

Pros and Cons of Distributed Workflow Execution Algorithms

Hans Schuster

Consileon Business Consultancy GmbH, Germany

hans.schuster@consileon.de

Abstract. As an implementation of business processes workflows are inherently distributed. Consequently, there is a considerable amount both of commercial products and research prototypes that address distribution issues in workflow execution and workflow management systems (WfMS). However, most of these approaches provide only results focussed on the properties of a specific workflow model, workflow application, and/or WfMS implementation. An analysis of generic requirements on distributed workflow execution algorithms and their applicability, advantages, and disadvantages in different workflow scenarios is still missing but will be shown in this paper. A comprehensive requirements analysis on distributed workflow execution forms the basis of our discussion of distributed workflow execution. In contrast to existing work that primarily focuses on non-functional requirements, this paper explicitly considers issues that originate in the workflow model as well. Subsequently, four basic algorithms for distributed workflow execution are presented, namely remote access, workflow migration, workflow partitioning, and subworkflow distribution. Existing WfMS approaches use combinations and/or variants of these basic algorithms. The properties of these algorithms with respect to the aforementioned requirements are discussed in detail. As a primary result, subworkflow distribution proves to be a well-suited application-independent and thus generally applicable distributed execution model. Nevertheless, application-specific optimizations can be accomplished by other models.

1 Motivation

Today's enterprises are highly distributed. The trend to enter cooperations among different enterprises in order to build virtual enterprises even intensifies this distributed nature. Workflow management is a promising technology to implement business processes in traditional as well as virtual enterprises through workflows that are enacted by workflow management systems (WfMS). WfMS integrate both human and fully automated activities within workflows [17].

Due to the distributed nature of the application environment, workflows, i.e. their executions, have to be distributed themselves. Hence, there is a strong need for suitable methods for distributed workflow execution. Although, there are many commercial workflow products and academic WfMS prototypes that all support some kind of distributed workflows, there is still no commonly agreed upon workflow model although the Business Process Execution Language for Web Services (BPEL4WS, [1]) is on a good way to establish an agreed standard for fully automated workflows. There is also no commonly agreed distributed execution model for workflows. The existing approaches address only specific problems of distributed workflows in the context of a particular workflow model without tackling the general problem. A comprehensive analysis of the basic requirements on distributed workflow execution and alternative solutions is still missing. However, only such an analysis can build up a solid foundation for a decision how to implement a workflow application in a particular (virtual) enterprise. A taxonomy for the suitability of distributed workflow execution algorithms in specific application scenarios is highly desirable.

The main contribution of this paper is the identification of four generic algorithms for distributed workflow execution, and their thorough discussion with respect to a comprehensive list of requirements covering model-related (e.g., synchronization of activities) and behavioral (e.g., performance, scalability) requirements. 'Generic' in this context means that—in contrast to other work done in this area—we do not assume a specific workflow model and/or WfMS implementation. In contrast, this work is based on a set of rudimentary assumptions that are fulfilled by virtually all existing workflow models. Distributed workflow execution algorithms proposed in the literature [3, 16, 4, 5, 6, 10, 11, 14, 21, 22, 27, 25] are a combination of the four generic algorithms. Therefore, it is not our objective to propose a new way to implement distributed workflows but to improve the understanding of the problem of distributed workflow execution. The results of this paper can serve as a foundation when deciding which approach to choose in a concrete workflow application scenario.

The next section presents the requirements that have to be fulfilled by distributed workflows. Section 3 introduces the four basic distributed workflows execution algorithms and discusses their general properties and combinations. The efficiency of these algorithms is the focus of section 4. Finally, section 5 presents a brief overview on existing work in analyzing distributed workflow approaches and Section 6 gives a short conclusion.

2 Requirements on Distributed Workflow Execution

Distributed workflow execution is not an end in itself. On the contrary, it is the answer on the demand for the adaptation of workflow execution to the distributed nature of the application environment in today's enterprises. This answer is naturally constrained by requirements that result from the workflow model used in a WfMS. Additionally, a distributed WfMS has to meet the non-functional requirements of application, like scalability and efficiency. section 2.1 defines the terminology used in this paper and our as-