Documentation of the SDC Prototype

This document provides the detailed technical documentation of the SDC prototype implementation.

In accordance to the general usage policy of the WSMX system, the SDC prototype is implemented as open-source software. Technically, it consists of two Java packages: the first one provides the implementations for the central components of the SDC technique, and the second one provides the implementation of the used matchmaking techniques. The following provides general information on the availability of the SDC prototype software, and then explains the technical realization in more detail.

Availability

SDC Homepage: `http://members.deri.at/~michaels/software/sdc/`

Owner & Contact: Michael Stollberg, `http://members.deri.at/~michaels/`

Nature: Java Application

Platform: JDK 1.5

Licensing: GNU Lesser General Public License (LGPL)

Copy Right: DERI Innsbruck 2007

Version: 1.3, date: 19 October 2007

Download: `http://members.deri.at/~michaels/software/sdc/SDCstandalone.zip`

Source Control: CVS of the WSMX project (`http://wsmx.cvs.sourceforge.net/`)

Required Libraries: all included in the "SDCstandalone.zip" archive

- WSMO4J – the WSMO API for Java (`wsmo4j.sourceforge.net/`)
- Apache AXIS 2 – an open source SOAP engine (`http://ws.apache.org/axis2/`)
- Apache Log4J – an open source API for inserting logging statements into Java code (`http://logging.apache.org/log4j/`).

Package: `org.deri.wsmx.discovery.caching`

This package contains the main classes of the SDC prototype, in particular the implementations of the three main components `SDC Runtime Discoverer`, `SDC Graph Creator`, and `Evolution Manager` as defined in Figure **??** (see Section **??**). Figure 1 provides comprehensive overview in form of a UML class diagram; it consists of the following classes:

SDCResourceManager loads and stores WSML descriptions (ontology, goal, and Web service descriptions, as well as the SDC Graph Ontology); uses the WSMX Resource Manager for storage in a file system on the local machine

SDCGraphManager provides the basic facilities for managing the SDC Graph Ontology, including ontology instance management for SDC graph elements (goal templates, intersection goal templates, goal graph arcs, discovery cache arcs) and routines for handling the SDC Graph knowledge

SDCGraphCreator implements the `SDC Graph Creator`

SDCGraphCreatorHelper provides helper methods for the `SDC Graph Creator`

SDCGraphEvolutionManager implements the `Evolution Manager`

GoalInstanceManager creation, management, and validation service for goal instances

GoalInstanceSDCDiscoverer implements `SDC Runtime Discoverer` .

Package: `org.deri.wsmx.discovery.caching.matchmaking`

This package contains the implementation of the matchmaking techniques used by the SDC prototype. Figure 2 provides the UML class diagram, consisting of the following classes:

Matchmaker defines all matchmaking needed for the SDC technique

- uses the Web service *VampireInvoker* to perform the matchmaking (see below)
- uses the *POGenerator* for generating the proof obligations (see below)

- the interface *Matchmaker* defines the method skeletons for all matchmaking operations needed for the SDC technique; other implementations of this interface may use different reasoning environments

POGenerator generates the TPTP proof obligations on the client side

- a proof obligation is a logical statement which defines a particular matchmaking operation; this is to be proved by VAMPIRE
- the interface *POGenerator* defines the method skeletons for all types of proof obligations needed for the SDC technique

VampireInvoker Web service implementation class for invoking VAMPIRE on a remote web server; this is a generic facility for invoking VAMPIRE for any proof obligation

- intermediately stores the TPTP proof obligation, invokes VAMPIRE for proving it, and returns the result (as a boolean)
- the Web service is publicly available at `http://138.232.65.138:8080/axis2/services/VampireInvoker?wsdl`; Listing 1 shows the WSDL description that has been generated with the Java2WSDL tool provided by Apache AXIS2

VampireInvokerStub client stub for the Web service (generated by Apache AXIS2).

```
< wsdl:definitions
xmlns:axis2="http://matchmaking.caching.discovery.wsmx.deri.org"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:ns0="http://matchmaking.caching.discovery.wsmx.deri.org/xsd"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:ns1="http://org.apache.axis2/xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://matchmaking.caching.discovery.wsmx.deri.org">
<wsdl:documentation> runs a proof obligation that is provided as
input </wsdl:documentation> <wsdl:types>
 <xs:schema xmlns:ns="http://matchmaking.caching.discovery.wsmx.deri.org/xsd"
    attributeFormDefault="qualified" elementFormDefault="qualified"
    targetNamespace="http://matchmaking.caching.discovery.wsmx.deri.org/xsd">
  <xs:element name="check"> <xs:complexType> <xs:sequence>
     <xs:element name="poContent" nillable="true" type="xs:string"/>
 </xs:sequence> </xs:complexType> </xs:element>
 <xs:element name="checkResponse"> <xs:complexType> <xs:sequence>
    <xs:element name="return" nillable="true" type="xs:boolean"/>
```

```
 </xs:sequence> </xs:complexType> </xs:element>
 </xs:schema>
</wsdl:types> <wsdl:message name="checkMessage">
 <wsdl:part name="part1" element="ns0:check"/></wsdl:message>
 <wsdl:message name="checkResponse">
  <wsdl:part name="part1" element="ns0:checkResponse"/></wsdl:message>
<wsdl:portType name="VampireInvokerPortType">
 <wsdl:operation name="check">
   <wsdl:input xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
        message="axis2:checkMessage" wsaw:Action="urn:check"/>
   <wsdl:output message="axis2:checkResponse"/>
  </wsdl:operation>
</wsdl:portType> <wsdl:binding name="VampireInvokerSOAP11Binding"
    type="axis2:VampireInvokerPortType">
 <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
 <wsdl:operation name="check">
   <soap:operation soapAction="urn:check" style="document"/>
   <wsdl:input><soap:body use="literal"/></wsdl:input>
   <wsdl:output><soap:body use="literal"/></wsdl:output>
 </wsdl:operation>
</wsdl:binding> <wsdl:binding
    name="VampireInvokerSOAP12Binding"
    type="axis2:VampireInvokerPortType">
 <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"
     style="document"/>
<wsdl:operation name="check">
 <soap12:operation soapAction="urn:check" style="document"/>
  <wsdl:input> <soap12:body use="literal"/> </wsdl:input>
   <wsdl:output> <soap12:body use="literal"/> </wsdl:output>
 </wsdl:operation>
</wsdl:binding> <wsdl:binding
    name="VampireInvokerHttpBinding"
    type="axis2:VampireInvokerPortType">
  <http:binding verb="POST"/>
  <wsdl:operation name="check">
    <http:operation location="check"/>
    <wsdl:input><mime:content type="text/xml"/> </wsdl:input>
    <wsdl:output><mime:content type="text/xml"/> </wsdl:output>
  </wsdl:operation>
</wsdl:binding> <wsdl:service name="VampireInvoker">
 <wsdl:port name="VampireInvokerSOAP11port_http"
    binding="axis2:VampireInvokerSOAP11Binding">
   <soap:address location="http://138.232.65.138:8080/axis2/services/VampireInvoker"/>
 </wsdl:port>
 <wsdl:port name="VampireInvokerSOAP12port_http"
    binding="axis2:VampireInvokerSOAP12Binding">
   <soap12:address location="http://138.232.65.138:8080/axis2/services/VampireInvoker"/>
  </wsdl:port>
   <wsdl:port name="VampireInvokerHttpport1"
    binding="axis2:VampireInvokerHttpBinding">
      <http:address location="http://138.232.65.138:8080/axis2/rest/VampireInvoker"/>
  </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

Listing 1: WSDL Description of VAMPIRE Invoker Web Service

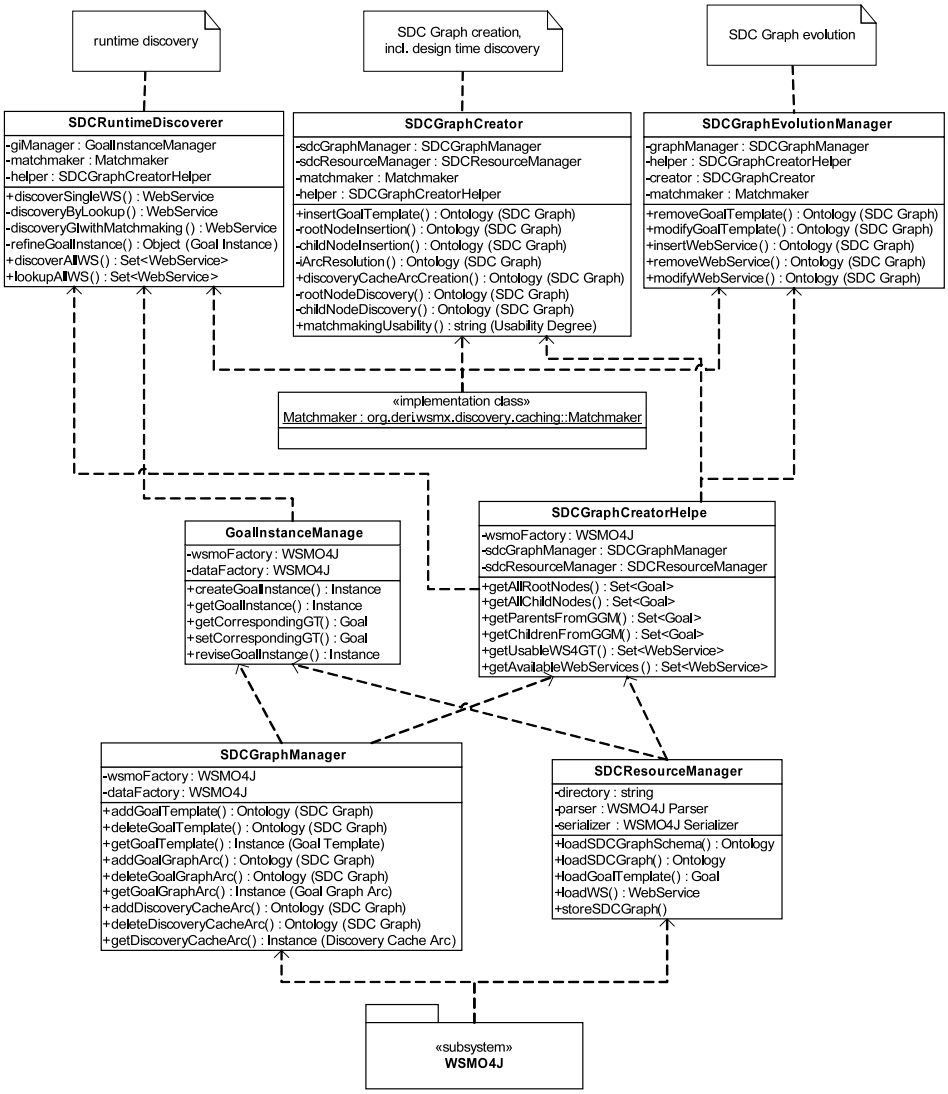

Figure 1: UML Class Diagram of SDC Prototype

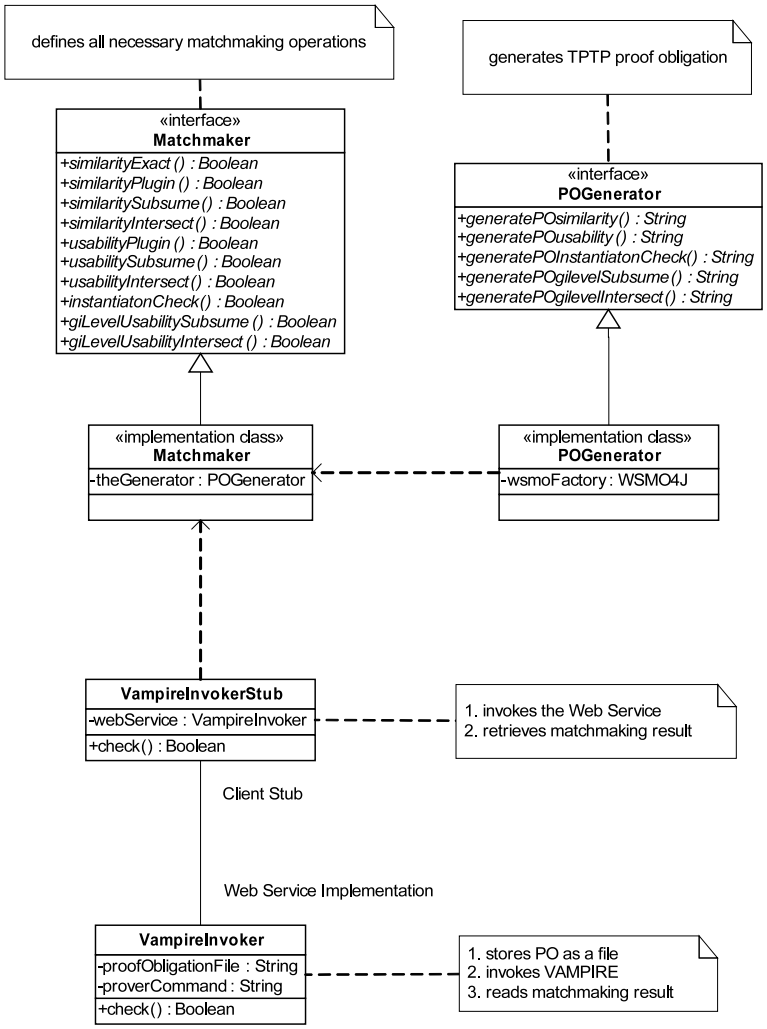

Figure 2: UML Class Diagram of Matchmaker used in SDC Prototype