

- (1) サービスレベル項目
可用性など SL のプロパティを定義する。
- (2) サービスレベル項目保証
保証 SL のプロパティのメトリクスを設定する。
- (3) 評価方法
サービスの実行状態からプロパティ値を測定し、保証 SL を満たすか確認する方法を定義する。
- (4) 対処方法
サービスの修復方法など保証 SL を下回った時のルールを定義する。
- (5) 罰則内容
保証 SL を下回った時のペナルティを定義する。

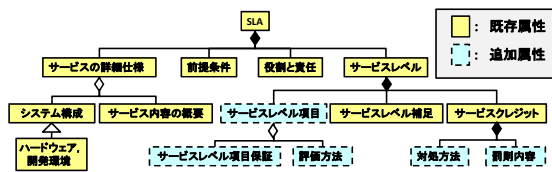


図2 拡張SLAの属性モデル

6.2. SLA メタモデルの作成

SLA メタモデルを図3に示す。SLA 管理に関わるステークホルダ、SL 監視の対象となるサービス、保証SLの関係を表現する。

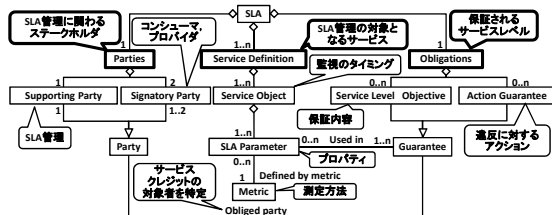


図3 SLAメタモデル

6.3. 監視データモデルの定義

SLA メタモデルに基づき、監視データモデルのスキーマを定義する。監視データモデルは、以下の四つのデータモデルに分類される。

- (1) タイミングモデル
タイミングモデルは、プロパティの監視タイミングを定義するデータモデルである。SLA 管理アーキテクチャ内では評価、測定を行う際、トリガとして利用される。タイミングは、周期起動、非周期起動を実現するためタイムトリガ、イベントトリガの2種類に分類する(図4)。



図4 タイミングモデル

- (2) サービスリポジトリ
サービスリポジトリは、監視の対象となるサービス情報、

プロパティ、プロパティのメトリクスを定義するデータモデルである。SLA 管理アーキテクチャ内では測定時に、測定の対象となるサービスを検索するために利用される(図5)。

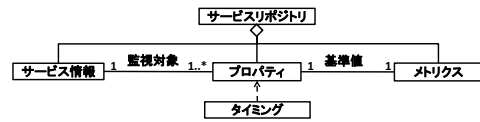


図5 サービスリポジトリ

- (3) サービスクレジットリポジトリ
サービスクレジットリポジトリは、測定結果が保証 SL に満たない条件を定義するデータモデルである。SLA 管理アーキテクチャ内では評価時に測定結果が保証 SL を満たすか確認するために利用される(図6)。

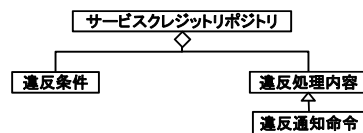


図6 サービスクレジットリポジトリ

- (4) ステークホルダリポジトリ
ステークホルダリポジトリは、SLA 管理のステークホルダを定義するデータモデルである。SLA 管理アーキテクチャ内では違反通知を送信する対象者を特定するため利用される(図7)。

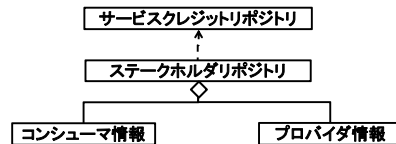


図7 ステークホルダリポジトリ

7. SLA 管理アーキテクチャ

7.1. SLA 管理アーキテクチャの機能

アーキテクチャの機能には、サービスの実行状態の取得、測定、評価、違反通知が含まれる。本稿では、SLA の多様化の課題を解決するため、SLA をモデル化する機能を追加する。また測定と評価のタイミングが異なっても SL 監視ができるよう、タイミング調整機能を追加する(図8)。

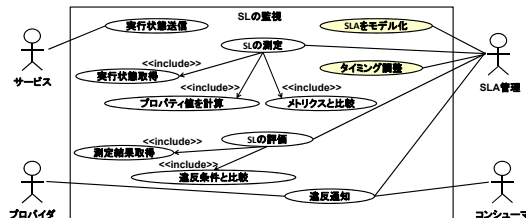


図8 アーキテクチャの機能

7.2. SLA 管理アーキテクチャの構造

アーキテクチャの構造を図9に示す。

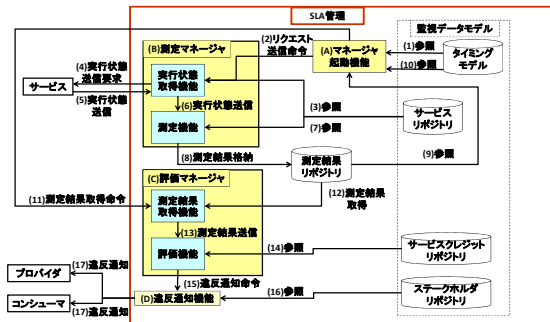


図9 アーキテクチャの構造

アーキテクチャの主な機能を以下に示す

- A) マネージャ起動機能
 タイミングモデルを用いて、測定マネージャ、評価マネージャを起動する。
- B) 測定マネージャ
 サービスリポジトリ内のサービス情報を用いて、サービスから実行状態を取得する。取得した実行状態をサービスリポジトリ内のメトリクスを用いて測定し測定結果を出力する。
- C) 評価マネージャ
 測定結果リポジトリから測定結果を取得する。取得したデータをサービスクレジットリポジトリ内の違反条件を用いて評価を行う。保証 SL に満たない場合、違反通知機能に違反通知命令を送信する。
- D) 違反通知機能
 ステークホルダリポジトリ内のコンシューマ情報とプロバイダ情報を用いて、対象者に違反通知を送信する。

7.3. SLA 管理アーキテクチャの振舞い

アーキテクチャの振舞いを、測定、評価、違反通知に分けて以下に示す。

(1) 測定の振舞い

マネージャ起動機能が、タイミングモデルを参照し、トリガを選択し、測定マネージャを起動する。起動の際、実行状態送信要求を送る命令(以下、リクエスト送信命令と略記)を測定マネージャに送信する。測定マネージャ起動後、測定マネージャは、測定対象サービスから実行状態を取得する。実行状態の取得後、実行状態の測定を行い、測定結果を測定結果リポジトリに格納する(図 10)。

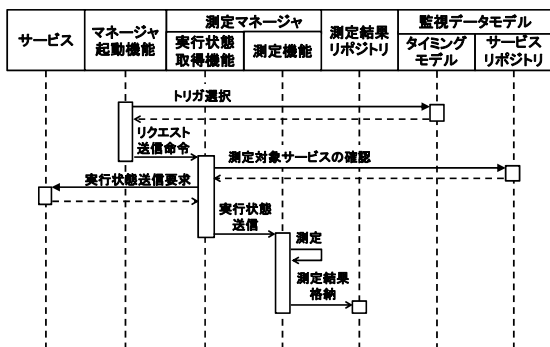


図 10 測定の振舞い

(2) 評価の振舞い

マネージャ起動機能が、測定結果が測定結果リポジトリに格納されていることを確認する。その後、マネージャ起動機能がタイミングモデルを参照し、トリガを選択し、評価マネージャを起動する。起動の際、測定結果を取得する命令(以下、測定結果取得命令と略記)を評価マネージャに送信する。評価マネージャ起動後、評価マネージャは、測定結果を取得、評価し、測定対象サービスが違反条件にあてはまる場合、違反通知機能に違反通知命令を送信する(図 11)。

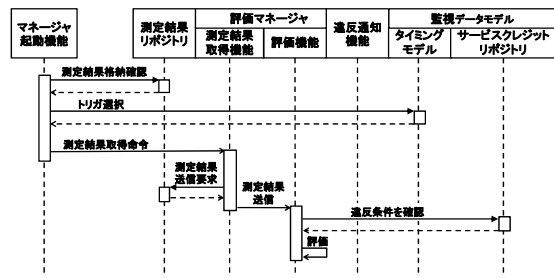


図 11 評価の振舞い

(3) 違反通知の振舞い

違反条件にあてはまる場合、評価機能が違反通知機能に違反通知命令を送信する。違反通知機能は、違反通知を送信する対象のステークホルダを特定し、違反通知を送信する(図 12)。

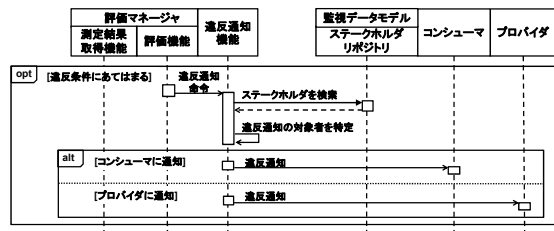


図 12 違反通知の振舞い

8. クラウドサービスへの適用

提案アーキテクチャを実際のクラウドサービスに適用し、提案アーキテクチャの有効性を示す。提案アーキテクチャ内で有効性を示す対象となるのは、タイミングモデルとマネージャ起動機能とする。

8.1. 適用方法

適用は、以下の手順に従い行う。

(1) プロパティの抽出

実際のクラウドサービスの SLA 内に複数存在するプロパティの中から、サービスの実行状態を取得し測定、評価できるプロパティを抽出する。クラウドサービス内に複数存在するプロパティの数を項目数とし、抽出したプロパティの数を抽出数とする。対象となるクラウドサービスは SaaS, PaaS, IaaS とする(表 1)。

表1 適用対象のクラウドサービスとSLA

クラウド階層	プロバイダ	サービス名	項目数	抽出数
SaaS	Google	Google Apps	6	1
PaaS	Salesforce.com	Force.com	6	1
IaaS	Amazon	Amazon EC2	6	1

(2) タイミングモデルの利用

タイミングモデル内のタイムトリガとイベントトリガをトリガと定義し、イベント指定と起動周期を起動条件と定義する。

(3) 監視タイミングの定義

上記(1)で抽出したプロパティと、上記(2)で定義したトリガと起動条件を利用し、測定と評価処理の観点から、監視タイミングを定義する。

8.2. 適用結果

8.1 節の手順に従い提案アーキテクチャを適用した結果を表2に示す。各プロパティは、プロバイダがコンシューマに対してサービスを提供すると測定を開始する。よって、イベントトリガを利用してサービスの実行状態の測定を行う。測定結果の評価はタイムトリガを利用して行う。稼働率、月間稼働率、年間稼働率の評価の時間間隔はそれぞれ日単位、月単位、年単位である。日単位の評価とは、評価する日から数えて1日分の測定結果を評価することを表す。月単位の評価とは、評価する日から数えて1月分の測定結果を評価することを表す。年単位の評価とは、評価する日から数えて1年分の測定結果を評価することを表す。

表2 適用結果

サービス名	プロパティ		監視タイミング	
			測定マネージャ	評価マネージャ
Google Apps	月間稼働率	起動条件	サービス提供イベント	月単位
		トリガ	イベントトリガ	タイムトリガ
Force.com	稼働率	起動条件	サービス提供イベント	日単位
		トリガ	イベントトリガ	タイムトリガ
Amazon EC2	年間稼働率	起動条件	サービス提供イベント	年単位
		トリガ	イベントトリガ	タイムトリガ

9. 評価と考察

9.1. SLA の多様化への対応

表2より実際のクラウドサービスのSLAが多様化していることを確認した。SLAの多様化により、SLAのモデル化が困難であるという課題を、WSLAに基づきメタモデルとして拡張SLAを形式的定義することで解決する。WSLAに基づき拡張SLAを形式的定義することで、SLAが機械可読可能となる。機械可読可能となったSLAは、監視タイミング、監視対象サービス、SLA管理のステークホルダ、保証SL

のスキーマを定義する監視データモデルとして表現できる。監視データモデルによって監視タイミング、監視対象サービス、SLA管理のステークホルダ、保証SLが特定可能となり、SL監視を自動化できる。SL監視の自動化によって、SLAの多様化に対応可能となる。

9.2. 監視タイミングを用いたSL監視への対応

表2より実際のクラウドサービスの監視対象となる各プロパティの監視タイミングが異なることを確認した。監視タイミングの差異によってプロパティの監視が競合するという課題を、タイミングモデル、マネージャ起動機能を用いることで解決する。タイミングモデルを用いることで、プロパティの監視タイミングを定義することが可能となる。マネージャ起動機能が、定義された監視タイミングを、測定マネージャと評価マネージャの起動に用いることで測定と評価の協調制御が可能となる。測定と評価の協調制御により、プロパティの監視の競合を軽減できる。

10. 今後の課題

10.1. SLAの変化への対応

クラウドにおいてSLAが変化することが想定される。例えば、プロバイダが提供するクラウドサービスの性能を上げた場合、SLAの変更をSLA管理に反映させる必要がある。

10.2. 違反検出方法を用いた違反通知対象者の特定

評価と違反通知処理の協調制御を行う場合、実行状態が保証SLに満たない原因を検出する処理が必要となる。

11. まとめ

本稿では、SLAのプロパティ毎にSL監視可能なSLA管理アーキテクチャを提案した。提案アーキテクチャをSaaS、PaaS、IaaSのSLAに適用し、アーキテクチャの有効性を示した。SLAメタモデルにより、SLAの多様化に対してSLAのモデル化が可能となり、提案アーキテクチャにより各プロパティの監視タイミングが異なってもSL監視が可能となる。

参考文献

[1] Amazon.com Int'l Sales, Inc., Amazon EC2, <http://aws.amazon.com/jp/ec2/>.
 [2] Google, Google Apps for Business, <http://www.google.com/intl/ja/enterprise/apps/business/>.
 [3] J. Happe, et al., A Reference Architecture for Multi-Level SLA Management, Springer, 2011.
 [4] H. Ludwig, et al., Web Service Level Agreement (WSLA) Language Specification, Jan. 2003, <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>.
 [5] 森下 月菜, ほか, SLAに基づいたクラウドサービス選択のための要求工学プロセス, SES 2012, Aug. 2012, pp. 1-8.
 [6] セールスフォースドットコム, Force.com, <https://www.salesforce.com/jp/>.