

Composição autônômica de serviços web para prevenir ou minimizar interações de características

Roberto de Cerqueira Figueiredo¹, Daniela Barreiro Claro¹,
Raimundo José de Araújo Macedo¹, Aline Maria Santos Andrade¹

¹FORMAS-Formalismos e Aplicações Semânticas,
Laboratório de Sistemas Distribuídos (LASID)
Universidade Federal da Bahia
Av. Adhemar de Barros, s/n, Ondina – Salvador – BA – Brazil

{roberedo, dclaro, macedo, aline}@ufba.br

Resumo. *Este trabalho propõe uma solução autônômica para composições de serviços web com a meta de minimizar futuras violações dos requisitos funcionais e não funcionais da composição. Estas violações são denominadas interações de características (IC). A solução consiste em dois tipos de gerentes autônômicos, com funções distintas: o gestor distribuído para classificação dos serviços quanto a qualidade de interação de características (QIC) e o gestor de adaptação para seleção de novos serviços através do QIC. A solução autônômica é promissora na prevenção de IC, pois utiliza uma QIC que representa serviços concretos.*

1. Introdução

Um serviço web é uma funcionalidade de um sistema de informação padronizada e publicada na internet [Alonso et al. 2004]. O seu objetivo é cumprir requisitos funcionais e não funcionais para atender as solicitações de usuários. Para tal, um único serviço pode não ser suficiente, sendo então necessário compor serviços [Amorim et al. 2011].

Apesar da execução de um único serviço não violar os seus requisitos funcionais e não funcionais, a execução da composição pode causar tal violação. Essa violação é denominada interação de características [Weiss et al. 2007]. Para solucionar interação de características, autores em [Keck and Kuehn 1998] definem três macro etapas: prevenção, detecção e resolução. A etapa de prevenção intervém no projeto da composição, no intuito de evitar que interações ocorram. A etapa de detecção identifica a existência de interações de características. Enfim, a etapa de resolução trata de eliminar ou minimizar os efeitos indesejáveis da interação de características detectadas.

Prevenir, detectar e resolver interações de características na composição de serviços web é um grande desafio principalmente devido ao dinamismo e heterogeneidade da web. Devido a tal dinamismo, mudanças frequentes nos serviços da composição para atender os requisitos do usuário podem ocasionar interações de características. Além disso, cada desenvolvedor pode criar o seu próprio serviço e publicá-lo na internet, caracterizando uma composição de serviços heterogêneos. Compor serviços heterogêneos na web aumenta a possibilidade de interações de características. Desta forma, devido à frequência de mudanças e a heterogeneidade da composição web, prevenir, detectar e resolver interações de características tornam-se tarefas complexas.

Assim, o presente trabalho propõe uma solução autônoma para prevenção, detecção e resolução de interação de características na composição de serviços web. Autores em [Kephart and Chess 2003] definem um sistema autônomo como um sistema computacional que se auto-gerencia baseado em metas definidas por um administrador. O autônomo pode então gerenciar a complexidade inerente à composição para prevenir, detectar e resolver interações de características. O presente artigo está organizado como segue. A seção 2 apresenta um exemplo de IC. Na seção 3, a proposta deste trabalho é descrita e a seção 4 discute a solução proposta.

2. Interação de características na composição de serviços web

As mudanças em uma composição podem ser implementadas por seleção de serviços. A seleção de serviços é definida como a seleção de serviços concretos (implementação do serviço) para cada serviço abstrato (funcionalidade desejada) da composição. Tal seleção pode ocasionar IC entre serviços. Por exemplo, um cliente deseja comprar itens em uma aplicação de comércio eletrônico. Os requisitos desejados pelo cliente são definidos nos serviços abstratos $SA1$ e $SA2$, conforme Figura 1. Os serviços concretos $S3$ e $S5$ são então selecionados para cumprir os requisitos do cliente $SA1$ e $SA2$, respectivamente. Se o serviço fornecedor $S5$ não tiver itens no seu estoque, encaminhará a solicitação de itens ao fornecedor $S4$. Se ocorrer o mesmo com $S4$, este encaminhará a solicitação de itens a $S5$, formando assim um laço infinito de solicitações. Isso ocorre, pois ambos serviços foram projetados para encaminhar solicitação de itens caso não os tenham em seu estoque.



Figura 1. Interação de características na composição de serviços web

O laço infinito viola o requisito funcional fornecer itens, de $S5$. Além disso, requisitos não funcionais atribuídos a $S5$, por exemplo, tempo de resposta, também serão violados. Uma possível solução para esta interação de características seria substituir um serviço $S5$ por um serviço similar $S8$ que não seja projetado com encaminhamento de solicitação de itens. Assim, se $S8$ não tiver itens em seu estoque, informa ao cliente.

3. Proposta

Autores em [Kephart and Chess 2003] definem um sistema autônomo como um sistema computacional capaz de se auto-gerenciar, tendo como referência metas de alto nível definidas por um administrador. Para tal, um gerente autônomo controla partes do sistema denominadas de elemento gerenciado através de um ciclo de gerenciamento denominado MAPE-BC (Monitorar, Analisar, Planejar, Executar, e Base de Conhecimento).

A proposta deste trabalho é prevenir ocorrências de interações através da seleção de novos serviços concretos para os serviços abstratos da composição abstrata. Para isso, é preciso também detectar e resolver IC. Serão propostos gerentes autônomos distribuídos e um gerente autônomo de adaptação, conforme ilustrado na figura 2. Gerentes distribuídos detectam e resolvem interação de características nos provedores. Após detecção da interação, é feita uma análise para classificar cada serviço concreto com uma medida de

interação (QIC). Esta medida é armazenada na base de conhecimento de cada gerente distribuído. Diante disso, o gerente autônomo de adaptação seleciona os melhores serviços concretos baseado na medida de interação. Se esta solução for aplicada ao exemplo da seção 2, o serviço *S8* obterá bom QIC enquanto que o serviço *S5* será classificado com um QIC ruim, determinando assim, a substituição de *S5* por *S8*.

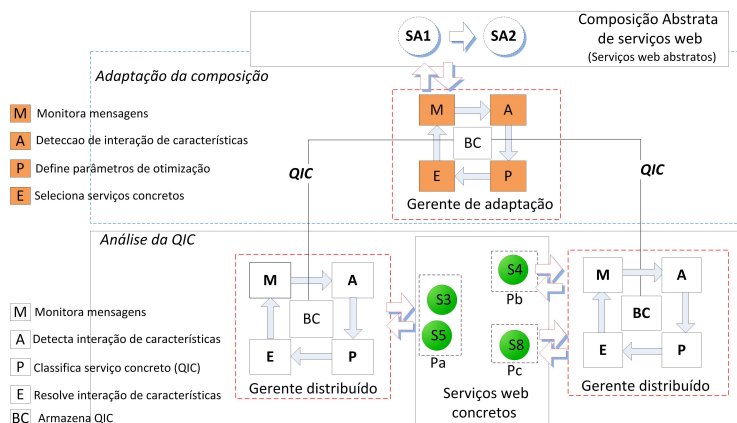


Figura 2. Solução autônoma para prevenção ou minimização de interação de características

A detecção da interação avalia mensagens enviadas e recebidas por um serviço concreto, no intuito de encontrar um padrão de interação de características entre os parâmetros das mensagens. A classificação do serviço concreto ocorrerá através de uma medida de probabilidade atribuída ao serviço concreto de modo a representar a chance deste serviço ocasionar interação de características nas próximas interações. A etapa de resolução estabelece qual mensagem recebida por um serviço será executada e quais mensagens serão enviadas. A definição dos parâmetros de otimização representa a medida QIC atualizada. Enfim, a seleção de serviços será realizada por algoritmos genéticos.

4. Discussão

Alguns autores vem propondo soluções para IC na composição de serviços web [Tang and Ai 2010, Xu et al. 2011, Lu and Tosic 2011]. Autores em [Tang and Ai 2010], utilizaram um algoritmo genético HGA para prevenção baseada na seleção de serviços. [Xu et al. 2011] propõe uma técnica ESTRIPS para detecção baseada no raciocínio semântico com SWRL e OWL-S. Autores em [Lu and Tosic 2011] propuseram um módulo de resolução de conflitos MiniZnMASC que decide qual ação conflitante será executada. A tabela 1 mostra uma síntese de comparação dos trabalhos relacionados.

A prevenção ou minimização de interação de características na composição de serviços web deve ser feita de modo autônomo, pois os serviços concretos para uma composição são heterogêneos e o ambiente web possui características dinâmicas. A abordagem autônoma proposta possui a vantagem de otimizar serviços através do QIC, enquanto que o HGA (conforme tabela 1) otimiza serviços através de restrições baseadas em conflitos que foram definidas em tempo de projeto. Além disso, o HGA não detecta IC em tempo de execução, enquanto que a solução QIC detecta em tempo de execução.

A interação de características pode ocorrer entre os serviços selecionados para a composição abstrata mas também pode ocorrer entre serviços que não estão na

Tabela 1. Prevenção, detecção e resolução de IC na composição de serviços web

<i>Propostas</i>	<i>Prevenção</i>	<i>Detecção</i>	<i>Resolução</i>	<i>Resultado</i>
HGA	Otimização por restrições estáticas	X	X	Algoritmo genético híbrido
ESTRIPS	X	Detecta FI na composição abstrata	X	Sistema especialista
MiniZnMASC	X	X	Resolução entre políticas	Middleware autônomo
QIC	Otimização através do QIC	Detecta FI na composição abstrata e nos serviços concretos	Resolução de propósito geral	Framework Autônomo

composição abstrata. Por exemplo, na seção 2, o serviço S_5 foi selecionado para a composição abstrata do cliente, ao contrário de S_4 . Portanto, a proposta para detectar interação de características ESTRIPS (conforme tabela 1) é limitada, pois detecta somente interação de características entre os serviços da composição abstrata. A complexidade desta detecção e análise de interação de características requer uma solução autônoma.

5. Conclusão

Neste trabalho, uma solução autônoma para composição de serviços foi proposta, no intuito de prevenir ou minimizar a ocorrência de interação de características. Dois tipos de gestores autônomos foram propostos. A solução autônoma é promissora na prevenção de IC, pois utiliza uma QIC que representa serviços concretos.

Referências

- Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web services: Concepts, architectures and applications*. Data-centric systems and applications. Springer.
- Amorim, R., Claro, D. B., Lopes, D., Albers, P., and Andrade, A. (2011). Improving web service discovery by a functional and structural approach. In *ICWS*, pages 411–418.
- Keck, D. O. and Kuehn, P. J. (1998). The feature and service interaction problem in telecommunications systems: A survey. *IEEE Trans. Softw. Eng.*, pages 779–796.
- Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, pages 41–50.
- Lu, Q. and Tasic, V. (2011). Support for concurrent adaptation of multiple web service compositions to maximize business metrics. In *(IM) IFIP/IEEE*, pages 241–248.
- Tang, M. and Ai, L. (2010). A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition. In *IEEE WCCI*.
- Weiss, M., Esfandiari, B., and Luo, Y. (2007). Towards a classification of web service feature interactions. *Comput. Netw.*, pages 359–381.
- Xu, J., Chen, K., Duan, Y., and Reiff-Marganiec, S. (2011). Modeling business process of web services with an extended strips operations to detection feature interaction problems runtime. *IEEE 19th ICWS*, pages 516–523.