

ENFOQUES EMERGENTES DE INGENIERÍA DE SOFTWARE APLICADOS A GRID COMPUTING

Natalia B. Trejo y Sandra I. Casas
Unidad Académica Río Gallegos, Universidad Nacional de la Patagonia Austral
Río Gallegos, Argentina, 9400
ntrejo@uarg.unpa.edu.ar, lis@uarg.unpa.edu.ar

J. Baltasar García Perez-Schofield
Departamento de Informática, Universidad de Vigo
Orense, España, 32004
jbgarcia@uvigo.es

RESUMEN

Las aplicaciones para Grid computing basadas en servicios Grid, pueden beneficiarse con el empleo de algunos de los emergentes enfoques de ingeniería de software. Los servicios Web, de los cuales extienden los servicios Grid, son intensamente investigados con el objetivo de desarrollar soluciones basadas en SOA que permitan reutilizarlos, evolucionar y soportar cambios de manera automatizada. Analizamos de qué manera los nuevos enfoques de ingeniería de software se aplicaron en el campo de Grid computing y proponemos el empleo de uno de ellos en la composición de servicios Grid.

Palabras clave: Grid services, feature interaction, grid computing

1. INTRODUCCION

Grid computing [FK2004], como nuevo paradigma de computación distribuida, permite gestionar y utilizar de forma segura y coordinada recursos heterogéneos (datos, almacenamiento, aplicaciones de software, servicios, redes, sensores, equipamiento científico, clústeres, etc.) que pueden encontrarse distribuidos geográficamente y pertenecer a diferentes dominios administrativos. Los usuarios obtienen una visión unificada de dichos recursos y pueden acceder a ellos de manera transparente, esto facilita el trabajo colaborativo a gran escala. Grid computing permite alcanzar de forma segura y económica capacidades computacionales para ejecutar eficientemente aplicaciones intensivas en datos ó computación.

Por otro lado, el paradigma de Computación Orientada a Servicios (*Service Oriented Computation, SOC*) surge para facilitar el desarrollo de aplicaciones distribuidas a gran escala, independientes de la plataforma y a bajo costo, facilitando la automatización de procesos de negocio a través de elementos básicos de construcción llamados servicios. La arquitectura de software para este tipo de aplicaciones es conocida como Arquitectura Orientada a Servicios (*Service Oriented Architecture, SOA*), en la cual los servicios Web proveen un marco de implementación.

Los servicios Web son componentes de software diseñados para proveer operaciones específicas

(“servicios”) que se encuentran accesibles usando tecnologías estándares de Internet como XML y protocolos de red a la vez que son independientes de la plataforma. Son de interés para Grid computing ya que permiten implementar la infraestructura Grid a través de la comunicación entre componentes Grid distribuidos y también como interfaz uniforme para que los usuarios puedan acceder a los recursos Grid. Un escenario para la utilización de servicios Web es usarlos como front-end de aplicaciones existentes que operen sobre una plataforma Grid, esto se realiza encapsulando el código de la aplicación para producir un servicio Web y accediendo a ella mediante la interfaz de dicho servicio.

Los servicios Web normalmente son *stateless*, es decir, que no conservan información entre una invocación y la siguiente. En las aplicaciones para Grid computing es de utilidad que los servicios Web sean *stateful*, es decir, que conserven información entre invocaciones y entre invocaciones realizadas por diferentes clientes para permitir que una secuencia de acciones a ejecutarse pueda depender de acciones y resultados anteriores. Dicha información o estado se almacena en uno o varios *recursos*, los que se encuentran separados del servicio Web, el que actúa como su front-end. Esta extensión de servicios Web para Grid computing se conoce como servicios Grid o servicios WSRF (*Web Service Resource Framework*) [FFG2004] que se basan en el estándar OGSA (*Open Grid Services Architecture*) [FKS2005].

Los desarrolladores de aplicaciones para Grid computing y los usuarios que despliegan y ejecutan aplicaciones en entornos Grid, precisan herramientas Grid en el nivel de aplicación [FK2004]. Es decir, software que se construya y utilice la infraestructura Grid proveyendo nuevas funcionalidades y abstracciones de alto nivel para facilitar las tareas de los desarrolladores y usuarios de Grid computing. Estas herramientas se sitúan sobre otros servicios Grid, en [FK2004] se presenta una taxonomía de ellas, entre las que se encuentran aquellas relacionadas con los entornos de ejecución de aplicaciones Grid, en particular las conocidas como workflows.

Un workflow puede ser definido como la automatización de un proceso, formado por varios participantes, llamados sub-procesos, donde la información o datos se transfieren entre ellos basándose

en un conjunto de reglas predefinidas para lograr un objetivo. Desde el punto de vista de Grid computing, una tarea en un Grid puede componerse a su vez de varias sub-tareas que frecuentemente tienen dependencias entre ellas.

Si consideramos los servicios Grid que forman la infraestructura Grid en muchos middleware, como Globus Toolkit 4, un nuevo servicio Grid puede estar formado a partir de la combinación de dos o más servicios Grid ya existentes. En este caso, un workflow define el orden de ejecución de estos servicios dentro del nuevo servicio Grid y los mensajes intercambiados entre ellos. Por lo tanto podría definirse un workflow Grid, como la automatización de una colección de servicios mediante la coordinación a través de dependencias de control y datos para resolver un problema específico ó generar un nuevo servicio Grid.

Grid computing se utiliza ampliamente en comunidades científicas relacionadas con bioinformática, astronomía, física de altas energías, etc. Tales aplicaciones requieren la orquestación de diversos recursos Grid y la ejecución de complejos workflows. Las ventajas de los workflows son diversas [YB2005], permitiendo la construcción de aplicaciones dinámicas a partir de recursos heterogéneos distribuidos y provenientes de diferentes dominios administrativos.

Al existir tal afinidad natural entre workflows y Grid computing y al ser los servicios Grid extensión de los servicios Web, existe un creciente interés en el estudio de cómo los nuevos y emergentes paradigmas de ingeniería de software (IS) pueden mejorar el desarrollo de aplicaciones basadas en servicios Grid. Presentamos un estudio de los principales enfoques emergentes de IS aplicados a Grid computing y la propuesta de utilización de uno de ellos para resolver el llamado *feature interaction problem (FIP)* [CKM2003] que puede surgir cuando se componen servicios Grid a través del uso workflows.

El resto de artículo se organiza de la siguiente manera: Sección 2 presenta algunos de enfoques emergentes de desarrollo de software que se pueden aplicar en el campo de Grid computing y las áreas en las que se aplicaron. La Sección 3 describe uno de enfoques que promete mejorar de manera significativa la composición de servicios Grid al permitir detectar y resolver las posibles interacciones indeseables originadas por la combinación de servicios Grid a través de un workflow. La última Sección presenta la conclusión y futuros trabajos.

2. NUEVOS PARADIGMAS DE INGENIERÍA DE SOFTWARE PARA GRID COMPUTING

En el ámbito de la ingeniería de software existen algunos enfoques emergentes que recientemente han sido aplicados en el campo de la Grid computing:

2.1 Arquitectura Dirigida por Modelos

La Arquitectura Dirigida por Modelos (*Model Driven Architecture, MDA*) [SOLEY2000], define un enfoque de especificación de sistemas de software mediante la

división del proceso de desarrollo en tres capas de modelo separadas y que permite automáticamente transformar los modelos de una capa a otra. El desarrollo del enfoque MDA comienza con la especificación de modelos a partir de un nivel alto de abstracción en UML, definiendo reglas de mapeo desde UML a una plataforma destino y luego utiliza la generación de código para derivar la mayor parte del código de implementación para la plataforma destino. En [HB2009, SFF2006] los autores proponen emplear este enfoque para la composición semiautomática y semántica de workflows de aplicaciones Grid a partir de servicios Grid existentes. [MVM2005] utiliza MDA para desarrollar aplicaciones para Grid computing ofreciendo una visión amigable para los usuarios y arquitectos de aplicaciones, ocultando la complejidad de la plataforma de ejecución final, a través del uso de modelos abstractos capaces de evolucionar y ser reutilizados en futuros desarrollos para Grid.

2.2 Desarrollo de Software Orientado a Aspectos

Desarrollo de Software Orientado a Aspectos (*Aspect-Oriented Software Development, AOSD*) [KMM1997] es un paradigma de desarrollo de software que tiene como objetivo modularizar los llamados *crosscutting concerns*, es decir, los intereses de los sistemas de software que no se alinean bien con la estructura establecida por la orientación a objetos ni con la descomposición funcional. AOSD busca modularizar los sistemas de software para aislar funcionalidades de soporte o secundarias de las funcionalidades básicas del sistema, es decir, aplicar la separación de intereses para obtener una mejor modularización del sistema de software. Así, se obtienen productos de software de calidad con partes más reutilizables y que evolucionen fácilmente en el tiempo. [SM2008] propone diferentes soluciones para facilitar la gridificación de aplicaciones científicas paralelas codificadas en Java aplicando el paradigma AOSD a través de AspectJ. En [SGN2008] se presenta un framework completo para la gridificación de aplicaciones secuenciales y paralelas mediante la aplicación de técnicas AOP evitando modificar los sistemas heredados de manera invasiva.

2.3 Ingeniería de Línea de Producto

Ingeniería de línea de Producto o Desarrollo de Software de línea de productos (*Product Line Engineering, PLE* ó *Software Product Line Development, SPL*) [WL1999] es un enfoque que aplica una forma especializada de reutilización de software mediante el empleo planificado de artefactos básicos de software (*core assets*) dentro del ámbito de un conjunto de productos relacionados, dichos artefactos de software básicos se utilizan en la producción de un producto en una línea de productos de software. En otras palabras, se crea una colección de sistemas de software que son similares a partir de un conjunto de activos de software usando un mismo medio de producción. [VGS2006] propone diferentes técnicas aplicables a servicios Grid que pueden ser utilizadas para realizar variantes de las

aplicaciones basadas en ellos con el fin de maximizar la reutilización en el desarrollo de familias de productos de software. [BP2005] aplica el enfoque SPL en el desarrollo de una familia de productos de software basada en servicios Grid, específicamente dentro del proyecto GeneGrid.

2.4 Desarrollo de Software Orientado a Características

Desarrollo de Software Orientado a Características (*Feature-oriented Software Development, FOSD*) [AK2009] es un paradigma reciente para la construcción, personalización y síntesis de sistemas de software a gran escala, donde el concepto central es la *feature*, que puede traducirse como atributo o característica. Una *feature* es una unidad de funcionalidad de un sistema de software que satisface un requerimiento, representa una decisión de diseño ó proporciona una opción potencial de configuración. El software se descompone en términos de las *features* que provee, para construir software bien estructurado que pueda ser adaptado a las necesidades del usuario y al escenario de aplicación. A partir de un conjunto de *features* comunes se puede generar una variedad de sistemas de software diferentes que comparten *features* comunes y difieren en otras, de manera similar al enfoque de SPL.

3 FOSD Y GRID COMPUTING

FOSD es un paradigma emergente de IS que promete ser de utilidad para mejorar el desarrollo de herramientas a nivel de aplicación para Grid computing, particularmente los workflows de servicios Grid, de igual manera también plantea desafíos ya que, al ser un enfoque novedoso, hasta la fecha se ha aplicado a los servicios Web con resultados satisfactorios en cuanto a permitir su modularidad, variabilidad, uniformidad y compatibilidad, entre otros beneficios. [AKL2008] describe los potenciales beneficios de combinar los enfoques basados en *features* y los enfoques basados en servicios Web para la construcción de sistemas de

software de manera tal que los mismos sean fáciles de desarrollar, mantener y evolucionar. Asimismo en este trabajo se plantean otros aspectos adicionales que deberían investigarse.

Por otro lado, la teoría de FOSD involucra el concepto de *feature interaction (FI)* que representa una situación en la que dos o más *features* exhiben un comportamiento inesperado que no ocurre cuando se utilizan aisladamente. Este problema surgió y fue analizado formalmente en el dominio de los servicios de telecomunicaciones [CKM2003], donde también suele ser llamado *feature interaction problem (FIP)*. Sin embargo, FIP se presentará toda vez que componentes desarrollados independientemente sean requeridos para trabajar de manera conjunta, como ocurre con los servicios Web y, por tanto, con los servicios Grid. Se desarrollaron diversas técnicas para detectar y manejar las interacciones, principalmente para especificar y resolver FI a nivel de diseño y especificación [RR2005].

Los servicios Web son aplicaciones de software cuyas interfaces se definen en WSDL (*Web Service Description Language*), sin embargo tales especificaciones de interfaces pueden resultar insuficientes y por lo tanto FIP es una cuestión a investigar. Los trabajos realizados para analizar FIP en sistemas distribuidos y específicamente relacionados con servicios Web son relativamente recientes [WE2005, WEL2007, ZHC2009]. Los servicios Web se consideran como *features* y las técnicas propuestas para detectar, analizar y resolver FIP en la orquestación de servicios Web proponen realizar una clasificación de FIP para servicios Web y en función de esta taxonomía proponer diferentes métodos para detectar y resolver las FI no deseadas [WEL2007, WBD2009, ZYS2007, XAR2009]. Sin embargo el análisis de FIP en el campo de los servicios Grid aún no ha sido abordado de manera tal de analizar cómo las características particulares de los servicios Grid deben ser consideradas en el momento de combinarlos a través de un workflow Grid.

La Fig. 1 describe la situación propuesta:

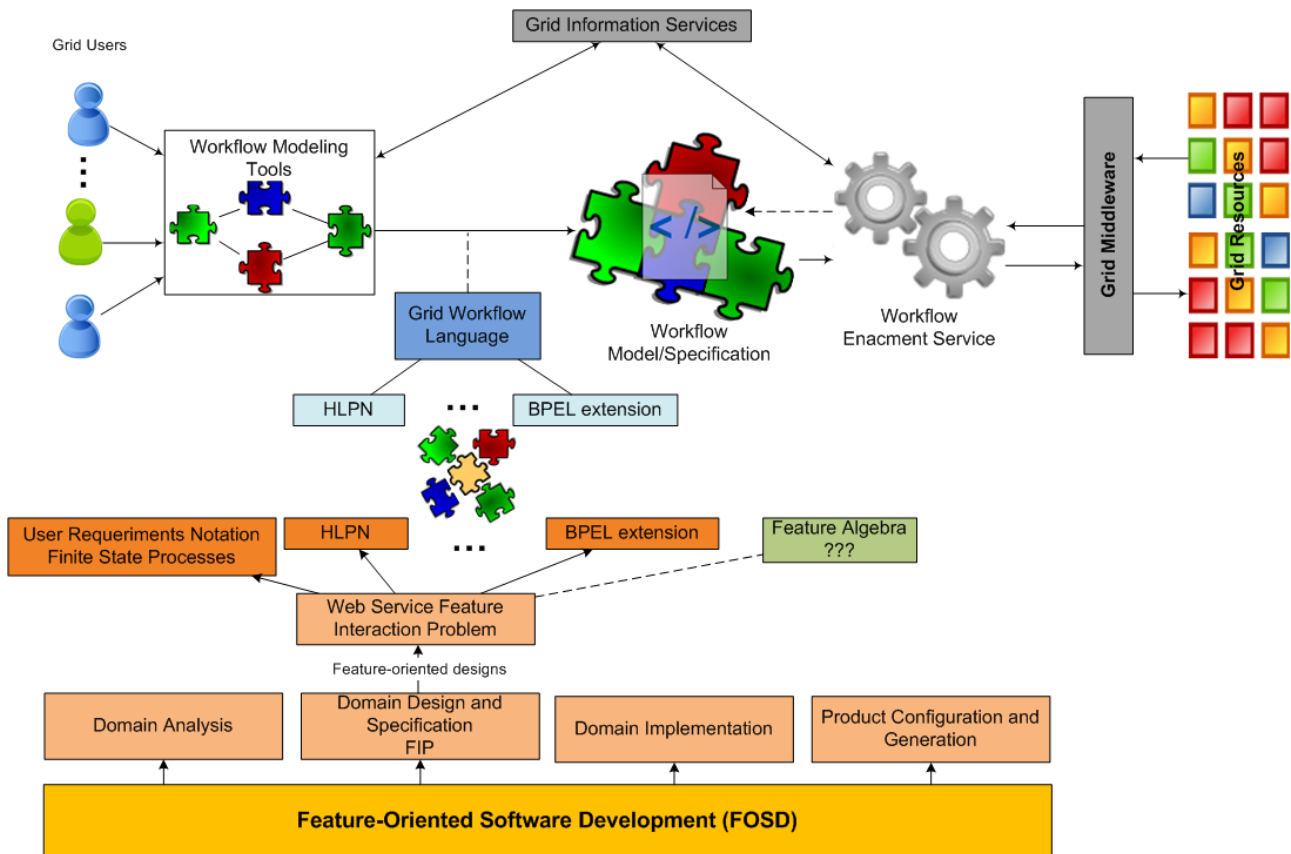


Figura 1 Feature Interaction Problem en la composición de servicios Grid

4. CONCLUSIÓN Y FUTUROS TRABAJOS

Hemos realizado una descripción de las áreas de Grid computing susceptibles de ser mejoradas por novedosos paradigmas de IS y hemos realizado una síntesis de aquellos se han aplicado en estas áreas de Grid computing. Frente a la creación de una aplicación Grid a partir de la combinación de servicios Grid, de los cuales sólo se encuentren disponibles sus interfaces, surge la necesidad de ofrecer una herramienta que permita detectar posibles combinaciones de servicios Grid que produzcan un comportamiento no esperado, entonces sería posible mejorar la forma automatizada de composición de servicios Grid a partir de servicios Grid existentes.

La línea de investigación que proponemos consiste en mejorar el desarrollo y orquestación de servicios Grid tomando como base estos nuevos enfoques de IS con el fin de construir servicios Grid que puedan ser reusables, fáciles de mantener y extensibles, permitiendo lograr diferentes soluciones de forma automatizada a partir de servicios básicos. Aplicando el paradigma *FOSD* y la teoría de *FIP*, las prioridades en esta investigación serán:

- Diseñar un modelo de features común definido para el dominio de servicios Grid que permita generar especificaciones de éstos basadas en features y que incluya una semántica expresada en un lenguaje de especificación formal, para definir ciertas restricciones.

- Determinar la interfaces mínimas que los servicios Grid tienen que comunicarse entre sí a fin de detectar y resolver las interacciones.
- Desarrollar herramientas que permitan realizar la detección y verificación de las interacciones entre servicios Grid basadas en sus interfaces, sus features y las restricciones de dominio y proponer un método para evitar y/o resolver las interacciones detectadas no deseadas.

5. BIBLIOGRAFÍA

- [AK2009] S. Apel and C. Kästner. An Overview of Feature-Oriented Software Development. *Journal of Object Technology*, 8(5):49-84, July-August 2009. http://www.jot.fm/issues/issue_2009_05/column5/
- [AKL2008] S. Apel, C. Kästner, and C. Lengauer. Research Challenges in the Tension Between Features and Services. In *SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments*, pages 53–58, Leipzig, Germany, 2008. ACM Press
- [BP2005] R. Bashroush, R. Perrott. Using a Software Product Line Approach in Designing Grid Services. <http://www.allhands.org.uk/2005/proceedings/papers/499.pdf>
- [CKM2003] M. Calder, M. Kolberg, E. Magill, and S. Reiff-Marganiec. Feature Interaction: A Critical Review and Considered Forecast. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 41(1):115–141, 2003

- [FK2004] I. Foster and C. Kesselman, eds 2004. *The Grid: Blueprint for a new computer infrastructure*, 2nd ed. San Francisco, CA: Morgan Kaufmann. ISBN: 1558609334
- [FKS2005] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. The Open Grid Services Architecture, Version 1.0. Informational Document, Global Grid Forum (GGF), 2005. <http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf>
- [FFG2004] I. Foster (ed), J. Frey (ed), S. Graham (ed), S. Tuecke (ed), K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, I. Sedukhin, D. Snelling, T. Storey, W. Vambenepe, S. Weerawarana. Modeling Stateful Resources with Web Services v. 1.1. , March 5, 2004. <http://www-106.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf>
- [HB2009] Y. BenDaly Hlaoui, L. Jemni BenAyed. An Interactive Composition Of Workflow Applications Based On UML Activity Diagram. Special Issue on ICIT 2009 Conference - Applied Computing. Volume Applied Computing, Special Issue, ISSN Online 1992-8424 ISSN Print 1994-4608, 2009. http://www.ubicc.org/files/pdf/7_378.pdf
- [KMM1997] Kiczales, G. J., Mendhekar, L. A., Maeda, C., Lopes, C., Loingtier, J. and Irwin, J. Aspect-Oriented Programming. In *Proc. of the 11th Eur. Conf. on Object-Oriented Programming (ECOOP'97)*, pages 220-242, Finland, June 1997. Springer-Verlag, LNCS Vol. 1241
- [MVM2005] Manset D., Verjus H., McClatchey R., Oquendo F. A Model-Driven Approach for Grid Services Engineering. In *Proceedings of the Eighteenth International Conference on Software and Systems Engineering and their Applications (ICSSEA'05)*, Paris, France, 2005
- [RR2005] S. Reiff-Marganiec and M. Ryan, editors. *Proceedings of the International Conference on Feature Interactions in Software and Communication Systems (ICFI)*. IOS Press, 2005
- [SFF2006] M. Smith, T. Friese, B. Freisleben. Model Driven Development of Service-Oriented Grid Applications. In *Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services*, pages 139, Guadeloupe, French Caribbean, February 19-25, 2006. IEEE Computer Society Press
- [SGN2008] E. Sousa, R. Gonçalves, D. Neves, J. Sobral. Non-Invasive Gridification through an Aspect-Oriented Approach. In 2nd Iberian Grid Infrastructure Conference (Ibergrid'08), Porto, Portugal, 2008. <http://alba.di.uminho.pt/aspectgrid/articles/ibergrid.pdf>
- [SM2008] J. Sobral, M. Monteiro. A Domain-Specific Language for Parallel and Grid Computing. In 3rd Workshop on Domain Specific Languages (DSAL 2008), Brussels, Belgium, April 2008
- [SOLEY2000] Richard Soley. Model Driven Architecture, White Paper, 2000. <http://www.omg.org/cgi-bin/doc?omg/00-11-05>
- [VGS2006] Jilles van Gorp, Juha Savolainen. Service Grid Variability Realization. In *Proceedings of 10th International Software Product Line Conference (SPLC 2006)*, Baltimore, Maryland, August 2006. <http://www.jillesvangorp.com/static/splc2006servicegridvariability.pdf>
- [WBD2009] Wu, X., Buford, J., Dhara, K., Krishnaswamy, V., and Kolberg, M. Feature interactions between internet services and telecommunication services. In *Proceedings of the 3rd international Conference on Principles, Systems and Applications of IP Telecommunications*, pages 1-12, Atlanta, Georgia, July 07 - 08, 2009. ACM, New York
- [WE2005] M. Weiss, B. Esfandiari. On feature interactions among web services. *International Journal on Web Services Research*, 2(4):21-45,2005
- [WEL2007] M. Weiss , B. Esfandiari , Y. Lu. Towards a classification of web service feature interactions. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(2):359-381, 2007. <http://www.michaelweiss.ca/courses/comp/5401/readings/Weiss-comnet07.pdf>
- [WL1999] D. Weiss and C. Lai. *Software Product-Line Engineering: A Family-Based Software Development Process*. MA: Addison-Wesley, 1999. ISBN-10: 0201694387
- [XAR2009] Xiangyu Luo, Aicheng Xuan, Rongsheng Dong. Detecting Feature Interactions in Web Services with Timed Automata. In *Proceedings of Third International Conference on Genetic and Evolutionary Computing*, pages 276-279, Guilin, China, 2009. IEEE
- [YB2005] Jia Yu and Rajkumar Buyya. A Taxonomy of Workflow Management Systems for Grid Computing. *Journal of Grid Computing*, 3(3-4):171-200. Springer Science+Business Media B.V., New York, USA, Sept. 2005
- [ZHC2009] Zhao, Qi and Huang, Jiyu and Chen, Xiangping and Huang, Gang. Feature Interaction Problems in Web-based Service Composition. In *Proceedings of 10th International Conference on Feature Interactions (ICFI)*, Lisbonne, Portugal, 2009. <http://hal.archives-ouvertes.fr/docs/00/45/96/27/PDF/4-2009-ICFI-iMashup.pdf>
- [ZYS2007] Jian-yin ZHANG and Fang-chun YANG and Sen SU. Detecting feature interactions in Web services with model checking techniques. *The Journal of China Universities of Posts and Telecommunications*, 14(3):108-112, 2007