Goal

The goal of the project is to propose a formalism for design of conversational AIs, and possibly also to implement a solution that allows the formalism to be used in practice.

Background

In recent years, various platforms for developing and designing conversational systems have emerged, such as DialogFlow, IBM Watson Assistant and Amazon Lex. These platforms enable users to create chatbots and virtual assistants, typically by providing a web-based graphical user interface, and in some cases also an API. By these means, a developer can express phrases that the system should be able to understand, and specify information that the system needs to gather from the user in order to proceed with a task.

Problem description

Despite the fact that the platforms share terminologies to a large extent - e.g. they refer to “intents”, “entities” and “slots” - they do not offer any official formalism that cover the concepts and relations underlying their user interfaces and APIs. (Existing formalisms such as VoiceXML and AIML are outdated, only deal with very simple forms of conversation and are not supported by the mentioned platforms.)

The project investigates whether platforms for conversational AI would benefit from a modernized formalism for conversation design, e.g. in the form of a programming language or data model. Several questions tie into this theme, such as: Would a formalism give developers more control? What kind of concepts and relations should the formalism contain? Should the formalism be declarative or imperative? Could a formalism be treated as a standard, serving as a benchmark for conversational capacities of a platform?

One concrete aim of the project is to propose a formalism. Furthermore, it may be interesting to implement a solution allowing a conversation developer to use the formalism in conjunction with at least one selected platform. This way, the student could demonstrate how a specific program or set of declarations results in a concrete and demonstrable system.

The project is proposed by Talkamatic, a company developing a platform for conversational AI called Talkamatic Dialogue Manager (TDM). Today, TDM uses an XML based formalism for dialogue development (see https://tdm.readthedocs.io/en/latest/tutorial/). The student is encouraged to study this formalism as an example and may use it as a source of inspiration, but the project is intended to be open-ended and not limited to Talkamatic’s platform.
Recommended knowledge and skills

- Formalizing concepts and relations (e.g. with ontologies and databases)
- Programming languages of different paradigms (e.g. declarative and imperative)
- Web APIs
- Interest in Conversational systems

Supervisors from Talkamatic

- Alex Berman
  alex@talkamatic.se