**Abstract**

Service discovery has been recognized as an important aspect in the development of service-centric systems, i.e., software systems which deploy Web services. To develop such systems, it is necessary to identify services that can be combined in order to fulfill the functionality and achieve quality criteria of the system being developed. In this paper, we present a framework supporting architecture-driven service discovery (ASD)—that is the discovery of services that can provide functionalities and satisfy properties and constraints of systems as specified during the design phase of the development lifecycle based on detailed system design models. Our framework assumes an iterative design process and allows for the (re-)formulation of design models of service-centric systems based on the discovered services. The framework is composed of a query extractor, which derives queries from behavioral and structural UML design models of service-centric systems based on graph matching techniques. The article describes a prototype tool that we have developed to demonstrate and evaluate our framework and the results of a set of preliminary experiments that we have conducted to evaluate it.

**Keywords:** similarity analysis system design; UML; Web service discovery WSDL

**Introduction**

The development of service-centric systems (SCS)—that is the construction of software systems that deploy autonomous Web services that can fulfill various functional and quality characteristics—is increasingly recognised as an important paradigm of software system development (Chammabasavaiah, Holley et al. 2003; Kramler, Kapsammer et al. 2005; Papazoglou, 2003).

This paradigm requires the extension of current software development practices with new processes, methods, and tools to support the effective discovery and composition of...
Web services into an SCS, in addition to such services, which may also use legacy code or software components. Depending on the stage occurring within the development life cycle of an SCS, Web service discovery (or simply “service discovery” for the purpose of this article) can be distinguished into (Jones, Kozlenkov et al. 2005):

- Early service discovery (ESD)—This is service discovery that occurs in the requirements analysis phase in the development life cycle of an SCS and is driven by its requirements specification,
- Architecture-driven service discovery (ASD)—This is service discovery that occurs during the design phase of the development life cycle of an SCS and is driven by the specification of the functionality, quality properties and constraints of the SCS and the services that the SCS is envisaged to deploy by its design models, and
- Run-time service discovery (RSD)—This is service discovery that occurs during the deployment of an SCS and is concerned with the replacement of existing services of an SCS that becomes unavailable or malfunction during the execution of the system.

The work presented in this article focuses on a framework to support architecture-driven service discovery. This form of service discovery requires the development of capabilities to address some important challenges including:

1. The extraction of service discovery queries from SCS architecture and design models specifying the functionality and quality properties of such systems;
2. The provision of a query language supporting both the expression of arbitrary logical combinations of prioritised functionalities and quality properties criteria for the required services, and similarity-based queries of the form “find a service that is similar to service X”;
3. The efficient matching of service discovery queries against service specifications and return of services that may have varying degrees of match with the queries;
4. The assistance to system designers to select services for an SCS in cases where the discovery process identifies more than one candidate services satisfying a query or services that do not satisfy a query entirely;
5. The integration of the discovered services into an iterative design process in which SCS architecture and design models may be re-formulated following the discovery of services.

The aforementioned challenges have been identified by industrial partners in the areas of telecommunications, automotive, and software in an integrated European project focusing on service-centric system engineering (SeCSE) (“Electronic source,” n.d.). These challenges constitute the main driver underpinning the ASD framework that we present in this article.

Our framework adopts an iterative architecture-driven service discovery process, and assumes the use of UML to specify structural and behavioral design models of SCS. The framework includes a query extractor, which derives queries from UML design models, and a query execution engine, which performs these queries against service registries. The execution of queries is based on a two-stage approach. In the first stage, services which satisfy certain functional and quality criteria are located and maintained for further processing. In the second stage, the fit of the services located in the first stage with the services required by the query is assessed by (a) computing distances between the descriptions of the former and the latter services, and (b) selecting the set of the former services that has the minimum aggregate distance to the services required by the query.

The remainder of this article is structured as follows. In the second section, we introduce a scenario for ASD that we use subsequently to demonstrate our approach. In the third section, we describe our ASD process and framework.
30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product’s webpage:

www.igi-global.com/article/architecture-driven-service-discovery-service/3100?camid=4v1


www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Cloud Computing for Rural ICT Implementations: Methods, Models, and Architectures
www.igi-global.com/chapter/cloud-computing-for-rural-ict-implementations/217883?camid=4v1a

AOP for Dynamic Configuration and Management of Web Services
www.igi-global.com/article/aop-dynamic-configuration-management-web/3043?camid=4v1a

Ubiquitous Provision of Context-Aware Web Services
www.igi-global.com/article/ubiquitous-provision-context-aware-web/3110?camid=4v1a
Quality Models for Multimedia Delivery in a Services Oriented Architecture
www.igi-global.com/chapter/quality-models-multimedia-delivery-services/26074?camid=4v1a