Supporting Strategic Decision Making on Service Evolution Context Using Business Intelligence

Ernando Silva, Karin Becker, Renata Galante
Instituto de Informática – Universidade Federal do Rio Grande do Sul
Porto Alegre, RS - Brazil
{ernando.silva, karin.becker, galante}@ inf.ufrgs.br

***Abstract*—With the growing demand for service-oriented applications, the complexity of service change management is increasing. Existing work essentially addresses change decisions from a technical perspective (e.g. versioning, compatibility), but providers need to make decisions considering the business impact in terms of clients affected, revenues, costs and penalties. This paper suggests the use of Business Intelligence and Data Warehousing techniques to support business-oriented decisions throughout service life-cycle in a deep change context, i.e. a portfolio of services consumed in large scale by direct/indirect clients. The approach is centered on financial and usage indicators related to the service provision business, a data warehouse that provides a unified and integrated view of these indicators according to different analysis perspectives, and a data warehousing architecture that integrates heterogeneous data sources. We illustrate the impact analysis support provided by the approach through a case study inspired by a real world scenario.**

Keywords- service evolution, deep change management, business intelligence, data warehouse, change impact

I. INTRODUCTION

The demand for service-oriented applications has increased in recent years and it is expected to grow even more in the short-term. Today, many companies have business segments focused on providing solutions based on the Software as a Service (SaaS) paradigm. Typically, this magnitude of service provision covers a portfolio of interrelated services and, consequently, includes a large web of clients [1]. As traditional software, services are subject to continuous cycles of improvement, where changes are motivated by new requirements or business opportunities, new regulations, performance, etc. Changes can affect the structure (interface), semantics or non-functional properties (e.g. QoS) of a service [1][2].

However, the life-cycle of service-based applications is decoupled from the one of the services they consume [3]. Therefore, applying changes to services that are incompatible with current usage will break clients, with particularly severe consequences in a large-scale usage scenario. Disruptions are not limited to direct clients, i.e. clients that make requests to the changed service. Changes can also affect indirect clients, who do not consume explicitly the changed service, but instead, a service that depends (directly or indirectly) on the service changed, and is effected in a ripple effect. The changes limited to a single service and its direct clients are known as shallow; and the ones that cover a portfolio of services and its whole set of direct/indirect clients are referred to as deep [4]. A change-oriented service life-cycle, integrated with methods and tools, is necessary to provide a sound foundation for planning changes.

Most approaches address the technical aspects of service changes in shallow change scenarios, such as versioning, compatibility, and functional components for hosting and handling versioned services [1][2][3][5][6]. Papazoglou et al. [4] address deep changes with a methodological approach. Their change-oriented service life-cycle is composed by phases for identifying if a change is needed, analyzing change alternatives and deploying changes. A change information model [7] relates types of changes and stakeholders, and their effects on each other. Service governance involves the management, creation and enforcement of policies and standards for all the processes related to service life-cycle management, highlighting the areas for which policies and standards should exist for supporting decision making [8]. These works highlight tasks and decision points involved in the evolution life-cycle, but lack support for the underlying decision making process related to several stakeholders (e.g. designers, providers).

In our previous work [9], we correlated service lifecycle phases, tasks and stakeholders as a foundation for identifying evolution decision support requirements. Our focus is on the provider's perspective for decision-making. Beyond the technical scope, deciding about changes requires estimating the impacts of change decisions and their alignment with business strategies. For providers, knowing about incompatibility issues between specific service versions is not enough: they need to understand the effects of incompatibility on their web of clients, and how their profit, reputation and market position will be affected. Although a more accurate impact analysis minimizes the difficulties on decision making, finding these answers considering a deep change context, is a non trivial challenge.

Business Intelligence (BI) has been applied to support decision making in several fields [10][11], but its potential for supporting service evolution decisions has not been addressed by existing works. BI refers to the use of internal and external organizational information assets to make better business decisions, supporting the transformation of data into information. A common approach is to provide analytical resources over a centralized, integrated, historical, and subject-oriented database referred to as Data Warehouse

(DW) [11]. Challenges towards the use of BI for supporting service life-cycle decisions include: a) characterizing the decisional needs in this scenario; b) identifying relevant data about service provision and service consumption (and their clients) that can meet these needs and organizing them in a proper multi-dimensional model that supports an integrated analysis of all perspectives; and c) designing a flexible architecture capable of dealing with the unpredictable heterogeneity of data sources, which may vary according the environment of the service provider and its business practices.

In this paper we explore how BI can support the measurement of the impact of changes during the service life-cycle, using financial and usage indicators. We propose a BI approach that encompasses: (i) the identification of metrics that measure change impact, (ii) how these metrics can be integrated using a multi-dimensional DW model that enable several analysis perspectives of service provision, and (iii) a data warehousing architecture that deals with the lack of standards and the heterogeneity of data sources in the service provisioning context. The proposed approach represents a novel contribution to enhance business decision making in the context of service evolution. We complement previous work [9] by detailing indicators to measure change impact, discussing their integration according to distinct analysis dimensions, and presenting a data warehousing architecture.

The rest of this paper is organized as follows. Section II explores the decisional needs of service providers and how KPIs can be used for decision support based on the impacts of changes. Section III specifies the BI proposal, detailing how indicators are stored in a data warehouse and the architecture. Section IV illustrates impact analysis through a case study. Section V summarizes related work, whereas Section VI presents conclusions and outlines future work.

II. Decisional Needs and Impact Indicators

Papazoglou et al. describe in [4] a methodology that defines a full cycle for service deep changes. It identifies the main decision points and the related tasks for evaluating or applying the changes. The initial phase, "*Need to evolve*", encompasses identifying the need for changes, their scope and extent, and collecting KPIs. The next phase, "*Analyze the impact of changes*", covers the change impact analysis, compliance with business rules, recognizing problems generated by the scope of changes, costs estimative and KPI (Key Performance Indicator) analysis. These analyses result in the decision of applying or not the change. When the decision to implement change occurs, the final phase, "*Align, refine, and define*", comprises services testing, as well as monitoring the alignment of changed service with business strategy. Despite the overall guidance provided by this lifecycle, several needs can lead to decisions about changes. Different stakeholders are involved the tasks of each phase, with different concerns for decisions, as highlighted in [9]. For instance, designers face decisions regarding whether requirements are met by changes in service design or implementation. Providers, on the other hand, are more concerned about business-oriented effects resulting from the changes. In this paper, we assume the providers' perspective, and the business impact of chances as the driver for decisions related to service changes.

A. *The Impact of Service Changes*

From a business perspective, changes can represent a competitive advantage, an alignment with regard to competitors, or financial adequacy based on profits/losses obtained with service provision. Usage levels tuning may also motivate changes. In fact, change motivations may have any bias, but change impact remains a critical and central decisional point for the service provider. Change effects can affect both direct and indirect clients, resulting in client attrition, financial losses or reputation damage. Because this criticality, it is important to understand decisional needs related to measuring impact, and related metrics that can support these impact analysis. An example illustrates this situation in a deep change scenario.

Example. Suppose a service provider that has a service portfolio with thousands of clients. The provider notices a decrease in business profit and wishes to analyze how this situation could be reversed. He considers two alternatives: to increase the service fees or to reduce provision costs by decommissioning older service versions. However, which alternative is the most appropriate, and more importantly, will any of them solve the perceived symptoms?

The provider has a financial need that requires adjustments in the service portfolio, but he lacks support for decision making. For example, is it possible to understand how profit is decreasing along the client (deep) chain? In the case of decommissioning versions, will cost reduction be affected by other financial variables, such as loss of revenue due to loss of broken clients, or SLAs (Service Level Agreements) penalties?

Therefore, the provider needs a decision support environment relating distinct analysis perspectives that represent the impact of changes for the business.

In this paper, we adopt two overall analysis perspectives, namely usage and financial. We present how financial and usage perspectives can be measured, combined and explored. However, it should be clear that the approach can be extended to other forms of impact measurement analysis (e.g. the service performance, geographic information, normative restrictions).

B. *Impact KPIs*

KPI analysis is central to understanding the need for changes and measuring its impacts, and it is one of the first tasks in the deep change life-cycle [4]. According to [12], KPIs are indicators used by organizations as a mix of performance measures, which cover both Key Result Indicators (KRIs) and simple Performance Indicators (PIs). We consider KPI as a measuring of organization's business performance and its results, and thus, each metric represents an aspect of overall organizational strategy. The analysis of these KPIs enable to answer questions related to decisional needs of a service provider, as in the scenario detailed on Section II.A. We propose an initial set of financial and usage KPIs, summarized in Table I. In the next section, we show

how to represent and integrate them in a DW.

From a financial perspective, typical metrics are related to revenues and costs of service provisioning, which need to be considered in the evolution context. It is often the case that, to avoid breaking clients due to incompatible changes, several versions of a same service are available. Metrics that consolidate infrastructure spending or make accounts about penalties caused by SLA disruptions (e.g. service unavailability) can be weighted on the decisions about providing a specific service or version, thus influencing on the decision of creating, maintaining or decommissioning versions. Additionally, with measures representing service revenue, the provider has interesting information to derive the profitability of each service/version.

Considering the service usage perspective, one can measure the amount of requests to services, specific versions of a service, or even specific operations of a service/version [5]. We can further distinguish between direct and indirect requests, in order to cope with both shallow and deep scenarios. These KPIs are valuable because they indicate how much a service/version/operation is used, influencing a possible decision to create new service/version/operation, or decommission existing ones.

TABLE I. KPIS CONSIDERING FINANCIAL AND USAGE PERSPECTIVES

Perspective	Direct KPI	Derived KPI
Financial	• Revenue • Estimate Infrastructure Spending • Penalties Costs	• Profit
Usage	• Quantity of Direct Requests • Quantity of Indirect Requests • Number of Client Applications	• Total Requests

III. A DATA WAREHOUSING ENVIRONMENT FOR ASSESSING CHANGE IMPACT

BI is a means to empower providers with insight about the impact of changes from a business perspective. Data warehousing [10][11] is a common foundation for BI, which is centered in providing analytical resources over a centralized, integrated, historical, and subject-oriented data warehouse. A complex process to Extract, Transform and Load (ETL) is required to access different internal and external data sources, and consolidate all this raw data in the DW.

In this section we detail the DW model and the ETL architecture for the described service evolution context. On the one hand, the design of a DW involves considering the available, correct and useful granularity of the data to compose useful indicators for business oriented analysis, as well as their integration to provide a unified view of the service provision business. On the other hand, the ETL process of the Data Warehousing architecture must deal with lack of standards and well-defined processes in the domain, distribution of data sources, and their heterogeneity. We assume that the DW can be explored using traditional BI analytical tools (e.g. pivot tables, dashboards, alerts) [10], adapted according to the decision-maker profile.

The ultimate goal is to lay foundations for an insightful environment supporting the service provider with useful analysis for assessing the impact of change alternatives in terms of business variables. The discussion in this paper is limited to the usage and financial perspectives, using the indicators defined in the previous section, but it should be clear the approach can be extended to other indicators and perspectives.

A. Data Granularity

The modeling of an integrated and unified view of business metrics about service provision and its environment requires identifying: a) the smallest granularity of relevant data as it exists, and b) the trade-offs involved in representing it in the appropriate detailed/aggregated level, according to the decisional needs. Next, we discuss the characteristics of data in the service context and the implications for DW modeling.

1) Quantity of Direct and Indirect Requests

A common scenario for providers is the existence of multiple active versions of a same service being used by different clients. So, usage information represents the interactions between clients and operation of a specific service version. Although very relevant for analysis purposes, collecting this data is challenging due to the distributed nature of services. Techniques to collect service usage data by monitoring, intercepting and logging of client requests are discussed in [13]. Each alternative imposes trade-offs in terms of scope of extractable data (ranging from service version to specific service operations), cost and performance of the monitoring capabilities, which must be carefully considered [5].

The choice about the approach to monitor interactions influences the level of detail of available information, and determines the metrics that can be represented and their usefulness, particularly when combining metrics for decision making. For instance, collecting usage data at service version level prevents one to have a correct perception about the key operations from the client perspective, as well as to (correctly) derive metrics about indirect requests. At most, one can assume an inaccurate worst-case scenario where a service can trigger requests in another one (e.g. by examining the services coordination model). On the other hand, capturing and representing information at service operation level enables one to understand exactly which operations clients tend to use most. This fine-grained data can derive useful information that can be used for various purposes, such as envisaging service design alternatives, deriving deep change impact measures, usage-oriented compatibility, and so forth [5]. In this paper, we consider the technique described in [9] to collect direct/indirect requests at service operation level, although there are other approaches (e.g. collecting direct requests and estimating indirect requests from a BPEL definition [14]).

Usage information analysis can thus be as detailed as per service operation, version and client, such that it represents the quantity of direct/indirect requests of some client in a specific time for a service operation belonging to some service version. This data can be aggregated in different ways, such as per service/version and per group of clients.

2) Revenue

Revenue is how much the provider charges clients according to respective consumption of service. The charging type varies according to different factors and this diversity causes difficulties to analyze in a unified and integrated manner. For example, services providing some cloud storage functionality may have at least two factors to determine the charging type: the period of usage (e.g. by month) and the amount of data stored. Services that focus on financial transactions outsourcing may have a charging type based on percentage of the value of the transactions. For services that offer cloud databases, the charging type may be composed by three factors: consumption per hours/month, storage space according specific plans, and amount of data transferred by month. Other services may be charged considering essentially the consumption volume, as in services that provide resizable computing capacity.

Unlike usage data, the granularity of revenue metrics is at service version level, since it is quite unusual for providers to charge according to each specific operation requested from a service. Revenue can be detailed per service version, per charging type, and per client, and it can be aggregated per service and per group of clients.

3) Costs and Profitability

Costs metrics can be divided into infrastructure spending for service provision and penalties caused by SLAs disruptions. While SLA fees or compensation costs can be related to specific clients and respective usage contracts, provision costs are much more complex. They can be a composition of fixed (e.g. same costs for all clients) or variable costs (e.g. considering some differenced distribution between distinct clients), according service characteristics or organizational policies. Determining provision costs on a service portfolio may be an arduous and complex process, because it implies defining dependency matrixes and several coefficients that are quite difficult to establish. Frequently, simplifications are done on organizational costs structure (e.g. prorating equally between each group of client, or with balancing costs according volume of client requests), making the analytical process simpler. In general, possible discrepancies due to costs simplification are covered by profit margin. Therefore, mechanisms of cost prorating can be applied in order to provide integration of this type of cost with the previously discussed financial metrics. In addition, it is essential to be able to combine costs and revenue to derive service profitability measures.

Assuming a proper prorating function for provision costs, it is possible to detail metrics related to cost and profitability per service version, per charging type, and per client. The aggregation can occur per service and per group of clients.

B. Representing Data in a Dimensional Model

Data Warehouses are modeled in terms of Fact and Dimension tables. The former contain the measurements, and the later represents the analysis perspective over measurements. A Dimension organizes a hierarchy of attributes that represent the ability to detail or aggregate measurement data.

It is important to establish a unified and integrated view of data, yet being able to preserve differences that are useful for analysis. Table II summarizes the minimum granularity of the data and their analysis dimensions, as discussed in the previous section. As it can be seen in Table II, despite the existence of common dimensions, indicators have differences that need to be taken into account.

TABLE II. GRANULARITY AND SCOPE OF FINANCIAL AND USAGE METRICS.

Dimension	**Requests, Applications**		**Revenue, Costs, Profit**	
	Analysis Dimensions	***Grain***	***Analysis Dimensions***	***Grain***
Time	Yes	Month*	Yes	Month*
Client	Yes	Client	Yes	Client
Service	Yes	Operation	Yes	Version
Status	Yes	Status	Yes	Status
Charging Type	No	-	Yes	Charging Type

Thus, we modeled the DW using a multiple fact table schema (MFTS), which is commonly used to model a set of multiple, interrelated subjects [11]. A MFTS schema is composed of several fact tables, relatable through a set of conformed dimensions, i.e. dimensions that have the same meaning at every possible fact table. We propose two facts tables, *FINANCIAL_FACT* and *USAGE_FACT*, which group the respective KPIs (Table I) and relate them through conformed dimensions. In this way, we are able to combine financial and usage information through conformed dimensions, yet preserving the differences between them, such as the different granularity of usage and financial with regard to service/operation, or the additional charging type dimension for financial facts. The resulting DW model is depicted in Fig. 1.

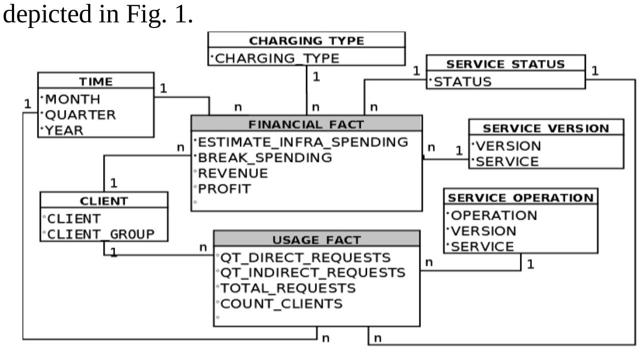

Figure 1. MFTS schema for the DW model.

The three common dimensions are *Time*, *Client* and *Service Status*. *Time* dimension indicates a time period that groups direct/indirect requests or the revenue/profit/losses obtained. We suggest month as the minimum granularity because generally the provider charges clients monthly, indicating that the data summarization by month is more appropriate. However, the analysis could be more detailed (e.g. two weeks, week), according to providers' needs. Notice that usage measurements could be related to a much smaller grain with regard to time (e.g. requests per hour, or day), but this is hardly the case for financial measurements. If that is the case, the same modeling alternative used for Service dimension could be used, as discussed below.

Client dimension is designed to characterize the origin of service requests or financial revenue/cost. Client dimension is described by a two level hierarchy, client and client group, the later considering the ability of grouping clients according to some similarity criteria. Several criteria could be used for grouping clients (e.g. strategic importance). In [3][9], we proposed a knowledge discovery process to group applications based on service usage patterns. Other dimensions could be used to characterize clients, such as geographic location, segment of business, which should simply be added as dimension to the fact tables.

Service Status dimension is used to indicate the stage in the provision life-cycle (e.g. active, deprecated, and decommissioned). This analysis perspective is related to both usage (e.g. how many clients are consuming a deprecated version) and finance. The later is especially interesting when the provider analyzes the profit of a deprecated service version or wishes to align an organizational strategy with results being obtained with a newly deployed service.

Another conformed dimension should be Service, but as discussed, usage and financial metrics have different minimum granularity for analysis, per operation and per version, respectively. In order to analyze the service usage, service dimension hierarchy is characterized as operation, version and service, whereas for financial metrics, operation detailing is a very unusual granularity. To consolidate distinct granularity information, each fact is related to a service dimension with specific granularity, *Service_Operation* and *Service_Version*, respectively. Thus, these dimensions are similar when usage facts are considered at aggregate level (per version or per service), at the same time preserving the ability of considering at operation level. As mentioned, this same alternative could be used if providers wish to maintain different *Time* granularity for usage and financial variables.

Finally, there are dimensions that are specific only to the financial context. *Charging Type* dimension is restricted to financial perspective because it is related with charging according the kind of service. This dimension allows the adequacy of model to the several charging methods, as previously discussed.

This model could be enriched with other fact tables, as additional impact indicators are considered, or dimensions, if more analysis details are required. In that case, the same considerations discussed here about data availability, granularity, and analysis detailing/aggregation apply.

C. ETL Architecture

The ETL architecture presented in Fig. 2 is responsible for extracting data from heterogeneous data sources, and loading it in the DW after the proper transformation.

1) Data Sources Area

A distinctive feature of this domain is the lack of standards and well established processes and practices. Large scale service provision is a relatively new business area and demands distinct operational management applications, which have their own repository with specific data models. Service providers can use several applications to store business and service data, as internal operational applications, CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning) systems, service metadata (e.g. the WSDL specification of services), SLAs, service usage log files, usage mining databases, etc. Also there is no standardization in the choice of applications, nor on the way providers store service data. Furthermore, some types of data may not even exist, and must be derived from raw data extracted from existent sources.

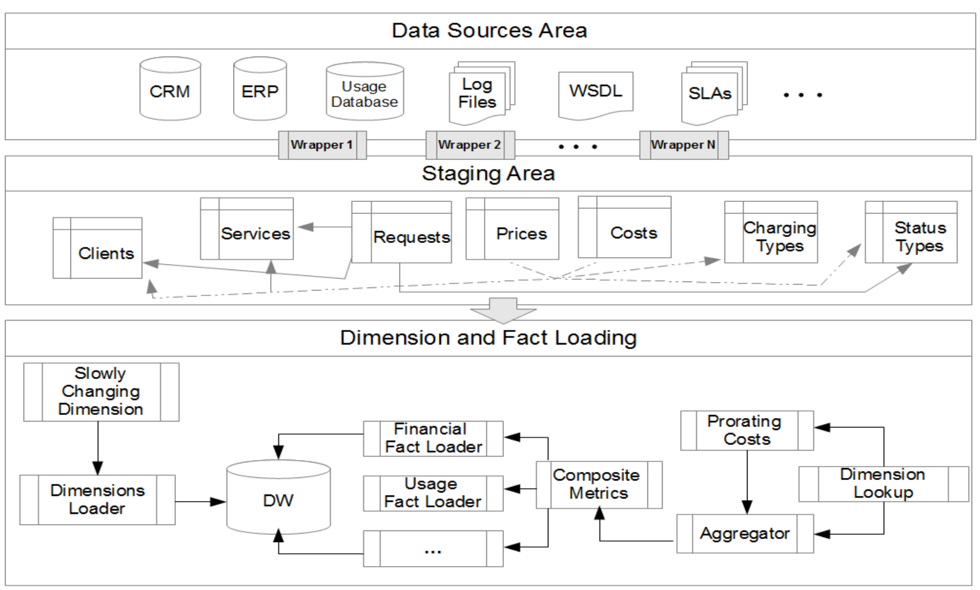

Figure 2. ETL architecture to populate the DW

2) *Staging Area and Extraction Wrappers*

The Staging Area aims at providing a normalized model in which the heterogeneous data from all data sources can be converted to. The approach proposed to deal with the different types of heterogeneity is to combine the data sources and the data staging layers using extraction wrappers, as in [15]. Wrappers are software artifacts that use a unique interface for encapsulating one or more applications. Wrappers will extract necessary data and transform them according to the normalized model established in the staging area. Wrappers may need to integrate raw data spread in different data sources in order to derive data for the normalized model.

For example, an organization may have part of the financial data in an ERP system, and the charging types and the factors that affect can be characterized in other types of systems. In this case, a wrapper should be implemented to extract data from various sources and store it into the staging area. All this information will then be consolidated in a single attribute that represents the revenue in a common unit, regardless the charging type.

With the wrapper-based approach, the process becomes more adaptable for adding of new and diversified sources, making the adoption of approach easier for distinct service provision environments.

3) *Dimension and Fact Loading*

The normalized model of the Staging Area enables to abstract from the heterogeneity and idiosyncrasies of the original, raw data sources. The Dimension and Fact Loading layer is responsible for transforming normalized data according to the multi-dimensional model concepts (dimensions and facts), and loading data in the DW.

The Dimension Loader performs the process of slowly changing dimensions. After handling data compliance and integrity checking, this component acts according strategies related to structural changes on dimensions. Although dimensions change infrequently over time [11], several alternatives may be adopted to represent these changes for the provider (e.g. overwrite old values or maintain historical values in specific fields), according the decisional needs. This component is interesting for the service domain because dimensions like *Service_Version/Service_Operation* and *Client* can be changed more frequently that the others. In first case, new service versions or even services can be regularly created and, in the later, new groups of clients can be detected or clients can be reclassified over time.

The Fact Loader has a set of components to transform measurable data and load it into the fact tables, namely:

- Lookup Function, used to match non-loaded fact data with existent dimension data in the DW, ensuring that only valid entries are inserted. This mechanism contributes to the consistency of the DW.
- Prorating Function. It is applied when prorating of costs is necessary to determine costs KPIs. For example, penalties can be calculated by client, but provision costs need to be prorated among a set of clients. This function can apply fixed prorating (homogeneous distributing of costs among clients), variable prorating according requests volume of clients, variable prorating according absolute number of clients, among others.
- Aggregator Function, which groups metrics according specific aggregation parameterizations and analysis levels. It is applied over raw data available at a more detailed grain than correspondent fact table. For example, direct/indirect requisitions must be grouped by month, lowest level of our time hierarchy, despite daily information may exist.
- Composite Metrics Function, which refers to the creation of derived metrics based on existing ones (e.g. profit, calculated based on difference between revenue and costs).

IV. ILLUSTRATION

To demonstrate how our proposal may be applied to the service evolution scenario, we present a hypothetical case study inspired by the AWS (Amazon Web Services [1]) portfolio, of which the relevant features are summarized in Table III. The portfolio includes services in different segments, of different complexities (as represented by number of operations), and distinct charging types. The dependencies between services allow us to assume a large web of direct and indirect clients for services in this portfolio. Also, the more operations a service offers (e.g. EC2), the more we can assume clients have several alternatives for using the service, implying distinctive usage patterns.

TABLE III. SELECTED SERVICES TO COMPOSE THE CASE STUDY BASED ON THE AWS PORTFOLIO.

Service	Segment	# Operations	Depends on	Charge by
FPS	Payment	25	Simple DB RDS	Financial transaction
S3	Storage	16	EC2	Storage by month
EC2	Computing	137		Consumption (by hour and by traffic)
Simple DB	Database	10	S3	Storage by month; Consumption (by hour)
RDS	Database	28	S3 EC2	Consumption; Data transfer; Storage plans

Considering this set of services, synthetic data was created to simulate aspects of this service portfolio, such as:

- Usage data at operation granularity, obtained from log files generated from the simulation of direct and indirect requests of clients. This data enables: (a) the measurement of the service usage level, by varying the set of operations used by each client; and (b) the grouping of the clients according to similar usage. We applied the service usage mining process proposed in [3][9], which clusters clients into groups according to similarity of operations requested (or their frequency).

[1] http://aws.amazon.com/

- Direct and indirect clients. Services can be used by client applications (direct clients), or by other services of the portfolio, which in turn have their own clients. Considering Table III, all services depend on EC2 either directly (S3 and RDS) or indirectly (FPS and SimpleDB). These interdependences enable to characterize the indirect clients of each service, which would be affected as a ripple effect.
- Frequent versioning of each service, as a result of some maintenance policy (e.g. monthly, as most AWS services). We assume dependent services and clients are not required to migrate to the newest version immediately, and therefore multiple concurrent versions of a same service exist, with their own clients and provision costs;
- Service revenue according service characteristics and distinct charging types, distributed by versions;
- Service costs, with an arbitrary estimative of spending related to provisioning service versions, and financial losses due violations of SLAs.

Considering this scenario, it is possible to explore how the BI approach supports some decisional needs of service provider, as detailed in Fig. 3.

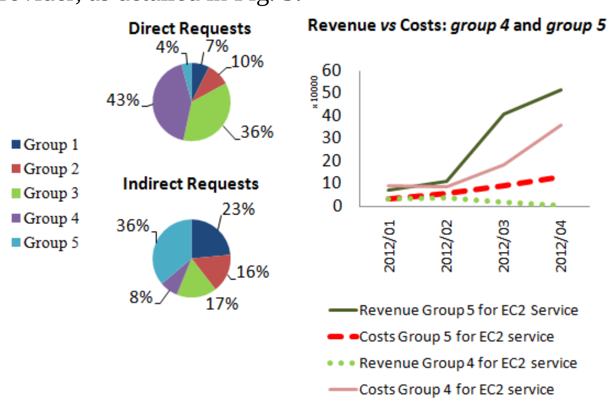

Figure 3. Dashboard that relates (i) EC2 direct/indirect requests by client group (left side) and (ii) the historical evolution of revenue and costs for a specific group (right side).

In this first illustration, the provider wishes to determine possible impacts resulting from changing the current version of the EC2 service. Since other services depend on the EC2 service, the provider must treat the spread of the impact over its indirect clients. As mentioned, we assume the use of typical analytical tools available in BI environments, such as pivot tables and dashboards. So, using the BI environment, the provider can develop the analysis in two fronts: (a) identifying client groups that make large amounts of direct and indirect requests, an indicative of the most (directly and indirectly) affected clients, as illustrated on pizzas chart in left side of Fig. 3; (b) detailing the evolution of revenue and costs associated that specific groups of clients over the last quarter, as illustrated by lines chart in right side of Fig. 3. With this consolidated perspective, the provider observes that clients in Group 5 make few direct requests to EC2 service, but consume it largely indirectly, whereas clients of Group 4 have the opposite behavior. When the provider details revenue and costs related to these two groups of clients over the last months, she discovers that the revenue related to Group 5 has grown considerably, whereas the cost of providing this service has remained stable. On the other hand, the revenue due to direct requests of Group 4 is much lower, and the costs are higher. So, it reveals that the deep indirect change impact can cause more negative financial consequences than the set of direct clients.

Another analytical possibility addresses the provider considering how to revert a situation of decreasing profit described in Section II.A. One of the considered solutions was decommissioning non-profitable service versions to reduce costs, a situation that can be analyzed using the proposed approach. Fig. 4 illustrates a pivot table resultant from a filtering of the 3 service versions with the least requests. The provider has an indicator that relates the quantity of requests and the service profit. The last column indicates profit or loss, using green or red arrows, respectively. The provider can also detail profit according charging types, which could reveal some pattern relating unprofitable services and some technical deficit (e.g. outages of storage mechanisms that cause breaking of SLAs). So the provider has more insight for decisions related to decommissioning versions unused or affecting negatively business financial health.

Service Version		Qt. Requests	Profit	
FPS	version 7	75,000	$ 31,000.00	▲
	version 14	88,000	$ 51,000.00	▲
	version 22	6,900	$ -12,000.00	▼
Simple DB	version 25	31,000	$ 41,000.00	▲
	version 26	8,600	$ -2,000.00	▼
	version 31	24,600	$ 58,000.00	▲
RDS	version 15	2,800	$ -128,000.00	▼
	version 27	6,500	$ -9,000.00	▼
	version 29	18,000	$ 2,000.00	▲
S3	version 10	41,700	$ 212,000.00	▲
	version 25	23,200	$ -183,000.00	▼
	version 27	27,700	$ -133,000.00	▼
EC2	version 9	700	$ -203,000.00	▼
	version 25	32,500	$ 37,000.00	▲
	version 27	4,300	$ -1,000.00	▼

Figure 4. Pivot table on report that lists the bottom 3 versions with least requests to each service, relating them to the quantity of requests, the profit and an indicator to demonstrate how profitable is the version.

V. RELATED WORK

Considering the diversity of proposals covering the service changes management, and targeting shallow changes or deep changes as introduced in [1], is possible notice that most works in the service evolution domain address the former. These approaches specially focus compatibility and versioning issues [1][2][3][5][6]. Despite the importance of this type of technical support for service designers to understand the effects of shallow changes and typically the worst-case impact scenario, it disregards the fact that services may be used differently. Usage oriented impact assessment is addressed in works such as [1][5], restricted to the context of shallow changes.

Few works are oriented towards deep change impacts, particularly according to a business perspective. In terms of deep impacts, a dependency model for quantifying the effect of changes considering a SOA ecosystem is presented in [14]. It assumes that boundaries, dependencies and components are previously known, but it does not specify how this data can be derived.

The change-oriented service life-cycle presented in [4] provides an important framework for the deep change scenario, but it needs to be refined for considering stakeholders, tools and models to support decision making and its features. In [9], we integrated this change-oriented methodology with a complementary set of tasks defined in [6], which relates change events, their relationship and stakeholders involved, as well as with service governance concepts [8]. SOA governance highlights the areas for which policies and standards should exist for supporting decision making, but there are not details about decision activities and stakeholders in the service evolution context. Our work is complementary to [4][6][8], by focusing on decision support. This integration supported the identification of decisional requirements related to service lifecycle decisions.

Also in [9], we proposed initial ideas towards a BI environment to support decision making in this context. The current work extends this previous work presenting a more mature model, and the data warehousing architecture.

VI. Conclusions and Future Work

In this paper we presented a BI proposal to support business-oriented decision making in service evolution management from the provider´s perspective. Our approach allows the assessment of change impacts according financial and usage indicators that provide insights about the consequences of changes in the business. A DW fact constellation schema was used to represent these measures and their respective analytical perspectives, allowing the representation of their common aspects, yet preserving the ability to analyze them according to specific dimensions. The data warehousing architecture proposed handles heterogeneous sources, aggregations and prorating of costs, which reflects the lack of standards and business practices typical of the current state of the practice. With a case study, we demonstrate how to combine distinct type of indicators to improve the support to the service evolution in a business perspective. The data is synthetic, but it presents properties that are typical of a real case scenario. The usage and financial indicators suggested provide important insight for typical service evolution dilemmas that providers face, but the approach goes beyond these specific indicators. Indeed, other classes of indicators relevant to business can be adopted, for which the tasks of analyzing data granularity, multidimensional modeling and insertion in the ETL structure must be developed, as discussed Section III. Addressing business decision needs, in addition to technical concerns, will grow in importance with the increasing investments on large-scale service-based applications.

As future work, we intend to explore analysis perspectives related to compatibility and versioning, deriving new metrics that translate technical aspects as impact indicators for the service provider. We also consider exploring multiversion DW capabilities to represent what-if analysis, allowing sensitivity analysis over the impact of changes according to given hypotheses, following new trends in BI [10]. We also want to evaluate the performance of our approach considering a big data scenario, applying our model in a real scenario of some large-scale service provider. The main challenge to do this validation is to get real data related to strategic needs of service providers.

Acknowledgment

This research is financially supported by FAPERGS and CNPq– Brazil.

References

[1] V. Andrikopoulos, S. Benbernou, and M. Papazoglou, “On the evolution of services,” Software Engineering, IEEE Transactions on, vol. 38, no. 3, May-June 2012, pp. 609-628.

[2] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, “End-to-end versioning support for web services,” Services Computing, 2008. SCC '08. IEEE International Conference on, 1, July 2008, pp. 59-66.

[3] K. Becker, A. Lopes, D. Milojicic, J. Pruyne, and S. Singhal, “Automatically determining compatibility of evolving services,” in Proc. IEEE International Conference on Web Services (ICWS’08), Sept. 2008, pp. 161-168.

[4] M. P. Papazoglou, V. Andrikopoulos, and S. Benbernou, “Managing evolving services,” IEEE Software, vol. 28, no. 3, May-June 2011, pp. 49-55, doi:10.1109/MS.2011.26.

[5] M. Yamashita, B. Vollino, K. Becker, and R. Galante, “Measuring change impact based on usage profiles,” Proc. IEEE 19th International Conference on Web Services (ICWS 2012), June 2012, pp. 226-233, doi:10.1109/ICWS.2012.35.D.

[6] Frank, L. Lam, L. Fong, R. Fang and M. Khangaonkar, “Using an interface proxy to host versioned web services,” Services Computing, 2008. SCC '08. IEEE International Conference on, vol. 2, July 2008, pp. 325-332, doi:10.1109/SCC.2008.84.M.

[7] Treiber, H. Truong, S. Dustdar, G. Feuerlicht, W. Lamersdorf, “On analyzing evolutionary changes of web services,” Lecture Notes in Computer Science, vol. 5472, 2009, Springer Berlin Heidelberg, pp. 284-297, doi:10.1007/978-3-642-01247-1_29.

[8] B. Woolf, “Introduction to SOA governance - Governance: The official IBM definition, and why you need it,” 2006. [Online]. Available: http://www.ibm.com/developerworks/library/ar-servgov/ [Accessed: February 03, 2013].

[9] E. Silva, B. Vollino, K. Becker, R. Galante, “A Business Intelligence approach to support decision making in service evolution management,” Proc. IEEE Ninth International Conference on Services Computing (SCC), 2012, June 2012, pp. 41-48.

[10] M. Golfarelli, S. Rizzi, I. Cella, “Beyond data warehousing: what's next in business intelligence?,” Proc. 7th ACM International Workshop on Data warehousing and OLAP, 2004, pp. 1-6.

[11] R. Kimball, M. Ross, The Data Warehouse Toolkit, 2nd ed., John Willey & Sons, 2002.

[12] D. Parmenter, Key performance indicators (KPI) - Developing, Implementing, and Using Winning KPIs, Hoboken: J. Wiley, 2010.

[13] A. Chuvakin, G. Peterson, “Logging in the age of web services,” IEEE Security & Privacy, vol. 7, no. 3, May-June 2009, pp. 82-85.

[14] S. Wang, M. Capretz, “A dependency impact analysis model for web services evolution,” Proc. IEEE 7th International Conference on Web Services (ICWS 2009), July 2009, pp. 359-365

[15] P. Silveira, K. Becker, D. Ruiz, “SPDW+: a seamless approach for capturing quality metrics in software development environments,” Software Quality Journal, vol. 18 (2), 2010, pp. 227-268.