BITS Pilani
Pilani Campus

Design of Conversational Experiences

Dr. Bharathi R
Off Campus, CSIS, BITS-Pilani

Slides by Dr. Shreyas Rao
Associate Prof. (Off Campus), CSIS, BITS-Pilani

Lecture 3 Recap

1. **Basics of NLP and NLU**
2. **Bot Basics**

- Bot Anatomy
- Bot Purpose
- Bot Branding
- Personality

Assignment I (10%)

Task: Comparative analysis of Bot Builders.

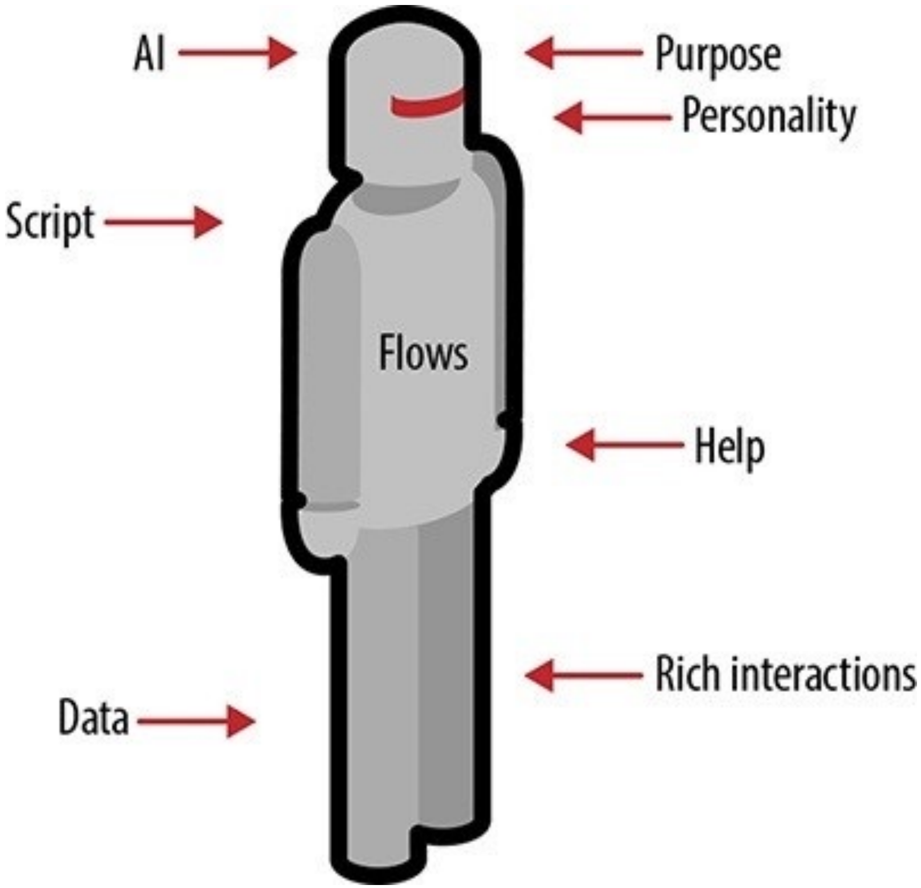
Sub-tasks:

1. Select one open source bot builder (Ex: Wit.ai) and one commercial bot builder (Ex: Amazon Lex).
2. Write a detailed report comparing the two selected bot builders, based on the following criteria:

a. Ease of Development (No code, drag and drop functionality etc.)
b. Features and capabilities.
c. Integration options with other tools and platforms.
d. Community support and documentation.
e. Real world use cases for the bot builder and its usage in the Industry.

- Individual Assignment
- Due date for submission: 20-Feburary-25

Bot Anatomy



BRANDING

BITS Pilani
Pilani Campus

Lecture No. 4

Agenda

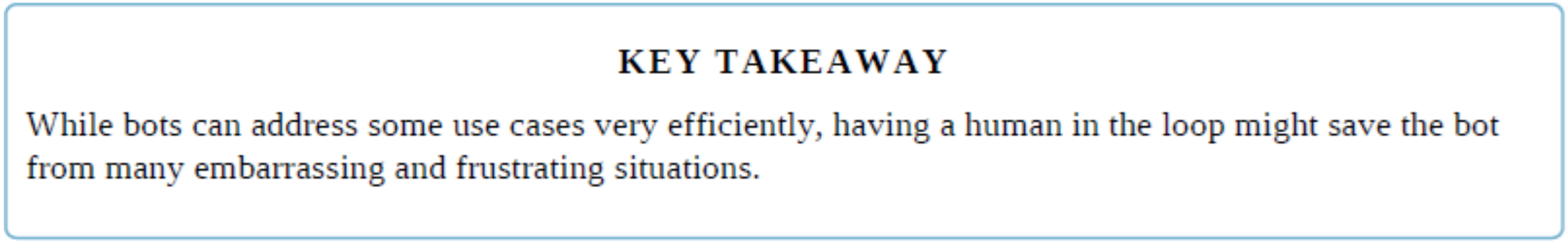
- Human Involvement (T1-chapter6)
- Designing Bot Conversations (T1- chapter 8)
- Demos / Hands-on exercises on designing bot conversations

References
T1: Chapter 6,8.

Human Intervention or Involvement

Human Intervention

- Bots require regular human supervision and intervention
- This is where Humans help bots



KEY TAKEAWAY

While bots can address some use cases very efficiently, having a human in the loop might save the bot from many embarrassing and frustrating situations.

Human Intervention

Identify scenarios where bots require Human Intervention?

1. Sales bot – financing services for a user may need human intervention.

2. IT support bot.

Enabling error

1. Bot → cannot understand the user's intent
2. It receives ambiguous i/p
3. user expresses frustration
4. conversation – unproductive way.

Human Intervention

1. Humans resolving ambiguity and providing response supervision

Ex: In banking websites or legal websites, before committing to user, certain transactions need human approval

Scenario 1: A human is talking with a sales bot about buying a car, and after a great conversation it is time for the bot to close the lead and offer a financing service to the user. A human sales rep is mandated by company process to manually go over the offer details and approve the loan.

Scenario 2: An IT support bot that gets a question with two possible answers. The bot then defers to a human IT professional (providing them with the two potential answers, preferably with a recommendation) and lets the IT professional pick the right answer out of the two.

Note that human supervision has a cost associated with it. Human responses are not instantaneous and will make your bot a little slower.

Human Intervention

2. Humans Enabling Error/Failure Escalation

Scenarios

- The bot does not know how to handle the user's intent or request.
- The bot does not understand the user input.
- The bot recognizes negative sentiment (for example, when the user is getting frustrated).
- The bot exposes a way for the user to ask for human assistance and that
- functionality is invoked. Ex: Mental health care or suicidal tendancies
- The conversation is taking too long, or is unproductive or circular (also known as the user getting lost in the conversation).

Human Intervention

3. Human Training Bots online / Humans improving Bot Conversation

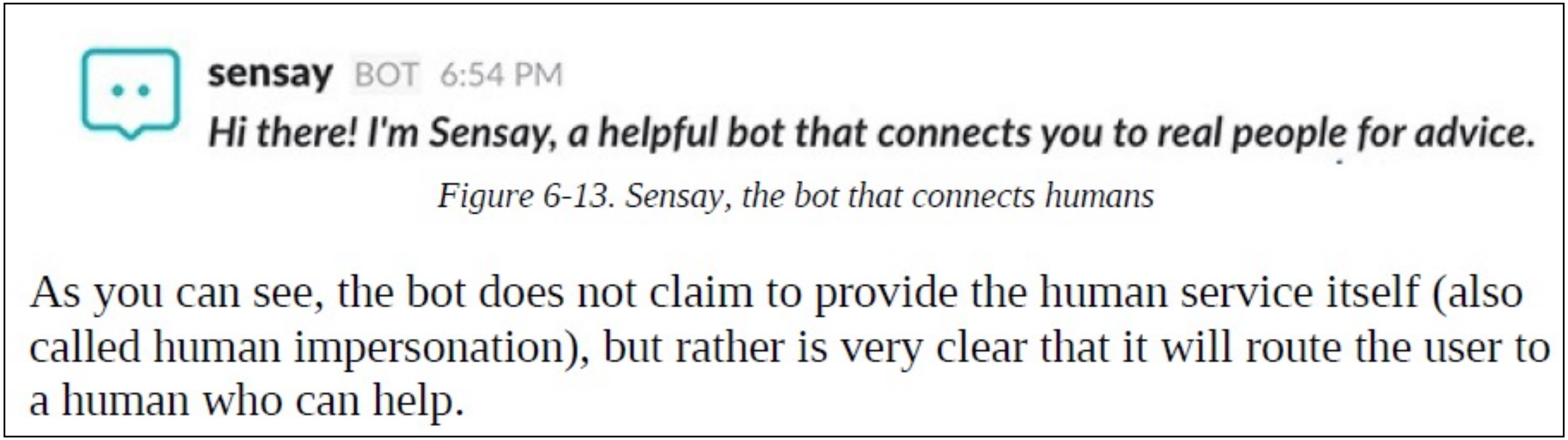
Ex: FAQs, Review the Conversational Logs, and train the bots to answer queries

One interesting real-life story regarding reading bot logs was with SmarterChild.
SmarterChild was one of the earliest bots, which ran on the AOL messaging app. The bot was a conversational mate for a teenage audience. Looking at the logs, the team discovered that some kids were confessing suicidal thoughts to the bot. At the time there was no hotline for such matters, and the team decided to direct the bot to not respond to such conversations. Talking with them now, they say it was one of the hardest design choices they had to make, and they only made it on the advice of their legal team. Today they would have directed these teens to the relevant hotline and reported the conversations.

Human Intervention

4. Bots as facilitators for Human Tasks

Some workflows and some tasks are better suited to humans, but bots can still play a facilitator role in the process



sensay BOT 6:54 PM

Hi there! I'm Sensay, a helpful bot that connects you to real people for advice.

Figure 6-13. Sensay, the bot that connects humans

As you can see, the bot does not claim to provide the human service itself (also called human impersonation), but rather is very clear that it will route the user to a human who can help.

innovate achieve lead

BITS Pilani
Pilani Campus

Designing Bot Conversations!

Steps in Designing Bot Conversations

1. Onboarding
2. Functionality Scripting
3. Task-led and topic-led conversations
    - Divergent flows and course correction
    - Entity extraction
    - Intent mapping
    - Conversational controls
    - Stories/ Flows
4. Decoration
5. Rich Interactions
6. Context and Memory
7. Bot Discovery and Installation
8. Engagement Methods

**

BITS Pilani
Pilani Campus

1.Onboarding

1.Onboarding

- Onboarding is the first interaction users see from the bot
- It could be a message that the bot sends to the installing user or a general message to a team
- It sets the first impression and tackles a set of tasks that can best be accomplished at the start of the conversation.

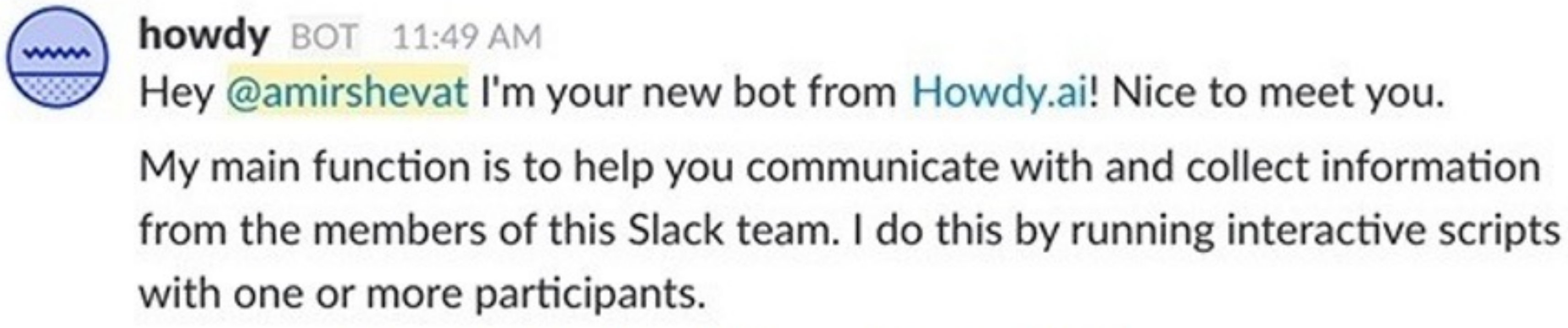
Involves:

a. Declaring the purpose
b. Configuration
c. Incite Action

**

1.a.Declaring the Purpose

- During onboarding the bot declares its purpose in the context of the conversation, making it transparent to the user or the team

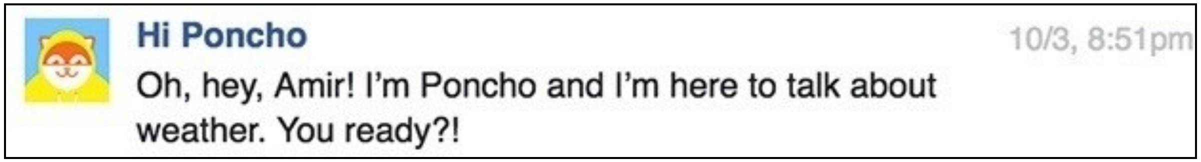


howdy BOT 11:49 AM

Hey @amirshevat I'm your new bot from Howdy.ai! Nice to meet you.

My main function is to help you communicate with and collect information from the members of this Slack team. I do this by running interactive scripts with one or more participants.

Figure 8-1. The Howdy bot introducing itself to the user



Hi Poncho 10/3, 8:51pm

Oh, hey, Amir! I'm Poncho and I'm here to talk about weather. You ready?!

Declaring the Purpose

- DoCE - Handbot
- Declaring the purpose of the bot, and additionally mentioning bot features

"Hi! I’m Chatbot, created by Dr. Shreyas to help you with the 'Design of Conversational Experiences' course. You can use me to:"
•📄 **Access course handouts**
•🔍 **Search for specific topics**
•💬 **Get answers to frequently asked questions**

1.b Configuration

- Bot will ask the user to supply information that is important to the core functionality of the bot

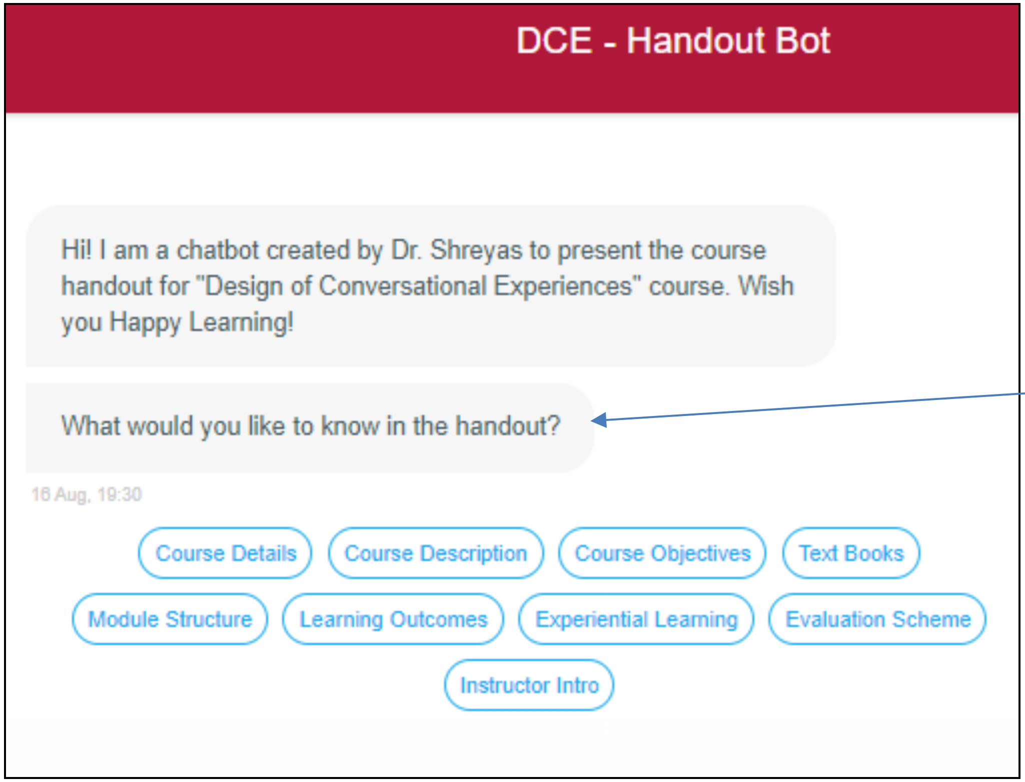
- It could be grant additional permissions to third-party services

OR

- Capture the responder's name, email, and other details

1.c Incite Action

- Call to action from the user



DCE - Handout Bot

Hi! I am a chatbot created by Dr. Shreyas to present the course handout for "Design of Conversational Experiences" course. Wish you Happy Learning!

What would you like to know in the handout?

16 Aug, 19:30

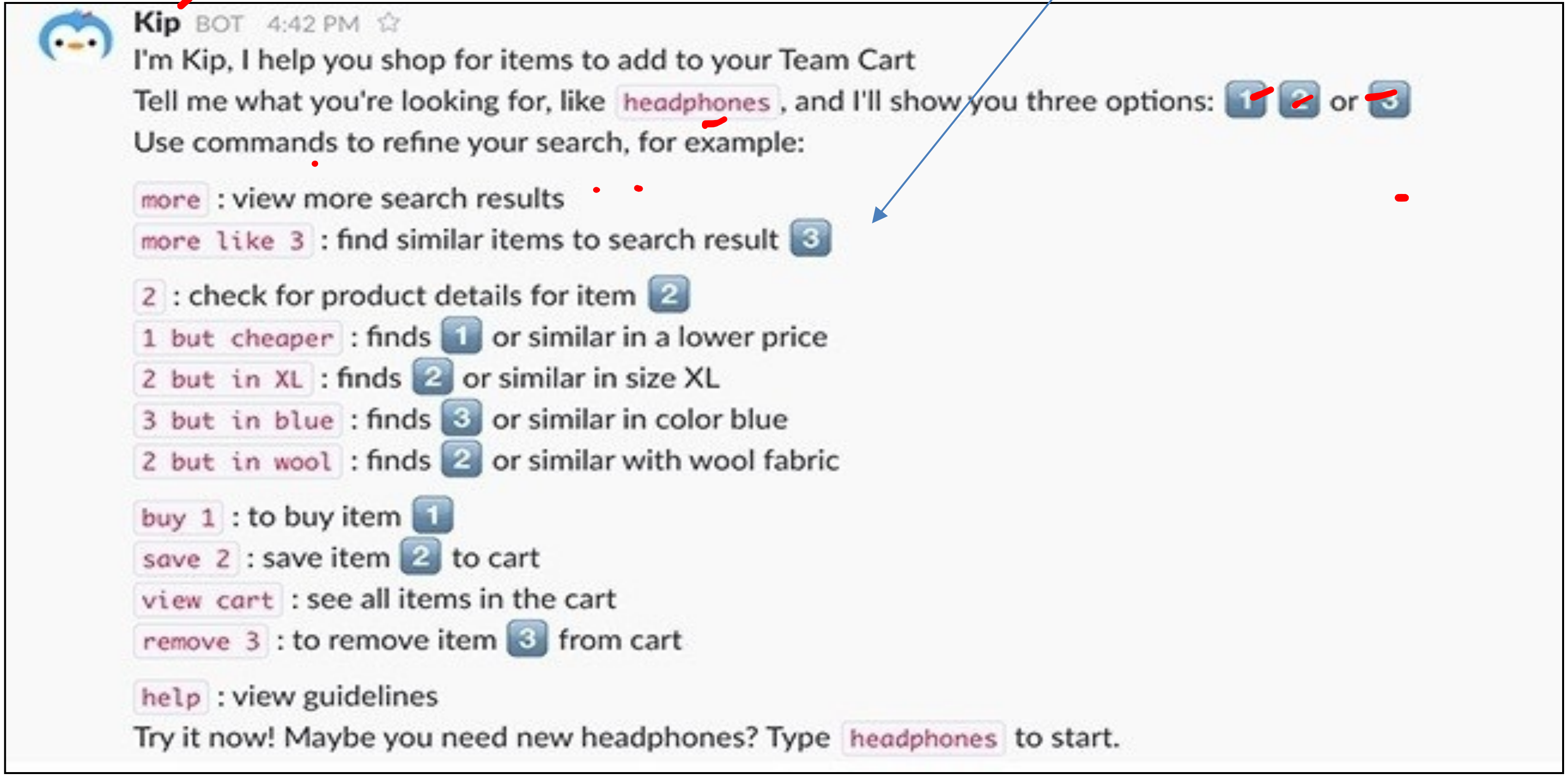
Call to Action

**

Incite Action

- Call to action from the user

Call to Action



Kip BOT 4:42 PM ☆
I'm Kip, I help you shop for items to add to your Team Cart
Tell me what you're looking for, like `headphones`, and I'll show you three options: 1 2 or 3
Use commands to refine your search, for example:

`more` : view more search results
`more like 3` : find similar items to search result 3

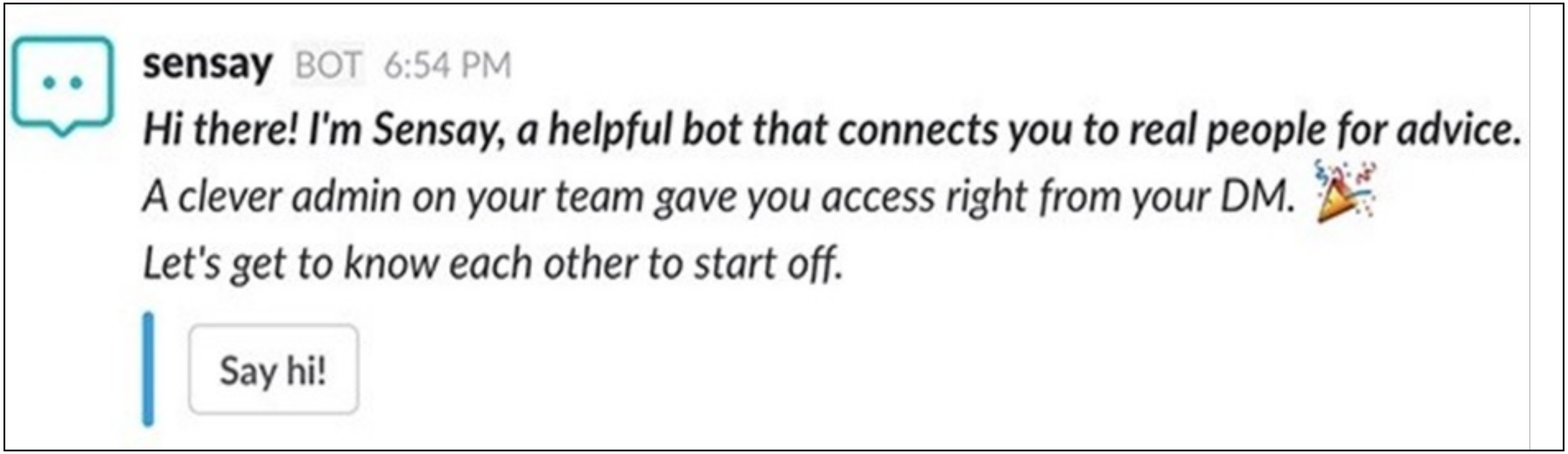
`2` : check for product details for item 2
`1 but cheaper` : finds 1 or similar in a lower price
`2 but in XL` : finds 2 or similar in size XL
`3 but in blue` : finds 3 or similar in color blue
`2 but in wool` : finds 2 or similar with wool fabric

`buy 1` : to buy item 1
`save 2` : save item 2 to cart
`view cart` : see all items in the cart
`remove 3` : to remove item 3 from cart

`help` : view guidelines
Try it now! Maybe you need new headphones? Type `headphones` to start.

Onboarding in Team Environment

- One of the challenges with onboarding a bot to a team is that the team members might not be aware that the bot was invited to the messaging platform, and might not know why they are being engaged by the bot.



sensay BOT 6:54 PM

Hi there! I'm Sensay, a helpful bot that connects you to real people for advice.

A clever admin on your team gave you access right from your DM. 🎉

Let's get to know each other to start off.

Good way to Introduce the Bot to the team!

**

2.Functionality Scripting

Functionality Scripting

- There are two ways to design conversations

1. Task-led conversation – Target is to accomplish a task
2. Topic-led conversation – Discuss information and exchange ideas around a specific subject(s)

2.1. Task-lead Conversation

Scenario: Planning a team meeting.
Conversation:
Person A: "We need to schedule the team meeting for next week. When is everyone available?"
Person B: "I'm available Monday at 10 AM or Wednesday after 2 PM."
Person C: "Wednesday at 2 PM works for me."
Person A: "Great, let's set the meeting for Wednesday at 2 PM. I'll send out the invite."

Scenario: Resolving a technical issue.
Conversation:
Person A: "The server is down, and we need to get it back online. Have you checked the logs?"
Person B: "Yes, the logs show a database connection error."
Person A: "Okay, let's restart the database service and see if that resolves it."
Person B: "Restarting now… It's back up! The site is live again."
Person A: "Perfect, let's monitor it for any further issues."

2.2 Topic-lead Conversation

Scenario: Discussing a new technology.
Conversation:
Person A: "Have you heard about the latest advancements in quantum computing?"
Person B: "Yes, it's fascinating! The potential for exponentially faster processing could revolutionize fields like cryptography."
Person A: "Absolutely, and I'm also curious about its applications in drug discovery. It could really speed up the process."
Person B: "That's true. It will be interesting to see how this technology evolves in the next few years."

Scenario: Exploring a business strategy.
Conversation:
Person A: "What do you think about adopting a subscription model for our product?"
Person B: "It could be a good move. Subscription models provide steady revenue and help build customer loyalty."
Person A: "I agree. Plus, it allows us to offer continuous updates and new features, which could enhance user satisfaction."
Person B: "Definitely. We'd need to assess the pricing strategy carefully, though, to ensure it's competitive."

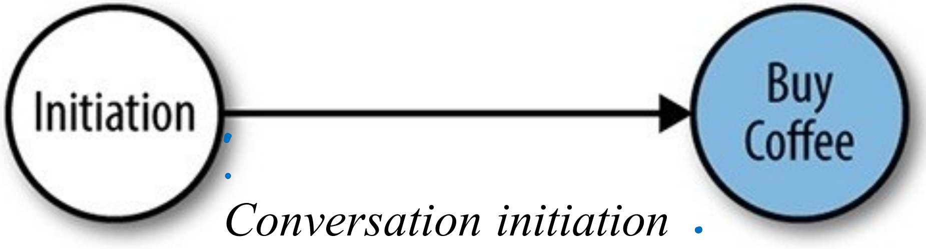
Designing Task-lead Bot Conversations

- The key for designing this type of conversation is **finding the optimal set of conversational interactions to complete a specific task**

Figure 8-12. The task to accomplish

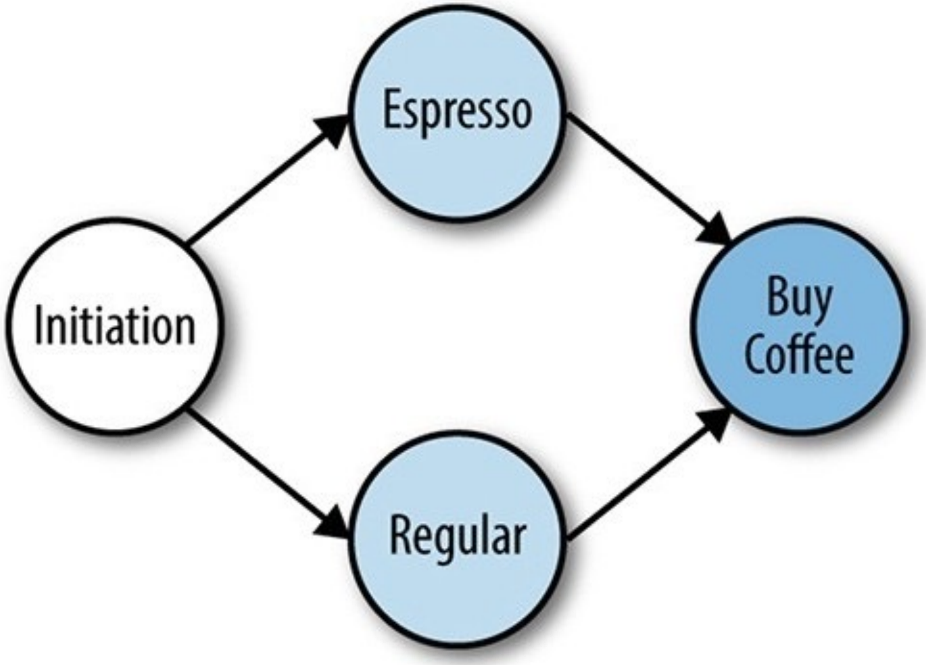
**

Designing a Coffee Bot



Conversation initiation .

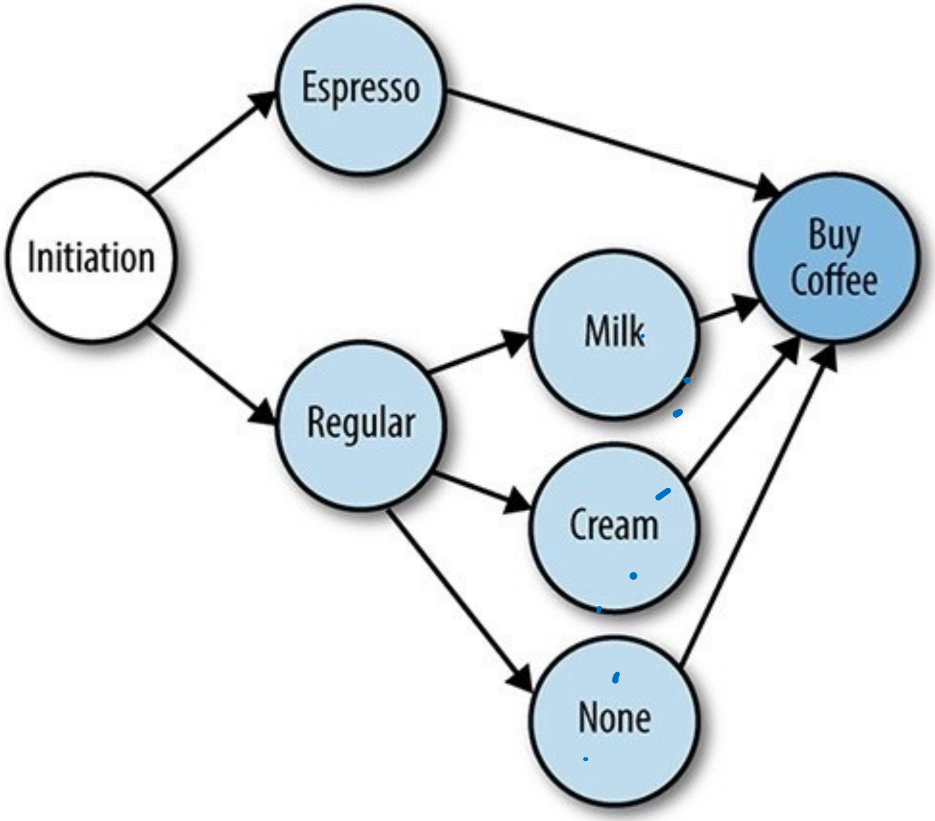
Designing a Coffee Bot



Initial Paths

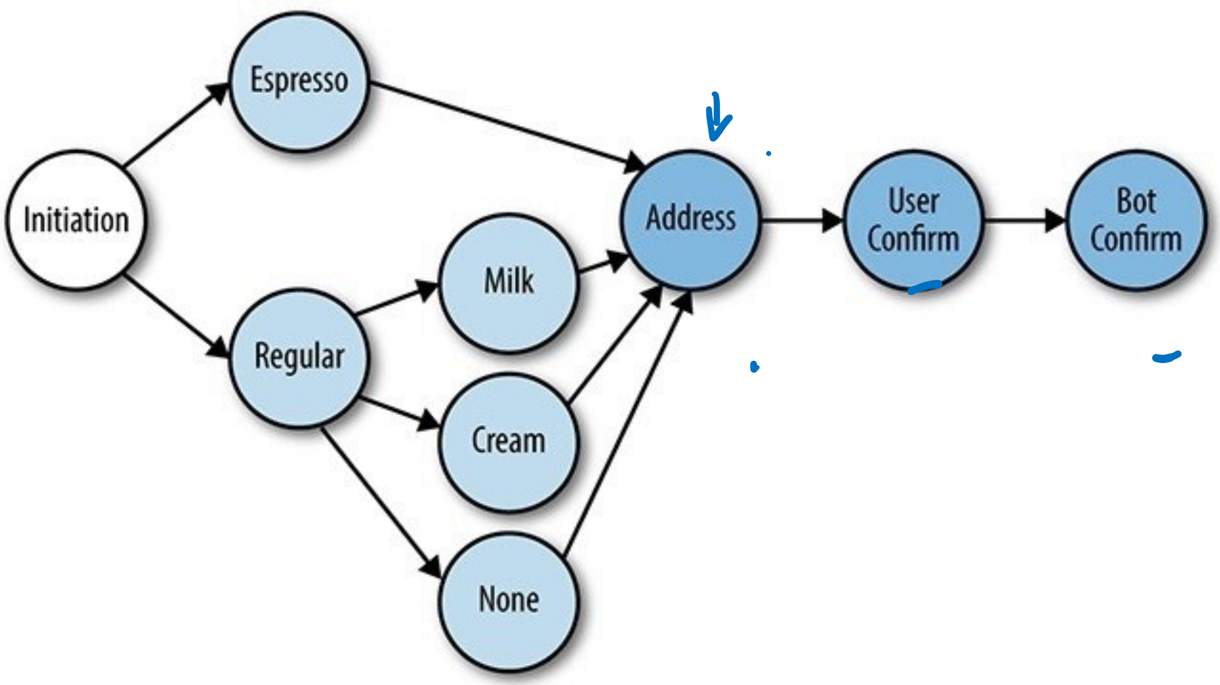
As you can see, one of these pathways might require milk

Designing a Coffee Bot



Additional Paths

Designing a Coffee Bot



Adding the Address and Confirmation Stages

Designing a Coffee Bot

Happy Path Flow

User: Hello @coffeebot

Coffeebot: What would you like today? We have regular coffee and espresso.

User: I will have an espresso

Coffeebot: Where would you like the coffee to be delivered?

User: 155 5th SF USA

Coffeebot: Please confirm - 1 espresso coffee to be delivered to 155 5th SF USA.

User: Confirmed

Coffeebot: Coffee en route, ETA 15 min.

Divergent Flows and Course Corrections

Happy Path Flow

User: Hello @coffeebot

Coffeebot: What would you like today? We have regular coffee and espresso.

User: Wait, no! I want a cappuccino!

The user has done something that the bot did not expect. They have diverged from the happy flow, and requested something that the bot was not prepared for.

There are two basic ways to handle this:

1. Course Correction
2. Human Intervention

Course Correction

In bot design, "**course correction**" refers to the process of guiding a conversation back on track when the interaction with the bot starts to deviate from its intended path.

To guide the user back, the bot needs to understand the “order” properly!

What was ordered -> Cappuccino or Espresso ??

Design an ENTITY to capture the User’s input

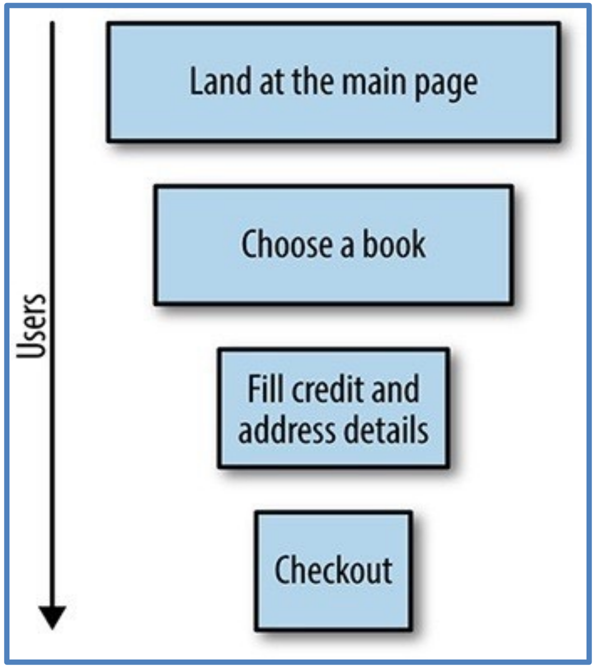
Entity

- In bot design, an entity refers to a **key piece of information or data** that the bot needs to recognize and extract from the user's input to fulfill a particular task or respond appropriately.
- Entities represent **specific values** within a conversation that are relevant to the bot's understanding and processing of user requests.

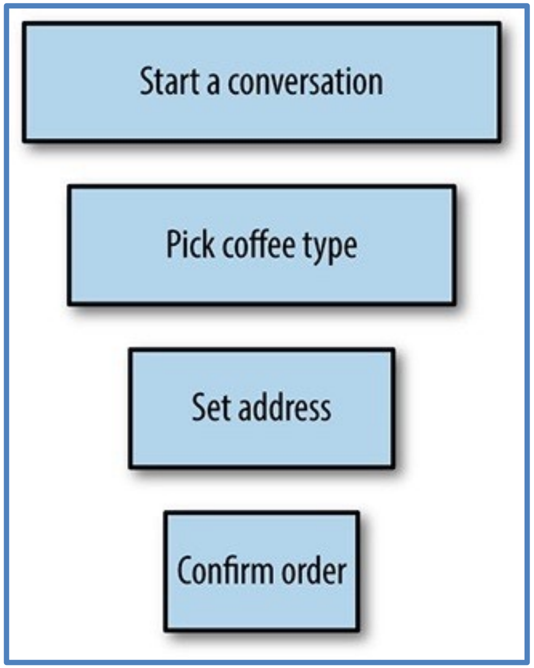
Examples for entitites - name, age, email, mobile number, type of coffee etc.

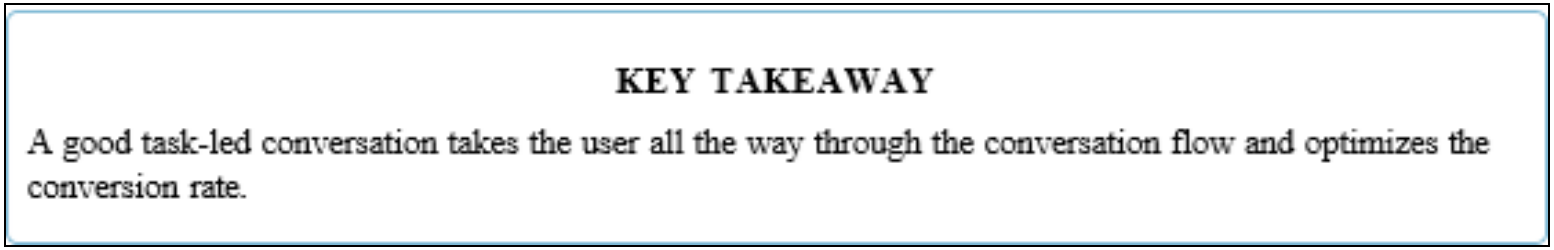
Conversation Funnel

Website Funnel



Coffee Bot Funnel





KEY TAKEAWAY

A good task-led conversation takes the user all the way through the conversation flow and optimizes the conversion rate.

AIML (Artificial Intelligence Markup Language)

Ref - AIMLhttps://kcir.pwr.edu.pl/~witold/ai/aie_aiml_d.pdf

AIML

- AIML is an XML-based description language designed for creating natural language software agents
- **AIML** stands for Artificial Intelligence Modelling Language.

We will use AIML for:

1. Designing Basic Bot Conversations
2. Onboarding
3. Handling Error Flows
4. Capture and set Entities within a conversation

Ref - AIML https://kcir.pwr.edu.pl/~witold/ai/aie_aiml_d.pdf

AIML Basics

- AIML is a rule-based language
- The rules are small entities consisting of two parts: **condition and action**.
- The system works by selecting one rule which has its condition satisfied, and then executing the action of the selected rule.

The AIML program

An AIML program is an XML document consisting of the elements defined in the AIML schema. It must consist exactly one aiml element:

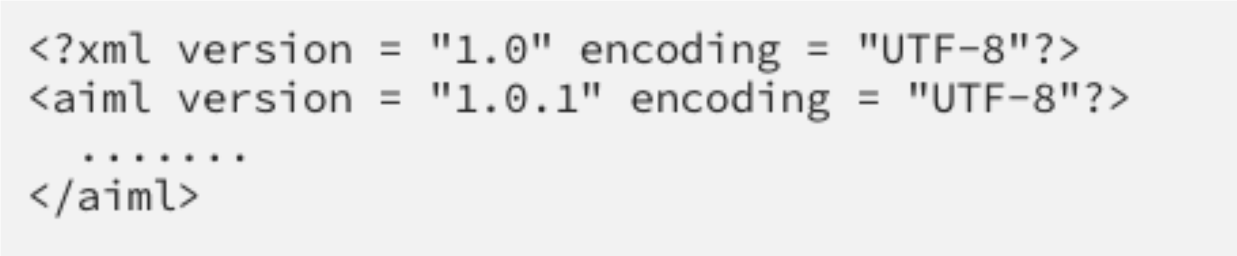
```
<aiml>
</aiml>
```

To start with the aiml, first of all, declare the starting tag i.e:

<?xml version=”1.0" encoding=”ISO-8859–1"?>

<aiml>

<aiml> tag is the parent tag, we need to close the aiml tag **</aiml>** as well at the end of the file.



```
<?xml version = "1.0" encoding = "UTF-8"?>
<aiml version = "1.0.1" encoding = "UTF-8"?>
  .......
</aiml>
```

AIML Basics

- A rule is defined with the category element, so the AIML document should contain a sequence of the category elements. Each one should contain one **pattern** element and one **template** element, which define the **condition** and **action** parts, respectively:

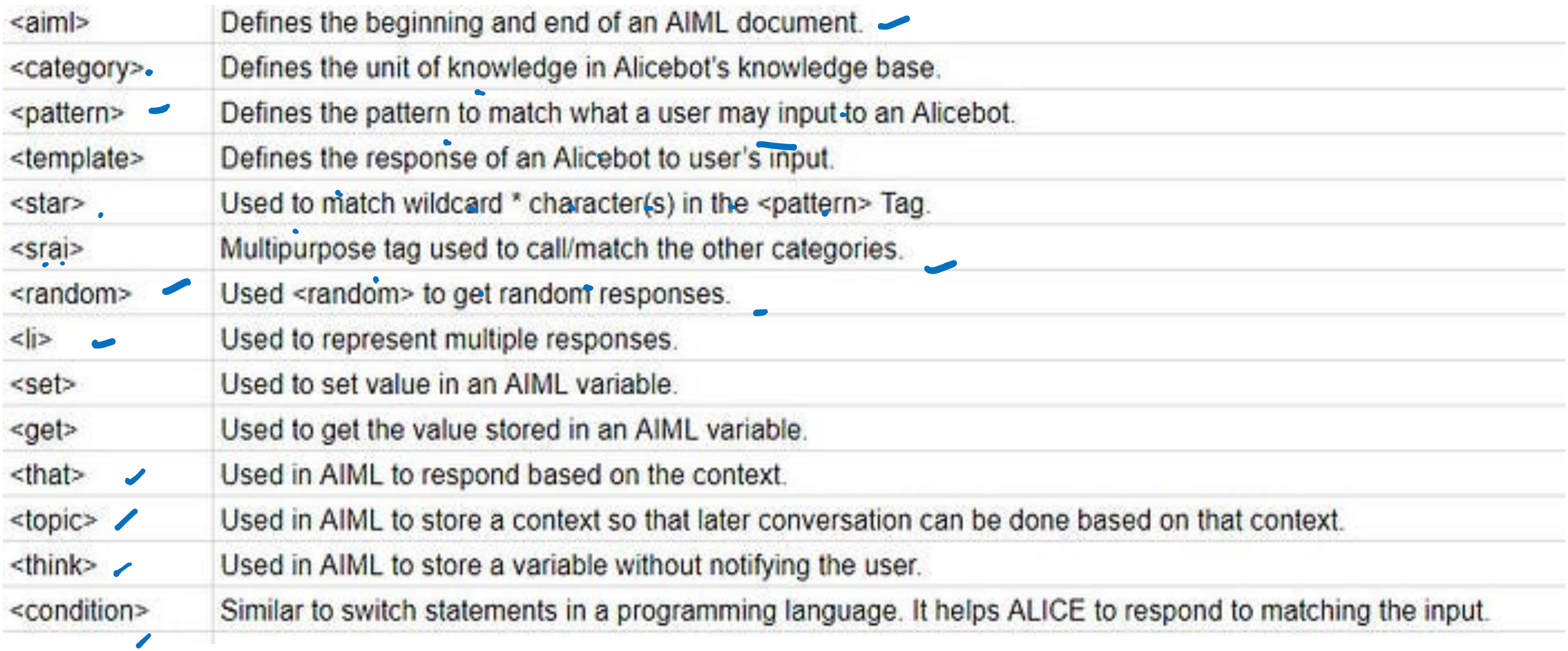
```
<aiml>
        <category>
                <pattern>HELLO</pattern>
                <template>Hi, how are you?</template>
        </category>
</aiml>
```

Rule

Condition

Action

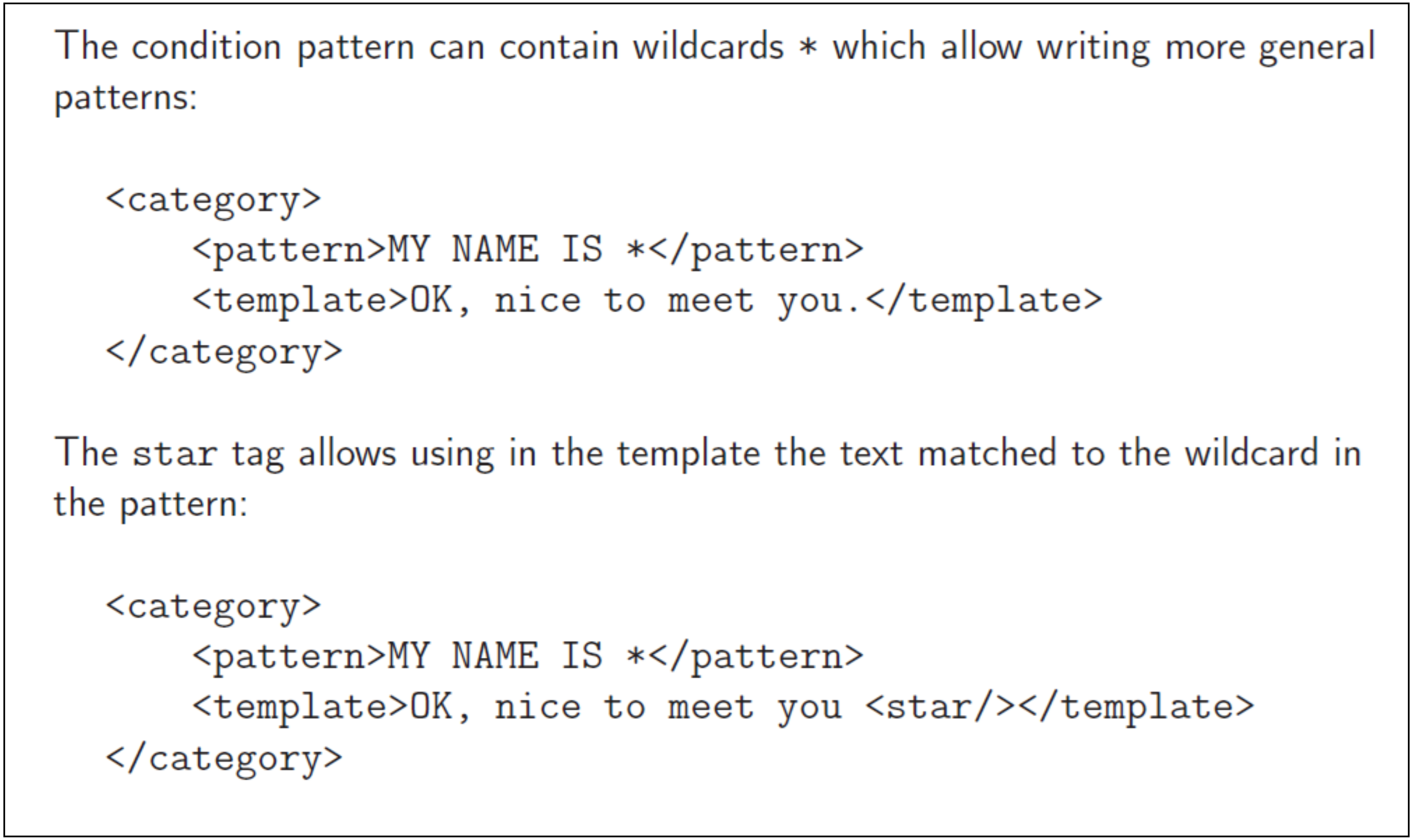
AIML tags



<aiml>	Defines the beginning and end of an AIML document.
<category>	Defines the unit of knowledge in Alicebot's knowledge base.
<pattern>	Defines the pattern to match what a user may input to an Alicebot.
<template>	Defines the response of an Alicebot to user's input.
<star>	Used to match wildcard * character(s) in the <pattern> Tag.
<srai>	Multipurpose tag used to call/match the other categories.
<random>	Used <random> to get random responses.
<li>	Used to represent multiple responses.
<set>	Used to set value in an AIML variable.
<get>	Used to get the value stored in an AIML variable.
<that>	Used in AIML to respond based on the context.
<topic>	Used in AIML to store a context so that later conversation can be done based on that context.
<think>	Used in AIML to store a variable without notifying the user.
<condition>	Similar to switch statements in a programming language. It helps ALICE to respond to matching the input.

ALICE (Artificial Linguistic Internet Computer Entity) framework.

Wild Cards and Star Tag



The condition pattern can contain wildcards * which allow writing more general patterns:

```
<category>
    <pattern>MY NAME IS *</pattern>
    <template>OK, nice to meet you.</template>
</category>
```

The star tag allows using in the template the text matched to the wildcard in the pattern:

```
<category>
    <pattern>MY NAME IS *</pattern>
    <template>OK, nice to meet you <star/></template>
</category>
```

AIML-Based Chatbot Workflow

Step 1: Create AIML Files

- Write AIML files containing categories (<category>) for all possible user inputs.
- Use patterns (<pattern>) to define user queries and templates (<template>) for bot responses.

Step 2: Use AIML Interpreter

- Use libraries like python-aiml to parse and execute AIML files.

Step 3: Interaction

- The chatbot matches user input to patterns in AIML files and provides the corresponding response.

Step 4: Deploy

- The bot can run locally or be integrated into web apps or messaging platforms.

MyFirstCode

```
<aiml>
  <category>
    <pattern>WHAT IS YOUR NAME</pattern>
    <template>My name is ChatBot!</template>
  </category>

  <category>
    <pattern>MY NAME IS *</pattern>
    <template>
      <think><set name="username"><star/></set></think>
      Nice to meet you, <get name="username"/>!
    </template>
  </category>
</aiml>
```

<srai> tag and recursion

SRAI (Set Response And Inheritance)

SRAI tag is used to redirect a response to another pattern. This allows you to reuse patterns and avoid redundancy in your AIML code.

```
<aiml>
  <category>
    <pattern>HELLO</pattern>
    <template>
      <srai>HI</srai>
    </template>
  </category>

 <category>
    <pattern>HI</pattern>
    <template>
       I am a chatbot. How can I assist you today?
    </template>
  </category>
</aiml>
```

```
<aiml>
  <!-- Primary response for "HELLO" -->
  <category>
    <pattern>HELLO</pattern>
    <template>Hello! How can I help you today?</template>
  </category>

  <!-- Primary response for "HI" using <srai> to redirect to "HELLO" -->
  <category>
    <pattern>HI</pattern>
    <template>
      <srai>HELLO</srai>
    </template>
  </category>

  <!-- Primary response for "HOW ARE YOU" -->
  <category>
    <pattern>HOW ARE YOU</pattern>
    <template>I'm doing great! How about you?</template>
  </category>

  <!-- Alternative phrasing using <srai> to avoid redundant code -->
  <category>
    <pattern>HOW DO YOU FEEL</pattern>
    <template>
      <srai>HOW ARE YOU</srai>
    </template>
  </category>

</aiml>
```

How This Works:

1. If the user says **"HELLO"**, the chatbot replies with **"Hello! How can I help you today?"**.
2. If the user says **"HI"**, the <srai> tag redirects it to the "HELLO" category, so the bot replies with **"Hello! How can I help you today?"**.
3. If the user says **"HOW ARE YOU"**, the chatbot responds with **"I'm doing great! How about you?"**.
4. If the user asks **"HOW DO YOU FEEL"**, <srai> redirects it to the "HOW ARE YOU" response, keeping the script concise.

Why Use **<srai>**?

- Reduces redundancy by reusing existing responses.
- Makes the AIML script easier to maintain and update.
- Ensures consistent responses for similar user inputs.

<random> and <li>

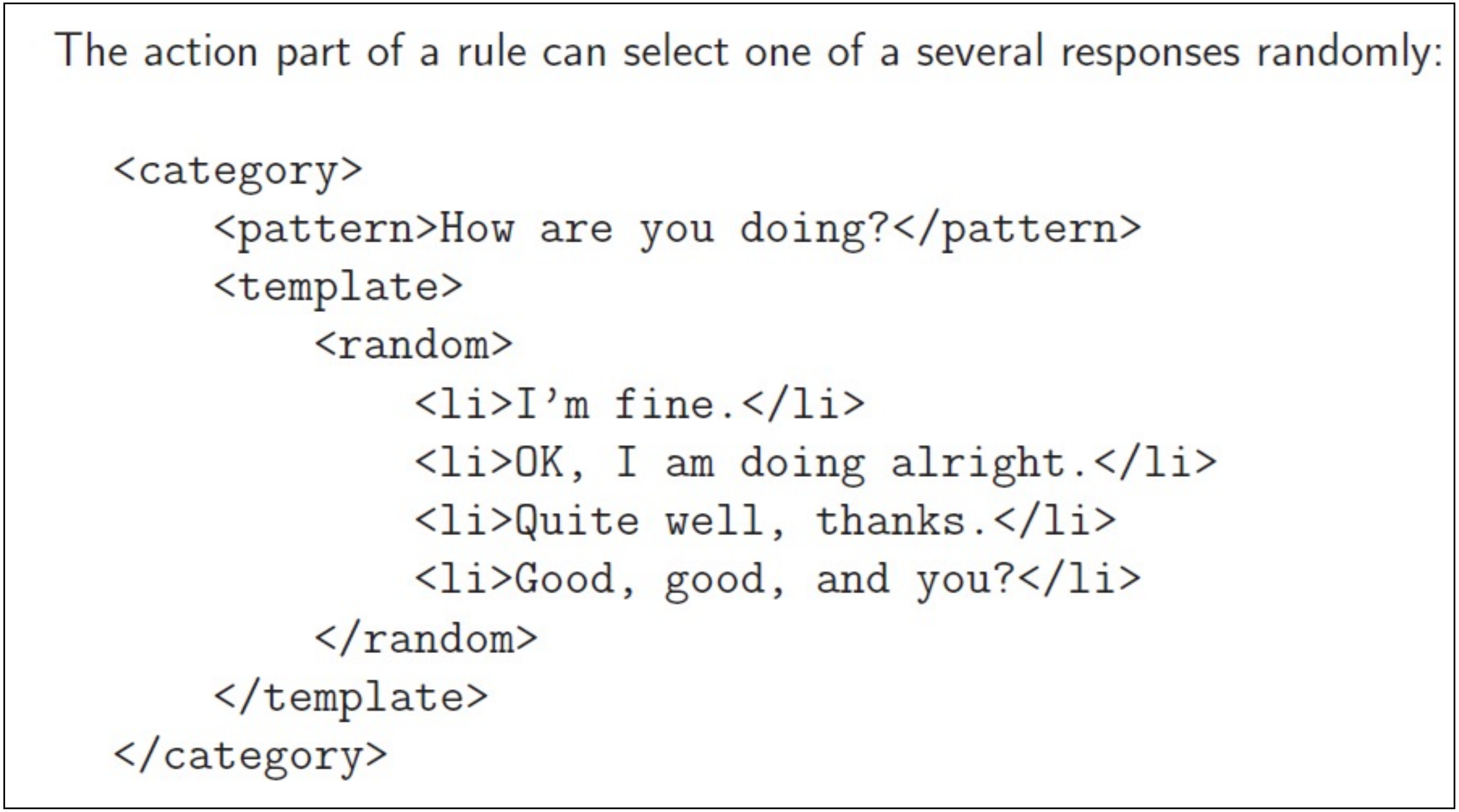
<random>

- Used to provide multiple random responses.
- Helps make chatbot interactions more dynamic.

<li>

- Used inside <random> to specify multiple response options.

Randomized Actions



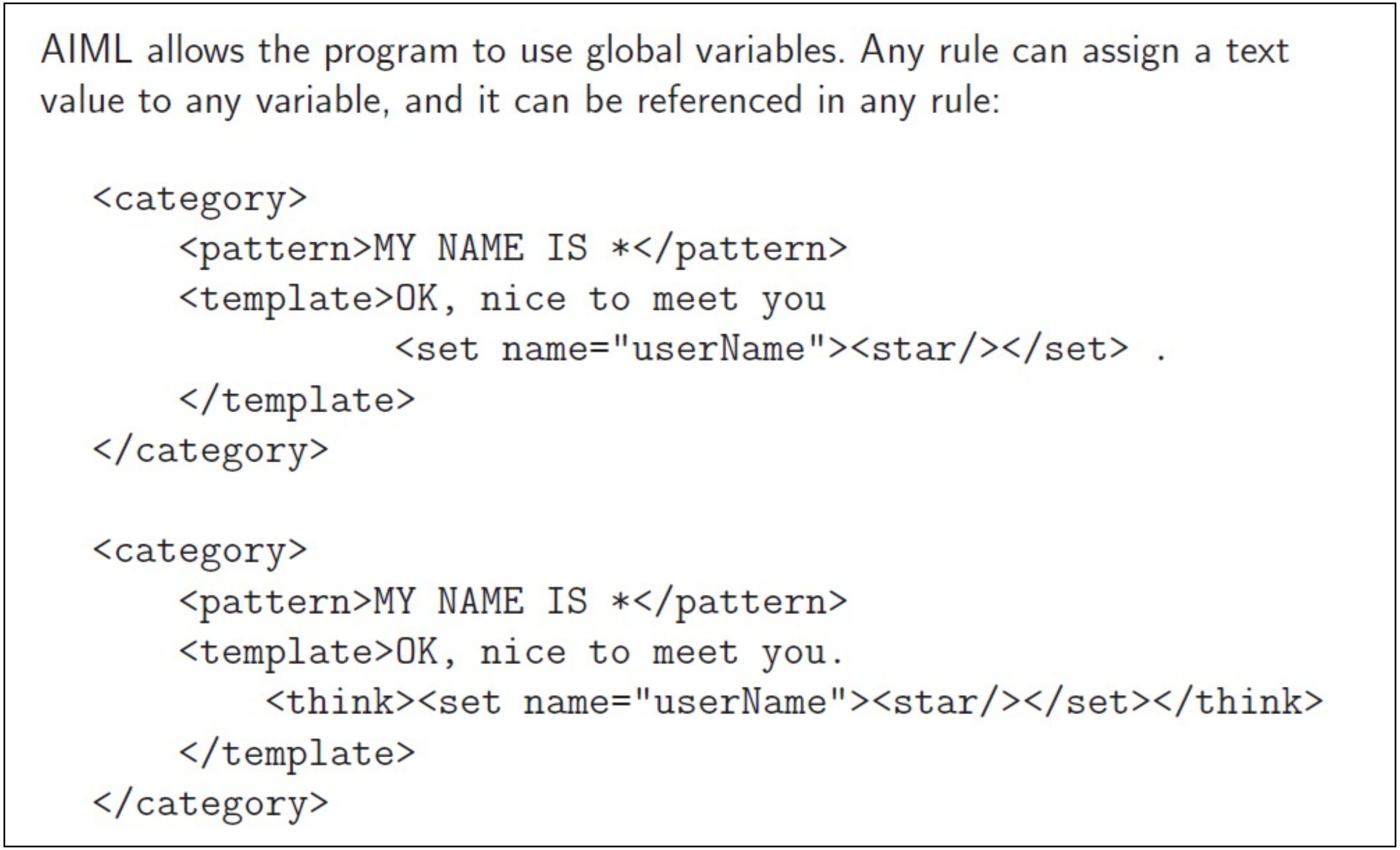
The action part of a rule can select one of a several responses randomly:

```
<category>
    <pattern>How are you doing?</pattern>
    <template>
        <random>
            <li>I'm fine.</li>
            <li>OK, I am doing alright.</li>
            <li>Quite well, thanks.</li>
            <li>Good, good, and you?</li>
        </random>
    </template>
</category>
```

Why Use <random> and <li>?

- Makes chatbot conversations feel more natural and engaging.
- Prevents repetitive responses, improving user experience.
- Useful for casual greetings, general questions, and small talk.

Setting Entity Values (Global Variables)



AIML allows the program to use global variables. Any rule can assign a text value to any variable, and it can be referenced in any rule:

```
<category>
    <pattern>MY NAME IS *</pattern>
    <template>OK, nice to meet you
              <set name="userName"><star/></set> .
    </template>
</category>

<category>
    <pattern>MY NAME IS *</pattern>
    <template>OK, nice to meet you.
        <think><set name="userName"><star/></set></think>
    </template>
</category>
```

innovate achieve lead

BITS Pilani
Pilani Campus

Design a Doctor Appointment Bot using AIML

Steps

1. Identify the main stories in the conversation flow [Ex: book appointment, reschedule appointment, cancel appointment etc.]
2. Design the AIML file, considering the category, patterns, and templates
3. Setup the Environment

- Install an AIML Interpreter
- You'll need an AIML interpreter to run the AIML files.
- One of the most popular options is **Program O** (for PHP) or **AIMLBot** (for .NET), or **Python's PyAIML**.

Steps

- **Install PyAIML:** If you have Python installed, you can use PyAIML, a Python implementation of AIML.

➢ pip install python-aiml

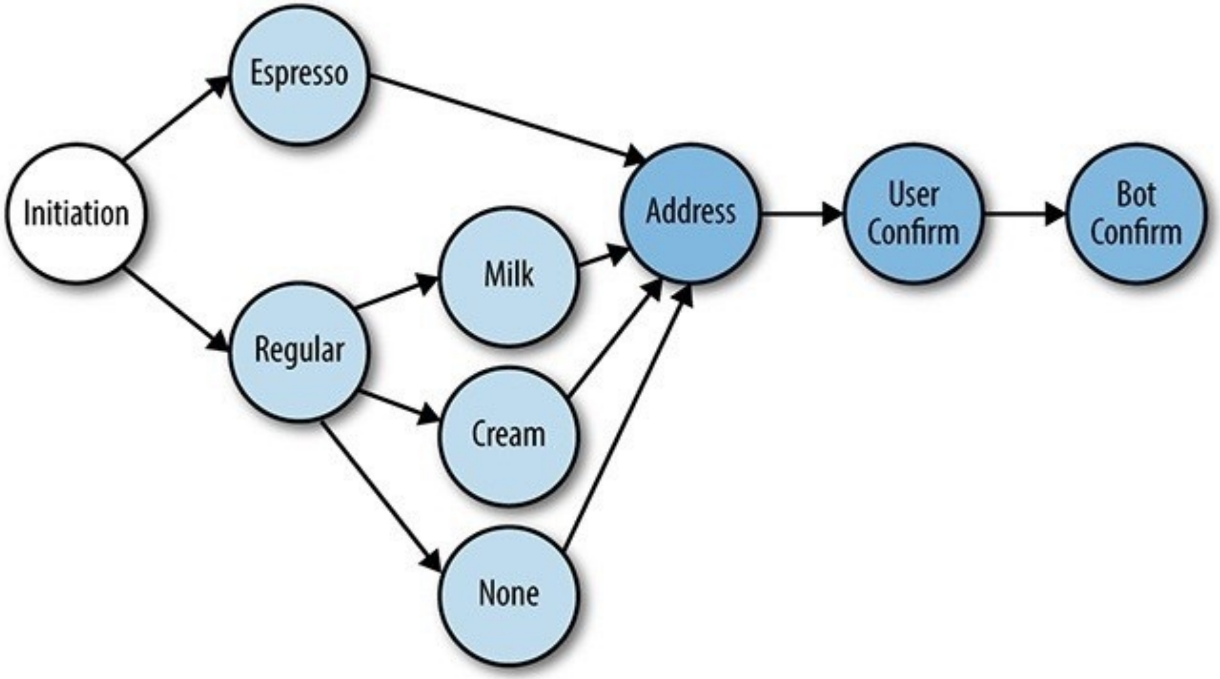
4. Write a Python script to run the AIML files and start the conversations

Steps

1. Main code file is run_bot.py
2. AIML files
    i. onboarding.aiml
    ii. onboarding_redirection.aiml
    iii. user_name.aiml
    iv. default_handle.aiml
    v. all_scenario.aiml
3. Run the python file in Terminal with the command “python run_bot.py”

Exercise [15 mins]

Design the conversation flow for “Coffee Bot” based on the below workflow. Make Suitable Assumptions as needed. No need to use wildcards, you can hardcode the patterns.



Thank You!