Discovering Relevant Services in Pervasive Environments Using Semantics and Context

Pan Xu

Semantics

- Wiki: Semantics is the study of meaning. It focuses on the relation between *signifiers*, like words, phrases, signs, and symbols, and what they stand for, their denotation.

Motivation

- Most of mobile devices have limited capability such as processing ability, storage space, battery life, network connectivity, etc.
- To provide the user most ``relevant`` services to minimize the wastage of precious devices capacity and bandwidth.

Strategy

- Current service discovery architectures utilize simple string attribute based techniques and often fail to discover the most relevant services efficiently.
- Our strategy/model is to bring together semantics and context for pervasive service discovery.

Main idea

- To express/describe context semantically so that valid deductions can be drawn from it.

Printer Service Scenario

- Bob wishes to print a document and wants to find a nearby printer on the campus. The service request is

Table 2. Bob's service request summary.

Attribute Name	Value	Vital/Rank	Weighting
Service type	Printer	Vital	
Colour	No	Vital	
Price	Cheap	Rank	0.3
Available	Yes	Vital	
Ready (has paper, no jam, etc)	Yes	Vital	
Location	Close	Rank	1
Paper Jams	Low	Rank	1
Print Queue	Short	Rank	0.7

Attribute

The service request includes two kinds of attributes:

- Static: the value changes rarely. e.g. color (Yes/No), price (cheap/expensive).
- Dynamic: the value changes regularly with time. e.g. print queue (short/long), ready(yes/no).

Attribute

- Vital indicates a service must match the attribute value such that it can be included in the recommendation list of discovered services.

- Rank indicates the attribute value will assist to determine a ranking index of relevance for a service.

Attribute

Summary of Bob request:

He wants a non-colored available (open to student) & ready printer which is as cheap and close as possible.

Bob’s current context

Bob’s current context:

(Location---> Outside building A)

Services currently on offer

Table 4. Services on offer.

#	Type	Col-our	Price	Avail-able	Ready	Location	Paper Jams	Print Queue	Rel-eva nt
1	Laser Printer	Yes	50c	Yes	Yes	Building A	0	5	No
2	Laser Printer	No	15c	No	No	Building B	0	0	No
3	Laser Printer	Neg-ative	15c	Affir-mative	Affir-mative	Building C	0.3	0	Yes
4	Laser Copier	No	10c	Yes	Yes	Building G	0.01	5	Yes
5	Ink Jet	Yes	30c	No	No	Building A	0.3	0	No
6	Ink Jet	No	15c	Yes	Yes	Building F	0.2	3	Yes
7	Printer	No	9c	No	Yes	Building L	0.05	0	No
8	Printer Magazine								No
9	Printing Agency					Closed			No

Two quality metrics

Here are two metrics to assess the quality of services returned:

- Recall
- Precision

Two quality metrics

- x: the number of relevant services returned.
- n: the total number of relevant services available.
- N: the total number of services returned.

- Recall=x/n
- Precision=x/N

Current architectures

- Current architectures match request attribute with service description attributes based on exact string matching and do not support context attributes.

Outcome

Here is a ranked service list from the discovery process using syntax based matching, in terms of service type alone:

Table 5. Service list returned by current architectures.

#	Type	Colour	Price	Relevant
7	Printer	No	9c	No
1	Laser Printer	Yes	50c	No
2	Laser Printer	No	15c	No
3	Laser Printer	Negative	15c	Yes
8	Printer Magazine			No
9	Printing Agency			No

Interpretation of the outcome

- A service will be returned if and only if its type value contains the keyword ‘printer’.
- Service 7 ranks first since its type value matches the keyword exactly.
- Recall=⅓; Precision=⅙.

Extension A

If static attributes ‘color’ is also considered, then services 1 and 3 will be ruled out. (‘negative’ does not match ‘no’.)

Table 5. Service list returned by current architectures.

#	Type	Colour	Price	Relevant
7	Printer	No	9c	No
1	Laser Printer	Yes	50c	No
2	Laser Printer	No	15c	No
3	Laser Printer	Negative	15c	Yes
8	Printer Magazine			No
9	Printing Agency			No

Extension B

If context attributes (‘available’ & ‘ready’) are also considered, then services 7 and 2 are removed.

#	Type	Col-our	Price	Avail-able	Ready	Location	Paper Jams	Print Queue	Rel-eva nt
7	Printer	No	9c	No	Yes	Building L	0.05	0	No
2	Laser Printer	No	15c	No	No	Building B	0	0	No

Extension B

The final list returned: services 9 and 8. (9 ranked higher since requested ‘location=close’ matches services 9 ‘location=closed’. Syntactically similar while semantically different.) Recall=Precision=0.

Table 6. Service list returned by current architectures with context.

#	Type	Colour	Price	Available	Ready	Location	Paper Jams	Print Queue	Rele-vant
9	Printing Agency					Closed			No
8	Printer Magazine								No

Semantic data representation

- Context information must be utilized and represented semantically.
- Each requested attribute is matched with service description based on meaning (semantically) rather than symbol (syntax).

Semantic data representation

- Match ‘type=printer’ with all values which has similar meanings. e.g. ‘Ink Jet’, ‘laser copier’, etc.
- Match ‘Ready=yes’ with other words semantically, such as ‘OK, yes, affirmative, good, right’, etc.

Service list returned

Service list returned: Recall=Precision =1.

We see Recall and Precision both improves dramatically by bringing together ‘semantics’ and ‘context’.

Table 7. Discovered service list returned by proposed model using vital attributes.

#	Type	Colour	Available	Ready
3	Laser Printer	Negative	Affirmative	Affirmative
6	Ink Jet	No	Yes	Yes
4	Laser Copier	No	Yes	Yes

How to rank the service list

- Each rank attribute of each service is ranked relative to the corresponding attribute of the other services in the discovered list.
- Each ranking are then weighted according to the weight input by the requester.
- The effective ranking for each attribute are then added to provide an overall rank index.

How to rank the service list

Table 8. Service list ranking achieved by proposed model.

#	Price	Effective Rank	Location	Effective Rank	Paper Jams	Effective Rank	Print Queue	Effective Rank	Overall Rank
3	15c	*2*0.3=0.6*	Building C	*1*1= 1*	0.3	*3*1= 3*	0	*1*0.7=0*	0.6+1+3 +0.7=5.3
6	15c	*2*0.3=0.6*	Building F	*2*1= 2*	0.2	*2*1= 2*	3	*2*0.7=1.4*	0.6+2+2 +1.4=6
4	10c	*1*0.3=0.3*	Building G	*3*1= 3*	0.01	*1*1= 1*	5	*3*0.7=2.1*	0.3+3+1 +2.1=6.4

Challenge

How to represent context information semantically?

Semantic Language

- Semantic languages are still in early stage of development and evaluation.

- Latest research recommends the use of OWL-S. (Web Ontology Language for Services).

OWL-S

- **OWL-S** is an ontology, within the OWL-based framework of the Semantic Web, for describing Semantic Web Services.
- It will enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints.
- Lots of research is ongoing here.

Thanks!

Thanks!