A CIM-based Framework to Manage Monitoring Adaptability

Audrey Moui, Thierry Desprats, Emmanuel Lavinal and Michelle Sibilla
IRIT, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse cedex 9, France
Email: {moui,desprats,lavinal,sibilla}@irit.fr

***Abstract*—Integrated management should cope with numerous, heterogeneous and complex systems. In this context, the monitoring activity should be sensitive and efficiently reactive to both variations of management applications requirements and environmental constraints. This paper introduces concepts, definitions, information models and algorithms to manage monitoring adaptation enforcement. A CIM/WBEM implementation has been realized as proof of concept and tested to measure the management overhead generated by the proposed approach. This framework aims at being generic enough to support any software solutions providing monitoring adaptation decisions.**

I. INTRODUCTION

The efficiency of the monitoring activity is a major preoccupation for recent management paradigms that are based on more and more autonomous and decentralized decision-making. This is particularly the case in the context of integrated management (network, systems, services) where complexity is increased due to the multiple components and the affluence of heterogeneous management information. Monitoring is required to become more flexible to autonomously tackle, in a relevant and efficient way, the variations of the management functional requirements (e.g. accuracy, freshness, granularity, scope of collected data) and self-optimization criteria (e.g. performance issues in constrained environments).

This paper presents another step performed in order to improve the adaptiveness level of monitoring activities, by providing facilities to support the enforcement of the adaptation of these activities in the general context of integrated management. A bottom-up approach has been adopted to define informational, operational models and algorithms. Based on these, a framework has been implemented and tested in a CIM/WBEM environment.

The remainder of this article is organized as follows. Section II presents motivations and related works. Section III gives an overview of the general architecture of the proposed framework, while section IV presents the CIM-based implementation. Section V presents a study scenario and some related evaluation results. Finally, the last section concludes the paper.

II. MOTIVATION

We define **adaptive monitoring** as the ability an online monitoring function has to decide and to enforce, without disruption, the adjustment of its behavior for maintaining its effectiveness, in respect of the variations of both functional requirements and operational constraints, and possibly for improving its efficiency according to self-optimization objectives.

A. Requirements for Adaptation Enforcement Management

The monitoring activity is a process which relies on the gathering of information data which can be raw, symbolic, aggregated, transformed or even filtered. The gathering is classically performed with the use of "polling" (i.e. pulling data, by periodically requesting some targeted source or aggregator for data) and "event reporting" (i.e. receiving, in an ad hoc, sporadic or periodic way, pushed data from a provider or an aggregator) mechanisms, that can be jointly used or combined for performance purposes as stated in [6] or in [18].

Automating monitoring adaptation requires to manage and to control the monitoring activity itself, leading to modify its current behavior. To do so, actions need to be achieved to enforce adaptation decisions that aim at varying the scope and/or the means of the observations. Typically, to reduce the CPU resource consumption, the number of basic monitoring running mechanisms has to be decreased; or to require a higher level of freshness of the collected data, the polling frequency has to be intensified. Therefore, in the large majority of cases, the achievement of a monitoring adaptation will be carried out by managing the operational life cycle of basic monitoring mechanisms as well as the parameters that can govern their behavior.

From these typical situations, we identified a set of requirements to operate such a management of the enforcement of a monitoring adaptation decision. We just mention here the two requirements on which we focus in this paper:

- The need to capture any operational request related to the behavior adjustment of the running monitoring activity. It implies to offer some well-defined interfaces allowing the decisional level to express actions to be done to the underlying enforcement infrastructure.
- The necessity to check at runtime the coherence and the feasibility of the requested actions.

In this paper, we do not focus on the decision making of a monitoring adaptation: we assume that it can be based on any of well-known solutions including constraints resolution, inference engine, management policies, rules, bio-inspired approaches, etc. In a complementary way, our concern is to focus more on the effective realization of the decided

978-3-901882-48-7 ©2012 IFIP

adaptation by providing a generic framework to support the management of this post-decision process.

B. Related Works

In the field of system and network management, numerous research works contribute on adaptive monitoring. We reference here a few of them that recently address programmability motivations that seem to be the closest to ours.

Duarte et al [5] proposes a policy- and event-based interesting approach to program configurable network monitoring applications, but does not really support their adaptations control at runtime; furthermore, it is dedicated exclusively to an SNMP/DISMAN [4] environment restraining the need to deal with several data gathering protocol as required in an integrated management approach. Ouda et al [12] investigates the possibility, when a PBM approach is adopted, to extract from a set of policies information used to configure the underlying monitoring infrastructure. Initialization and configuration of their Monitoring agents are implemented through a common code: neither precise interfaces nor the settings are explicitly mentioned. Roxburgh et al [14] adopts a pragmatic SLA-based approach to offer to the management applications a service-oriented interface. Their monitoring service allows them to dynamically control and scale the integrated monitoring activity deployment. Monitoring is based on two commercial products. To deal with their integration in the monitoring service, ad hoc adaptors need to be implemented: no common management information has been defined to support their drivability in a uniform way. Le Duc et al [7] presents ADAMO, an adaptive monitoring solution, inspired from [1], able “to tackle different QoI-aware data queries over dynamic data streams, transform them into probe configurations settings under resource constraints”. A constraint-solving approach has been chosen for adaptation decision making, particularly for the computation of the value of data source inquiry frequency. Consequently, the proposed adaptation modeling is strongly influenced by CSP. A fine-grained management information model is not clarified. Other works have designed adaptive protocol-based solutions allowing trade-offs between performance and quality of the information [13], which exclusively relies on push data models in large scale networks.

We presented in [8] a way to manage polling adaptability in a CIM-WBEM environment. The current paper presents another step of the work in progress, by adding an event reporting mechanism management model, and a more complete definition and implementation of the adaptability.

III. THE ADAPTIVE MONITORING FRAMEWORK

We introduced a control plane which aims at managing the set of the underlying basic mechanisms used to operate the monitoring of the managed system (operational plane). The controlled system is then the monitoring activity itself. Figure 1 gives, at a high level of abstraction, an overview of what should be the architectural framework of an adaptive monitoring service based on three capabilities that we introduced in [9]. It intentionally focuses on the control plane and clearly emphasizes the hierarchical dependencies existing between these capabilities.

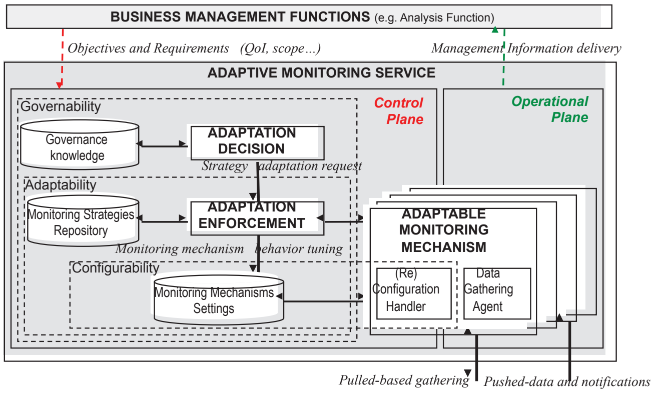

Fig. 1. Architecture of an Adaptive Monitoring Service

1) Configurability: The configurability is defined as the ability to initially acquire and then modify on demand, at runtime, the values of the parameters which govern the behavior of each basic monitoring mechanism: polling P and event reporting ER. For each of these two types of mechanism, we defined common parameters and algorithms for the mechanisms to govern initially and at runtime their operational behavior.

2) Adaptability: Adaptability is defined as the ability to dynamically change, in a consistent way, the behavior of a whole monitoring activity and, consequently, to modify the constitution of the set of the underlying monitoring mechanisms and/or the way they operate. This capability has to be performed by managing the life cycle of the basic monitoring mechanisms and relies on the configurability capacity of each related mechanism. To do so, we defined the concept of “strategy” and identified five basic adaptation operators.

A **monitoring strategy** S is an association between the mechanisms M applied on their respective targets T (with concerns of genericity, we consider targets and sources as any relevant information related to managed elements), their configuration C and their operational state Op:

$$S = (M_T, C, Op) = \{\langle M_{t_1}, C_1, Op_1\rangle, ..., \langle M_{t_n}, C_n, Op_n\rangle\}$$

An adaptation will result in a monitoring strategy change that will consequently modify the monitoring activity behavior. The evolution from a strategy S into S' is defined as an “adaptation” δ_S [3]: $\delta_S = S \rightarrow S'$

We have identified five basic adaptation operators that may be defined similarly as service interfaces for supporting at runtime the reconfiguration enforcement of a monitoring strategy adaptation. Then, a mechanism can be added by using the $\mathcal{A}$ operator (*adjunction*); a mechanism can be deleted by using the $\mathcal{D}$ operator (*deletion*); and a configuration of a mechanism can be updated at runtime, thanks to the $\mathcal{U}$ operator (*update*). This operator makes also possible to add or delete targets in case of polling. Moreover, rather than deleting the mechanism, it can be relevant to *suspend* it ($\mathcal{S}$ operator); then the polling mechanism can be used again later if needed, by *resuming* it

($\mathcal{R}$ operator). These operators have been formalized to express invocation constraints and execution post-conditions. As an example, the top of Figure 3 shows the adjunction operator formalization.

3) Governability: Piloting the adaptation operations to adjust and/or optimize the monitoring activity is provided by the governability capability. In order to enforce the actions resulting from the adaptation decisions, governability must rely on the adaptability level through the use of the adaptation operators, that definition constitutes the interface between these two layers. We have clearly defined the interface of each capability in order to ensure the framework's modularity and to allow the possibility to specialize their implementation on any technological environment.

IV. A CIM-BASED IMPLEMENTATION

First as a proof of concept, second to measure the overhead due to the adaptation management, and third to highlight some impacts on the way to use the operators, a prototype for monitoring configurability and adaptability has been developped in a CIM [2]/WBEM [16] environment. Management information model is needed to handle the enforcement of the adaptation. So, we chose CIM, as it is a DMTF standard bringing an already existing information model that could be extended to integrate a management view of our mechanisms. Moreover, it proposes some patterns through Settings that can be used to easily model our mechanisms configuration. Finally, it makes possible to simply connect, thanks to the managed elements concept, our specific model to the existing targets.

A. Monitoring Configurability

We implemented both polling and event reporting adaptable mechanisms. No polling management elements have been defined in DMTF standards and standard schemas concerning event reporting are limited to indications subscription and delivery but do not address a possible configuration of their listening behavior.

The **polling configurability** and the related identified parameters have already been detailed in [8]. The **event reporting configurability** is novel and more specific. Even if some of them are experimental, the classes defined in the `CIM_Event` schema present all the parameters to determine how notifications should be produced, filtered and delivered to the consumer when it subscribes to indications or pushed data. However, it can be interesting to observe when the notifications or events are received. Indeed, the notifications reception frequency can be relevant to detect any particular issue occurring on the managed system. We defined temporal settings to configure three specific listening mode: `silence`, `burst`, `heartbeat`.

The configurability has been modeled as an extension of the DMTF CIM common model as shown on Figure 2. Moreover, we implemented both polling and event reporting adaptable mechanisms, to acquire and then to apply the CIM configuration parameters that will govern the operational behavior of the mechanism, and to listen for and handle incoming behavior updating requests.

$$(M_T, C, Op) \xrightarrow{\mathcal{A}(m_{t_x}, c_x)} (M'_{T'}, C', Op')$$

$$\text{with } \left(M'_{T'}, C', Op'\right) = (M_T, C, Op) \cup \{m_{t_x}, c_x, running\}$$

```
if the parameters are coherent/consistent then                       Control plane
   if the mechanism does not already exist then
      if the targets exist and can be questionned then
         instanciate IRIT_PollingOperation class (and parameters value);
         instanciate IRIT_PollingSettingData class;
         instanciate IRIT_PollingData association;
         if there is only one target then
            instanciate IRIT_IndividualPollingTarget association;
         else
            instanciate IRIT_CollectionOfPollingTarget class;
            instanciate IRIT_CollectedMSEs association;
            instanciate IRIT_CollectionPollingTarget association;
         end if;                                                 Operational plane
         execute operationally the polling;
      end if;
   end if;
end if;
```

Fig. 3. Formalization and CIM Implementation of the Polling $\mathcal{A}$ Operator

The CIM model has been instrumented on a CIM server, offered by the Open Pegasus product [11], and its CIM repository to store the mechanisms settings.

B. Monitoring Adaptability

The previously formalized operators are the core of this module. They operate on the two planes: the management of the dynamic modifications on the CIM model (control plane) and the use of the model to control the execution of the monitoring mechanism (operational plane). Figure 3 shows, on the bottom, the example of the algorithm of the adjunction operator for a polling mechanism.

The CIM implementation of this interface (adaptation operators) is done with the Java SBLIM [15] API. It allows the dynamic management of the proposed CIM model thanks to operators able to see, modify, add and delete CIM objects and CIM associations.

V. EXPERIMENTAL RESULTS

All the presented results have been obtained by computing the average of fifty measures realized in favorable conditions. The CIM server is executed on a virtual machine running Linux Ubuntu LTS 8.04 "Hardy Heron"; the CIM client is running on the physical machine (hosting the virtual machine) running Mac OS X 10.5.8 "Leopard".

Figure 4 illustrates two bellow-described scenarios of dynamic adaptation performed thanks to the developed prototype. Black values show the total execution time needed to enforce each adaptation operator; whereas the red values show the time devoted to the execution of the control plane.

- Adaptation 1. A polling $P1$ is running and a listener $ER1$ is set on burst detection. When a burst occurs, the following adaptation, composed by three operations, is performed: the suspension of the running polling $P1$, the adjunction of a new one $P2$, and the update of the event reporting $ER1$ `ReceptionMode`. This adaptation is executed over 745 ms, in which 87% (653ms) of time is devoted to the control plane;
- Adaptation 2. It makes the system back to normal, by performing another adaptation: the resumption of the suspended polling $P1$, the deletion of the added polling $P2$ and the

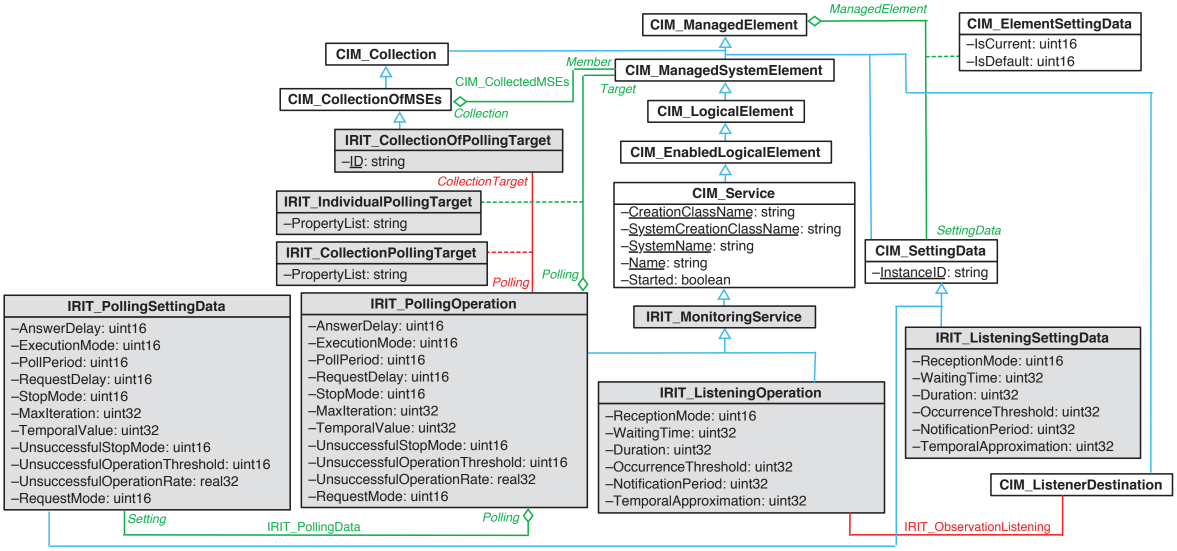

Fig. 2. CIM Common Model Extensions for Monitoring Configurability Level

update of the event reporting $ER1$. This adaptation is executed over 533 ms (control plane: 431ms/80%).

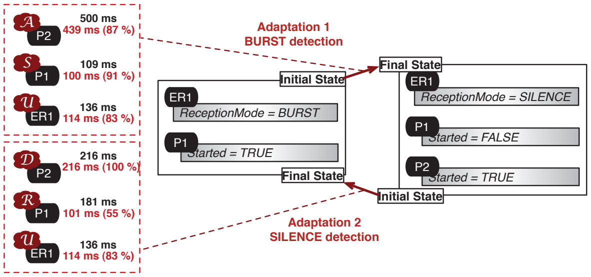

Fig. 4. Example of strategy adaptation

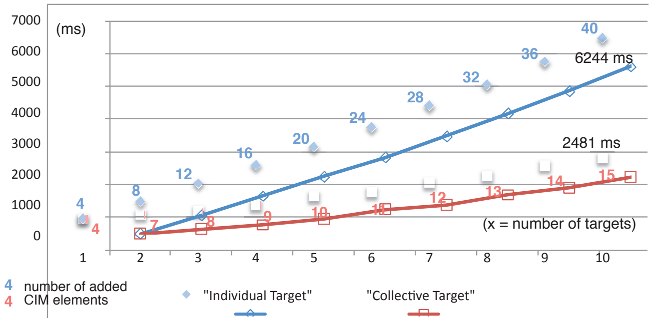

Fig. 5. Execution Duration for Adjunction Operator

1) Time and resource cost reduction: The scenario reveals that the suspension and resumption operators are really useful on an existing polling: the combined suspension and resumption of the polling $P1$ lasts **290ms**, while the combined deletion and adjunction of the polling $P2$ lasts **716ms**.

2) Automation of scope adaptation: We tested the adjunction of a polling $P2$ with more and more targets to enlarge the scope of the monitoring. As shown on Figure 5, the more the targets are numerous, the more the adjunction execution time will be high. But there is a difference appearing according to the way the adjunction operator is used: the adjunction of a polling operation applied on n targets (collective targets, in red) is globally more interesting than the adjunction of n polling operations applied on individual targets (in blue) as it is performed faster (the ratio is 2.5).

VI. CONCLUSION

This paper presented a generic monitoring framework that aims at enforcing the monitoring adaptation at runtime. When the adaptation is performed, the monitoring is made the less intrusive possible, by efficiently adjusting itself to every situation variation (business objectives changes, monitoring self-optimization, constraints variations on resources). Following a control-oriented viewpoint, three capabilities have been identified within the framework: configurability, adaptability and governability. In this paper, we offer solutions about the configurability and adaptability levels by giving definitions and algorithms to make a coherent and generic interface through well-defined parameters and operators. This interface has been implemented and tested to prove the feasibility of the approach so far and to measure the overhead due to adaptation management. As positionned in a high level of abstraction, and thanks to the facilities brought by the CIM common model, the implementation required very few additional modeling features. The experimentation shows that the genericity of the interface has been well-defined and is coherent, thanks to our operators and algorithms. Whereas, the first results indicate that the execution times related to the management model are correct and argue for the viability of the proposed framework.

Future works will tend now to integrate the governability capability to the presented framework. Management policies and inference engines are possible approaches which are currently under study to evaluate some criteria of suitability to support the governability capability, according to the underlying framework and needs.

REFERENCES

[1] Abid, Z.; Chabridon, S.; Conan, D.: **A Framework for Quality of Context Management**. in *QuaCom*. (2009). 120–131.

[2] DMTF, **Common Information Model (CIM) Infrastructure** specification v.2.6.0 [online], (2010) [retrieved: July, 2012], http://www.dmtf.org/standards/cim.

[3] Chung, L.; Subramanian, N.: **Adaptable Architecture Generation for Embedded Systems**. *Journal of Systems and Software*. 71(3) (2004). 271–295.

[4] IETF Working Group, **Distributed Management** [online], (2006) [retrieved: July, 2012], http://datatracker.ietf.org/wg/disman/.

[5] Duarte Jr, E.P.; Musicante, M.A.; Fernandes, H.H.: **ANEMONA: a Programming Language for Network Monitoring Applications**. *Internat. Journ. of Net. Managem.*. 18(4) (August 2008).

[6] Durai, B.; Gonsalves, T.A.; Sivalingam, K.M.: **Adaptive Push-Based Data Collection Method for Online Performance Monitoring**. *Communications (NCC), 2011 National Conference on*. (Jan. 2011). 1–5.

[7] Le Duc, B.; Collet, P.; Malenfant, J.; Rivierre, N.: **A QoI-Aware Framework for Adaptive Monitoring**. *2nd International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2010)*. (Nov. 2010). 133–141.

[8] Moui, A.; Desprats, T.; Lavinal, E.; Sibilla, M.: **Managing Polling Adaptability in a CIM/WBEM Infrastructure**. *Internat. DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and New Technologies (SVM10)*. (2010). 1–6.

[9] Moui, A.; Desprats, T.: **Towards Self-Adaptive Monitoring Framework for Integrated Management**. *5th IFIP Internat. Conference on Autonomous Infrastructure, Management and Security (AIMS 2011)*. (2011). 160–163.

[10] Munawar, M.A.; Jiang, M.; Ward, P.A.S.: **Monitoring Multi-Tier Clustered Systems with Invariant Metric Relationships**. *Proceedings of the 2008 Internat. Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2008)*. (2008). 73–80.

[11] Open Pegasus, **OpenPegasus "C++ CIM/WBEM Manageability Services Broker"** [online], (2012) [retrieved: July, 2012], https://collaboration.opengroup.org/pegasus/.

[12] Ouda, A.; Lutfiyya, H.; Bauer, M.: **Automatic Policy Mapping to Management System Configurations**. *Policies for Distributed Systems and Networks (POLICY), 2010 IEEE International Symposium on*. (2010). 87–94.

[13] Prieto, A.G.; Stadler, R.: **Controlling Performance Trade-offs in Adaptive Network Monitoring**. *Proceedings of IM 2009*. (2009). 359–366.

[14] Roxburgh, D; Spaven, D; Gallen, C.: **Monitoring as an SLA-Oriented Consumable Service for SaaS Assurance: a Prototype**. *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. (2011). 925–939.

[15] Sourceforge SBLIM, **Main Page** [online], (2012) [retrieved: July, 2012], http://sourceforge.net/apps/mediawiki/sblim/index.php?title=Main_Page.

[16] DMTF: "Web-Based Enterprise Management" specifications [online], (2012) [retrieved: May, 2012], http://www.dmtf.org/standards/wbem.

[17] Xue, Z.; Dong, X.; Wu, W.: **AOCMS: An Adaptive and Scalable Monitoring System for Large-Scale Clusters**. *Proc. APSCC*. (2006). 466–472.

[18] Yang, G.; Wang, K.; Zhou, X.: **An Adaptive Resource Monitoring Method for Distributed Heterogeneous Computing Environment**. *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on*. (Aug. 2009). 40–44.