UFPB CCT Departamento de Sistemas e Computação

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia

DSC
Departamento de Sistemas e Computação

Av. Aprígio Veloso, 882 — Bodocongó
Caixa Postal 10.106
58.109-970 — Campina Grande — PB — Brasil
Fone: 310-1119 — Fax: 310-1124
e_mail: dsc@dsc.ufcg.edu.br

Relatório Técnico

RT DSC-001/2004

AN IT BUSINESS IMPACT MANAGEMENT FRAMEWORK

J. P. Sauvé, J. A. B. Moura, M. C. Sampaio, J. F. H. Jornada, E. Radziuk

Universidade Federal de Campina Grande
Hewlett-Packard Brazil

(13 páginas)

junho 2004

An IT Business Impact Management Framework

J. P. Sauvé[1], J. A. B. Moura[1], M. C. Sampaio[1], J. F. H. Jornada[2], E. Radziuk[2]
[1]*University of Campina Grande (UFCG), Brazil* - [2]*Hewlett-Packard Brazil*
{sampaio, jacques, antao}@dsc.ufcg.edu.br - {joao.jornada, eduardo.radziuk}@hp.com

Abstract

A Business Impact Management (BIM) framework is developed. It enables IT infrastructure and services management – particularly with respect to fault and performance management – to be performed with a business perspective, that is, by revealing the impact that IT faults and performance degradations have on the business. The framework explicitly includes a business layer and business metrics and allows drill-down operations to analyze cause-and-effect relationships. The framework was designed with the particular aim of permitting low-intrusion BIM, which requires relatively little effort to be constructed and which does not rely on heavy instrumentation of the IT infrastructure. The models so constructed can be used in an operations setting, since they provide instantaneous measures of the business impact of IT problems. New metrics (health, loss and SLA non-compliance risk) are introduced as part of the framework.

1. Introduction

Aligning Information Technology (IT) with the business has developed into an important facet of the task of managing IT [3,6,10,13,16]. In this paper, we focus on the tasks of managing the IT infrastructure, particularly with regard to fault and performance management. The IT manager is now asked to manage the IT infrastructure from the business perspective, that is, using metrics understandable to business executives, and reflecting business priorities. This is considered important since typical metrics used in IT management (availability, response time) mean nothing to a business executive. Climbing away from the IT perspective to reach closer to the business perspective is being tackled in different ways, the most important among which are Business Process Management (BPM) [18,20] and Business Impact Management (BIM) [8,13]. Although both BPM and BIM take business processes into account, they are fundamentally different in that BPM aims to measure and improve business process execution quality whereas BIM aims to measure the impact of IT faults and performance degradation using metrics that are couched in business terms, frequently of a financial nature.[1]

BIM can be used by IT managers and also by business executives. For the IT manager, BIM helps to prioritize IT actions so that trouble conditions with the most business impact are fixed first; it allows one to drill down from high-level effect to the causes of the problems by following a most-critical path; it is helpful in gathering data to perform capacity planning with business priorities in mind. For the business executive, BIM provides measures concerning the alignment between IT and the business.

BIM is supported by a service impact model, or, more generally, a *BIM model*. This model includes all relevant entities and their metrics, defines their relationships, and specifies how business metrics are measured or calculated. Current work in BIM and BPM frequently uses the term *business metric* but really deals with *business process metrics* such as "rate of business transactions completed in a business process" and "end-to-end response time" [13]. These metrics are closer to the business than the typical IT metrics but are still not what we call business metrics. In our definition, a business metric is one that may readily be understood by a top business executive and that is related to the final performance of the business, in terms of revenue, profit, cost, etc. By comparison, a business process metric is an attribute of a business process. Observe that, although BP metrics are related to the business (they may be considered *performance drivers* for Business Metrics), they are not indicative of final business performance. It is of interest to map BP metrics to business (performance) metrics.

Articles on BIM and BPM to date offer little insight into the mappings of IT or BP metrics to business metrics. Most BIM articles are white papers put out by companies that may consider such mappings trade secrets and are understandably unwilling to divulge them [3,8]. Some of the proposed solutions are hardly business-oriented nor they can be created simply, requiring a lot of instrumentation effort to gather data about the BPs of interest [17].

Considerable work in the BPM area is being devoted to BP specification and execution languages [5]; with major focus on Web services [4,21]. Extensions to such languages to facilitate performance evaluation measurements are also being proposed [12,14]. Performance monitoring of BPs [15] – also known as BP Intelligence, BPI [6], and Business Activity Monitoring,

[1] There is a sorry lack of standard terminology here. BIM is also variously called "Business Service Management", "Service Impact Management", "Business Centric Service Level Management", etc. See for example [22].

BAM [7] - is another hot area. In BPI, a fair amount of instrumentation is tied to some data warehousing and data mining facility. The term "BAM" was coined by the Gartner Group to refer to "software products aimed at providing real-time access to critical business performance indicators to improve speed and effectiveness of business operations". This definition notwithstanding, most BAM solutions are discrete event monitors lacking overall BP context and do not consider nor offer clues as to how business level metrics are to be brought into the picture. Most companies recognize the importance of BAM but have "no effective way to collect, aggregate and analyze execution statistics" [20]. Modeling and simulation of BPs [1], complement well "real-life" executing BP monitoring proposals and are less intrusive. Simple low-intrusion models could be an initial step towards assisting them in aggregating and analyzing BP performance data.

Most of the literature on BIM and BPM (and BAM) is company produced. Academic production for these topics is scarcer so far. We believe – and the reviewed literature seems to indicate – that an interesting academic contribution to BIM / BPM – see [16], for instance - would be the construction of low-intrusion, theoretical models that could provide preliminary information for IT alignment with business – i.e., for measuring the IT impact on business. This paper contributes with such models.

Our objectives are to develop BIM models for *operations* BIM, that is, BIM used for short-term problem evaluation and resolution of IT operations, rather than long-term tactical or strategic BIM. We aim for a full model that does not stop at the business processes but goes on to explicitly include the business in the model, especially through the proper application of business metrics. Finally, a very important objective is to produce *low-intrusion* BIM solutions; this is in response to comments garnered from users that highlight the high cost of deployment of currently available BPM and BIM solutions [8].

The paper is organized as follows: we review the requirements for our BIM model in section 2 and describe it in section 3; validation considerations are discussed in section 4; our conclusions follow.

2. Model Requirements

The BIM model must satisfy the following requirements in order to accomplish its objectives.

1. **Business Impact**. The model must show the impact of faults or performance degradation in the IT infrastructure couched as Business Metrics or other Key Performance Indicators (KPI). A KPI is similar to a business metric but may be more abstract, not representing something tangible to a top business executive. An example is an "alignment index" that measures how aligned IT is to business objectives, but only has relative meaning (a higher value means "more alignment"). Thus, the model must provide mappings between the IT layer and the top business layer.
2. **Drill-down**. The model must provide drill-down capabilities in order to navigate from top level effects to underlying causes. In the drill-down model, the intermediate entities must be familiar either to a business executive (a business unit, for example) or to an IT manager (an IT service or component). The model must therefore provide metrics at all layers of the drill-down model, in order to satisfy top-level managers, middle-level managers, line-of-business managers, IT managers, etc. With a proper drill-down function, the model will allow IT staff to prioritize service requests (trouble tickets) based on business impact.
3. **IT measurements**. The model must accept as input measurements performed on the IT infrastructure and relate those values to the business metrics. Thus the model must include IT entities on which measurements are performed.
4. **Low intrusion**. The model should be as low-intrusion as possible. A BIM model is said to be low-intrusion if it exhibits the following characteristics:
   - It requires little modeling effort (the model is simple);
   - It requires as little instrumentation as possible on the IT infrastructure;
   - It requires little or no access to other business data (such as financial data, business process execution data, etc.);
   - It requires little or no modification to be implemented in legacy software;
   - It requires little or no input to define final-user requirements;
   - It requires little or no configuration (parameterization, etc.);
   - It is robust to changes in the business and IT entities and their relationship in the sense that changes can rapidly be effected over time to keep the model synchronized with reality;
   - It requires little or no effort to calibrate or for baselines to be established.
5. **Flexibility**. The model should be flexible, allowing the addition or removal of entities.
6. **Operations model**. The model must be usable in an operations environment; in other words, it must be possible to *instantaneously* calculate changes in business metrics as a result of changes in infrastructure measurements.

7. **Framework**. The model must provide a standard, generic abstract part and allow concrete instantiation is several ways. In other words, the abstract part says *what* the model requires, while the concrete instantiation describes *how* the information is provided. The abstract model may thus be considered a *framework* for the construction of a family of concrete models. The differences between the concrete models involve the entities modeled and the particular ways in which certain data is obtained or calculated within the model entities.
8. **Business changes**. The model should allow business managers to change business priorities frequently and expose how misaligned IT becomes as a result.
9. **Base for strategic model**. The operations model should produce values that will allow a long-term strategic BIM model to be developed later on.

3. The Framework

We now proceed to describe the framework for Business Impact Management.

3.1 A Layered Model

Requirements 1 [Business Impact], 2 [Drill-down] and 3 [IT measurements] suggest a layered model with at least two basic layers: the IT layer and the business layer. The IT layer is below the business layer[2] and feeds information from the IT infrastructure to calculate business metrics. More layers can be added, for two reasons: additional layers make it easier to generate final business metrics since they are stepping stones with intermediate metrics that make crossing the IT-business gap more intuitive; secondly, additional layers can provide a better drill-down model, as long as the entities encountered in the intermediate layers are familiar.

In fact, our model includes four layers (see Figure 1). The lowest layer is the IT Components Layer and includes such entities as routers, switches, host computers, and other basic IT components such as database services. The second layer is called the IT Services layer and comprises final IT services as accessed by users or other systems; these typically include software applications. The Business Process Layer is next and consists of those processes that make use of IT services. Finally, the Business Layer is where business entities are modeled and final business metrics calculated. Each layer will be described in more detail in further sections. For now, it is sufficient to say that each layer is responsible for measuring or calculating *metrics*. These can then be subjected to Service-Level Agreements, can be shown on management dashboards used by IT managers or business executives, and can also percolate up to be used by the above layer for further calculations. Drill-down operations can proceed from the top layer to lower layers to investigate cause-and-effect relationships. The topmost layers are more appropriate to business executives while all layers are important to an IT manager.

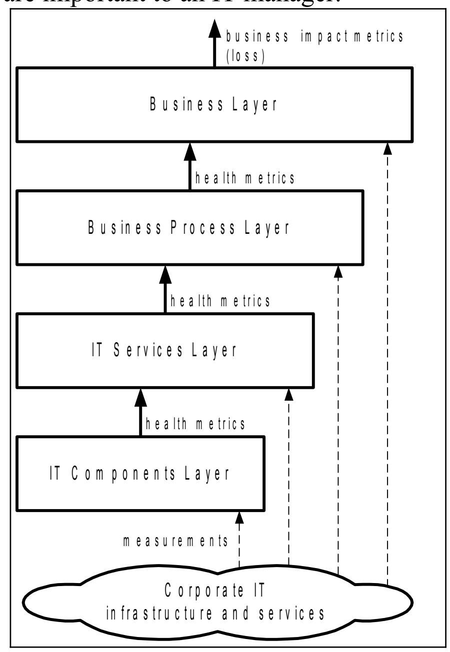

Figure 1: The basic four-layer model

So far, the model is quite abstract and generic in that no particular metric or formula used in calculating metrics is imposed. The model will become less abstract as we describe it more fully but will still retain a lot of elbow room to create different concrete models. This is necessary since different enterprises have very different instrumentation realities; for example, some enterprises have Business Process Management Systems (BPMS) in place while others do not. It is very important to us that the framework created enable enterprises with very little instrumentation to be able to perform BIM, using as little intrusion as possible. In general, with more instrumentation, metrics are more likely to be measured; with little instrumentation, they are more likely to be calculated.

3.2 General Layer Organization

The framework's four layers have many characteristics in common and these are described in this section. Figure 2 will be of help in understanding a general layer's organization. A layer is composed of *entities* of a particular kind; for example, in the business process layer, entities are business processes. Entities possibly have *attributes* and always have one or several associated *metrics*. Notice that a layer may have *neighboring layers*,

[2] Following convention, a layer obtains information from the layer below and provides information to the layer above it. "Above" and "below" merely refer to the way the layers are drawn in figures.

either above, or below, or both. Neighboring layers are linked through dependencies and metrics. The aim of a layer is threefold: first, a layer must produce metrics for SLA auditing and exhibition in management software dashboards. This is show on the left of the figure. Second, a layer must produce metrics for consumption by the layer directly above it. Metrics can either be based on measurements obtained from the IT infrastructure or can be calculated from lower-layer metrics plus suitable internal layer information (such as mapping functions, parameters, an entity's attributes, etc.). It is more common to measure closer to the bottom and to calculate closer to the top where metrics are not usually measurable quantities. Third, a layer must provide a drill-down model to be used inside the layer itself.

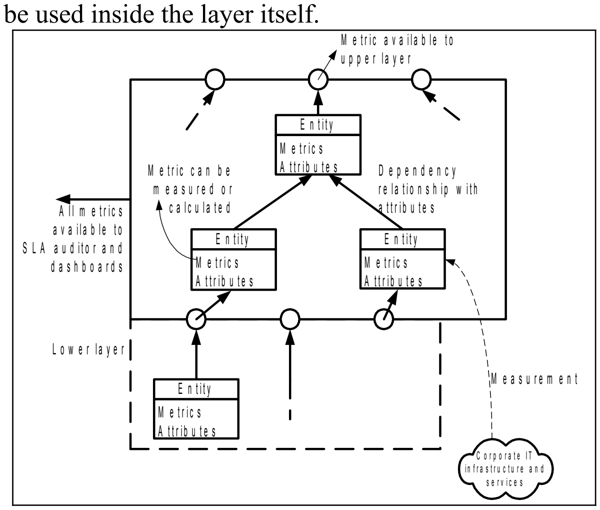

Figure 2: General layer organization

Entities are organized as a hierarchical structure (also called a *Composite*) using the notion of dependency. An entity has several children, or *antecedents*, on which it depends. The relationship "A depends on B" has the following semantics: the performance of entity A depends on the performance of entity B, entity A knows that entity B is an antecedent and entity A can access entity B's attributes and metrics. Dependencies themselves can have attributes; a possibility that will be discussed later on is the attribute *importance* which may be used to tag an entity's antecedents with their relative importance.

All entities must produce at least one metric but may produce more. Typically, an entity produces a *health* metric that captures entity status (up/down) and performance degradation of the entity. Some entities provide metrics for the whole layer and these are available to the upper layer (together with the identification of the entity that produced it).

We now proceed to describe each layer in more detail, from bottom to top.

3.3 The IT Component Layer

In this layer, entities are basic IT components such as routers, switches, host computers, network links, etc. All components in this layer must provide a *health* metric, a value between zero and one. The value 'zero' means that the component is down or has unacceptably bad performance and the value 'one' means that the component is working and has good performance. It is common to use *status* or *availability* to qualify an IT component. We prefer the health metric for several reasons. First, status is binary and can't capture performance degradation; second, availability can only be measured over a period of time whereas health is an instantaneous metric, as requirement 6 necessitates.

For individual IT components, health can be measured (or calculated from measurements as shown below). For composite entities with antecedents, health is calculated as a function of the antecedents' health. The framework does not specify particular functions for calculated metrics. For a composite entity, it is typically the function "worst health among antecedents". Several examples will clarify how a health measure can reasonably be obtained for a variety of simple and composite components.

Basic component with queuing. Many IT components suffer performance degradation with increasing load as a result of queues forming inside the component. Examples include routers, switches, host computers, databases, network links, etc. For simplicity, let us first consider the case of a single queue; more complex examples will follow. In this case, one can evaluate the health of the component as a function of the "server utilization" where the server is that for which a queue forms. For example, in a host computer, processes queue up to access the CPU (the server). Health can be inferred from the server utilization, if one examines Figure 3. The figure shows how response time (performance) degrades with server utilization. Response time is good up to about 70% utilization, degrades significantly by 85% and becomes unbearable thereafter. As a result, we suggest that a utilization measure be converted to a health measure according to the following mapping function, where h is health and ρ denotes utilization:

$$h = \begin{cases} 1 & if\ \rho \le 70\% \\ 0.5 & if\ 70\% < \rho \le 85\% \\ 1-\rho & if\ \rho > 85\% \end{cases} \qquad \text{Equation 1}$$

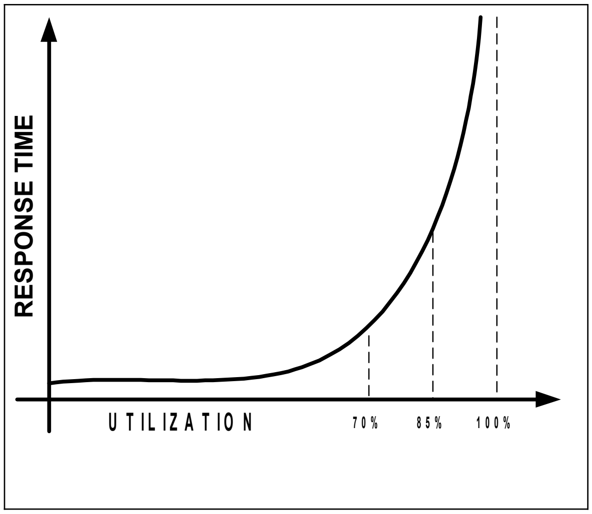

Figure 3: Response time versus server utilization

Router or switch. In the case of a router or switch component, two subcomponents affect health: the internal CPU and the memory. The health of the CPU subcomponent is given by Equation 1. For memory, one may also use memory utilization (since there is no virtual memory) or one may use the packet drop rate with health reaching zero when, say, 1% of packets or frames are dropped due to lack of memory. The overall health of the router or switch is the worst health between CPU and memory.

Host computer. A host is healthy if four subcomponents are healthy: CPU, memory, disk and network. The health of three of these can be obtained from Equation 1: CPU (from CPU utilization), disk (from the highest utilization of any physical disk) and network (from network interface utilization). Memory is special, due to the effects of virtual memory. A good way of measuring health for memory is to examine the page-out activity. Using today's disk technology, health can have value 1 if up to about 5 page-out operations occur per second, on average. Thereafter, health decrease linearly with page-out operations until it reaches zero with say 20 page-out operations per second, on average. Observe that we are referring to page *operations* and not *pages* brought out, since a single page operation typically pages out several pages at a time. The overall health of the host computer is the worst health between all four subcomponents.

Database. A database is made up of two subcomponents: the host and the DBMS software. Health for the host has already been discussed. Health for the DBMS component can be simply 1 if the DBMS is up and 0 if not. We do not have to consider degrading the health of a slow DBMS since this will be done at the next layer (IT Services). Once more, the overall health is the worst among subcomponents.

Network. A network is a complex entity and is not used in its entirety by any single service. What is actually used by a user accessing an IT service is a *network path* and this is what we choose to model. The whole network is modeled as a list of network path entities. Each path corresponds to a user-community/server pair. For example, suppose that a user community in New York uses a database server in San Francisco. We then model that combination as an entity and characterize health through the response time between that user community and the server. The mapping function between response time and health is discussed in the next section. Of course, this assumes that response time can be measured with the instrumentation used by the enterprise. If only link utilizations are being measured (a more common state of affairs), an alternative is to consider the health of the network path as a function of the worst link utilization on the path and use Equation 1.

In all the above case, it is important to note that, although only the health metric is fed into the next layer, the basic metrics (utilization, page-out activity, etc.) can be brought out to a dashboard and can be part of the drill-down model, as we discuss further below. We prefer to feed a single health metric into the next layer for simplicity (requirement 4).

3.4 The IT Services Layer

In the IT Services layer, entities are applications or other final services provided by IT, such as mail, a web service, etc. IT services may depend on other ancillary services; for example, an application may depend on mail service which in turn depends on DNS. In this way, a hierarchy of services is the main structure exhibited by entities in this layer. The bottommost services depend directly on IT components from the previous layer; these are the components used to support the service.

As before, the main metric used here is health, a value between zero and one that must capture the performance degradation of a service. Two situations must be considered: at times health can be measured and at others it must be calculated. In the second case, a service's health is a function of the health of antecedents. This can be any function but is typically the "worst health among antecedents", as seen previously. When an entity's health is based on measures performed directly, then one may ask of what use may the health metrics of that entity's antecedents be? The answer is that, even though the antecedents' health metrics are not used in calculating the entity's health, they are useful when drilling down to find cause-and-effect relationships.

Let us consider what we believe would be the most common situation: health is calculated from the *measured* response time offered by an IT service. How can one map response time to health? It is common knowledge in the performance management community that user services should aim for a response time lower than 100 ms, since this is the value below which the user thinks the service is "instantaneous". From 100 ms up, health goes down until response time reaches a certain maximum, say 5 seconds, at which point health is considered to be zero. The drop in health between these two points is not linear but must follow a function similar to that shown in Figure 4. This particular function is defined by the following equation, where t is response time and h is health.

$$h = \begin{cases} 1 & t \le 100ms \\ 0{,}7 & 100ms < t \le 300ms \\ 0{,}5 & 300ms < t \le 700ms \\ 0{,}3 & 700ms < t \le 2s \\ 0{,}1 & 2s < t \le 5s \\ 0 & t > 5s \end{cases} \qquad \text{Equation 2}$$

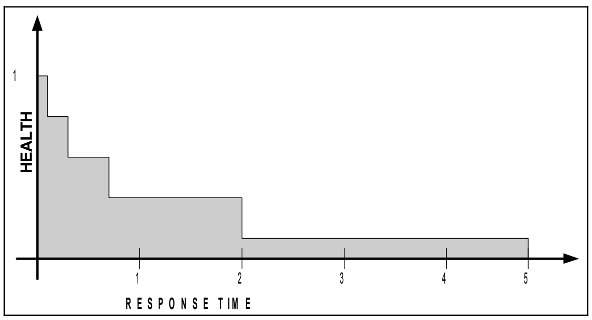

Figure 4: Response time to health mapping

We conclude this section with three comments regarding the application of this model to real-world situation.

Database application. In this case, one must clearly differentiate between database transactions and business transactions; the latter is a single operation from the user's point of view although the application may map into a set of several database transactions. One can measure the response time for a business transaction using, for example, Oracle's SQL trace facility, as long as one knows a typical sequence of SQL statements for that business transaction. Thus the response time for the whole business transaction can be obtained and transformed to a health metric according to Equation 2.

Aggregate response time. It is frequently possible to measure not only the total response time for an IT service but also the component times that add up to this total response time. For example, the database application response time can be broken down into CPU time and queue time, with this last further subdivided into sequential disk access time, direct disk access time, buffer wait, latch times, etc. When this is done, it is not necessary to model this hierarchy of metrics as entities in the model; the metrics themselves, if hierarchical in nature, can be used to assist a drill-down operation.

Geographical view of IT services. We expect that managers may wish the drill-down model to provide a geographical dimension. For example, saying that the mail service has health 0.7 does not mean that all users of this service experience the same performance. Users located in Brasília may experience good service while users in Ottawa experience large delays. This can be modeled simply by splitting the IT service into several sub-services, and to make a particular service (say Ottawa mail) dependent on the appropriate IT components, including the correct network path (refer to last section about modeling the network through the concept of paths). We will return to this example in the next section to show how geographic considerations can be included in other layers and even take into account the user population present in each region.

3.5 The Business Process Layer

Business processes (BPs) are the entities in this layer. A business process is a group of business activities organized as a workflow and undertaken by an organization in pursuit of a common goal. Because of our requirement for low intrusion, we choose not to model the internal organization of a BP; at most, we allow a BP to be broken down into sub-processes, using the dependency relationship provided by the framework. Of course, the dependencies that a BP has on IT services from the lower layer must be discovered and this often will force analysts to model the workflow to some extent.

For modeling, it is thus sufficient to *identify* the BPs and their dependence on IT services. However, it is also desirable to know the *importance* of each BP to the business. This will be further discussed in the next section. One initial way of discovering the importance of a BP is to classify it as being an operate process, a support process or a manage process. An *operate* BP is a process that is part of the main value chain of an enterprise; the main operate BPs are typically Get Order (or Generate Demand), Develop Product, Fulfill Order and Support Product. A *support* BP exists only to support the operate and manage processes by providing human resources, financial, technology and learning support. Finally, *manage* BPs are concerned with managing the performance of the firm and also exist only insofar as operate processes exist [2].

Once again, in this layer, the basic entity metric is *health*: how well is the BP performing? It is crucial to note that we are only speaking of BP performance as affected by the IT services supporting them. A process may not be performing well due to non-IT reasons and it is not the aim of BIM to capture this (although it is an important part of general BPM). BP health must capture BP performance degradation. How this is done really depends on how much instrumentation is available. If a BPMS is available, such measures as end-to-end response time, number of trades performed, etc. can be leveraged to calculate health.[3] We prefer not to assume that a BPMS is available (for low intrusion) and assign a function to calculate the health of a BP from the health of its antecedents. Let us discuss two examples:

BP where all antecedents are necessary. Assume that a BP depends on several IT Services and that all these services *must* be operational for the BP to work satisfactorily. In this case, we can choose the function "worst antecedent health" as a BP health function or even "product of antecedent health metrics". In either case, a faulty IT service (with health metric equal to 0) would cause the BP to have health 0.

Widely geographically dispersed BP. We continue the example presented in the last section where the user population for the BP is geographically dispersed. During drill-down, we wish to provide a geographical dimension that will allow one to see where bad BP health is coming from. Recall that we can model IT services by geographic region. Suppose that we have a Get Order BP spanning two regions, Brasília and Ottawa (Figure 5). In Brasília, 14 people are using the Get Order Application while 6 people do so in Ottawa. Assume further that the Get Order Application is down in Ottawa (health is 0). Then we can calculate the final Get Order BP health as a weighted sum of antecedent health with the normalized population as weight. Thus, BP health is 0.7. Recall that the framework allows dependencies to have attributes; this is what we have done here: we have used normalized population as the dependency attribute and have used this attribute in calculating health. Of course, it is perfectly acceptable and frequently desirable, to use other types of weights, closer to business results: consider, for example, how well business priorities are captured if we use Brasília's and Ottawa's sales targets as weights.

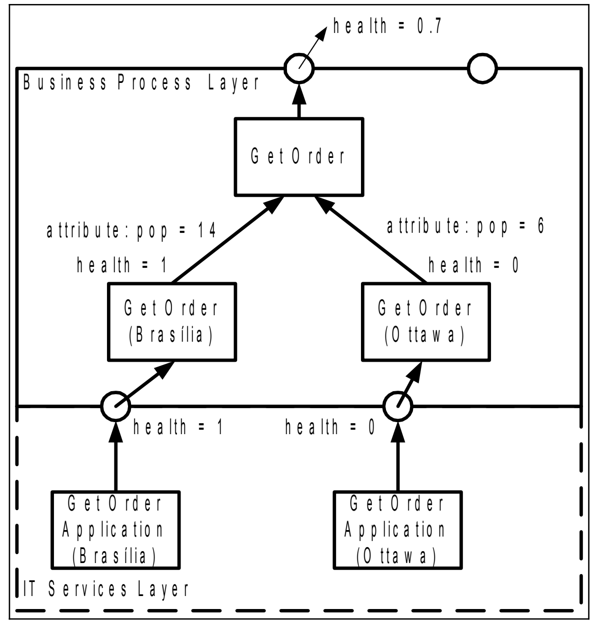

Figure 5: Health of a geographically dispersed business process

3.6 The Business Layer

This is the layer where couching metrics in business terms becomes important and where the model must capture the final effects of IT faults and performance degradation on the business. Entities are corporate entities such as the corporation, business units, or lines of business to which business metrics can be applied. Once more, these entities form a hierarchy that can be as small or extensive as desired. Figure 6 shows a situation with 2 business units.

Although one may calculate health as in other layers here also, this is not sufficient since health is not a business metric. We introduce *loss* as the basic measure captured by the business layer. Loss is simply a measure of the *negative impact* that bad BP health has on the business. How is loss calculated? For upper entities in the layer, loss is simply the sum of the loss metric over all antecedents. For lower entities, which have BPs as direct antecedents, loss is calculated as a function of the health of these antecedents. For perfect health, loss should be 0. Loss is actually a rate at which loss is instantaneously accumulated at any given instant in time. Thus, two important health metrics exist for any entity in this layer: (*rate of*) *loss* and *accumulated loss*. Accumulated loss is an important metric since it captures the adverse effects on business when IT faults or performance degradations are not resolved over a period of time; while rate of loss may remain constant over this period, accumulated loss will steadily worsen.

Clearly, the health-to-loss function is extremely important. As a first observation, we note that one must

[3] Observe that many authors call such measures "Business Metrics"; we prefer to call them Business Process Metrics. Business metrics will appear in the next layer.

capture the criticality of each business process to a business entity in order to calculate loss. Recall the discussion about operate/support/manage BPs in the last section. It is clear that important operate BPs can have a much larger impact on business loss than other BPs. We therefore include a *criticality weight* as an attribute of all dependencies between business entities and BPs (see Figure 6). The sum of all criticality weights over an entity's antecedents is 1. This criticality weight is taken into account when mapping from health to loss. It can be based on a BP's revenue-generating power, on the number of people using the BP, or on other estimates of the importance of the BP to the business.

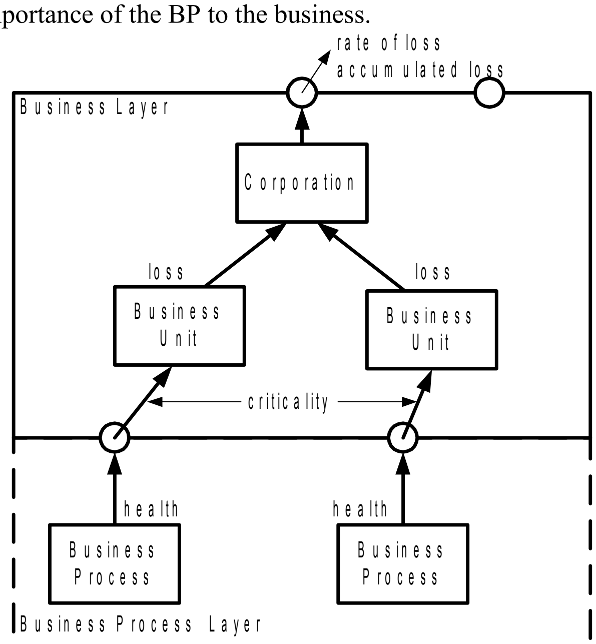

Figure 6: The business layer

Many functions may be used for health-to-loss mapping. Ideally, loss represents an absolute financial value directly interpretable by business executives. Examples are lost revenue and cost of productivity losses (due to stopped or slow business processes). However, even when accurate financial values may not be calculated, capturing relative values for loss is very important since relative metrics also have the important property of carrying *business priority information*, as is required in any BIM solution. Such relative values may be used by IT management to prioritize IT actions according to business priorities.

Example health-to-loss mapping. We provide an example of a health-to-loss mapping function applicable to a general business enterprise. The enterprise has *m* business units (BUs), B_1 *to* B_m. Business unit B_i generates hourly[4] revenue r_i. Let there be n_i BPs used by BU B_i, and let these processes be denoted by P_1^i to $P_{n_i}^i$. A business process, say P_j^i ,has the following attributes:

- A health measure, h_j^i ;
- A class: a BP may be an operate BP, a support BP or a manage BP (see above);
- The fraction the BP contributes to the revenue of the BU; this is assumed to be zero for support and manage processes; the total of 100% is distributed among the operate BPs. We denote this fraction by v_j^i. Note that $\sum_{j=1}^{n_i} v_j^i = 1$.
- The hourly fixed cost associated with the BP. We denote this value by c_j^i.

Now the (rate of) loss Li for BU Bi is simply the sum of lost revenue and lost productivity (as measured by the BP's fixed cost): see Equation 3.

$$L_i = \sum_{j=1}^{n_i} \left(v_j^i r_i + c_j^i\right)\left(1 - h_j^i\right) \qquad \text{Equation 3}$$

For the corporation, we sum over all BUs to get total loss, L:

$$L = \sum_{i=1}^{m} L_i$$

Accumulated loss is simply the time integral of the (hourly) rate-of-loss function over time and is an estimate of financial losses due the current state of the IT infrastructure.

This is still an unrefined calculation of loss, since it mixes revenue and costs and varies linearly with health. We are investigating better models.

3.7 Service-Level Agreements and Alignment Metrics

The metrics we have so far introduced, particularly rate of loss and accumulated loss, do serve the objective of capturing the impact of IT on the business from a business perspective. However, we have not yet mentioned Service-Level Agreements (SLAs) [19], which may be an important component of BIM. A further contribution of our work is to numerically capture IT-business alignment through SLAs. This discussion follows.

When IT managers meet business executives to understand business priorities, the end result is frequently expressed as SLAs. These are typically included at the IT service layer, using IT metrics such as response time and availability. In our model, we go further and allow the inclusion of SLAs at any layer, and using any of the metrics that appear in the model, be they IT metrics or

[4] It is more common to speak of the monthly revenue generated by a business unit. However, in the context of fault management, smaller time scales are more useful. We have chosen one hour as the basic time interval although any other convenient value can be used.

higher-level business process metrics or business metrics. Health and loss can also be used when expressing SLAs. Observe also that SLAs may also be used in drill-down operations since they have "antecedents", that is, the metrics used to check compliance.

We now propose new aggregate metrics that are particularly useful in checking *instantaneous* IT-business alignment. It may seem strange to the reader to discover that we base our instantaneous measures on SLAs, since SLAs are not themselves instantaneous but typically involve metrics averaged over a period of days or weeks. A basic BIM measure we introduce is the SLA *non-compliance risk* metric: how much is SLA compliance being compromised *now* by current IT service quality? In other words, we want to capture how much tougher attaining SLA compliance is getting with the current state of health metrics in the model. This metric is also a rate (like loss) and may be useful as a rate or aggregated over time. We do not call this index a business metric since it is too abstract and is not couched in business terms. Still, it may be a very valuable Key Performance Indicator since it reflects how *misaligned* IT currently is with business objectives, as expressed by SLAs. Observe that each Service-Level Objective (SLO) mentioned in the SLA is subject to its own SLA non-compliance risk metric.

The exact way in which the metric is calculated may vary. As an example, suppose that an SLA contains the following SLO: "Metric X, sampled every s minutes and averaged over a period of p minutes must be less than or equal to m". Over a period of p minutes, there will be $N = p/s$ samples taken. Let us say that t minutes ($t < p$) have elapsed since the beginning of the SLO evaluation period. At this time, $n = t/s$ samples have been taken and let the sum of theses samples be $S(t)$. These samples have already "compromised" SLA compliance by $S(t)$ units out of the total of Nm units that may be accumulated over the averaging period and still maintain compliance. How can we represent how bad this situation is? We show two alternatives:

First index: $$A = \frac{S(t) \cdot p}{Nm \cdot t}$$ Equation 4

In this case, the index assumes value less than 1 when SLA compliance is not compromised; in other words, so far, over t minutes, the average to be maintained over p minutes is being maintained; when the index goes above 1, the average being maintained so far, if kept over the whole averaging period, will make the metric overshoot the maximum value (m) to $A\,m$.

Second index: $$A = \frac{m}{\dfrac{Nm - S(t)}{N - n}}$$

Equation 5

Let us examine the equation by parts. The term $Nm - S(t)$ is the total number of units of the measure that may still be accumulated until the end of the audit period and still maintain compliance; thus, $(Nm - S(t))/(N-n)$ is the average for the metric that will have to be maintained from now on and still maintain compliance. Finally, the value of A captures how much better the new average to be maintained needs to be compared to the desired average (m) in order to maintain compliance. For example, when the instantaneous measure $A = 5$, an average metric 5 times better than the desired value needs to be maintained until the end of the audit period. A negative value indicates that compliance is impossible. It therefore clearly captures "how bad things are" as far as SLA compliance is.

If every SLO in an SLA has such a metric, an overall *Non-Compliance Risk* can be calculated by using a weighed average, each SLO receiving an appropriate weight according to its importance.

4. Results and Validation

This section discusses the methodology used to validate our work and discusses preliminary results obtained to that effect. Our validation methodology rests on answering two questions: Have we satisfied the requirements? Are results promising?

Have we satisfied the requirements? Let us take up the requirements one by one (see section 2) and verify how well we have satisfied them with the model presented here.

1. **Business Impact**. The model produces business metric showing the (negative) impact (loss) due to IT component faults and performance degradation. The model takes into account the importance of business processes to the business and can therefore represent business priorities. Furthermore the SLA non-compliance risk metric is based on SLAs which, by definition, capture business priorities.
2. **Drill-down**. The hierarchical structure of the model allows drill-down and roll-up operations to be performed. Specifically, it is possible to drill down within a layer and also from layer to layer using dependency relationships. Additionally, one may start at the SLA, drill down through SLOs and into the causes of non-compliance risk. Finally, some measures (host health, for example) may themselves be organized hierarchically and depend on other measures (e.g., CPU health, memory health, disk health, network interface health), allowing for drill-down operations within a particular metric. The model can cater to many stakeholders: some layers are meant for the business executives while others are more appropriate.

3. **IT measurements**. IT entities on which measurements are performed are explicitly represented in the model.
4. **Low intrusion.** Since we have produced a framework, many models can be instantiated, some simpler than others. Simple models are certainly possible and require little modeling; as an example, it is possible to include BPs without detailing the complete workflows. The framework does not require BP execution data (it does not require a BPMS), although such data may be used if available. We do believe, however, that more experience with the model will be necessary before we can pass final judgment on this requirement.
5. **Flexibility**. No entity is obligatory and entities can be freely introduced in the model. Furthermore it is even possible to remove a whole layer if the modeler can map metrics between the remaining layers.
6. **Operations model**. The main metrics involved in the model all represent an instantaneous state. These metrics are health, loss and SLA non-compliance risk.
7. **Framework**. The model entities are generic, the metrics are generic, the mapping functions are generic, and what is measured and what is calculated can be chosen by the modeler. Observe that having a "health metric" or a "loss metric" is generic; how they are calculated is specific to an instantiated model.
8. **Business changes**. Weights on the dependencies present in the business layer and between this layer and the BP layer allow business priorities to be changed as frequently as desired.
9. **Base for strategic model**. Although we have not dwelt on this in the paper, we have actually produced an add-on to this model that captures long-term effects. It is based on analyzing the effects on the business of SLA violations using a Balanced Scorecard [11].

Are the results promising? We will need more time to answer this question. We will have to depend on the opinion of IT and business managers as to the usefulness of the framework and the models it produces. Objectively, we will need to baseline SLA compliance and verify that compliance improves with time by using the model. Furthermore, we can baseline business metrics and see if they improve with time. All of these steps take quite a bit of time to perform and validate.

5. Conclusions

What have we achieved? We have produced a framework for Business Impact Management that explicitly includes a business layer and generates business metrics and other high-level performance indicators. The framework provides flexibility in modeling the different enterprise instrumentation realities. We have also introduced new metrics to measure the alignment between IT and business; these are based on SLAs.

What are the consequences? The framework allows low-intrusion BIM solutions to be built. Cost of acquisition and deployment significantly hinders the adoption of BIM and BPM solutions [8] and one of the main objectives of our work is to lower the entry barrier for BIM. The use of the framework allows IT management to prioritize IT actions based on business impact. IT staff can drill-down to focus on highest-return problems. On the negative side, one must say that a framework is not, in itself, a complete model but must be instantiated to be useful; this requires some work and we will need more investigation to decide whether we have struck the right balance between flexibility and ease of use.

What will we do in the future? Future investigations will occur in four directions. First, the framework needs more validation. We must use the framework in many situations to verify the balance between flexibility and ease-of-use, to check whether it can capture business priorities well, and whether the proper use of the framework really can help to improve IT quality.

Secondly, the framework itself can be expanded in many ways. We believe that more work is needed to try to capture business metrics, especially of a financial nature. To that end, we will try to factor into the framework considerations about regulation, responsiveness, revenue growth, revenue efficiency (profit), and risk management. It is not clear that the model can handle entities external to the enterprise in a realistic fashion and this will have to be investigated. Finally, the framework does not yet consider highly dynamic infrastructure (grids [9]) where dependencies can change over a very short period of time.

In a third direction, we will pursue new low-intrusion avenues. For example, we want to refactor the framework to explicitly include a core framework applicable to all enterprises, an industry segment part which would be the same for any enterprise in a particular segment and an enterprise-specific part. This would allow more reuse between models and would diminish the amount of work needed to create models.

Finally, this work has considered *operations* BIM. The model must be expanded to capture long-term strategic effects; a Balanced Scorecard (BSC) approach is likely to be useful here.

This work was developed in collaboration with HP Brazil R&D.

6. References

[1] Barnett, M.W., “Modeling & Simulation in Business Process Management”, *Gensym Corporation*, 2003, 10 pp.

[2] Bititci, U.S., “Managing with Measures – Using Performance Measures to manage the Performance of your Business”, *Centre for Strategic Manufacturing, University of Strathclyde*, Glasgow, UK, 12 pp.

[3] BMC Software, “Align your IT to your Business”, *White Paper, online*, http://www.bmc.com

[4] Curbera, F. et alli., “Business Process Execution Language for Web Services Version 1.0”, http://www.ibm.com/developerworks/library/ws-bpel/, julho 2002.

[5] Thiagarajan, R.K., Srivastava, A.K., Pujari, A.K, and Bulusu, V.K., “BPML: A Process Modeling Language for Dynamic Business Models”, *Proceedings of the 4th IEEE Internationalk Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWS 2002)*, 1530-1354/02, 3pp.

[6] Casati, F., Shan, E., Dayal, U., and Shan, M-C.,. “Business-Oriented Management of Web Services”. *ACM Communications*, October 2003.

[7] DeFee, J.M., Harmon, P., “Business Activity Monitoring and Simulation”, *BPT White Paper*, www.bptrends.com, Feb. 2004

[8] Dubie, D., “Never-fail business services”, *Network World Fusion*, online http://www.nwfusion.com/buzz/2002/bim.html, September 2002.

[9] Foster, I., Kesselman, C., “The Grid 2: Blueprint for a New Computing Infrastructure – 2nd Edition”, *Morgan Kaufmann*, November 2003, 748 pp.

[10] Van Grembergen, W., and Saull, R. “Aligning Business and Information Technology through the Balanced Scorecard at a Major Canadian Financial Group: its Status Measured with the IT BSC Maturity Model”, *Proceedings of the 34th Hawaii International Conference on Systems Sciences* – 2001, 10 pp.

[11] Kaplan, R., and Norton, D., “The Balanced Scorecard: Measures that Drive Performance”, *Harvard Business Review*, 70(1), 1992, pp.71-79.

[12] Keller, A., and Ludwig, H., “The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services”, *IBM Research Report, Thomas J. Watson Research Center*, May 2002, 21 pp.

[13] Mason, P., “A New Culture for Service-Level Management: Business Impact Management”, *IDC White paper, online*, http://www-900.ibm.com/cn/software/tivoli/products/download/analystsreports/ar-service-level-mgt.pdf, 2002.

[14] McGregor, C., “A method to extend BPEL4WS to enable Business Performance Measurement”, *School of Computing & Information Technology, University of Western Sydney*, Locked Bag 1797, Penrith South DC, NSW 1797, Australia, Technical Report No. CIT/15/2003, 6pp.

[15] zur Muehlen, M., and Rosemann, M., “Workflow-based Process Monitoring and Controlling”, *Proceedings of the 33rd Hawaii’s International Conference on Systems Sciences*, 2002.

[16] Nelson, K. M., Nelson, H. J., Armstrong, D., and Richardson, V. J. “Creating Business Value through Business-IT Alignment, Shared Knowledge, Commitment and Accountability”, *37th Hawaii International Conference on System Sciences (HICSS’04)*, 0-7695-2056-1/04 IEEE Computer Society, 2004, 1pp.

[17] Rainge, E., “Maximizing the Value of IT Investments Through Business-Aware Performance Management”, *IDC White paper*, January 2004. http://www.bitpipe.com/detail/RES/1075811758_780.html

[18] Smith, H. and Fingar, P. “Business Process Management: The Third Wave”, *Meghan-Kiffer Press*, December 2002, 311 pp.

[19] Sturm, R., Wayne, M., Jander, M., “Foundations of Service Level Management”, *Sams*, 2000.

[20] Verner, L., “BPM - The Promise and the Challenge”, *ACM Queue Magazine*, March 2004, pp. 83-91

[21] Zhang, L-J., Chang, H. and Chao, T., “Web Services Relationships Binding for Dynamic e-Business Integration”, *IBM T. J. Watson Research Center Report*, 2002, 6 pp.

[22] The Mercury Interactive Corporation, “Topaz for SLM - The Solution for Business-Centric Service Level Management (SLM)”, *online*, http://www.mercuryinteractive.com/products/topaz/sm/, 2004