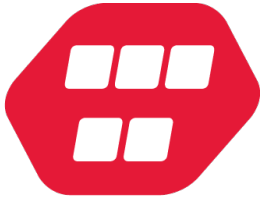


Talkamatic

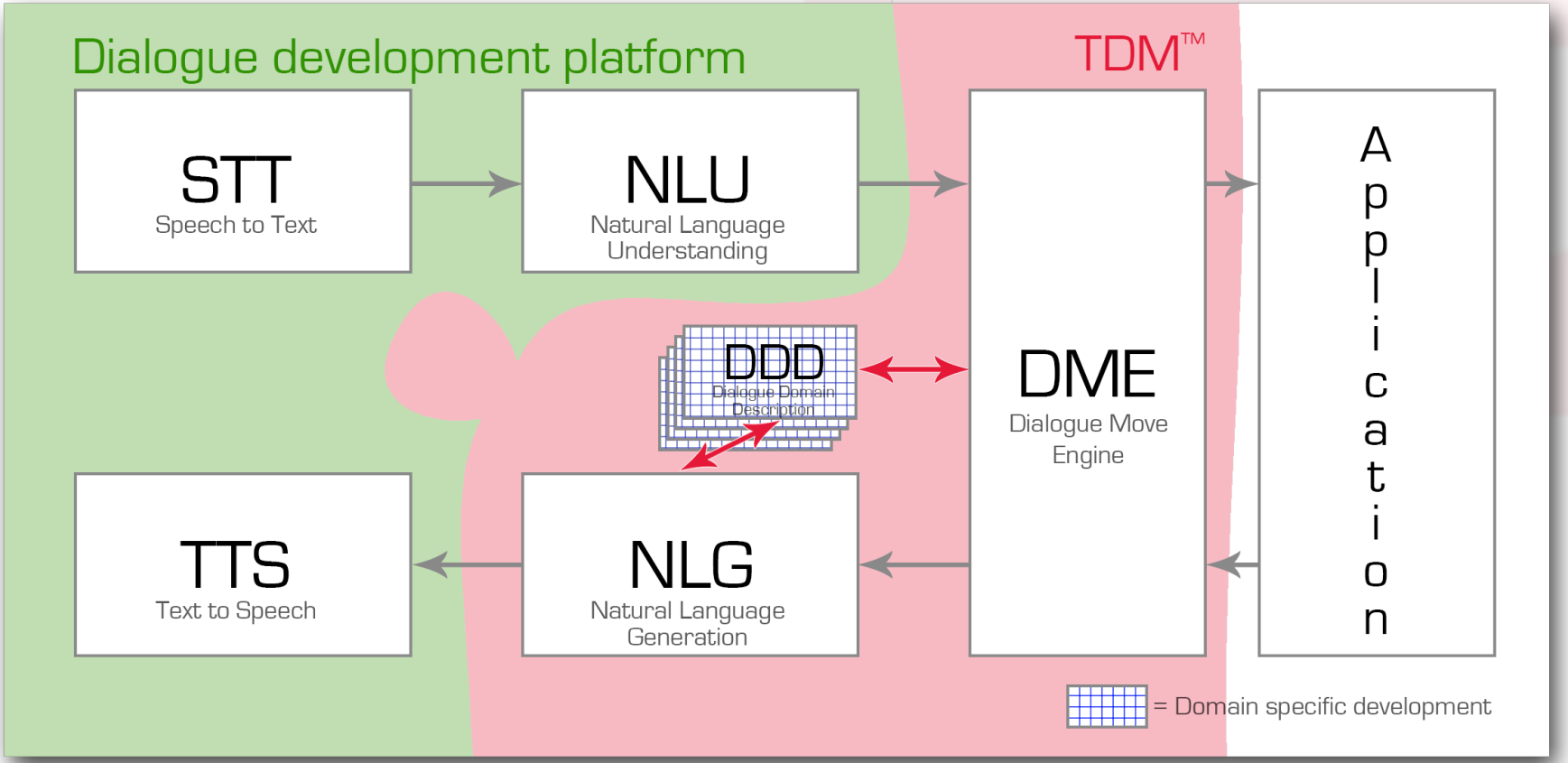
NEXT GENERATION OF CONVERSATION

Dialog Development with TDM

2019

Marta Pantaleeva • Ph: +46-739-241308 • Mail: Marta@talkamatic.se

TDM Integration



TDM can be integrated into any dialogue platform of choice adding shorter development times, support for automated AI dialogue features, multi-language support and multimodality*. TDM is accessible in the cloud or locally to match your performance and security demands. Our DDD approach allows dialogue across multiple domains as well as parallel development and a system that is easier to upgrade and maintain.

* *Synchronised integration between input and output from any sensory system including GUI, voice, text and others*

Talkamatic
NEXT GENERATION OF CONVERSATION

Introduction to TDM

- TDM = Talkamatic Dialogue Manager
- Enables natural, easy-to-use dialogue
- Features low distraction and high efficiency
- Allows rapid prototyping thanks to built-in dialogue design

Talkamatic
NEXT GENERATION OF CONVERSATION

Introduction to TDM

- TDM is generic
- Easy to add new apps / domains
- Supports multimodality (GUI + speech)
- Supports Multilinguality
- Supports many different speech recognizers and text-to-speech engines

Technical architecture

- TDM runtime consists of a backend and frontend
- Backend supports multiple simultaneous frontends
- Backend can run in the cloud or locally
- Supported frontends:
  - Android
  - Google Assistant
  - Alexa
  - Twilio
  - Text I/O via console (cross-platform, for development)
- API for HTTP frontends
- API for HTTP services

Dialogue development philosophy

- All dialogue logic is in TDM, not in the dialogue apps
- Dialogue apps are resources, not programs
- Dialogue apps provide high-level information
- Dialogue developers do not need to implement dialogue logic (questions, answers, feedback, etc.)

TDM SDK

- App formalism:
  - Ontology
  - Grammar
  - Domain
  - Service interface
  - Interaction tests (optional)
  - (Services)
- XML format
- Console-based tools
- Future GUI

Talkamatic
NEXT GENERATION OF CONVERSATION

Ontology

- Actions
  - *'make_call', 'send_message', 'receive_call', 'receive_message'*
- Sorts
  - *'contact'*
  - Built-in sorts: integer, float, string, boolean
- Predicates
  - *'contact_to_call'* of sort *'contact'*
  - *'contact_to_message'* of sort *'contact'*
- Individuals
  - *'otto'* of sort *'contact'*
  - However: individuals are typically added on the fly, rather than pre-defined

Talkamatic
NEXT GENERATION OF CONVERSATION

Grammar

- Describes how user and system phrase themselves
- Used for understanding what the user said
- Used for generating system responses
- Consists of mappings between words and semantics
- Examples:
  - “make a phone call” means *request the action 'make_call'*
  - “call X” means *request the action 'make_call' with 'contact_to_call=X' (or with 'contact=X')*

Talkamatic
NEXT GENERATION OF CONVERSATION

Domain

- Describes how actions are carried out and what information they need
- Describes how questions from user are answered
- Example:
  - *To perform 'make_call', first find out 'contact_to_call'*
  - *To answer 'who_called', query the 'phone' service*

Dialogue capabilities

- Over answering
- Other answering
- Answer revision
- Topic shifts
- Task recognition

Talkamatic
NEXT GENERATION OF CONVERSATION

Answers

Over-answering: Handling answers to questions that give more (task-relevant) information than was requested by the system.

```
U (User): Make a call
S (System): OK, who do you want to call?
U: John's mobile [not requested by the system] S: OK,
placing the call.
```

Other-answering: Handling answers to questions that give different (task-relevant) information than was actually requested by the system.

```
U: Make a call
S: Who do you want to call?
U: The mobile number [not requested by the system] S:
OK. Who do you want to call?
U: John
S: Calling John's mobile
```


Talkamatic
NEXT GENERATION OF CONVERSATION

Correcting the system

Denial:

S: Who do you want to call?
U: Call John
S: John's home or mobile number?
U: **Not John**
S: **OK, who do you want to call?**

Repair:

S: Who do you want to call?
U: Call John
S: John's home or mobile number?
U: **Not John, Steve!**
S: OK, Steve's home or mobile number?

Post-answer revision:

U: Do I have a meeting tomorrow at 9?
S: You don't have anything in your calendar at 9 pm.
U: **No, a.m.**
S: You have a meeting tomorrow at 9 am.

Talkamatic
NEXT GENERATION OF CONVERSATION

Topic shifts

Returning from two sub-dialogues across independently developed domains.

U: **Call John [Phone domain]**

S: What number? Work, home or mobile?

U: **When does the train leave? [Public transport domain]**

S: Which train?

U: **I'm freezing! [Climate domain]**

S: The temperature was increased to 21 degrees.

Returning to public transportation. What station are you interested in?

U: Oslo

S: The next train to Oslo leaves in 20 minutes.

Returning to calling. What number do you want to call?

U: The home number

Talkamatic
NEXT GENERATION OF CONVERSATION

Task recognition

Recognizing task from user-supplied answer

U: John
S: OK, you want to make a call. John's home or mobile number?
U: ...

Clarify unspecified goal inferred from user-supplied answer(s).

U: **John**
S: Do you want to make a call or send a text message?

Talkamatic
NEXT GENERATION OF CONVERSATION