Invisible Deployment of Integration Processes

Matthias Boehm, Dirk Habich, Wolfgang Lehner, and Uwe Wloka

1 Dresden University of Applied Sciences, Database Group
mboehm@informatik.htw-dresden.de, wloka@informatik.htw-dresden.de
2 Dresden University of Technology, Database Technology Group
dirk.habich@tu-dresden.de, wolfgang.lehner@tu-dresden.de

Abstract. Due to the changing scope of data management towards the management of heterogeneous and distributed systems and applications, integration processes gain in importance. This is particularly true for those processes used as abstractions of workflow-based integration tasks; these are widely applied in practice. In such scenarios, a typical IT infrastructure comprises multiple integration systems with overlapping functionalities. The major problems in this area are high development effort, low portability and inefficiency. Therefore, in this paper, we introduce the vision of invisible deployment that addresses the virtualization of multiple, heterogeneous, physical integration systems into a single logical integration system. This vision comprises several challenging issues in the fields of deployment aspects as well as runtime aspects. Here, we describe those challenges, discuss possible solutions and present a detailed system architecture for that approach. As a result, the development effort can be reduced and the portability as well as the performance can be improved significantly.

Keywords: Invisible deployment, Integration processes, Virtualization, Deployment, Optimality decision, Heterogeneous integration platforms.

1 Introduction

Integration processes—as an abstraction for workflow-based integration tasks—gain in importance because data management continuously changes towards the management of distributed and heterogeneous systems and applications. There, the performance of complete IT infrastructures depends on the central integration platforms. In this context, different integration system types are used. Examples for those types are Federated DBMS, EAI (Enterprise Application Integration) servers, ETL (Extraction Transformation Loading) tools, and WFMS (Workflow Management Systems). However, the boundaries between these different classes of systems begin to blur due to overlapping functionalities of concrete products. Major problems in this context are posed by the high development effort for integration task specification, the low degree of portability between those integration systems, and the possible inefficiency. The inefficiency problem (optimization potential) is caused by system-inherent assumptions about the primary application context. If the actual workload characteristics (process types, data
size) differ from those assumptions, the execution performance can be significantly improved by changing the used integration system.

Our main hypotheses are (1) that a typical IT infrastructure comprises multiple integration systems with overlapping functionalities, and (2) that we can generate platform-specific integration task specifications from platform-independent models. The opportunities arising from these hypotheses led us to our idea of invisible deployment. Here, a user models an integration process in a platform-independent way and deploys it using a central deployment interface. Now, there is the general possibility to decide on the optimal integration platform to execute the specified integration process. This decision should consider workload execution statistics in order to be based on objective online statistics with regard to changing workload characteristics. Clearly, this general idea can overcome the problems of high development effort and low portability (generation of process descriptions) as well as inefficiency (optimality decision, load balancing), but it comes with several inherent challenges.

In order to overcome the described problems and to convey the core idea of invisible deployment, in this paper, we make the following contributions.

- In Section 2, we introduce the vision of invisible deployment and describe the major challenges that arise when realizing that vision.
- Subsequently, in Section 3, we describe our approach for the deployment of integration processes. Here, we focus on the selected aspects of integration process generation and functional candidate set determination.
- Furthermore, in Section 4, we discuss a possible runtime approach, where we investigate cost modeling and cost normalization, optimality decisions and the heterogeneous load balancing.
- Based on the proposed solution, in Section 5, we present a system architecture for the realization of invisible deployment in the context of integration platforms.

Finally, we survey related work in Section 6 and conclude the paper in Section 7.

2 Vision Overview

Based on the described problems, in this section, we pose our main hypotheses and present the resulting conceptual architecture for the vision of invisible deployment. Additionally, we point out the major challenges that arise here.

2.1 Assumptions and Hypotheses

In fact, our vision is based on empirically evaluated assumptions. Here, we conclude the following two hypotheses.

**Hypothesis 1.** Model-Driven Generation: Integration processes can be modeled in a platform-independent way. Based on those models, we can generate proprietary (platform-specific) integration task specifications for concrete integration platforms.

**Hypothesis 2.** Optimality Decision: A typical IT infrastructure comprises multiple integration systems with overlapping functionalities. Hence, there is the possibility to decide on the optimal integration system based on functional and non-functional properties according to given integration processes.