2012 IEEE International Conference on Systems, Man, and Cybernetics
October 14-17, 2012, COEX, Seoul, Korea

Dashboard by-Example: A Hypergraph-based approach to On-demand Data warehousing systems

Duong Thi Anh Hoang*, Thanh Binh Nguyen†, A Min Tjoa*
*Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria
Email: {htaduong,amin}@ifs.tuwien.ac.at
†International Institute for Applied Systems Analysis (IIASA)
Vienna, Austria
Email: nguyenb@iiasa.ac.at

***Abstract*—The usefulness and ease of use of dashboards are essential elements in supporting interactive queries in data warehousing systems, as they provide the analysts the view of critical business metrics that reflect the business performance. However, on-demand dashboard development presents a challenging task for linking semantic equivalent data facts, discovering structural dependencies between and within data sources. In this paper, we introduce an improved approach for dashboard development that is supported by by-example framework and semantic-based data linkage management. In this Dashboard-by-example (DBE) framework, hypergraph-based techniques are adopted to acquire knowledge from heterogeneous and disparate data into homogeneous ontological clusters and partitions. Moreover, the use of hypergraph-guided data linkage provides multiple perspectives on the knowledge space, supporting interaction model to explore, contextualize and aggregate the data depending on the application needs. The applicability of our approach is discussed in a case study scenario, highlight the flexibility and efficiency in handling on-demand requirements in data warehousing systems.**

***Index Terms*—Data warehousing, by-example, query processing, dashboard, hypergraph-based**

I. Introduction

The diversity and complexity of analytical requirements has sparked a growing interest in On-demand Data warehousing systems (DWH systems) [1], where interactive analytical data processing often makes a qualitative difference in data exploration, monitoring and other tasks. This has led to the popularity of dynamic dashboard applications which monitor rapid, flexible customization of information to users' specific needs [2]. In these environments, the personalization of user interactions and context representations is most important where formulating queries should be easily, effectively and should not require programming skills.

However, the use of large amounts of large-scale information has become extremely cumbersome, which mainly relates to exploring the data, rather than actually storing or exchanging it. Critical knowledge may be hidden in the huge amount of data, and the cost of exhaustively identifying, retrieving and reusing information across this fragmentation is very high [3]. Unfortunately, schema-based techniques lack the ability to allow computational linkage using domain information. In this context, discovering of interesting collections of data facts from multiple sources [4] is a significant problem of current dashboard design process. Meanwhile, the usage of traditional query languages [5] requires knowledge on the language syntax, technical background, information on both the system application domain and its interaction mechanisms.

In this paper, we present a knowledge-based QBE method that maintains the flexibility and performance in the personalized dashboard development process. The proposed DBE framework takes advantage of ontological knowledge to formalize the content and context of heterogeneous data sources and structures it in a semi-supervised manner. As an enrichment of QBE paradigm, DBE query language defines queries not only on the base attributes in the data sources, but also on the semantics of the base data encoded in the ontology and the connections between the analyzed data and the ontology data. Moreover, this query formulation mechanism is to address the urgent need of exploiting implicit knowledge relationships from various viewpoints to eventually obtain integrated understanding of the ontology and its target data domain.

With the fact that ontology exploration is closely related to the well-known problem of hypergraph transversals [6], we employ hypergraph clustering and partitioning techniques [7] to provide rich sources of information for semantic browsing and data linkage tasks. Based on this strategy, dashboard development can then be supported in pulling together and designing of different views of the data resources, in a way for managing the exploration of large and complex search spaces. We illustrate our approach using a real-world scenario, demonstrating the benefits of our approach with regards to the design flexibility of the solution.

The rest of the paper is organized as follows. Section 2 reviews related background of this work. In Section 3, we introduce the basic concepts of the by-example query language for handling dashboard designs. In section 4, we give an overview of our conceptual DBE framework along with the process of capturing semantic hierarchies, based on core idea of ontology and hypergraph theory. Finally, section 5 will conclude with a summary and an outlook on further research.

978-1-4673-1714-6/12/$31.00 ©2012 IEEE

II. Related works

Extending the related works through developing correspondence between the user given query scenario and the result dashboard design, this discovery process is rooted in several areas of research, including the trends and concepts of QBE approaches, the use of hypergraph theory in supporting data modeling as well as its utilization in the querying process.

QBE is a well-known concept and has been very popular since its introduction decades ago [8] and its variants are currently being used in most modern database products. Most research work around QBE has also been focused on enrichment and extension of QBE as a query language and developing efficient methods for generating and processing the queries defined by the examples [9]–[12], e.g. Query-by-visual-example (QBVE) and Query-by-semantic-example (QBSE). Dynamic dashboards can take advantage of a query-by-example interface, which users can draw the sequence that they are looking for as a query, specify flexible search criteria, and search for similar scenarios in a more efficient, and user-friendly manner.

However, due to the complexity of the semantic information associated with the data sources, the underlying query is defined not only on the base attributes, but also on the semantics and the connections of the base data. Currently, query expansion appends the original query with specialized terms that have a statistical co-occurrence relationship with original query terms [13]. Although appending such specialized terms makes the expanded query a better match, the expansion is not scenario-specific, which leads to the retrieval of data that are irrelevant to the original query's scenario.

The applicability of hypergraphs for modeling paradigm was studied in [14]–[16] and query languages for graphs and hypergraph-based representation have been developed, e.g. HQL defined for the Hypergraph Data Model [17]. Inspired by the hypergraph formalism [18], our approach employs hypergraphs for multidimensional modeling [19] as well as the knowledge space used to represent data information that facilitate also the process of dashboard design . The mathematic soundness of hypergraph theory [14], [15] enables the flexible relationship mechanism for a dashboard development process where hypergraph builder analyzes the ontology-based descriptions of the published data sources, and the query solver analyses the hypergraph, given example data that specifies the set of inputs and outputs of the desired data dashboard.

III. The definition of Hypergraph-based QBE language

In our framework, we assume that all the necessary data can be stored in a data warehouse with the availability of meta information about the content of and access mechanisms to the sources. Dashboard template can be associated with one or more scenarios, and may refer to several metric groups. When the scenario is displayed on the dashboard, the information about all the metrics corresponding to the dashboard templates is presented on the interface.

A. Illustrative scenario

This section provides a simplified scenario development related to analyze the food sales indicators [1]. Figure 1 includes two different seasonal timelines related to dashboard scenarios for different areas. The first scenario covers the data period from the start of 2009 through the end of August 2010. Meanwhile, the second scenario deals with how food sales is affected by a number of factors, including food production, political stability, infrastructure and natural hazards. The dashboard related to these activities requires access to data for many different resources, such as crops and livestock, at varying levels of detail. Information on food production (demand and supply), price levels, and population is also expected.

Therefore, the dashboard development problem may be categorized into three important dimensional issues, (1) granularities of location and time, (2) overlapping time domains and (3) the aggregation and disaggregation of information at different dimensional hierarchies. In this context, to ensure that decision makers have all the necessary information and to ensure that analysts have an opportunity to explain why things may turn out differently than anticipated, it is required to identify the key indicators, though they may not be included in the main scenario, which are possible and would result in different food sales analysis than those identified.

B. Dashboard data model

The dashboard model artifacts [2] used in our approach are related to modeling the data that are necessary for the dashboard, including the data and the indicator models. Moreover, the dashboard model aslo define an abstract presentation layer, including navigation and the dashboard template models.

- *Dashboard navigation model*: In a typical scenario, the analyst starts by defining some scenario concepts, and then associates these scenario concepts with indicators widgets. In the last step, DBE introduces a navigation element, in order to capture the navigation paths among the indicator widgets, which eventually form the dashboard reports.
- *Logical Connectives*: Given a chosen value for a widget, we can bind a subset of tuples in the related indicators widgets. The decomposition is done by hierarchically partitioning the widgets in the dashboard.
- *Interaction Semantics*: On-demand dashboard design have to deal with different kinds of interactions, including users adding a new value to the set of chosen values. Therefore, interaction semantics involves a consistent view of the data and allow the user to traverse to other view of data in a correct and complete manner.

The dashboard definition within the DBE illustrated in Figure 2 involves in understanding the multi-perspective dashboard by means of the concepts known in the core ontology. Based on the QBE ideas, the users can interactively provide a small number of examples that satisfy the dashboard they has in mind. Using these examples as seeds, DBE can derive a set

[1] www.fews.net/

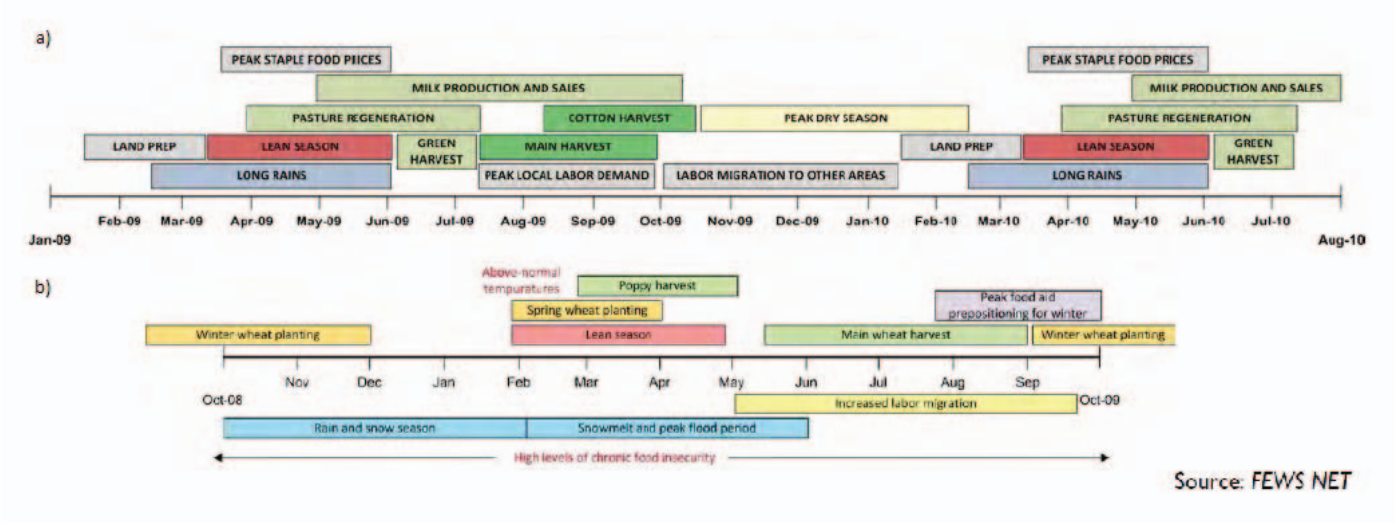

Fig. 1. Seasonal calendar and critical events for food sales

of concepts that are related to the key concept of the given query. However, only a subset of these concepts are relevant to the given dashboard scenario, and the knowledge space can be incomplete and fail to include all possible query scenarios. This is where the hypergraph-based ontological clusters can be applied to further explore the semantics of the query from the given examples and related ontology clusters.

For an undefined scenario, we retrieve from the knowledge space an incomplete relationship graph. Starting from this set of hypergraph transversals, DBE explores each edge of the hypergraph, one edge in each step, and generates a set of candidate transversals by computing all the possible unions between the candidates generated in the previous step and each vertex in the considered edge. When no more hyperedge remains, the algorithm backtracks to the previous level and picks the next transversal, etc. This new relationship graph will be appended to the knowledge space and can be used for future related dashboard scenario.

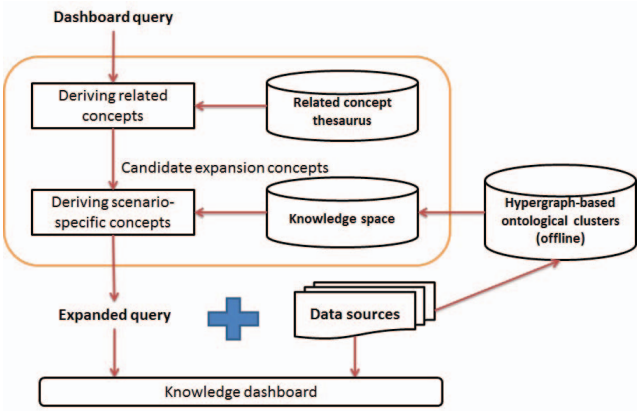

Fig. 2. Dashboard retrieval processes

C. The intuition of DBE language

The introduction of the hypergraph structure is a generalization of graph theory, in which a hypergraph is the family of these edges (called hyperedges). In this approach, a DBE query is seen as a query graph specifying the set of inputs and outputs of the desired data, eased by a suitable interface that displays the available concepts (e.g., with an expandable structure) and highlights the equivalences between concepts.

Each condition hyperedge is a restriction on a property of the query inputs. Relation hyperedge can can be expanded to allow sub-hypergraph, called query paths, which allows one to navigate through the underlying dataset and build complex queries. A query can return either a node or an edge. For query formulation, we only need to retrieve the last node/edge in the path; that is, we need to retrieve either the edge or the node. For example, a query path as an expression of the form: $Wheat\ Sales\ O_2\ P_2\ \ldots\ P_n\ O_n$, where O_i is a node, and P_i is an edge, retrieves the properties of wheat sales; or $Wheat\ Sales\ O_2\ Prices\ \ldots\ P_n\ O_n$ retrieves the prices of wheat sales.

In this way, users can navigate and query data source(s) without any prior knowledge about it. We may now construct the search space, which consists of the access paths and operations on base nodes, and various groupings of related nodes. Thus, hypergraph model is used for discovering and traversing data entities (commonly called ”nodes” or ”vertices”) linked by associations (edges) to reveal relationships at multiple levels of separation. Such structures are more flexible than any that can be represented relationally (which cannot natively support recursive relationships), and the edges enable much more rapid traversal of relationship structures than is possible through relational joins [17], [18].

D. The syntax and semantics of DBE language

One of the important abstraction in the query language is the classification of the building commponents in a DBE design

which can be defined into various generic classes as *Constant elements* which the user cannot interact directly with; *Indexed example elements* which enable the user to choose a value using the indexing capability, in the food sales example, this provides the capability to state range restrictions such as the presentation of *sales with wheat price less than 4 dollars per bushel*irrespective of whether *sales with 4 dollars per bushel exists or not*; or *Set example elements* which allow the entire data to be presented to the user in some construct and allows the user to choose a subset of the data, e.g. hypergraph and hierarchy constructs.

There are various types of semantic links, i.e. associating elements to the attributes of the target data sources; and hierarchical decomposition of the elements. Every elements is associated with one or more attributes in the data sources and each association is classified either as for output or a condition. Moreover, to reduce the data to be presented and the dimensionality of the data, the decomposition is done by hierarchically partitioning the elements and ensures the nested level consistency. Each node in the hierarchy is called a level and reduces the data to be presented in the lower level. In food sale example, the dashboard is a multidimensional widget that is nested below the sliders for Price and Supply total.

Intuitively, a dashboard design should present a consistent view of the data and allow the user to interact with other view of data in a correct and complete manner. There are various possible interactions to dashboard design, e.g. Choosing values in Constant elements or Indexed example elements; Choosing subsequent values in Set example elements, adding a new value to the set of chosen values. To assure the view of the data presented to the user must be consistent with the target data sources, the dashboard have to satisfy that the set presented in the nested levels is the partition that corresponds to the chosen value at the previous levels.

IV. Knowledge-based Dashboard-by-example Framework

By definition, the DBE holds an integrated view and focuses on using a semantic knowledge dashboard at different granularity levels to support users in quickly gathering a broad insight of their datasets from differing perspectives. Regarding data management for an on-demand data warehousing system, DBE knowledge only includes an abstract view of multiple data sources based on a set of interlinked ontologies, which structure the knowledge from the different data sources and define relations between found entities. This approach shows two benefits: firstly, sources can autonomously maintain their data, while the DBE is just responsible for suggesting to users where to find appropriate sources. Secondly, the sources can protect their data based on their own privacy regulations.

Figure 3 shows the conceptual architecture of the proposed dashboard framework, based on the use of a scenario-specific knowledge representation in the form of ontological clusters. The dashboard models capture the definition of metrics and related context information to be displayed on the dashboard as well as the interactive navigation paths. The series of

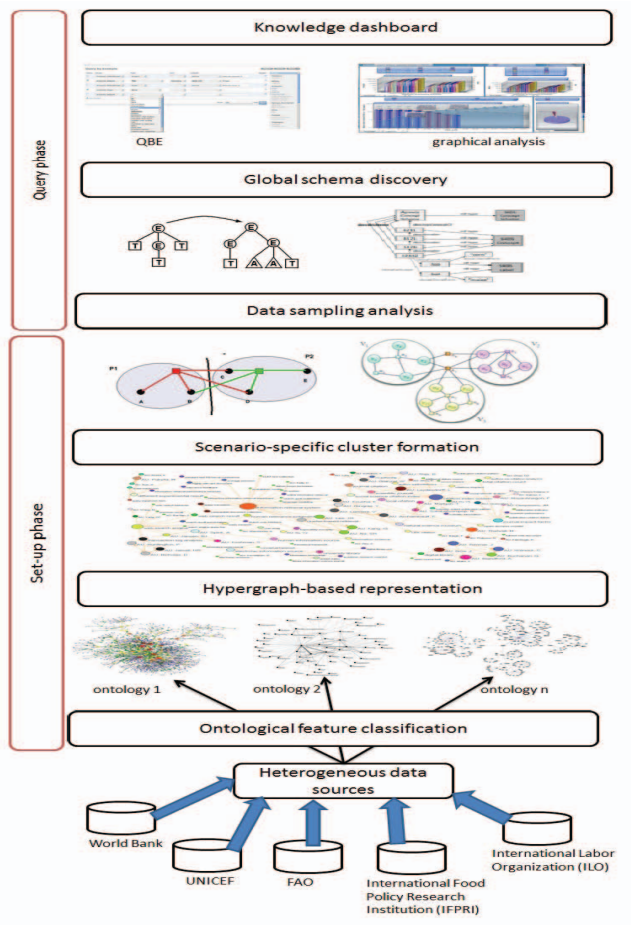

Fig. 3. General architecture of DBE framework

steps performed as part of our DBE framework are aimed at discovering data linkage across multiple data sources with minimal user involvement.

Our semantic dashboard by example approach consists of a set-up phase and query phase. In the knowledge acquisition phase, the base table data is scanned to associate the related ontology instance with each record. In the dashboard design phase, a user poses a query on the integrated data tables by giving a set of example tuples that satisfy the query. The on-line phase exploits the hypergraph-based data linkages to a model from the examples supplied, appending the original query with additional terms that are specifically relevant to the query's scenario(s). The base data is scanned again, and the DBE decides what are the tuples satisfy the query.

Most of semantic web applications use ontologies as vocabularies to describe metadata and are aimed at semantic processing of them [20]. Regarding ontology as the exploration target, the main DBE approache is composed of: (1) the systematized conceptual structure of ontologies and (2) on the fly conceptual structures identification from the ontologies to cope with various dashboard interests. Whenever the DBE is unable to define the dashboard data view because the local data does not lead to a possible answer, it may forward the query to

neighboring clusters and partitions. A transversal result of the defined hypergraph then corresponds to a set of data sources that cover all the indicators requested by the user.

A. Hypergraph-based knowledge acquisition

In order to be able to clearly identify the driving indicators behind the dashboard and trends or patterns related to scenario-specific queries, our paper applies hypergraph indexing and transversal techniques [21] as a formal model for the process of computing scenario-specific of data sources, which is a complementary rather than an alternative to the current QBE approaches. The idea of the ontological hypergraph is to summarize ontological concept graph, so that the background queries can be answered from this summary. Because the size of the summary is smaller than the original graph, queries can be faster. Given an ontological graph, its ontological hypergraph is a twofold summary: of the original graph such that nodes that have the same outgoing paths are grouped together and of a graph summerized by grouping nodes that have the same incoming paths.

1) Hypergraph builder : The hypergraph builder analyses the ontology-based descriptions of the registry-published data sources in order to build a labeled directed hypergraph, which synthesizes all the data dependencies of the data services. Although this module performs a time consuming task, it does not affect the efficiency of the dashboard design process, as the hypergraph construction is completely query independent and can be pre-computed off-line before query answering time [6]. The vertices of the hypergraph constructed by the hypergraph builder correspond to the concepts defined in the ontologies employed, while the hyperedges correspond to a set of vertices which offer particular ontological relationships, e.g. is-a (subclass-of) relations, part-of or attribute-of relations.

The hypergraph builder firstly adds to the hypergraph the concepts defined in the ontologies. Next, it draws the hyperedges representing the relationships and the dependencies between the added ontology concepts. Moreover, the inter-source dependencies are directly represented by the hypergraph and updated whenever a source is added to the registry.

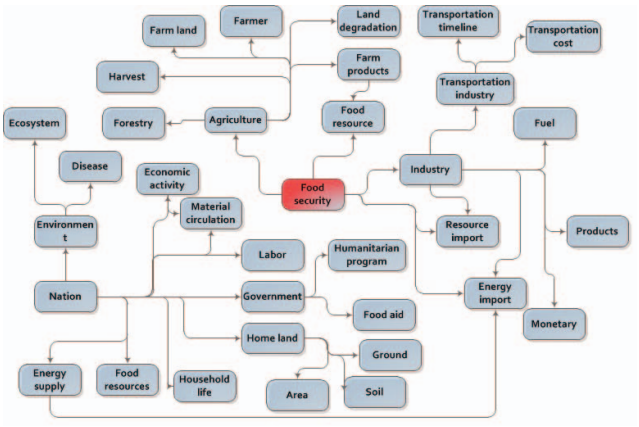

Fig. 4. A simple dependency hypergraph of Food indicators

Figure 4 shows example of the resulting hypergraph based on the information in the Food Systems Network[2]. The concepts are associated with their corresponding structures, showing a set of relatively simple relationships among a set of structural domains in one of the initial partitions of the hypergraph.

2) Hypergraph definition of scenario-specific views: Viewpoints are defined as a combination of a main concept and related aspects, and an aspect is represented according to relationships defined in an ontological hyperedge. In order to discover the flow of semantic information in multiple dimensions, the framework performs clustering process at different levels to capture different sets of ontological concepts [22]. At the concept level, case entities are collection of indicator concepts along with their ontological resource to perform structural level matching in a cluster. The results obtained from concept level clusters are used to perform field matches during schema scans. Meanwhile, at the instance level, case entities are collection of source data formed in the data sample analysis stage. Finally we end up with a set of structural domain clusters. Each cluster is characterized by combinations of memberships to indicators families.

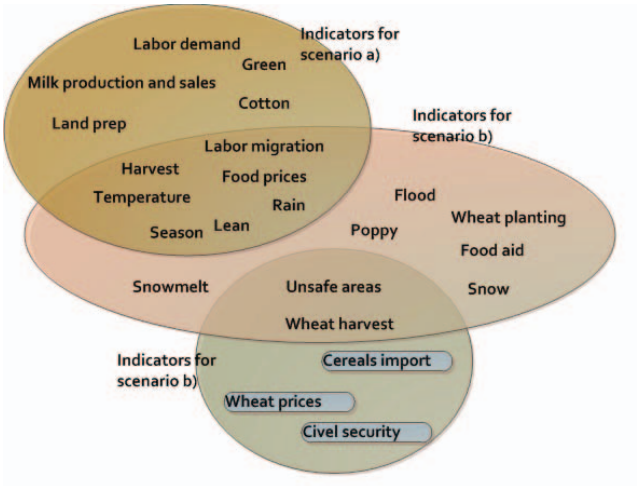

Fig. 5. Indicators families and cluster borders

Since each hyperedge represents a sequence property shared by structural domains, a partitioning can minimize the relationships among structural domains across the different partitions. This cluster mapping strategy enables the structural relationships between different ontological clusters in different arrangements. Instead of any predetermined order, this approach is used in an anticipation of capturing the flow of meaningful semantic data and to construct self-expanding hierarchical semantic trees crucial for discovering correlations between clusters, thereby optimizing a combined indicator measure.

The given query hypergraph is first traversed and a search path list of predecessor, successor query vertex is defined, by considering the compositions of vertex as well as the

[2]www.fews.net/

ontological relationships to cope with different concepts. The large number of vertices taken into account by the search component can be reduced by introducing a suitable vertex pre-selection phase. The efficiency of the query formation can be improved by enumerating the non-deterministic generation of the possible solutions in order to return the successful vertices only. The sub-hypergraph search algorithm naturally allows the user to provide disconnected query graphs. In this case, the result hypergraph is obtained as a union of the results for the individual disconnected query graphs.

B. A simple DBE example

Imagine that an analyst is developing an dashboard using scenario a and retrieving "The rice sales in relations with Farm price, Supply total, Total demand, World price from 2009 to 2010". This initiates several queries to be sent to the backend from the interface. For example, in this scenario, the concepts derived from the ontology can be *Farms for food production, Food aids, Problem of fixed area, Rise in food prices* and the corresponding relationships clarified.

Analysts are then provided with related indicator elements, each of which presents different facets based on different relations in the underlying semantic knowledge (figure 5), i.e *Rice prices, Government budget* and *Environment factors*. These indicator elements present the entire data to be presented in graphing or hierarchy constructs and allow users to choose a subset of data. Thanks to the semantic knowledge and the hypergraph-based ontologies, the dashboard goal can be visualized at different levels of granularity. From the multiple layers a summation of the knowledge emerges that can highlight previously unseen patterns. Supposed that a new dashboard scenario focuses on wheat sales. Based on scenario b design and the ontologcal clusters in figure 5, DBE could indentify the key indicators which are particularly important to the changing dashboard outcomes. Thus, this assumption will affect other indicators about *crop harvest, labor demand, food prices* , and, eventually, *food sale details*.

V. Conclusions and Future works

Inspired by the growing need of on-demand business analysis, we presented a hypergraph-based QBE approach for developing dashboard design satisfying given user query. In this approach, the multidimensional hypergraph, provides more flexibility for changing user requirements in the condition of highly dynamic analysis. Specifically, on top of the conceptual model for BDE we now describe how complex data can be organized into dashboard components with related interaction paths. We are working on extending the model to detailed DBE query language which model the semantics of user interactions in a declarative way, providing a quick and on-demand specification of dashboard interfaces. The implementation design for our DBE framework has been designed and are under development, enabling us to empirically evaluate the applicability and impact of the proposed approach.

References

[1] T. B. Pedersen, D. Pedersen, and K. Riis, "On-demand multidimensional data integration: toward a semantic foundation for cloud intelligence," *The Journal of Supercomputing*, Oct. 2011.

[2] T. Palpanas, P. Chowdhary, G. Mihaila, and F. Pinel, "Integrated model-driven dashboard development," *Information Systems Frontiers*, vol. 9, no. 2-3, pp. 195–208, May 2007.

[3] S. Mazumdar, A. Varga, and V. Lanfranchi, "A Knowledge Dashboard for Manufacturing Industries," *Proceedings of the 1st International Workshop on Ontology and Semantic Web for Manufacturing*, pp. 51–63, 2011.

[4] T. B. Nguyen, A. M. Tjoa, and O. Mangisengi, "MetaCube-X: An XML metadata foundation for interoperability search among Web warehouses," in *Proceedings of the 3rd International Workshop on Design and Management of Data Warehouses*. Citeseer, 2001, pp. 1–8.

[5] T. Catarci and L. Tarantino, "A Hypergraph-based Framework for Visual Interaction with Databases," *Journal of Visual Languages & Computing*, vol. 6, no. 2, pp. 135–166, Jun. 1995. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1045926X85710087

[6] A. Brogi, S. Corfini, and J. Aldana, "Automated discovery of compositions of services described with separate ontologies," *Proceedings of the 4th international conference on Service-Oriented Computing ICSOC'06*, pp. 509–514, 2006.

[7] D. Papa, "Hypergraph partitioning and clustering," *Approximation algorithms and metaheuristics*, pp. 1–38, 2006.

[8] M. Zloof, "Query-by-example: the invocation and definition of tables and forms," in *Proceedings of the 1st International Conference on*, 1975, pp. 1–24.

[9] I. Song, "Semantic Query-by-Example for RDF data," in *Proceedings of the First International Conference on Emerging Databases*, 2009, p. 8.

[10] P. Moreno and N. Vasconcelos, "Bridging the Gap: Query by Semantic Example," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 923–938, Aug. 2007.

[11] N. Rasiwasia and N. Vasconcelos, "A study of query by semantic example," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Ieee, Jun. 2008, pp. 1–8.

[12] L. Lim and H. Wang, "Semantic queries in databases: problems and challenges," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009*, 2009, pp. 1505–1508.

[13] Z. Liu, "Knowledge-based query expansion to support scenario-specific retrieval of medical free text," *Information Retrieval*, pp. 1–32, 2007.

[14] G. Gallo, "Directed hypergraphs and applications," *Discrete Applied Mathematics*, vol. 42, no. 2-3, pp. 177–201, Apr. 1993.

[15] G. Gallo and M. G. Scutellà, "Directed hypergraphs as a modelling paradigm," *Decisions in Economics and Finance*, vol. 21, no. 1-2, pp. 97–123, Jun. 1998.

[16] E. Grabska, B. Strug, and G. ŚŚlusarczyk, "Hypergraph-based Evolutionary Design System," in *Proceedings of the 10th Generative Art Conference, GA2007*, 2007, pp. 41–51.

[17] D. Theodoratos, "Semantic integration and querying of heterogeneous data sources using a hypergraph data model," in *Proceedings of the 19th British National Conference on Databases: Advances in Databases*. Springer-Verlag London, UK, 2002, pp. 166–182.

[18] A. Sarkar, S. Choudhury, N. Chaki, and S. Bhattacharya, "Implementation of Graph Semantic Based Multidimensional Data Model: An Object Relational Approach," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 1–26, 2007.

[19] D. Hoang and T. Priebe, "Hypergraph-based multidimensional data modeling towards on-demand business analysis," *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services iiWAS '11*, pp. 36–43, 2011.

[20] K. Kozaki and T. Hirota, "Understanding an ontology through divergent exploration," *Poster & Demo Notes of the 7th International Semantic Web Conference (ISWC 2008)*, pp. 305–320, 2011.

[21] T. Eiter, "Identifying the minimal transversals of a hypergraph and related problems," *SIAM J. Comput.*, vol. 24, no. 6, pp. 1278–1304, 1995.

[22] M. Gollapalli, X. Li, and I. Wood, "Ontology Guided Data Linkage Framework for Discovering Meaningful Data Facts," *Proceedings of the 7th international conference on Advanced Data Mining and Applications ADMA'11*, pp. 252–265, 2011.