LNCS 5788

Emilio Corchado
Hujun Yin (Eds.)

Intelligent Data Engineering and Automated Learning - IDEAL 2009

10th International Conference
Burgos, Spain, September 2009
Proceedings

Springer

Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at
www.springer.com/lncs

Proposals for publication should be sent to
LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany
E-mail: lncs@springer.com

ISSN 0302-9743

Lecture Notes in Computer Science
LNCS LNAI LNBI

springer.com

Agent Negotiation Protocols in Time-Bounded Service Composition

Martí Navarro, Elena del Val, Miguel Rebollo, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia, Spain
{mnavarro,edelval,mrebollo,vinglada}@dsic.upv.es

Abstract. Time is an important non-functional parameter to consider in service compositions, especially in environments where a service must be provided before a deadline. In this paper service composition takes service execution time into account and, to provide the composition with more reliability, considers the workload and availability of service providers. Furthermore, negotiation protocols are used to add flexibility between clients and providers that participate in the service composition. Negotiation protocols increase the chances of reaching an agreement by allowing both, providers and clients, to exchange their proposals in order to adjust the negotiation terms related to service execution time.

1 Introduction

One of the main difficulties in service-oriented environments is how to create added-value services dynamically by composing elemental services. Semantic annotations help machines to deal with services, but service discovery and composition are complex tasks that need extra intelligence doses to achieve proper results, especially in open and dynamic environments where services are not always available. Despite all of the work in the area of service composition, there is a major problem: attend client requests in a bounded time. The late fulfillment of some requests could reduce the quality of the response offered by the system. Thus, compositions of services must be carried out taking temporal restrictions established by the client into account.

If a provider agent is already executing several services it may not have enough time to deal with a new request, and therefore the service composition becomes impossible. So, in order to evaluate whether an initially suitable service composition is feasible, it is necessary for each provider agent to report its availability in order to deal with the request.

There are several proposals that consider time in service specifications expressing temporal constraints [1,4,10]. Some of them also verify temporal properties and in some approaches the specifications can be executed [9]. Furthermore, there are other proposals regarding service composition that consider time[8,3]. All of the proposals presented consider time from a description level, as a static parameter regardless of the provider's workload or the availability of the resources

E. Corchado and H. Yin (Eds.): IDEAL 2009, LNCS 5788, pp. 527–534, 2009.

similar, but the system utilization is higher using the negotiation in the SAES because all of the service executions are scheduled in order to optimize the CPU utilization. The main difference between a system with negotiation and one without is the quality of the service response. With the negotiation the quality increases considerably. This is because the SAES, in the case that a client request cannot be provided before a deadline, does not flatly reject the client request, but instead offers an alternative, so the client can decide whether to accept a new deadline in order to get the service response. Using the negotiation an alternative deadline is provided, and although the quality of the service response is lower, this is better than rejecting the client's requests or not satisfying the client's expectations. Therefore, the use of negotiation provides a system with better CPU perfomance and quality, due to the fact that the SAES does not refuse services that could be fulfilled with a little more time.

References

1. Pan, F.: Temporal aggregates for web services on the semantic web. In: IEEE International Conference on Web Services, pp. 831–832 (2005)
2. WS-agreement, `http://www.ogf.org/documents/GFD.107.pdf`
3. Fernández-Olivares, J., Garzón, T., Castillo, L., García-Pérez, O., Palao, F.: A middle-ware for the automated composition and invocation of semantic web services based on temporal htn planning techniques, pp. 70–79 (2007)
4. Hao, J., Zhi-jian, S.: The tcpn-based verification of temporal consistency in web service process. In: ICEBE 2006: Proceedings of the IEEE International Conference on e-Business Engineering, pp. 302–306 (2006)
5. Fox, M., Long, D.: PDDL2.1: An extension to PDDL for expressing temporal planning domains. J. Artif. Intell. Res (JAIR) 201, 61–124 (2003)
6. Navarro, M., del Val, E., Rebollo, M., Julian, V.: Composing and ensuring time-bounded agent services. In: Cabestany, J. (ed.) IWANN 2009. LNCS, vol. 5517, pp. 553–560. Springer, Heidelberg (2009)
7. Navarro, M., Julian, V., Soler, J., Botti, V.: jART: A real-time multi-agent platform with rt-java. In: IWPAAMS 2004, pp. 73–82 (2004)
8. Naseri, M., Towhidi, A.: Qos-aware automatic composition of web services using ai planners. In: ICIW 2007: Proceedings of the Second International Conference on Internet and Web Applications and Services, pp. 29–34 (2007)
9. Solanki, M.: Tesco-s: A framework for defining temporal semantics in owl enabled services. In: W3C Workshop on Frameworks for Semantics in Web Services (2005)
10. Martín-Díaz, O., Cortés, A.R., Durán, A., Müller, C.: An approach to temporal-aware procurement of web services. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 170–184. Springer, Heidelberg (2005)
11. The real-time for Java expert group, `http://www.rtj.org`