Monitoring and Diagnosing Indicators for Business Analytics

Konstantinos Zoumpatianos, Themis Palpanas, John Mylopoulos
Department of Computer Science
University of Trento
Trento, Italy
{zoumpatianos,themis,jm}@disi.unitn.eu

Alejandro Maté, Juan Trujillo
Lucentia Research Group
Software and Computing Systems
University of Alicante, Spain
{amate,jtrujillo}@dlsi.ua.es

Abstract

Modeling the strategic objectives has been shown to be useful both for understanding a business as well as planning and guiding the overall activities within an enterprise. Business strategy is modeled according to human expertise, setting up the goals as well as the indicators that monitor activities and goals. However, usually indicators provide high-level aggregated views of data, making it difficult to pinpoint problems within specific sub-areas until they have a significant impact into the aggregated value. By the time these problems become evident, they have already hindered the performance of the organization. However, performing a detailed analysis manually can be a daunting task, due to the size of the data space. In order to solve this problem, we propose a user-driven method to analyze the data related to each business indicator by means of data mining. We illustrate our approach with a real world example based on the Europe 2020 framework. Our approach allows us not only to identify latent problems, but also to highlight deviations from anticipated trends that may represent opportunities and exceptional situations, thereby enabling an organization to take advantage of them.

1 Introduction

Modeling the business strategy has been shown to be useful both for understanding a business [3] as well as planning and guiding the activities within an enterprise [11]. Most enterprises represent the business strategy in a textual fashion, captured in the business plan. Then, once the business plan has been established, the business intelligence system of the organization helps to monitor business performance by means of Key Performance Indicators (KPIs) [16]. For example, an organization may have the goal of "Increasing revenue" by achieving "Increasing market share". These goals could be monitored by the KPIs, "Revenue" and "Market share".

Traditionally, organizations are interested in monitoring these KPIs in order to identify unwelcome or unexpected situations, either positive or negative, that affect their goals. Currently, this kind of analysis is done by measuring how close or how far the values of KPIs are from their targets [16], defined in terms of an acceptable range. It is often the case, however, that these KPIs do not reflect the anomalies within the sub-areas that are being monitored by the KPI, since they represent high levels of aggregation. We refer to these sub-areas as *sub-markets*. For example, sales may have decreased dramatically in Trento, but this may not be noticeable by looking at national aggregates for Italy. Moreover when KPIs do deviate significantly from their target values it is mostly the task of an analyst to seek the reasons behind these exceptional events. Finally, such KPI deviations are only monitored in isolation, non-systematically, rather than within the context of the strategic model. As a result, the impact analysis for a deviation is limited.

Although current dashboards and scorecards allow users to analyze the data in detail, performing the monitoring and analysis processes manually can be a daunting task, since (i) the underlying data warehouse commonly contains multiple dimensions [13], thus making analysis a time-consuming process, (ii) identifying a significant event is challenging, due to the knowledge required to interpret the data for each specific part of the market, and (iii) explaining the results in the context of a strategic model is even more difficult, since it presupposes understanding how the results may affect every relevant goal and indicator.

In order to tackle these problems, we propose a semi-automated method for generating a set of monitoring and diagnostic queries from a strategic point of view, in order to (i) identify unexpected and/or unwelcome situations in the context of a strategic model, (ii) explain why these situations occur, and (iii) identify how they may affect other strategic elements. Furthermore, we show how all the steps in our approach can be applied to a real case by means of an illustrative running example based on the Europe 2020 framework, to be described in the following section.

Specifically, in our approach, we monitor groups of related indicators for identifying if certain goals are going to be achieved on time. Moreover, we are able to detect outliers within such groups that could point us to anomalies that should be either corrected or confirmed, i.e., “Why Sales have dropped in Barcelona but not in Madrid?”. Additionally, we are able to explain such outliers by focusing on specific areas of the data warehouse that are responsible for these anomalies. Finally, we consider that our approach could be applied to work with big data analysis techniques, thus allowing the results to be evaluated from an strategic perspective and helping users to understand what these results mean for the business.

The remainder of this paper is structured as follows. Section 2 presents the background work, including an illustrative example. Section 3 describes an overview of the steps involved in our approach applied to our case study. Section 4 presents the related work. Finally, Section 5 discusses the results and sketches directions for future work.

2 Background & Problem Formulation

2.1 Modeling the Business Strategy: An Illustrative Example

In this section we describe the basic elements within the business strategy in our approach, by means of an illustrative example based on the Europe 2020 framework.

The Europe 2020 framework aims to specify a set of strategic goals to be met by the European Union (EU) by 2020[1]. Some of these goals have clear target values and indicators established, allowing the EU to monitor their performance and be aware of deviations from the initial plan. Additionally, as the EU is integrated by several countries, each one of them with its own characteristics such as population, industry, etc., each country has its own particular objectives. As a result, under-performers may be compensated by over-performers, resulting in a high level indicator that correctly shows the UE is meeting its goal, without reflecting outliers that represent a potential threat. Furthermore, much like in traditional business plans, the descriptions provided in the framework only highlight a handful of relationships between goals, with no claims of completeness or consistency. We can see an example of these goals, indicators and the relationships between them in the Europe 2020 employment axis represented in Figure 1.

In order to model the business strategy, we make use of a simplified version of the Business Intelligence Model (BIM) [9]. According to this metamodel, the elements involved in the Europe 2020 framework are as follows:

First we have **Goals**, which capture the objectives of the organization to be achieved. There are three kinds of goals: Strategic (long-term), Operational (medium-term) and Tactic (short-term).

[1] For more information see: ec.europa.eu/europe2020/index_en.htm

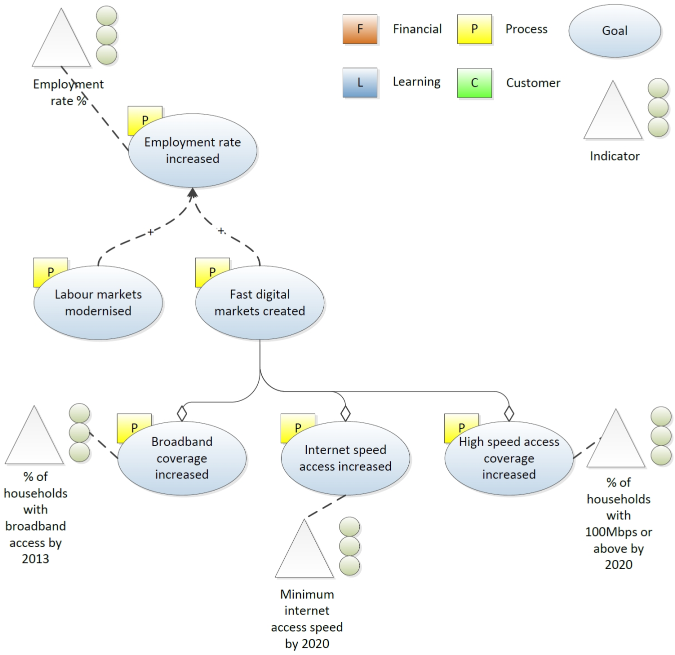
F
Financial
P
Process
Goal
L
Learning
C
Customer
Indicator
Employment rate %
P
Employment rate increased
+
+
P
Labour markets modernised
P
Fast digital markets created
P
Broadband coverage increased
P
Internet speed access increased
P
High speed access coverage increased
% of households with broadband access by 2013
Minimum internet access speed by 2020
% of households with 100Mbps or above by 2020

Figure 1: Excerpt of the European 2020 strategic model

Within the Europe 2020 framework we have Strategic Goals that define the axis of the strategy, such as "Employment rate increased", "Expenditure on R&D increased", or improve "Environmental care". As some of these goals can not be monitored directly, they are refined into additional strategic goals. For example, "Environmental Care" is refined into diminishing "Green house gas emissions", increase the "Share of renewable energy in gross final consumption", and diminish "Primary energy consumption". In order to achieve these high-level goals, a set of Operational Goals, that influence the strategic goals, are defined. Examples of operational goals are "Fast digital market created" or "Individual skills developed". Finally, since Europe 2020 is a long-term plan, it includes no Tactic goals.

Second, we have a set of **Indicators** that monitor the performance of Europe 2020 goals, and alert about deviations in the targeted values. Some of the indicators included in the Europe 2020 strategy are "Employment rate %", "Early leaver % between 18 and 24", or "Index of Greenhouse gas emissions". Each of thse indicatorsr presents a target value (value to be achieved), a threshold (margin between good and bad performance), a current value and a worst value. According to these values we can analyze how much we are deviating from our targets. In addition to these attributes, in this work we extend BIM with two additional attributes. The first one is the "time to target" often employed in scorecards and required in order to perform time series analysis. This attribute describes how much time is left to achieve the expected target value. The second attribute is the refresh rate of the indicator, which is required for monitoring purposes, and describes how often the indicator should be monitored.

Third, we have **Situations**, which describe external or internal influences that may affect the business strategy and its goals, positively or negatively. An example situation would be "Economic Crisis". However, since the Europe 2020 framework does not explicitly mention this situation (it is managed by the EU on its own), it has been omitted from the model.

Finally, we add the concept of **Strategic query** over the model. A strategic query formalizes a goal, allowing to check if it is achieved or not. A simple query for "Employment rate increased" goal would be $EmploymentRate >= 75\%$. However, these queries can be made more complex, including subtargets $\forall C_i \in Countries, EmploymentRate_C >= Target_C$ and trends $EmploymentRate_{2020} >= 75\%$, or involving multiple goals [9].

By modeling these elements, and by associating KPIs to business goals, we obtain a clear view of the strategy that our business is following, as well its current status.

2.2 Monitoring a Business Strategy

Monitoring a business strategy by relying only on KPI information (current vs target values, time left) can conceal lurking problems, especially when KPIs are highly aggregated. Therefore, it is necessary to issue queries that monitor the evolution of both KPIs and sub-areas within these KPIs periodically.

For example, consider again the case of Europe 2020. The aggregated indicator for Employment Rate currently shows a deviation of less than 7% compared to its target value. However, is this deviation distributed equally among each country? It may be known that in some countries unemployment rates are increasing, leading to increasing deviations from their targets. However, the community as a whole may not be aware of these deviations until they become problems that threaten the global target.

In order to adequately monitor these situations, we need to pose strategic queries. For example, in the case of Europe 2020 a manager could pose queries described in natural language such as "Is it expected that we meet our Employment Rate goal?" This query would derivate into several other queries like "Are there other countries displaying a similar pattern to those that are struggling?" "What countries are close to their own sub-targets?" The information gathered from these queries helps to monitor the status of the strategic goals and identify potential problems arising, even if the aggregated indicator is not accurately reflecting them. However, it is often the case that anomalies are latent and thus, the number of monitoring queries that can be posed increases exponentially, thus complicating the monitoring task. Therefore, in order to restrict the search space it is necessary to determine two aspects:

1. **Dimensionality**. What are the relevant dimensions that we are interested in analyzing in detail? In Europe 2020, we are interested in finding anomalies within Country and Time dimensions, whereas if we were analyzing the evolution of sales we might be interested in Customer Segments. These relevant dimensions are not necessarily those restricting the calculation of the indicator to be monitored.

2. **History**. What are the relevant periods of time for the queries of interest? In the case of Europe 2020, we may be interested in analyzing only the period after the start of the economic crisis, or we may actually want to consider the data before the beginning of the crisis.

Finally, once these aspects have been determined, we need to transform strategic queries into one or more data warehouse queries that monitor the strategy and identify potential outliers and anomalies that can be highlighted for the analyst.

3 Proposed Approach

In this section we describe our proposal for monitoring the business strategy and detecting anomalies. An overall view of our process can be seen in Figure 2. First, we gather input from the user. This input is composed by the set of goals, situations, and KPIs to be monitored. Then, for each KPI, we gather the relevant dimensions from the data warehouse that should be considered (Dimensionality) and the relevant period of time (History). As a data warehouse may contain several dimensions, performing a search on every possible combination can be very costly, thus the initial knowledge from the user can speed up the analysis process. After gathering the input from the user, we proceed to the setup step. The setup step calculates the necessary data for the analysis. Specifically, we identify the target values for each sub-market to be considered. Afterwards, we proceed to the monitoring step. In the monitoring step, a set of algorithms analyze the data and identify the existence of deviations and outliers, whether in the aggregated values or in any of the sub-markets.

Our process requires us to be able to decompose business strategies into components, and specialize them to lower level sub-markets. All of the above

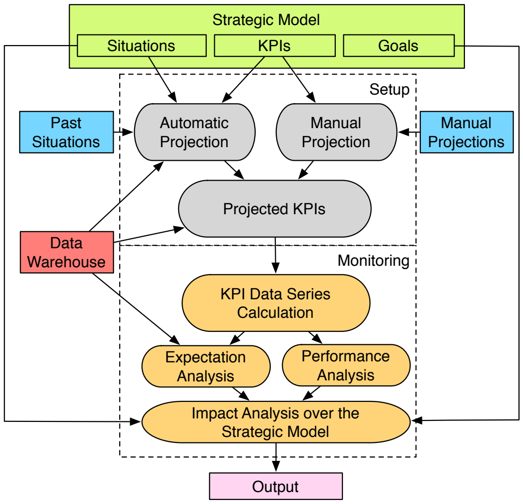

Figure 2: Overall view of the monitoring process

in an automated manner that will assist analysts to monitor their strategic goals by pointing out:

- The performance for each goal based on the targets for their KPI values. Either at the highest level of aggregation, or at different submarkets. *For example we might need to know that a specific goal has failed for Western Europe, even if it has succeeded for Europe in overall.*

- The KPIs that demonstrate unexpected behavior with regards to previously correlated markets, or their parent market segments. *For example we might need to know if Italy is following a different trend than the rest of Europe, or if Italy is following a different trend than Greece, even though they were correlated in the past.*

Based on the above requirements, we define two different kinds of diagnostics:

- **Performance diagnostics:** How are we doing with regards to a goal, based on the KPI value and our targets?

- **Expectation diagnostics:** Is the current value/trend expected based on the data in other parts of our data warehouse?

After providing an overall view of the process, in the following subsections we describe our process in detail using the Europe 2020 framework, previously introduced in Section 2.

3.1 User preferences

The first step in our process is gathering user preferences. Recently, employment rates in certain countries as well as the importance of education have been a hot topic. The current Employment and Education aggregated indicators do not show extreme deviations, and we have no knowledge about the potential influences among them. Therefore, as users, we will choose to focus on the Employment and Education axis from the Europe 2020 framework. The description of these axis is as follows:

- **Employment** axis has the target of achieving a 75% aggregated Employment Rate in the whole EU. Furthermore, each country has assigned its own sub-targets, such as 74% for Spain, or 67% for Italy. This axis is modeled in Figure 1.

- **Education** axis is subdivided into two main indicators. The first, measures the rate of **Early leavers from education and training**, and its target value is set to 10% or less of abandonment rate for people between 18 and 24 years. In addition to having different subtargets for each country, the comparability of this indicator is restricted over countries and time, due to different implementations in the way of measuring its value. Second, the EU aims to achieve a **Tertiary educational attainment** rate of 40% or more for people between 30 and 34 years, with each country having its own sub-target. This axis is modeled in Figure 3.

Furthermore, each of these axis is supported by one or more initiatives planned by the EU commission, and grouped into pillars. Each initiative represents a course of action to be followed in order to achieve the strategic and operational goals, and may include milestones and sub-indicators to measure their progress. The pillars that support Employment and/or Education are:

1. **Smart Growth** pillar includes the initiatives “Creating a single digital market based on fast

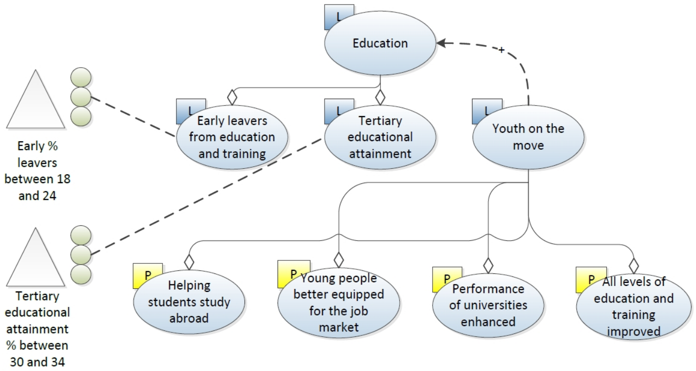
L
Education
L
Early leavers from education and training
L
Tertiary educational attainment
L
Youth on the move
Early % leavers between 18 and 24
Tertiary educational attainment % between 30 and 34
P
Helping students study abroad
P
Young people better equipped for the job market
P
Performance of universities enhanced
P
All levels of education and training improved
+

Figure 3: Education axis within the Europe 2020 strategy

and interoperable applications", focused on increasing internet speeds throughout Europe, and "Youth on the move", focused on improving individual skills and foster student mobility.

2. **Inclusive Growth** includes the initiatives "Creating an agenda for new skills and jobs", focusing on helping people acquire new skills and modernizing labor markets to raise employment levels.

In order to support the analysis of this strategic model we require a data warehouse. A data warehouse stores information in terms of facts [13], and is structured according to a Dimension Schema $\mathcal{D}$, a Measure Schema, and a Fact table. The Dimension Schema defines a set of dimensions that provide context information. The Measure Schema contains all the measures available in the data warehouse. These are real numbers on which we can apply functions and aggregate them over the different dimensions defined in the Dimensions Schema. Finally, the Fact table stores the fact data. On top of a data warehouse, analysts can define KPIs, by combining aggregations (KPI terms) into complex formulas and by assigning target values. Further on, these KPIs can be restricted to various sub-dimensions e.g. specific countries, via KPI-Restriction operations that drill down in the data warehouse, for all the terms of a KPI.

An example Dimensions Schema is that of Figure 4, where the non-sequential dimensions are $\mathcal{D}_{NS} = \{Country,\ Euro\}$, the sequential[2] are $\mathcal{D}_S = \{Time,\ Total\ area\}$, and we have the following hierarchies $I = (i_{Country} = (Region, Country),\ i_{Euro} = (In\ Euro),\ i_{Time} = (Year, Quarter, Month, Day, Hour),\ i_{Total\ area} = (Area\ in\ km^2))$.

Note that we make a distinction between sequential and non-sequential dimensions, such that we are able to monitor trends of measures over the values of sequential dimensions, and compare these trends among different parts of the data warehouse. The most intuitive sequential dimension is that of Time, where analysts need to compare how the values of various indicators fluctuate over the course of time. Other examples can be relatively stable dimensions that can not be considered measures, such as the area of a country, where analysts are interested on monitoring the trends by using the area of each country in the horizontal axis.

With these considerations, we gathered the data for our case study from the official source for Europe 2020 data, the Eurostat[3]. Eurostat provides data in the form of tables for each indicator both at aggregated EU level, as well as for each country. A subset of the data warehouse schema can be seen in Figure 4.

According to this schema, we choose to use both Time and Country dimensions for the analysis of Employment and Education indicators, and we in-

[2] A sequential dimension contains instances that have an order established

[3] http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/

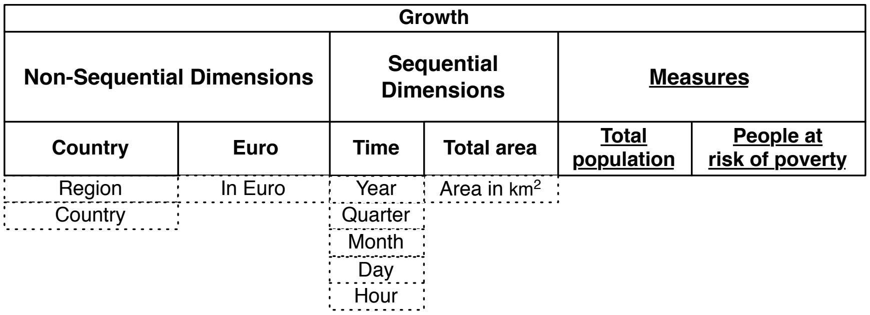

Figure 4: Europe 2020 Framework data warehouse

clude all the data available since year 2000 into the analysis.

3.2 Setup Step

After the user preferences step, we proceed to the setup step. During the setup step, if the user has not selected specific KPIs, we identify all the KPIs for each goal by scanning the strategic model. Then, for each KPI, we need to identify all the KPI target values for each of its sub-markets, in order to support **Performance Diagnosis**. This can be done by either scanning a Knowledge Base where these projections are predefined, or in case that it does not exist, use the data warehouse for adjusting these values for each specific sub-market. This is done based on the relative size of this sub-market with regards to the higher level aggregation. In our scenario, we want to analyze Employment Rate, where we can adjust the target value for specific countries by finding the ratio of the historic average employment for a specific country, over the European historic average. An issue that arises here is that there might be more than one parent markets when we have sub-markets that are restricted by more than one dimension or that have multiple hierarchies of aggregation. So we need to decide, for example, if it makes sense to compare Italy to the countries that are in Eurozone (Eurozone classification), to the countries that are in South Europe (Region classification), or to all the countries. One could consider all of them and average the ratios, or pick the one that makes more sense. In the following subsections we analyze these aspects in more detail.

3.2.1 KPI Value Segmentations for Performance Diagnostics

Being able to produce sequences of observations for each KPI, and specialize them to various sub-dimensions or hierarchy levels, will give us the raw data. But we still need to address two important points in order to support Performance Diagnosis.

1. The first problem is that of missing values. There are cases where we have data for all the years except one, or cases where we are interested on extrapolating our data for predicting future values based on the past trends. In such a scenario, we need to be able to either interpolate the data, in order to produce an approximation of a missing value, or if we are interested in forecasting future values, use a time-series forecasting algorithm to do so. This will help us in two directions:

    - First, we will be able to create complete data series even if some of the intermediate values are missing.
    - Second, we will be able to produce ahead of time (if the sequential dimension is time) performance analytics, and thus identify if a goal is going to fail or succeed in the near future.

2. The second problem is that of measuring the degree of performance success for different KPI-Restrictions. For example, given that a value $> 1\%$ of people in risk of poverty is bad, we would like to find out what this number is for other dimension restrictions e.g. per country, in/out of Eurozone, or even different

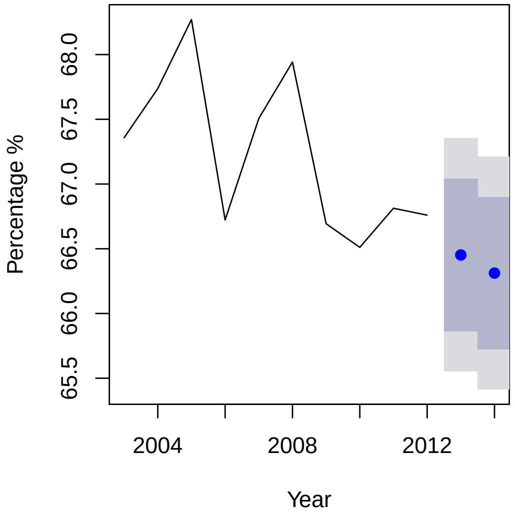

Figure 5: Total employment forecasting for EU

> years. This will help us automatically identify our target values for sub-markets as well as sub-periods for a specific high-level KPI, that is defined in a strategic model.

Missing and Future Values

Initially, given a set of observations seen so far for a KPI, we are interested in forecasting its future values, such that we provide in-time insights. The values that compose each KPI can be seen as a training set, based on which a prediction model p can be trained. This prediction model can then be used to forecast the observations that have not been recorded yet. For example, in Figure 5, we can see Employment Rate values projected into the missing years, in order to analyze if we will meet our goal.

Definition 1 (KPI Data-Series Value Forecasting Query) *Formally, given a KPI Data-Series $ds = DS_{k_i}^{d_j}(i_s, i_e)$ for a period i_s to i_e, on a sequential dimension d_j, we are interested in training a prediction model p in order to predict values of ds for time points $i_f > i_e$.*

$$forecast(DS_{k_i}^{d_j}(i_s, i_e), p, i_f) \; returns \; x \in \mathbb{R}$$

In the same spirit, we can define an operation that interpolates missing values within a certain range. In both forecasting and interpolation, queries can be answered using well known statistics algorithms. Some common forecasting methods are those of moving averages, ARIMA models and the Box-Jenkins methodology [4].

Projecting KPI Value Segmentations

As we stated earlier, being able to project target KPI Value Segmentations to KPI-Restrictions (sub-markets) is crucial for monitoring sub-markets. This is because we need to be able to monitor our expectations on a higher-level market aggregation (e.g. unemployment all over Europe) as well as for different sub-markets (e.g. unemployment in Italy, France, etc.) and not only in overall.

In order to do so we need to be able to extract the value segmentations for a KPI-Restriction in any given dimension-instance pair, either sequential or not. So for example, given that an Employment Rate of 75% is “good” for EU in overall, it is not trivial to project what is good to different sub-markets such as different countries, countries within Eurozone and more. Given this, we define a KPI Value Segmentation Projection Query that should be able to mine a database for identifying such KPI-Segmentation breakdowns.

This is a process that can be done either in an automatic, or in a manual way. This means that either the system must be able to project the value segmentation [15] to lower level markets, for example by comparing market shares of previous years; or that the analyst has to manually specify a projection function that maps segmentations to the lower-level.

The most intuitive way of breaking down KPI segmentations is that of allowing the analyst to **manually define** them. These are two examples that demonstrate this case.

Example 1 (Temporary Reforms: Manual Projection on the Time dimension) *After applying some temporary economical reforms, we could specify that we expect a slight decrease in unemployment within a certain period. As a result, we would need to specify a 20% unemployment target for the first year, while for the second and third year we should have a 16% and 10% unemployment rate, even if our final target was 10%.* If we had not projected our expectations for all the years, we would have been failing during each year until we reached our target, even though this should not be the case.

Example 2 (Development Level: Manual Projection on the Space dimension) *In the same sense, one would expect that unemployment should be lower in more industrialized and developed countries than in developing ones. As a result we might have expected that the unemployment in a certain country should be lower than others over the years. This is the case for Europe 2020, where the Employment Rate target for each country is set individually.*

The second way for projecting segmentations is that of **automatically mining** the data warehouse. Given a database with values from previous years and a KPI. We can calculate, for every KPI-Restriction, the value of this KPI. Moreover, for each term of the KPI we can identify what is the market share of this term when compared to the "larger" parent market. This is an essential operation on data warehouses, where we calculate what is the contribution of a base-cell on a higher level aggregation. Since KPIs are complex functions, their values can not always additively calculate the higher level KPI. However, we are able to identify the contribution of each simple aggregation term of a KPI-Restriction to the general KPI. By feeding these ratios in the KPI calculation formula, we can identify the expected value for this KPI-Restriction. This process can be seen in Algorithm 1, where we iterate over all the KPIs defined in the strategic model. For each one of them, we identify every possible dimension-instance pair set and we calculate the value of the KPI for it. We then find all the parent market segments of this sub-dimension, and calculate the ratio of its value to the value of the KPI for the parent market segment. Then using an user defined function, we combine all these ratios in order to choose either the most meaningful or an aggregation of them.

An example for the employment KPI is the following. Starting with the aggregation of employment all over Europe, we calculate the employment for all countries, then for all regions and finally for the eurozone dimension. At each step we compare this KPI to the KPI of its parent market segments. For example, we compare Spain to Europe, Spain to Eurozone, Spain to Southern Europe. We then use the user defined function to choose the most appropriate ratio, or to aggregate them, thus adjusting the global target value to this sub-dimension. In the same way we compare Eurozone to Europe, each Region to Europe, and adjust the target values for the related sub-dimensions.

Example 3 (Temporary Reforms: Automatic Projection on the Time dimension) *As per our previous example, we could take a look at similar situations, e.g., previous Temporary Reforms, and mine the percentages over the different years with regards to the overall target value.*

Example 4 (Development Level: Automatic Projection on the Space dimension) *We could automatically calculate the average unemployment rates for the past years for all the countries in East EU, and compare each country's average to the total average. This ratio can then be used to adjust the expectations for each country on its own.*

Partial History Selection

Further on, in our scenario we only want to select parts of the data warehouse (from 2000 onwards) for performing the target value projections, instead of the complete historic knowledge. Partial history selection can become more complex, selecting isolated parts of the history. An example could be an economic crisis, where we want to get unemployment data only from periods where there was a financial crisis affecting some parts of Europe, and use them to adjust the target values for each sub-market. In order to deal with these situations, we can have a Knowledge Base that contains historic situations pointing to parts of the data warehouse that contain data related to them. Thus, we retrieve only the values from the relevant time intervals.

Summarizing, before proceeding to the monitoring step, we need to know how to fill the missing and target values of the KPIs, both for the aggregated data and for each sub-market. As we have shown, we can project segmentations using the previous data, either by:

- Using the current information available about the evolution of each KPI.

- Performing Partial History Selection for focusing on specific situations that affect this KPI that have occurred in the past, using a pre-annotated knowledge-base of past situations and types of situations.

```
Data: KPIs: $\mathcal{K}$, Predefined Target Value Projections: $P$, Data Warehouse: $DW$
Result: $\mathcal{K}_{proj}$
$\mathcal{K}_{proj} = \mathcal{K}$;
foreach $k \in \mathcal{K}$ do
    foreach Possible Sub-Dimensions $d$ of $k$ do
        $k_{restricted} = SelectSubDimension(k, d)$;
        if hasPredefinedTargetValues($k_{restricted}$, $P$) then
            $k_{restricted}$.targetValues = getTargetValues(k,d,P);
        else
            restrVal = calculateKPI($k$, d, $DW$);
            $p_{all}$ = findParentMarketDimensions($k_{restricted}$);
            parentKPIValueRatios = [];
            i = 0;
            foreach $p \in p_{all}$ do
                parVal = calculateKPI(k, p, $DW$);
                parentKPIValueRatios[i++] = restrVal / parVal ;
            end
            $k_{restricted}$.targetValues = calculateTargetValues(parentKPIValueRatios);
        end
        append($k_{restricted}$, $\mathcal{K}_{proj}$);
    end
end
```

Algorithm 1: Performance Diagnosis Setup

3.2.2 Expectation Diagnostics

In the previous sections we have focused on preparing the necessary data to answer **Performance Diagnostics** questions such as "Will we meet our goal for Employment Rate by year 2020?". However, as introduced in Section 2, we are also interested in being able to identify if there are hidden anomalies such as countries deviating from their usual behavior. For example, it is relevant to know if Spain or Greece are deviating from the employment behavior of other countries and when they started deviating. These questions fall into the **Expectation Diagnostics** category.

In order to support **Expectation Diagnostics**, we need to perform clustering at the base level of our data warehouse, such that we group our data and introduce hidden dimensions that can be used for aggregating their members, and producing trends that describe them. These aggregated trends, can be used to monitor for parts of the warehouse that are deviating from their previous clusters.

Expectation Diagnostics is the process that allows an automated monitoring system to compare current KPI trends to expected KPI trends. Expectation differs from Performance in the sense that it tries to capture differences among different layers of aggregation and different parts of the data warehouse, where the same trends should traditionally be observed. An example is that sub-markets should most of the time follow the same trends as their parent market segments. For example, unemployment in Spain could be expected to follow the trend of unemployment of Europe. If this is not the case, then Spain constitutes a special case that is worth being reported to an analyst. At the same time, this might not always be the case, as there might be sub-clusters within a certain hierarchy level, that do not necessarily follow the same trends [20]. For our scenario, we can see the trends and clusters of Spain and Greece compared with the trend of EU in Figure 6.

In order to be able to capture such insights, we need to support data series similarity queries for performing the following actions.

1. Clustering data series from different parts of the data warehouse, thus producing more meaningful levels of aggregation, where the aggregations follow the same trends as their components. An example would be the creation of clusters of countries that follow the

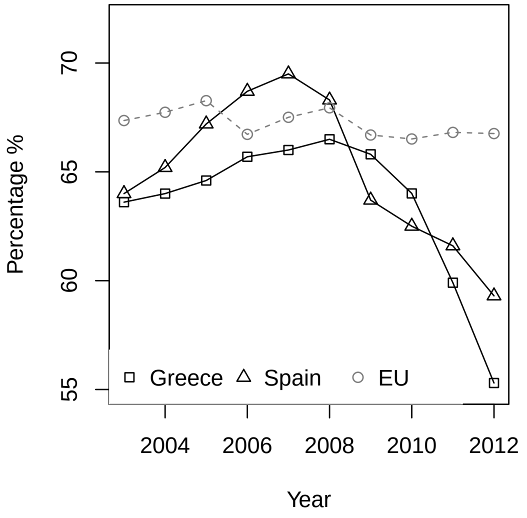

Figure 6: Cluster of Spain and Greece, over EU employment percentage

same unemployment trends, when this clustering is not provided by any of the current dimensions (e.g. Region, Eurozone, etc.), but rather from a hidden dimension that could be introduced by a clustering algorithm.

2. Comparing data series with their parent market segments, either computed via clustering, or physically located in the Dimensions Schema, in order to identify unexpected trends.

Consequently, similarity queries have to be used both while monitoring for unexpected deviations, and while periodically trying to find or update existing clusters in the data warehouse. As a result, such systems should be able to efficiently answer similarity queries on large collections of data series defined as follows.

Definition 2 (KPI Data-Series Similarity Query) *Given a two KPI Data Series* $ds_1 = DS^{d_1}_{k_1}(p_{1_s}, p_{1_e}, p_{1_{in}})$ *and* $ds_2 = DS^{d_2}_{k_2}(p_{2_s}, p_{2_e}, p_{2_{in}})$ *of equal size, for two KPIs* k_1 *and* k_2*, we are interested in finding out the distance between* ds_1 *and* ds_2*, by using a distance function* δ *that returns a real number representing this distance.*

$$series\_similarity(ds_1, ds_2, \delta)\ returns\ x \in \mathbb{R}$$

The problem of answering similarity queries in databases of data series was first introduced by [1] in 1993. A common approach to handle such queries is by reducing the dimensionality of the data using a dimensionality reduction technique [7, 12] and then building a specialized index structure [2, 18, 6]. Some example distance functions that can be used on top of such indices are the Euclidean Distance, Dynamic Time Warping [17] and others.

After the setup step, we should end up with a set of KPIs and their corresponding target value projections for each different sub-part of each KPI. Subsequently these new Sub-KPIs will be used for Performance Diagnosis both at a high level as well as at a lower level, thus allowing the analyst to adjust a strategy to all the important parts of the data that demonstrate a bad performance. Moreover, for Expectation Analysis we should end up with a new set of dimensions inserted in the Dimensions Schema, such that we are able to use them as if they were preexisting in the data warehouse.

3.3 Monitoring Step

The last step in the process is the monitoring step. The monitoring step is responsible for monitoring all the KPIs related to each goal, aggregate their statuses, and provide insights for each one of them. The first step of the process is the one related to **Performance Diagnosis**. Here, the system should be able to identify whether a goal is failing or not. In our scenario, we need to know if we are currently meeting or not our target for Employment and Education.

- If the overall goal is failing, or about to fail, we need to drill down in our data warehouse, by restricting the KPIs to various dimensions, in search of the sub-markets responsible for the overall failure.
- If the overall goal is succeeding, we need to point out to the analyst the parts of the Warehouse with the greatest success, that are probably responsible for this good status, as well as the ones that can still be improved.

A baseline algorithm would start with the general, unrestricted KPIs, and gradually produce restricted KPIs for all dimension-instance pairs. Subsequently, by calculating their values it should

```
Data: Goals: $G$, Data Warehouse: $DW$, Time Dimension: $t$, forecast step: f, forecast
      prediction model: p
Result: statuses
statuses = [];
foreach $g \in \mathcal{G}$ do
    foreach Possible Sub-dimensions d, except time do
        goalKPIStatuses = [];
        foreach $k \in getKPIs(g)$ do
            $k_{restricted} = SelectSubDimension(k, d)$;
            $k_{data\_series} = DataSeriesGrouping(k, t)$;
            ds = ComputeDataSeries($k_{data\_series}$);
            futureVal = forecast(ds, p, f);
            goalKPIStatuses.add(getKPIStatus($k_{restricted}$, futureVal));
        end
        statuses.add(aggregateKPIStatuses(kpiStatuses, g));
    end
end
```

Algorithm 2: Performance Diagnosis Monitoring

be able to produce analytics for each sub-market. Moreover, by performing Data-Series Grouping operations for each one of the restricted and unrestricted KPIs, forecasting algorithms can be used for performing this kind of analysis ahead of time. The output of the algorithm should be an overall status for the general goal and various insights for the status of this goal for various sub-markets. This algorithm can be seen in Algorithm 2, where we start by iterating over all the goals, for each goal we iterate over all the related KPIs and calculate every possible dimension restriction in all dimension-instance pairs. For each one of them we calculate the KPI Data Series on the time dimension, until the current time point. We then try to forecast the value for a future time f, using a prediction model p, both of which we have as input from the analyst. This future value (if f is set to 0, corresponds to now), is translated to a status. By aggregating all these statuses we can calculate the overall status of each KPI-restriction, as well as of the general KPI.

Obviously, the search space can explode really fast, as the number of combinations of dimensions and instances is very large. To overcome this problem various pruning techniques can be used, such as stopping to drill in when a sub-market of the original KPI has been marked as failed. For example, when we identify that East Europe is failing, we could either choose to drill in to the Eurozone dimension, or report this to the analyst and only drill in, on demand.

The last monitoring step is that of Expectation Diagnosis, where we are interested in finding market segments that are outlying with regards to their parent market segments. This process can be done in a top-down way, as described in [15], where we start at the highest level of aggregation and only explore the dimensions that are part of the most dissimilar descendants of this aggregation. This step makes use of the KPI Data Series and clusters previously calculated, and helps the user to find out diverging trends within the data as will be shown in the following section.

3.4 Result Analysis

In this section we present the results of following our process for analyzing Employment and Education with regards to Europe 2020 goals. As we can see in Figure 7, the **Performance Diagnosis** suggests that we are not expected to meet our goal in Employment if we follow the current trend. However, if analyzed in detail, we can see in Figure 7 that some countries present more significant differences with their targets than Europe overall. Malta for example, has already succeeded, while Germany is about to succeed, and Greece will probably fail to do so. In order to discover such insights, Algorithm 1 is run as a setup step to identify the target values for EU and for all the other sub-dimensions in the Data Warehouse. In this case, EU has defined specific targets for each country, and these

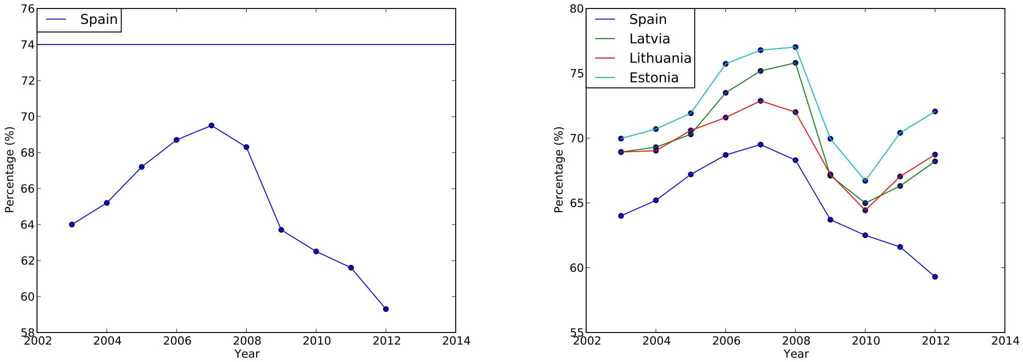

Figure 8: Spain failing to succeed as well as it deviates from its previous cluster in 2012

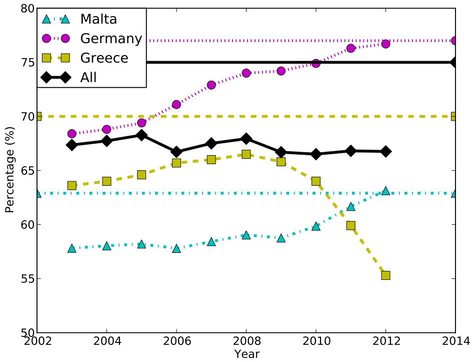

Figure 7: EU failing to meet its target, but Malta succeeds, Germany is about to succeed, and Greece will probably fail.

targets are monitored in a top down fashion by Algorithm 2.

By means of **Expectation Diagnostics**, we can find out what countries started to diverge from their traditional behavior, and when this started to happen. An example can be seen in Figure 8, where we can see that Spain fails to succeed to its target, something that constitutes a Performance Diagnostic. Moreover, while traditionally correlated with the Baltic countries, it starts to deviate and this is an Expectation Diagnostic, as the last years' course is not expected considering previous clusterings.

Finally, with regards to Education, Spain had been deviating from its target until the beginning of the economic crisis, when, unexpectedly, it changed its trend.

As we can see, with our process, we can obtain **Performance and Expectation Diagnostics** that provide important information in order to analyze in-depth the performance of the business. Furthermore, these results can be reflected into the strategic model in order to analyze their impact. It is worth noting however, that finding the reasons for latent anomalies requires additional considerations and is out of the scope of this paper.

4 Related Work

In this section we summarize the related work in the area of business monitoring. The most common asset used for monitoring tasks by enterprises until now has been the Balanced Scorecard [11]. A scorecard integrates several high level KPIs in order to provide an overall view of the performance of the business. However, a scorecard only provides an aggregated view of the data and does not consider the relationships between indicators. In an attempt to provide deeper analysis, businesses also include several Dashboards [8] that provide more detail about each indicator. Despite this effort, the potential number of different sub-market aggregations make it next to impossible to ensure that no anomaly goes unnoticed.

In terms of business modeling, Strategy maps [10] provide an overall view of the strategy of the enterprise. However, they are not (i) completely formal and (ii) they do not provide an integrated view of the status of the business. Similarly, in other areas, such as Software Engineering, proposals as [19] have been defined in order to specify monitoring conditions over requirements. Yet, in most cases the monitoring task is still left to the user. In [5] the authors propose the Goal-Question-Metric approach in order to monitor software development. However, when the approach is applied to business environments, the result obtained is a set of KPIs for monitoring the business performance that suffer from the drawbacks presented in the introduction. Finally, In [14] the authors propose the Willow architecture for system survivability. It aims at making systems avoid, eliminate and tolerate faults. It is able to monitor fault sequences, their inter-dependencies, as well as fault hierarchies. Each fault sequence is modeled by a finite state machine (FSM) which is triggered by system events. When certain FSMs reach a fault state, action is taken. Yet, although such systems work well for system survivability, their purpose is to identify events and perform actions accordingly, rather than monitor trends towards the fulfillment of a set of goals as well as their statuses.

Considering the fact that data warehouses commonly contain a large number of dimensions, identifying significant events as well as explaining them in the context of a Strategic Model, can be a daunting task for the analyst to be done in a manual way. As a result, automated tools that allow for the evaluation of continuous queries in regards to a Strategic Model have to be developed. One of the most significant challenges though, is that of converting the sets of formal goals specifications into well defined queries that can be continuously evaluated on top of a data warehouse.

5 Conclusions

Monitoring the business requires an in-depth analysis of the Key Performance Indicators (KPIs), as otherwise, problems within the sub-markets will go unnoticed until they threaten the global target. However, given the complexity of the search space it can become a daunting task if performed manually. In this paper we have presented a semi-automatic approach to tackle this problem by (i) modeling the business strategy and deciding on what KPIs should be monitored, (ii) describing a process to analyze sub-markets and evaluate their performance, and (iii) specifying a set of algorithms to perform this process. Furthermore, we have tested our approach by means of a real case study on the Europe 2020 framework, publicly available. The great benefits of our proposal are that we can identify anomalies, relationships, and deviations in the data that are not reflected at an aggregated level. Therefore, we are able to diagnose the existence of anomalies before they become threats for the business.

Our future work is focused on identifying the potential causes and solutions of a diagnosed problem. Additionally, we plan to analyze the impact of data quality in the process and testing scalability of the approach with larger data samples. Finally, an interesting research path is that of identifying hidden relationships in the strategic model being used.

Acknowledgements

This research was partially funded by the FP7 EU ERC Advanced Investigator project Lucretius (grant agreement no. 267856), and developed in the context of the projects MESOLAP (TIN2010-14860) and GEODAS-BI (TIN2012-37493-C03-03) funded by the Ministry of Education and the Ministry Economy and Competitiveness (MINECO) of Spain. Alejandro Maté is funded by the Generalitat Valenciana (ACIF/2010/298).

About the Authors

Konstantinos Zoumpatianos is a PhD student at the dbTrento group, University of Trento, Italy. His research involves data warehouses, business intelligence and data series management. He holds a MSc Degree in Information Management and a BSc degree in Information and Communication Systems Engineering, both from the University of the Aegean, Greece.

Alejandro Maté is a PhD student at the University of Alicante, Spain. He received a BS and a MSc Degree in Computer Science from the University of Alicante. He has published several papers in international conferences such as CAiSE, ER, and RE. His research involves data warehouses, model driven development, and requirements engineering.

Themis Palpanas is a professor of computer science at the University of Trento, Italy. He received a BS degree from the National Technical University of Athens, Greece, and the MSc and PhD degrees from the University of Toronto, Canada. His research solutions have been implemented in world leading data management products and he is the author of five US patents. He is the recipient of three Best Paper awards (PERCOM 2012, ICDE 2010, ADAPTIVE 2009) and founding member of the Event Processing Technical Society.

Juan Trujillo is a professor at the Computer Science School at the University of Alicante, Spain. Trujillo received a PhD in Computer Science from the University of Alicante in 2001. His research interests include database modeling, conceptual design of data warehouses, and data warehouse quality and security. He has published over 40 papers in national and international conferences and journals such as ER, ADBIS, JDMS, and DSS journal. He is a Program Committee member of several conferences such as ER, DOLAP, and SCI.

John Mylopoulos holds a distinguished professor position (chiara fama) at the University of Trento, and a professor emeritus position at the University of Toronto. He earned a PhD degree from Princeton University in 1970 and joined the Department of Computer Science at the University of Toronto that year. His research interests include conceptual modeling, requirements engineering, data semantics and knowledge management. Mylopoulos is a fellow of the Association for the Advancement of Artificial Intelligence (AAAI) and the Royal Society of Canada (Academy of Sciences). He has served as general chair of international conferences in Artificial Intelligence, Databases and Software Engineering.

References

[1] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. *Foundations of Data Organization and Algorithms*, pages 69–84, 1993.

[2] Ira Assent, Ralph Krieger, and Farzad Afschari. The TS-tree: efficient time series search and retrieval. *EDBT*, pages 252–263, 2008.

[3] Daniele Barone, Thodoros Topaloglou, and John Mylopoulos. Business intelligence modeling in action: a hospital case study. In *Advanced Information Systems Engineering*, pages 502–517. Springer, 2012.

[4] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*, volume 734. Wiley, 2011.

[5] Victor R Basili1 Gianluigi Caldiera and H Dieter Rombach. The goal question metric approach. *Encyclopedia of software engineering*, 2(1994):528–532, 1994.

[6] Alessandro Camerra, T Palpanas, J Shieh, and Eamonn Keogh. iSAX 2.0: Indexing and Mining One Billion Time Series. *ICDM*, pages 58–67, 2010.

[7] KP Chan and AWC Fu. Efficient time series matching by wavelets. *Data Engineering, 1999. Proceedings*, pages 126–133, 1999.

[8] Wayne W Eckerson. *Performance dashboards: measuring, monitoring, and managing your business*. Wiley, 2010.

[9] Jennifer Horkoff, Daniele Barone, Lei Jiang, Eric Yu, Daniel Amyot, Alex Borgida, and John Mylopoulos. Strategic business modeling: representation and reasoning. *Software & Systems Modeling*, pages 1–27, 2012.

[10] Robert S. Kaplan and David P. Norton. *Strategy maps: Converting intangible assets into tangible outcomes*. Harvard Business Press, 2004.

[11] Robert S. Kaplan, David P. Norton, RC Dorf, and M Raitanen. *The balanced scorecard: translating strategy into action*, volume 4. Harvard Business school press Boston, 1996.

[12] Eamonn Keogh, Kaushik Chakrabarti, and Michael Pazzani. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, August 2000.

[13] Ralph Kimball and Margy Ross. *The data warehouse toolkit: the complete guide to dimensional modeling*. Wiley, 2011.

[14] John Knight, Dennis Heimbigner, Alexander L. Wolf, Er L. Wolf, Antonio Carzaniga, Antonio Carzaniga, Jonathan Hill, Jonathan Hill, Premkumar Devanbu, Premkumar Devanbu, Michael Gertz, and Michael Gertz. The willow architecture: Comprehensive survivability for large-scale distributed applications. In *Distributed Applications., Intrusion Tolerance Workshop, Dependable Systems and Networks (DSN 2002), Washington DC*, 2001.

[15] Xiaolei Li and Jiawei Han. Mining approximate top-k subspace anomalies in multi-dimensional time-series data. In *VLDB*, pages 447–458, 2007.

[16] David Parmenter. *Key performance indicators (KPI): developing, implementing, and using winning KPIs*. Wiley, 2010.

[17] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[18] Jin Shieh and Eamonn Keogh. iSAX: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, 2009.

[19] Vítor E Silva Souza, Alexei Lapouchnian, William N Robinson, and John Mylopoulos. Awareness requirements for adaptive systems. In *Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems*, pages 60–69. ACM, 2011.

[20] Konstantinos Zoumpatianos, Themis Palpanas, and John Mylopoulos. Strategic management for real-time business intelligence. In *International workshop on business intelligence for the real, time enterprise (BIRTE)*, 2012.