inherently complex. In order to reduce the complexity of these models, we propose using a qualitative multiple-criteria decision modelling method that is based on using a hierarchical tree of the criteria to decompose the larger problem into a group of smaller problems. The final value is derived by aggregating the criteria values using simple "if-then" rules, which form the knowledge-based expert rules in the hierarchical criteria tree that reflect users' preferences. The multiple-criteria analysis of the multidimensional model structure results in a multidimensional model that exhibits a reduced complexity and is adapted to users' needs. The model was validated using sales data from a medium-size enterprise. The qualitative (through questionnaires) and the quantitative (through usage mining) evaluation of the proposed methodology both showed that the proposed approach increases the ease-of-use of business intelligence systems and also contributes to a higher user satisfaction.

Key words: data warehousing, multidimensional data model, automatic construction, user profile oriented, multiple-criteria decision analysis.

1. Introduction

Although business intelligence (BI) products provide a wide spectrum of functionality, they are not "one-size-fits-all" solutions. Moss and Shaku (2003) estimated that 60% of BI projects either fail or are abandoned. According to Gartner (2008), 10–20% of business users used their BI tools proactively in 2008. According to another study (DecisionPath, 2012), only 10% of organisations did not have BI projects, but only 20% of organisations stated that BI was adopted throughout the entire organisation. One of the reasons for this low utilisation of BI projects is the fact that BI systems are considered to be highly technical. However, users of BI tools demand products that are easy-to-use and flexible and are accessible through handheld devices; such tools should incorporate one-third of the

*Corresponding author.

BI functionality (Gartner, 2012). BI systems should become more human-centric by considering the impact of information systems on humans and the integration of information systems and human work (El Sawy, 2003).

Even though not often referenced in the literature, practice shows that BI users rarely take full advantage of BI systems. Johnson (2002) estimated that 64% of BI system functionalities are never or rarely used. Korelič and Škedelj (2008) showed that 80% of searches in BI systems are performed on 20% of the data. Those studies have shown that BI systems need to be context- or user-aware; that is, provide the same services in different ways in accordance with the usage context. The present paper focuses on the adaptation of the multidimensional data models (MDMs) of BI systems to the needs of the user.

We use an example to present the need for this adaptation. At the reporting level, a typical BI system contains a predefined set of MDMs that are grouped according to the business segment they serve (such as sales, production, and finances). With the aim of covering all of the information related to a business segment, one of the following approaches is used: (1) a small number of MDMs with a large number of dimensions and measures (approximately 30 dimensions and 20 measures) is created for each segment; (2) a larger number of MDMs with a small number of dimensions and measures (approximately five dimensions and three measures) is created for each segment; or (3) a combination of (1) and (2).

Although all business data is integrated by this system, users must exert effort to find the information they need. This is especially true for busy business users who, unlike power users, do not have deep knowledge of the underlying structure of the MDMs or do not have experience with BI systems. Several author (Pendse, 2006; Korelič and Škedelj, 2008; Finucae *et al.*, 2010) have shown that business users perceive such BI systems as complex and therefore do not adopt them. Although not referenced in the literature, our practical experience shows that BI users can be separated into a small number of groups according to their needs. In sales-oriented companies, for example, BI users can be divided in the following three groups: (1) operative sellers, (2) regional sales supervisors, and (3) strategists. Each group typically uses a subset of MDMs, which represent data with a subset of dimensions and measures at a concrete level of detail. By presenting users with only a subset of MDMs, which have customised dimensions and measures in accordance with their needs, it is possible to obtain high-quality information in a straightforward manner.

Figure 1 presents the MDM set of a typical BI system before (MD^{in}) and after the adaptation to user needs ($MD^{out\text{-}opt\text{-}ut}$) proposed in the paper. The MDM set denoted as MD^{in} contains a large number of MDMs from all business segments. The number of dimensions and measures per MDM varies from small ($\leqslant 5$, $\leqslant 3$ in most cases) to large ($\geqslant 30$, $\geqslant 20$ in some cases). The MDM set denoted as $MD^{opt\text{-}out\text{-}ut}$ presents a BI system that has been adapted to the needs of regional sales supervisors. The system contains five pivot tables, each of which has up to three dimensions and two or three measures per view (with aggregates).

In this study, we present a novel methodology denoted Multidimensional Data Model Profile Adaptation (MDM-PAD), which was designed to adapt multidimensional data

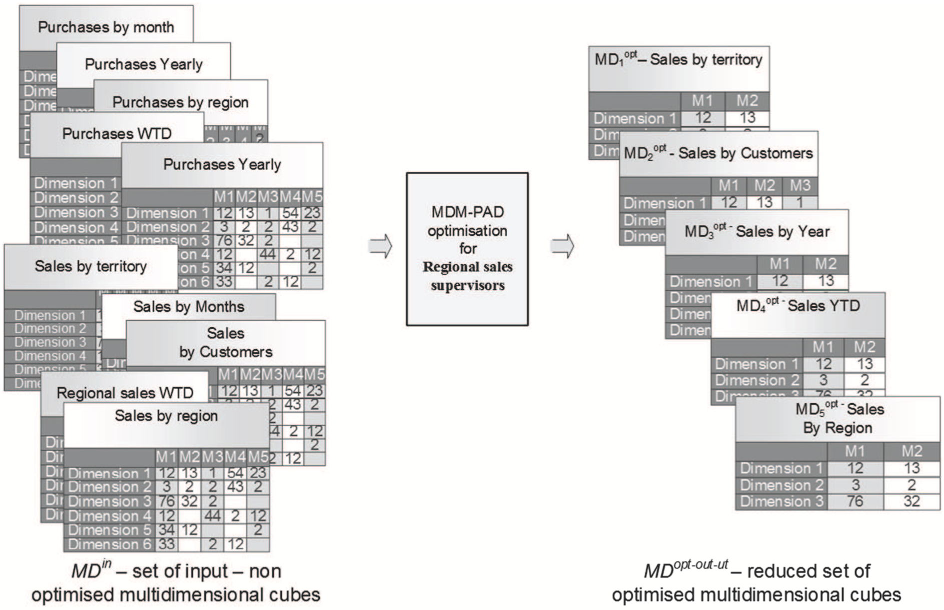

Fig. 1. Typical BI system covering all business segments and presenting all of the business data, before and after adaptation.

models to user needs; that is, to create the adapted set $MD^{out\text{-}opt\text{-}ut}$ from MD^{in}. First, the user types are determined, each of which is represented by a user profile. The user profile contains the typical information needs of a particular user type. The set of MDMs adapted to the information needs of the users in each user type ($MD^{out\text{-}opt\text{-}ut}$) is then determined through qualitative multiple-criteria decision analysis. This is followed by evaluation of the system performance. The user feedback, both qualitative using questionnaires and quantitative through an analysis of the view click rate and the dwell time, is used to better understand the users' needs and provide an improved adaptation. The test results show that MDM-PAD increases the ease-of-use of BI systems and contributes to higher user satisfaction.

The paper is organised as follows. Section 2 summarises previous work in this field, before Section 3 presents the proposed methodology. Section 4 evaluates MDM-PAD, and Section 5 discusses the findings and open questions that need to be addressed in future work.

2. Related Work

BI systems must be context-aware; that is, provide the same services, in different ways, in accordance with the usage context. This problem has been addressed by several authors. Golfarelli *et al.* (2011), and Bellatreche *et al.* (2005) customised the presentation of on-line analytical processing (OLAP) query results by sorting them according to the user preferences, which are expressed qualitatively; that is, as binary relations on the space of tuples. Sarawagi (2000) also addressed the customisation of the presentation of OLAP query results and developed a tool for the enhanced exploration of OLAP data that is adaptive to the user's prior knowledge of the data. Duan *et al.* (2011) developed a method for ranking query results according to the history of the user's actions in a sequential decision-making domain.

In addition to adapting BI systems to the usage context, the literature has also discussed various approaches to assisting users during OLAP analysis sessions by recommending executable queries. Jerbi *et al.* (2009) generated query recommendations based on user preferences. Giacometti *et al.* (2008) analysed executed sessions on an OLAP server. Sarawagi (1999) presented an automatic explanation of value changes (decreases or increases) of an aggregated measure.

MDM-PAD also adapts OLAP query results (MDMs) to a particular user. It also filters the set of available MDMs based on what the specific user type typically analyses. In contrast to the previously described approaches, which help users who are familiar with the multidimensional data model and OLAP analysis, our methodology is primarily beneficial for new users with little or no experience with the underlying BI system technology. For experienced users, the benefit of our method is shown in cases when new cubes are introduced to the system or significant changes to existing cubes are introduced to the BI system.

2.1. *User Profiles*

The techniques for automatic user profile extraction can be divided into three groups: content-based filtering (Basilico and Hofmann, 2004), collaborative filtering (Wang and Lin, 2002), and usage mining (Deshpande and Karypis, 2004). Depending on how the user profiles are used to provide customised content, two approaches can be distinguished: user segmentation and personalisation. In MDM-PAD, the content customisation is based on user segmentation in order to provide content that is customised to specific user features. The low number of users in the average analytical BI system means that standard user profiling methods are not suitable for this domain. We present a method for user profile extraction that is appropriate for the analytical domain of BI systems.

2.2. *Multiple-Criteria Decision Methods*

Multiple-criteria decision methods (MCDM) play an important role in decision support systems (DSS). In this paper, we have used a qualitative method called DEX (Bohanec and Rajkovič, 1990), which was upgraded to DEXi (Bohanec, 2008). DEX has been applied to real-life decision problems in various fields, including transportation (Cundrič *et al.*, 2008), agriculture (Pavlovič *et al.*, 2011; Rozman and Pažek, 2005), ecological modelling (Žnidaršič *et al.*, 2006), industrial applications (Bohanec and Rajkovič, 1999), environment (Kontič *et al.*, 2006), and food processing (Rozman *et al.*, 2006).

As is the case with our approach, there are implementations of MCDM that utilise MCDM in combination with other methodologies. An example of this type of implementation was presented by Barfod *et al.* (2010), who combined MCDM with cost-benefit analysis to evaluate partial results from both methods before obtaining the final decision. The study conducted by Bouyssou *et al.* (2006) exhibits some similarity to the approach in the present paper because it combines different methodologies with MCDM. Hung *et al.* (2011) used MCDM in the domain of knowledge management for SMEs. MCDM combined with data integration and data mining has been used for incident information management (Peng *et al.*, 2011). Turskis and Zavadskas (2010) developed the ARAS-G method for resolving grey areas for criteria evaluation. Zavadskas *et al.* (2009) conducted similar research applying the COPRAS-G method, while Stanujkic *et al.* (2013) used a ranking procedure to determine performance rating of alternatives.

The MCDM methodology was also used for resolving classification problems (Chen, 2006) as a method that simplifies complex models using ranging, weighting and grouping technologies to create homogeneous groups (Chen, 2006; Roberts and Goodwin, 2002; Žnidaršič and Bohanec, 2007). Our method uses more models but fewer nodes and attributes (rules).

2.3. *Methods for Optimising or Enhancing Multidimensional Data Models*

Some authors have introduced innovative methods for optimising or enhancing multidimensional data models (MDM). Espil and Vaisman (2002) introduced Intensional Redefinition of Aggregation Hierarchies (IRAH), which concentrates on methods that redefine single dimensions, but differs from our method in that it does not take user profile parameters into consideration. Mansmann and Scholl (2007) proposed methods for enhancing MDMs by improving the navigational hierarchy, homogeneity, an unbalanced and irregular hierarchy, and summarisability problems. Although these enhanced methods have resulted in an increased capacity of the OLAP technology, they have not addressed the differences between different user types.

Malinowski and Zimanyi (2006) proposed the adoption of the conceptual perspective and used ER-like graphical notations to design multidimensional models. Other authors proposed UML-like approaches (Lujan *et al.*, 2006; Prat *et al.*, 2006) or special text and graphical notations (Dori *et al.*, 2008; Pardillo *et al.*, 2010) to express the semantics of the multidimensional model. In addition, some researchers have developed fully automated methods for generating MDMs (Romero and Abelló, 2010). Some researchers (Dzemyda and Sakalauskas, 2011) have used heuristic methods in areas of large data for predictive learning. Although these approaches improve MDM design to make it faster and better defined, these methods require special domain knowledge that only data warehouse and professional IT personnel have. Therefore, these approaches differ from ours in that they cannot be automated or adapted to all user types.

Other researchers have proposed enhancements of relative multidimensional scaling method, for visualisation of multidimensional data in order to lower the difficulty of understanding the multidimensional data, such as different techniques for exploratory analysis of multidimensional data (Žilinskas, 2008), dimensionality reduction methods (Karbauskaite and Dzemyda, 2009), multimodal evolutionary algorithm (Redondo *et al.*, 2012) and advanced modification of the base method (Bernatavičienė *et al.*, 2007; Žilinskas and Podlipskyte, 2003).

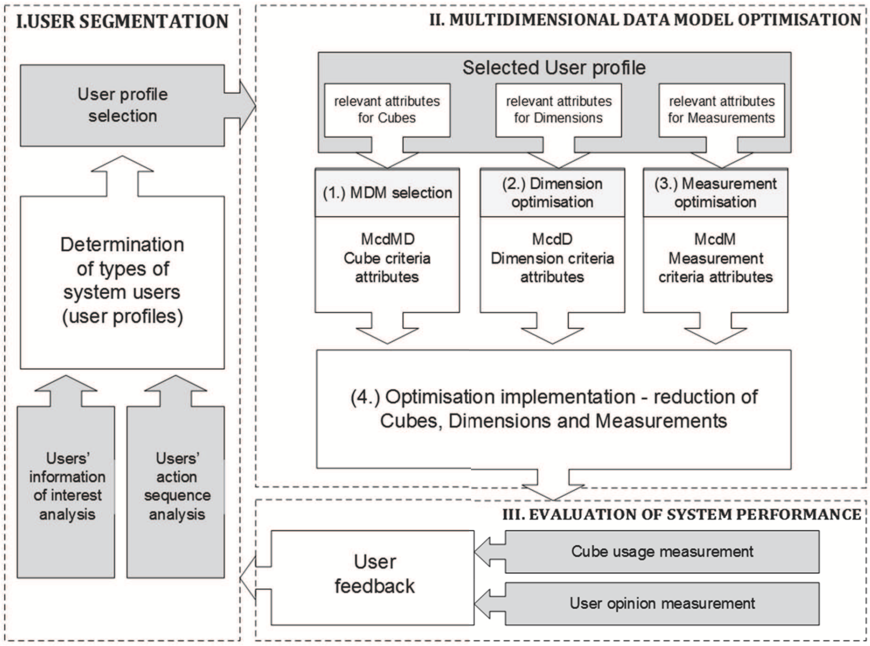

Fig. 2. Multidimensional Data Model Profile Adaptation (MDM-PAD).

3. Multidimensional Data Model Profile Adaptation (MDM-PAD)

This section presents the MDM-PAD methodology for adapting multidimensional data models to user-specific needs using the multiple-criteria decision modelling methodology DEX. The methodology (Fig. 2) can be divided into three parts: (I) user segmentation, (II) multidimensional data model optimisation, and (III) evaluation of the system performance. Part I determines the types of system users, and the categorisation is based on analyses of the users' interest and the users' observed action sequences. Each user type is represented by a user profile. Part II determines the optimal data presentation for each user type through multiple-criteria decision modelling and MDM operations. Three optimisations steps are performed for each user type using separate multiple-criteria decision models: (1) McdMD – MDM selection model, (2) McdD – dimensions optimisation model, and (3) McdM – measures optimisation model. Part III closes the loop that enables iterative improvement of the alignment of the MDMs in the BI system to different user types. Direct (through questionnaires) and indirect (through usage mining) user feedback provides a measure of the relevance of the optimised MDMs for each particular user. This feedback makes it possible to refine the user segmentation part of the method, which helps improved MDM optimisation. Because the main goal of the evaluation part is to collect additional system usage data, thereby improving MDM-PAD's input information, this section focuses on the user segmentation part (Section 3.1) and the multidimensional data model optimisation part (Section 3.2).

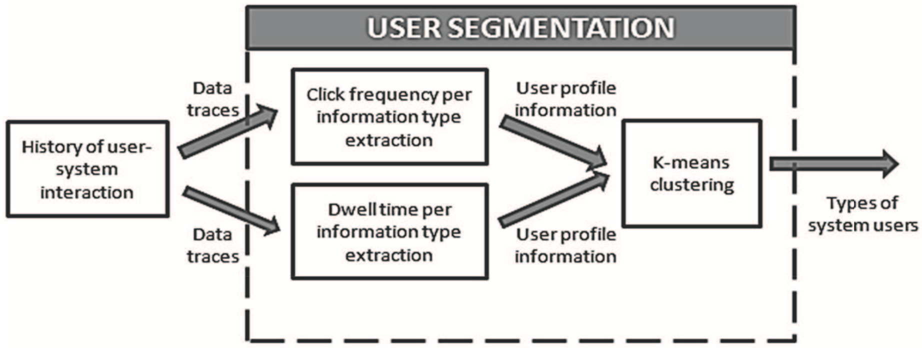

Fig. 3. MDM-PAD user segmentation for determining user types.

3.1. *User Segmentation*

The procedure for providing customised content to users in accordance to their information needs and preferences is based on user segmentation. Figure 3 presents the approach used to determine the user types.

The first step is the automatic extraction of the information needs and preferences of the system users encoded in user profiles.

Definition 1. A user profile is a vector of attributes $Pu = (attr_1, attr_2, \dots, attr_N)$, which represent the degree of interest of the user in a specific piece of information.

For example, the attributes in the user profile vector may represent the level of detail at which the user analyses the data, the time period that the user typically utilises when examining the data, and the data type (for example, sales per items, sales per customers). The level of interest that a user has in the specific pieces of information encompassed by the user-profile attributes is determined using the following aspects:

- Click frequency – the frequency of clicks on MDMs that provide a specific piece of information (expressed as a percentage of all of the clicks).
- Dwell time – the time that a user spends viewing MDMs that provide a specific piece of information (expressed as a percentage of all of the system's usage time).

The degree of interest is a value in the range [0, 1], where higher values indicate higher degrees of interest. With users' information needs and preferences encoded in user profiles, we apply clustering (Witten and Frank, 2011) to obtain the user types. User segmentation in MDM-PAD uses the k-means clustering algorithm, which outputs a set N of clusters, each of which represents one type of N BI-system users $UT = \{ut_1, ut_2, \dots, ut_N\}$, where N is the number of user types. The cluster centroids represent the typical information needs and preferences of the users that belong to each user type. A symbolic description of each user type is assigned by a domain expert based on cluster centroid analysis.

3.2. *Multidimensional Data Model Optimisation*

The next step in our method is the optimisation of the MDMs. The optimal MDMs for each user type are determined using the multiple-criteria decision modelling methodology DEX. The purpose of this step is to adjust the MDMs to the user type in order to ultimately optimise the BI system usage (for example, fewer clicks and less time to obtain the answer). Section 3.2.1 introduces MDMs and then Section 3.2.2 presents DEX. We then describe the MDM optimisation method (Section 3.2.3).

3.2.1. *The Multidimensional Data Model*

The format of historical business-data snapshots is defined using MDMs (generalised, conceptual data models). We use an extension of the Thomas and Datta MDM model (Thomas and Datta, 2001), which can handle uncertain and incomplete measures, as proposed by Moole (2003), and uncertainty in dimensional hierarchies using fuzzy logic (Delgadom *et al.*, 2004).

Definition 2. A multidimensional data model MD_i is a seven-tuple $MD_i = \langle C, A, f, d, O, L, H\rangle$, where:

- C is a set of m characteristics that includes all of the concepts needed for analysis; that is, all of the dimensions and measures;
- A is a set of t elements (attributes) encompassing all of the values on all of the levels of abstraction that may be assigned to a characteristic;
- f is a one-to-one mapping, $f : C \to 2^A$, which defines the set of attributes that may be assigned to each specific characteristic in C;
- d is a Boolean-valued function, $d(C) : C \to 2^C$, that partitions C into a set of dimensions *DS* and a set of measures *MS*, where measures are designated with the value TRUE and dimensions with the value FALSE;
- O is a set of hierarchies, one for each characteristic. It specifies how the elements of each characteristic, specified by A, are organised into levels with respect to their generality with the use of fuzzy hierarchical relations. Fuzzy hierarchical relations are used as proposed by Delgadom *et al.* (2004);
- L is a set of cube cells, each having the structure ⟨address, (content, probability stamp)⟩, represented by $\langle L.AC, (L.CC, L.PS)\rangle$ as proposed by Moole (2003);
- H an element of type history of the form

$$H = \begin{cases} \Omega \\ (A, f, O, V, G, H'), \end{cases}$$

  where Ω is a recursive closure, A is a set of attributes, f specifies which attributes are related to each characteristic, O is a set of characteristic hierarchies, V specifies how the cell values are obtained, G is an aggregation operator and H' is an element of type history.

Depending on the outcome of the utility functions, the following multidimensional data model operations were used in MDM optimisation: *roll-up* (increasing the level of aggregation along one or more dimension hierarchies), *slice* (selecting dimensions to be shown in the hypercube), and *dice* (restricting the values of one or more dimensions based on conditions specified in the form of predicates).

3.2.2. *DEX*

We used DEX to define the optimisation parameters for cube optimisation (Bohanec and Rajkovič, 1990). DEX has the following characteristics (Bohanec, 2003): (a) the model consists of hierarchically structured variables called attributes, (b) all of these attributes are qualitative rather than numerical, (c) the attributes can take only a finite (and usually a small) number of discrete symbolic values, and (d) the aggregation of values in the model is defined by rules. The distinguishing characteristic of DEX is its ability to utilise qualitative variables instead of numerical variables, which typically constitute traditional quantitative MCDM models. In the DEX method, the utility functions are represented by decision rules, which are typically formulated by decision makers or domain experts. DEX models are hierarchically structured into a tree of attributes and all MDM_PAD models are also trees. A tree represents the structure of a decision problem. Leaves of the decision tree represent basic attributes, while internal nodes are used to represent aggregate attributes. The models are built using two steps. In the first step, the domain expert develops the structure of the particular MCDM model using the DEX decision tree editor. The second step involves defining simple if-then decision rules for each internal node of the decision tree. DEX provides a number of utility functions to support the creation, verification and analysis of decision trees and decision rules.

Definition 3. A DEX multiple-criteria decision model is a qualitative decision-tree model that decomposes the decision problem to a set of smaller, less-complex sub-problems, such that each decision sub-problem is resolved by a set of if-then rules.

3.2.3. *MDM Optimisation Method*

The aim of the MDM optimisation method is to determine the most suitable data representation for each user type identified in the first part of MDM-PAD (user segmentation). As input, the optimisation method uses a set of M predefined MDMs $MD^{in} = \{MD_1^{in}, MD_2^{in}, MD_3^{in}, \dots, MD_M^{in}\}$ that cover all of the information related to the business segment of interest (for example, sales and marketing), where $MD_i^{in} = \langle C^{in}, A^{in}, f, d, O^{in}, L, H\rangle$.

The method then outputs a set of N adapted MDM sets

$$MD^{opt\text{-}out} = \{MD^{opt\text{-}out-1}, MD^{opt\text{-}out-2}, \dots, MD^{opt\text{-}out-N}\},$$

where N is the number of extracted user types and $MD^{opt\text{-}out\text{-}ut}$ is the set of adapted MDMs for the user type ut. $MD^{opt\text{-}out\text{-}ut}$ contains an optimised subset of the MDMs from

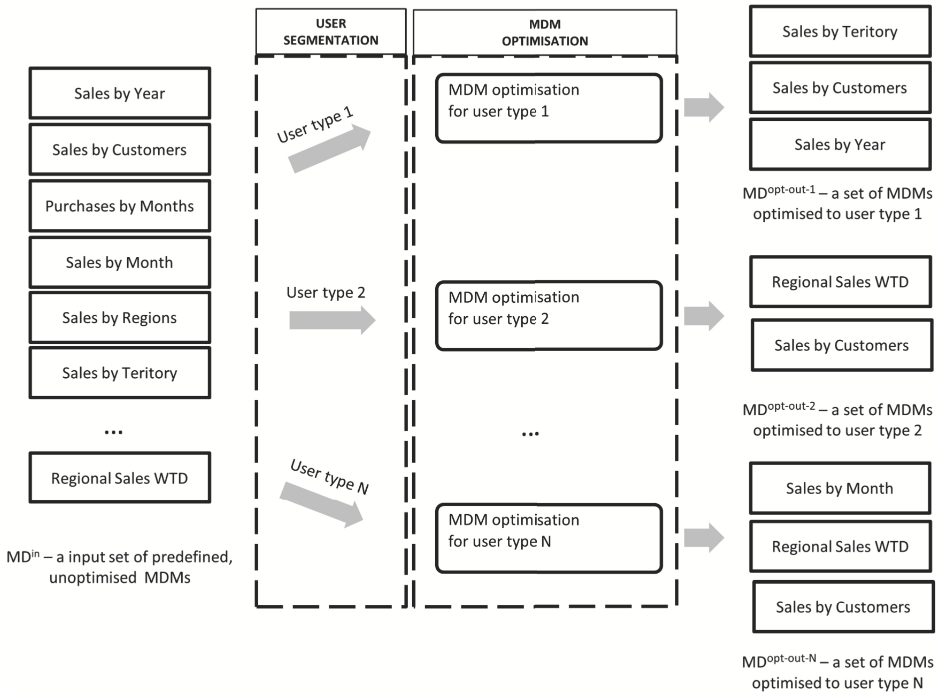

Fig. 4. MDM optimisation – global view.

the input set MD^{in} with MDM characteristics C^{out} and attributes A^{out} adapted to the needs and preferences of the users of user type ut; namely,

$$MD_i^{opt\text{-}out\text{-}ut} = \langle C^{out\text{-}ut}, A^{out\text{-}ut}, f, d, O^{out\text{-}ut}, L, H\rangle.$$

The following is the formal definition of MDM-PAD's MDM optimisation function. Let S be the result set of the utility function of MCDM model

$$S = \{\text{“hide”}, \text{“keep”}, \text{“highlight”}\}.$$

The mapping function is then defined as $f_{CU} : MD^{in} \to S$; by applying f_{CU} to each of the MDM from MD^{in}, we get

$$MD^{out\text{-}ut} = \{k \mid k \in MD^{in};\ f_{CU}(k) = \text{“keep”} \vee f_{CU}(k) = \text{“highlight”}\}.$$

Let F be a function with the property

$$|F(MD^{out\text{-}ut})| \leqslant |MD^{out\text{-}ut}|.$$

Let $MP = 2^{MD^{out\text{-}ut}}$ be the power set of the MDM set $MD^{out\text{-}ut}$; therefore, $M^{opt\text{-}out\text{-}ut} \subseteq MP$. We also define the following functions:

- $f_{DS} : K \to \{\text{‘}drop\text{’}, \text{‘}simplify\text{’}, \text{‘}keep\text{’}\}$;
- $\forall k \in K,\ f_{DS}: k \mapsto \{\text{‘}drop\text{’}, \text{‘}simplify\text{’}, \text{‘}keep\text{’}\}$;
- $f_{MS} : K \to \{\text{‘}drop\text{’}, \text{‘}simplify\text{’}, \text{‘}keep\text{’}\}$;
- $\forall k \in K,\ f_{MS}: k \mapsto \{\text{‘}drop\text{’}, \text{‘}simplify\text{’}, \text{‘}keep\text{’}\}$, where k is a cube from a set K;
- $G_1 : MP \to MP$

$$G_1(k) = \begin{cases} slice(d); & f_{DS}(k) = \text{‘}drop\text{’}, \\ dice(d); & f_{DS}(k) = \text{‘}simplify\text{’}, \\ k; & f_{DS}(k) = \text{‘}keep\text{’}, \end{cases}$$

  where d is a dimension in DS;
- $G_2 : MP \to MP$

$$G_2(k) = \begin{cases} hide(m); & f_{MS}(k) = \text{‘}drop\text{’}, \\ roll\_up(m); & f_{MS}(k) = \text{‘}simplify\text{’}, \\ k; & f_{MS}(k) = \text{‘}keep\text{’}, \end{cases}$$

where m is a measure in MS. Therefore, the previously defined function F can be rewritten as:

$$F : MP \to MP,$$
$$F(k) = (G_1 \circ G_2)(k).$$

We can see that the final optimised set of MDMs is

$$MD^{opt\text{-}out\text{-}ut} = \bigcup_{k \in MD^{out-ut}} (G_1 \circ G_2)(k) = \bigcup_{k \in MD^{out-ut}} F(k)$$

and, moreover,

$$\left|F\left(MD^{out\text{-}ut}\right)\right| \leqslant \left|MD^{out\text{-}ut}\right|.$$

The MDM optimisation method consists of the following four steps: (1) MDM selection, (2) dimension optimisation, (3) measurement optimisation, and (4) optimisation implementation. It is applied to each user type identified in the first part of MDM-PAD (user segmentation) separately. The first three steps are performed using DEX. Three multiple-criteria decision models are created for this purpose:

1. The McdMD – this DEX model makes it possible to select the MDMs of the input set MD^{in} that are relevant for a particular user type ut. The DEX decision tree structure of McdMD is represented in Fig. 5. Input to McdMD are six MDM characteristics: suitability of the MDM's content for user type ut (Ca_1), the MDM's detail level (Ca_2), its refresh time (Ca_3), the time frame of the data in the MDM (Ca_4), its number of dimensions (Ca_5) and number of measures (Ca_6). McdMD contains two aggregate attributes: importance

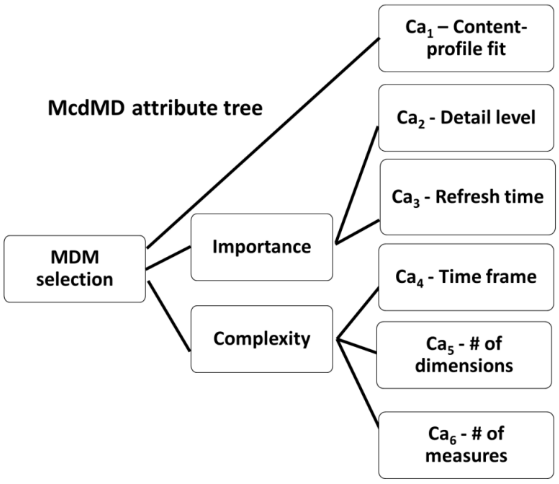

Fig. 5. DEX decision attribute tree for McdMD – MDM selection.

and complexity. MDM's importance for user type *ut* is assessed from the MDM's detail level (Ca_2), and refresh time (Ca_3). MDM's complexity for user type *ut* is assessed from the time frame of the captured data (Ca_4), its number of dimensions (Ca_5), and measures (Ca_6). McdMD's output (that is, MDM selection) depends on the suitability of the MDM's content (Ca_1), the MDM's importance, and the MDM's complexity for user type *ut*. While the DEX decision tree structure is domain-independent, the aggregation rules in the DEX model need to be specified for each domain separately. The utility function output of the McdMD model is "hide", "keep", or "highlight".

2. The McdD – this DEX model optimises the set of MDM dimensions $DS_i = \{d_{1,}d_2, \ldots, d_{Ndim}\}$ and their attribute set, where *Ndim* is the number of dimensions for a particular MDM from $MD^{out\text{-}ut}$ set. The DEX decision tree structure used in this step is presented in Fig. 6. Input to McdD are five dimension characteristics: the dimension size (Da_1), the dimension relation dependency (Da_2), its data type (Da_3), the characteristics of its hierarchy (Da_4), and its logical meaning (Da_5). McdD contains two aggregate attributes: dimension metrics and dimension kind. The suitability of the dimension's metrics for user type *ut* is assessed from the dimension size (Da_1), and the dimension relation dependency (Da_2). The suitability of the dimension's kind for user type *ut* is assessed from the dimension's data type (Da_3), its hierarchy characteristics (Da_4), and its logical meaning (Da_5). McdD's output (i.e. dimension adjustment) depends on the suitability of the dimension's metrics and kind for user type *ut*. Similarly to McdMD, McdD's DEX decision tree structure is domain-independent, while the aggregation rules in the DEX model need to be specified for each domain separately. The output of the DEX utility function for model McdD is defined as "drop", "simplify", or "keep" and is used to create the output dimension set.

The McdM – this DEX model optimises the set of measures $MS_i = \{m_1, m_2, m_3, \ldots, m_{Nmes}\}$ and their attribute set, where *Nmes* is the number of measures for a particular

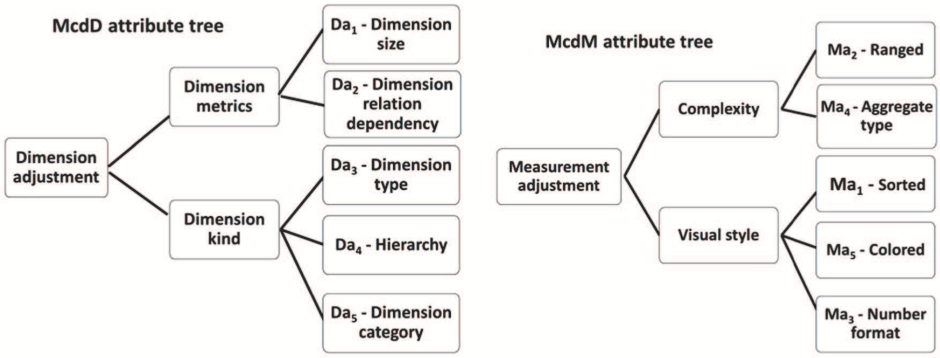

Fig. 6. DEX attribute tree for MCDM models McdD and McdM.

MDM in $MD^{out\text{-}ut}$. The DEX decision tree structure used in this step is presented in Fig. 6. Input to McdM are five measure characteristics: the sorting of the measure values (Ma_1), the clustering (ranging) of the measure (Ma_2), its formatting (Ma_3), its aggregate type (Ma_4), and its visual type (Ma_5). McdM contains two aggregate attributes: complexity and visual style. The suitability of the measure's complexity for user type *ut* is assessed from the clustering (ranging) of the measure (Ma_2) and its aggregate type (Ma_4). The suitability of the measure's visual style for user type *ut* is assessed from the sorting of the measure values (Ma_1), its visual type (Ma_5), and its formatting (Ma_3). McdM's output (that is, measure adjustment) depends on the suitability of the measure's complexity and visual style for user type *ut*. McdM's DEX decision tree structure is also domain-independent, while the aggregation rules in the DEX model need to be specified for each domain separately. The output of the DEX utility function for model McdM is defined as "drop", "simplify", or "keep" and is used to create the output measures set MS_i^{opt}.

All of the MDMs in MD^{in} are processed by McdMD. The model outputs "hide" for those MDMs that are not relevant to user type *ut*, "keep" for the MDMs that are relevant to user type *ut*, and "highlight" for the MDMs that are the most relevant to user type *ut*. The MDM selection (McdMD) is executed once for each user type and provides $MD^{out\text{-}ut}$, which is the set of all MDMs that are relevant to a specific user type *ut*. The characteristics of all the MDMs in the set $MD^{out\text{-}ut}$ are then processed by either the McdD and the McdM model, which output the operations that need to be performed on these characteristics in order to make them suitable for a particular user type *ut* and obtain $C^{out\text{-}ut}$ and $A^{out\text{-}ut}$.

The fourth step (optimisation implementation) concerns the application of the MDM operations that were found to be necessary by the outputs of the McdMD, McdD and McdM models. The optimal set of MDMs in the optimal format for user type *ut* ($MD^{opt\text{-}out\text{-}ut}$) is obtained, which contains a subset of the MDMs in MD^{in}. The input attributes and the decision-tree structure of the three MCDM models are the same for each user type identified in the first part of MDM-PAD (user segmentation). However, each MCDM model contains separate utility functions for each user type; that is, *N* utility functions are associated to each MCDM model.

MDM-PAD optimisation was performed using the following algorithm:

Data: MD^{in}, UT
Result: $MD^{opt\text{-}out}$

```
for each user type ut in set of user types UT do
    for each MDM k in MDM set MD^in do
        if f_CU(k) = 'keep' then
            copy(k) to MD^out-ut
        end
        if f_CU(k) = 'highlight' then
            copy(k) to MD^out-ut
        end
        if f_CU(k) = 'hide' then
            continue;
        end
    end
    for each MDM k in MDM set MD^out-ut do
        for each dimension d in dimension set DS_in do
            if f_DS(K) = 'drop' then
                slice(d);
            end
            if f_DS(k) = 'simplify' then
                dice(d);
            end
            if f_DS(k) = 'keep' then
                continue;
            end
        end
        for each measure m in MS_in do
            if f_MS(k) = 'drop' then
                hide(m);
            end
            if f_MS(k) = 'simplify' then
                roll_up(m);
            end
            if f_MS(k) = 'keep' then
                continue;
            end
        end
    end
end
```

The characteristics (C^{in}) in $MD^{out\text{-}ut}$ are processed by either McdD (when $d(C_i) =$ FALSE; that is, the characteristic is a dimension) or McdM (when $d(C_i) =$ TRUE; that is, the characteristic is a measure), which output "drop" for characteristics that are irrelevant to user type *ut*, "simplify" for relevant characteristics that need to be simplified (that is, the range of values needs to be simplified), and "keep" for relevant characteristics that are in the correct format for user type *ut*. The "drop" output of McdD and McdM is performed using the MDM operation slice. The MDM $MD_i^{out\text{-}ut}$, the characteristic to be dropped C_i, and the predefined aggregation operator G of characteristic C_i are the inputs to the slice operator, which outputs a reduced MDM. The "simplify" output is performed using roll-up and dice. We use two approaches to simplify a characteristic in the MDM. The first approach presents the user a predefined number K of attribute values of the characteristic (say, 10) to be simplified. The attribute values are selected so that they provide a uniform distribution of data instances per attribute value or a uniform distribution of the measure values per attribute value. They also define the level of abstraction l_r of characteristic C_i that is most suitable for user type *ut*. This operation is performed using the roll-up operator. The MDM $MD_i^{out\text{-}ut}$, the characteristic C_i (which has an abstraction level that needs to be increased), the abstraction level l_r, and the predefined aggregation operator G of characteristic C_i are used as the input to the roll-up operator, which outputs a simplified MDM. The second approach presents the user with the best-ranked attributes of a characteristic according to a predefined ranking function. The attribute values of the characteristic to be simplified are sorted according to their rank values. The top M (e.g., 10) attribute values are kept, whereas the other attribute values are aggregated to a single group that is denoted as "other" and presented by a single measure value. This operation is performed using the dice operator. The MDM $MD_i^{out\text{-}ut}$, the characteristic to be simplified C_i, a predicate P that specifies the simplification heuristic, and the predefined aggregation operator G of characteristic C_i are used as the input to the dice operator, which outputs a simplified MDM.

The size of $MD^{opt\text{-}out\text{-}ut}$ ranges from 0 to the size of MD^{in}. When the number of MDMs in $MD^{opt\text{-}out\text{-}ut}$ is close to 0 (or equal to 0), the following two situations are possible: (a) none of the MDMs in the input set MD^{in} are relevant to the user type and/or (b) none of the MDMs present in the optimised set $MD^{out\text{-}ut}$ satisfies the criteria of the MCDM models for dimension and measure optimisation. If the number of MDMs in $MD^{out\text{-}ut}$ approaches the size of MD^{in}, the optimisation can be considered useful if an MDM operation, such as roll-up, slice or dice, has been applied to the MDMs in $MD^{out\text{-}ut}$ to make them more appropriate for a specific user type. Otherwise, the MCDM models need to be refined or an additional set of MDM attributes needs to be added.

4. Empirical Verification

MDM-PAD has been applied to real-world data in sales-oriented decision-making, which uses the "BIView" business intelligence tool to capture the interaction of employees of a medium-size enterprise (Result, 2011). In this specific application, users were given access to 48 cubes that contained between three and 34 dimensions and 7–10 measures. There were an average of 30 dimensions per cube, and an average of eight measures per cube. The number of dimension members varied from 17 to 123 756. The maximum time frame was five years and the user interaction with BIView was recorded for a period of approximately two years. The user interaction records are expressed in the form $R = (user, action, cube, time)$, where *user* represents the user that interacted with the BIView application, *action* represents the action that the user performed, *cube* represents the cube on which the action was performed, and *time* represents the time when the action was performed. This data was recorded for 16 users.

4.1. *User Segmentation*

To determine the user types in the sales-oriented decision-making dataset, the user profile was defined as

$$\begin{aligned} Pu = {} & (cbAllGroup, cbComp, cbGr, cbSample, cbEntry, tmCurrent, tmLast2Y, \\ & rtWeekly, rtMonthly, objItems, objCustomers, objSales), \end{aligned}$$

where:

- *cbAllGroup*, *cbComp*, *cbGr*, and *cbSample* represent the degree of user's interest in information at several levels of detail; that is, the company as a whole, its individual offices, its office groups, or user-selected set of offices, respectively, mapped to input attribute Ca_2, which is the *Detail level* from the DEX McdMD model used in the optimisation in step 1.
- *tmCurrent* and *tmLast2Y* represent the degree of interest in information from different time frames, mapped to the input attribute Ca_4, which is *Time frame* from the DEX McdMD model used in the optimisation in step 1.
- *rtWeekly* and *rtMonthly* represent the degree of interest in information that is aggregated over different time periods; that is, weekly and monthly aggregations, respectively, which are partially used to define input attributes Ca3 and Ca4 from the DEX McdMD model.
- *objItems*, *objCustomers*, and *objSales* represent the degree of interest in information belonging to different cube categories, mapped to the input attribute Ca1, which is the *Content-profile fit* from the DEX McdMD model used in the optimisation in step 1.

The degrees of a user's interest in the specific pieces of information encompassed by the user profile attributes were determined using: (a) click frequency, (b) dwell time and (c) both click frequency and dwell time.

Having the user profiles, we applied K-means clustering to determine the user types. Figure 7 visualises the obtained clusters – that is, the user types for each of the three cases (a, b and c) – using the Radviz technique (Hoffman *et al.*, 1997) in Orange (Demšar *et al.*, 2013). Each of the three large circles represents a user-profile space. The attributes in the user-profile vector *Pu* are represented as points on the circle boundary. Each object inside a circle corresponds to a user-profile instance. The position of a user-profile instance is

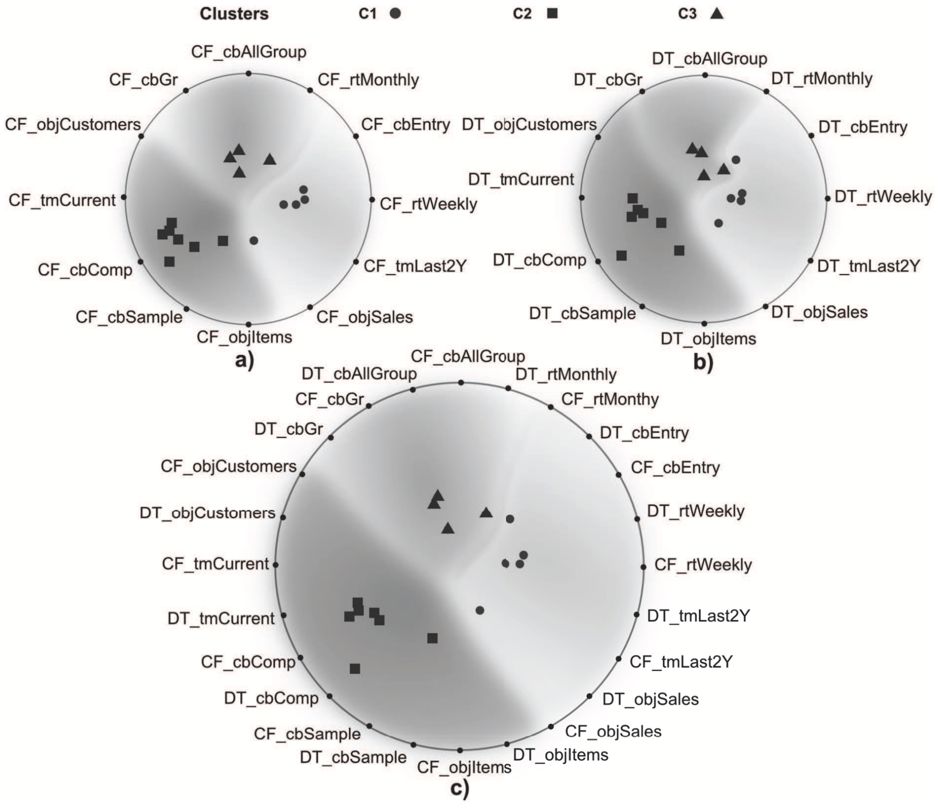

Fig. 7. User types when the degree of interest is calculated using (a) click frequency, (b) dwell time and (c) both click frequency and dwell time.

determined by a metaphor from physics – the spring equilibrium. Imagine an object that represents a user-profile instance and a set of $|Pu|$ springs, each of which connects an attribute point with the user-profile point. Imagine that the springs' stiffness corresponds to the attribute values in the user-profile instance. The user-profile object is located at the place where the spring forces are in equilibrium.

Figure 7 shows that, in all three cases (a, b and c), the user profile space can be divided into three disjoint subspaces. The analysis of the cluster centroids revealed the following types of users and their respective information of interest:

1. Operative seller (C2 – square in Fig. 7) – addresses business on an operational level with an owned set of business partners and a subset of the organisation articles. This user type is characterised by $ut1 = (0.16, 0.42, 0.00, 0.09, 0.00, 0.03, 0.00, 0.00, 0.00, 0.46, 0.08, 0.00)$.
2. Regional sales supervisor (C3 – triangle in Fig. 7) – supervises the sales of strategically important articles with the aim of determining potential deviations from business plans. This user type is characterised by $ut2 = (0.29, 0.18, 0.14, 0.06, 0.14, 0.02, 0.27, 0.01, 0.99, 0.17, 0.15, 0.00)$.

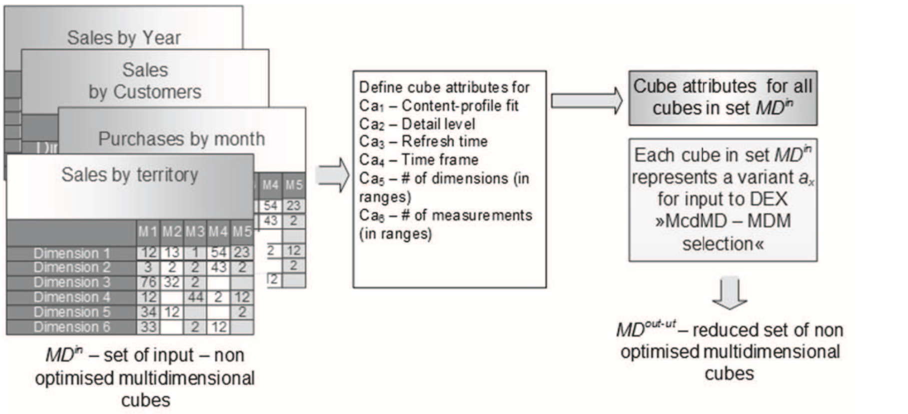

Fig. 8. Multidimensional model selection process – using DEX multi-attribute decision model McdMD.

3. Strategist (C1 – circle in Fig. 7) – regulates the products' life cycles; characterised by $ut3 = (0.70, 0.22, 0.00, 0.00, 0.00, 0.00, 0.25, 0.90, 0.10, 0.50, 0.03, 0.12)$.

4.2. *MDM Optimisation in Sales-Oriented Decision-Making Systems*

This section illustrates the process of MDM optimisation for regional sales supervisor user type. The McdMD, McdD, and McdM utility functions for this user type and a description of their respective input attribute values are presented in detail.

Step 1. MDM selection. This step determines $MD^{out\text{-}ut}$, which is the set of relevant MDMs in MD^{in} for the profile of a specific user type (regional sales supervisor in this case), based on the McdMD model and optimisation process shown in Fig. 8. Most of the attributes are obtained automatically from the definition of the MDM or MDM running attributes or statistics, while others are defined by domain experts.

The McdMD attribute qualitative value scales in this specific case are defined as follows.

Ca_1 – *Content-profile fit*: (*Fit*, *No fit*, or *Undefined*); obtained as an MDM attribute, defined manually or extracted automatically from the MDM name, definition, and the description. This obtained value is used for comparing to the list of business names assigned to each user type *ut*. The value *Fit* is assigned whenever the observed user type business area name, such as Sales, Purchases, and Finance, is found in the name, definition or description of the MDM. The value *No fit* is assigned whenever the MDM business area cannot be matched to any of existing user types. *Undefined* is assigned where no match or multiple match of business area names was found.

Ca_2 – *Detail Level*: (*Low, Mid*, and *High*); defines the level of detail of the data that are represented in the MDM, depending on the number of dimension members. The value *Low* is assigned when the number of dimension members is $<= 25$; value *High* is assigned whenever the number of dimension members is over 250; otherwise, value *Mid* is assigned.

Ca_3 – *Refresh time*: (*Yearly, Monthly*, *Weekly*, Daily, and *Hourly*); the time-frame that the data in the MDM was (or should be) refreshed.

Table 1
High-level decision rules for McdMD DEX model - adaptation of an MDM to a specific user profile.

	Complexity	Importance	Ca_1 – content-profile fit	MDM optimisation
1	**High**	>= mid	**Fit**	**Highlight**
2	*	**high**	**Fit**	**Highlight**
3	Mid	**low**	**Fit**	**Highlight**
4	**Low**	**low**	>= undefined	**Highlight**
5	**Low**	>= mid	**Fit**	**Highlight**
6	Mid	mid	**Fit**	Keep
7	>= Mid	**high**	>= undefined	Keep
8	**Low**	**low**	**Fit**	Keep
9	**Low**	>= mid	undefined	Keep
10	**High**	**low**	*	**Hide**
11	**High**	*	>= undefined	**Hide**
12	<= Mid	<= mid	>= undefined	**Hide**
13	*	mid	**No fit**	**Hide**

Ca_4 – *Time frame*: (*Multi-Year*, *2-Year*, *1-Year*, *Month*, *Week*, and *Day*); the range of the largest time dimension in the MDM is grouped in ranges.

Ca_5 – *# of dimensions*: (*1–5*, *5–10*, *10–20*, and > *20*); the number of dimensions in an MDM defined in ranges.

Ca_6 – *# of measures*: (*1, 2, 3–5*, and > *5*); the number of measures (and aggregates) in a specific MDM (in ranges).

Table 1 represents high-level decision rules in the McdMD model (Fig. 5) for the regional sales supervisor user type. These rules are used to compute the root attribute MDM utility value. Each row in the table represents one rule, each of which is indexed with the number presented in the first column. Each rule contains three conditions, one for each MDM optimisation descendent in the McdMD attribute tree. The second and third columns represent rule conditions concerning aggregate attributes Complexity and Importance. The fourth column represents the conditions concerning the attribute *Content-profile fit* (Ca1). The last column on the table represents rules' consequents; that is, the value assigned to the utility value of the McdMD model.

The symbol '*' in rule tables denotes 'any value', '>=' represents 'better or equal', '<=' indicates 'worse or equal', and 'value1:value2' denotes the interval between and including the two values. Rules that correspond to a single aggregate attribute are grouped together. To improve readability, in all decision tables, the highest and the lowest values of each attribute are emphasised using bold italic and bold type-face, respectively. For example, rule in row 1 in Table 1 states that: if Complexity is 'High' and Importance is better than or equal to 'mid' and *Content-profile fit* (Ca1) is 'Fit' then MDM optimisation utility output value is 'Highlight'.

This model contains two aggregate attributes – *Complexity* and *Importance* – which represent the intermediate results of the assessment depending on previously defined basic attributes. For each aggregate attribute, a set of if-then-else rules was composed to define corresponding utility function depending on the values of its descendant attributes.

Table 2
Decision rules corresponding to the *Complexity* attribute.

	Ca$_6$ – # measures	Ca$_5$ – # of dimensions	Ca$_4$ – time frame	Complexity
1	*	>= 10-=20	**Day**	**High**
2	>= 2	*	>= **Week**	**High**
3	>= 3–5	*	>= Month	**High**
4	>= 3–5	>= 5–10	>= 1-Year	**High**
5	> **5**	*	>= 1-Year	**High**
6	> **5**	> **20**	*	**High**
7	**1**	<= 5–10	>= Month	Mid
8	**1**	*	Month: **Week**	Mid
9	<= 2	*	Month	Mid
10	**1**	5–10	>= 1-Year	Mid
11	**1**	>= 5–10	1-Year: **Week**	Mid
12	<= 2	5–10	1-Year: Month	Mid
13	**1**	>= 10–20	**2-Years: Week**	Mid
14	<= 2	> **20**	**2-Years:** Month	Mid
15	<= 3–5	> **20**	**2-Years**	Mid
16	2	5–10	**2-Years:** Month	Mid
17	>= 2	5–10	**2-Years**	Mid
18	3–5	**1–5**	**2-Years:** 1-Year	Mid
19	3–5	*	**2-Years**	Mid
20	>= 3–5	<= 10–20	**2-Years**	Mid
21	3–5	>= 5–10	<= **2-Years**	Mid
22	>= 3–5	5–10: 10–20	<= **2-Years**	Mid
23	> **5**	<= 10–20	<= **2-Years**	Mid
24	**1**	<= 5–10	<= **2-Years**	**Low**
25	<= 2	**1–5**	<= 1-Year	**Low**
26	<= 2	*	**Multi-Year**	**Low**
27	<= 3–5	**1–5**	**Multi-Year**	**Low**
28	2	10–20	<= 1-Year	**Low**

Table 3
Decision rules corresponding to the *Complexity* attribute.

	Ca$_2$ – detail level	Ca$_3$ – refresh time	Importance
1	**High**	>= Monthly	**Low**
2	<= Mid	>= Daily	**Low**
3	*	**Hourly**	**Low**
4	**High**	**Yearly**	Mid
5	Mid	Monthly: Weekly	Mid
6	**Low**	Daily	Mid
7	>= Mid	**Yearly**	**High**
8	**Low**	<= Weekly	**High**

Table 2 shows the corresponding decision rules for the *Complexity* and Table 3 shows the corresponding rules table for *Importance*.

Table 4 presents sample alternatives (MDM cubes from MD^{in}) with values for each input attribute. All of the 48 alternatives were described in this manner. The final results

Table 4
Sample criteria for cubes Cu_1 and Cu_{10} through Cu_{12}.

MDM – cube name	Ca_1	Ca_2	Ca_3	Ca_4	Ca_5	Ca_6	Result
Cu_1 – Yearly sales	Fit	Mid	Daily	Multi-year	> 20	> 5	Highlight
Cu_{10} – Monthly sales	Fit	Low	Monthly	1-Year	> 20	> 5	Highlight
Cu_{11} – Weekly sales	Fit	High	Weekly	1-Year	> 20	> 5	Keep
Cu_{12} – Purchases by customers	No Fit	Mid	Monthly	Multi-year	10–20	> 5	Hide

Table 5
Top-level aggregation rules for dimension optimisation model.

	Dimension metrics	Dimension kind	Dimension adjustment
1	Useless	$\leqslant$ Fair	Drop
2	$\leqslant$ Tie	Low	Drop
3	Useless	High	Simplify
4	Tie	Fair	Simplify
5	Usable	Low	Simplify
6	$\geqslant$ Tie	High	Keep
7	Usable	$\geqslant$ Fair	Keep

show that 16 cubes should be hidden, nine cubes should be left intact, and 23 should be highlighted for operative seller user type ut_1; that is, 48% of the cubes are of interest to this specific user type.

Step 2. Dimension optimisation. This step determines which dimensions of the MDMs in $MD^{out\text{-}ut}$ are relevant to a specific user type DS_i^{opt} based on McdD model. The McdD model attribute tree is presented in Fig. 6, and the top-level aggregation rules for this model are presented in Table 5. The decision rules for the *Dimension metrics* and *Dimension kind* aggregate functions are defined using decision rules similar to those described for model McdMD. Twenty-five rules were defined for *Dimension metrics*, providing *Useless, Tie*, and *Usable* intermediate results, and 100 rules were defined for *Dimension kind*, giving intermediate results of *Low, Fair, and High*. In the process of defining basic decision rules, DEX supports the indirect definition of weights in the process of defining non-entered function values.

The McdD attribute values in this particular case are as follows:

Da_1 – *Dimension size*: (*1–10, 10–100, 100–1000*, and $>$ *1000*); number of elements in the dimension in ranges.

Da_2 – *Dimension relation dependency*: (*None, One, Two, More-than-Two*, and *Unknown*); inherits the dependency from the original underlining table(s).

Da_3 – *Dimension type*: (*Undefined, Text, Numeric-and-text, Numeric, Date*, and *Date-time*); this attribute represents a logical dimension value type that can be partially mapped to a general dimension source data type.

Da_4 – *Hierarchy*: (*None, Simple, Two-level*, and *More-than-two-level*).

Da_5 – *Dimension category*: (*Real, Location*, and *Data*); describes the logical meaning of the dimension value.

Table 6
Sample criteria for dimensions.

Dimension	Da_1	Da_2	Da_3	Da_4	Da_5	Result
D1 – Customer name	> 1000	One	Numeric-and-text	Simple	Real	Keep
D2 – Country	1–10	2	Text	Two-level	Location	Drop
D3 – Item description	< 1000	None	Numeric-and-text	None	Real	Keep
D4 – Posting date	10–100	None	Date-time	More-than two-level	Real	Drop
D5 – Invoice reference	> 1000	> 2	Numeric	Simple	Data	Simplify

Table 7
Decision rules for measure analysis.

	Complexity	Visual style	Measure adjustment
1	$\geqslant$ Mid	Overloaded	Drop
2	High	$\geqslant$ Generic	Drop
3	Low	Overloaded	Simplify
4	Mid	$\leqslant$ Generic	Simplify
5	$>$ Mid	Clean	Simplify
6	Low	$\leqslant$ Generic	Keep

All of the dimension attribute values were determined automatically, except for "Da_2 – *Dimension relation dependency*", which must be set by the BI expert. Table 6 presents the values of the input attributes for a sample of alternatives (dimensions) of one particular cube (CU_{10}), which was selected in the first optimisation step.

For operative sellers, the dimensions "D1 – Customer Name" and "D3 – Item Description" should be kept, "D2 – Country" and "D4 – Posting Date" should be dropped, and "D5 – Invoice Reference" should be used only if it can be simplified (for example, by grouping or ranking the elements of the dimension). The cube operations that can be used to simplify a dimension are roll-up and slice.

Step 3. Measure optimisation. This step determines which measures of the MDMs in $MD^{out\text{-}ut}$ are relevant to a specific user type MS_i^{opt} based on the McdM model. The McdM model attribute tree is presented in Fig. 6, and the top-level aggregation rules for this model are presented in Table 7.

The McdM attribute values in this case are as follows:

Ma_1 – *Sorted*: (*Ascending, Descending*, and *None*); describes the display order.

Ma_2 – *Ranged*: (*None, Equal-Density, Equal-Range-Size*, and *Custom*); defines the clustering(ranging) of the measure. *Equal-Range* and *Equal-Density* assume that the measures were transformed into ranges by internal algorithms, whereas *Custom* denotes non-normal distributions of the values into ranges.

Ma_3 – *Number format*: (*Simple, Detailed*, and *Percentage*); *Simple* indicates integer numbers with no decimals; *Detailed* indicates that values are shown in a full format with rich formatting; and *Percentage* indicates that the values shown are percentages.

Ma_4 – *Aggregate Type*: (*Computed, Sum, Index*, and *Ratio*). *Sum* indicates generic measure behaviour with no internal aggregation. *Computed* indicates that the displayed measure value is computed from any combination of measures in the MDM. *Index*

Table 8
Sample criteria for measures.

	Ma_1	Ma_2	Ma_3	Ma_4	Ma_5	Result
M1 – Quantity of item	None	None	Simple	Sum	None	Keep
M2 – Net value	Ascending	Equal-density	Detailed	Computed	None	Keep
M3 – Gross value	None	None	Detailed	Sum	Max	Keep
M4 – Discount value	None	None	Percentage	Ratio	Range	Simplify
M5 – Super rebate value	None	None	Percentage	Ratio	Range	Simplify
M6 – Extended rebate value	Ascending	None	Percentage	Ratio	Range	Simplify
M7 – Bonus rebate value	Ascending	None	Percentage	Ratio	Range	Simplify

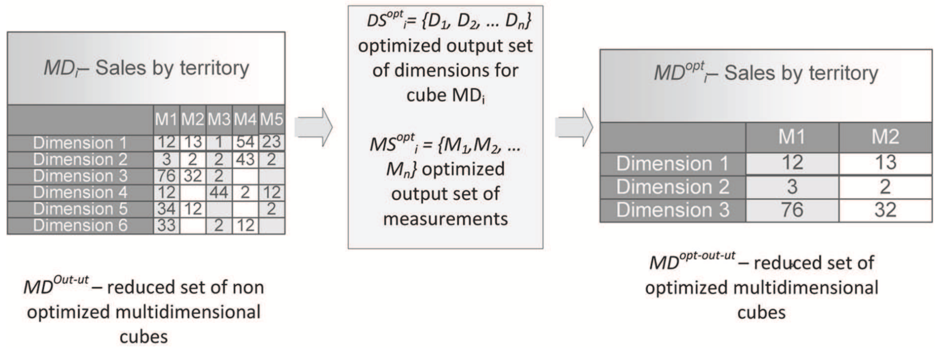

Fig. 9. Creation of derived reduced MDM with optimised set of dimensions and measures.

indicates that the value is a variant of the computed value and that there is a relationship between two measures of the same dimension (for example, Month Index and Year Index).

Ma_5 – *Coloured*: (*None, Min, Max*, and *Range*). The visual style of the measures in the user interface; the colour of the values depends on the values.

Aggregate rule Complexity contains 16 discrete if-then-else decision rules; while aggregate function Visual style contains a total of 36 decision rules.

Table 8 presents the values of the input attributes for a sample of alternatives (measures) of one particular cube (CU_{10}), which was selected in the first optimisation step.

For operative sellers, the "M1 – Quantity of Item", "M2 – Net Value", and "M3 – Gross value" measures of cube CU_1 should be kept, whereas the measures "M4 – Discount Value", "M5 – Super Rebate Value", "M6 – Extended Rebate value", and "M7 – Bonus Rebate value" must be simplified. For simplification, the drill-down and slice cube operations can be used.

Step 4. Optimisation implementation. The final step (Fig. 9) of the optimisation process is the implementation of the optimisation. The optimised set of dimensions and measures is obtained using the output of the McdMD, McdD, and McdM.

4.3. *Case Study*

To evaluate the benefit of MDM-PAD, we compared the ease-of-use and user satisfaction of three user interfaces (UI):

1. Typical UI – typical BI user interface that covers all of the relevant business information and contains a large number of views with a smaller number of dimensions and measures.
2. MDM-PAD UI – BI user interface optimised using the proposed MDM-PAD method. It presents only a subset of the MDMs with customised dimensions and measures in accordance with the needs of regional sales supervisors.
3. Topsis UI – BI user interface optimised using the multiple-criteria decision analysis method Topsis (Hung *et al.*, 2011; Peng *et al.*, 2011), which has been successfully applied in studies similar to ours. We used Topsis for MDM optimisation (part II in MDM-PAD), while keeping the user segmentation part (part I in MDM-PAD) the same. Because Topsis does not allow hierarchical decision models and qualitative criteria, we adjusted the decision models used for MDM optimisation with the help of the Topsis Solver. In particular, we flattened the decision models and performed the procedures for criteria-weight estimation as proposed in Topsis.

The ease-of-use and user satisfaction of these UIs was estimated in two ways:

1. Quantitative: Users were given problems to solve with the typical, the MDM-PAD, and the Topsis UI. Each user chose the UI and the problem order. The ease-of-use of each UI was estimated by the percentage of correctly answered problems, the average time, and the average number of clicks that users needed to obtain an answer.
2. Qualitative: Each user completed a questionnaire through which we evaluated the user satisfaction of the typical, the MDM-PAD and the Topsis UI.

The problems were defined in a form of tasks and questions, some examples of which are provided below:

1. Find a list of salespersons needed to be invited to the next presentation of products in the product group "LL Touring cycles" – up to five persons.
2. Compile a list of customers with the sales contract due day less than or equal 15 within the active contracts in year 2011. Use only customers with annual gross amount over 100K EUR and average past due days greater than 15 days.

The tests were performed by 25 users: 16 were beginners (group A), seven had occasionally used the BI applications previously (group B), and two were BI experts (group C).

The quantitative analysis revealed the following:

- The difference in the expertise of users of different user types was apparent. Group A (beginners) had the lowest percentage of correctly answered problems. Group B (occasional BI users) had a higher percentage of correctly answered problems than group A, while group C had the highest percentage of correctly answered problems.

Table 9
Percentage of correct answers, and time and number of clicks used to obtain an answer using the typical UI, the MDM-PAD UI and the Topsis UI. The cells represent the average value in each case.

	User interface	% correct	Average time	# of clicks
Group A	Typical UI	15	6.8	44
	MDM-PAD UI	44	3.6	25
	Topsis UI	19	3.6	29
Group B	Typical UI	38	5.9	40
	MDM-PAD UI	62	4.3	22
	Topsis UI	10	5.6	30
Group C	Typical UI	83	3.3	11
	MDM-PAD UI	100	2.9	11
	Topsis UI	33	7.2	33
All users	Typical UI	27	6	38
	MDM-PAD UI	53	3.8	23
	Topsis UI	17	4.5	30

- The MDM-PAD UI had the highest number of correct answers. Overall, 53% of the problems were correctly answered with the MDM-PAD UI, compared to 27% for the typical UI and 17% for the Topsis UI. MDM-PAD UI outperformed the typical UI and the Topsis UI on four out of 10 problems, while Topsis UI had the best performance on one problem. A best-performing UI could not be identified on five of the problems.
- The MDM-PAD UI required the lowest time and number of clicks to obtain an answer (correct or partially correct). On average, 3.8 minutes and 23 clicks were required to obtain an answer with the MDM-PAD UI, compared to 6 minutes and 38 clicks for the typical UI, and 4.5 minutes and 30 clicks for the Topsis UI.

The qualitative analysis revealed the following results:

- MDM-PAD UI contains the most suitable subset of MDMs: 63% of the users found the MDM subset in MDM-PAD UI suitable, compared to 25% and 46% for the typical UI and the Topsis UI, respectively.
- The clarity of the dimension and measure layout of the MDM-PAD UI is the highest: 63% of the users found the dimension and measure layout of the MDM-PAD UI clear, compared to 33% and 35% for the typical UI and the Topsis UI, respectively.
- The MDM-PAD UI is the simplest to use: 50% of the users found the MDM-PAD UI simple to use, compared to 25% and 23% for the typical UI and the Topsis UI, respectively.

In summary, MDM-PAD contributed to the highest number of correct answers. According to user feedback, it was also the most convenient UI. Although this result was also found for expert BI users, the best results were obtained for regular and occasional BI users. However, the results also showed that by using MDM-PAD, not all of the MDMs were properly adapted to the particular user type, which resulted in a high percentage of unanswered or partially answered questions for one of the 10 problems.

5. Conclusions

The main contribution of this paper is the development of a novel methodology called Multidimensional Data Model Profile Adaptation (MDM-PAD) for the adaptation of the MDMs of BI systems to the specific needs of its users. Using near-automatic user profile attribute definitions combined with the advantages of DEX MCDM qualitative attribute possibilities, makes it possible to overcome many obstacles for the faster implementation and modification of BI systems. To meet the expectations of the growing number of information consumers, BI systems must deliver functionality and content that is customised to the needs of each specific system user. In existing systems, this need is achieved by segmenting users based on their analytical habits and requirements and delivering pre-defined customised functionality and content for each user group.

MDM-PAD performs the adaptation of MDMs in two steps. First, the types of system users are determined based on their respective needs and preferences. This step is performed using clustering methods to analyse historical data of user interactions with the system. Second, for each user type, MCDM models are defined by determining the best-fit visualisation MDM for each specific user type. Using a number of small MCDM models instead of one large MCDM model, our method achieves a large degree of usability and flexibility after being implemented in real BI systems. This method does not limit the number of MCDM models used or the degree of user segmentation. Different degrees of user segmentation can be used for different business or non-business areas, and a higher number of user types results in the definition of simpler MCDM models containing fewer, but more accurate, decision rules. The results of the MCDM models are directly used for the adaptation of the MDMs without any transformation, which enables the end-user OLAP BI tools to be automatically customised. This method assumes that users have been properly clustered based on the activity records of the BI system. Proper clustering requires a larger number of activity records, but the fact that the BI system is not a transactional system means that the activity record collection process can be timely. To overcome this problem, the user activity should be monitored at the level of whole desktop (or web) systems, which would provide additional input data for clustering.

The evaluation showed that users find the MDM-PAD-customised MDMs more convenient than the typical MDMs that are provided by default in the current BI systems, and more convenient than the MDMs customised using the Topsis approach. In the tests, the customisation of the MDMs to the user type with the MDM-PAD approach contributed to the highest percentage of correctly solved problems and the lowest average answering time and the lowest average number of clicks required to answer each question. Because the tests were performed on a limited number of test subjects, future tests should verify these observed improvements. Moreover, additional tests need to be performed on additional domains. Despite these limitations, the chosen problems represented typical BI tasks, and the improvements were consistent, which indicates that the developed method is indeed beneficial.

As future work, we are considering three main method enhancements. First, through self-rule adoption, end users can eventually change their role in the business environment.

Because this type of event can occur frequently, the user must be able to select his/her business role dynamically, and the MDM adaptation should be performed "on-the-fly" depending on the current user type. This functionality can be applied after implementing the fully automated discovery of all of the MDM/dimension/measure attributes needed as input for the MCDM. Second, the addition of more measurement points in the BI tool will provide more detailed description of the user behaviour, which will increase the accuracy of the self-segmentation. Third, this methodology can be extended to optimise graphs (not only MDMs). This extension will require the addition of many new measurements in the audit process of the BI tool.

Acknowledgements. This research was partly financed by the European Social Fund of the European Union.

References

Barfod, M.B., Salling, B.K., Lelur, S. (2010). Composite decision support by combining cost-benefit and multi-criteria decision analysis. *Decision Support Systems*, **51**(1), 167–175.

Basilico, J., Hofmann, T. (2004). Unifying collaborative and content-based filtering. In: *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D. (2005). A personalization framework for OLAP queries. In: *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP, DOLAP '05*, Bremen, Germany, pp. 9–18.

Bernatavičienė, J., Dzemyda, G., Marcinkevičius, V. (2007). Conditions for optimal efficiency of relative MDS. *Informatica*, 18(2), 187–202.

Bohanec, M. (2003). Decision support. In: *Data Mining and Decision Support: Integration and Collaboration*. Kluwer Academic, Dordrecht, pp. 23–35.

Bohanec, M. (2008). Program for multi-attribute decision Making, user's manual. http://kt.ijs.si/MarkoBohanec/pub/DEXiManual30r.pdf.

Bohanec, M., Rajkovič, V. (1990). DEX: an expert system shell for decision support. *Sistemica* 1, 145–157.

Bohanec, M., Rajkovič, V. (1999). Multi-attribute decision modelling: industrial applications of DEX. *Informatica (Ljublj.)*, 23(4), 487–491.

Bouyssou, D., Marchant, T., Pirlot, M., Tsoukias, A., Vincke, P. (2006). *Evaluation and Decision Models with Multiple Criteria: Stepping Stones for the Analyst*. Springer, New York.

Chen, Y. (2006). Multiple criteria decision analysis: classification problems and solutions. *PhD thesis*, University of Waterloo.

Cundrič, A., Kern, T., Rajkovič, V. (2008). A qualitative model for road investment appraisal. *Transport Policy*, 15(4), 225–231.

DecisionPath (2012). http://www.decisionpath.com/wp-content/uploads/2011/01/How-is-BI-being-used.pdf.

Delgadom, M., Molina, C., Sanchez, D., Vila, A., Rodrigez-Ariza, L. (2004). A fuzzy multidimensional model for supporting imprecision in OLAP. In: *Proceedings, 2004 IEEE International Conference*.

Demšar, J., Curk, T., Erjavec, A. (2013). Orange: data mining toolbox in python. *Journal of Machine Learning Research*, 14(1), 2349–2353.

Deshpande, M., Karypis, G. (2004). Selective Markov models for predicting web page accesses. *ACM Transactions on Internet Technology*, 4(2), 163–184.

Dori, D., Feldman, R., Sturm, A. (2008). From conceptual models to schemata: an object-process-based data warehouse construction method. *Information Systems*, 33(6), 567–593.

Duan, L., Street, W.N., Xu, E. (2011). Healthcare information systems: mining methods in the creation of a clinical recommender system. *Enterprise Information Systems* 5 (2), 169–181.

Dzemyda, G., Sakalauskas L. (2011). Large-scale data analysis using heuristic methods. *Informatica*, 22(1), 1–10.

El Sawy, O.A. (2003). The IS core – IX the 3 faces of IS identity: connection, immersion and fusion. *Communications of AIS*, 12(39), 588–598.

Espil, M.M., Vaisman, A.A. (2002). Revising aggregation hierarchies in OLAP: a rule-based approach. *Data & Knowledge Engineering*, 45(2), 225–256.

Finucae, B., Bange, C., Mack, M., Vierkorn, S., Keller, P. (2010). *The BI Survey 9 – The Customer Verdict*. Business Application Research Center BARC.

Gartner (2008). *Business Intelligence Summit by Gartner*. Gartner, Chicago.

Gartner (2012). *Magic Quadrant for Business Intelligence Platforms*.

Giacometti, A., Marcel, P., Negre, E. (2008). *A framework for recommending OLAP queries*. In: *Proc. DOLAP*, Napa Valley, CA, USA, pp. 73–80.

Golfarelli, M., Rizzi, S., Biondi, P. (2011). MYOLAP: an approach to express and evaluate OLAP preferences. *IEEE Transactions on Knowledge and Data Engineering*, 23(7), 1050–1064.

Hoffman, P. et al.. (1997). DNA visual and analytic data mining. In: *Proceedings of the IEEE Visualization*, pp. 437–441.

Hung, Y.H., Chou, S.C., Tzeng, G.H. (2011). Knowledge management adoption and assessment for SMEs by a novel MCDM approach. *Decision Support Systems*, 51(2), 270–291.

Jerbi, H., Ravat, F., Teste, O., Zurfluh, G. (2009). *Preference-based recommendations for OLAP analysis*. In: *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery, DaWaK'09*, Linz, Austria, 31 August–September, 2009, pp. 467–478.

Johnson, J. (2002). *Jim Standish Group Study Reported at XP2002 by Jim Johnson*. Jim Standish Group, Alghero, Sardinia.

Karbauskaite, R., Dzemyda, G. (2009). Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatica*, 20(2), 235–254.

Kontič, B., Bohanec, M., Urbančič, T. (2006). An experiment in participative environmental decision making. *The Environmentalist*, 26, 5–15.

Korelič, I. and F. Škedelj (2008). *Obstacles for OLAP tools implementation in small enterprises*. Portorož, FOV.

Lujan, M.S., Trujillo, J., Song, I.Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, 59(3), 725–769.

Malinowski, E., Zimanyi, E. (2006). Hierarchies in a multidimensional model: from conceptual modeling to logical representation. *Data & Knowledge Engineering*, 59(2), 348–377.

Mansmann, S., Scholl, M.H. (2007). Empowering the OLAP technology to support complex dimension hierarchies. *International Journal of Data Warehousing and Mining*, 3(4), 31–50.

Moole, B.R. (2003). *A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems*. Wonder Technol. Corp., Herndon, VA, USA, IEEE.

Moss, L.T., Shaku, A. (2003). *Business Intelligence Roadmap*. Addison-Wesley, Reading.

Pardillo, J., Mazóna, J., Trujilloa, J. (2010). Extending OCL for OLAP querying on conceptual multidimensional models of data warehouses. *Information Sciences*, 180(5), 584–601.

Pavlovič, M., Čerenak, A., Pavlovič, V., Rozman, Č., Pažek, K., Bohanec, M. (2011). Development of DEX-HOP multi-attribute decision model for preliminary hop hybrids assessment. *Computers and Electronics in Agriculture*, 75(1), 181–189.

Pendse, N. (2006). *Olap Survey*. Optima Publishing Ltd., London.

Peng, Y., Zhang, Y., Tang, Y., Li, S. (2011). An incident information management framework based on data integration, data mining, and multi-criteria decision making. *Decision Support Systems*, 51(2), 316–327.

Prat, N., Akoka, J., Comyn, W.I. (2006). A UML-based data warehouse design method. *Decision Support Systems*, 42(3), 1449–1473.

Redondo, J.L., Ortigosa, P.M., Žilinskas, J. (2012). Multimodal evolutionary algorithm for multidimensional scaling with city-block distances. *Informatica*, 23(4), 601–620.

Result (2011). *Biview Business Intelligence Tool*. http://www.biview.com.

Roberts, R., Goodwin, P. (2002). Weight approximations in multi-attribute decision models. *Journal of Multi-Criteria Decision Analysis*, 11(6), 291–303.

Romero, O., Abelló, A. (2010). A framework for multidimensional design of data warehouses from ontologies. *Data & Knowledge Engineering*, 69(11), 1138–1157.

Rozman, C., Pažek, K. (2005). Application of computer supported multi-criteria decision models in agriculture. *Agriculturae Conspectus Scientificus*, 70(4), 127–134.

Rozman, C., Pazek, K., Bavec, M., Bavec, F., Turk, J., Majkovic, D. (2006). The multi-criteria analysis of spelt food processing alternatives on small organic farms. *Journal of Sustainable Agriculture*, 28(2), 159–179.

Sarawagi, S. (1999). Explaining differences in multidimensional aggregates. In: *Proc. VLDB*, Edinburg, Scotland, pp. 42–53.

Sarawagi, S. (2000). User-adaptive exploration of multidimensional data. In: *Proc. VLDB*, Cairo, Egypt, pp. 307–316.

Stanujkic, D., Magdalinovic, N., Jovanovic, R. (2013). A multi-attribute decision making model based on distance from decision Maker's preferences. *Informatica*, 24(1), 103–118.

Thomas, H., Datta, A. (2001). A conceptual model and algebra for on-line analytical processing in decision support databases. *Information Systems Research*, 12(1), 83–102.

Turskis, Z., Zavadskas, E.K. (2010). A novel method for multiple criteria analysis: grey additive ratio assessment (ARAS-G) method. *Informatica*, 21(4), 597–610.

Wang, J., Lin, J. (2002). *Are Personalization Systems Really Personal? Effects of Conformity in Reducing Information Overload*. IEEE Computer Society, Hawaii.

Witten, I.H., Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed., Morgan Kaufmann.

Zavadskas, E.K., Kaklauskas, A., Turskis, Z., Tamošaitiene, J. (2009). Multi-attribute decision-making model by applying grey numbers. *Informatica*, 20(2), 305–320.

Žilinskas, A., Podlipskyte, A. (2003). On multimodality of the SSTRESS criterion for metric multidimensional scaling. *Informatica*, 14(1), 121–130.

Žilinskas, J. (2008). On dimensionality of embedding space in multidimensional scaling. *Informatica*, 19(3), 447–460.

Žnidaršič, M., Bohanec, M. (2007). Automatic revision of qualitative multi-attribute decision models. *Foundations of Computing and Decision Sciences*, 32(4), 315–326.

Žnidaršič, M., Jakulin, A., Džeroski, S., Kampichler, C. (2006). Automatic construction of concept hierarchies: the case of foliage-dwelling spiders. *Ecological Modelling*, 191(1), 144–158.

I. Korelič is the head of research and development department from industry at Result d.o.o. and PhD candidate with the University of Maribor, Faculty of Organisational Sciences. His research focuses on business intelligence systems with the emphasis on smart user interfaces and mobile solutions.

V. Mirchevska is a researcher at the Department of Intelligent Systems at Jožef Stefan Institute, Slovenia. She holds a ScD degree in computer and information science from the Jožef Stefan International Postgraduate School. Her research focuses on modelling agent behaviour based on observing low-level action sequences by leveraging both domain knowledge and machine learning. The main application areas of her research are user profiling, remote health monitoring and security

V. Rajkovič is a professor emeritus of information systems at University of Maribor, Faculty of Organisational Sciences, and a senior researcher at the Intelligent Systems Department at Jožef Stefan Institute. His research interests focus on information systems and artificial intelligence methods for supporting decision processes and education.

M. Kljajić Borštnar holds a PhD in the area of Information Systems Management from the University of Maribor. Her research work covers expert systems, multi-criteria decision-making, and Information Systems development methods. She holds a position of an Assistant Professor and is a member of Laboratory for Decision Processes and Knowledge-Based Systems. Her recent research work is focused on experiments with decision groups applying system dynamics simulators in experimental, interactive learning, and living laboratory environments. She is involved in several EU and industry projects. She is the author and co-author of several scientific articles published in recognised international journals including Group Decision and Negotiation and System Dynamics Review. She is also a program committee chair of annual international Bled eConference.

M. Gams is a professor and the head of the Department of Intelligent Systems at Jožef Stefan Institute, Slovenia. He is member of executive Boards of the Engineering academy of Slovenia, and a cofounder of the Engineering Academy, Artificial Intelligence Society and Cognitive Sciences Society in Slovenia. He is an executive contact editor of the journal Informatica (Ljubljana) and member of the editorial board of several international journals. His research interests include intelligent systems and agents, machine learning, hybrid learning and reasoning, cognitive sciences, and ambient intelligence

Daugiakriteris būdas daugiamačiams duomenų modeliams optimizuoti

Igor KORELIČ, Violeta MIRCHEVSKA, Vladislav RAJKOVIČ,
Mirjana KLJAJIĆ BORŠTNAR, Matjaž GAMS

Šiame straipsnyje siūlomas naujas būdas daugiamačių duomenų modelių adaptavimui naudotojų poreikiams. Daugiamačių duomenų modeliai, naudojami šiuolaikinėse verslo analizės sistemose, iš prigimties yra sudėtingi. Siekiant sumažinti šių modelių sudėtingumą, straipsnyje pasiūlyta naudoti kokybišką daugiakriterių sprendimų modeliavimo metodą, pagrįstą hierarchinio medžio kriterijų naudojimu, išskaidant didesnį uždavinį į mažesnius. Galutinė reikšmė, gaunama agreguojant kriterijų reikšmes, naudojant paprastas „jei...tai“ taisykles, kurios hierarchiniame kriterijų medyje sudaro žiniomis grįstas ekspertines taisykles, atspindinčias naudotojų pageidavimus. Modelis yra patikrintas, naudojant vidutinio dydžio įmonės pardavimų duomenis. Ir kokybinis (iš apklausų), ir kiekybinis (naudojant duomenų tyrybą) pasiūlytos metodologijos vertinimas parodė, kad pasiūlytas būdas palengvina naudojimąsi verslo analizės sistemomis ir padeda užtikrinti aukštesnį naudotojų pasitenkinimą.