IT service management driven by business objectives: an application to incident management

Claudio Bartolini, Mathias Sallé, David Trastour
HP Laboratories Palo Alto
HPL-2006-38
February 22, 2006*

service level incident management, Openview, service desk, service level management, decision support

In this paper we address the problem of ensuring business-IT alignment. We describe a method and a system for decision support in IT Service Management driven by alignment with the business objectives of the enterprise that the IT supports. Our technical proposition, called IT Management by Business Objectives (MBO) is applicable to most of the domains of IT Service Management, such as incident management, change management, and others. The technology consists of some components that are reusable across domains, together with guidelines and patterns for building complementary components in order to develop domain-specific solutions.

* Internal Accession Date Only
Published in and presented at the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), 3-7 April 2006, Vancouver, Canada
Approved for External Publication

IT service management driven by business objectives An application to incident management

Claudio Bartolini, Mathias Sallé
HP Laboratories
Palo Alto, USA
claudio.bartolini@hp.com, mathias.salle@hp.com

David Trastour
HP Laboratories
Bristol, UK
david.trastour@hp.com

Abstract — **In this paper we address the problem of ensuring business-IT alignment. We describe a method and a system for decision support in IT Service Management driven by alignment with the business objectives of the enterprise that the IT supports. Our technical proposition, called IT Management by Business Objectives (MBO) is applicable to most of the domains of IT Service Management, such as incident management, change management, and others. The technology consists of some components that are reusable across domains, together with guidelines and patterns for building complementary components in order to develop domain-specific solutions.**

Keywords Service Level Incident Management, Openview, Service Desk, Service Level Management, Decision Support

I. Introduction and Motivation

Whether running an in-house IT department or managing it on behalf of an outsourcing customer, IT managers are required to align IT service delivery to the business that it supports. This requires them not only to ensure smooth running of their IT operations, but also to be aware of the effect that alternative decisions may have on the business. Besides and beyond the definition of service level objectives, business-level indicators need to be taken into account that represent the concerns of various levels of management within the business, from line of business managers supported by the IT all the way up to the executives. Examples of such indicators are:

- Guarantee a given return on investment.
- Work with an upper bound on fund for IT investment
- Comply with Sarbanes-Oxley 404 regulations
- Increase customer satisfaction
- Improve voice-of-the-workforce scores

Control Objectives for Business Information-related Technology (COBIT) [1] and Balanced Scorecards [2] specify mechanisms to specify and monitor such objectives and provide metrics and key performance indicators to assess IT performance related to them. There are, however, few tools or technologies available to support IT managers as they struggle with the ever-rapidly changing needs of the business. Currently available tools for IT Service Management (as defined in ITSM [3]) help ensure that the IT organization uses standard methods and procedures for ensuring that smooth operation of the IT resources. However, none of the commercially available tools today help an IT manager plan and schedule their courses of action by taking into account the risk of effecting the actions and their impact on the business.

We are proposing an approach based on building decision support tools that suggest the most important things to do with respect to the impact on the business of actions taken on IT systems and processes. The decision support tools recommend the course of action to take based on the prospective alignment with business objectives that capture concerns spanning from smoothly running of daily IT operations all the way up to long-term strategic objectives at the executive level. Our tools can be used in a standalone way or can be embedded in more comprehensive solutions for IT Service Manager such as HP Openview Service Desk™ (OVSD) for example.

From the available literature on the matter [4] and from our conversations with IT managers, it appears very difficult to understand the business impact of course of actions because the various layers of the IT infrastructure (networking, servers, OS, storage, etc...) are owned and managed by different organizations, and there is no cross-organization view of the service model. Our approach aims at providing a comprehensive information model for defining business objectives and the key performance indicators that they are based upon. On the other hand, we aim at keeping the cost of modeling IT systems, resources, and processes at a minimum, while still providing meaningful decision support. We do this by identifying the metrics collected at IT level that really have impact at the business level.

Our claim is that the techniques we use are applicable to various sub-domains of IT service management. To exemplify the applicability of the approach we have built a complete solution for *incident management* driven by business objectives, which we describe in this paper.

The Information Technology Infrastructure Library (ITIL) [5] defines an *incident* as a deviation from the (expected) standard operation of a system or a service that causes, or may cause an interruption to (or a reduction in) the quality of the service. Most targets (or *service level objectives*, SLO) set in a *service level agreement* (SLA) are subject to direct financial penalties or indirect financial repercussions if not met. It is therefore critical for this management process to flag when service levels are projected to be violated in order for an IT organization to take proactive actions to address the issue. The objective of *incident management* is to provide continuity by restoring the service in the quickest way possible by whatever means necessary (temporary fixes or workarounds). Example of incidents may be degradation in the quality of the service according to some measure of quality of service; unavailability of a service; a hardware failure; the detection of a virus.

In the incident management process it is of fundamental importance to classify, prioritize and escalate incidents. Priority of an incident is usually calculated through evaluation of impact and urgency. However, these measures usually refer to the IT domain. The central claim of our work is that in order to achieve the strategic alignment between business and IT, the enterprise needs to drive incident prioritization from its business objectives. This starts from evaluating the impact that an incident has at the business level, and its urgency in terms of the cost to the business of not dealing with it in a timely fashion.

The paper is structured as follows. In section II we introduce our approach to providing decision support tools for IT service management driven by business objectives and in section III we exemplify its applicability by describing the design and implementation of a solution for incident management that uses MBO. In section IV we describe the integration effort of the MBO technology into the current version of the Service Level Management (OVSD-SLM) component of HP Openview Service Desk. In section V we touch on future work, and in section VI we discuss related work in the IT management literature, before moving on to the conclusions in section VII.

II. IT SERVICE MANAGEMENT DRIVEN BY BUSINESS OBJECTIVES (MBO)

Our current research on *IT Service Management driven by Business Objectives* (in short MBO) is aimed at developing solutions for IT management driven by an enterprise's business objectives. We approach this goal through the development of a methodology for decision support in IT management. The methodology is based on a reasoning engine (the MBO *alignment engine*, a.k.a. *Aline*) that solves the following class of decision problems: it computes the alignment to objectives that is expected for each of the possible given management options, or course of action aimed at managing the IT delivery systems. The alignment engine uses the measure of alignment thus derived as a value of the utility to the business of carrying which is used to rank the alternative courses of action. On ranking the options, Aline returns a suggestion on what course of action to take, substantiated by the evidence that it has for assessing the alignment with respect to the business objectives.

In the development and the deployment of the solutions, we follow the principle that the **cost of modeling should be kept low**; so that it is easily offset by the benefit obtained from the decision support.

For each of the various IT management domains the generic decision problem is specialized into a decision problem that pertains to that domain. This requires a mapping of the domain specific concept onto the generic concepts that are defined in the MBO information model.

1) MBO Information model

The MBO business objectives information model (Figure 1, top of next page) is articulated around a set of key concepts: *objectives*, *key performance indicators* (KPI), and *perspectives*. The terminology used in this information model borrows where possible from the lexicon of the COBIT [1] (Common Objectives for Information and related Technology) framework and from balanced scorecard [2]. COBIT is a framework addressing the management's need for control and measurability of IT. It provides a set of tools and guidelines to assess and measure the enterprise's IT capability for the principal IT processes. Balance scorecard is a tool for management that enables organizations to clarify their vision and strategy by capturing them into actionable objectives

In the remainder of this section, we briefly describe the principal concepts defined in the MBO Information Model.

OBJECTIVES

COBIT introduces key goal indicators (KGI) as measurable indicators of the business objectives. In our model the *objectives* are the corresponding concept to COBIT's KGIs. They are represented by expressing one or more target values[1] over a key performance indicator, or KPI – see below.

KEY PERFORMANCE INDICATORS

As defined by COBIT, *key performance indicators* (KPI) are measurable indicators of performance of the enabling factors of IT processes, indicating how well the process enables the goal to be reached.

PERSPECTIVES

Perspectives are used to bundle objectives together that concern a certain angle of the business. The concept of perspectives is borrowed from the balanced scorecard [3]. A

[1] In all the examples given in the remainder of this paper, only one target region per objective is defined, in order to help the flow of the discussion. An example of objective with multiple targets is one defining a first threshold of acceptability and a further threshold that represents a *stretch goal*. Example: revenues for the quarter must increase 15% year over year, with a stretch goal of 20% increase. When multiple targets are defined for an objective, the measure of alignment with the objective needs the definition of *importance weights* for all the target regions.

.

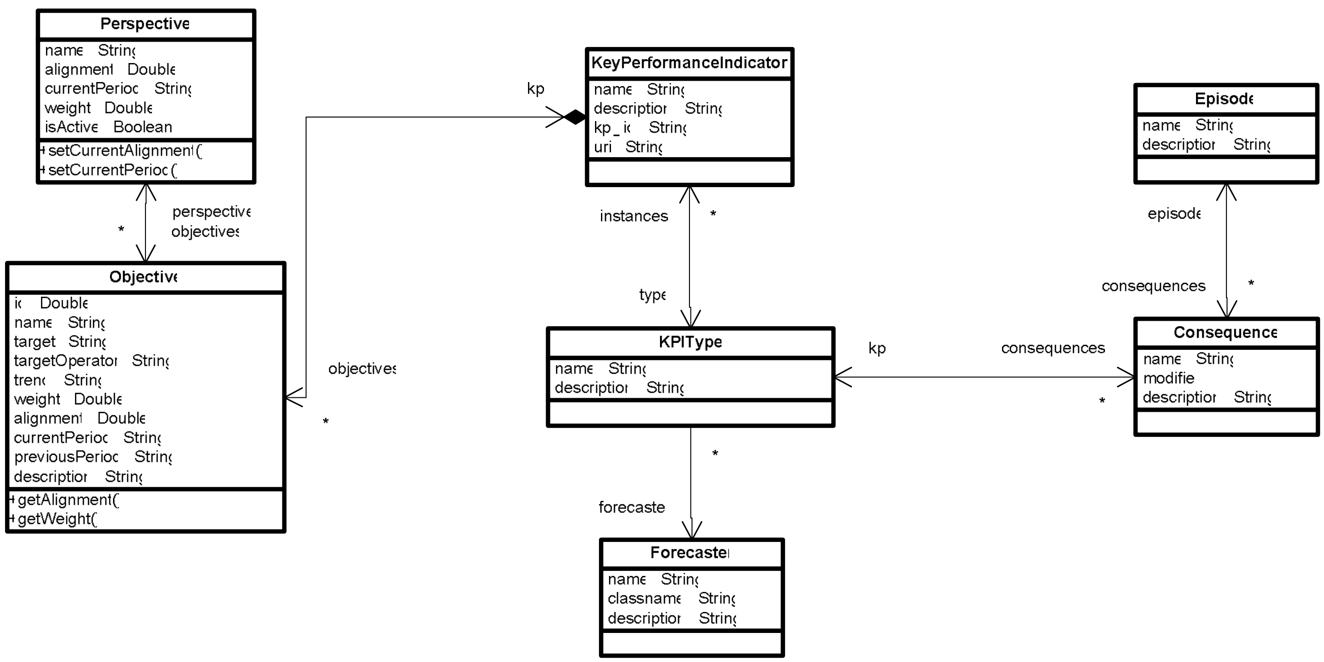

Figure 1. The MBO business objectives information model

balanced scorecard defines four perspectives: *financial*, *customer*, *business process* and *learning and growth*. Our model defines a perspective as a first class object, not limiting its usage to the traditional balance scorecard model. Perspectives do not represent a partition over the set of objectives defined. An objective can belong to more than one perspective.

EXAMPLES

An example of an objective defined through the model is "the aggregate service revenue generated over the current three-month period must be above 100,000 $". This is modeled in MBO by defining a KPI ι representing the aggregate cost of SLO penalty paid over the current three month period, represented by a dollar amount. The target of the objective is the region in the KPI space characterized by the inequality $\iota < 100,000\$$.

An example of perspective is a financial perspective, containing objectives such as the one listed above on the aggregate cost of SLO violations, or an objective that defines a target over a KPI representing the aggregated revenue generated in a given time period. A customer perspective could contain objectives defining targets over some KPIs representing quantitative measures of the customer satisfaction (measures of TCE: total customer experience), and so on.

MBO assigns importance weights to objectives and perspectives. The weights are used by the MBO engine to compare utility values of different objectives. The weight assigned to one perspective is propagated down to the objectives belonging to that perspective, as exemplified in Table 1.

Perspective	**Financial**		**Customer**
weight	80%		20%
Objective	Aggregated revenue in three month period	Aggregated cost of penalties for SLA violation in three month period	Total customer experience
weight	40%	60%	100%
adjusted weight for perspective	32%	48%	20%

TABLE I. OBJECTIVES, PERSPECTIVES AND IMPORTANCE WEIGHTS

2) *The MBO alignment computation engine: Aline*

The MBO alignment computation engine, Aline, reasons over the objectives and KPI defined through the MBO information model in order to assign a value of alignment to a given course of action among the ones available to the IT manager. Because in the decision support process we use this value of alignment as the utility of carrying out a given course of actions, we require a formal and quantitative definition of alignment, which we give in the next sub-section.

ALIGNMENT WITH BUSINESS OBJECTIVES AS THE UTILITY OF THE MANAGEMENT OPTIONS

We define the alignment with a given business objective as **the measure of the likelihood –** *given the best knowledge about the current situation* **– that the objective will be met**.

Before applying this definition of alignment, let us discuss why it behaves better for our purposes than other definitions of alignment that are sometimes used.

Let us recall the simple objective given in the previous section: the aggregate service revenue generated over the current three-month period (KPI) must be above 100,000$ (target). Let us suppose that 2 months into the period, the aggregate revenue figure amounts to 60,000$. A naïve measure of the alignment is derived by dividing the current amount by the target threshold, obtaining a figure for the 'alignment' of 60,000$ / 100,000$ = 60%. There are a number of problems with this definition. To begin with, the measure so defined does not take into account how far into the time period the aggregate value of penalties is measured. It is obvious to anyone that an organization is much better poised to meet (i.e. *aligned to*) its objective if the figure reads 60,000$ one month into the period than two months. But the naïve definition of alignment would miss this.

Having disposed of this, one possible improvement is to take the time dimension into account and compare the revenue/time interval figure generated so far with the one that characterizes the target. In this case, the situation given in the original example would amount to an alignment measure of (60,000$ / 2 mo) / (100,000$ / 3 mo) = 90%. In the example where the figure of 60,000$ refers to a one month period, the alignment measure would be 180%. The problems with this definition, in reverse order of gravity are: 1) it is difficult to associate an interpretation of a value for the alignment that exceeds 100% or results in a negative figure (which is possible with this definition); 2) it's really difficult to compare alignment across objectives (how to compare between an alignment of 1000% and 2000%?), and most importantly 3) it does not require one to take into account forseeable events that might impact the likelihood of the organization to eventually meet their objectives. About this third problem, suppose that in the example above (60,000$ at 2 months, 90% aligned) it's known that the third and last quarter the revenue slows down because of seasonality of the business. Now the figure for the alignment derived with this method is completely useless, as it is evident that it will be utterly improbable that the organization meets its objective.

It's easy to see that our definition of alignment behaves well against all the objections made to the alternative definitions. First off, the alignment always results in a figure between 0 and 1, which makes it easier to compare among alignment figures for different objectives. Most importantly, our definition copes well with the "seasonality" problem that was highlighted by the last variant of the given example. By reminding the reader that alignment is defined as the likelihood – to the best of one's knowledge – that the objective will be met, suppose an estimation is made that the revenue for the last month is uniformly[2] distributed in the interval [0$, 45,000$]. The likelihood of meeting the objective is equal to the likelihood of posting a top line figure for the last month of the quarter in excess of 100,000$ - 60,000$ = 40,000$, that amounts to (45,000$ - 40,000$) / 45,000$ = 11.11%.

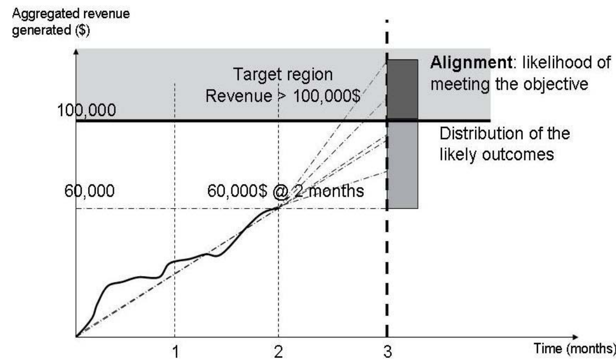

Figure 2. Alignment as likelihood of meeting the business objective

From the definition of alignment used here, and the kind of business objectives that we consider, it follows that our method requires some estimate of the future value of the KPIs. The estimate is captured as a distribution of probability over the relevant KPI spaces. In the rest of the document, we refer to such an estimate as a *likely outcome*. An outcome is characterized by the distributions of probability over the KPI spaces that it entails.

Our working hypothesis is that the actual method that is used to estimate the likely outcome does not matter so much as long as there is a simple way to estimate the likelihood of meeting the objective. Once again, the principle we follow is that we keep the cost of modeling low. Our hypothesis is that very sophisticated models will only add marginally to the accuracy in the computation of utility.

We plan to validate this hypotesis by carrying out experiments to determine the sensitivity of any measure of the goodness of the decisions suggested to the complexity of the methods used for determining workable figures for the alignment. In the worked example above, three alternative methods to estimate the alignment of the seasonal-sensitive organization to its revenue objective could be:

- elicit knowledge from a business expert (or a pool thereof) through the question "how likely do you think it is that you'll post a a revenue figure in excess of 40,000$ dollars this month
- use the simple uniform distribution model given above
- use a very complicated method that keeps into account a great number of variables, such as Box-Jennings' ARIMA [6].

[2] Used here for simplicity of calculation.

Our conviction is that the simplest method will be "just good enough" for the analysis that the decision making engine will have to perform. Accuracy is not the most important quality of the prediction, as it would be for a system that predicts revenue for business managers, where an error greater than 3% would be considered bad. The most important quality required to the model in our framework is that it be **low cost**. In a case where the complex prediction models would give figures of say 11.11% and 13.245%, using a "guesstimate" interval of say "15% to 20%" may still result in good enough suggestions. As we hint in figure 2, large increments in the complexity of the model (to which the cost of modeling is proportional) result in decreasing marginal quality of prediction.

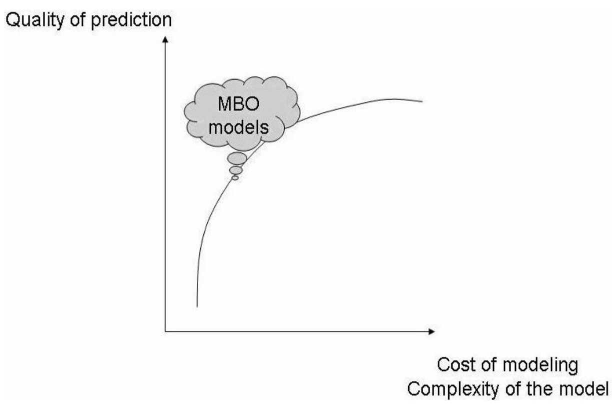

Figure 3. Low cost modeling yielding "just good enough" accuracy of prediction

The objective used in the example above was useful to compare our definition of alignment with alternative ones. Let's now switch to an example that is more relevant to IT management to see how we use alignment with the objectives as a measure of utility to rank alternative management options. The objective we consider here is "the aggregate cost of paid penalties for not meeting SLOs over the current three months period (KPI) must be below 10,000$ (target)".

We consider a situation where, two months into the period, the aggregated penalty cost amounts to 6,000$. For the sake of this example, we'll use a prediction model that assumes that the aggregated penalty cost over the last month of the quarter is uniformly distributed in [1500$, 4500$] (but a guesstimating oracle would do just as fine, as discussed above). The current measure of the alignment is therefore computed to 83.33% (the probability that the aggregated cost of penalty in the last month will exceed 4000$). At this point an incident occurs that is likely to disrupt the service being provided to a customer. The penalty associated to the service disruption amounts to 2000$. The options available to the IT manager are defined by the priority value to assign to the incident. If a) the incident is dealt with the highest priority, it is expected to result in a 25% probability of incurring in the penalty. If b) the incident is dealt with with lower priority the likelihood of ending up breaching the SLA is 75%[3]. Assuming risk-neutrality, these two options are characterized by expected new values of aggreagate cost of penalty of a) 6,000$ + 25%*2,000$ = 6,500$ and b) 6,000$ + 75%*2,000$ = 7,500$. Assuming independence of incidents, the distribution of the aggregated cost of penalty for the last month is unchanged. The resulting alignment resulting by acting on each option is therefore 66.66% and 33.33% respectively[4].

The measure of the alignment thus far calculated will be used to describe a (Von Neumann – Morgenstern[5]) utiliy function defined over the set of the management options. By summing up the value so obtained with the cost of carrying out the course of action related to the option, and then applying a convex combination of the utility-as-alignment of the option over the set of the objectives - each taken with its given weight representing the relative importance of the objective to the business – we obtain a (Von Neumann – Morgenstern again) utility function that the engine will use to rank the management options.

To exemplify, assume that three objectives were defined, and given weights representing the objectives relative importance. After calculating the alignment of the two options given above with respect to all the objectives, a total utility value is calculated for each of the options (table 2).

Objectives	**Aggr. revenue**	**Aggr. penalty**	**Cust. sat'n**	**Utility**
Weights	32%	48%	20%	
Alignment of Option a: ***assign high priority***	86%	67%	90%	Score: **0.7768** Alignment: **Good**
Alignment of Option b: ***assign low priority***	86%	33%	50%	Score: **0.5336** Alignment: **Fair**

TABLE II. CALCULATION OF UTILITY AS ALIGNMENT OF THE OPTIONS WITH THE BUSINESS OBJECTIVES

We cannot stress enough the fact that the utility values are only useful as an indication of which option will be preferrable to achieve a better alignment with the business objectives. The values so obtained are not meaningful per se;

[3] The same observations made above on how to obtain these values apply. Here an oracle is assumed in order not to break the flow of the argumentation. In the next section we give a fully worked example including how to derive these probability values.

[4] For option a), the alignment is equal to the likelihood that the aggregate cost for the last month does not exceed 10,000$ - 6,500$ = 3,500$. For b) the threshold figure is 2,500$.

[5] That the utility function so defined is a Von Neumann - Morgenstern utility function is quite easy to prove, but the proof would take us beyond the scope of this paper. Von Neumann – Morgenstern utility functions have the useful property of behaving well with respect to probabilistic outcomes.

they only make sense as an arbitrary utility value used to rank the options. This observation reinforces the hypothesis of the relative relevance of the particular method used for prediction to the final decision.

An alternative way of perceiving the utility value of the available options is through a *monetization process*, which states a mapping between the utility values calcolated through the alignment definition, and a monetary measure of the perceived goodness of the option. This is useful in that it allows instant comparison with measures of the monetary cost of executing the option. However, it's very important to note that the output of the monetization process is not meant to be an accurate monetary evaluation for the option, but rather just an input to the ranking process of the available options.

Without loss of generality, whatever the method chosen for forecasting the value of the KPIs at the end of the period, we will indicate the *default outcome* with $pdf_{default}(kpi)$.

MODELING THE IMPACT OF COURSES OF ACTION ON THE BUSINESS OBJECTIVES

This sub-section describes the core of the computation that the alignment engine carries out.

To follow our principle of keeping the cost of modeling low, we have to be very selective in choosing the relevant information to model. In general, the complexity required in modeling the effect that actions taken by the IT manager have on the KPI - and consequently on the business objectives – can quickly grow beyond acceptability. The main inventive step is to break down the chain of dependencies between actions and KPI values by inserting an intermediate step in the middle. We identify *episodes* that can have an impact on the KPIs. Unlike the actions that can be taken, the episodes that we model are usually of few different types and can be described quite easily in terms of the metrics underlying the KPIs. For the sake of an example, let us now introduce a couple of business objectives and their relative KPIs that we will refer to time and again within this paper. The first objective states that the aggregate cost of penalty for SLO violation in a three month should be less than 10,000$. The second objective states that the total number of SLO violations for all customers in a three month period should be less than 15. We note here that whatever the actions that an IT manager can take, the only relevant episodes to the KPIs above described are *SLO violations*. On identifying the relevant episodes, the calculation of the alignment for any given course of action is therefore split in two steps:

1) estimate the likelihood that a given course of action will result in a relevant episode (e.g. SLO violation) will take place;

2) calculate the alignment with the business objectives both when the episode takes place and when it does not; and use these values to compute a final measure of the alignment given the likelihood of the occurrence of the episode that was computed in the previous step.

The method here described results in much simplified calculations because the only information that is required for a given course of action is what is needed to estimate the likelihood of the episode. In the next section we fully work out an example of how to effectively utilize the alignment engine in building a solution for incident management. Here we skip the first step (in the next section we present a fully worked example that applies to the incident management domain) and describe in detail the calculations that the engine carries out.

First, we need to model the impact that a likely SLO violation has on the KPIs described above. In the current version we use deterministic[6] functions to express that dependency. For example, when considering the effect of a likely SLO violation on the aggregated cost of penalties KPI, we represent it through the function

$$f_{impact}(kpi_{penalty}) = kpi_{penalty} + penalty(SLO) \qquad (1)$$

The formula is intended to mean that when a SLO violation occurs, the value of the KPI measuring the aggregate cost of penalty is increased by the penalty relative to the SLO violation. Similarly, considering a customer related KPI that measures the number of violations experienced by all customers in a period of time, we write

$$f_{impact}(kpi_{violation}) = kpi_{violation} + 1 \qquad (2)$$

The outcome that follows an SLO violation is therefore obtained by composing the effect of the SLO violation with the default outcome (no SLO violation) as indicated by the following formula

$$pdf_{violation}(kpi) = pdf_{default}(f_{impact}^{-1}(kpi)) \qquad (3)$$

We now determine the likely outcome of closing an incident by a given time. We have already determined the likelihood of SLO violation λ in function of the time taken to close the incident. The likely outcome is given by the combination of the default outcome if the violation doesn't occur with the modified outcome if the violation does occur.

$$pdf_{combined}(kpi) = \\ (1-\lambda)\,pdf_{default}(kpi) + \lambda pdf_{violation}(kpi) \qquad (4)$$

The method followed is illustrated in Figure 4, next page.

[6] We have built a prototype of Aline that can work with dependencies expressed through probabilistic functions. Besides impacting the ease of computation, this leaves the other steps of the computation unchanged.

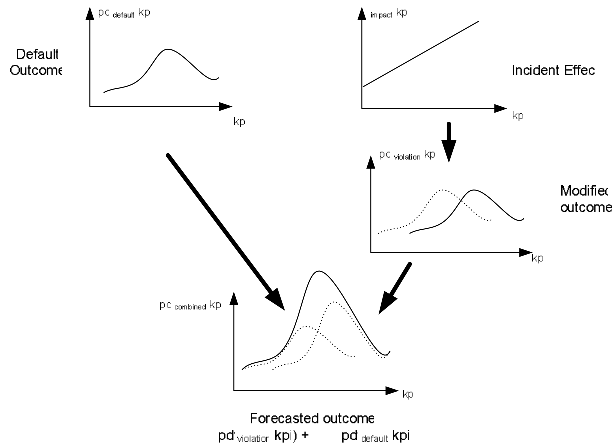

Figure 4. Forecasting the effect of SLO violations on the KPI space

COMPUTATION OF ALIGNMENT

To compute the alignment of the forecasted outcome with a given business objective, we simply integrate the probability density function for the outcome within the target region defined by the objective in the KPI space

$$alignment_{objective} = \int_{kpi \in Target(objective)} pdf_{combined}(kpi)dkpi \quad (5)$$

The alignment of the outcome with all the business objectives is finally obtained by summing the contribution of the alignment with each of the business objectives, each taken with their own weight.

$$alignment = \sum_{\forall objective} weight_{objective} \cdot alignment_{objective} \quad (6)$$

It can be noted that because of the linearity of the combination operated above, we also could first independently compute the default alignment and the alignment in case of violation; and later linearly combine them with the likelihood that the violation will in fact happen. The end result does not change, and in this way the calculations are much simpler and faster to carry out, since it is much simpler to deal with scalars than with probability density functions, and the number of integration required is dramatically reduced.

3) Building solutions for IT Service Management driven by business objectives using MBO

We conclude this section with a discussion on how MBO, can be used to build solutions for IT Service Management driven by business objectives. So far we have described the components of MBO that are common across the various IT Service Management domains; those are the MBO information model and the alignment engine. However, in order to employ MBO in a full solution, the MBO reusable components need to be complemented with corollary components that are specific to the domain.

In particular we need a *forecasting module* that can estimate the likely outcomes in terms of KPI value (what we called the default outcome in the section on impact modeling). Once Aline is given the default outcome then it can compute the alignment for each alternative course of action, but the default outcome is necessary for bootstrapping the calculation. The problem of forecasting from time series is well studied in the literature [6]. It has to be noted that, since the output of the forecasting module is only one of a chain of steps that Aline goes through, accuracy in the prediction is not so important here.

For some domains where the space of possible courses of action is limited, the alignment engine could be used in a brute-force mode to calculate the alignment of each of the options and choose the course of action which results in the optimal alignment with the business objectives. However, life is not always so simple. When the number of alternative courses of action is susceptible of combinatorial explosion (as is the case for incident prioritization), the approach that we take is to model the decision problem according to multi-attribute utility theory ([7]). In cases like these, it will be necessary to develop a *solver* module that tackles the decision problem using the output of Aline as its input to calculate the utility of alternative options as alignment with the business objectives.

The conceptual architecture of a solution based on MBO is represented in Figure 4.

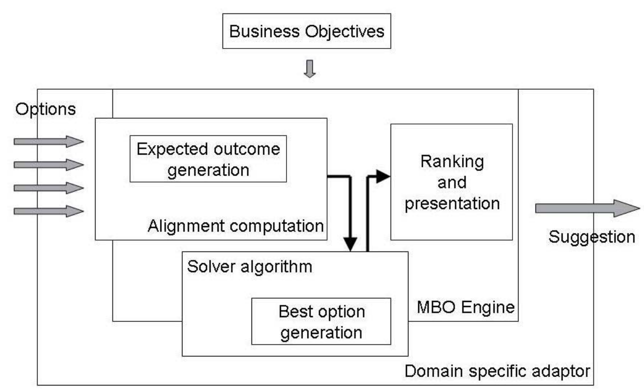

Figure 5. Conceptual architecture of MBO-based solutions

Following the flow in Figure 3, the alternative options that are available to the IT manager are passed on to Aline (the alignment computation module) to obtain a value of utility. Aline in turn uses a forecasting module (expected outcome generation) to calculate the value of alignment according to the method described above. The output of Aline is then used by a solver module that determines the best option among the ones available.

The solver and forecasting modules that we introduced in abstract terms in this section are described in much more concrete terms and detail in the next section, where we present a fully fledged MBO solution for incident management.

III. Application of MBO to Prioritization of Service Level Degradation Incidents

The problem solved by MBO in the incident prioritization solution is to assign priority levels to a set of service level degradation incidents so as to maximize the alignment with a given set of business objectives. Let us begin by introducing (or recalling) some of the lexicon that we use during the description of the incident prioritization solution.

A *Service Level Agreement* (SLA) is contracted with a *customer* and contains a set of *Service Level Objectives* (SLO). Each SLO specifies an acceptable range of values for a given system metric, through the definition of a *violation threshold* for the SLO. A penalty cost is associated to SLO violation, which occurs when the metric value surpasses the violation threshold. Besides the violation threshold, a *jeopardy threshold* is also specified. Metric values are obtained by probes deployed by the management system and monitored by a monitoring component. A degradation of service level incident for an SLO occurs when the monitoring component reports on a metric value surpassing the jeopardy threshold for the SLO. An incident management system collects and organizes the information on the degradation of service level incidents by assigning priority values to them together with other information on the lifecycle of the trouble ticket associated to the incident. The problem that we solve is to suggest how to deal with the incidents so as to minimize the expected cost of violation of the SLOs. In this work we only consider incidents generated on detection of service level degradation or violation, although the general techniques that we present are more widely applicable.

To solve the incident prioritization problem, we operate on the following steps:

- Compute the likelihood of violation of an SLO in function of the time taken to close a jeopardy incident, using an ad-hoc module developed for the incident prioritization problem;
- Compute the alignment with the business objectives in function of the likelihood of violation, using Aline;
- Prioritize the incidents in function of the alignment with the business objectives, using a solver module based on Integer Linear Programming (ILP) [8].

A. Likelihood of SLO violation as a function of the time of closure of the service degradation incident

We make the assumption that the IT manager is required to specify a time value that represents the expected time that it will take for the system to move from the jeopardy state to the violation state if no measures are taken (expected time from jeopardy to violation). We assume an exponential distribution of the time from jeopardy to violation if no corrective actions are taken. The parameter of the exponential distribution, λ, is defined as the inverse of the expected time from jeopardy to violation.

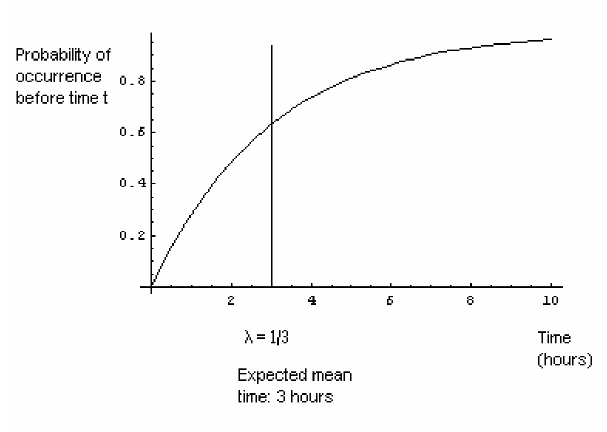

Figure 6. Likelihood of SLO violation as a function of the time of closure of the service degradation incident

The plot in figure 5 represents the cumulative distribution function of the distribution of probability associated with the time from jeopardy to violation. The analytic form of the curve is given by the equation $p = 1 - e^{-\lambda t}$, where p represents the probability of violation if the incident is closed after t from its starting time. In the example given, for a value of $\lambda = 1/3$, corresponding to an expected mean time of 3 hours, the probability of a violation occurring within 4 hours from the jeopardy alarm if no corrective measures are taken equals to $1-e^{-4/3}$=76%.

B. Alignment with the business objectives in function of the likelihood of violation

Aline is invoked to compute the alignment with the business objectives. The alignment computation steps are described in detail in the previous section on the alignment engine. Here we exemplify the computation by presenting a fully worked out example.

Worked Example of Alignment Computation

Let's consider a service degradation incident that relates to an SLO for which the expected time from jeopardy to violation is three hours. From the previous sub-section, the likelihood of SLO violation expected from closing the incident in four hours is given by $1 - e^{-4/3}$ = 76%. The penalty associated to violating the incident is set at 1,000$.

Let's assume that two simple business objectives have been defined. The first objective states that the aggregate cost of penalty for SLO violation in a three month should be less than 10,000$. The second objective states that the total number of SLO violations for all customers in a three month period should be less than 15. The objectives are deemed to be equally important, therefore each carrying an importance weight of 1/2.

Two months into the period, the current readings of the KPIs are 6,000$ in penalties paid for 10 SLO violations. For simplicity's sake and without loss of generality we here assume that the forecasting module predicts a default outcome characterized by uniform distributions with extremes [7,500$, 10,500$] for the penalty KPI and {13, 14, 15, 16, 17} for the violations KPI.

The alignment with the business objectives entailed by the default outcome is computed as the probability of meeting the objectives given the expected default outcome. For the first objective we obtain an alignment measure of 5/6 (such the probability that the penalty KPI wind up be below 10,000$ given that it's distributed uniformly in [7,500$, 10,500$]. For the second objective it's obvious to observe that the alignment measure is 2/5. The alignment with all business objectives is therefore 1/2*5/6 + 1/2*2/5 = 31/60 = 0.517 or 51.7%.

The effect of the likely SLO violation on the KPIs is to increase the penalty figure by 1,000$ and the number of violations by 1. This would therefore define an outcome characterized by uniform distributions with extremes [8,500$, 11,500$] for the penalty KPI and {14, 15, 16, 17, 18} for the violations KPI. If the violation were to occur, that would entail level of alignment of 1/2 and 1/5 for each objective respectively, and therefore an alignment with all business objectives of 1/2*1/2 + 1/2*1/5 = 7/20 = 0.350, or 35%.

Since the violation is expected to occur with a 76% likelihood, the measure of alignment for the combined case becomes: 24%*0.517 + 76%*0.350 = 0.390, or 39%.

Once again, the reader should not read too much into the actual figure for the alignment, besides considering it a useful way to discern among alternatives.

Forecasting

As far as the forecasting module is concerned, the method is only relatively sensitive to the accuracy of the prediction of the forecaster, as we noted above. This means that that MBO's suggestions are good if the accuracy is just good enough.

The simple method that we have chosen for this system predicts the mean value of the KPI at the end of the period simply by extrapolating its current value. The forecasted value of the KPI is considered to be normally distributed with mean calculated as above and variance set at a sensible customized value (for example the square of 5% of the mean value). For the example given above, when the penalty KPI reads 6,000$ two months into the period, we forecast its value at the end of the three month period to be normally distributed with mean 9,000$ and a variance of $(450\$)^2$, and therefore characterized by

$$pdf_{default}(kpi_{penalty}) = NormalDistribution(9{,}000\$, (450\$)^2)(kpi_{penalty}) \quad (7)$$

We plan to experiment with more advanced methods such as ARIMA [6] and carry out sensitivity analysis experiments to assess the goodness of the prioritization that we obtain.

C. *Incident prioritization to maximize alignment with business objectives*

Once the business impact of the incidents has been computed, we are faced with the problem of prioritizing them so as to minimize the total impact on the business. Our system requires the use of a *priority scheme*. Together with the definition of a set of priority levels that are used to classify the incidents (defined by the ITIL [5] guidelines for incident management), we require the user to express constraints on what are the acceptable distributions of incidents into priority levels. For any priority level the users can either force the incidents to be classified according to some predefined distribution (e.g. 25%-30% high, 40%-50% medium, 25%-30% low), or define a minimum and maximum number of incidents to be assigned to each priority level. Our method finally requires an *expected maximum closing time* for the incidents that are assigned to a certain priority level.

1) *The incident prioritization problem*

We here present a mathematical formulation of the incident prioritization problem as an instance of the generalized assignment problem. The generalized assignment problem is an integer optimization problem that is well studied in the operation research literature and for which very efficient algorithms have been developed.

Suppose we are required to prioritize between n incidents $i_1..i_n$ into m priority levels $p_1..p_m$. We introduce a variable x_{jk}, $j=1..m$, $k=1..n$ that assumes the value $x_{jk}=1$ if the k^{th} incident is assigned to the j^{th} priority level and $x_{jk}=0$ otherwise.

By observing that the alignment of each incident can be calculated depending on what priority level it is assigned to, if t_j is the expected time of completion for incidents assigned to priority level j, then obviously the alignment yielded by assigning the k^{th} incident to the j^{th} priority level is given by the alignment of closing the incident by the time t_j, which we know how to compute from the previous sections. We'll call this measure of alignment $a(i_k,t_j)$ for short

The next thing to be noticed is that the constraints that the user imposes on the distribution of the incidents into priority levels can be trivially translated into minimum and maximum capacity constraints for the priority levels. For example, when dealing with n=200 incidents, the requirement that at least 40% of the incidents be assigned medium priority (assume that is priority level p_2) would read: $\sum_{k=1}^{n} x_{2k} \geq 80$

$$\sum_{k=1}^{n} x_{jk} \geq c_j \quad and \quad \sum_{k=1}^{n} x_{jk} \leq C_j \quad \forall j = 1..m \quad (8)$$

In general we assign a minimum (c_j) and maximum (C_j) capacity constraint for a priority level j that are symbolized as

The mathematical formulation of the incident prioritization problem (IPP) becomes:

$$(IPP) \quad \min \quad \sum_{j=1}^{m} \sum_{k=1}^{n} a(i_k, t_j) \cdot x_{jk} \tag{9}$$

$$s.t. \quad \sum_{k=1}^{n} x_{jk} \geq c_{jk} \quad and \quad \sum_{k=1}^{n} x_{jk} \leq C_{jk} \quad \forall j = 1..m$$

$$\sum_{j=1}^{m} x_{jk} = 1 \quad \forall k = 1..n$$

$$x_{jk} = 0 \; or \; 1 \quad \forall j = 1..m, k = 1..n$$

The solution of this problem yields the optimal assignment of priorities to the incidents.

INCIDENT PRIORITIZATION ALGORITHM

Here is a pseudo-code description of the incident prioritization algorithm

Input:

A number of SLOs, each modeled with the following information:

- Violation threshold for an SLO
- Jeopardy threshold for an SLO
- Penalty cost for SLO violation
- Expected time from jeopardy to violation (if no corrective measures are taken)
- A set of priority levels for incidents, modeled with the following information:
- Constraints over the number of incidents to be assigned to each priority level
- Expected maximum closing time of incidents assigned to the priority level
- A number of service degradation incident, modeled with the following information:
- Incident start time (the time when a jeopardy alarm was raised)

Output:

A complete prioritization of the incidents that assigns to each of them a priority level

Steps:

1. Compute the default alignment with business objectives (section 3.2)

2. For each incident:

2.1 For each priority level:

2.1.1 Compute the likelihood of violation of this incident if assigned to this priority level (section 2, using the expected maximum closing time for the priority level)

2.1.2 Compute the alignment with business objectives if a violation occurs (section 3.4)

3. Solve the incident prioritization problem (section 4, using the values of alignment derived from the previous step)

IV. FUTURE WORK

A. *Experiments to evaluate the goodness of the decision support*

In order to evaluate the practical viability (*efficiency*) of the approach, we have carried out performance experiments to verity that the prototype can scale up to a number of instances of incidents that would occur in a real life scenario. However a much more interesting set of experiments could be designed to assess the *effectiveness* of the decision support provided by the MBO technology. In order to do so, we would need to deploy the solution in a real life IT environment and measure metrics of goodness such as "how often does a skilled operator override the calls of our systems".

The method here exposed is depending on forecasting of KPI values, and may therefore be subject to the garbage-in-garbage-out principle. However, one of our main underlying assumptions is that because the forecasting is but an intermediate step in the whole process, the final suggestions will not be so sensitive to the goodness of the forecasting. We plan on carrying out sensitivity analysis experiments to validate our assumption.

B. *Extension to change management and other ITIL processes*

The approach here described lends itself quite naturally to be extended to the ITIL process of change management. In the short term, we are planning to apply this method to the prioritization of request for changes. In the medium term we will experiment with ideas in the space of planning and scheduling for change management, much along the same lines as Keller's CHAMPS work [10], but with a direct link to optimizing planning and scheduling with respect to alignment with business objectives.

C. *Integration with tools of the HP Openview™ Suite*

HP Openview Service Desk™ (OVSD) is the tool of the Openview management suite that provides coverage for a number of ITIL processes, such as incident management, service level management, change management and others. OVSD allows a user to define a hierarchical service structure with multi-tiered SLA capabilities to describe the relationship between a higher level business service and the supporting operation management service.

In [9] we described a prototypical implementation of the integration of a solution for incident management based on MBO with the Service Level Management (SLM) capability of Openview Service Desk v. 4.5 (OVSD 4.5). At the moment when we write, there are plans for an incident prioritization module based on a simplified version of the algorithms here described to be made commercially available as part of a future release of OVSD.

D. Continued development of the Aline prototype

The current research prototype of Aline is under continued development and can be made available for research purposes on contacting the authors.

V. Related Work

The problem of business-IT alignment has recently risen in importance within the IT community. However, most of the approaches (of which [11, 4] are notable examples) are targeted at business executives and try to quantify the value of IT investments, therefore taking a strategic – long term view. In contrast, our work is targeted at IT managers, as it is demonstrated by our choice of making it commercially available through an IT management software suite such as HP Openview.

In the IT management research community literature, Sauvé et al [12] take a very similar approach to ours. However, they put more emphasis than we do on the systems and metrics modeling, which makes our works nicely complementary.

Buco et al [13] present a business-objectives-based utility computing SLA management system. The business objective(s) that they consider is the minimization of the exposed business impact of service level violation, for which we presented a solution in [14]. However, in this work we go far beyond just using impact of service level violations. We provide a comprehensive framework and a method for incident prioritization that takes into account strategic business objectives such as total customer experience thereby going a long way towards the much needed alignment of IT and business objectives.

An aspect of Keller's CHAMPS [15] that is relevant to this work is the automation of change management based on electronic contracts expressed through WS-Agreement[16]. As in Buco's [13] case, the cost and profit information are derived through SLA definitions.

Jeng et.al [17] take an interesting approach to business performance management that in the tradition of [ponder, LGA] is based on prescribing and enforcing logic-based policies. In contrast our approach is based on quantitatively assessing the utility of alternative options for IT management against business objectives (which are in turn express quantitatively).

VI. Conclusions

In this paper we have addressed the problem of ensuring business-IT alignment, by describing a system for decision support in IT Service Management driven by alignment with the business objectives of the enterprise that the IT supports. Our technical proposition, called IT Management by Business Objectives (MBO) is applicable to most of the domains of IT Service Management, such as incident management, change management, and so on. The technology consists of some components that are reusable across domains, together with guidelines and patterns for building complementary components in order to develop domain-specific solutions.

References

[1] Control Objectives for Business Information-related Technology COBIT. http://www.isaca.org/cobit.htm

[2] R. S. Kaplan, D. P. Norton, "The Balanced Scorecard: Translating Strategy into Action" Harvard Business School Press, 1996.

[3] ITSM reference. http://www.itsm.org

[4] B. H. Reich, I. Benzabat, "Measuring the linkage between business and information technology objectives", MIS quarterly, Society for Information Management and The Management Information Systems Research Center, 1996, vol 20, pages 55-81.

[5] IT Infrastructure Library, "ITIL Service Support", Office of Government Commerce, June 2000

[6] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. Jenkins, "Time series analysis: forecasting and control", Holden-Day, San Francisco 1976

[7] P. C. Fishburn, "Utility theory and decision making", Krieger Publication Co., Huntington, NY, 1979

[8] A. Schrijver, "Theory of linear and integer programming", John Wiley and Sons, 1986

[9] C. Bartolini, M. Sallé, "Business-driven prioritization of service incidents", in Proc. Distributed Systems: Operations and Management, (DSOM 2004), LNCS Springer 2004

[10] A. Keller, J. Hellerstein, J. Wolf, K. L. Wu, V. Krishnan "The CHAMPS System: Change Management with Planning and Scheduling". In Proc. Network Operations and Management Symposium (NOMS 2004), IEEE Publishing, 2004

[11] T. Pisello, P.A. Strassmann, "IT Value Chain Management– Maximizing the ROI from IT Investments", Digital Publications from the Information Economics Press, 2004

[12] J. Sauvé, F. Marques, A. Moura, M. Sampaio, J. Jornada, "SLA Design from a Business Perspective", In Proc. Distributed Systems: Operations and Management, (DSOM 2005), LNCS Springer 2005

[13] M. J. Buco, R. N. Chang, L. Z. Luan, C .Ward, J. L. Wolf, and P. S. Yu, "Utility computing: SLA management based upon business objectives", IBM Systems Journal, 2004.

[14] M. Sallé, C. Bartolini, "Management by Contract", In Proc. Network Operations and Management Symposium (NOMS 2004), IEEE Publishing, 2004.

[15] A. Keller, "Automating the Change Management Process with Electronic Contracts", In Proc. Service Oriented Solutions for Cooperative Organization (SoS4CO 2005), IEEE Publishing, 2005

[16] A. Andrieux et al. "Web Services Agreement Specification (WS-Agreement)", Global Grid Forum, 2004

[17] J. J. Jeng, H. Chang, K. Bhaskaran, "Policy Driven Business Performance Management", In Distributed Systems: Operations and Management, (DSOM 2004), LNCS Springer 2004

URLs last checked on December 2005.