

ORACLE®

Oracle Digital Assistant The Complete Training

Composite Bag

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Topic agenda

- 1 Entities and why we need them
- 2 Composite bag basics
- 3 Composite bag error handling
- 4 Working with entity values
- 5 Slotting entities out of order

Topic agenda

- 1 Entities and why we need them
- 2 Composite bag basics
- 3 Composite bag error handling
- 4 Working with entity values
- 5 Slotting entities out of order

Entities and why we need them – a recap

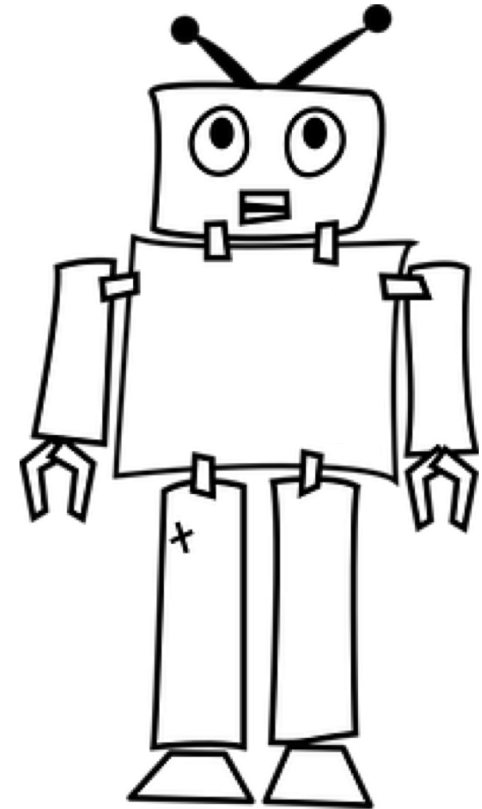
- Important variable elements related to an intent
 - Date, bank account, amount to transfer, pizza size, pizza topping
- Entity slotting
 - Process of filling those variable elements

```
startBalances:  
  component: "System.SetVariable"  
  properties:  
    variable: "accountType"  
    value: "${iResult.value.entityMatches['AccountType']}[0]}"  
  transitions: {}
```

```
askBalancesAccountType:  
  component: "System.List"  
  properties:  
    options: "${accountType.type.enumValues}"  
    prompt: "For which account do you want your balance?"  
    variable: "accountType"  
  transitions: {}
```

```
askBalancesAccountType:  
  component: "System.List"  
  properties:  
    options: "${accountType.type.enumValues}"  
    nlpResultVariable: "iResult"  
    prompt: "For which account do you want your balance?"  
    variable: "accountType"  
  transitions: {}
```

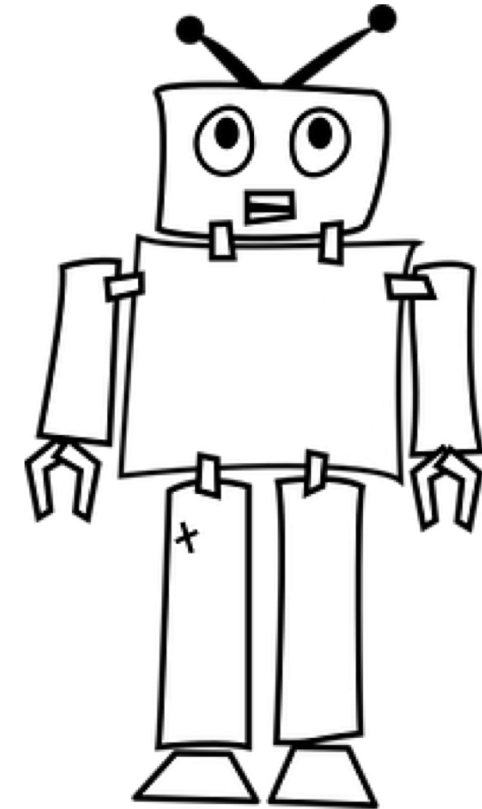
Can anyone see the **problem** with
this approach?



```

resolvesize:
  component: "System.SetVariable"
  properties:
    variable: "size"
    value: "${iResult.value.entityMatches['PizzaSize'] [0]}"
  transitions: {}
resolvecrust:
  component: "System.SetVariable"
  properties:
    variable: "crust"
    value: "${iResult.value.entityMatches['PizzaCrust'] [0]}"
  transitions: {}
resolvetype:
  component: "System.SetVariable"
  properties:
    variable: "type"
    value: "${iResult.value.entityMatches['PizzaType'] [0]}"
  transitions: {}
askage:
  component: "System.Output"
  properties:
    text: "How old are you?"
  transitions: {}
checkage:
  component: "AgeChecker"
  properties:
    minAge: 18
  transitions:
    actions:
      allow: "crust"
      block: "underage"
crust:
  component: "System.List"
  properties:
    options: "Thick,Thin,Stuffed,Pan"
    prompt: "What crust do you want for your Pizza?"
    variable: "crust"
  transitions: {}
size:
  component: "System.List"
  properties:
    options: "${size.type.enumValues}"
    prompt: "What size Pizza do you want?"
    variable: "size"
  transitions: {}
type:
  component: "System.Text"
  properties:
    prompt: "What Type of Pizza do you want?"
    variable: "type"
  transitions: {}

```



The challenge of “real world” entity slotting

- Many values required for an intent
- Error handling for each entity value
- Different prompts should the user error
- Out of order information
- Validation
- Allow multiple values or not?
- Slot entity if specifically said, otherwise, default it

Topic agenda

- 1 Entities and why we need them
- 2 Composite bag basics**
- 3 Composite bag error handling
- 4 Working with entity values
- 5 Slotting entities out of order

Composite bag entity

- Models a business domain object
 - Pizza order, holiday request, expense
- Each composite bag is composed of one to many items
 - Custom entities
 - Built-in entities
 - String, location and attachment
- All contained entities get resolved automatically in a single dialog flow state
 - System.ResolveEntities
 - Automatically displays enumerated values as a list and provides pagination
 - System.CommonResponse

Ordering a pizza with composite bag entity

variables:

order: "PizzaOrder"
iResult: "nlresult"

Configure

states:

intent:

component: "System.Intent"
properties:
variable: "iResult"

"A **cheese** pizza please."

Composite Bag:
PizzaOrder

Entity: PizzaType

Entity: PizzaSize

Entity: PizzaCrust

orderState:

component: "System.ResolveEntities"
properties:
variable: "order"
nlpResultVariable: "iResult"
[...]

Check for Resolved Entities

- PizzaType
- PizzaSize
- PizzaCrust

What size of pizza do
You want?

And what kind of
crust?



+ Entity

More

Filter

Sort By Created Ascending

PizzaBag

PizzaDough

PizzaSize

PizzaTopping

ADDRESS

CURRENCY

DATE

DURATION

EMAIL

NUMBER

PHONE_NUMBER

Description

Name *

PizzaBag

Description

Configuration

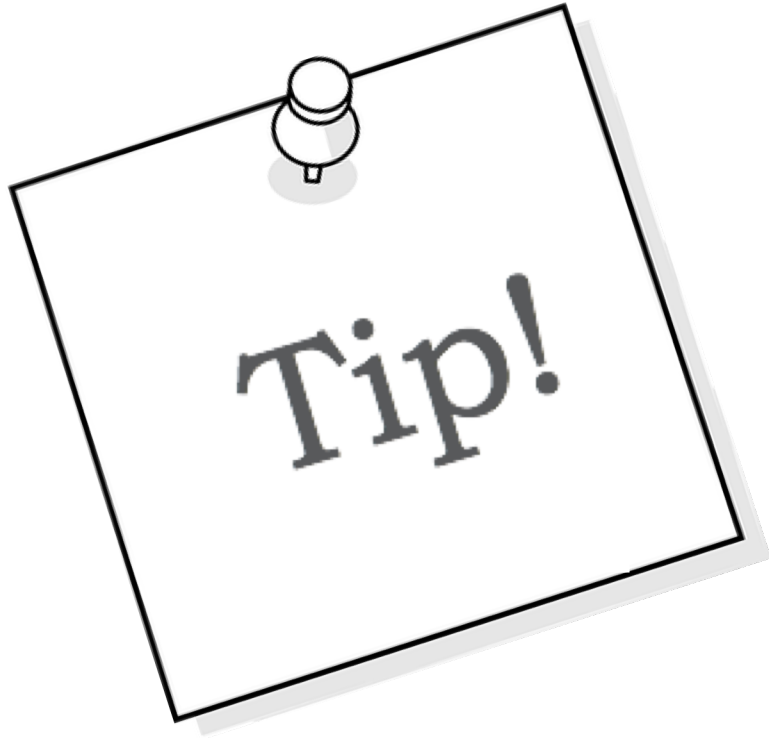
Type ?

Composite Bag

Bag Items

+ Bag Item

Name	Type	Entity Name
PizzaSize	ENTITY	PizzaSize
PizzaTopping	ENTITY	PizzaTopping
DeliveryTime	ENTITY	TIME



Remember to train if
you add an item to
composite bag


```
1 #metadata: information about the flow
2 # platformVersion: the version of the bots platform that this flow was written to work with
3 metadata:
4   platformVersion: 1.0
5 main: true
6 name: GR_Pizza_Composite_Bag
7 #context: Define the variables which will used throughout the dialog flow here.
8 context:
9   variables:
10    iResult: "nlpresult"
11    pizza: "PizzaBag"
12
13 states:
14
15   intent:
16     component: "System.Intent"
17     properties:
18       variable: "iResult"
19       optionsPrompt: "Do you want to"
20     transitions:
21       actions:
22         OrderPizza: "startOrderPizza"
23         WelcomePizza: "startWelcome"
24         unresolvedIntent: "startUnresolved"
25
26   resolveEntities:
27     component: "System.ResolveEntities"
28     properties:
29       variable: "pizza"
30       nlpResultVariable: "iResult"
31       maxPrompts: 3
32       cancelPolicy: "immediate"
33     entityOrder:
34     transitions:
35       actions:
36         cancel: "maxError"
37         next: "showPizzaOrder"
```

Ordering a pizza with composite bag entity

I want a large pizza

ok lets get that order sorted

What kind of pizza would you like

- Meaty
- Veggie
- Hot and Spicy
- American Hot

Meaty

When would you like us to deliver your pizza?

anytime you want

Ok, we need a valid time for delivery. When would you like us to deliver your pizza?

now?

Description

Name *

PizzaBag

Description

Configuration

Type ?

Composite Bag

Bag Items

+ Bag Item

Name	Type	Entity Name
PizzaSize	ENTITY	PizzaSize
PizzaTopping	ENTITY	PizzaTopping
DeliveryTime	ENTITY	TIME

Composite bag prompts

I want a large pizza

ok lets get that order sorted

What kind of pizza would you like

Meaty
Veggie
Hot and Spicy
American Hot

Meaty

When would you like us to deliver your pizza?

anytime you want

Ok, we need a valid time for delivery. When would you like us to deliver your pizza?

now?

- Multiple Values
- Fuzzy Match

Disambiguation Resolution

- Prompt for Disambiguation
- Disambiguation Prompt

Entity Extraction

- Out of Order Extraction
- Extract With
- Prompt for Value

Prompts

+ Prompt

Prompt	Sequence Number
What kind of pizza would you like	1
Which one of our pizzas would you like to try?	2

Composite bag prompts

I want a large pizza

ok lets get that order sorted

What kind of pizza would you like

Meaty
Veggie
Hot and Spicy
American Hot

Meaty

When would you like us to deliver your pizza?

anytime you want

Ok, we need a valid time for delivery. When would you like us to deliver your pizza?

now?

Error Message: Ok, we need a valid time for delivery.

Multiple Values:

Fuzzy Match: Off

Disambiguation Resolution

Prompt for Disambiguation:

Disambiguation Prompt:

Entity Extraction

Out of Order Extraction:

Extract With: PizzaSize

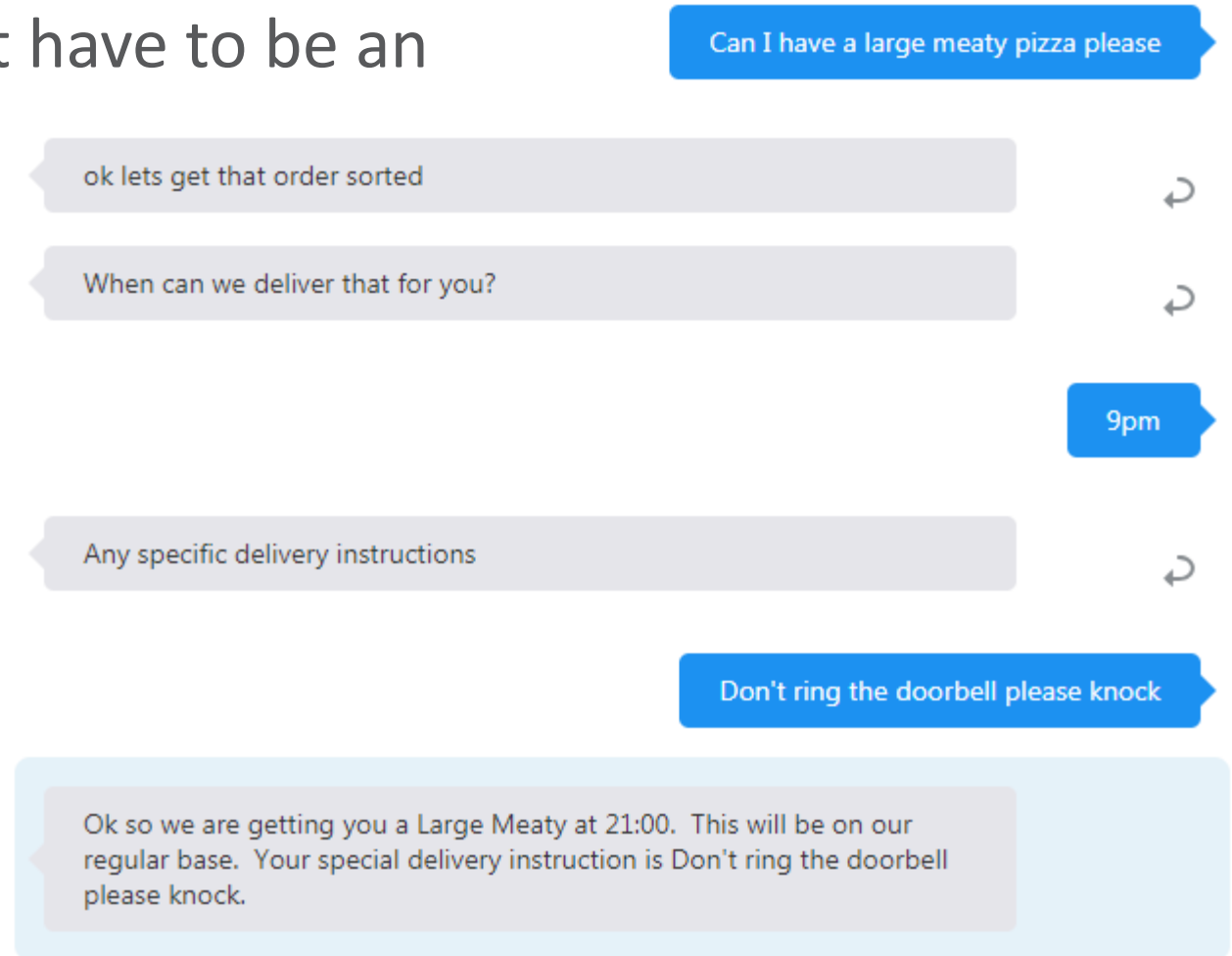
Prompt for Value:

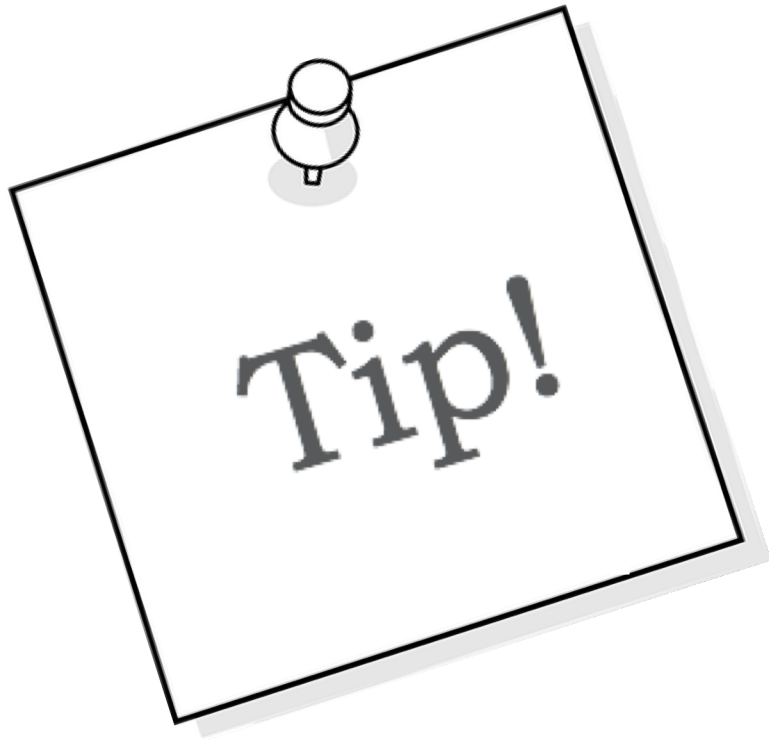
Prompts

Prompt	Sequence Number
When would you like us to deliver your pizza?	1

Composite bag string

- A composite bag entity item doesn't have to be an entity
 - String, location, attachment





Entities are resolved in the **order** in which they appear in the composite bag – you can change the order at design time

resolveEntities resolves composite bag in dialog flow

```
resolveEntities:  
  component: "System.ResolveEntities"  
  properties:  
    variable: "pizza"  
    nlpResultVariable: "iResult"  
    maxPrompts: 3  
    cancelPolicy: "immediate"  
    headerText: "This message appears for each entity"  
  transitions:  
    actions:  
      cancel: "maxError"  
      next: "setPizzaDough"
```

Topic agenda

- 1 Entities and why we need them
- 2 Composite bag basics
- 3 Composite bag error handling**
- 4 Working with entity values
- 5 Slotting entities out of order

Composite bag validation and error handling

- You can define error messages for invalid input
 - This can also include Apache FreeMarker expressions
- You can define validation rules to enforce business rules
 - Also can include Apache FreeMarker expressions
- You can define maximum attempts for valid input
 - Error retries in composite bag overrides maxPrompts in dialog flow

Composite bag validation and error handling

I want a pizza

ok lets get that order sorted

What size of pizza would you like

Small
Medium
Large

Huge

Sorry, 'Huge' is not a valid size of pizza. Please choose small, medium or large pizza.

Small
Medium
Large

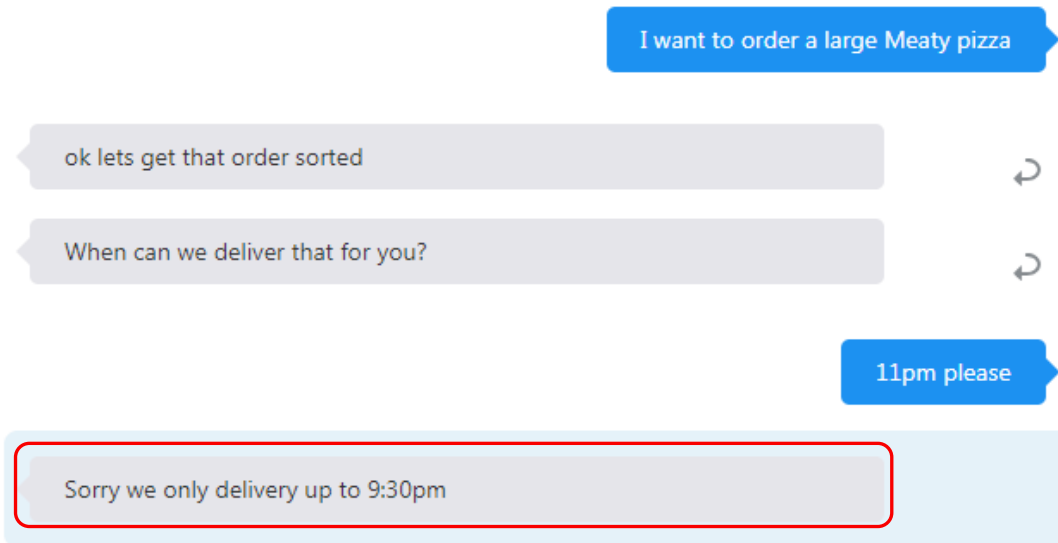
Maximum User Input Attempts 4

Error Message

Sorry, '\${system.entityToResolve.value.userInput!this}' is not a valid size of pizza.

Prompt	Sequence Number
What size of pizza would you like	1
Please choose small, medium or large pizza.	2

Composite bag validation and error handling



+ Validation Rule

Expression	Error Message
<code>\${(pizza.value.DeliveryTime.hrs?number < 10)?then('true','false')}</code>	Sorry we only delivery up to 9:30pm
<code>\${(.(now?date?number < pizza.value.DeliveryTime.date?number)?then('true','false')}</code>	OK we are quick but we don't have a time machine!

Composite bag validation and error handling

- Define the maximum number of retries in dialog flow
- Override within each entity
- Define if failure is
 - Immediate
 - On last entity only (backwards compatibility)

```
resolveEntities:  
  component: "System.ResolveEntities"  
  properties:  
    variable: "pizza"  
    nlpResultVariable: "iResult"  
    maxPrompts: 3  
    cancelPolicy: "immediate"  
  transitions:  
    actions:  
      cancel: "maxError"  
      next: "showPizzaOrder"
```

* Name	<input type="text" value="PizzaSize"/>
Type	<input type="text" value="Entity"/>
Entity Name	<input type="text" value="PizzaSize"/>
Description	<input type="text"/>
? Enumeration Range Size	<input type="text"/>
? Maximum User Input Attempts	<input type="text" value="4"/> <input type="button" value="v"/> <input type="button" value="^"/>

Topic agenda

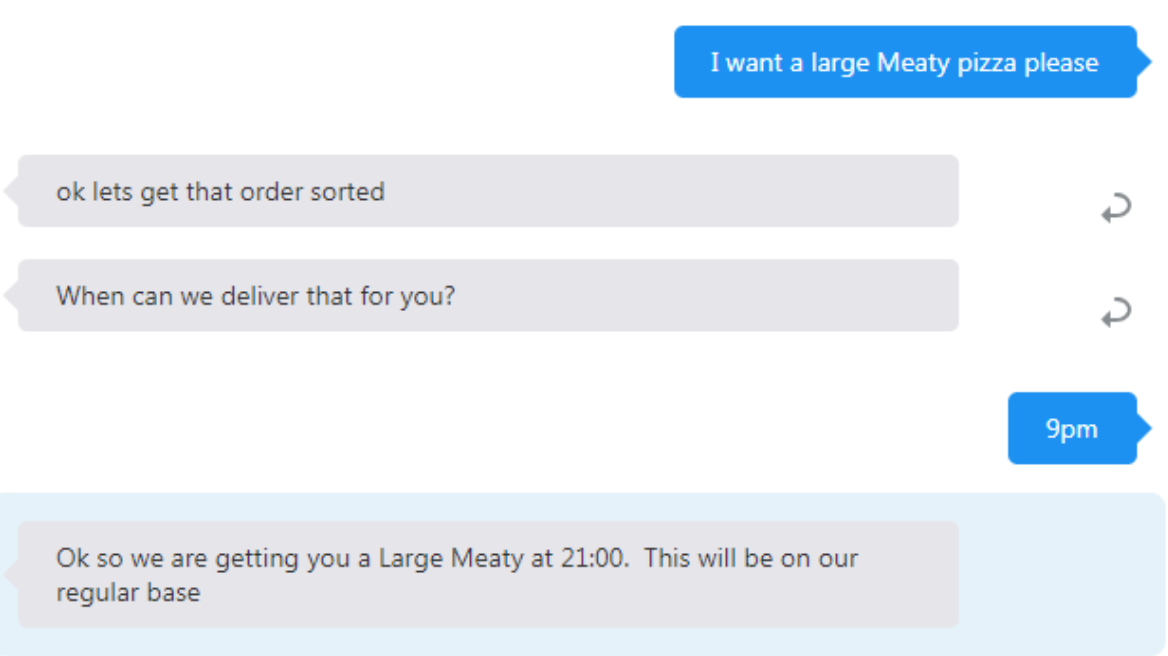
- 1 Entities and why we need them
- 2 Composite bag basics
- 3 Composite bag error handling
- 4 Working with entity values**
- 5 Slotting entities out of order

Defaulting entity values

- Composite bag will slot any entity values in initial sentence
- Then will prompt for other entity values
- What if you want to capture a entity value if mentioned, but not specifically prompt for it
 - Pizza dough
 - Assume regular unless someone specifically asks for gluten-free

- PizzaType
- PizzaSize
- PizzaCrust
- PizzaDough

Defaulting entity values



A chat interface showing a conversation about a pizza order. The user's request is in a blue bubble, and the system's responses are in grey bubbles. A '9pm' timestamp is shown in a blue bubble. The final system response is highlighted in a light blue background.

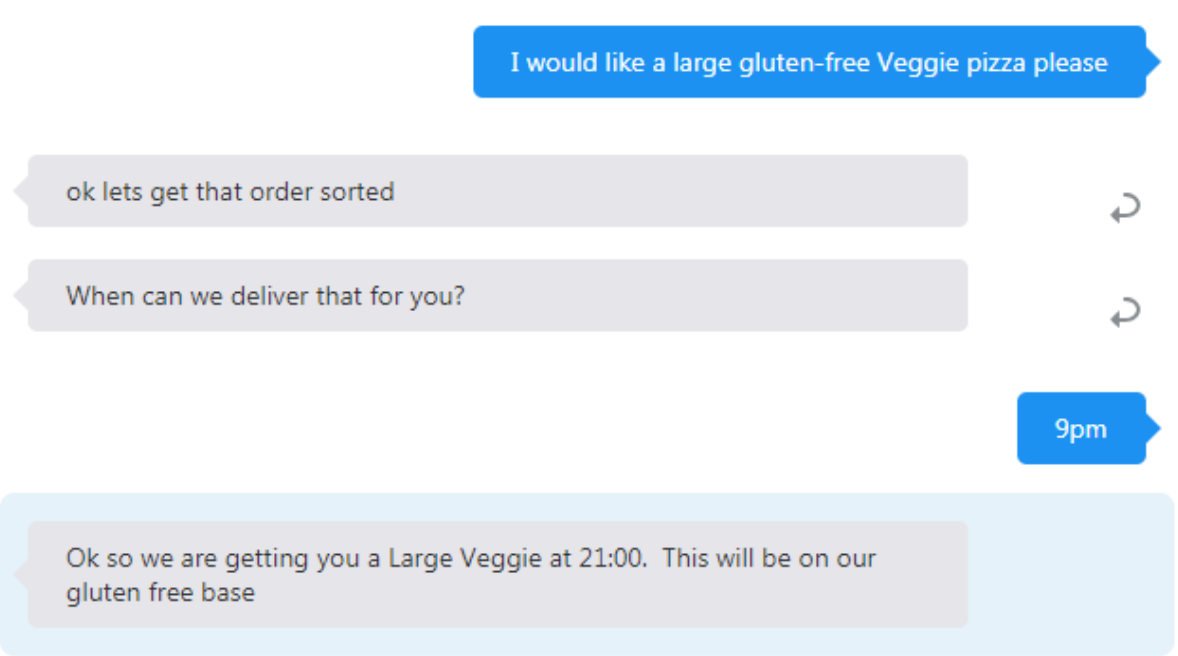
I want a large Meaty pizza please

ok lets get that order sorted

When can we deliver that for you?

9pm

Ok so we are getting you a Large Meaty at 21:00. This will be on our regular base



A chat interface showing a conversation about a pizza order. The user's request is in a blue bubble, and the system's responses are in grey bubbles. A '9pm' timestamp is shown in a blue bubble. The final system response is highlighted in a light blue background.

I would like a large gluten-free Veggie pizza please

ok lets get that order sorted

When can we deliver that for you?

9pm

Ok so we are getting you a Large Veggie at 21:00. This will be on our gluten free base

Defaulting entity values

- Set prompt to false, then populate default in the dialog flow after resolveEntities

Entity Extraction

? Out of Order Extraction

? Extract With

? Prompt for Value

setPizzaDough:

component: "System.SetVariable"

properties:

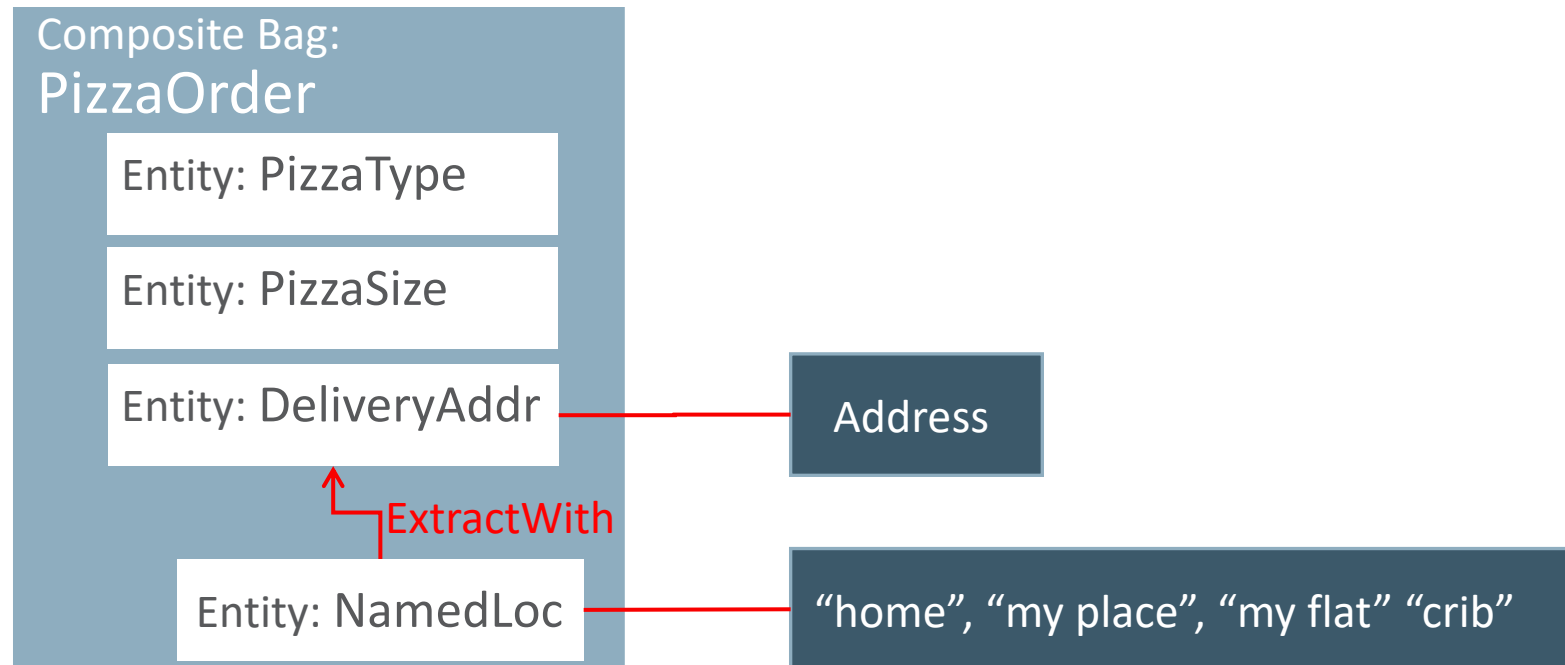
variable: "pizza.PizzaDough"

value set for the variable.

value: "\${pizza.value.PizzaDough?has_content?then(pizza.value.PizzaDough,'regular')}"

Allowing related terms for an entity value

- User may answer in a way which is different from the expected entity value
- DeliveryAddress (primary) NamedLocation (secondary)
 - “What is the delivery address for your pizza” – “home delivery please”



Allowing related terms for an entity value

- Create NamedLocation entity add to bag
 - Don't specifically prompt for secondary entity (Prompt for Value false)
 - Extract with Delivery Address
 - Only prompt for primary if secondary has no content

* Name	<input type="text" value="DeliveryAddress"/>
Type	<input type="text" value="Entity"/>
Entity Name	<input type="text" value="ADDRESS"/>

? Prompt for Value

* Name	<input type="text" value="NamedLocation"/>
Type	<input type="text" value="Entity"/>
Entity Name	<input type="text" value="NamedLocation"/>

? Extract With

? Prompt for Value

Topic agenda

- 1 Entities and why we need them
- 2 Composite bag basics
- 3 Composite bag error handling
- 4 Working with entity values
- 5 Slotting entities out of order**

Slotting entities out of order

- Sometimes the user might supply a new entity value whilst awaiting a value for a different entity

A chat interface illustrating a user providing information for one entity while the system is waiting for information for another. The system asks, "When can we deliver that for you?". The user responds with "Meaty". The system asks again, "When can we deliver that for you?". The user responds with "actually make it a medium". The system asks again, "When can we deliver that for you?". The user responds with "8pm". The system then provides a confirmation: "Ok so we are getting you a Medium Meaty at 20:00. This will be on our regular base". At the bottom, there is a text input field containing "8pm" and a send button.

A chat interface illustrating a user providing information for one entity while the system is waiting for information for another. The user starts with "I want a large meaty pizza please". The system asks, "ok lets get that order sorted". The system then asks, "When can we deliver that for you?". The user responds with "9pm but actually can you make it a medium veggie". The system then provides a confirmation: "Ok so we are getting you a Medium Veggie at 21:00. This will be on our regular base.".

Slotting entities out of order

- If Out of Order Extraction is set, composite bag will resolve if it finds any entities of that type in any user input within resolveEntities
 - Would not work for string (as every input could be a string)

Entity Extraction

Out of Order Extraction

Extract With

Prompt for Value

I want a large meaty pizza please

ok lets get that order sorted

When can we deliver that for you?

9pm but actually can you make it a medium veggie

Ok so we are getting you a Medium Veggie at 21:00. This will be on our regular base.

Slotting entities out of order

- Confirming to the user that the entity has changed

```
resolveEntities:  
  component: "System.ResolveEntities"  
  properties:  
    variable: "pizza"  
    nlpResultVariable: "iResult"  
    maxPrompts: 3  
    cancelPolicy: "immediate"  
    headerText: "<#list system.entityToResolve.value.updatedEntities>I have updated <#items as  
ent>${ent.description}<#sep> and </#items>."  
  transitions:  
    actions:  
      cancel: "maxError"  
      next: "setPizzaDough"
```

Slotting entities out of order

```
headerText: "<#list system.entityToResolve.value.updatedEntities>I have updated <#items as ent>${ent.description}<#sep> and </#items>. </#list>"
```

I want a large meaty pizza please

ok lets get that order sorted

When can we deliver that for you?

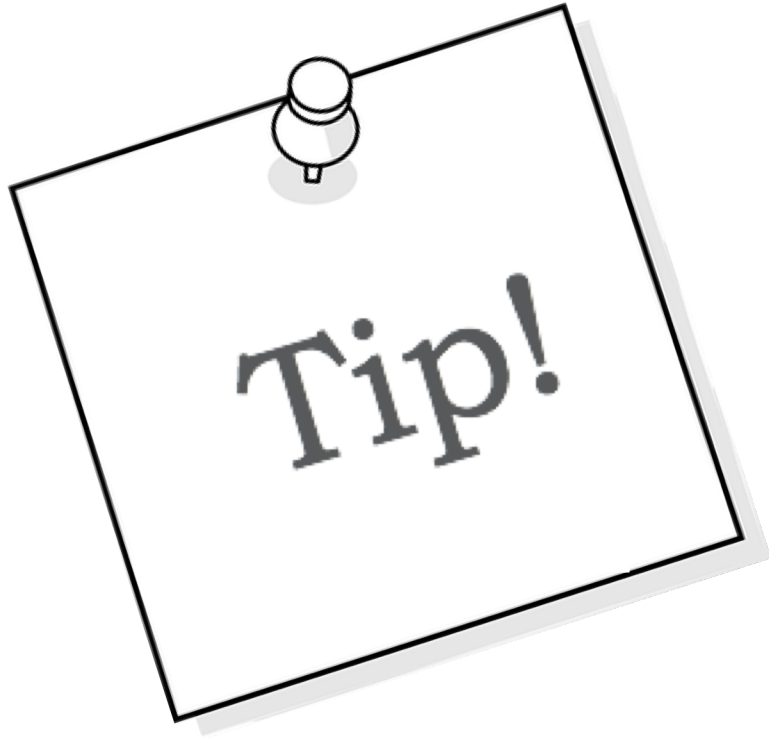
actually can you make it a medium

I have updated size of the pizza.

When can we deliver that for you?

9pm

Ok so we are getting you a Medium Meaty at 21:00. This will be on our regular base.



Going forward,
composite bag is your
primary “go to” for
entity resolution



Oracle Digital Assistant Hands-On

TBD

Integrated Cloud

Applications & Platform Services

ORACLE®