A P2P Service Description Language Specification:  
Technical Report

A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou

Dept. Of Informatics and Telecommunications, National and Kapodistrian University of Athens
{atsalga, gathanas, michaelp} @di.uoa.gr

Abstract: Service Oriented Computing is branded as the software engineering trend which will reshape the way distributed applications and systems are built. This new trend has reformed existing distributed application development paradigms such as grid and peer-to-peer (p2p) paradigms. Although, the notion of service has been firmly incorporated within the grid computing paradigm, the p2p computing paradigm has not ensued a similar approach. Hence, existing p2p computing frameworks and platforms have leveraged proprietary approaches in incorporating the notion of service.

Introduction

The term “peer-to-peer” refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner. These resources encompass computing power, data (storage and content), network bandwidth, and presence (computers, human, and other resources). The critical function can be distributed computing, data/content sharing, communication and collaboration, or platform services. Decentralization may apply to algorithms, data, and meta-data, or to all of them. This does not preclude retaining centralization in some parts of the systems and applications if it meets their requirements. Typical P2P systems reside on the edge of the Internet or in ad-hoc networks.

The emergence of the Service Oriented Computing (SOC) paradigm has affected the design of such p2p platforms and systems. The term p2p service has emerged as a result of the merging of the p2p and service oriented computing paradigms. Nevertheless, the p2p service term hasn’t kindled a consensus yet. Each of the existing p2p platforms and systems that have turned towards the service oriented computing model incorporates its own arbitrary set of properties and characteristics for the description of p2p services.

As it has been stated in [1], p2p services can be classified either as coarse services providing high level business logic, such as file sharing and instant messaging or as elementary services providing basic p2p network functionality, such as discovery of nodes and resources, message routing and message exchange. As far as coarse p2p services are concerned, these may be defined as "the provision of resources or the execution of tasks of one or more (temporarily provider) peers on behalf of one or
more (temporarily user) peers in a P2P network” [2], whereas elementary p2p service may be defined as “services that support basic functionality in a P2P system, such as discovery of peers or resources, peer membership management, query formulation and routing, etc”.

The diversity among the types of p2p services i.e. coarse and elementary p2p services, along with the proliferation of p2p systems has made it difficult to establish a widely accepted set of standards and protocols that would facilitate the description of p2p services. Each one of the contemporary p2p networks accommodates its own proprietary set of protocols, which are arbitrarily designed so as to enable their specific features and properties.

Since there is no common infrastructure, protocols and standards used for the specification of p2p services, we will have to look at each of the existing proprietary p2p platforms and systems. For each of the addressed p2p system/platform we will contrive a conceptual model describing its constructs and their interrelationships.

Existing P2P Service Platforms

The list of available p2p systems and platforms is enormous. Hence, we need to specify a set of criteria that are going to help us in identifying the systems and platforms that are going to be tackled. The list of selection criteria that have been identified consists of:

- Acceptance: The selected platforms and systems should be widely accepted and used by developers, p2p community, end users, etc. This will help us in having a gross effect on the existing technology and systems.
- Service notion support: It is desirable to select platforms and systems that accommodate the notion of service. Such systems will enable the specification of the “p2p service” term and the identification of properties and features that such services should have.
- Openness: The selected platforms and systems should follow an open source license approach and be based on open and extendable protocols and standards. This will facilitate the identification of their architecture and the provision of possible extensions that might be needed for accommodating the p2p service concept.
- Application domain: The selected platforms and systems should be able to support the development of applications for as many as possible application domains. This will enhance the effect of the systems as far as market penetration is concerned.

Based on the aforementioned criteria list the set of systems and/or platforms that are going to be addressed by this deliverable are:

- JXTA [3]
- Gnutella [4]
- Edutella [5]
JXTA

JXTA [3] is a set of open, generalized peer-to-peer protocols that allow network connected devices to communicate and collaborate as peers. JXTA protocols are independent of any programming language and up to now there have been several implementations for different environments. In general we may regard JXTA as a framework that can be used for the development of p2p networks. Daughter

The provided JXTA protocols standardize the services that are used for:

- Discovering peers
- Organizing peers into peer groups
- Advertising and discovering network services
- Handling peer communication
- Handling peer monitoring

![JXTA Platform Architecture](image)

A JXTA p2p network is composed of a set of interconnected peers which can be organized in peer groups. Peer groups provide a common set of services such as document sharing, instant messaging, etc. Peers and Peer Groups advertise themselves along with their provided services so that other peers may know how to connect and interact with their respective services. Peers can communicate with each other through pipes, which bind to specific endpoints (e.g. specific IP address and port) of the interacting peers and provide an asynchronous, one-way communication channel that is used for exchanging messages. Messages are simple XML documents whose envelop contains information such as routing, message digest and credential information and their body consists of application related data. The essential aspects of the JXTA architecture that differentiate it from other network models are:
A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou

- The use of XML documents for the description of the provided resources and the exchanged messages
- Independence of pipes and peers and of peers and endpoints that doesn’t rely on the use of a central naming mechanism such as DNS
- A unified naming scheme and mechanism for addressing resources

**Gnutella**

Gnutella [4] is a simple file sharing protocol. We consider the Gnutella file sharing service as adhering to the first definition of p2p service. More specifically it represents the coarse functionality of the Gnutella system, which is file searching and downloading.

Peers of the Gnutella network contain files and exchange messages, specified by the Gnutella protocol. The Gnutella protocol [4] specifies the processes that are used for bootstrapping a peer, querying for available resources and retrieving requested files. Therefore, upon instantiation a new peer this peer should know the IP address of another Gnutella peer in order to connect to the p2p network. After connecting to the network the peer sends a message (ping) to advertise its presence to all the peers it knows. Every peer, which receives a ping message, replies with a similar pong message that contains the number and the total size of files the peer has.

When a peer wishes to find a file it submits a keyword-based query to the other peers it knows. Pending on if they have such a file, these peers may respond with results and will forward the query to other peers within their knowledge. In order to avoid flooding Gnutella has equipped its queries with a Time-To-Live (TTL) field which specifies the max number of hops a query may have within the p2p network.

If a resource is found and it is selected for downloading, a direct point to point connection is opened between the client and the host of the resource, and the file is downloaded directly over HTTP.

**Edutella**

Edutella [5] is project that aims to provide a multi-staged effort to scope, specify, architect and implement an RDF-based metadata infrastructure for JXTA p2p applications. An initial implementation of this infrastructure has been incorporated in the development of a p2p system for the exchange of educational resources. The exchanged resources are described using schemas like IEEE LOM, IMS, and ADL SCORM.

The provided system was founded on the JXTA platform and is based on the exchange of RDF meta-data. It provides specific services which complement the JXTA layers that are illustrated in Fig. 1. Edutella Services (described in web service languages like DAML-S or WSDL, etc.) complement the JXTA Service Layer, building upon the JXTA Core Layer. Edutella Peers lie in the Application Layer and utilize the functionality provided by the Edutella services as well as possibly other JXTA services.

The initially specified Edutella services are the following:
- Query Service: standardized query and retrieval of RDF metadata (the core Edutella service)
- Replication Service: provides data persistence / availability and workload balancing while maintaining data integrity and consistency
- Mapping Service: translates between different metadata vocabularies to enable interoperability between different peers
- Mediation Service: defines views that join data from different meta-data sources and reconcile conflicting and overlapping information) and Annotation Service (annotate materials stored anywhere within the Edutella Network

Edutella peers are highly heterogeneous in terms of the functionality (i.e. services) they offer. Thus every peer provides a kind of local repository for RDF triples (e.g., a relational database) as well as a kind of local query language (e.g. SQL). Additionally peer might offer more complex services such as annotation, mediation or mapping services [5].

The Edutella query service is the most basic service within the Edutella network. Peers register the queries they may be asked through the query service by specifying supported metadata schemas (e.g., “this peer provides metadata according to the LOM 6.1 or DCMI standards”) or by specifying individual properties or even values for these properties (e.g., “this peer provides metadata of the form dc title(X,Y)” or “this peer provides metadata of the form dc title(X,’Artificial Intelligence’)”). Queries are propagated through the Edutella network to the subset of peers which have registered their interest in this kind of queries. The resulting RDF statements / models are sent back to the requesting peer.

Edutella provides a set of wrappers, which are used for translating queries and results from the common Edutella format to the local format of a peer and vice versa, as well as a set of JXTA-based libraries that are facilitating the connection of peers to the Edutella p2p network.

To handle queries the wrapper uses the common Edutella query exchange format and data model for query and result representation. For communication with the Edutella network the wrapper transaltes the local data model into the Edutella Common Data Model (ECDM) and vice versa, and connects to the Edutella Network using the JXTA p2p primitives, transmitting the queries based on the ECDM in RDF/XML form (Fig. 2). The ECDM is based on Datalog which is a non-procedural query language that shares with relational databases and with RDF the central feature, that data are conceptually grouped around properties.

In order to handle different query capabilities, several query exchange language levels (RDF-QEL-i) are defined describing what kind of queries a peer can handle (conjunctive queries, relational algebra, transitive closure, etc.)
Language Overview

PSDL (P2P Service Description Language) is an XML [ref] based language built for facilitating the annotation of p2p service descriptions with additional elements and concepts so as to leverage the provision of advanced service discovery and invocation mechanisms. Based on the service model described in [1] a p2p service has an interface similar to that of a web service and accommodates specific binding and invocation mechanisms which rely on the underlying p2p platform infrastructure.

WSDL [8] on the other hand is an XML language that has been branded as the de-facto standard for the description of web services. WSDL accommodates a conceptual model that enables the description of a service’s abstract interface along with its binding details. Although WSDL has primarily focused on the description of web services, its information model can be extended so as to support the definition of additional binding mechanisms apart from the ones that are originally supported, i.e. SOAP [7], HTTP [6], SMTP [ref].

WSDL therefore may easily provide the basis for the development of PSDL. Both languages accommodate the same constructs and structure for the description of a service’s abstract interface. Thus, PSDL can easily reuse the WSDL elements which cater for the description of a service’s abstract information part. This abstract part includes WSDL top level elements such as:
- Type
- Message
- Operation
- PortType

For the definition of a p2p service’s binding information, appropriate extensions to the WSDL elements which cater for the description of a service’s concrete information part can be provided. The WSDL top level elements which may serve as the basis for the provision of such extensions are:
- Binding
- Port
- Service

In the following we present the necessary concepts along with their associations that are used for the description of a p2p service.

Language Structure

PSDL is a language that has been built so as to enable the provision of advanced p2p service discovery and invocation mechanisms. Although, according to the service model presented in [1] a service description may cater for the definition of a service’s semantic and quality properties, PSDL in its current version focuses on the definition
of a P2P service’s syntactic concepts and provides the necessary extension points for accommodating quality and semantic information.

![Diagram of P2P Service Description Model](image)

**Fig. 3.** P2P Service description model

According to **Fig. 3** a P2P service represents a service that is provided by a peer or a set of peers in a P2P network. According to [1] a service is decomposed into a set of syntactic elements that are presented in **Fig. 4**.

![Diagram of Service's Syntactic Structure](image)

**Fig. 4.** A Service’s syntactic structure

A P2P service as it can be seen in **Fig. 3** may be provided by a peer or a set of peers in a P2P network. According to [1] a peer in a P2P network can invoke P2P services
offered by other peers within that network. Peers in a p2p network may formulate
groups which may declare a set of p2p services that are supported by all members of
that group. Therefore, we may identify two distinct types of services; a peer service
which may be provided by a peer in a p2p network and group services which are
provided by all members of a peer group.

A PSDL description therefore is a description document which apart from the
information that is normally conveyed within a service description document contains
information that supports the description the peer or the peer group which provide a
specific p2p service pending on the service type.

The p2p service model presented above can be further refined so as to cater for the
description of specific types of p2p services such as JXTA services. Such a
refinement is presented in the following section where extensions for JXTA to the
above mentioned concepts are presented.

**JXTA P2P Service**

The p2p service concepts introduced in the previous section are generic enough so as
to support the description of any type of p2p service. Nonetheless, in the case of
JXTA service’s further refinements may be provided so as to facilitate a service
description with additional elements that are provided by the JXTA platform to
support the notion of service.

A JXTA refined PSDL description document is a specialization of the PSDL
concept that is presented in Fig. 3 which caters for the description of additional JXTA
related constructs. The additional constructs that are illustrated in Fig. 5 cater for the
description of a JXTA service’s binding information.

A JXTA refined PSDL description document is thus associated to the
ModuleSpecificationAdvertisement construct which is a JXTA specific advertisement
of the service and to the PipeAdvertisement which is a JXTA specific advertisement
of the Pipe that is used for exchanging messages with that service. According to the
JXTA p2p service model presented in Fig. 5 PSDL can be used for the description of
JXTA services which use pipes for exchanging messages between the service and its
respective client.
The aforementioned constructs along with their respective associations can be used in the specification of the PSDL language. The XML elements that were built for the representation of the aforementioned concepts are presented in the following section.

Language Elements

WSDL served as the basis for the provision of the necessary constructs, illustrated in the previous sections that facilitate the description of p2p services. As it has been stated before, both WSDL and PSDL accommodate the same concepts for the description of a service’s abstract information part, hence there we are not going to present them again. Within this section we are going to illustrate the provided extensions to the WSDL elements for the representation of a p2p service’s binding mechanisms.

PSDL is XML-based hence the language that will be used hereafter for the definition of the necessary extensions is going to be XML Schema [9]. The namespaces that were used as the basis for the development of appropriate extensions along with their associated prefixes are presented in Table 1.

Table 1. Imported Namespaces and associated prefixes

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
</tr>
<tr>
<td>tns</td>
<td><a href="http://metis.di.uoa.gr/Sodium/gesmo/services/p2p/">http://metis.di.uoa.gr/Sodium/gesmo/services/p2p/</a></td>
</tr>
<tr>
<td>wsdl</td>
<td><a href="http://schemas.xmlsoap.org/wsd/">http://schemas.xmlsoap.org/wsd/</a></td>
</tr>
</tbody>
</table>

In the following we illustrate the extensions that were provided so as to accommodate the description of p2p services. For each of the introduced elements we present its description, its XML structure and details on its attributes and properties.

PSDL Extensions

The provided WSDL extensions have been primarily influenced by the JXTA platform. Nonetheless, extensions can be built so as to cater for the description of p2p services which are provided by different platforms. In the following we describe only the first level types and elements that were introduced for supporting the description of JXTA services. The details of additional types of elements that were introduced so as to enable the structuring the provided extensions can be found at [ref to psdl schema].

BindingType

In order to support the specification of JXTA binding schemes we introduced the “bindingType” type. This type is used for extending the wsdl:Binding type so as to declare that this is a JXTA binding as well as for specifying the action and the interaction pattern that will be supported by this binding.
As it is illustrated in Fig. 6 an element of this type has two optional attributes; the “action” attribute which is used for the specification of the type of actions that are supported i.e. send-receive, send or receive, and the “pattern” attribute which is used for the specification of the interaction type, i.e. synchronous or asynchronous.

The type of binding that is supported by a p2p service depends on the type of the pipe that will be used for communicating with that service. Hence, if a service uses a bidirectional pipe for receiving and replying to incoming requests this service supports send-receive actions. Pending on if this is a blocking or non-blocking interaction this binding could be synchronous or asynchronous interaction pattern.

```xml
<xsd:complexType name="bindingType">
  <xsd:attribute name="action" type="tns:bindingActionType" use="optional" default="sendreceive"/>
  <xsd:attribute name="pattern" type="tns:bindingActionModel" use="optional" default="synchronous"/>
</xsd:complexType>

<xsd:complexType name="tJXTABindingType">
  <xsd:complexContent>
    <xsd:extension base="wsdl:tBinding">
      <xsd:sequence>
        <xsd:element name="binding" type="bindingType" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Fig. 6. BindingType type declaration

**PipeListType**

“PipeListType” type was introduced so as to cater for the specification of the JXTA Pipes that will be used when interacting with a JXTA service. An element of this type contains the list of pipes that one may use to communicate with a service. Each list of pipes has a unique name which is used for referring to that list (see Fig. 7).

For each pipe offered by a service a link to its associated PipeAdvertisement along with its name are provided. Furthermore, each pipe is associated to a specific binding scheme which defines the supported interaction patterns as well as the operations accessible through this pipe.
The “jxtaGroup” group of elements defines the necessary additions to the wsdl:Port element which will facilitate the specification of the concrete information that is needed when invoking JXTA services. This group of elements includes information about the provider of a p2p service, the list of pipes one may use to access this service as well as the associated ModuleSpecificationAdvertisement of that service (see Fig. 8).

According to the p2p service model introduced in [1] a p2p service can be provided either by a peer or by a group of peers. In order to facilitate the specification of the p2p service provider we have introduced the “provider” element which provides the necessary attributes so as to enable the specification of the provider’s unique id, its name and its type i.e. either a peer or a group.

Based on JXTA [3] a service may incorporate a set of pipes to support the exchange of messages with that service. Therefore, we have introduced the “pipes” element which is of “PipeListType” type.

Additionally, in order to support the association of a PSDL description with a specific ModuleSpecificationAdvertisement of a p2p service as it’s illustrated in Fig. 5 we have introduced the “moduleAdv” element. This element provides attributes which cater for the specification of the advertisement’s name as well as a link to the advertisement file.
A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou

**Fig. 8. JxtaGroup group declaration**

```xml
<xs:group name="jxtaGroup">
  <xs:sequence>
    <xs:element name="provider">
      <xs:complexType>
        <xs:attribute name="id" type="xs:NCName"/>
        <xs:attribute name="name" type="xs:NCName"/>
        <xs:attribute name="type" type="ProviderType" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="pipes" type="pipeListType"/>
    <xs:element name="moduleAdv">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:anyURI">
            <xs:attributeGroup ref="tns:jxtaAdv"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:group>
```

**tJXTAServicePort**

The “tJXTAServicePort” type extends the wsdl:tPort type by appending the additional elements contained in the “jxtaGroup” group of elements. The provided type can be used in a WSDL description document instead of the wsdl:tPort type.

```xml
<xs:complexType name="tJXTAServicePort">
  <xs:complexContent>
    <xs:extension base="wsdl:tPort">
      <xs:group ref="jxtaGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**Fig. 9. JxtaServicePort type declaration**
Conclusions

The emergence of the service oriented computing paradigm has reformed the existing distribute application development approaches. The peer-to-peer computing paradigm has been influenced by this trend and some of the existing platforms and systems are moving towards the service oriented model. The merging of the p2p and service oriented models has given rise to the “p2p service” paradigm.

However, the “p2p service” term hasn’t yet reached a consensus among the peer to peer community members. Each of the existing platforms and systems supporting the notion of service has attributed p2p services with its own specific properties and characteristics and incorporates its own mechanisms and protocols to support the description, discovery and invocation of such services.

In order to overcome this heterogeneity a generic service model has been provided in [1], which caters for the modeling of common as well as distinct characteristics of various types of services. This model provided the basis for the development of a p2p service description language that facilitates the description of JXTA services. The structure and the concepts of this language are presented within this report.

PSDL (P2P Service Description Language) is an XML based language which has been based on WSDL v1.1 [8] that supports the description of p2p services. Although, PSDL in its current version caters for the description of JXTA services its underlying conceptual model enables the description of p2p services which may be provided by other kinds of p2p platforms and systems.

PSDL in its current version caters primarily for the description of syntactic information enabling thus the development of advanced discovery and invocation mechanisms. Nevertheless, its underlying conceptual model caters for the specification of semantic as well as quality information. Additional semantic and quality concepts and properties may be used in the development of advanced description and discovery mechanisms.

The introduced p2p service description language should not be considered as another service description language. PSDL is a set of extensions to the WSDL elements used for the specification of a service’s concrete information model. The extended concepts and their corresponding elements that were introduced within this report can be used in the description of a p2p service by replacing the WSDL elements they extend.

We choose to use WSDL as the basis for the provision of appropriate extensions in order to be able to reuse the existing middleware and tools that have already been built upon it. Furthermore, since WSDL is considered as the de-facto standard in service description we are able to reuse the existing expertise and knowledge of service developers and thus avoiding the introduction of a cumbersome language that one should learn from scratch.
References