

Saraswathy K
14-07-2025

Discussion & Improvement Suggestions


Enhancing flow continuity

Existing flow:

- The system forgets that the user just got their account ID from the bot.
- It doesn't reuse the slot clarification that was just resolved, even though that information was already made available.
- The user has to repeat their intent and provide the ID again, creating unnecessary steps.

Enhancements:

- Avoid forcing the user to repeat intent or known information.
- Automatically resume the flow once missing slots are available.
- Respect contextual memory — especially when the bot just offered the answer.



check my balance

Got it, for security reasons can you please provide me your account number please?

AB1234

Alright.

I can only check the balance for you for the accounts you own.

Unfortunately, The id you just provided "AB1234" is not in the list of your accounts.

You are actually authenticated, in case you don't recall your account id

I can look it up for you.

Would you like that?

Yes

No

/SetSlots(provide_account_id=True)

Sure thing.

Here is the account id EF1234 associated to your customer id 125.

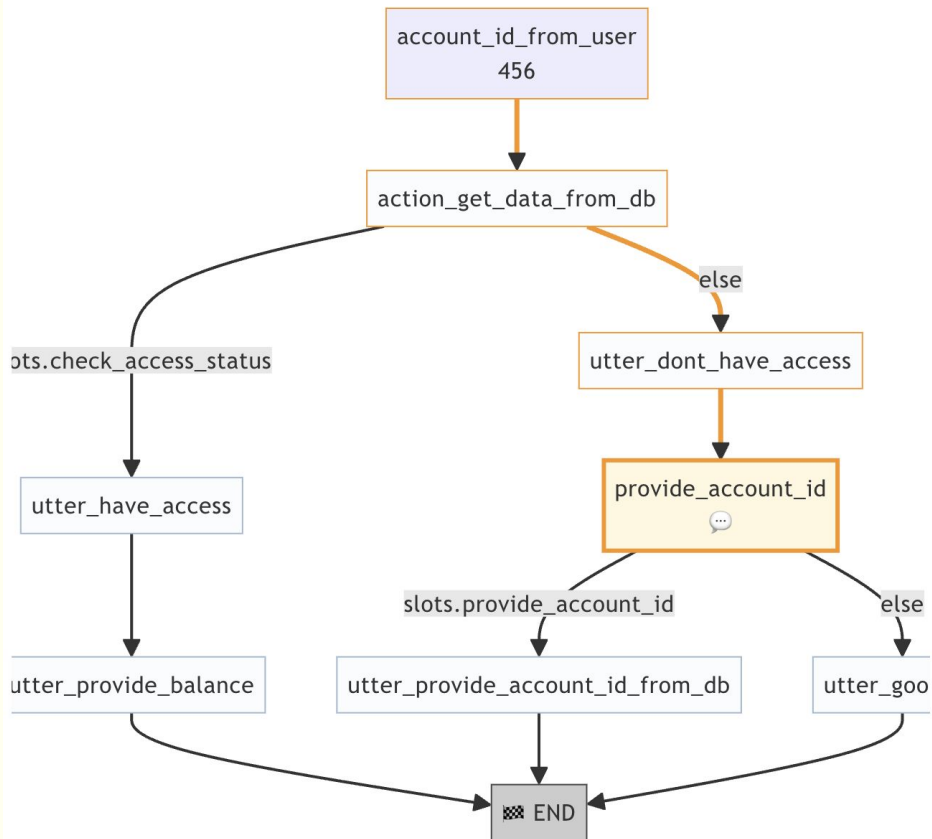
check my balance then

Got it, for security reasons can you please provide me your account number please?

EF1234

Existing Flow

Flow: check_balance



Enhancement

utter_provide_account_id_from_db

Confirm slot correction
(account id from user)

Confirm check_balance from id

account_id_from_user

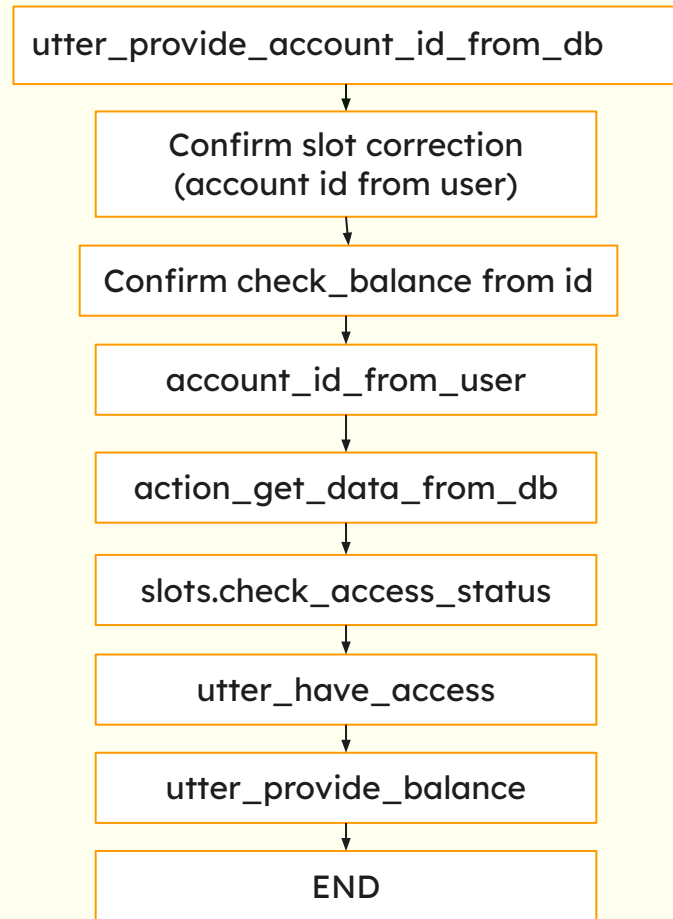
action_get_data_from_db

slots.check_access_status

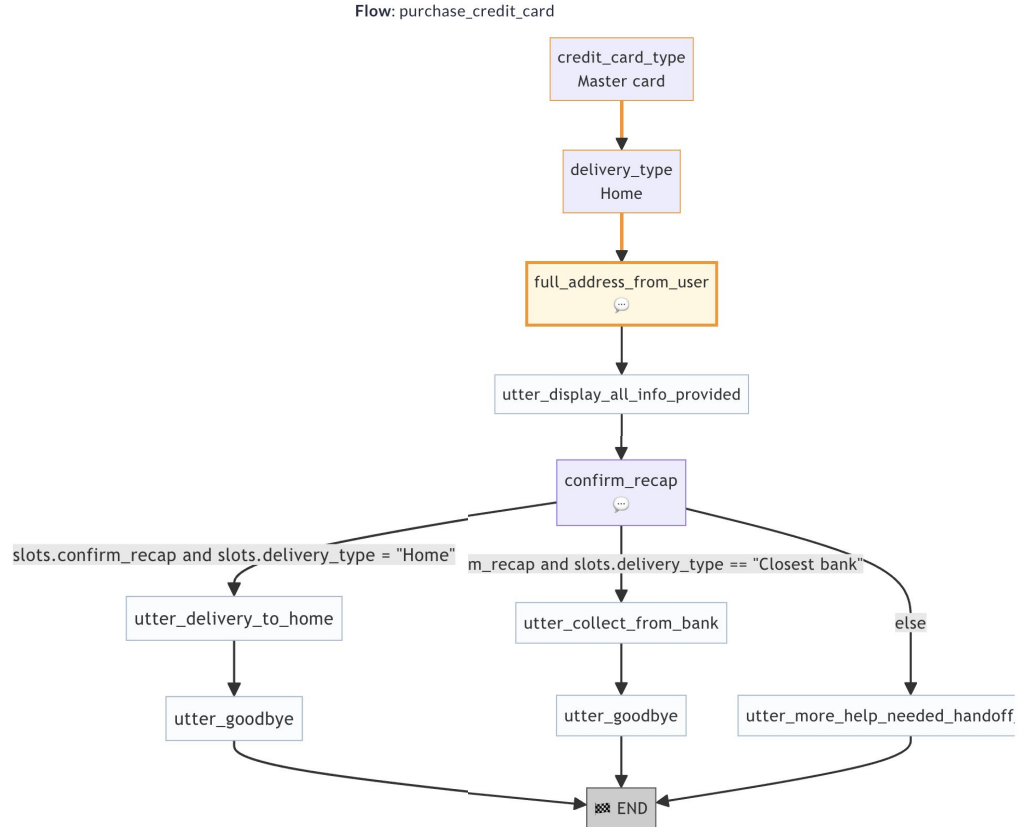
utter_have_access

utter_provide_balance

END



Enhanced Purchase credit card flow



For real world applications



- The CALM approach avoids brittle intent taxonomies and reduces manual NLU overhead, which makes it faster to iterate and easier to scale conversational coverage.
- Ensure flow and slot names are semantically rich and clearly separated to avoid hallucination by the model.
- Use a modular structure where each flow is defined in a reusable YAML allowing team-based development. e.g. the `purchase_credit_card` flow can be maintained independently of `check_balance`.
- Maintain a registry of flows and render only relevant ones per context to improve model performance and latency.
- Integrate `embedding-based` flow retrieval with `flow descriptions` to dynamically surface the most relevant flows in the prompt, reducing prompt size and improving action accuracy.
- Use prompt versioning so you can test changes in prompt instructions without breaking existing logic.
- Add a config param like `max_prompt_tokens`, and enforce it before sending the request to limit cost.

For real world applications



```
===
Here is what happened previously in the conversation:
USER: hello help me buy a credit card
AI: Certainly, I can help you with that. what kind of credit card you are looking to buy
USER: /SetSlots(credit_card_type=Master card)
AI: Okay let's move to the next step of the credit card purchase Would you like your new credit card to be delivered
USER: /SetSlots(delivery_type=Closest bank)
AI: Understood! Can you provide me your full address please?
USER: oh wait I want visa to deliver to home
AI: Do you want to update the credit card type to Visa delivery type to Home ?
USER: /SetSlots(confirm_slot_correction=True)
AI: Ok, I am updating credit_card_type, delivery_type to Visa, Home respectively.
AI: Understood! Can you provide me your full address please?
USER: 123, Bakers st
AI: Let's make sure all information provided are in order Credit card type: Visa Delivery type: Home Home address: 123, Bakers
AI: Can you confirm that all the information captured above is accurate:
USER: /SetSlots(confirm_recap=True)
AI: Thank you! Your credit card order has been placed successfully. It will be delivered to your address within 7-10 business
you have any questions, feel free to ask!
AI: I hope I was helpful today! Let me know if you need more assistance.
USER: check my balance
```

- Manage current_conversation length with summarization or sliding window to keep the prompt within token budget while maintaining context. Eg: even after transitioning to a new flow, the conversation history still retains the previous flow's dialogue, which could be discarded once the previous pattern is completed and not contextually linked.



**THANK
YOU**