NEXOF-RA Invitation to Contribute: SERVICE DISCOVERY

Contribution Lead
- **Name: Andras Micsik**
- **Affiliation: SZTAKI**
- **Email: Micsik@sztaki.hu**

TITLE: Semantic service discovery for the masses

DEPENDENCIES: NONE DEFINED

BASIC IDEAS:
- Service discovery needs to integrate semantic approaches but possibly without its complexities.
- Services found should be rank able and comparable to each other
- re-use easily existing service descriptions, and to integrate service discovery with other sides of service management.
- Semantic annotations are of key importance in this respect.
- The discovery engine developed in the INFRAWEBS project combines keyword-based and semantic matching to provide an efficient and flexible solution
- From the usability viewpoint it is very important to be able to compare, rank and explain service matches (or non-matches).
- user preferences should be handled as well

involved in the INFRAWEBS project, where we implemented a service discovery mechanism with several features enhancing the users' comfort

Contribution Lead
- **Name: Giuseppe Laria**
- **Affiliation: Centro di Ricerca in Matematica Pura ed Applicata (CRMPA)**
- **Email: laria@crmpa.unisa.it**

TITLE: Service Discovery

DEPENDENCIES:
- -Services Description Techniques.
- -Design Time Service Composition.

BASIC IDEAS:
- Design of a discovery pattern based on the concept "separation of concerns"
- Objective of reducing the overhead of exchanged messages introduced by the necessary "separation of concerns"
- Agreed conceptual model (both for functional and non-functional properties)
- Current business models demand more automation for service discovery to reduce the human intervention in the discovery of services.
- Ontologies can solve the problem based on the use of semantic descriptions of services as well as the "separation of concerns".
- Service descriptions have to be matched against both the functional properties and non-functional properties.

SEPARATION OF CONCERNS DESCRIPTION:
- Functional properties service discovery involves the process to obtain services which match customer services functionalities.
- The functional properties filter implies the elimination of those services that do not satisfy customer constrains regarding the profile.
- The non-functional properties selecting will match from the initial set of matching services, a ranked list of services that match the non-functional properties as well.

The contributors are involved in the BREIN FP6 European Project that focuses on the integration between semantics, grid and agents technologies.

Contribution Lead
- **Name: Aliaksandr Birukou**
- **Affiliation: University of Trento**
- **Email: birukou@disi.unitn.it**

TITLE: Collaborative web service discovery

DEPENDENCIES:
1) Service description – user requests in our approach can contain partial descriptions of the desired services
2) Design time service composition – our approach can be extended to support discovering of sequences or constellations of web services
3) Context model and universal API – the concept of scene in our approach represents the context of the application where the service is used

BASIC IDEAS:
- This approach falls into the category of "techniques, algorithms, best practices, etc. for semi-automatic service ranking and selection among those service specifications retrieved by the discovery process."
- Service-based application developers often base their services' selections on information from partners, experts in the field, friends, or other people who have had experience with a certain service.
- A system to support such information exchange has been developed.
- Currently developed approach aims at helping one to find a suitable web service at design time based on a "Culture Framework" that provides knowledge based on previously faced similar needs might know suitable services and have experience-based preferences about which of them to use.
- Can be useful for finding services using high-level requests, while when the system is being designed and new specific needs for services are identified
- The Implicit Culture Framework assumes that it's possible to elicit this knowledge by observing the behavior of the involved parties and then encouraging newcomers to behave similarly to more experienced members of a community.
- The System for Implicit Culture Support (SICS) use information about observed actions of the community of developers in order to produce recommendations.

The web service discovery system:
- A developer submits a request, which returns a list of recommended services.
- The request contains:
  - textual description of the goal
  - the name of the desired operation
  - a description of its I/O parameters
  - a description of a desired Web service
  - optional list of preferred features

Contribution Lead
- **Name: Mike Boniface**
- **Affiliation: IT Innovation**
- **Email: mjb@it-innovation.soton.ac.uk**

TITLE: Service Description

DEPENDENCIES: Service Discovery

BASIC IDEAS:
- Service Discovery objective "Strategies, techniques for service discovery applied to some frequent service discovery scenarios in composite services".
- Our analysis of the business model unveils that discovery is a procurement process with different phases and accountable actors establishing business relationships and limiting the liability of purchasing decisions
- We have therefore adopted a discovery through procurement approach that links service discovery to procurement processes by combining workflow, contextualised service registries and security.

Contribution Lead
- **Name: Valentin Sánchez**
- **Affiliation: TECNALIA - Infotech**
- **Email: vsanchez@robotiker.es**

TITLE: Service Discovery

DEPENDENCIES: NONE DEFINED

BASIC IDEAS:
- Services should be supplemented with communicative metadata by which they can be effectively discovered and interpreted.
- The Service Registry/Repository approach:
  - A common mechanism for mediated awareness in the industry is a registry-repository
  - Mediated Registry-Repository: a mediation facility containing a registry and a repository. The registry stores links or pointers to service description artifacts.
  - Federated Registry-Repository: According to OASIS "when like SOA IT mechanisms interoperate with one another, the IT mechanisms may be referred to as federated".
  - Distributed Registry Repository:
    - Services Provider
    - Service Consumer
    - Services Registry-Repository: It stores information about the adjacent nodes

Contribution Lead
- **Name: Professor Neil Maiden**
- **Affiliation: City University London**
- **Email: N.A.M.Maiden@city.ac.uk**

TITLE: User Request-Driven Service Discovery

DEPENDENCIES:
- Service Description and Specification
- Service Representation
- Service Registries and other sources

BASIC IDEAS:
- SeCSE delivers effective service discovery methods, techniques and software tools for use during different phases
- Requirements-based Service Discovery:
  - The environment has three main components:
    - UCaRE, a module to document requirements and generate service queries
    - EDDIE, the service discovery engine
      - EDDIE algorithm supports:
        - 1. Query expansion – the addition of terms in the service query that have the same or similar meaning
        - 2. Term disambiguation – selecting the meaning
      - The EDDIE algorithm has been evaluated experimentally and shown to have good precision and recall.
      - With respect to NEXOF , EDDIE offers a proven service discovery technology using the most common language specification for user requests – structured natural language completely independent of the underlying technologies.
    - the service registries
- Architecture-based Service Discovery:
  - Architecture-based service discovery in SECSE supports the identification of services that can fulfill certain functional roles within the design of a system
  - discovery activity is driven by queries incorporating structural and behavioural design models of systems which need to use software services
  - The architecture-driven discovery process is based on a query engine that supports the matching of the discovery queries against service interfaces expressed in WSDL.
  - algorithms can perform inexact distance based matching, and distinguish between hard and soft constraints
  - The architecture-driven discovery platform has undergone an extensive performance evaluation as an add-on of IBM's Rational Modeller. Both implementations are open source.
- Run-Time Service Discovery:
  - This involves identifying alternative services to replace services that are already part of a service based system but may become unavailable
  - The SECSE runtime discovery platform:
    - provides integrated support for the evaluation of structural, behavioural and context conditions at runtime service discovery
    - supports the discovery of services which are described in widely adopted industrial standards such as BPEL and WSDL.
    - assumes a generic mechanism for the description and acquisition of context information through context operations
    - provides a query language that enables the specification of complex conditions in discovery queries
  - One of the novel characteristics of this platform is its ability to cope with the potential heterogeneity of the context operations of different services.
  - Current work on the platform focuses on its integration with a monitoring framework that would enable the automatic generation of queries from failed monitoring conditions

The contribution reported in this paper was developed and evaluated in the FP6 SeCSE (Service-Centric Systems Engineering) Integrated Project.