# Web Services Composition

Presented by: Noran Hasan

# Web Services

- Web services allow web applications to publish a  functions and make them available to the world.
- Examples: currency conversion, weather reports, language translation

# Web Service Description Language (WSDL)

- XML-based
- Example: findRestaurant

```
<message name='findRestaurant_Request'>
    <part name='zip' type='xs:string'>
    <part name='foodPref' type='xs:string'>
    </message>
<message name='findRestaurant_Response'>
    <part name='name' type='xs:string'>
    <part name='phone' type='xs:string'>
    <part name='addr' type='xs:string'>
</message>
```

# Web Service Discovery (WSD)

- Finding a web service that can fulfill a request is known as the web service discovery (WSD) problem.

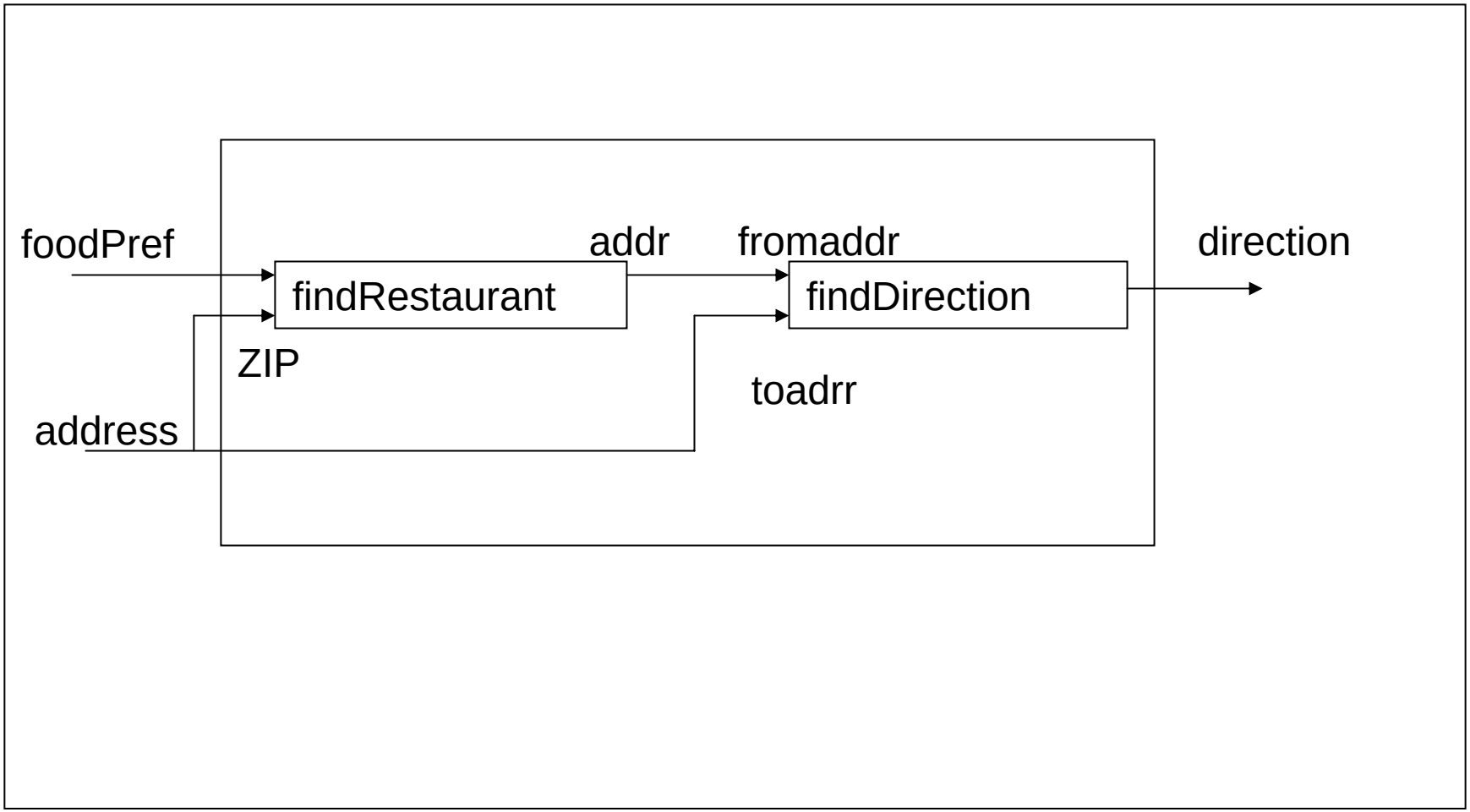
# Web Service Composition (WSC)

- Sometimes, no single web service can satisfy a request on its own.
- Multiple web services need to be combined to provide the desired functionality. This is known as the web service composition (WSC) problem.

# WSC Example

- Request: Find Thai restaurant near hotel with driving directions.
- Web services available:
  1. findRestaurant:
     - Input: ZIP, food preference
     - Output: name, phone, address
  2. findDirection:
     - Input: from address, to address
     - Output: direction

# WSC Example cont.



# Classification of WSC Problem

- Manual vs. Automatic
  - Manual: domain experts
  - Automatic: software programs through matching of parameters
- Simple vs. Complex
  - Simple: AND only
  - Complex: OR, XOR, NOT and constraints
- Small-scale vs. Large-scale
  - Small-scale: exhaustive search
  - Large-scale: approximate algorithms

# Matching schemes

- Exact matching: lexicographically the same
- Approximate matching: similarity is determined from a distance function e.g. “password” and “passwd”, “license-Fee” and “Fee for license”
- Semantic matching: semantic meaning e.g. “cost” and “fee”

# Recent work

- Akkiraju et al. calculate a semantic similarity score which is used by the AI planning algorithm to compose the service.
- This similarity score is determined using both domain-independent ontology, derived from the English thesaurus, and domain-dependent ontology.

# Recent work cont'd

- Blanco et. Al. try to prune the space of possibilities based on a cost metric that estimates the cost of a particular solution.
- They use dynamic programming

# Recent work cont'd

- Hashmian and Mavaddat  store Web services in the form of a graph, which contains relationships between inputs and outputs (dependencies, generalizations and composition) of web services and build a composite web service by finding paths in the graph.

# Recent work cont'd

- Shin an Lee claimed that chaining input and outputs does not guarantee that the composed service will provide the requested functionality.
- They consider functional semantics and is represented by action and object e.g. {Calculate, Distance}.
- The use a data ontology and a domain ontology.
- They use a Service Relation Graph.
- This improves correctness and reduces time-complexity.

# Possible future work

- Client may use more than one action and object to specify functionality needed.

Thank you.