Adaptable decentralized Service Oriented Architecture.

ABSTRACT

In the Service Oriented Architecture (SOA), BPEL specified business processes are executed by non-scalable centralized orchestration engines. In order to address the scalability issue, decentralized orchestration engines are applied, which decentralize BPEL processes into static fragments at design time without considering runtime requirements. The fragments are then encapsulated into runtime components such as agents. There are a variety of attitudes towards workflow decentralization; however, only a few of them produce adaptable fragments with runtime environment. In this paper, producing runtime adaptable fragments is presented in two aspects. The first one is frequent-path adaptability that is equal to finding closely interrelated activities and encapsulating them in the same fragment to omit the communication cost of the activities. Another aspect is proportional-fragment adaptability, which is analogous to the proportionality of produced fragments with number of workflow engine machines. It extenuates the internal communication among the fragments on the same machine. An ever-changing runtime environment along with the mentioned adaptability aspects may result in producing a variety of process versions at runtime. Thus, an Adaptable and Decentralized Workflow Execution Framework (ADWEF) is introduced that proposes an abstraction of adaptable decentralization in the SOA orchestration layer. Furthermore, ADWEF architectures Type-1 and Type-2 are presented to support the execution of fragments created by two decentralization methods, which produce customized fragments known as Hierarchical Process Decentralization (HPD) and Hierarchical Intelligent Process Decentralization (HIPD). However, mapping the current system conditions to a suitable decentralization method is considered as future work. Evaluations of the ADWEF decentralization methods substantiate both adaptability aspects and demonstrate a range of improvements in response-time, throughput, and bandwidth-usage compared to previous methods.