Business Aware Policy Based Management

Issam Aib[1,4], Mathias Sallé[2], Claudio Bartolini[2], Abdel Boulmakoul[3],
Raouf Boutaba[1], Guy Pujolle[4]

[1] Networks and Distributed Systems Laboratory, Univ. Waterloo, Canada
{iaib@bbcr.uwaterloo.ca}
[2] HP Research Labs, Palo Alto, USA
[3] HP Research Labs, Bristol, UK
[4] PHARE Group, LIP6 Laboratory, Univ. Paris 6, France

Abstract. *In this paper, we introduce a Business Aware Framework for the Management of policy enabled IT Systems and its application to Utility Computing Environments. 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

Utility Computing (UC) is a paradigm where shared infrastructure can be provided on demand to multiple applications [15]. IT resources in a UC (compute-power, storage, network bandwidth, etc.) service are provisioned on an demand 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 difficult to accurately predict applications demands in terms of IT resources but also that demand generally varies considerably over 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 timescales involved, the decision making process has to be implemented through *management policies* whose objective is to maximize the *business profit* (value) of the services offered by IT system.

At the same time, traditional policy-based management (PBM) promises to reduce IT costs while simultaneously improving quality of service and adaptability to change [19]. Research in policy-based management systems in various application areas including networking, security, and enterprise systems has been going on for more than a decade, although it is still struggling to make its 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 its wide acceptance.

Our approach stems from the observation that however successful an enterprise might be with its adoption of policy-based management solution, it must be remembered that its IT infrastructure is aimed at the provision of a service which is exchanged for an economic value. Therefore, it is extremely important to make the policy-based management capability *clearly aware* of business level considerations.

This consideration is central to our approach which defines a management stack including a *business aware management Layer* (BAL) and an underpinning *policy-based resource control layer* (RCL). The BAL provides business context to the RCL.

The BAL is based on two components that we have developed previously: Management by Business Objectives (MBO) [5][22] 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 allows modeling SLAs so as to link service quality specifications to the policy based control layer.

The contribution of this paper consists of the business aware management layer and its interaction with the Policy Decision Point (PDP). The resource control layer is described here for completeness and is based on the policy-based management concept. We rather propose an enhancement of the currently accepted PBM architecture of the IETF.

Our proposed framework is widely applicable. However, in this work we describe its application to utility computing environments (UCE). It allows achieving closed-loop management of a UCE by (i) deriving low level management policies from SLAs and business context; and (ii) coupling a reactive policy-based system with a proactive business driven reasoning engine.

This paper is structured as follows. Section 2 describes the BDMF framework. Section 3 describes the BAL layer of BDMF in more detail focusing on the way we model SLAs and Business Objectives; as well as a description of the MBO engine. The use case shows how the BDMF succeeds in aligning IT management with Business objectives.

2 The 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. In the following, we will consider without loss of generality the IT infrastructure as a Utility Computing (UC) infrastructure.

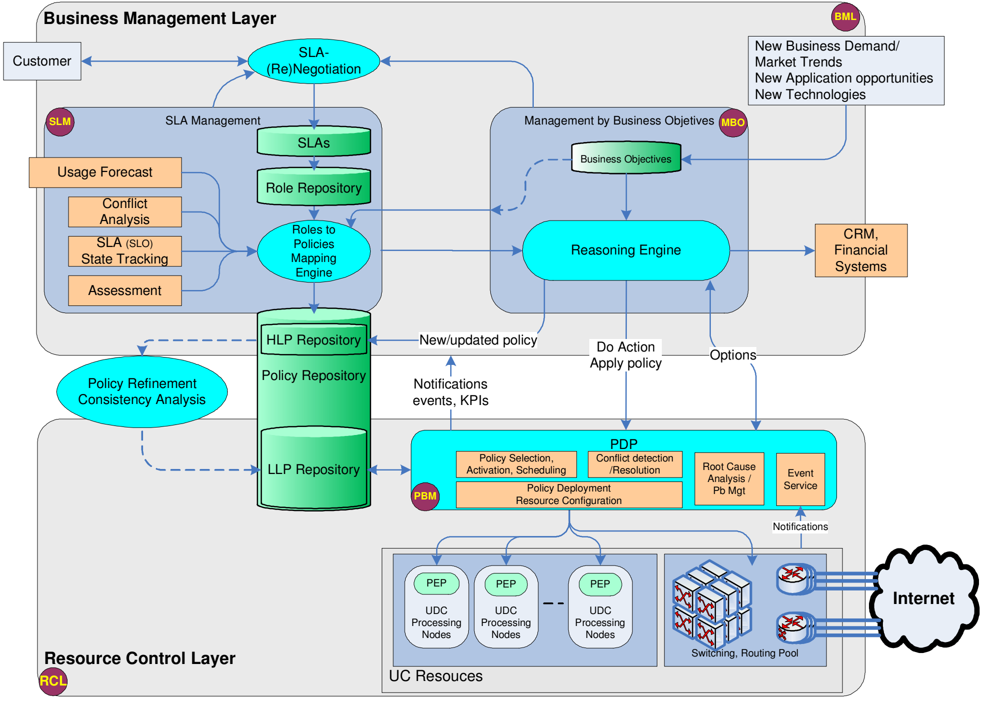


Fig. 1. The BDM Framework.

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

The Business Management Layer is responsible for optimizing the alignment of UC resources usage with the objectives of the utility provider 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.

Business relationships contracted by the utility provider are formalized by SLAs and modeled using the GSLA information model [1]. 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 of 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, we call the Role-to-policies mapping 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 correctly drive the utility resources.

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 takes place at the long term time scale on the SLA database.

LLPs are dealt with by the Policy Decision Point (PDP) module [10][16] 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 utility 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 the generate of appropriate configuration flows in order to enforce active policies.

As it is impossible to define policies upfront to cover all run-time events, it will happen that LLPs may not be sufficient to deal with certain conditions. In those cases, our PDP passes up the control to the BDM. 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 also the opportunity for the architecture to learn and refine the policy repository.

3 Business Management Layer of the BDM Framework

Our 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.

The BAL is architected around two key components (Fig. 1) *(1)* SLA Management (SLM) and *(2)* Management by Business Objectives (MBO). Interactions between the SLM and MBO modules are twofold: *(i)* First, business objectives are used by the roles-to-policies engine to drive policy derivation, and *(ii)* Second, changes in business objectives are reflected by the introduction/update of existing policies.

3.1 Management by Business Objectives (MBO) Engine

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.

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 quickly and reactively deal with arising situations that do not require human intervention.

We develop a mathematical formulation of the MBO Engine functionality [5] which makes of the incident prioritization problem an instance of the well known integer assignment problem:

The *impact* of an *incident i* on *business indicator* I_j when i is supposed to be dealt with within a time of at most t_i of its occurrence is represented by $I_j(i,t_i)$. Business indicators can be of various types and might concern for example "total customer satisfaction", "total SLA compliance", etc. The calculation of the impact of an incident on a given business indicator is an inherently complex process and is to be assured by the MBO engine. Besides, at system run time, multiple incidents can occur concurrently and there is a need to prioritize between them in order to determine which incident to deal with first and how so, so as for the MBO engine to find:

$$\text{Minimum Impact (incident-set)} = \text{Min} \left(\sum_i I\,(i,t_i) \right) \mid i \in \text{incident-set} \qquad \mathbf{(1)}$$

$$\text{Where, } I\,(i,t_i) = \sum_j \omega_j\, I_j\,(i,t_i) \;\; \text{where } \sum_j \omega_j = 1 \qquad \mathbf{(2)}$$

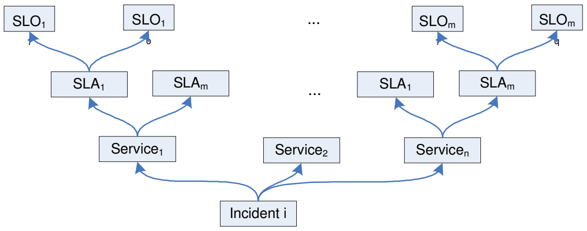


Fig. 2. SLO and SLA related Impact tree of an incident i. Fig. 2 shows an example of a complex dependency that the MBO engine needs to process to calculate the business impact of an "SLO compliance" business indicator at the occurrence of a single incident

The I_j's represent the different business indicators taken into consideration. An ω_j represents the "importance" that the service provider gives to business indicator j. Hence, depending on what is *currently* most important to the service provider (as the priorities might change by time) different optimization choices could be taken. Such high-level driven decisions are by no means at the grasp the PDP and this is why we advocate that it might often end-up optimizing a local function (a specific SLO for example) if it 'blindly' applies a set of policies based on some 'default' incident prioritization scheme. It is then of first importance to consider appending the BDM layer role onto the current state-of-art conception of policy based management architectures.

3.2 SLA Contract Model

We model Service Level Agreements using the Generalized SLA (GSLA) model introduced in [1]. 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.

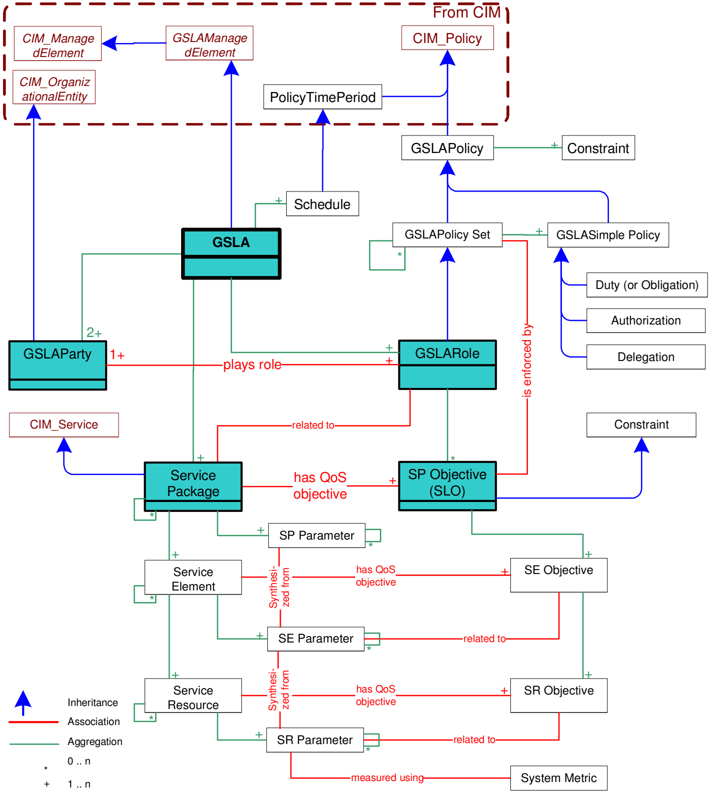


Fig. 3. The GSLA Information Model

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 role is modeled at first approximation by a set of Rules. Hence, *GSLARole* inherits indirectly from GSLA-Policy. 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 represents a group of related Service Elements that are instantiated and managed as a whole and/or are offered altogether to customers.

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 [1].

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 for enforcing its SLOs. Such enforcement policies are normally only viewed by the party related to that role and need not be specified at the common SLA unless explicitly requested by the concerned service customer party.

The behavior of a party is ultimately modeled through policy [3]. A GSLARole is modeled through a set of policies as well as the set of SLOs it is required to ensure as part of its responsibilities in the GSLA [1]. 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 the result of no SLO mapping.

4 An Incident Management Use case

We modify the use case we did in [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.

We assume an IT hosting infrastructure for a Utility Computing Provider named UCP. As depicted in Fig. 4, UCP hosts services for two companies: C1 and C2. Both services are web-based and require Web Servers, Application Servers and back-end systems such as Databases and Enterprise Resource Planning Systems. Customers of C1 and C2 access these services by connecting to the servers hosted onto UCP' IT infrastructure. This infrastructure takes the form of a set of shared IT resources inside UCP' utility computing infrastructure.

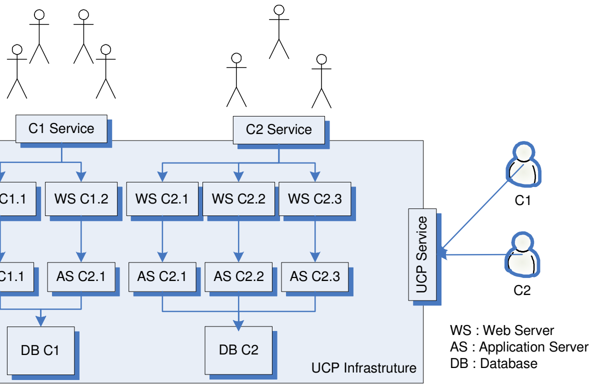


Fig. 4. C1 and C2 hosted services running on UCP utility infrastructure

UCP customers, C1 and C2, are provided similar services but with different QoS assurances. These guarantees are negotiated prior to service deployment and can be renegotiated over time as the service requirements of the customers evolve. We show the parts of SLAs that are of concern for the use case. Because of lack of space we avoided to include the full (lengthy) XML specification [2] of the involved GSLAs:

GSLA 1. UCP-C1 GSLA

UCP offers C1 a Web-Server Service with schedule SC1
Supported capacity is of 1000 simultaneous connections
C1 is monthly charged $0.1*Server.Capacity
Monthly availability of the hosted service will be 99%.
Otherwise, fully refund C1 the extra period over which the breach occurred.
Average time to process any customer service request over a month period will be less than 300 ms.
Otherwise, credit C1 20% of the monthly charge
IF C1 fails to pay the monthly charge for 3 successive months then the contract will be terminated.

GSLA 2. : UCP-C2 GSLA

UCP offers C2 a Web-Server Service with schedule SC2
Supported capacity is of 5000 simultaneous connections
C2 is monthly charged $0.15*Server.Capacity
Monthly availability of the hosted service will be 99.9%.
Otherwise, credit C2 100% of the monthly charge
Any service unavailability will be fixed within 20 min of the receipt of a trouble ticket.
Otherwise, C2 will be fully refunded the charge of the period over which the breach occurred.
Average time to process any customers service request over a month period, will be less than 200 ms
Otherwise, credit C2 80% of the monthly charge
IF C2 fails to pay the monthly charge for a successive 6 months then the contract will be terminated.

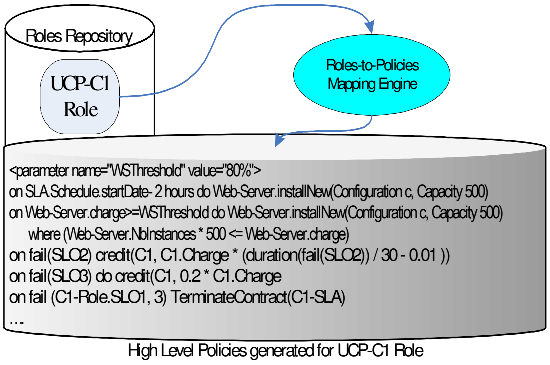
Although C1 and C2 SLAs are specified over two service parameters, availability and service latency, they each define different service levels and have different penalties. For the purpose of keeping the use case simple we consider that the UCP has two parameters to maximize within its business profit (value) function:

$$BP = \alpha\ \text{direct-financial-profit} + \beta\ \text{customer-satisfaction; where } \alpha+\beta=1 \qquad (\mathbf{3})$$

Direct financial profit = money that clients pay minus service provision charges

We assume here that based on local information related to the performance of the UCP web servers the roles-to-policies mapping engine generated the set of high-level policies for the UCP-C1 role depicted in Fig. 5.

Given that the UCP Roles-to-policies mapping engine knows that a UCP web server resource instance can serve up to 500 clients without reducing the contracted QoS, we understand from the policy set of Fig. 5 that the UCP considered the "lazy policy" of per-need provisioning to meet its GSLA with C1. Whenever there is a need, an additional web server instance is installed and provisioned for C1 customers. We assume that UCP took the same approach for mapping its UCP-C1 role.


Roles Repository
UCP-C1 Role
Roles-to-Policies Mapping Engine
High Level Policies generated for UCP-C1 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 fail(SLO2) credit(C1, C1.Charge * (duration(fail(SLO2)) / 30 - 0.01 ))
on fail(SLO3) do credit(C1, 0.2 * C1.Charge
on fail (C1-Role.SLO1, 3) TerminateContract(C1-SLA)
…
```

Fig. 5. Generated Policies for UCP-C1 Role

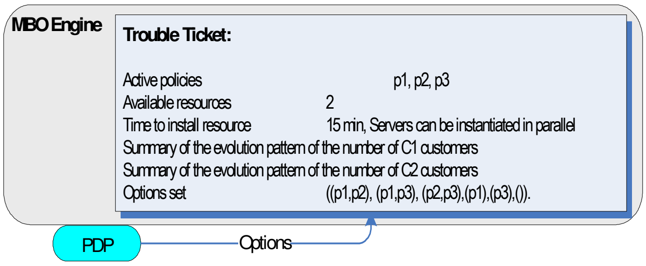
Let's assume that C2 and C1 services reach the configuration of 1 and 4 web servers respectively and there are only two free resources that can be allocated to C2 or C1 services. Then, a sudden increase in the number of C1 and C2 customers is noticed leading to the following set of active LLP policies (WS21: virtual instance of the C1 web server):

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

In addition, policies are not taken care of immediately after their activation as they are first queued within a set of activated policies waiting for the PDP to treat them. So we suppose that p1, p2, and p3 are noticed by the PDP at the same time. The PDP is confronted here with a run-time policy conflict. In contrast to a static policy conflict [3], this is a type of conflict that cannot be predicted by the role-to-policies mapping process of the BAL layer (Fig. 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 symptom of service degradation that the PDP cannot measure correctly. The PDP hence sends a set of options for the MBO to decide which to apply (Fig. 6).

The MBO engine will take into consideration service and business level parameters related to *(i)* C1 and C2 GSLAs (current total time of service unavailability, time to recover from unavailability, penalty amounts, Expectation of the evolution pattern for the number of customers for C2 and C1 (to decide whether to allocate resources or just do not if the congestion period is temporary); and *(ii)* Current customer satisfaction indicator value for C1 and C2.

After calculating the impact on Business Profit (utility) value for each option of the options set based on equations 3, 2, and 1, the MBO engine determines the option which maximizes the utility value and sends it back to the PDP for execution. It is clear that it is not necessarily the FIFO treatment of active policies which will lead to the best business profit. When the PDP knows that there will be inevitably a degradation of service (that might lead or not to some SLO violation) the MBO answers about the best options (strategy) to follow so as to achieve minimal degradation of the business profit function.



Trouble Ticket:

Active policies	p1, p2, p3
Available resources	2
Time to install resource	15 min, Servers can be instantiated in parallel
Summary of the evolution pattern of the number of C1 customers	
Summary of the evolution pattern of the number of C2 customers	
Options set	((p1,p2), (p1,p3), (p2,p3),(p1),(p3),()).

Fig. 6. Options generated and sent by the PDP to the MBO

5 Related Work

Driving IT management from business objectives is quite a novel proposition. In [6][7], 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 [22]. 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 MBO capability applied to the incident management domain see our work in [5].

In another respect, the area of SLA-driven management has been closely interested in the subject of SLA modeling. WSLA, [12][13], from IBM research and WSMN [14][17] from HP Labs analyze and define SLAs for Web Services by building new constructs over existing Web Services formalisms. [17] 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 (Finite State Automata) 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. [12][13] 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. [9] 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. Finally, [8] considers a direct application of WLA within UCEs.

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. [18] 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. The main contribution of our work is in complementing the responsibilities of the standard policy decision point (PDP) of a policy-based management system. 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. To this extent, 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.

References

1. Aib, I.; N. Agoulmine, Pujolle, G.; "A Multi-Party Approach to SLA Modeling, Application to WLANs", IEEE CCNC'05, January 2005, USA.
2. Aib I.; Salle M.; Bartolini C.; Boulmakoul A.; "A Business Driven Management Framework for Utility Computing Environments", HPL-2004-171, HP Labs, Bristol UK, September 2004.

3. Aib I., N. Agoulmine, M.S. Fonseca, G. Pujolle, "Analysis of Policy Management Models and Specification Languages", IFIP Net-Con 2003.
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. Buco M. J., Chang R. N., Luan L. Z., Ward C., Wolf J. L., Yu P. S., "Utility computing SLA management based upon business objectives", IBM Systems Journal, Vol 43, No 1, 2004.
8. Dan A., Davis D., Kearney R., Keller A., King R., Kuebler D., Ludwig H., Polan M., Spreitzer M., Youssef A., "Web services on demand: WSLA-driven automated management", IBM Systems Journal, Vol 43, No 1, 2004.
9. Debusmann, M.; Keller, A.; "SLA-driven Management of Distributed Systems using the Common Information Model", IEEE IM 2003.
10. DMTF, "CIM Core Policy Model", May 12 2000.
11. IT Governance Institute (ITGI), "Control Objectives for Information and related Technology (CobiT) 3rd Edition", 2002. Information Systems Audit and Control As-sociation.
12. 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.
13. Keller A. , Ludwig H., IBM Research Division, "Web Service Level Agreement (WSLA) Language Specification", Version 1.0, Revision wsla-2003/01/28.
14. Machiraju, V.; Sahai, A.; van Moorsel, A.; "Web Services Management Network: an overlay network for federated service management", IFIP/IEEE IM 2003.
15. Machiraju, V.; Bartolini, C.; Casati, F.; "Technologies for Business-Driven IT Management", HPL-2004-101.
16. Moore B. et al., "Policy Core Information Model", IETF, RFC 3090, February 2001.
17. Sahai, A.; Machiraju, V.; Sayal, M.; Moorsel, A.; Casati, F.; "Automated SLA Monitoring for Web Services", IEEE/IFIP DSOM 2002.
18. Sahai, A.; Singhal, S.; Machiraju, V.; Joshi, R.; "Automated policy-based resource construction in utility computing environments", IEEE/IFIP NOMS 2004.
19. Scott, D. et al.; "The Evolution toward Policy-Based Computing Services", Gartner 2002.
20. Sloman, M.; "Policy Based Management: the Holy Grail?", Panel, Policy Workshop 2004.
21. Strassner, J.; "Policy Based Management Thoughts and Observations from a Network Management Perspective"; Panel, IEEE Policy Workshop 2004.
22. Sallé M.; Bartolini C.; "Management by Contract", IEEE/IFIP NOMS 2004.
23. W.Van Grembergen; D. Timmerman; "Monitoring the IT process through the balanced scorecard", IRMIC 1998.