Specifying Compensations with WS–Agreement

C. Müller, P. Fernández, O. Martín, A. M. Gutierrez, M. Resinas and A. Ruiz

Abstract— During the last years the use of service level agreements (SLA) is rising uncontrollably to describe the rights and obligations of parties involved in service provisioning (typically the service consumer and the service provider); amongst other information, SLA could define guarantees associated with the idea of service level objectives (SLOs) that normally represent key performance indicators of either the consumer or the provider. In case the guarantee is under or over fulfilled SLAs could also define some compensations (i.e. penalties or rewards). In such a context, there have been important steps towards the automation of the analysis of SLAs. One of these steps is a characterization model of SLAs with compensations proposed by the authors in a previous work; and another step is the standardisation effort in the SLAs notation made by WS–Agreement. However, real-world SLAs includes complex concepts that must be considered, namely: (i) SLA terms that specify compensations without an explicit SLO; and (ii) a limit for the compensations. In this paper we extend our prior characterization model considering these complex concepts. Specifically, (i) we provide up to five real-world scenarios whose SLAs incorporate aforementioned new concepts; (ii) we extend our model for compensable guarantees considering terms without an explicit SLO; and (iii) we provide a novel WS–Agreement-based syntax to model SLAs with compensations considering these concepts. These contributions aim to establish a foundation to elaborate tools that could provide an automated support to model and analyse SLAs with compensations.

Keywords— Compensation, Penalty, Reward, Service Level Agreement, SLA, WS–Agreement.

I. INTRODUCCIÓN

EN los últimos años el uso de los Acuerdos de Nivel de Servicio (ANS, o SLA en inglés) se encuentra en auge como mecanismo para describir los derechos y obligaciones de las partes en procesos de prestación de servicios. De esta manera, los ANS están aportando su granito de arena para alcanzar una prestación de servicios de calidad y confiable. Concretamente, los ANS se componen de diversos tipos de términos que definen garantías asociadas a ciertos Objetivos de Nivel de Servicio (ONS, o SLO en inglés) que se definen sobre métricas del servicio (ej.- tiempo de respuesta, tiempo de resolución, etc) y suelen ser asegurados por una parte (garante) a otra parte (beneficiario); en la mayor parte de los casos el primero corresponde con el proveedor y el segundo con el consumidor del servicio. Por ejemplo, Amazon, como proveedor de sus servicios web (AWS), garantiza una disponibilidad $\geq 99.95\%$ como ONS general.

Además, los ANS comerciales suelen incluir un conjunto de compensaciones que se corresponden con las consecuencias del incumplimiento (penalización), o sobrecumplimiento (recompensa) del ONS. Por ejemplo, Amazon define una penalización si la disponibilidad de su servicio cae del 99.95%. En un artículo previo [1] se acuñó el concepto de ANS compensable para referirnos a aquellos ANS que incluyen al menos una acción de compensación, ya sea una penalización o una recompensa.

En este contexto se han dado numerosos e importantes avances hacia la automatización del análisis de ANSs. Uno de dichos avances es un modelo que caracteriza los ANSs con compensaciones propuesto por los autores del presente artículo en [1]; y otro avance es el esfuerzo de estandarización de la descripción de ANSs realizado por WS–Agreement. Sin embargo, los ANSs comerciales suelen estar descritos concienzudamente para evitar problemas futuros y, por lo tanto, a menudo incluyen conceptos complejos que deben ser considerados en los modelos y descripciones de ANSs. Por ejemplo, (i) los términos de garantía que describen algunos gobiernos regionales de Canadá y España suelen especificar compensaciones sin definir explícitamente un ONS asociado; y (ii) la mayor parte de las compañías suelen definir un límite para las compensaciones del servicio. Por ejemplo, Amazon limita la penalización de su servicio "Elastic Cloud service" (AWS EC2) al 10% de la factura mensual pagada por usarlo (<http://aws.amazon.com/es/ec2/sla/>).

En el presente artículo proponemos un paso determinante hacia la automatización del análisis de ANSs compensables que consiste en dar soporte a los conceptos complejos mencionados anteriormente. En primer lugar, extendemos nuestro modelo de caracterización de ANSs incluyendo la posibilidad de definir valores umbrales óptimos de las métricas del servicio para definir de manera óptima las compensaciones. Dichos umbrales pueden ser inferidos automáticamente y tienen en cuenta si el ANS ha sido definido por el garante o el beneficiario. Estos umbrales son usados para validar las compensaciones en aquellas garantías que no especifican explícitamente un ONS. En segundo lugar, proponemos una notación novedosa y fácil de entender para el usuario que se corresponde con una versión actualizada de la sintaxis del lenguaje iAgree, basado en WS–Agreement [3] [4], [5]. El principal motivo para usar iAgree es poder beneficiarse de las herramientas de soporte que ya están desarrolladas para dicho lenguaje y que incorporan el chequeo automático de consistencia y validez, aunque sólo aplicado a los ONS y no a la definición de compensaciones [4].

C. Müller, E.T.S.I.I., Univ. de Sevilla, España, cmuller@us.es.
P. Fernández, E.T.S.I.I., Univ. de Sevilla, España, pablofm@us.es.
O. Martín, E.T.S.I.I., Univ. de Sevilla, España, omartindiaz@us.es.

A. M. Gutierrez, E.T.S.I.I., Univ. de Sevilla, España, amgutierrez@us.es.
M. Resinas, E.T.S.I.I., Univ. de Sevilla, España, resinas@us.es.
A. R. Cortés, E.T.S.I.I., Univ. de Sevilla, España, aruiz@us.es.
Corresponding autor: Carlos Müller.

Este artículo está estructurado de la siguiente manera: la sección II resume el modelo de caracterización de compensaciones propuesto en [1]. Dicho modelo es extendido con la definición de los umbrales óptimos en la sección III. En la sección IV proponemos una extensión del lenguaje iAgree para modelar las compensaciones. Nuestra propuesta de modelado es validada en cinco escenarios reales en la sección V. La sección VI analiza el estado del arte e identifica los trabajos relevantes que tratan con compensaciones en ANSs. Finalmente, en la sección VII comentamos algunas conclusiones y el trabajo que queda por hacer.

II. RESUMEN DEL MODELO DE COMPENSACIONES

En esta sección resumimos el modelo de compensaciones propuesto en [1], que está compuesto por: (i) la conceptualización de la función de compensación para expresar de manera conjunta y consistente tanto las penalizaciones como las recompensas; y (ii) un modelo de garantías compensables que asocian los ONSs con las funciones de compensación.

Funciones de Compensación

Las Funciones de Compensación (FC) se definen sobre métricas del servicio en el contexto de una garantía otorgada por un garante a un beneficiario. Concretamente, asocian dos tipos de compensaciones dependiendo de quién compensa a quién: por un lado, una penalización representa una compensación de un garante a un beneficiario y, por otro lado, una recompensa representa una compensación de un beneficiario a un garante. A continuación, formalizamos las FCs a través de un conjunto de definiciones básicas:

Definición 1 (Valores de las métricas): El conjunto M_m representa el conjunto de todos los valores posibles de una métrica m ($M_m = \{v_1, \dots, v_n\}$).

Definición 2 (Función de Utilidad): Una Función de Utilidad para una métrica específica, representado como FU_m , es una función de M a $\mathbb{R}$ que asocia una utilidad a cada uno de los valores de dicha métrica; e.d. define qué valores de la métrica M_m son más interesantes para una de las partes.

Definición 3 (Precedencia de Utilidad): Siendo v_1 y v_2 valores del conjunto M_m de una métrica m , y FU_m una función de utilidad definida sobre la misma métrica; se define una relación de precedencia llamada Precedencia de Utilidad sobre M_m por FU_m . Y representamos que v_1 es menos interesante que v_2 de la siguiente manera: $v_1 \boxtimes v_2$.

Definición 4 (Función de Compensación): Una Función de Compensación para una métrica determinada m , representado por FC_m , es una función de M a $\mathbb{R}$ que asocia una compensación para cada uno de los valores de la métrica.

Definición 5 (Regiones de Compensación): Una función de compensación de una métrica determinada FC_m define hasta tres regiones distintas de compensación:

$$\begin{aligned} \text{Penalizada}(FC_m) &= \{v_i \in M_m \cdot FC_m(v_i) > 0\}, \\ \text{Neutral}(FC_m) &= \{v_i \in M_m \cdot FC_m(v_i) = 0\}, \text{ y} \\ \text{Recompensada}(FC_m) &= \{v_i \in M_m \cdot FC_m(v_i) < 0\}. \end{aligned}$$

La Fig. 1 muestra una función de compensación típica con las tres regiones mencionadas.

La validez de las FCs se formaliza como una propiedad relacionada con: (i) su consistencia con la función de utilidad; y (ii) la saturación de las compensaciones.

Definición 6 (Consistente_{FC}): Se dice que una función de compensación FC_m es consistente si un valor menos interesante de su métrica tiene mayor o igual compensación que un valor más interesante de acuerdo con la precedencia de utilidad definida por la función de utilidad del beneficiario.

$$\text{Consistente}_{FC}(FC_m) \iff \forall v_1, v_2 \in M \cdot v_1 \leq v_2 \Rightarrow FC_m(v_1) \geq FC_m(v_2)$$

Definición 7 (Saturada): Se dice que una función de compensación FC_m está saturada si existen dos valores (v_{min} y v_{max}) para la métrica, que delimitan la mayor compensación, ya sea penalización o recompensa.

$$\text{Saturada}(FC_m) \iff \forall v_i \in M, \exists v_{max}, v_{min} \in M \cdot FC_m(v_i) \leq FC_m(v_{max}) \wedge FC_m(v_i) \geq FC_m(v_{min})$$

Definición 8 (Valida_{FC}): Se dice que una función de compensación FC_m es válida si es consistente y saturada. $\text{Valida}_{FC}(FC_m) \iff \text{Consistente}_{FC}(FC_m) \wedge \text{Saturada}(FC_m)$

Garantías Compensables

Uno de los elementos principales de los ANS son los términos de garantía [3] que suelen estar definidos en torno a un ONS. Basándonos en esta conceptualización, acuñamos el concepto de garantía compensable (GC) para referirnos a aquellas garantías que incluyen una FC. Por tanto, un ANS compensable es todo ANS que incluye al menos una GC.

Definición 9 (Objetivo de Nivel de Servicio): Un ONS_m es una condición válida (En [4] se formaliza el criterio de validez para los ONSs.) definida sobre una métrica m .

Definición 10 (Regiones de Cumplimiento): La condición definida por un ONS_m determina dos regiones sobre los valores de las métricas con la que trata: $\text{Cumplida}(ONS_m) = \{v_i \in M_m \cdot ONS_m\}$, e $\text{Incumplida}(ONS_m) = \{v_i \in M_m \cdot \neg ONS_m\}$. Estas regiones son delimitadas por un umbral U_{ONS} .

Definición 11 (Garantía Compensable): Una Garantía Compensable GC_m es una tupla (ONS_m, FC_m) en la que ONS_m es un objetivo de nivel de servicio y FC_m es una función de compensación, ambos definidos sobre la misma métrica m . $GC_m = \langle FC_m, ONS_m \rangle$

La Fig. 1 muestra las relaciones existentes entre las regiones de cumplimiento definidas por el $ONS(U_{ONS})$ y las regiones de compensación definidas por la FC . La validez de las GCs se formaliza como una propiedad relacionada con la consistencia entre su ONS y su FC .

Definición 12 (Consistente_{GC}): Se dice que una garantía compensable GC_m es consistente si existe al menos un valor neutro de la métrica y que cumple el ONS (que sería el umbral U_{ONS}) y las regiones de cumplimiento son coherentes con las de compensación. Es decir, los valores que cumplen el ONS deben ser o neutros o recompensados, y de manera complementaria, los valores que no cumplen el ONS deben ser neutros o penalizados.

Definición 13 (Valida_{GC}): Siendo GC_m una garantía compensable, se dice que es válida si es consistente y contiene una FC válida.

$$\text{Valida}_{GC}(GC_m) \iff \text{Valida}_{FC}(GC_m \cdot FC) \wedge \text{Consistente}_{GC}(GC_m)$$

Figura 1. FC genérica con GC, incluyendo las regiones de cumplimiento y compensación.

III. EXTENSIÓN DEL MODELO DE GARANTÍAS COMPENSABLES

Algunas garantías compensables que se pueden encontrar en ANS públicos o comerciales no expresan de manera explícita ningún ONS, sino que la garantía es expresada en términos de las compensaciones asociadas a ellos (ver los escenarios GNWT y IT_AHS en la sección V). En esos casos se pueden inferir dos umbrales óptimos para delimitar las regiones de cumplimiento. Nótese que estos umbrales son óptimos con respecto a las compensaciones, pero no necesariamente con respecto al coste o cualquier otro aspecto del ONS.

Definición 14 (Umbrales Óptimos): Siendo GC_m una garantía compensable sin definir un ONS explícito ONS_m , se pueden inferir dos umbrales óptimos: (i) el umbral óptimo para el garante (U_{Gte}) que corresponde con el valor de la métrica que implica una menor compensación para el beneficiario, sin incurrir en una penalización (ej.- el valor b de la Figura 1); y (ii) el umbral óptimo para el beneficiario (U_{Ben}) que corresponde con el valor de la métrica que implica una mayor compensación para el beneficiario, sin incurrir en una recompensa (ej. el valor a de la Fig. 1).

$$U_{Gte} = v_j | \forall v_i, v_j, v_k \in M \cdot v_i \leq v_j \boxtimes v_k \wedge FC_m(v_i) \leq 0 \wedge FC_m(v_j) = 0 \wedge FC_m(v_k) > 0$$

$$U_{Ben} = v_j | \forall v_i, v_j, v_k \in M \cdot v_i \boxtimes v_j \leq v_k \wedge FC_m(v_i) < 0 \wedge FC_m(v_j) = 0 \wedge FC_m(v_k) \geq 0$$

Estos umbrales óptimos son necesarios para comprobar la validez de las garantías compensables cuando estas no definen explícitamente el ONS. Además, los umbrales óptimos son también útiles para comprobar si el ONS definido en una garantía compensable ha sido definido de manera óptima o no respecto de las compensaciones. Es decir, un ONS está definido de manera óptima si el ONS y el umbral óptimo delimitan las mismas regiones de cumplimiento.

```

Agreement_Name:
...
context:
  provider: //Nombre del Proveedor
  consumer: //Nombre del Consumidor
...
definitions:
  schemas:
    p/r: //variables de penalización/recompensa
        //incluye descripción, tipo, y unidad
...
terms:
  pricing:
    billing:
      period: //monthly, weekly, ...
      initial: //instante ej.-'2016-05-12T10:35:36.000Z'
    penalties:
      - over: p
    rewards:
      - over: r
  metrics:
    m: //definición de la métrica
      schema: //incluye descripción, tipo, y unidad
    guarantees: //Garantía compensable (GC)
      - id: //GT_id
      of:
        - objective: CondSLO(m) //ONS definido sobre m
          precondition: //habilita el ONS
          window:
            period: //monthly, weekly, ...
            initial: //instante
          penalties: //Penalización de la FC
            - over: p
              of:
                - value: AsigP(p) //asignación de p
                  condition: CondP(m) //habilita penaliza.
            - upTo: limit
          rewards: //Recompensa de FC
            - over: r
              of:
                - value: AsigR(r) //asignación de r
                  condition: CondR(m) //habilita recompensa
            - upTo: limit

```

Figura 2. Esquema de garantías compensables en sintaxis iAgree.

IV. EXTENSIÓN DE IAGREE CON COMPENSACIONES

El principal motivo para extender el lenguaje iAgree para modelar las compensaciones es la posibilidad de desarrollar una nueva versión de sus herramientas de soporte. Actualmente iAgree es soportado por IDEAS (<http://www.isa.us.es/IDEAS/>), una aplicación basada en servicios web que permite, entre otras cosas, analizar y editar de manera sencilla los ANSs.

La Fig. 2 muestra un esquema de la notación de iAgree (se muestra una versión actualizada respecto a la incluida en [4] y [5]) con la extensión que proponemos para modelar las compensaciones. Dicha notación permite expresar GCs como guarantees que incluyen tanto los ONS definidos sobre las métricas (ver CondSLO(m)), como las FCs. La FC se especifica por medio de sus penalizaciones y recompensas de manera separada. Concretamente, las penalizaciones y recompensas se definen mediante varios elementos: (1) sus variables (ver p and r), (2) sus asignaciones de valores (ver AsigP(p) y AsigR(r)), y (3) las condiciones que habilitan las compensaciones (ver CondP(m) y CondR(m)). Las figuras 3 y 5 incluidas en la siguiente sección muestran garantías compensables descritas en base a esta extensión de iAgree.

Figura 4. Garantía Compensable de los ejemplos de GNWT.

se le aplica una recompensa, y si ninguno de dichos problemas se resuelve en menos de 4 horas se le aplica una penalización. Este ANS también incluye una garantía, la GNWT-4, sobre la fecha prevista para la entrega de los proyectos y la fecha real de la entrega del proyecto. Dicha garantía define una penalización para el proveedor si los días transcurridos hasta la entrega son más de un 20% más de lo planificado; pero también incluye una recompensa para el proveedor si los días transcurridos hasta la entrega son exactamente un 20% menos de los planificados. En este caso, la recompensa no está definida de manera correcta [1]. Aunque el ONS no está explícito en este caso, el documento iAgree de la Fig. 5 incluye el umbral óptimo inferido para el garante.

ANS de IT_AHS: El gobierno regional de Andalucía, en España, también subcontrata los servicios de soporte de TI para el servicio público de salud. El ANS que deben seguir los proveedores es público (<http://ow.ly/4mO6cB>) e incluye garantías sobre los servicios requeridos además de ciertas compensaciones. Por ejemplo, se establece que, si la disponibilidad del servicio de mejora continua de las TI es menor del 95%, se aplica una penalización que depende del precio/hora del perfil de recurso humano demandado. Además, el ANS limita las penalizaciones o bien al 25% de los gastos mensuales o bien al 10% del presupuesto total destinado al servicio de soporte de TI.

ANS del servicio de Verizon Terremark Cloud Backup: Verizon Terremark ofrece diversos servicios web, entre los que se encuentra el servicio de copia de seguridad en la nube (<http://ow.ly/4mORhQ>).

```
GNWT SLA:
id: GNWT
version: 1.0
type: agreement
context:
  provider: Provider
  consumer: Gobierno del noroeste de Canadá
  definitions:
    schemas:
      InvoicePercentage:
        description: Porcentaje para factura mensual
        type: integer... unit: '%'
  terms:
    pricing:
      billing:
        period: monthly
        initial: '2016-05-12T10:35:36.000Z'
      penalties:
        - over: InvoicePercentage
      rewards:
        - over: InvoicePercentage
    metrics:
      elapsedDaysPercent:
        schema:
          description: Días pasados hasta la entrega (%)
          type: integer... unit: '%'
      resolutionHours:
        schema:
          description: Horas para resolver incidencias
          type: integer... unit: 'hours'
      severityType:
        schema:
          description: Tipo de incidencia (1-4)
          type: integer... unit: ''
    guarantees:
      - id: GNWT-2
        of:
          - objective: resolutionHours <= 4
            precondition: severityType = 1
            window:
              type: static
              period: monthly...
            penalties:
              - over:
                  InvoicePercentage:
                    of:
                      - value: '-10'
                        condition: resolutionHours > 4
            rewards:
              - over:
                  InvoicePercentage:
                    of:
                      - value: '10'
                        condition: resolutionHours < 2
      - id: GNWT-4
        of:
          - objective: elapsedDaysPercent < 120
            window:
              type: static
              period: monthly...
            penalties:
              - over:
                  InvoicePercentage:
                    of:
                      - value: '-10'
                        condition: elapsedDaysPercent >= 120
            rewards:
              - over:
                  InvoicePercentage:
                    of:
                      - value: '5'
                        condition: elapsedDaysPercent == 80
```

Figura 5. Garantía Compensable del escenario GNWT en iAgree

El ANS (<http://ow.ly/4mOFFd>) especifica una disponibilidad mayor del 99.9% bajo la aplicación de una penalización de un crédito de servicio (definido como 1/30 de la factura mensual del servicio) al que se sumará un crédito más por cada 100

minutos adicionales en los que la disponibilidad del servicio no sea reestablecida. Sin embargo, dicha penalización no podrá sobrepasar el 50% de la factura mensual pagada por el servicio. Lo que indica que, aunque el servicio no esté disponible en ningún momento del mes el cliente deberá pagar la mitad del mes.

VI. TRABAJO RELACIONADO

Hasta donde sabemos, no existe ninguna propuesta para modelar las compensaciones de los ANSs, que permita automatizar su análisis.

La propuesta de Leitner et al en [6] formaliza el problema de encontrar un conjunto óptimo de adaptaciones que minimice los costes totales debido a las violaciones del ANS y las adaptaciones que permitan prevenirlas. En este trabajo, se presenta un modelo de funciones de penalizaciones que hemos utilizado como punto de partida para caracterizar el modelo de compensaciones presentado en [1] (y que hemos resumido en la Sección II). En [7], los mismos autores presentan una aproximación para la programación óptima de peticiones a recursos computacionales virtuales en la nube, de manera que se minimicen tanto el pago de los recursos como las pérdidas debidas a las violaciones del ANS. El ejemplo incluido en este trabajo relaciona la penalización con la métrica del servicio que representa la duración de las peticiones a los recursos computacionales virtuales en la nube.

Otros ejemplos revisados son: Buco et al. proponen en [8] un sistema de gestión de ANSs llamado SAM, que define penalizaciones para la gestión del nivel de servicio. Grabarnik et al. proponen en [9] un modelo para reducir el coste total de la provisión de servicios de TI usando equipos alternativos y proveedores de servicio externos. Rana et al. identifican en [10] como los ONSs se pueden ver afectados por la elección de cláusulas específicas de penalización. Paschke et al., [11], modelan el ANS para automatizar su gestión. Este ANS define los límites máximo y mínimo para compensar el incumplimiento o el sobrecumplimiento del ANS, pero esta compensación se hace mediante llamadas a eventos de manera personalizada.

En el estudio de los modelos de negocio, se analizan modelos de funciones de utilidad que dependen del comportamiento y las preferencias de los clientes. Por ejemplo, en [12] se describe un escenario de negocio con variables de coste, expectativas del cliente y reputación donde la función de compensación sigue un comportamiento no monótono (que depende del cumplimiento de las preferencias de los distintos clientes). De igual manera, Fenghui Ren et al. analizan en [13], como obtener la función de utilidad a partir de la función objetivo del cliente (ej. como las preferencias de horario de los clientes afectan a la distribución de transacciones durante el horario comercial).

Angelov et al. proponen, en [14], una representación formal para los contratos que permite detectar y solucionar ciertos conflictos. Aunque el modelo propuesto soporta penalizaciones y recompensas mediante términos de reparación, no se comprueba la validez de los mismos frente a las funciones de utilidad, tal como se propone en el presente trabajo.

VII. CONCLUSIONES Y TRABAJO FUTURO

En este artículo proponemos una manera de modelar ANSs compensables extendiendo iAgree, un lenguaje [4], [5] basado basado en WS-Agreement, [3]. La notación extendida soporta: (i) Un modelo de compensaciones propuesto anteriormente en [1]; (ii) la definición de límites de compensaciones, que es una práctica común, como se muestra en los cinco escenarios de ANSs reales incluidos. También se han considerado aquellos casos en los que no existen ONSs explícitos en las garantías del ANS. De esta manera, hemos definido los umbrales óptimos del ONS que se pueden inferir en función de si el ANS fue especificado por su beneficiario o su garante.

Esta contribución constituye un importante paso adelante en la automatización del análisis de ANSs compensables porque iAgree tiene un conjunto de herramientas de soporte públicas denominado IDEAS. Además, se provee un espacio de trabajo público en IDEAS con todos los ejemplos de este artículo, modelados con iAgree. Como trabajo futuro, se actualizarán las operaciones de análisis de IDEAS para validar tanto las funciones de compensación como los términos de garantía compensables, de manera que se detecten situaciones singulares o errores comunes.

VIII. AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por proyectos de investigación españoles y andaluces (TIN2014-53986-REDT, P12-TIC-1867, TIN2015-70560-R).

REFERENCIAS

- [1] Carlos Müller et al., Towards a Formal Specification of SLAs with Compensations, OTM Conferences: CoopIS, Amantea, Italy, 2014, pp. 295–312. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-45563-0_17
- [2] Carlos Müller et al., “Supporting compensations with ws-agreement, xii jornadas de ciencia e ingeniería de servicios (jcis): Salamanca, España,” 2016.
- [3] Andrieux et al., “Web Services Agreement Specification (WS-Agreement) (v. gfd-r.192),” 2011, OGF - Grid Resource Allocation Agreement Protocol WG.
- [4] C. Müller, M. Resinas, and A. Ruiz-Cortés, “Automated analysis of conflicts in ws-agreement,” *IEEE Transactions on Services Computing (TSC)*, 2013.
- [5] C. Müller, “On the Automated Analysis of WS-Agreement Documents. Applications to the Processes of Creating and Monitoring Agreements,” International dissertation, Universidad de Sevilla, 2013. [Online]. Available: <http://www.isa.us.es/sites/default/files/muller-Phd-PTB.pdf>
- [6] P. Leitner, W. Hummer, and S. Dustdar, “Cost-based optimization of service compositions,” *Services Computing, IEEE Transactions on*, vol. 6, no. 2, pp. 239–251, April 2013.
- [7] P. Leitner et al., “Cost-Efficient and Application SLA-Aware Client Side Request Scheduling in an Infrastructure-as-a-Service Cloud,” in *Cloud Computing (CLOUD)*, IEEE 5th International Conference on, June 2012, pp. 213–220.
- [8] M.J. Buco et al., “Utility computing sla management based upon business objectives,” *IBM Systems Journal*, vol. 43, no. 1, pp. 159–178, 2004.
- [9] G. Grabarnik et al., “Management of service process qos in a service provider - service supplier environment,” in *IEEE Int. Conf. on Ent. Comp., E-Commerce, and E-Services. (CEC/EEE)*, July 2007, pp. 543–550.
- [10] O. F. Rana, M. Warnier, T. B. Quillinan, F. Brazier, and D. Cojocararu, “Managing violations in service level agreements,” in *Grid Middleware and Services Chapter Title - Managing Violations in Service Level Agreements*, 2008, pp. 349–358.
- [11] A. Paschke and M. Bichler, “Knowledge representation concepts for automated SLA management,” 23 Nov. 2006.

- [12] H. Bar-Isaac and J. Deb, "What is a good reputation? career concerns with heterogeneous audiences," *Journal of Industrial Organization*, vol. 34, no. 0, pp. 44 – 50, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167718714000253>
- [13] F. Ren and M. Zhang, "Bilateral single-issue negotiation model considering nonlinear utility and time constraint," *Decision Support Systems*, vol. 60, no. 0, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167923613001668>
- [14] Krasimir Angelov et al., "A framework for conflict analysis of normative texts written in controlled natural language," *The Journal of Logic and Algebraic Programming*, vol. 82, no. 57, pp. 216 – 240, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1567832613000143>.

Carlos Müller es profesor a tiempo completo y cada (ISA, www.isa.us.es) en la Universidad de miembro del grupo de Ingeniería del Software Apli- Sevilla, España. Obtuvo su doctorado en Ingeniería del Software en dicha Universidad. Su línea de investigación actual incluye computación orientada a servicios, centrado en el análisis automático de acuerdos de nivel de servicio y su aplicación en el diseño y monitorización de los acuerdos.

Pablo Fernández es profesor a tiempo completo y miembro del grupo de Ingeniería del Software Aplicada (ISA, www.isa.us.es) en la Universidad de Sevilla, España. Obtuvo su doctorado en Ingeniería del Software en dicha Universidad. Su línea de investigación actual está centrada en el gobierno automático de organizaciones basadas en compromisos y acuerdos de nivel de servicio.

Octavio Martín-Díaz es profesor a tiempo completo y miembro del grupo de Ingeniería del Software Aplicada (ISA, www.isa.us.es) en la Universidad de Sevilla, España. Obtuvo su doctorado en Ingeniería del Software en dicha Universidad. Su línea de investigación actual incluye el estudio de técnicas de compra en computación en la Nube y la computación orientada a servicios, centrándose en la gestión del tiempo en acuerdos de nivel de servicio.

Antonio M. Gutierrez es estudiante de Doctorado y miembro del Grupo de Ingeniería del Software Aplicada (ISA, <http://www.isa.us.es>) en la Universidad de Sevilla, España. Tiene un grado y un máster en Ingeniería del Software. Trabajó como ingeniero para diferentes compañías antes de unirse a la academia. Su línea de investigación aborda la computación orientada a servicios y la gestión de procesos de negocio.

Manuel Resinas es profesor a tiempo completo en la Universidad de Sevilla, España. Obtuvo su doctorado en Ingeniería del Software en dicha Universidad. Su línea de investigación actual incluye el análisis y la gestión de acuerdos de nivel de servicio, conformidad en procesos de negocio, y gestión del rendimiento en procesos de negocio. Anteriormente trabajó en la negociación automática de acuerdos de nivel de servicio.

Antonio Ruiz-Cortés es catedrático e investigador principal del grupo de Ingeniería del Software Aplicada (ISA, www.isa.us.es) en la Universidad de Sevilla, España. Obtuvo su doctorado con honores en Ingeniería del Software en dicha Universidad. Sus líneas de investigación actuales incluyen la computación orientada a servicios, las líneas de productos software, y la gestión de procesos de negocio. Anteriormente trabajó en ingeniería de requisitos e interacciones multiparte.