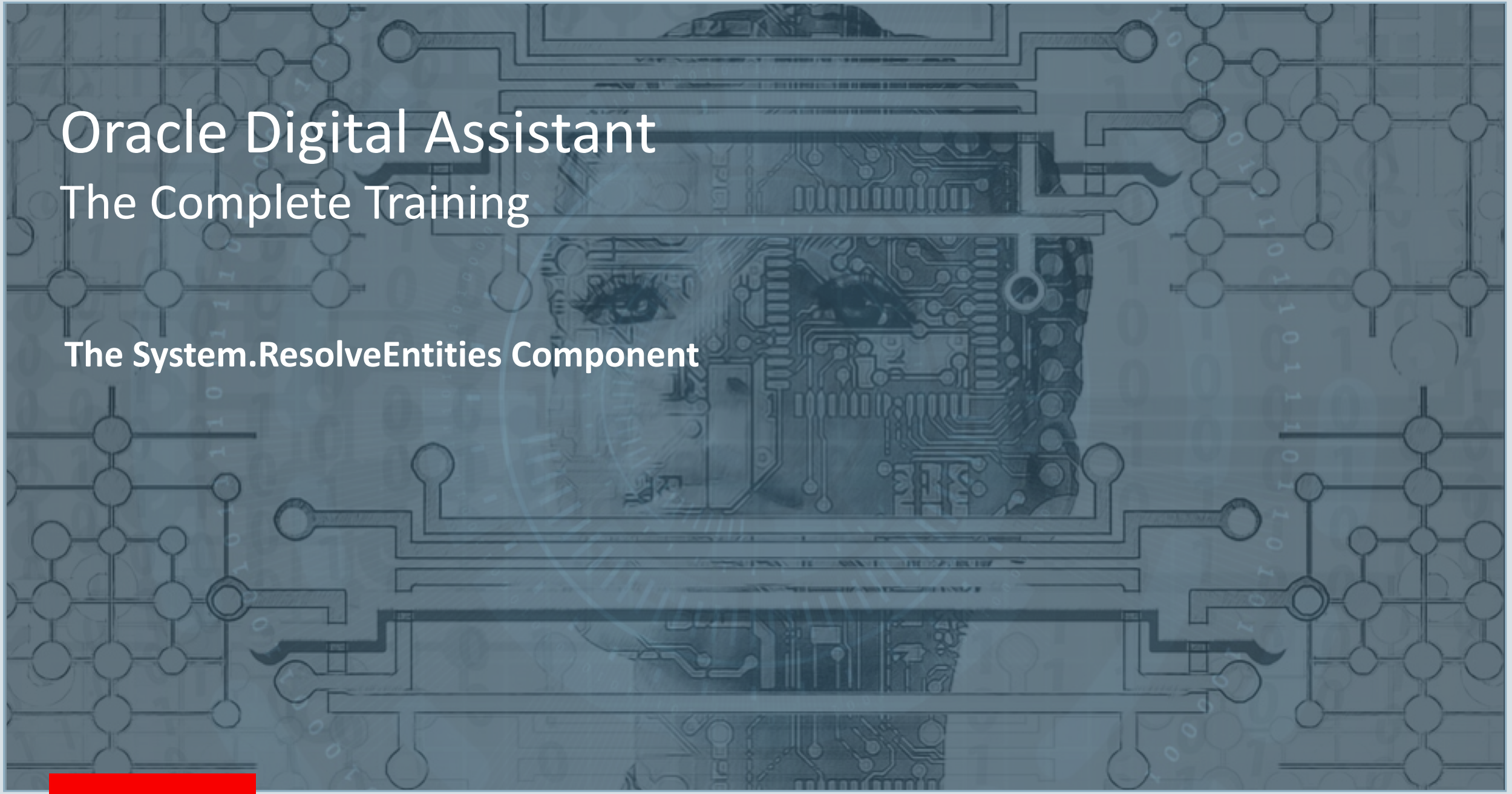

ORACLE®

Oracle Digital Assistant

The Complete Training

The System.ResolveEntities Component



ORACLE

Image courtesy of pixabay.com

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. Entity derived conversations
2. Component overview
3. Use with composite bag entities

Topic Agenda

1. Entity derived conversations
2. Component overview
3. Use with composite bag entities

Dialog flow is the conversation script that is followed by a skill in a user interaction. However, the **best dialog flow is no dialog flow**.

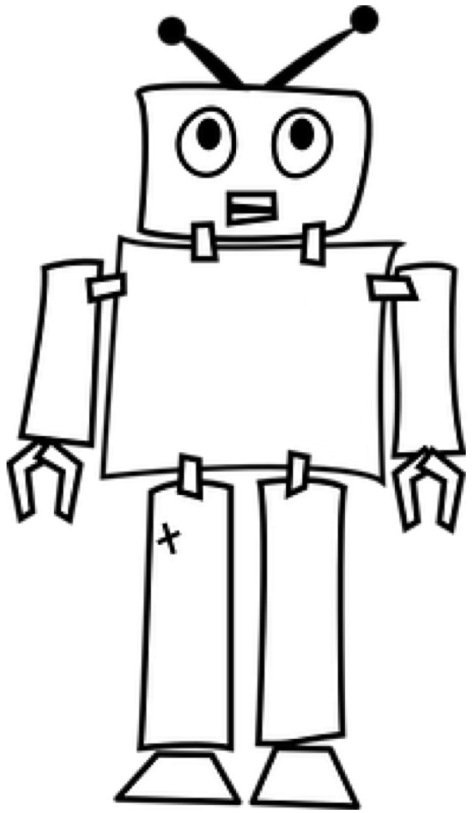


Image courtesy of pixabay.com

What wrong with dialog flows

- Users are not good at giving a single answer to a question
  - In human-to-human interaction it is natural to overload answers with information
    - Bot: "what pizza type do you like?"
    - User: "a large salami with extra cheese"
- 'story telling' vs. 'data driven'
  - Natural conversation design is chatty
  - Skills only need data input to complete a task
- Violates the DRY principle (don't repeat yourself)
  - Bot response configured on component
    - Prompt, error message, validation, range size etc.
  - No reuse of settings if configuration is on the component

Entity driven bot conversations

- Reduce the amount of dialog flow steps to write at design time
- Dynamically generate UI at runtime
  - Bot UI rendered based on entity type
    - Simple entities have a single user prompt
    - Composite bag entities may prompt users multiple times
  - All configurations and behaviors are defined on the entity
    - Prompts, error message, validation rules, range size
    - Entity extraction, out-of-order message handling (composite bag entity only)
- Oracle Digital Assistant promotes entity derived conversations
- Require use of System.ResolveEntities and System.CommonResponse components

Entity driven conversation **delegates common component configurations to the entity** level, which is a much better model for reuse.

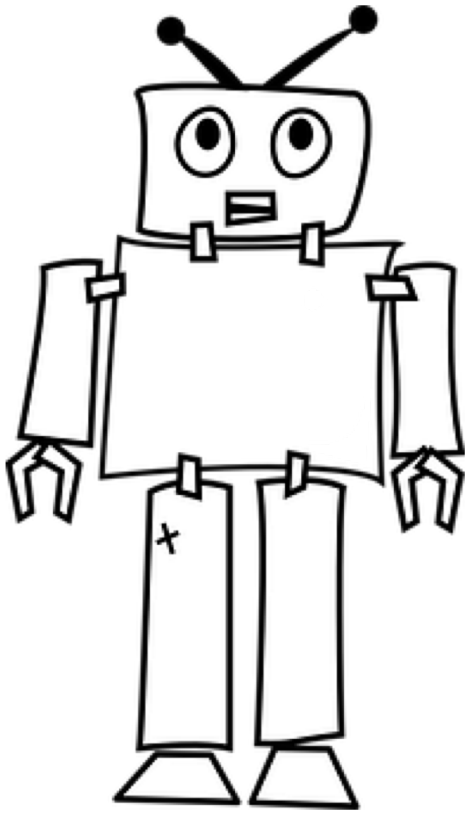


Image courtesy of pixabay.com

Topic Agenda

1. Entity derived conversations
2. Component overview
3. Use with composite bag entities

You can use the System.ResolveEntities component with system, custom and **composite bag entities**. It generates input fields (prompts) and value lists.

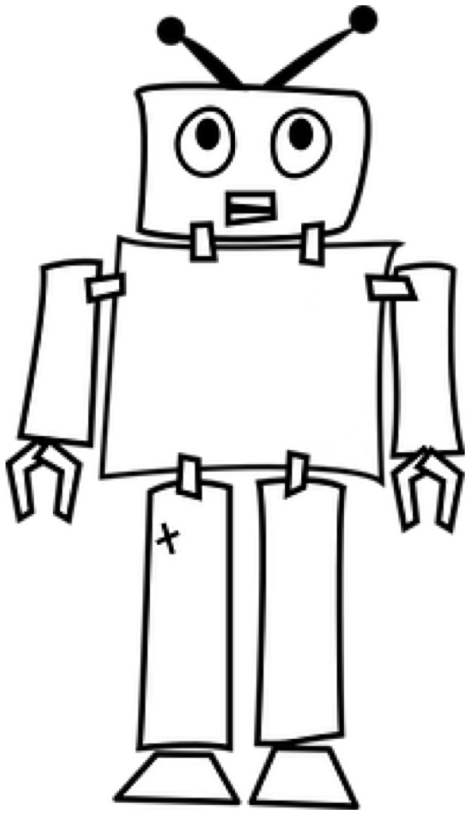
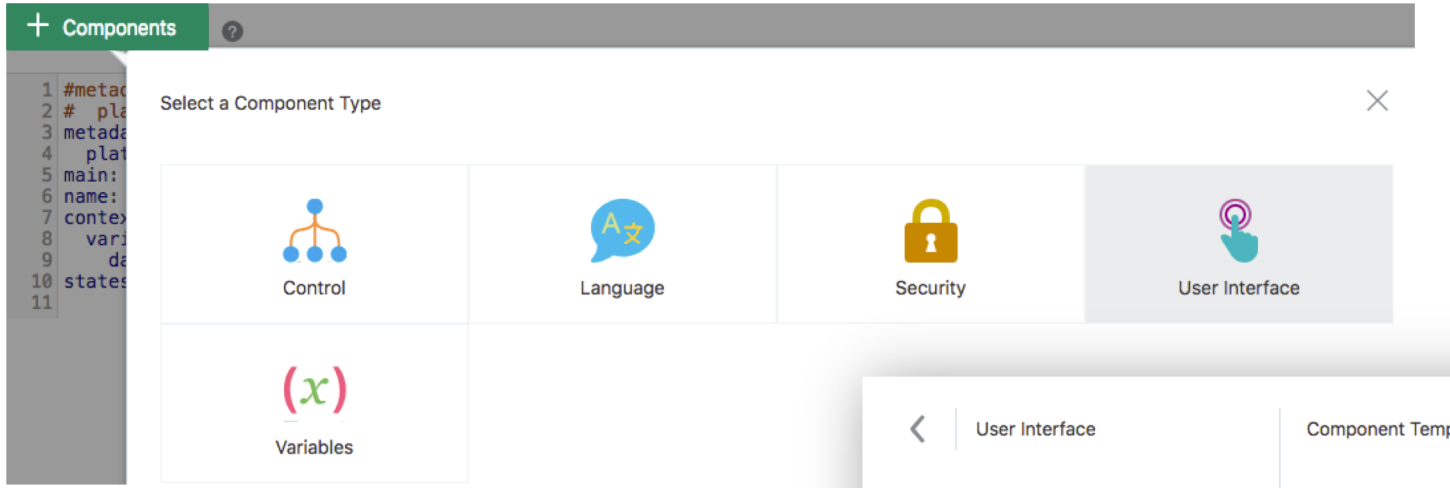
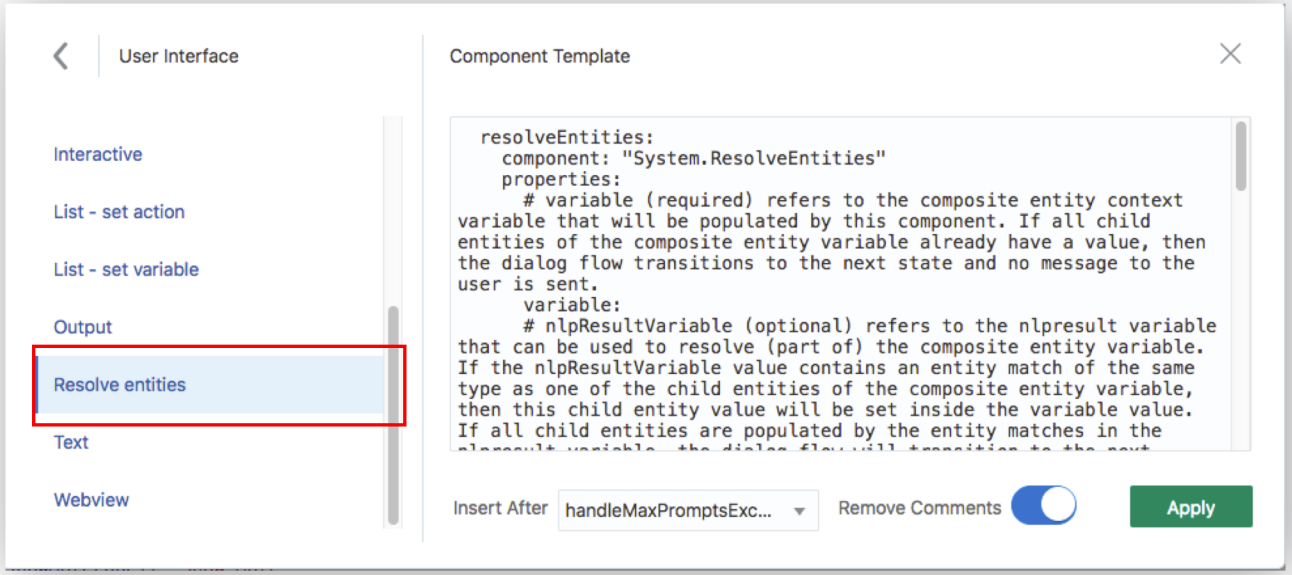


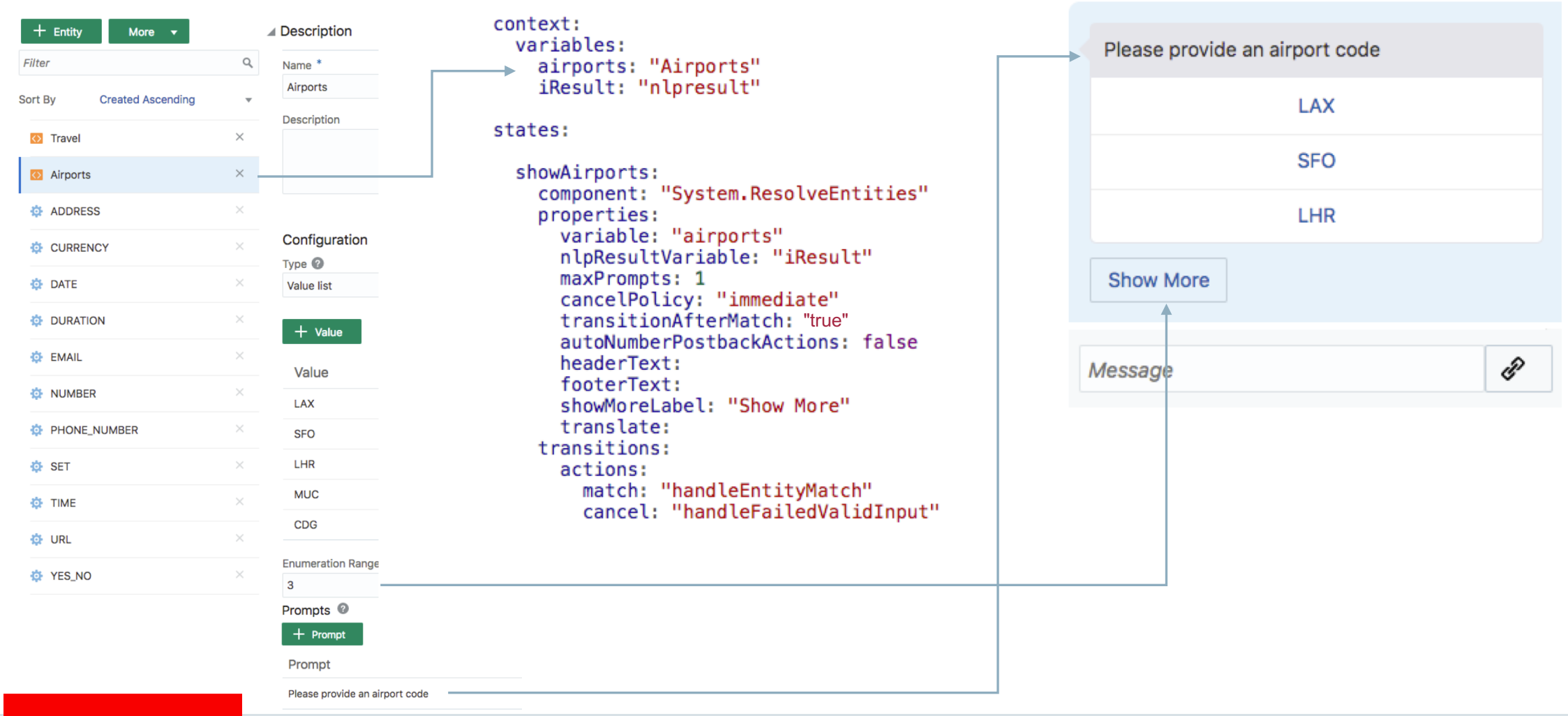
Image courtesy of pixabay.com

Building System.ResolveEntities from component template





System.ResolveEntities component with custom entity



Topic Agenda

1. Entity derived conversations
2. Component overview
3. Use with composite bag entities

You need to train **the skill bot model** before using System.ResolveEntities with composite bag entities.

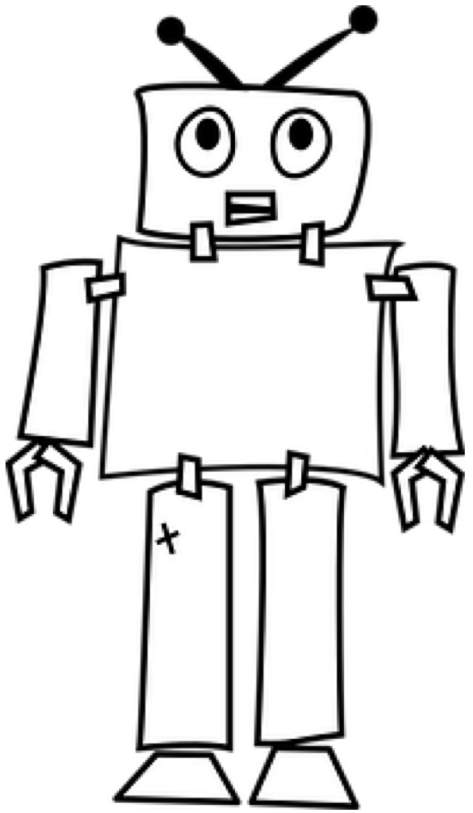
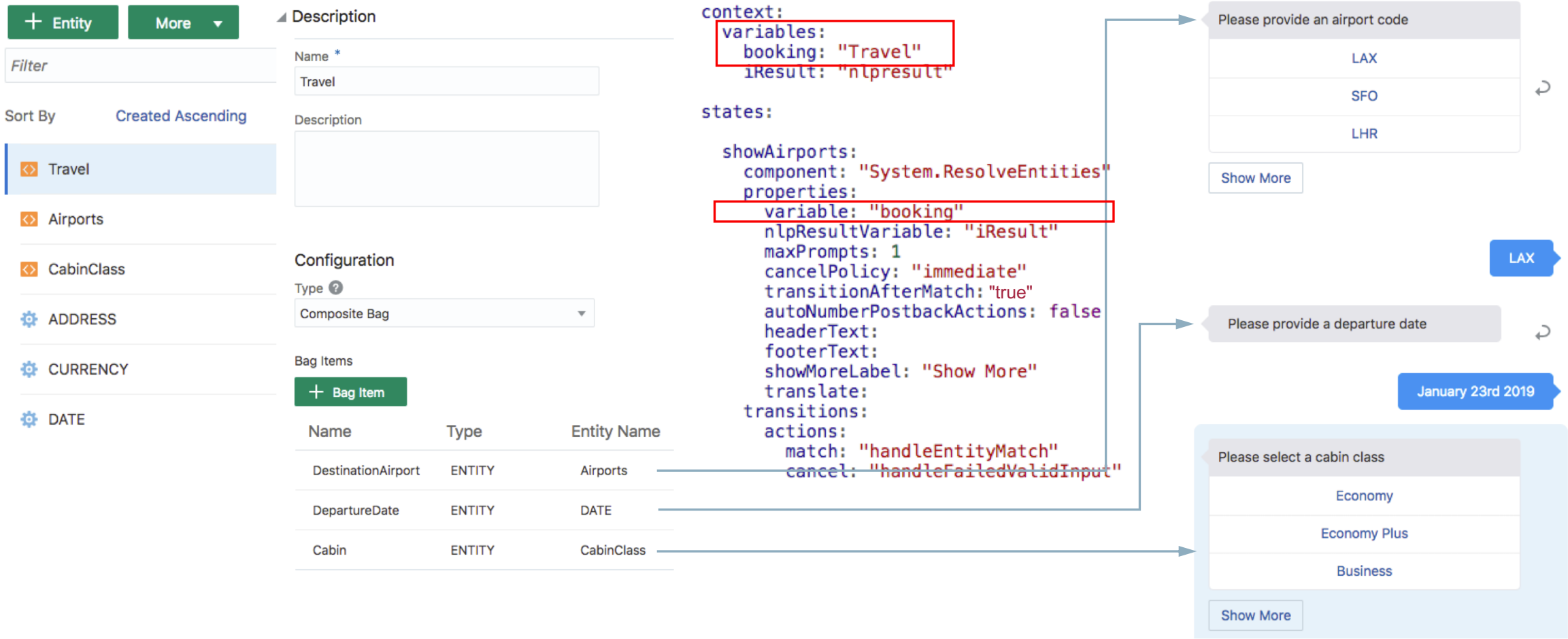


Image courtesy of pixabay.com

System.ResolveEntities with composite bag entity



Okay. We need to talk. What if you want to **perform additional validation or** just need to **invoke a custom component in response to a matched entity**?

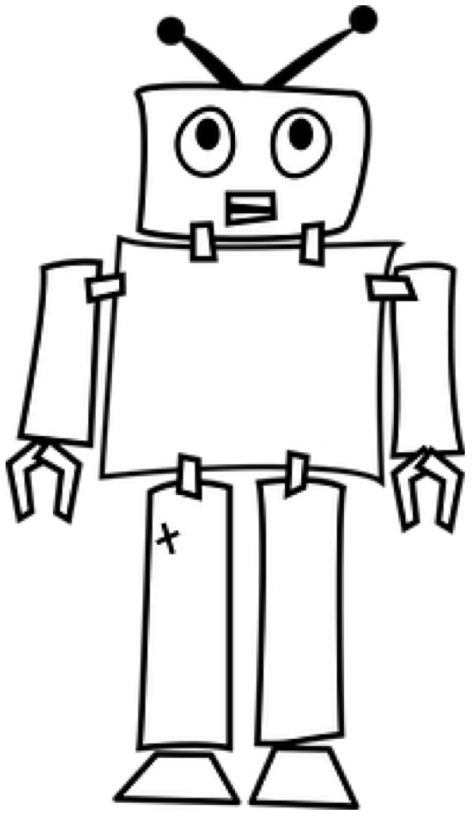
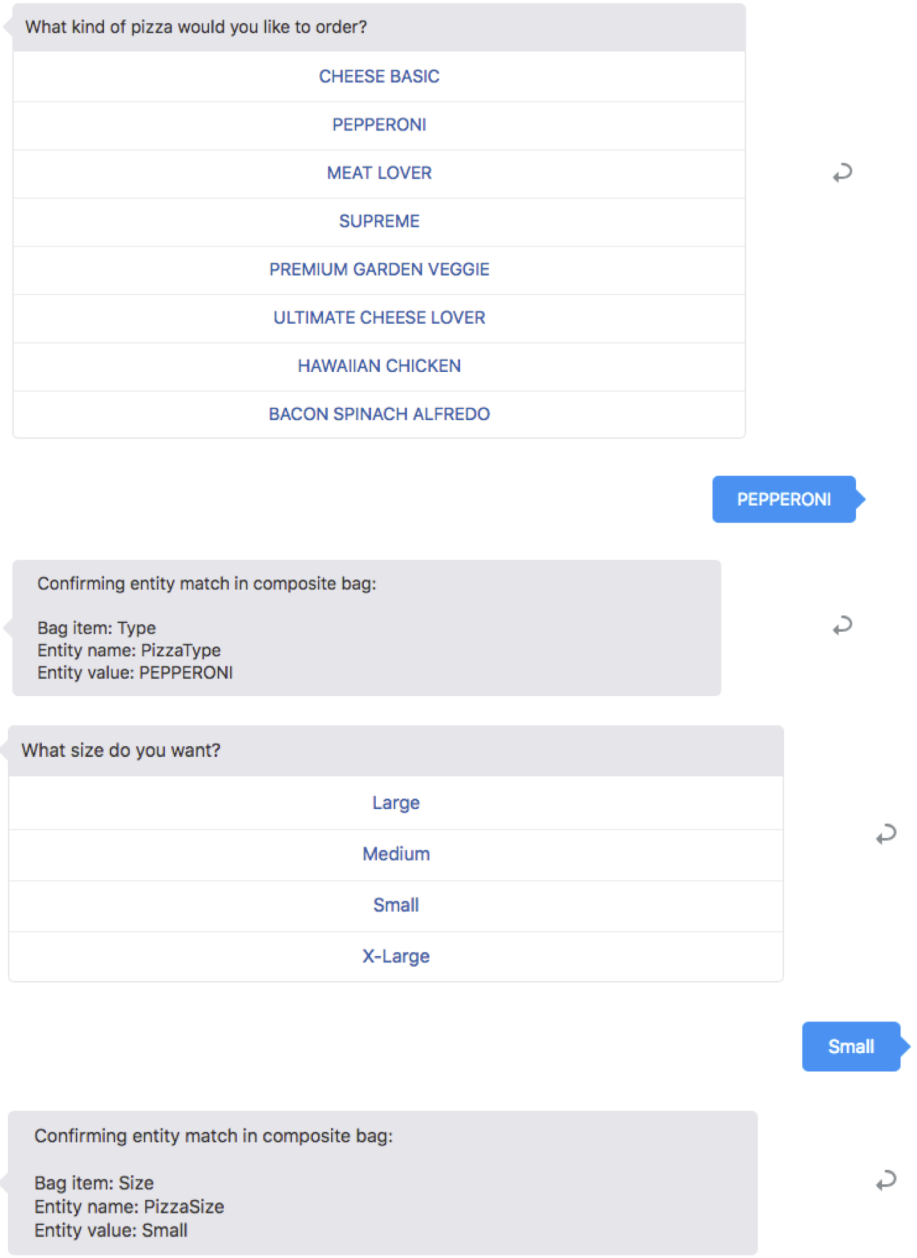


Image courtesy of pixabay.com

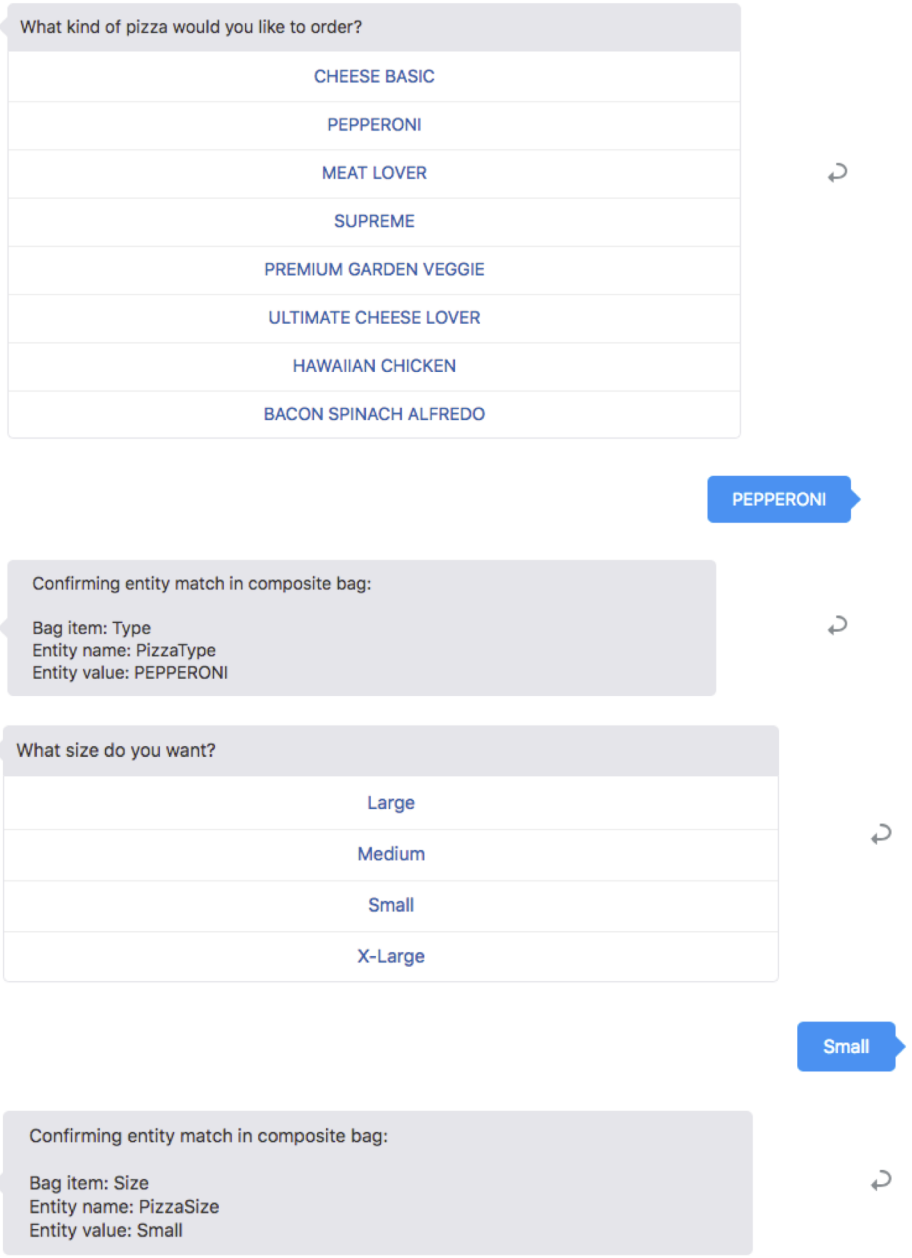
'transitionAfterMatch' property

- If set to "true", component transitions to dialog flow state upon entity match
  - String "true", not the boolean true
  - Bot designers can call custom component or just acknowledge the value match
- 'match' action transition called for each entity match



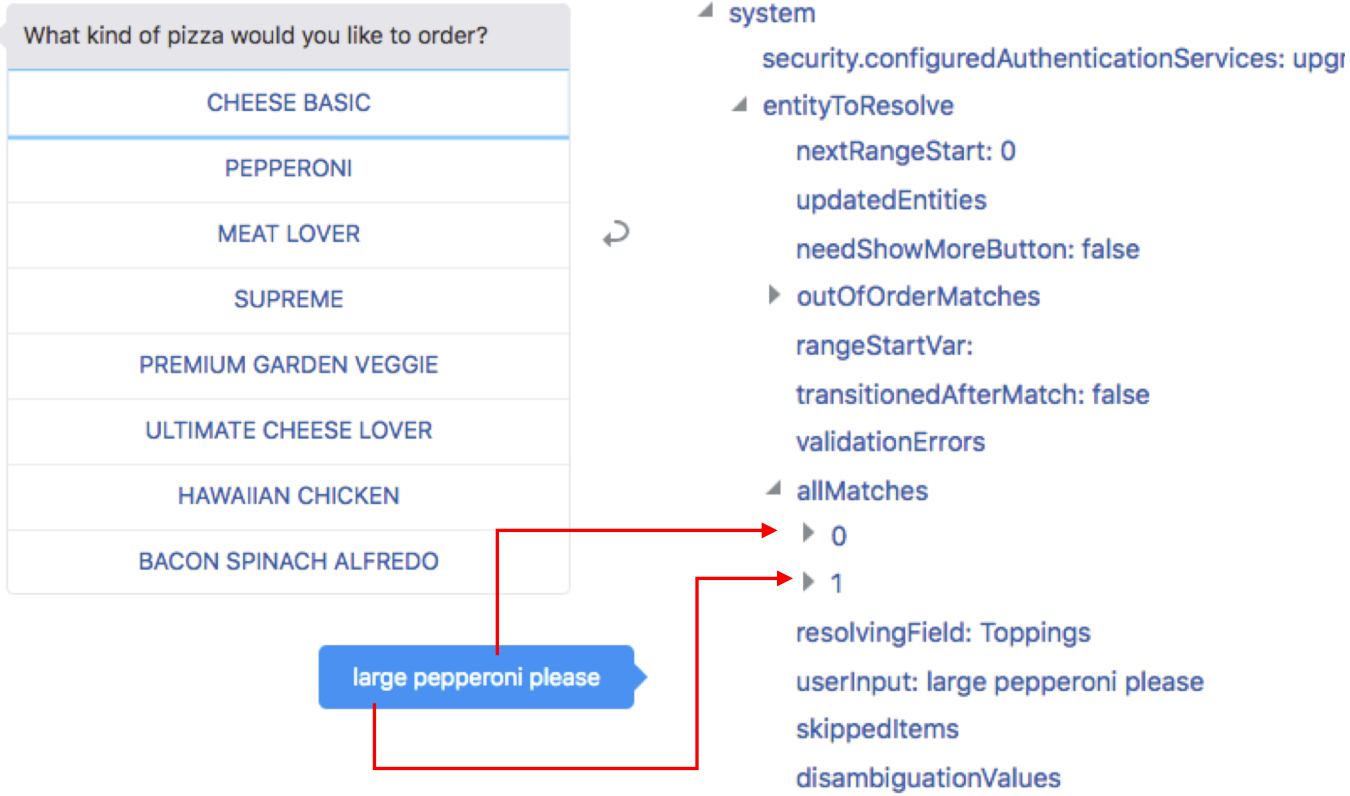
Navigating on entity match

```
orderPizza:
  component: "System.ResolveEntities"
  properties:
    variable: "pizza"
    nlpResultVariable: "iResult"
    maxPrompts: 2
    transitionAfterMatch: "true"
    showMoreLabel: "More"
    translate:
  transitions:
    next: "confirmation"
    actions:
      match: "handleAfterMatch"
      cancel: "cancelorder"

handleAfterMatch:
  component: "System.Output"
  properties:
    text: |-
          Confirming entity match in composite bag:

          Bag item: ${system.entityToResolve.value.resolvingField}
          Entity name: ${system.entityToResolve.value.allMatches[0].entityName}
          Entity value: ${pizza.value[system.entityToResolve.value.resolvingField]}

    keepTurn: true
  transitions:
    #resume orderPizza
    next: "orderPizza"
```



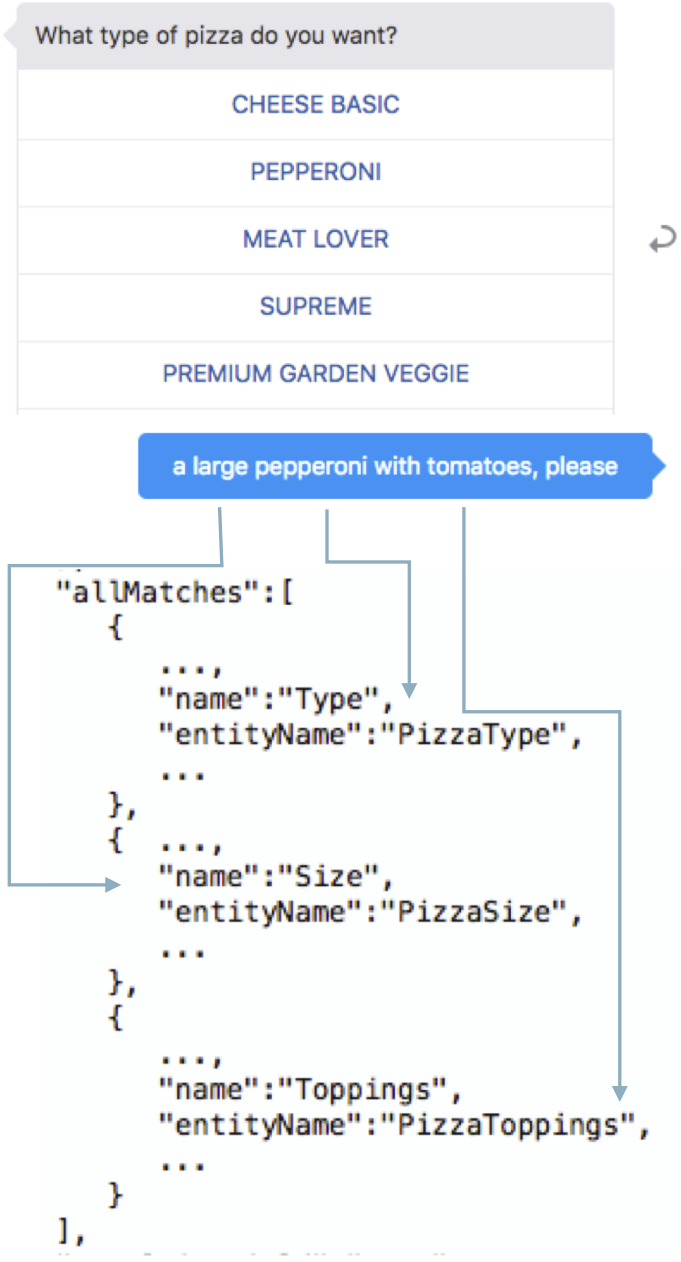
Behavior when multiple entities are getting resolved

- User input may lead to multiple entity matches
  - Out-of-order extraction
- 'match' transition is called only once
- Access matched bag items
  - ${system.entityToResolve.value.allMatches[n].entityName}
  - ${system.entityToResolve.value.allMatches[n].name}



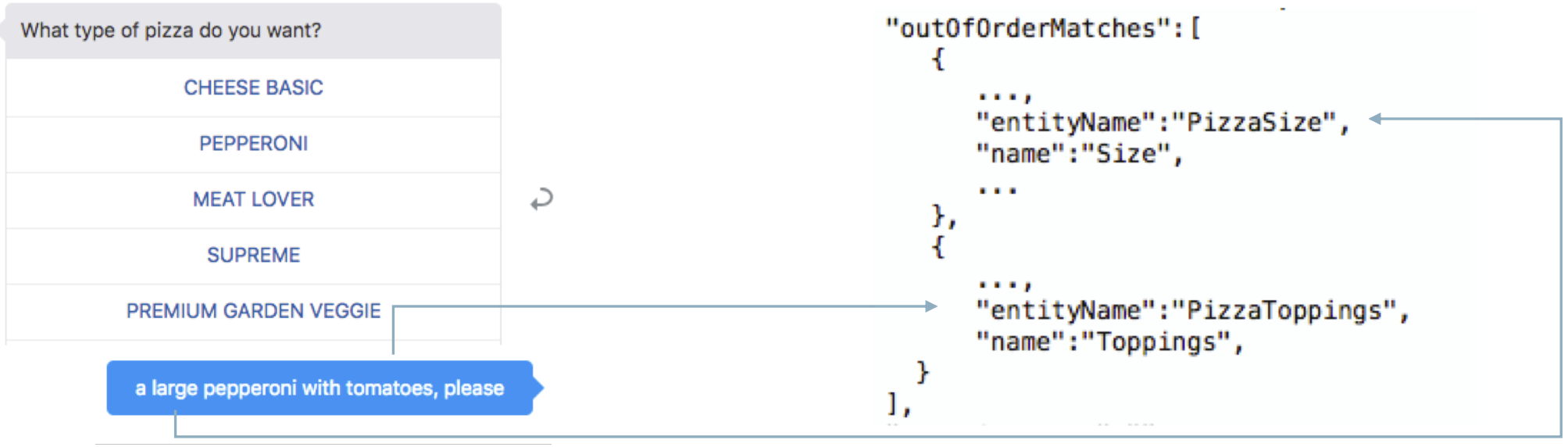
Accessing matched entities

- All updated entities from a user input
  - E.g. User provides more information than prompted for
    - Bot: "what pizza type do you like?"
    - User: "a large salami with tomatoes"
  - Updated entities: PizzaSize, PizzaType, PizzaToppings
- Expression to access matched entities
  - ${system.entityToResolve.value.allMatches?size}
  - ${system.entityToResolve.value.allMatches[n].entityName}
  - ${system.entityToResolve.value.allMatches[n].name}



Accessing out-of-order entity matches

- Updated entities from a user input for which there was no prompt
  - ${system.entityToResolve.value.outOfOrderMatches[n].entityName}
  - ${system.entityToResolve.value. outOfOrderMatches[n].name}
  - ${system.entityToResolve.value.outOfOrderMatches?has_content?then(...,...)}



Integrated Cloud

Applications & Platform Services


ORACLE®