A Business Driven Management Framework for Utility Computing Environments

Issam Abi[1], Mathias Salle, Claudio Bartolini, Abdel Boulmakoul
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2004-171
October 12, 2004 *

E-mail: firstname.lastname@hp.com

utility computing, management by business objectives, service level agreement (SLA), service level management (SLM), policy based management (PBM)

In this paper, we introduce a Business Driven Framework for the Management of Utility Computing Environments (BDMF). The framework couples two main subsystems on top of an IETF-like policy-based resource control layer. They are MBO (Management by Business Objectives) where the decision ability supported by analysis of business objectives resides, and GSLA (Generalized SLA), an advanced framework for SLA driven management that lends itself quite naturally to the derivation of IT management policies from the SLAs that the enterprise has contracted. We discuss the advantages and the limitations of the state-of-art policy-based approach to systems management, mainly the lack of business and service level context to drive policy-related decisions at system run-time. We then explain how this is remedied in our framework through the interaction mechanism between the reactive policy-based resource control layer and the more proactive business driven decision-making engine.

* Internal Accession Date Only
[1]issam.aib@lip6.fr PHARE Group, LIP6 Laboratory, University of Paris 6, France
Approved for External Publication

A Business Driven Management Framework for Utility Computing Environments

Issam Aib[*]

PHARE Group,
LIP6 Laboratory,
University of Paris 6,
France

Mathias Sallé, Claudio Bartolini

HP Research Labs
1501 Page Mill Rd
Palo Alto, CA 94304
USA

Abdel Boulmakoul

HP Research Labs
Filton Rd, Stoke Gifford,
Bristol BS34 8QZ
United Kingdom

Abstract

In this paper, we introduce a Business Driven Framework for the Management of Utility Computing Environments (BDMF). The framework couples two main subsystems on top of an IETF-like policy-based resource control layer. They are MBO (Management by Business Objectives) where the decision ability supported by analysis of business objectives resides, and GSLA (Generalized SLA), an advanced framework for SLA driven management that lends itself quite naturally to the derivation of IT management policies from the SLAs that the enterprise has contracted. We discuss the advantages and the limitations of the state-of-art policy-based approach to systems management, mainly the lack of business and service level context to drive policy-related decisions at system run-time. We then explain how this is remedied in our framework through the interaction mechanism between the reactive policy-based resource control layer and the more proactive business driven decision-making engine.

Keywords*: Utility Computing, Management by Business Objectives, Service Level Agreement (SLA), Service Level Management (SLM), Policy Based Management (PBM).*

1 Introduction

Over the years, the Information Technology (IT) function has gained an increasing strategic role in the enterprise as it is becoming the backbone of businesses to the point that it would be impossible for many to function, let alone succeed, without it [5]. Lines of business are increasingly seeking to consolidate and to generate higher return on IT investment by sharing a common, agile and highly adaptable infrastructure run by the IT function.

Utility Computing (UC) is a paradigm where shared infrastructure can be provided on demand to multiple applications [13]. IT resources in a UC (compute-power, storage, network bandwidth, etc.) service are provisioned on a per-use basis in the same manner as a public Electric Energy service is used. Traditionally, IT resources are allocated in a dedicated manner. Capacity planning technology determines the optimal amount of resources to be provided based on some analysis criteria (average usage, peak usage, expected usage growth, etc.). However, practice has shown that not only it is almost impossible to fully predict applications demands in terms of IT resources but also that demand generally varies dramatically in time.

In this context, IT management solutions take on a new crucial role. With the increasing number, complexity and frequency of IT related decisions, the mechanisms to determine the optimal utilization of IT resources must become an integral part of automated IT Operations. Given the

[*] Work done while author was intern at HP Research Labs, Bristol, UK.

timescales involved, the decision making process has to be implemented through management policies whose objective is to maximize the business value of the services offered by IT. This has to be done while keeping the cost of eliciting knowledge about the business value of the services acceptably low. Therefore any piece of information that is available within the enterprise that enables assessing the business value of the IT services is a very precious source of knowledge. Examples of such information are the Service Level Agreements (SLA) that the enterprise has contracted, and other ways available of representing strategic and tactical business objectives of the enterprise in such a way that it is possible to understand the dependencies of the business objectives from measures taken at the IT level. Examples of such information can come from objectives defined in the enterprises IT balanced scorecard [22].

At the same time, traditional policy-based management (PBM) promises to reduce IT costs while simultaneously improving quality of service and adaptability to change [18]. Research in policy-based management systems in various applications areas including networking, security, and enterprise systems has been going on for about a decade [19], although they are still struggling to make their way into industrial applications. This is partly understandable as policy-based management represents a paradigm shift and there are a number of economical, political, and social considerations to deal with before it becomes widely used [20].

In fact, however successful an enterprise may be with its adoption of policy-based management, it must be remembered that the IT infrastructure of the enterprise is finalized to the provision of a service which is exchanged for economic value. Therefore, it is extremely important to make the policy-based management capability aware of business level considerations.

This consideration is central to our approach to defining a management stack that at a first level of detail is neatly separated into a business driven management Layer and an underpinning policy-based resource control layer (RCL), to which the former offers support and business context.

The main components of the business driven management layer are based on two technologies that we have developed: Management by Business Objectives (MBO) [5][21] and Generalized Service Level Agreements (GSLA) [2]. MBO is a proactive and business oriented decision making engine offering a high-level reasoning over the IT system state based on high-level business oriented data such as current business objectives and SLAs states. GSLA is our approach to SLA modeling that links service quality specifications to the policy based management layer.

The main novelty brought about by this work is positioned at the business driven management layer and its interaction with the UC PDP. The resource control layer is described here for completeness and is centered on policy based management technology that is currently state-of-the-art.

Although the framework we propose is more widely applicable, in this work we describe its application to utility computing environments. The framework helps achieve closed-loop management of a UC environment by (i) deriving low level management policies form SLAs and business context; and (ii) coupling a reactive policy-based system with a proactive business driven reasoning engine.

The paper is structured as follows. Section 2 will describe the BDM framework. In section 3, we describe the BML layer of BDMF in more detail focusing on the way we model SLAs and Business Objectives; as well as a detailed description of the MBO engine. Finally, we develop a use case showing how the BDMF succeeds in aligning utility-computing management with utility provider business objectives. The paper then concludes by an analysis of related and future work.

2 Business Driven Management Framework

The main objective of the business driven management (BDM) framework is to drive the management of IT resources and services from the business point of view. Most of the times, when tradeoff kind of decisions are to be made, the IT managers have a feeling for which is the option

available to them that guarantees the minimum cost or least disruption to the service. But unless the impact of carrying out the chosen course of action onto the business layer is understood, one may run the risk of solving the wrong problem optimally. Because of this the BDM framework was designed according to the principle of making information that pertains to the business visible from the IT and vice versa.

As presented in Figure 1, the BDM architecture is divided into two main layers. On the top is the business management layer, which is intended to host the long term control of the UC infrastructure based on the business objectives and market strategy of the utility provider. Beneath it is a resource control layer that hosts the real time logic for the reactive, short term control of the utility computing infrastructure.

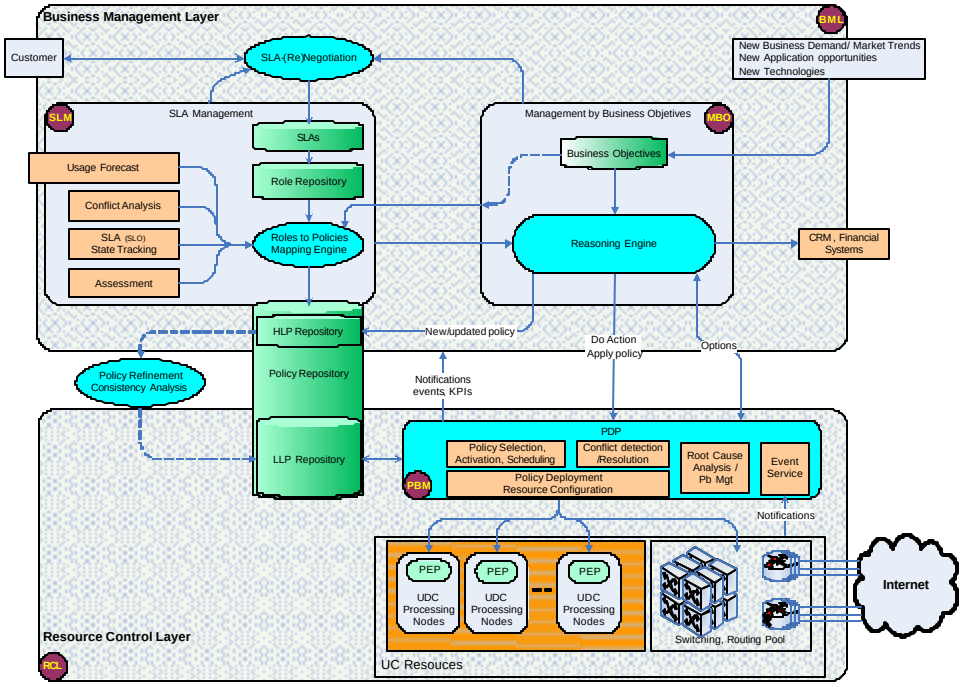


Figure 1: The BDM Framework

The Business Management Layer is responsible for optimizing the alignment of the UC resources usage with the objectives of the UC provider. The reasoning is based on a set of business objectives defined and audited over relatively long periods of time (monthly, quarterly, etc.). Business objectives are the reflection of the utility provider's business strategy and range over diverse key performance indicators, which can be related to service operations, service level agreements, or any other business indicators. Examples of business objectives are given below:

- Customer satisfaction should be improved to at least 70%.
- Customers with revenue greater than $xxxx should be classified as Premium Customers.
- Lowest importance (class 5) should be given to contracts related to services A, B, ..., C.
- Class 5 Services should not consume more than 20% of the UC resources.
- All SLAs related to Service K must operate at highest security and confidentiality levels (top confidential).
- Assign priority level 2 (important) to "Customer Satisfaction" Business Indicator.

Example 1: Sample Business Objectives

Business objectives are not defined once and for all, but often change due to multiple factors. Examples of factors affecting business objectives are: feedback from the assessment of all SLAs that the utility provider contracted over a given period of time; the internal prediction of the future

business relationships (number and type of customers, services gaining more interest, those getting back in the market share, and expectations on the business demand curve)

The business relationships contracted by the utility provider are formalized by SLAs and modeled using the GSLA information model introduced in [3]. Using the GSLA, each contracted service relationship is modeled as a set of parties playing an SLA game in which each party plays one or more roles to achieve the SLA objectives. Each role in the SLA is associated with a set Service Level Objectives (SLOs) to be achieved; as well as a set of intrinsic policies related to the role behavior per see. A special engine translates Roles, SLOs and rules into a set of enabling policies. These policies are further refined to lower level policies (LLPs) that enclose all the low level logic required to drive the UC resources as required by the business objectives and SLAs.

Business objectives affect the way SLAs are defined and managed at the resource control layer. So whenever a business objective is changed, added, or removed, important impact taking place at the long term time scale will be noticed on the SLA database.

Low level policies are dealt with by the Policy Decision Point (PDP) module [8][14] of the resource control layer. Part of the PDP`s task is to monitor and respond to system events and notifications by selecting, activating, and scheduling the enforcement of the appropriate policies at the appropriate UC resources. The PDP contains also sub-components for policy run-time conflict detection, root cause analysis, generation of the set of options available in the presence of some incident or problem, as well as a component dedicated to generate the appropriate configuration flow in order to enforce an active policy.

As it is impossible to define policies upfront to cover all run-time events, it will happen that LLPs may not be sufficient or there might not be specific LLPs to deal with certain conditions (servers going down, network pool congestion, etc.). In those cases, it is necessary to take a holistic approach to the situation and the PDP will pass the control to the BDM to deal with it. Given the various options, the BDM will select the one that will maximize the value to the utility provider. That is, the option that will result in the closest alignment to the business objectives. Such interactions offer the opportunity to learn and refine the policy definition process.

Finally, as external factors change (demand in the market place, introduction of new hardware in the UCE, etc.), policies might need to be modified. These changes are triggered by the Business Management Layer and propagated down to the Resource Control Layer ensuring a business driven management of the lifecycle of the LLPs.

In the remainder of this paper, we concentrate on the Business Management Layer of the BDM framework by focusing on our key technological contributions and illustrating them in a Utility Computing specific use case.

3 Business Management Layer of the BDM Framework

The Business Management Layer of the BDM framework is responsible for (i) managing the lifecycle of the SLAs contracted with the utility provider customers; (ii) managing unexpected events by maximizing the alignment to the utility provider business objectives; and (iii) deriving low level policies that will ensure compliance to the contracted SLAs and business objectives.

As depicted in Figure 1, the BML is architected around two key components, the SLA Management (SLM) and the Management by Business Objectives (MBO) components. The interactions between the SLM and MBO modules are twofold. Firstly, the business objectives are used by the roles to policies mapping engine during the policy derivation process. For example, if a business objective states "*achieve high availability levels (>98%) for Service S*", then the policy generation engine will override during the derivation process the SLOs in which service S is guaranteed with an availability of less than 98%. Finally, changes in the business objectives are reflected by the introduction or update of existing management policies.

3.1 Management by Business Objectives (MBO)

The Management by Business Objectives reasoning engine solves the following decision problem: it computes the *alignment* to objectives that is expected for each of the possible given *options*, or course of action aimed at managing the IT delivery systems. The engine is able to monetize the measure of alignment thus derived and use the monetization value together with other information on the cost of carrying out the respective course of action to rank the available options. On ranking the options, it 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.

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.

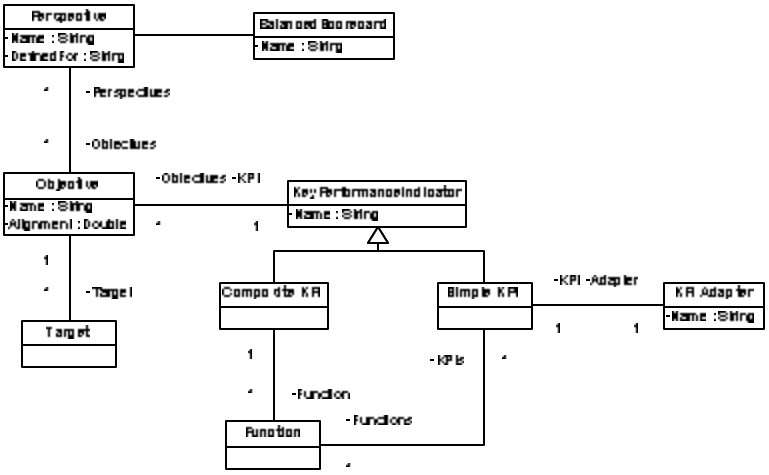
For instance, in the incident management domain, MBO has been applied to the problem of prioritizing among concurrent service incidents based on their impact on business objectives [5]. In that context, each option is a possible assignment of a priority value to the incidents. The prioritization that is finally chosen is the one that guarantees the optimal alignment with objectives that were propagated down from the business level, such as maximization of profit, or maximization of total customer experience (TCE), defined as a function of some key performance indicators (KPI). Knowledge about the domain is necessary to assess the impact of the incidents onto the value of the KPIs.

Although this can vary for different IT management domains, the timescale at which MBO works tends to be of the same order of magnitude as the IT decisions that require humans in the loop. Depending on the domain, that can be of seconds, minutes or even hours. The timescale is therefore much longer than the one at which the PDP works, which has to automatically and reactively deal with arising situations that do not require human intervention.

A. MBO Information Model

Because of the wide applicability of the MBO framework described above, it is a requirement for our underpinning information model to be as generic and as flexible as possible in order to be used in these various domains.

The MBO information model (Figure 3) 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 (Common Objectives for Information and related Technology [9]) framework and from Balance Scorecard [15]. 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.


Perspective
- Name : String
- DerivedFor : String
Balanced Scorecard
- Name : String
- Perspectives
- Objectives
Objective
- Name : String
- Alignment : Double
- Objectives -KPI
Key Performanceindicator
- Name : String
- Target
Target
Composite KPI
Simple KPI
-KPI -Adapter
KPI Adapter
- Name : String
- KPIs
- Function
- Functions
Function

3.2 SLA Contract Model

Service Level Agreements are modeled using the Generalized SLA (GSLA) model introduced in [2]. The GSLA is defined as a contract signed between two or more parties relating to a service relationship and that is designed to create a clear measurable common understanding of the role each party plays in the GSLA. A party role represents a set of objectives and rules which define the minimal service level expectations and service level obligations it has with other roles and at which constraints.

A GSLA contract can be viewed as a set of *parties* that come together according to a certain *schedule* in order to realize the contract each by playing one or more *roles*. During the GSLA life cycle, a required behavior or constraint related to a GSLA role is captured in the model through the abstract *GSLAPolicy*. A *GSLARole* inherits indirectly from the GSLAPolicy. This is to catch the idea that a role is modeled at first approximation by a set of Rules. The *Schedule* component represents the temporal scope during which a GSLA component is valid. A Schedule class is a specialization of a Constraint. A *Constraint* is an abstract class intended to capture any type of logical predicates over parameters of GSLA components. Finally, a GSLA is related to one or more *Service Packages* to each of which is associated a *Service Package Objective* that some GSLA party is required to guarantee as is specified in the role(s) it is related to.

A *Service Package* (SPg) is composed of a set of one or more *Service Elements*, each of which is related to one or more Service Resources. An SPg represents a group of related Service Elements that are instantiated and managed as a whole and/or are offered altogether to customers.

```
<SPG template="Internet Services Package">
<SE template="Web Server">
   <SR>VMWS422</SR> ": This information is normally not viewed by the customer.
</SE>                          "The UP added it to when considering the SLA instantiation."
<SE template=" Email Server"> <SR>VMES134</SR> </SE>
<SE template="FTP Server"> <SR>VMFTPS221</SR> </SE>
</SPG>
```

Example 2 : An Internet Services Package

To each offered service is associated an expected run-time quality as is promised by the Service Provider and as should be experienced by the service customer. Service quality is captured through a set of Service Level Objectives (SLOs). The modeling of SLOs is always faced with the tradeoff between Customer facing QoS parameters and Provider facing technical QoS spread within technical details related to service resources . We propose a modeling that bridges both QoS levels.

Considering Figure 3, the Service Package Objective (SPO) component defines an SLO over parameters of a SPg. An SPO is mainly a constraint which can be defined in two different ways. First, it can be a set of predicates or logical expressions over one or more SPg Parameters. SPg parameters are synthesized (calculated using aggregation functions) from basic System Metrics up through System Resource (SR) parameters and System Element (SE) Parameters. The other way around is to define an SLO as a compilation of QoS appreciations coming from subordinate SLOs. This second approach in appreciating the overall Service Quality reflects better the way users appreciate a given service infrastructure, i.e, by giving a final appreciation based on separate 'sub' appreciations over the different service components.

In the GSLA information model, multiple party service relationships are supported and each party has a set of SLOs to assure and some behavior to follow with respect to the other parties. Also, to each SLO are normally associated policies that specify actions to take in case the SLO has not been respected or some warning-level has been reached. Policies are also generated by a role (i.e the party responsible for the role) for the enforcement of its SLOs. Such enforcement policies are only viewed by that party and need not be specified at the common SLA unless explicitly requested by the concerned service customer party.

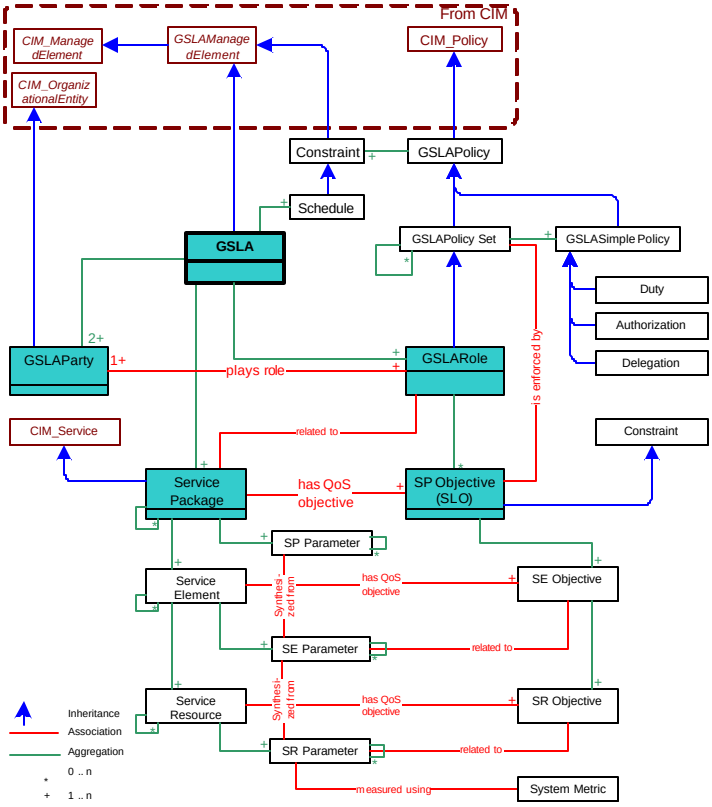


Figure 3: The GSLA Information Model

The behavior of a party is ultimately modeled through policy [1]. A GSLARole is modeled through a set of duty, authorization, and delegation policies, having their subject domain the party or the group of parties playing that role, as well as the set of SLOs that it is required to ensure as part of its responsibilities in the SLA [3]. A role contains two different types of policies: Role intrinsic policies and SLO enforcement policies. Role intrinsic policies are not linked to a specific SLO and are not a result of any SLO mapping. An interesting usage of Role intrinsic policies concern the constraint based scheduling of services, also named Service Package Graph in [6].

Examples of Role Intrinsic Policies
Send monthly report about availability and performance parameters of service X to the SC role.
IF ((terminated(Service1) or terminated (Service2)) and (paid(Service1, SC))) **THEN** Activate Service3
IF (failed (SLO) or failed (SLO_m)) **THEN** (SLOn.time += BONUS days)

4 An Incident Management Use case

To illustrate our work, we modify a use case from [4] in order to describe the combined use of policies, SLAs and business objectives to drive IT-related decisions. The use case shows the usage of MBO in the incident management IT management domain (3.1)

We assume an IT hosting infrastructure for an electronic outsourcing company EOS. As depicted in Figure 4, EOS hosts services for two companies: KPU and EDF. Both services are web-based and require Web Servers, Application Servers and back-end systems such as Databases and Enterprise Resource Planning Systems. KPU and EDF customers access these services by

connecting to the servers hosted onto EOS IT infrastructure. This infrastructure takes the form of a set of shared IT resources inside EOS utility computing infrastructure.

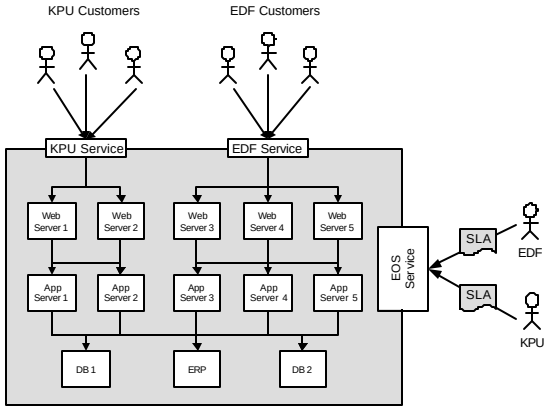


Figure 4: KPU and EDF hosted services running on EOS utility infrastructure

EOS's customers, KPU and EDF, are provided similar services but with different QoS assurances. These guarantees are negotiated prior to service deployment and can be renegotiated over time as EOS customer's service requirements evolve. We show, through a XML-like pseudo syntax (we avoid a fully XML notation for clarity), the parts of SLAs that are of concern for the use case:

<**GSLA** name=" **EDF- SLA**"> <id>..</id><schedule>…</schedule>	Description="EOS-EDF SLA"
<Party name="**EOS**">…</Party><Party name="**EDF**">…</Party>	
<**SPG**…>< **SE** template="Web-Server">	
<ServiceParameter name="Availability">…</ ServiceParameter>	Description="Offered services and their parameters"
<ServiceParameter name="Latency">…</ ServiceParameter>	
<ServiceParameter name="Capacity">…</ ServiceParameter>	
<ServiceParameter name="NbClients">…</ ServiceParameter>	
</**SE**>…</ **SPG**>	
<**Role** name="**EDF-Role**" SubjectParty=" **EDF**" Type="Mandatory">	
<Policy name="p1"> <param name="Charge" value="€0.1*Server.Capacity"/>	description="EDF is monthly charged
"**on** month **do** credit(EOS-Role, Charge)"</Policy>	€0.1*Server.Capacity"
</**Role**>	
<**Role** name="**EOS-Role**" SubjectParty="**EOS**" Type="Mandatory">	description="supported capacity is of 1000
<SLO name="SLO1"> Web-Server.Capacity = "1000" </SLO>	simultaneous connections"
<SLO name="SLO2">	
"Web-Server.NbClients<Capacity ==> Web-Server. Availability >= 99%"	description="Availability of the hosted service will be
<Policy type="duty">"**on** not(SLO2) **do** Credit(EDF, 0.1 * EDF.Charge)"</Policy>	99% at all time, Otherwise, KPU will be credited
</SLO>	20% of the monthly charge for the next pay period"
<SLO name="SLO3" >	description="Average time to process any customers
"Average(Web-Server.Latency, month)<=5ms"	service request over a month period, will be less than
<Policy type="duty" >	5 ms"
"**on** fail(SLO3) **do** credit(EDF, 0.05 * EDF.Charge * duration(fail(SLO3) / 30)"	description="Otherwise, KPU will be granted a 5%
</Policy></SLO>	credit of the period over which the breach occurred"
<Policy type="duty">	
"on not (EDF-Role.SLO1, 3) do TerminateContract(EDF-SLA)"</Policy>	description="if EDF fails to pay the monthly charge
</**Role**></GSLA>	for successive 3 months then the contract will be
	terminated"

Figure 5: EOS-EDF GSLA

<GSLA name="**KPU-SLA**"> <id>..</id><schedule>…</schedule> Description="EOS-KPU SLA"

<Party name="EOS">…</Party>
<Party name="KPU">…</Party>
<SPG>...</ SPG> description="KPU is monthly charged €0.15*Server.Capacity on return to service usage"

<**Role** name="**KPU-Role**" SubjectParty="**KPU**" description=" Availability of the hosted service will be 98%"
Type="Mandatory">...</**Role**>

<**Role** name="**EOS-Role**" SubjectParty="**EOS**" Type="Mandatory"> description=" Otherwise, KPU will be credited 20% of the monthly charge for the next pay period"

<SLO name="SLO1"> Web-Server.Capacity = "5000" </SLO>
<SLO name="SLO2" >…</SLO>
<SLO name="SLO3" > description="Service unavailability will be fixed within 4 hours of the receipt of a trouble ticket, otherwise KPU will be credited 3% of the monthly charge for the next pay period"
"Time(Web-Server.availability==true)-Time(Web-Server.availability==false) <= 4 hours"
<Policy type="duty">
on fail(SLO3) **do** credit(KPU, 0.03 * KPU.Charge)"</Policy> description="Average time to process any customers service request over a month period, will be less than 10 ms"
</SLO>
<SLO name="SLO4" >...</SLO>
<Policy type="duty"> description="Otherwise, KPU will be granted a 30% credit of the period over which the breach occurred"
" **on** not (KPU-Role.SLO1, 3) **do** TerminateContract(KPU-SLA)</Policy>
</Role></GSLA>

Figure 6: EOS-KPU GSLA

Although KPU and EDF SLAs are specified over two services parameters, availability and service latency, they each define different service levels and have different penalties. As new contracts are signed between the outsourcing company and its customers, the hosting IT infrastructure is further solicited: hosted service will share servers, bandwidth, disk arrays and sometimes applications, business processes, etc.

As a service provider, EOS operates under a set of specific business objectives which are captured within a balanced scorecard. Examples of business objectives are presented below.

Financial Perspective	**Customer Perspective**
· Increase revenue by 20% · Reduce SLA penalty cost by 30%	· Increase total customer experience by 10%
Growth Perspective	**Internal Perspective**
	· Decrease incident processing time by 15% · Increase application deployment time by 50% · Minimize spare pool · Achieve SLA compliance rate of 98%

Table 1: EOS balanced scorecard (Business Objectives)

EOS resource allocation strategy relies on the concept of policy-driven and business-driven entitlement. For each of the services deployed on the infrastructure, according to the SLA requirements, the usage forecast for the service and the capacity model, a set of policies is generated to control the EOS resources. The actual (run-time) allocation of those resources to the various applications composing that service is left to the PDP. This is referred to as *policy-driven entitlement*. The *business-driven entitlement* concerns the overall management of the entire resources of the infrastructure and is under the control of the Business Driven Management systems.

As previously explained, the process to generate low-level configuration policies that will enforce a certain GSLA Role or SLO is not straightforward. Let's assume here that based on local information related to EOS Web-Servers performance, its Roles to Policies engine generated the following set of high-level policies for the EOS-EDF role:

<parameter name="WSThreshold" value="80%">
on SLA.Schedule.startDate- 2 hours do Web-Server.installNew(Configuration c, Capacity 500)
on Web-Server.charge>=WSThreshold do Web-Server.installNew(Configuration c, Capacity 500) where (Web-Server.NbInstances * 500 <= Web-Server.charge
on not(SLO2) do Credit(EDF, 0.1 * EDF.Charge)
on fail(SLO3) do credit(EDF, 0.05 * EDF.Charge * duration(fail(SLO3) / 30)
on not (EDF-Role.SLO1, 3) do TerminateContract(EDF-SLA)
…

Given that an EOS web server resource instance can serve up to 500 clients without reducing the required QoS, we understand from the above policy set that EOS took the approach of provisioning *per-need* to meet its SLA with EDF. Whenever there is need, an additional web server instance is installed and provisioned for EDF customers. We assume that EOS took the same approach for mapping its EOS-KPU role.

The policy refinement engine then takes care of generating low-level policies within the LLP repository. It also registers required events within the PDP's event service and the PDP is informed about the LLP repository change.

Let's assume that EDF and KPU services reach the configuration of 1 and 4 web servers respectively and there are only two free resources that can be allocated to EDF or KPU services. Then, a sudden increase in the number of KPU and EDF customers is noticed leading to the following set of active LLP policies (WS21: virtual instance of the KPU web server):

(p1) on WS21.threshold do Web-Server.installNew(Configuration c, Capacity 500)	activated at time T
(p2) on WS21.threshold do Web-Server.installNew(Configuration c, Capacity 500)	activated at time T+ 5 s
(p3) on WS22.threshold do Web-Server.installNew(Configuration c, Capacity 500)	activated at time T + 7 s

The PDP is confronted here with a run-time policy conflict. In contrast to a static policy conflict [1], this is a type of conflict that cannot be predicted by the policy refinement process of the BML layer (Figure 1). To resolve the conflict, the PDP needs the assistance of the MBO for a wiser (business-driven) decision as a run-time policy conflict is generally synonym of service degradation that the PDP cannot measure. The PDP hence sends the following set of options for the MBO to decide which to apply.

- Active policies p1, p2, p3
- Available resources 2
- Time to install resource 1 hour, Servers can be instantiated in parallel
- Summary of the evolution pattern of the number of KPU customers
- Summary of the evolution pattern of the number of EDF customers
- Options set ((p1,p2), (p1,p3), (p2,p3),(p1),(p3),()).

The MBO engine will take into consideration service and business level parameters related to:

- EDF and KPU SLAs (current total time of service unavailability, time to recover from unavailability, penalty amounts) Expectation of the evolution pattern for the number of customers for EDF and KPU (to decide whether to allocate resources or just do not if the congestion period is temporary).
- Current customer satisfaction indicator value for EDF and KPU,
- Other Business indicators related to scorecards :
  - Reduce SLA penalty cost by 30%.
  - Increase total customer experience by 10%
  - Achieve SLA compliance rate of 98%

A utility value is generated for each option of the options set [5]; and then the MBO engine determines the option which maximizes the utility value and sends it back to the PDP for execution.

5 Related Work

Driving IT management from business objectives is quite a novel proposition. In [6], Buco et. al. 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 [21]. However, the Management by Business Objectives component of the Framework presented in this paper goes far beyond just using impact of service level violations. It provides a comprehensive method for IT management that can take into account strategic business objectives; thereby, going a long way towards the much needed synchronization of IT and business objectives. For a more detailed discussion of the MBO capability applied to the incident management domain see [5].

In another respect, the area of SLA-driven management has been closely interested in the subject of SLA modeling. WSLA, [10][11], from IBM research and WSMN [12][16] from HP Labs analyze and define SLAs for Web Services by building new constructs over existing Web Services formalisms (WSDL, WSFL, XLANG or BTP/ebXML, etc.). [16] specifies SLOs within SLAs and relates each SLO to a set of Clauses. Clauses provide the exact details on the expected service performance. Each clause represents an event-triggered function over a measured item which evaluates an SLO and triggers an action in the case the SLO has not been respected. In a recent work [4], we defined an FSA for SLA state management in which each state specifies the set of SLA clauses that are active. Transitions between states can be either events generated by an SLA monitoring layer or actions taken by parties in the SLA.

Keller A. and Ludwig H. [10][11] define the Web Service Level Agreement (WSLA) Language for the Specification and Monitoring of SLAs for Web Services. The framework provides differentiated levels of Web services to different customers on the basis of SLAs. In this work, an SLA is defined as a bilateral contract made up of two signatory parties, a Customer and a Provider. Service provider and service customer are ultimately responsible for all obligations, mainly in the case of the service provider, and the ultimate beneficiary of obligations. WSLA defines an SLO as a commitment to maintain a particular state of the service in a given period. An action guarantee performs a particular activity if a given precondition is met. Action guarantees are used as a means to meet SLOs. [7] adds on this work by proposing an approach of using CIM for the SLA-driven management of distributed systems. It proposes a mapping of SLAs, defined using the WSLA framework, onto the CIM information model.

The GSLA model we propose for SLA specification has the novelty of considering each contracted service relationship as a set of parties playing an SLA game in which each party plays one or more roles to achieve the SLA objectives. GSLA party behavior is captured into a unique semantic component; modeling a role that the party plays. SLOs are specified for each role and enforcement policies are generated to meet them. These policies need not be specified at contract sign time, they can change according to run-time circumstances. Ultimately, roles represent a high-level representation of a set of low-level enforcement policies which are generated, enabled, disabled, and removed as a whole and help keep a consistent relationship between what is high-level behavior and its corresponding low-level actions.

Finally, the use of policies for the management of utility computing infrastructures has been recently addressed by Akhil et al. [17] from HP Labs where policy is used to assist in service deployment. We consider this component as part of the policy deployment and resource configuration component of the PDP.

6 Conclusion

In this paper, we have presented a framework for IT systems management that goes beyond the capabilities currently made available by state-of-art technology on policy-based management. Although the principles that we followed for architecting the framework make it so that it can

enjoy a wider applicability, we chose utility computing as an application domain for our work. The main contribution of our work is in complementing the responsibilities of the standard policy decision point (PDP) of a policy-based management system. The framework extends policy-based management with a wider scope decision ability that is informed and driven through the business objectives and the contractual obligations of the enterprise supported by the IT systems being managed. To this extent, our framework defines two main subsystems on top of a policy-based resource control layer. They are MBO (Management by Business Objectives) where the decision ability supported by analysis of business objectives resides, and GSLA (Generalized SLA), an advanced framework for SLA driven management that lends itself quite naturally to the derivation of IT management policies from the SLAs that the enterprise has contracted. It has to be said that an integrated system built on top of the overall framework has yet to be implemented. Yet, judging from the results we obtained from previous work on the various subsystems that have to various degrees been deployed on real life systems, the approach seems quite promising.

References

[1] Aib I., N. Agoulmine, M.S. Fonseca, G. Pujolle, "Analysis of Policy Management Models and Specification Languages", IFIP Net-Con 2003.

[2] Aib I., N. Agoulmine, G. Pujolle, "Capturing adaptive B2B Service Relationships Management through a generalized Service Information Model", HPOVUA 2004.

[3] Aib, I.; N. Agoulmine, Pujolle, G.; "A Multi-Party Approach to SLA Modeling, Application to WLANs", July 2004, in review.

[4] Bartolini, C.; Boulmakoul, A; Sallé, M.; et al; HP Labs."Management by Contract: IT Management driven by Business Objectives", HPOVUA, June 2004.

[5] Bartolini, C.; Salle, M.; "Business Driven Prioritization of Service Incidents", DSOM 2004.

[6] Buco M. et al., "Managing of eBusiness on Demand SLA Contracts in Business Terms Using the Cross-SLA Execution Manager SAM", IBM, IEEE ISADS, 2003.

[7] Debusmann, M.; Keller, A.; "SLA-driven Management of Distributed Systems using the Common Information Model", IEEE IM 2003.

[8] DMTF, "CIM Core Policy Model", May 12 2000.

[9] IT Governance Institute (ITGI), "Control Objectives for Information and related Technology (CobiT) 3rd Edition", 2002. Information Systems Audit and Control Association.

[10] Keller A.; Ludwig H.; IBM Research Division, "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", JNSM, Vol .11, No. 1, March 2003.

[11] Keller A. , Ludwig H., IBM Research Division, "Web Service Level Agreement (WSLA) Language Specification", Version 1.0, Revision wsla-2003/01/28.

[12] Machiraju, V.; Sahai, A.; van Moorsel, A.; "Web Services Management Network: an overlay network for federated service management", IFIP/IEEE IM 2003.

[13] Machiraju, V.; Bartolini, C.; Casati, F.; "Technologies for Business-Driven IT Management", HPL-2004-101.

[14] Moore B. et al., "Policy Core Information Model", IETF, RFC 3090, February 2001.

[15] R.Kaplan and D.Norton, "The balanced scorecard: translating vision into action", Harvard Business School Press, 1996.

[16] Sahai, A.; Machiraju, V.; Sayal, M.; Moorsel, A.; Casati, F.; "Automated SLA Monitoring for Web Services", IEEE/IFIP DSOM 2002.

[17] Sahai, A.; Singhal, S.; Machiraju, V.; Joshi, R.; "Automated policy-based resource construction in utility computing environments", IEEE/IFIP NOMS 2004.

[18] Scott, D. et al.; "The Evolution toward Policy-Based Computing Services", Gartner 2002.

[19] Sloman, M.; "Policy Based Management: the Holy Grail?", Panel, Policy Workshop 2004.

[20] Strassner, J.; "Policy Bas ed Management Thoughts and Observations from a Network Management Perspective"; Panel, IEEE Policy Workshop 2004.

[21] Sallé M.; Bartolini C.; "Management by Contract", IEEE/IFIP NOMS 2004.

[22] W.Van Grembergen; D. Timmerman; "Monitoring the IT process through the balanced scorecard", IRMIC 1998.