On the Topological Landscape of Web Services Matchmaking

PENNSTATE

Hyunyoung Kil, Seog-Chan Oh, and Dongwon Lee

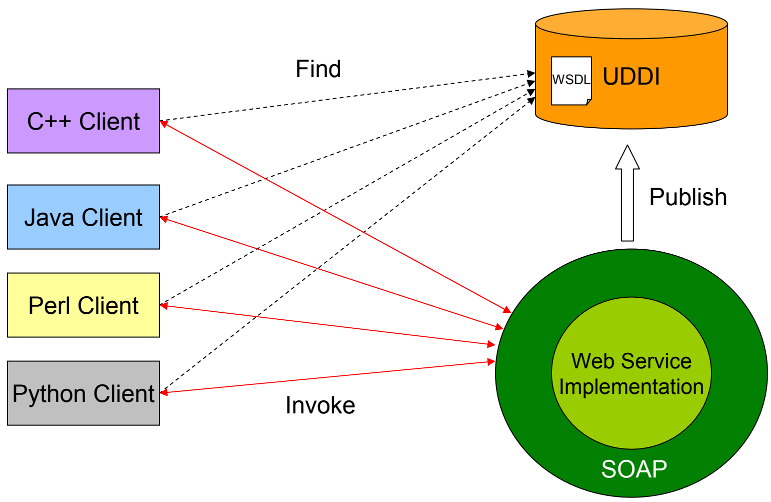
The Pennsylvania State University
USA

SMR 2006 Talk

Motivation

- How useful are current web services?
  - Very little research
- Web services in WSDL form a network
  - Network analysis technique
- The topology of networks often help understand its behavior:
  - How to describe a large network quantitatively ?
  - How were networks formed?
  - What are the consequences of a specific network organization?

Web Services



WSDL

- Web Services Definition Language
- WSDL provides a way for service providers to describe the basic format of web service requests over different protocols or encodings
- WSDL is used to describe *what* a web service can do, *where* it resides, and *how* to invoke it
- Similar to IDL of CORBA

WSDL

- A *web service, ws*, consists of operations
- An *operation, op,* consists of input and output parameters: *op(IN, OUT)*
  - One-way
  - Request-response
  - Solicit-response
  - Notification
- A *parameter, p,* has name and type
  - *p(name, type)*

Example

findRestaurant(16801, “Korean”)

```
<message name=‘findRestaurant_Request’>
    <part name=‘zip’ type=‘xs:string’>
    <part name=‘foodPref’ type=‘xs:string’>
</message>
<message name=‘findRestaurant_Response’>
    <part name=‘zip’ type=‘xs:string’>
    <part name=‘phone’ type=‘xs:integer’>
</message>
<portType name=‘allRestaurant’>
    <operation name=‘findRestaurant’>
        <input message=‘findRestaurant_Request’/>
        <output message=‘findRestaurant_Response’/>
    </operation>
</portType>
```

op: findRestaurant

IN={zip, foodPref}
OUT = {zip, phone}

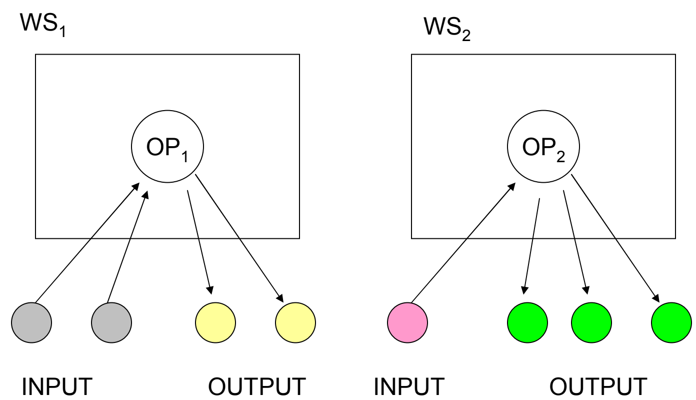
Matchmaking Framework: Parameter Matching

- For two parameters:
  - $p_1(name_1, type_1)$ and $p_2(name_2, type_2)$,
  - Type-Match: p_1 type-matches p_2 iff
    (1) p_1.type == p_2.type or (2) p_1.type is derived from p_2.type.
  - Name-Match: p_1 name-matches p_2 iff
    dist(p_1.name, p_2.name) <= threshold
- *p_1 matches p_2*, ($p_1 \sim p_2$), if p_1 and p_2 are type-match and name-match
- Eg. p("password", xs:string) ~ q("pwd", xs:string)

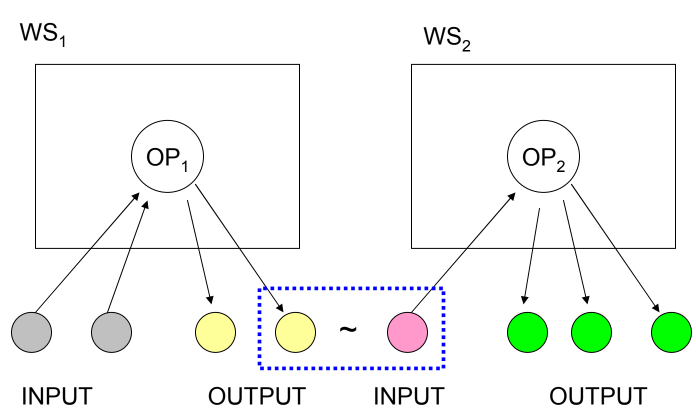
Matchmaking Framework: Operation Invocation

- For two operations
  - $op_1(IN_1, OUT_1)$ and $op_2(IN_2, OUT_2)$,
- Full Invocation (FI)
  - *op_1 fully invokes op_2* if for every mandatory input parameter p in IN_2, there exists an output parameter q in OUT_1 such that q~p.
- Partial Invocation (PI)
  - *op_1 partially Invokes op_2* if there exists any mandatory input parameter p in IN_2, an output parameter q in OUT_1 such that q~p.

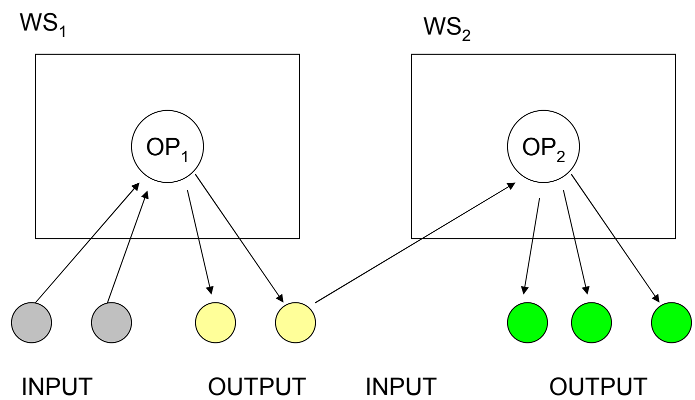
Matchmaking Framework: Web Service Network Model



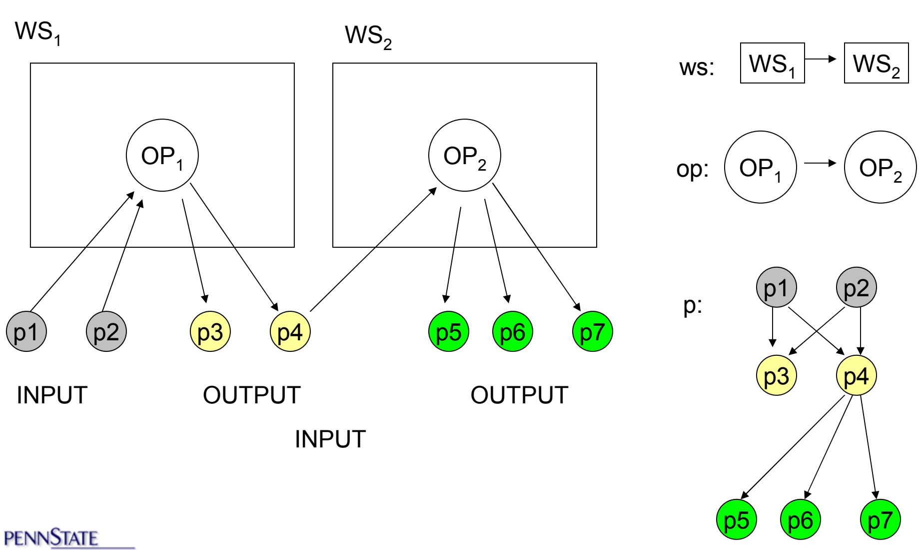
Matchmaking Framework: Web Service Network Model



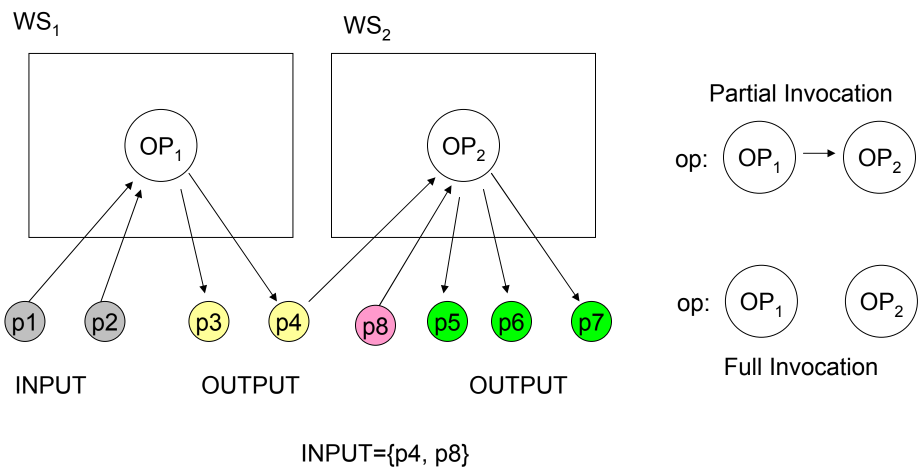
Matchmaking Framework: Web Service Network Model



Matchmaking Framework: Web Service Network Model



Matchmaking Framework: Web Service Network Model

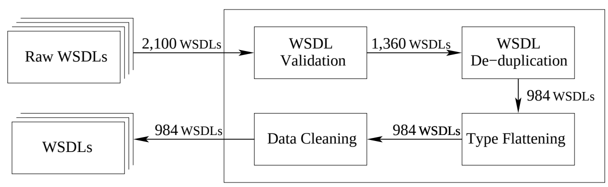


Plan of Study

- Use real-world public WSDL files
- Study the topology of various networks formed from the downloaded WSDL files
  - Small-world
  - Power-law
- Use different distance metrics and thresholds in parameter matching
  - Exact, Cosine with TF/IDF, WordNet

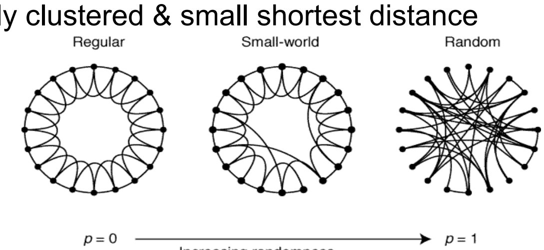
Data Pre-Processing

- Data Gathering: 2,100 WSDLs
  - 1,554 files from Fan et al. + top-1,000 WSDLs from Google
- WSDL Validation: 1,360 WSDLs are left
  - After removing 740 invalid WSDL files based on WSDL standard.
- WSDL De-duplication: 984 WSDLs are left
  - After removing 376 duplicate WSDL files at operation level.
- Type Flattening
  - When a parameter has not a simple type, we flatten the type.
  - ex) p1(address, addressType1), where addressType1 is {integer zipcode, string street, string city, string state}, changes to a set of parameters, p11(zipcode, integer), p12(street, string), p13(city, string) and p14(state, string).
- Data Cleaning
  - Improve data quality, e.g., replacing too general names such as “return”, “result”.



Small-World Network Model

- Watts & Strogatz (1998)
- Arrange N nodes in a ring and connect each node to k others in each direction. (each node degree = 2k.)
- With probability *p* “re-wire” each connection from node i to a new node
- Small-world networks show both regularity and randomness
  - Highly clustered & small shortest distance



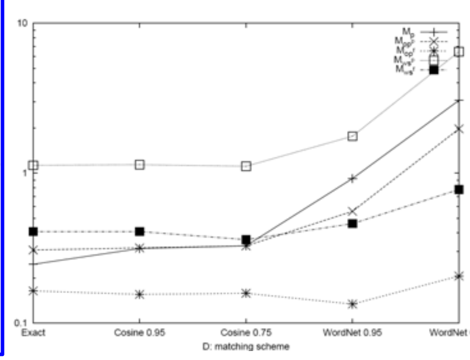
Small-World Network Model

- *L*: average shortest distance among all reachable pairs of nodes
- *C*: average clustering coefficient of all nodes

- $Index_{SN} = |C_{actual} – C_{random}| / |L_{actual} – L_{random}|$
- If a network is small-world, then
  - $C_{actual} >> C_{randdom}$
  - $L_{actual} >\sim L_{random}$
  - => Large $Index_{SN}$

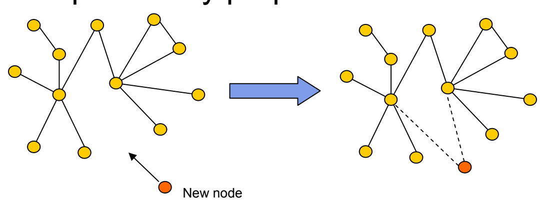
Result1: Small World

Matching Scheme	Network	L_{actual}	L_{random}	C_{actual}	C_{random}
Exact matching	$\mathcal{M}_p$	4.31852	3.42441	0.2229	0.0021
	$\mathcal{M}_{op}^p$	2.8590	1.9830	0.3056	0.0362
	$\mathcal{M}_{op}^f$	3.7605	2.5628	0.2147	0.0180
	$\mathcal{M}_{ws}^p$	2.2710	1.9222	0.4809	0.0874
	$\mathcal{M}_{ws}^f$	2.9659	2.4250	0.2610	0.0405
Cosine (0.95)	$\mathcal{M}_p$	4.1760	3.4442	0.2324	0.0022
	$\mathcal{M}_{op}^p$	2.8651	1.9881	0.3125	0.0340
	$\mathcal{M}_{op}^f$	3.7538	2.5787	0.2001	0.0173
	$\mathcal{M}_{ws}^p$	2.2847	1.9254	0.4925	0.0833
	$\mathcal{M}_{ws}^f$	3.0046	2.4803	0.2499	0.0359
Cosine (0.75)	$\mathcal{M}_p$	4.1981	3.4730	0.2397	0.0020
	$\mathcal{M}_{op}^p$	2.8671	1.9925	0.3190	0.0326
	$\mathcal{M}_{op}^f$	3.7392	2.5910	0.1990	0.0172
	$\mathcal{M}_{ws}^p$	2.2923	1.9307	0.4822	0.0801
	$\mathcal{M}_{ws}^f$	2.9990	2.4394	0.2392	0.0375
WordNet (0.95)	$\mathcal{M}_p$	3.6088	3.3282	0.2612	0.0027
	$\mathcal{M}_{op}^p$	2.4234	1.9425	0.3251	0.0574
	$\mathcal{M}_{op}^f$	3.4165	2.4440	0.1493	0.0190
	$\mathcal{M}_{ws}^p$	2.1222	1.8865	0.5290	0.1138
	$\mathcal{M}_{ws}^f$	2.6215	2.1839	0.2527	0.0510
WordNet (0.75)	$\mathcal{M}_p$	3.2656	3.3665	0.3118	0.0030
	$\mathcal{M}_{op}^p$	2.0546	1.8743	0.4818	0.1256
	$\mathcal{M}_{op}^f$	2.6506	1.9657	0.1842	0.0429
	$\mathcal{M}_{ws}^p$	1.8484	1.7790	0.6697	0.2214
	$\mathcal{M}_{ws}^f$	2.2226	1.9023	0.3487	0.0993



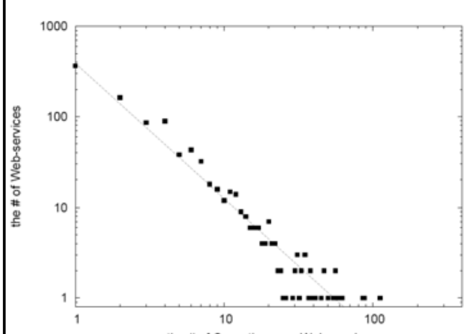
Scale-Free Network Model

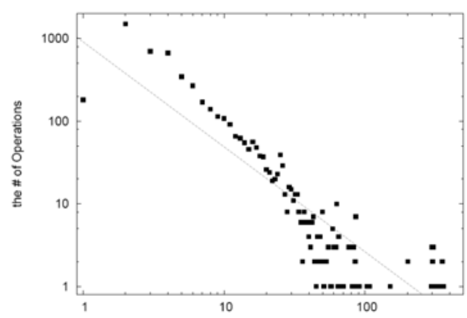
- Barabasi & Albert (1999)
- Small number of nodes with many links (Hubs) and many nodes with only a few links
- Scale-free link distribution often follows power-law
  - the proportion of nodes with a given number of links n is $P(n)=1/n^k$.
- Network grows by addition of new nodes with preferential attachment to the existing nodes based on their number of links with a probability proportional to their degrees



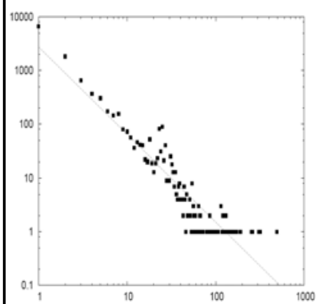
Result2: Power-law

- Complexity of Web services

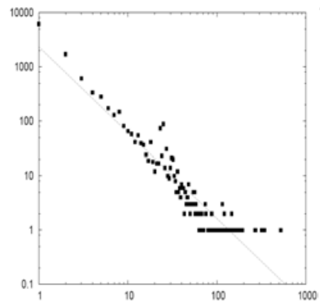




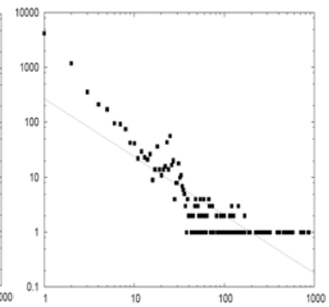
Result2: Popularity of Parameter Names



Exact

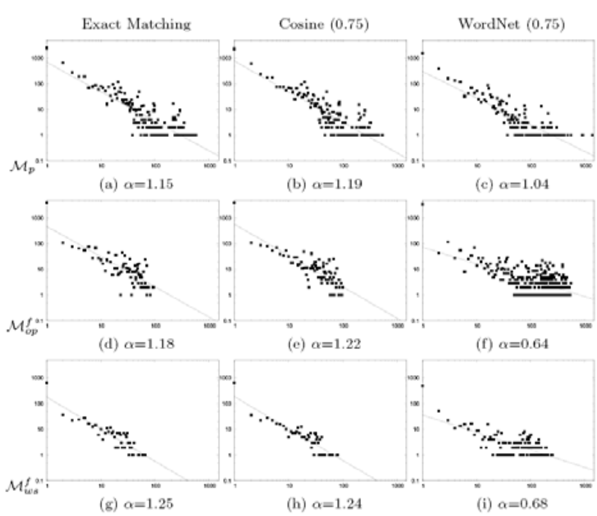


Cosine (0.75)



WordNet (0.75)

Result2: Out-degree Distribution



Conclusion

- Using five matching schemes and three network granularities:
  - Public web services networks show small-world network property
  - Public web service networks follow power-law like distribution pattern
- Semantic Web is needed:
  - Using only exact matching in web service composition is too rigid
  - Using approximate matching helps but not sufficient
  - Semantic web services matchmaking is needed (eg, WSDL-S)
- More study is needed:
  - Giant component size, diameter, formation of networks, etc.