

Definición de una Estrategia de Pruebas basada en Acuerdos de Nivel de Servicio

Marcos Palacios¹, José García-Fanjul², Javier Tuya²

Departamento de Informática. Campus Universitario de Gijón. Universidad de Oviedo.
¹palacios@lsi.uniovi.es, ²{jgfanjul, tuya}@uniovi.es

Abstract. Durante la última década muchas empresas han optado por implantar sus procesos de negocio mediante el desarrollo de software basado en servicios y se espera que esta tendencia continúe durante los próximos años. En este contexto es habitual que las condiciones negociadas entre los proveedores de los servicios y los clientes estén especificadas de manera formal en un contrato que recibe el nombre de Acuerdo de Nivel de Servicio (SLA). Habitualmente se suelen utilizar técnicas de monitorización para determinar si los términos del SLA están siendo cumplidos o violados. Sin embargo es posible que, en ocasiones, los problemas no sean detectados al estar relacionados con situaciones que no se han ejercitado durante el período de monitorización. En este artículo se propone la definición de una estrategia de pruebas que, basándose en la especificación de los SLAs, permite identificar y priorizar situaciones de pruebas aplicando para ello una lógica cuatrivaluada.

Keywords: SLAs, lógica cuatrivaluada, monitorización, pruebas, servicios.

1 Introducción y Motivación

En el ámbito de la Ingeniería de Servicios es necesario que tanto los proveedores como los consumidores establezcan un conjunto de condiciones a cumplir en la ejecución de los servicios. Estas condiciones se especifican en un contrato o documento técnico que, en el contexto del paradigma XaaS (Todo como Servicio o Everything as a Service), recibe el nombre de Acuerdo de Nivel de Servicio (SLA – Service Level Agreement). La gestión de los SLAs incluye el proceso de elaboración y negociación de los términos así como la evaluación de las condiciones acordadas.

Una característica inherente al establecimiento de los SLAs es la capacidad que deben tener tanto proveedor como consumidor para poder determinar si las condiciones acordadas están siendo cumplidas o no. Para llevar a cabo esta toma de decisión es necesario observar el comportamiento del software durante su ejecución y obtener la información adecuada. Estas tareas son conocidas en el ámbito de las pruebas del software con el nombre de monitorización. Actualmente la mayoría de trabajos que abordan la prueba de SLAs usan esta técnica para detectar fallos en el software bajo prueba [9] aunque el hecho de que un sistema haya cumplido las condiciones del SLA durante el período monitorizado no garantiza que ese software esté libre de fallos. Por ejemplo, si un sistema se ha ejecutado con éxito un número

elevado de veces siempre bajo las mismas condiciones, el mecanismo de monitorización habrá reflejado el cumplimiento sistemático del SLA durante el período observado. Sin embargo, no es posible conocer el comportamiento del software en futuras ejecuciones cuando éstas se realicen en condiciones diferentes a las previamente observadas ni, por tanto, garantizar que se vaya a seguir cumpliendo lo acordado en el SLA.

Los enfoques basados en monitorización detectan los fallos en el software cuando éstos ya han aparecido en un entorno de producción (enfoques reactivos). Por este motivo, estos enfoques pueden ser adecuados en sistemas donde el descubrimiento y solución de problemas no representan un grave perjuicio para las partes implicadas en el SLA. Sin embargo, hay escenarios críticos donde la aparición de un problema puede acarrear consecuencias irreparables para los usuarios (por ejemplo, un sistema de salud para la gestión de alarmas médicas). En estos escenarios no se debe esperar a que los problemas se detecten en el entorno de producción sino que hay que anticipar su detección mediante la identificación, diseño y ejecución de las pruebas correspondientes (enfoques proactivos).

En este artículo se propone la definición de una estrategia de pruebas que permite la detección anticipada de problemas en escenarios críticos donde la monitorización puede resultar insuficiente. Esta estrategia identificará y priorizará un conjunto de situaciones que deberían ser ejercitadas mediante un adecuado diseño de pruebas. Para ello y tras analizar los actuales estándares para la especificación de SLAs, se utilizará una lógica cuatrivaluada para analizar las posibles situaciones que pueden darse durante el proceso de evaluación de un SLA.

La estructura del artículo será la siguiente: en la sección 2 se introducen los conceptos básicos sobre SLAs. En la sección 3 se describe el proceso de evaluación de los acuerdos. En la sección 4 se presenta la estrategia de pruebas propuesta. Finalmente, en la sección 5 se resumen las conclusiones y el trabajo futuro.

2 Acuerdos de Nivel de Servicio (SLAs)

La utilización de SLAs para gobernar la ejecución de software basado en servicios ha provocado que diferentes lenguajes hayan sido propuestos con la finalidad de estandarizar la especificación de los acuerdos. Entre estas propuestas se puede destacar WSLA [7] o SLAng [11], aunque ha sido WS-Agreement (WSAG) [1] quien últimamente ha recibido más atención. Un SLA especificado en WSAG se compone de un conjunto de *términos* que, combinados entre sí usando elementos *conectores* (equivalentes a los operadores lógicos), gobiernan el comportamiento del software al que hace referencia el acuerdo. Cada uno de estos términos incluye un *ámbito* (*Scope*) de aplicación, es decir, una lista de servicios cuya ejecución debe satisfacer las condiciones especificadas en dicho término. De la misma forma, un término contiene una *precondición* (*Qualifying Condition*) que gobierna la evaluación o no del término: si la precondición no se cumple, ese término no es tenido en cuenta durante el proceso de evaluación del SLA; por el contrario, en el caso de cumplirse la precondición el término tiene validez y la ejecución del software debe cumplir la condición especificada como *SLO* (*Service Level Objective*).

3 Evaluación de SLAs

La evaluación del SLA global requiere tomar una decisión sobre cada uno de los elementos incluidos en el acuerdo. Esta evaluación puede llevarse a cabo en un entorno de producción utilizando información obtenida mediante monitorización y, también, en un entorno de integración mediante la ejecución de un conjunto de casos de prueba. Habitualmente, la evaluación de un SLA se puede representar con un semáforo que, con dos únicas luces (verde y roja), indica si el SLA ha sido cumplido o no. A efectos de tomar esta decisión, es necesario combinar los resultados de la evaluación de cada elemento del SLA teniendo en cuenta los conectores utilizados en el acuerdo, pudiendo considerarse el *término* como el elemento más indivisible. En tiempo de ejecución y tras analizar la información obtenida, el término puede ser considerado *Cumplido* o *Violado*.

- Un término se considera ***Cumplido*** si y sólo si se han ejecutado los servicios especificados en el *ámbito* del término, se cumple la *precondición* que hace que el término tenga validez y se cumplen las condiciones especificadas en el *SLO*.
- Un término se considera ***Violado*** si y solo si se han ejecutado los servicios especificados en el *ámbito* del término, se cumple la *precondición* que hace que el término tenga validez pero no se cumplen las condiciones del *SLO*.

Además de los valores *Cumplido* y *Violado* es posible identificar y, de hecho, la especificación del estándar WS-Agreement lo hace, un tercer valor denominado *No Determinado*.

- Un término se considera ***No Determinado*** si y sólo si no se han ejecutado los servicios especificados en el *ámbito* del término.

La interpretación de este valor según WS-Agreement es que, en el momento de su evaluación, no ha habido actividad de los servicios asociados al término o éstos se encuentran en ejecución y aún no es posible determinar el valor exacto de la evaluación del término. Por tanto, representa situaciones que, desde el punto de vista de las pruebas, no han sido ejercitadas aún y necesitan ser probadas.

Tras estudiar los elementos internos de un término y su interpretación de acuerdo al estándar, es posible identificar una nueva situación en la que el término puede encontrarse tras su evaluación y que no ha sido identificada explícitamente en WS-Agreement. Esta situación se da cuando los servicios especificados en el *ámbito* del término se han ejecutado pero dichas ejecuciones no cumplen la *precondición* especificada, lo cual hace que el término no tenga validez y no deba ser aplicado a efectos de evaluar el SLA. Se puede decir entonces que el término tiene un valor *Inaplicable* y, al igual que ocurre con el valor *No Determinado*, es una situación de la que se desconoce su comportamiento en ejecuciones futuras.

- Un término se considera ***Inaplicable*** si y sólo si se han ejecutado los servicios especificados en el *ámbito* del término pero no se cumple la *precondición* que hace que el término tenga validez.

En la Fig. 1 se representa un ejemplo concreto de término de garantía especificado según el estándar WS-Agreement y su evaluación. En dicho ejemplo el término afecta a un servicio WSDoctor que recibe una alarma y debe proporcionar una respuesta en un tiempo máximo especificado. En el caso de que el servicio no se ejecutase, el término se considera *No Determinado*. Si el servicio se ejecuta pero no recibe una alarma de tipo Emergencia, el término no es válido y se evalúa como *Inaplicable*. Si el término se ejecuta y recibe una alarma de tipo Emergencia, el tiempo empleado en responder determinará si el término ha sido *Cumplido* o *Violado*.

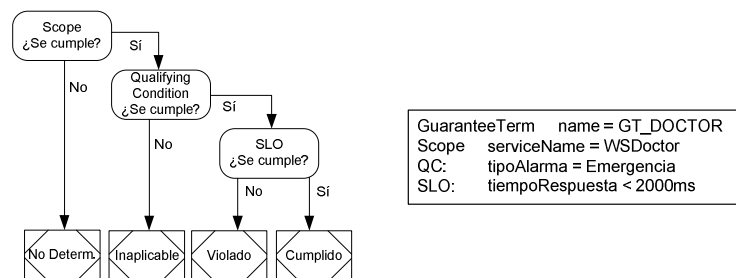


Fig. 1. Evaluación de un término de garantía especificado en WS-Agreement.

4 Propuesta de Estrategia de Pruebas

En el contexto de los SLAs, varios trabajos han sido propuestos para probar el contenido de los acuerdos usando tanto enfoques reactivos [8][6] como proactivos [4][10]. En este apartado se propone el uso de una estrategia que permite, a partir de la especificación de un SLA, identificar y priorizar situaciones de prueba. Para ello se define una lógica que permite evaluar el SLA teniendo en cuenta la evaluación de cada uno de sus términos. Tomando como base esa lógica se podrán definir criterios que permitan identificar un conjunto de situaciones de prueba y determinar el nivel de cobertura del SLA en función de las situaciones ejercitadas. En principio se podría pensar en el uso de una lógica convencional con dos valores de verdad: verdadero/falso (Cumplimiento/Violación). Sin embargo, tal como se describe en el apartado anterior, desde el punto de vista de las pruebas hay dos nuevas situaciones que deben contemplarse durante la evaluación del SLA. Es necesario, por tanto, incluir en la lógica estas dos evaluaciones adicionales donde los valores *No Determinado* e *Inaplicable* se corresponden con dos interpretaciones similares al tratamiento del valor nulo de la lógica trivaluada, ampliamente estudiada en el contexto de los Sistemas de Gestión de Bases de Datos (SGBD) y aplicada en el ámbito de las pruebas [2][3][5].

A partir de la especificación del SLA y teniendo en cuenta los posibles valores que pueden darse durante la evaluación, se pueden identificar los siguientes conjuntos de situaciones de pruebas (ordenados de mayor a menor prioridad).

1. El primer conjunto de situaciones a tener en cuenta es aquel en el que la evaluación del acuerdo ya debería detectar problemas en el software (el SLA se evaluaría como *Violado*). Estas situaciones son de máxima prioridad ya que

pueden acarrear consecuencias en el presente (por ejemplo, el pago de sanciones económicas) por lo que es necesario probar que los mecanismos de detección de violaciones del software bajo prueba funcionan adecuadamente y advierten a los usuarios de los problemas encontrados.

2. La evaluación de algún término con el valor *No Determinado* es también una situación muy interesante a probar ya que representa que los servicios asociados a dicho término no han sido ejecutados y, por tanto, se desconoce su comportamiento en futuras ejecuciones. Como ejemplo, supongamos un WSAG con dos términos incluidos en un conector *OneOrMore* (el SLA se cumple si al menos uno de esos términos se cumple) y, tras su evaluación, un término se ha *Violado* mientras el otro ha sido evaluado como *No Determinado*. En esa situación no es posible asegurar nada respecto a la evaluación del SLA por lo que es interesante ejercitar la situación en la que los servicios asociados al segundo término se ejecuten para, con un cierto nivel de confianza, tener la garantía que en futuras ejecuciones se van a cumplir las condiciones especificadas en ese término
3. La utilización del cuarto valor de esta lógica cuatervaluada permite identificar otro conjunto de situaciones de prueba interesantes. Un término evaluado como *Inaplicable* no afecta a la evaluación global del SLA dado que ese término no cumple la *precondición* y no tiene validez. Como ejemplo, supongamos que un conector *All* (todos los términos han de cumplirse) contiene varios términos asociados a un servicio que proporciona las medidas realizadas por diferentes aparatos médicos. La precondición de cada término especifica qué aparato está siendo consultado. Si durante la evaluación del SLA todos los términos se cumplen pero hay un término que se evalúa constantemente como *Inaplicable* significa que el aparato médico asociado a ese término nunca ha sido consultado. En este escenario, el SLA puede considerarse *Cumplido*. Sin embargo, es necesario diseñar y ejecutar pruebas que consulten al aparato en cuestión ya que la posible aplicación de ese término afecta directamente a la evaluación del SLA.

La identificación de las situaciones de prueba permite determinar el nivel de cobertura del SLA para un determinado conjunto de casos de prueba. Además de la ejecución de dichos casos, es posible monitorizar la ejecución de los servicios desplegados en operación para conocer cuáles de las situaciones han sido ya ejercitadas durante las ejecuciones observadas. La información obtenida de la monitorización puede utilizarse para definir criterios que permitan diseñar y ejecutar pruebas para las situaciones no ejercitadas.

5 Conclusiones y Trabajo Futuro

En este artículo se propone la definición de una estrategia de pruebas que permite identificar y priorizar, a partir de la especificación de un SLA, un conjunto de situaciones de prueba con el objetivo de detectar fallos. El contexto de aplicación de esta estrategia son aquellos sistemas de carácter crítico donde los problemas pueden ocasionar graves consecuencias a los usuarios y, también, en situaciones donde la monitorización no es suficiente para garantizar un determinado nivel de calidad del software observado. La estrategia hace uso de una lógica cuatervaluada para

representar los potenciales valores con los que puede ser evaluado un término del SLA.

Tras definir el planteamiento general de la estrategia, como trabajo futuro será necesario definir cómo se aplica la lógica presentada en los conectores que combinan jerárquicamente los términos del SLA para obtener diferentes niveles de cobertura. Además, se estudiará cómo complementar la ejecución de las pruebas en el entorno de integración con la utilización de técnicas de monitorización. Observando las ejecuciones del software desplegado en producción y aplicando criterios de cobertura del SLA, se podrán reducir, priorizar y seleccionar las situaciones a probar.

Por otra parte, los SLAs representan habitualmente aspectos no funcionales (p.ej: rendimiento). La estrategia propuesta podría ser aplicada para completar el proceso de pruebas cuando en el SLA se especifiquen características funcionales del software bajo prueba. Finalmente, la validación del enfoque supone un problema añadido debido a la falta de ejemplos reales de SLAs suficientemente complejos, tanto a nivel académico como industrial. Para mitigar esta limitación, se estudiará complementar el contenido de los SLAs con información funcional relevante para las pruebas.

Agradecimientos. Este trabajo ha sido financiado por el Programa Nacional de I+D+i del Ministerio de Ciencia e Innovación con el proyecto Test4DBS (TIN2010-20057-C03-01) y por el Gobierno del Principado de Asturias con la beca FICYT BP09-075.

References

1. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Services Agreement Specification (2007)
2. Belnap, N.D.: A Useful Four-valued Logic. In: J.M. Dunn, G. Epstein (eds.), *Modern Uses of Multiple-Valued Logic*, Dordrecht: Reidel, 8--37 (1977)
3. Codd, E.F.: *The Relational Model for Database Management - Version 2*. Addison-Wesley, Reading, MA, (1990)
4. Di Penta, M., Canfora, G., Espósito, G., Mazza, V., Bruno, M.: Search-based Testing of Service Level Agreements. In: 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 07), pp. 1090--1097, London, ACM, New York (2007)
5. Gessert, G.: Four Valued Logic for Relational Database Systems. *Sigmod Rec.* 19 (1), 29--35 (1990)
6. Leitner, P., Michlmayr, A., Rosenberg, F., Dustdar, S.: Monitoring, Prediction and Prevention of SLA Violations in Composite Services. In: 8th IEEE International Conference on Web Services (ICWS), pp.369--376, Miami, USA (2010)
7. Ludwig, H., Keller, A., Dan, A., Franck, R., King, R.P. *Web Service Level Agreement (WSLA) Language Specification*. IBM Corporation (2002)
8. Mahbub, K., Spanoudakis, G.: Monitoring WS-Agreements: an Event Calculus Based Approach. *Test and Analysis of Service Oriented Systems*, pp. 265--306, Springer V. (2007)
9. Palacios, M., García-Fanjul, J., Tuya, J.: Testing in Service Oriented Architectures with Dynamic Binding: A Mapping Study. *Information and Software Technology*, 53 (3), 171--189 (2011)
10. Palacios, M., García-Fanjul, J., Tuya, J.: A Proactive Approach to Test Service Level Agreements. In: 5th International Conference on Software Engineering Advances (ICSEA'10), pp. 454--458, Niza, Francia (2010)
11. Skene, J., Raimondi, F., Emmerich, W.: Service-Level Agreements for Electronic Services, *IEEE Transactions on Software Engineering*, 36 (2), 288--304, (March-April 2010)