

Flexible Provisioning of Service Workflows

(Extended Abstract)¹

Sebastian Stein

Nicholas R. Jennings

Terry R. Payne

University of Southampton, SO17 1BJ, United Kingdom
{ss04r,nrj,trp}@ecs.soton.ac.uk

1 Background

Computer systems are becoming increasingly complex, distributed and open in nature. In so doing, they pose new challenges to contemporary software engineering techniques, as applications need to deal with many widely distributed and dynamic resources. In this context, service-oriented computing has gained popularity as an appropriate paradigm to build such systems. In this approach, resources are offered as services, which are advertised to consumer applications using computer-readable descriptions and then provisioned dynamically when needed, often as part of complex workflows.

A defining feature of these distributed systems is that participants are usually heterogeneous, self-interested agents that act autonomously in accordance with their own private goals. As such, agents choose actions that maximise their own welfare, and so they cannot be assumed to honour every request, to be available whenever they are needed or even to report their capabilities and status truthfully. Additionally, network problems, competition for resources or even software bugs exacerbate the potential for service failures. Thus, when interacting with service providing agents and making real investments that depend on their performance (whether in terms of money, time or other resources), dealing with service failures becomes a vital issue to consider.

Against this background, we believe that flexible service provisioning (i.e., selecting and allocating service providers to specific tasks) is the key to handling and controlling service failures. It allows the consumer to dynamically select service providers based on their performance characteristics and provision replacement providers when services fail. Specifically, provisioning services in the context of a workflow enables a consumer agent to identify particularly failure-prone tasks and invest additional resources in them. To date, however, this provisioning process has received comparatively little attention. In particular, the fact that a type of service might be offered by multiple providers is seldom explored, and most research is concerned with finding and invoking the first matching service that helps fulfil a given goal. Such a naïve approach means that a single service failure will result in the failure of the whole workflow, which is highly undesirable, especially when success is critical to the interests of the consumer agent.

2 Flexible Provisioning

To address this issue, in our paper [1] we investigate the process of provisioning service workflows in environments where service success is not generally guaranteed. We show that the current approach of provisioning single providers is insufficient in such environments, and then advance the state of the art by developing several strategies that deal proactively with unreliable providers. In particular, we propose two key techniques:

Parallel Provisioning: To increase the probability of success, n agents that provide the same service can be provisioned in parallel for a single task in the workflow. The task will then succeed as long as at least one service completes successfully.

¹This is an extended abstract of [1].

Serial Provisioning: Providers can also be re-provisioned when it becomes obvious that they have failed. As failure may not be communicated explicitly by the providers, this process involves provisioning an initial set of agents, and then waiting for some time w , after which a new set of agents is provisioned for the same task (and so on).

Now, the appropriate choice of n and w for each task depends on the performance characteristics of the available services. For example, if services are cheap and unreliable, a high value for n might be chosen. If services generally take a long time to complete, then a high w would reflect this.

Because finding the right level of parallelism and waiting times is non-trivial to perform manually, and because it depends on the consumer’s valuation of the workflow and associated time-constraints, we automate this choice by employing decision theory. To this end, we attach a utility function to each workflow, in order to represent the value of completing the workflow after some time t . As provisioning providers optimally (i.e., maximising the expected utility) is intractable, we then use a hill-climbing algorithm that uses a heuristic estimation of the true expected utility and explores a small subset of the decision space by iteratively modifying a candidate solution.

3 Results

In order to verify our approach, we also report our empirical results. These clearly demonstrate the value of flexible service provisioning over current approaches. For example, Figure 1 (taken from [1]) shows our provisioning strategies in environments with varying service uncertainty. In particular, the *naïve* strategy represents current approaches that rely on single providers, *parallel(2)* and *parallel(6)* are strategies where the number of parallel providers (n) is fixed statically, and the *flexible* strategy is our heuristic provisioning approach.

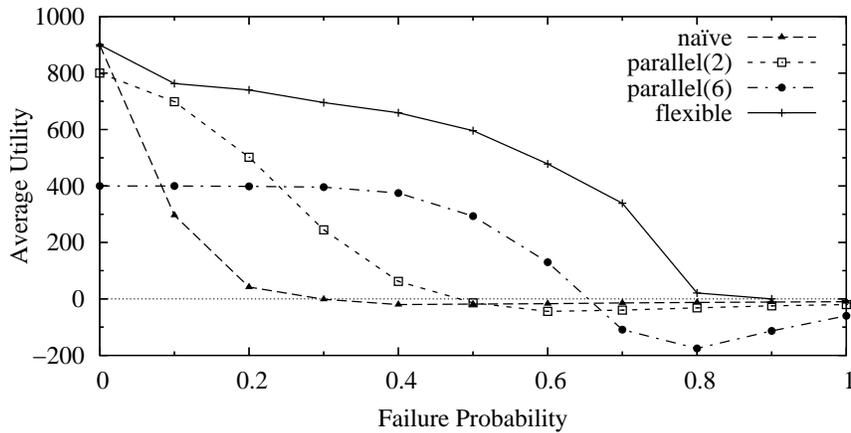


Figure 1: Average utility in environments with increasing service failure probability.

The *flexible* strategy clearly dominates the others, and, in fact, we show that it successfully completes over 90% of its workflows in most environments where current approaches complete none. Furthermore, our results indicate that it achieves a 350% improvement in average utility over the *naïve* strategy.

References

- [1] Sebastian Stein, Nicholas R. Jennings, and Terry R. Payne. Flexible provisioning of service workflows. In *Proceedings of 17th European Conference on Artificial Intelligence (ECAI-06)*, Riva del Garda, Italy, pages 295–299. IOS Press, 2006.