Natural Language Queries on Natural Language Data: a Database of Meeting Dialogues

Maria Georgescul
ISSCO / ETI
University of Geneva, Switzerland

Burg (Spreewald), Germany, Jun 23, 2003
8th International Conference on Applications of Natural Language to Information Systems

Acknowledgements

- This work is part of the Multimodal Dialogue Management (MDM) project
  - http://www.issco.unige.ch/projects/im2/mdm/
  - http://www.im2.ch/
- Participants:
  - ISSCO/ETI/UNIGE: Susan Armstrong, Alexander Clark, Maria Georgescul, Andrei Popescu-Belis, Marianne Starlander
  - LITH/EPFL: Giovanni Coray, Vincenzo Pallotta
  - LIA/EPFL: David Portabella, Martin Rajman
- Thanks to ICSI (Berkeley, USA) Meeting Recorder project for access to their data

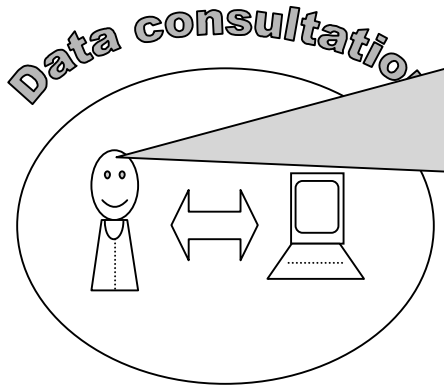
Outline

- Introduction
- Source Data and Initial Formatting
- Shallow Dialogue Analysis
- Annotation of Argumentation Structure in Meetings
- Database Structure and Access
- Multimodal Interactive Query Interface
- Conclusion

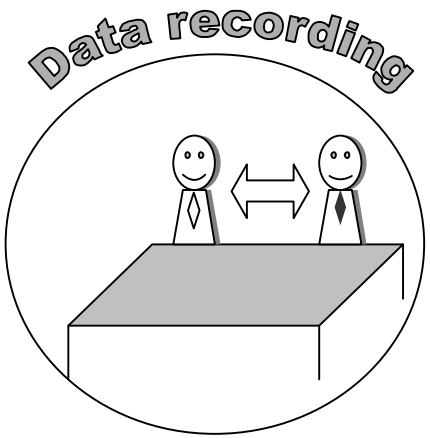
Introduction

- Objectives: processing, storage and retrieval of natural language dialogues that occur in meetings (e.g. staff meetings)
- Our approach:
  - extraction of semantic content of dialogues, mainly from the linguistic surface expression
  - incorporate additional information extracted from other communication modalities (such as expression of emotions, pointing, etc.)
  - multimodal user interface for dialogue retrieval

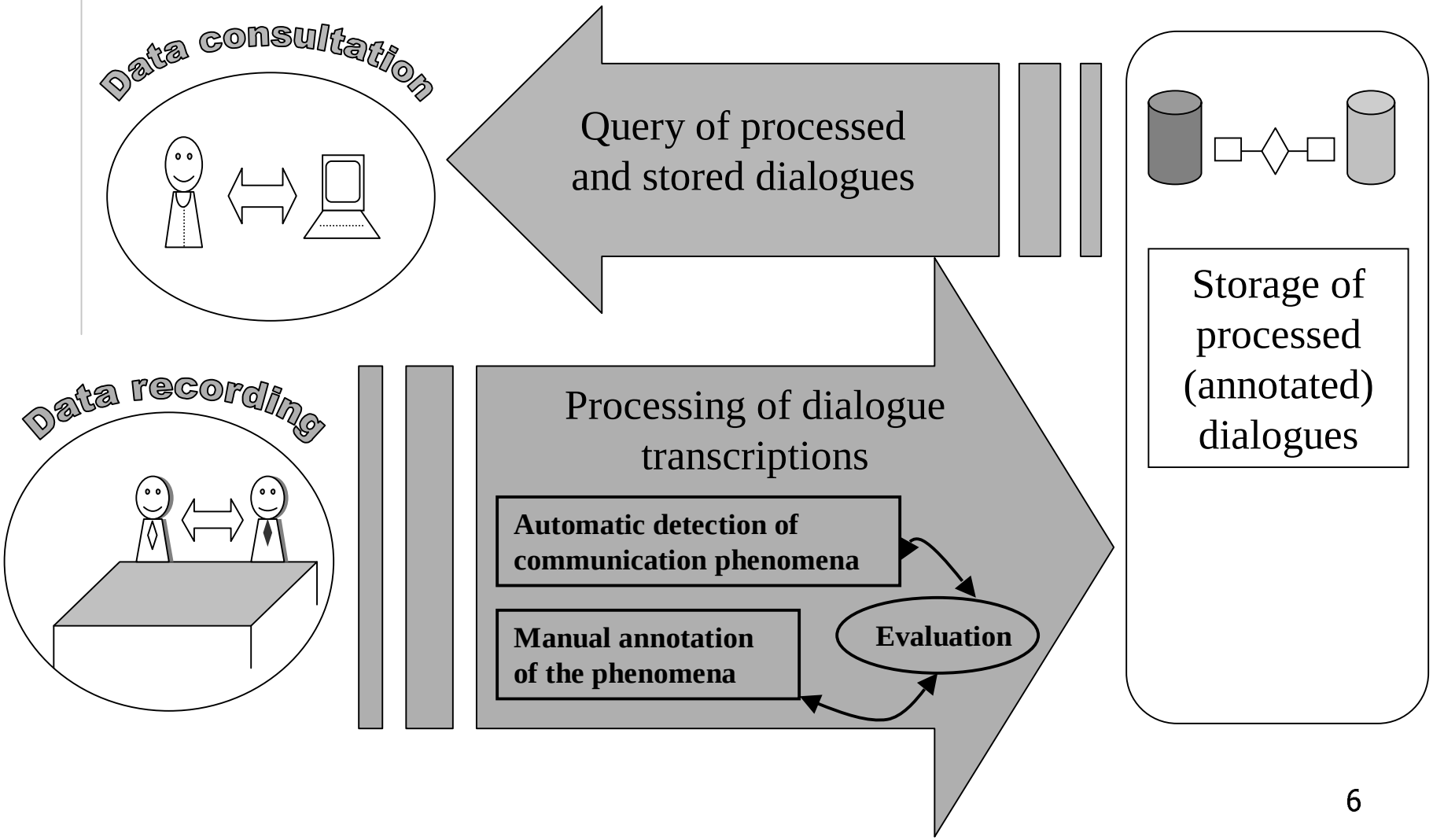
Application: Database of Meetings



- **Information about**
  - **the participants**
  - **the issues**
  - **the decisions**
  - **the documents (such as presentation handouts, reports, agenda, etc)**



Application: Database of Meetings



Source Data

- Resources
  - Meeting recording rooms having (ideally)
    - *individual microphones* for each participant
    - *far-field microphones*
    - *individual cameras* and *overview cameras*
    - cameras or intelligent boards for meeting material (slides, documents, drawings)
  - ICSI (Berkeley, USA): meeting transcriptions - EN
  - IDIAP (Martigny, Switzerland) : meeting transcriptions – EN
  - ISSCO(Geneva, Switzerland): transcription of TV show – FR

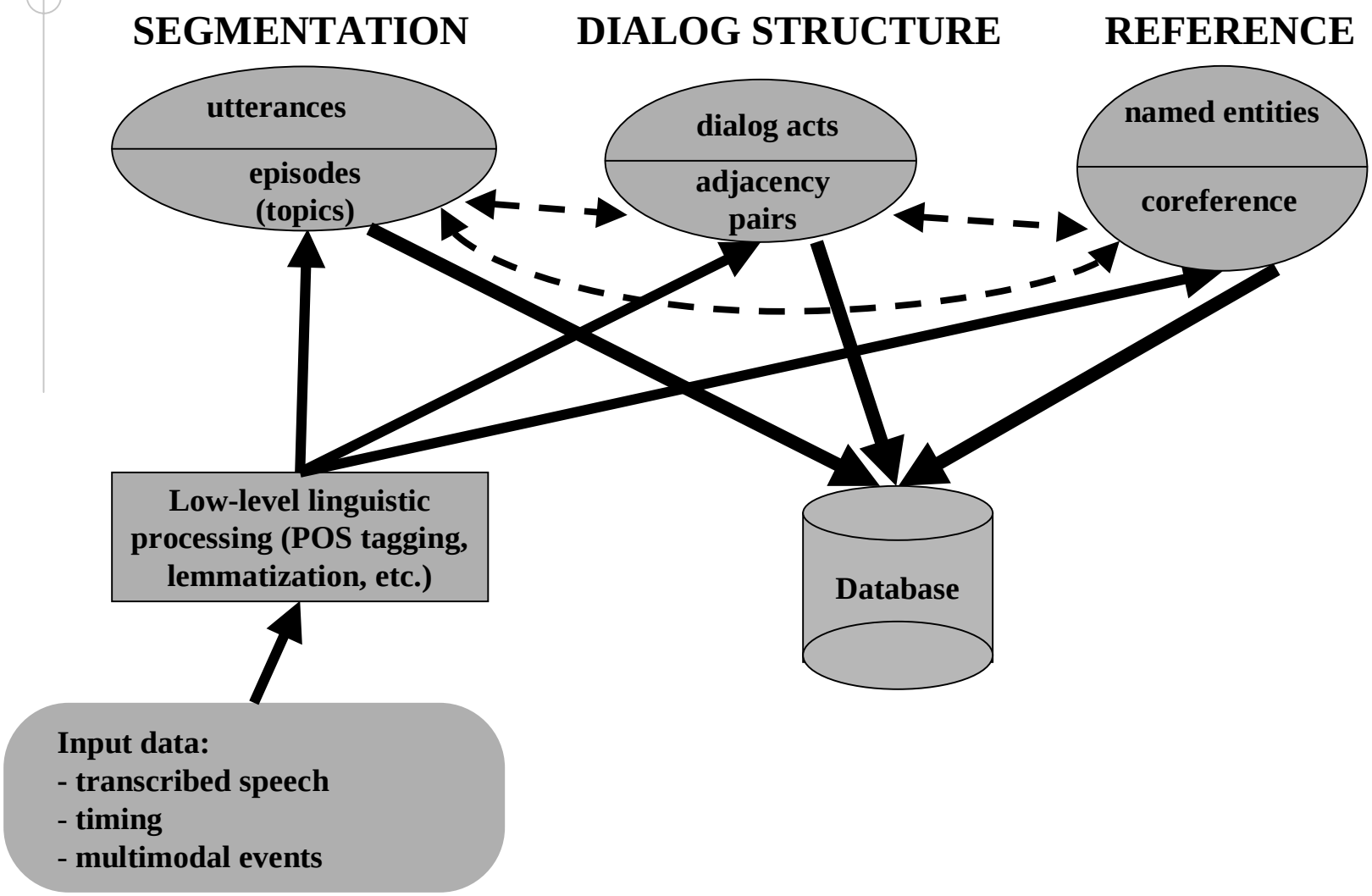
Source Data and Initial Formatting

- Human transcriptions of each channel/speaker's discourse
- Manual annotation of shallow dialogue phenomena in order to prepare test data
- Data conversion
  - 1: to XML from various annotation formats
  - 2a: from XML to HTML for visualization
  - 2b: from XML to tabular format for the database

Shallow Dialogue Processing

- Simplified linguistic descriptions
- Computationally feasible to automatically derive these features with reasonable accuracy
- Extracts dialogue information for intelligent and flexible queries and retrievals

Overview of Shallow Dialogue Processing



Dialogue Segmentation

- Minimal structure
  - dialog = **[**episode**]+** (topic change)
  - episode = **[**turn**]+** (speaker change)
  - turn = **[**utterance**]+** (function change)
  - utterance = **[**a complete sentence **|**
    a disfluent sentence aborted in mid-utterance **|**
    an interjection **|**
    a back-channel response indicating attentiveness **|**
    a stand-alone phrase **]**

Issues in Dialogue Segmentation

- Utility
  - utterances: detect dialogue acts, align multilingual texts, translation
  - episodes: topical indexing, dialogue tracking and understanding, summarization
- Challenges
  - meetings are topic-oriented, not task-oriented (as were previously studied domains)
  - spoken dialogues have many non-propositional utterances: hesitations, repairs, backchannels
  - dialogue transcriptions lacks typographic paragraph-marking

Dialogue Structure

1. Dialog acts
   - Task: tagging each utterance with dialog acts
     - Dialog act = the function of an utterance (e.g. "question", "statement", "backchannel", etc.)
   - Many different annotation schemes based on different theories of discourse
     - ICSI set: 57 dialog acts - simplified for our needs in ~20 classes
2. Adjacency pairs
   - Adjacency pair = functional link between two utterances (e.g. "question/answer", "invite/accept", "offer/decline", "greeting/greeting", "request/accept")

How to Detect the Dialogue Structure?

- Use Machine Learning techniques to induce classifiers
- Different sources of information
  - Local information – sequence of words that make up the utterance
  - Prosodic information – intonation, pause lengths, etc.
  - Contextual information - surrounding utterances

Named entities and Coreference (1)

- Named entities (NE)
  - names of people, companies, places, etc.
  - times, dates, etc.
- Referring expressions (RE)
  - refer to a person, object, event, … = *entities*
- Tasks:
  - detect occurrence and type of NE and RE
  - coreference resolution: detect relation between REs that refer to the same entity -> sets of coherent REs

Named entities and coreference (2)

- Utility
  - relation between salient referents and topics
  - indexing and retrieval of persons and entities
- Meeting dialogues
  - multimodal referring acts: pointing
  - reference to objects in other modalities
  - reference to documents: slides, reports, agenda

Annotation of Argumentation Structure (1)

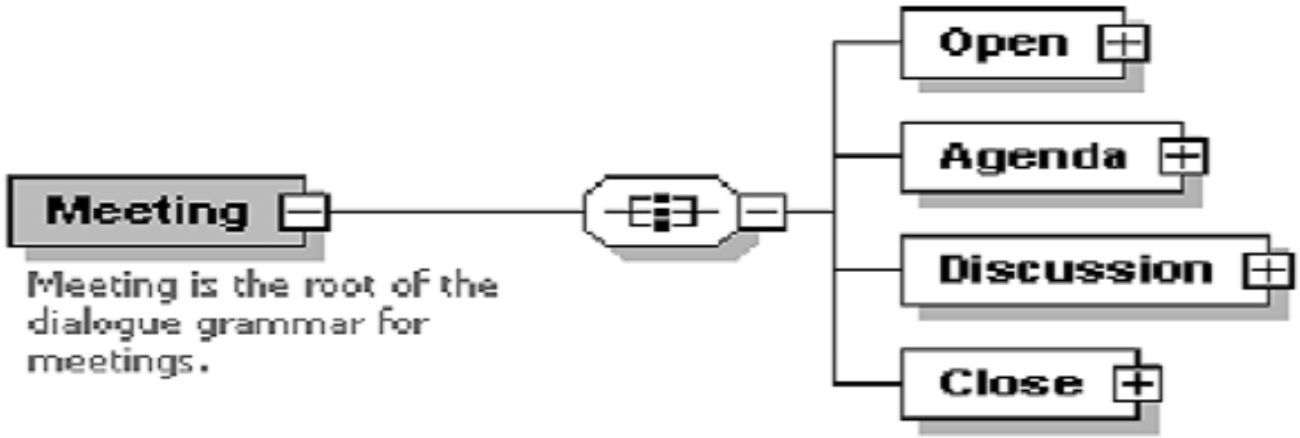
- Goal: formally define the protocols of meetings where a decision-making process is taking place
- Find relevant answers to user queries…
  - About discussion, arguments invoked, questions raised:
    - "Let me see the moment when _XX_ and _YY_ strongly disagree"
    - "What were the objection to the proposal _ZZ_?"
  - About decisions, consensus achieved
    - "Which arguments did the members who disagreed on the decision _D1_invoke?"
  - About consistency, fairness
    - "Was the decision about issue X democratically made?"

Annotation of Argumentation Structure (2)

- Our approach:
  - Based on formal theories of argumentation
  - Annotate the parts of the meeting where a decision process takes place with a suitable "argumentative structure":
    - what issues have been discussed?
    - what alternatives have been proposed and accepted by the participants?

Annotation of Argumentation Structure (3)

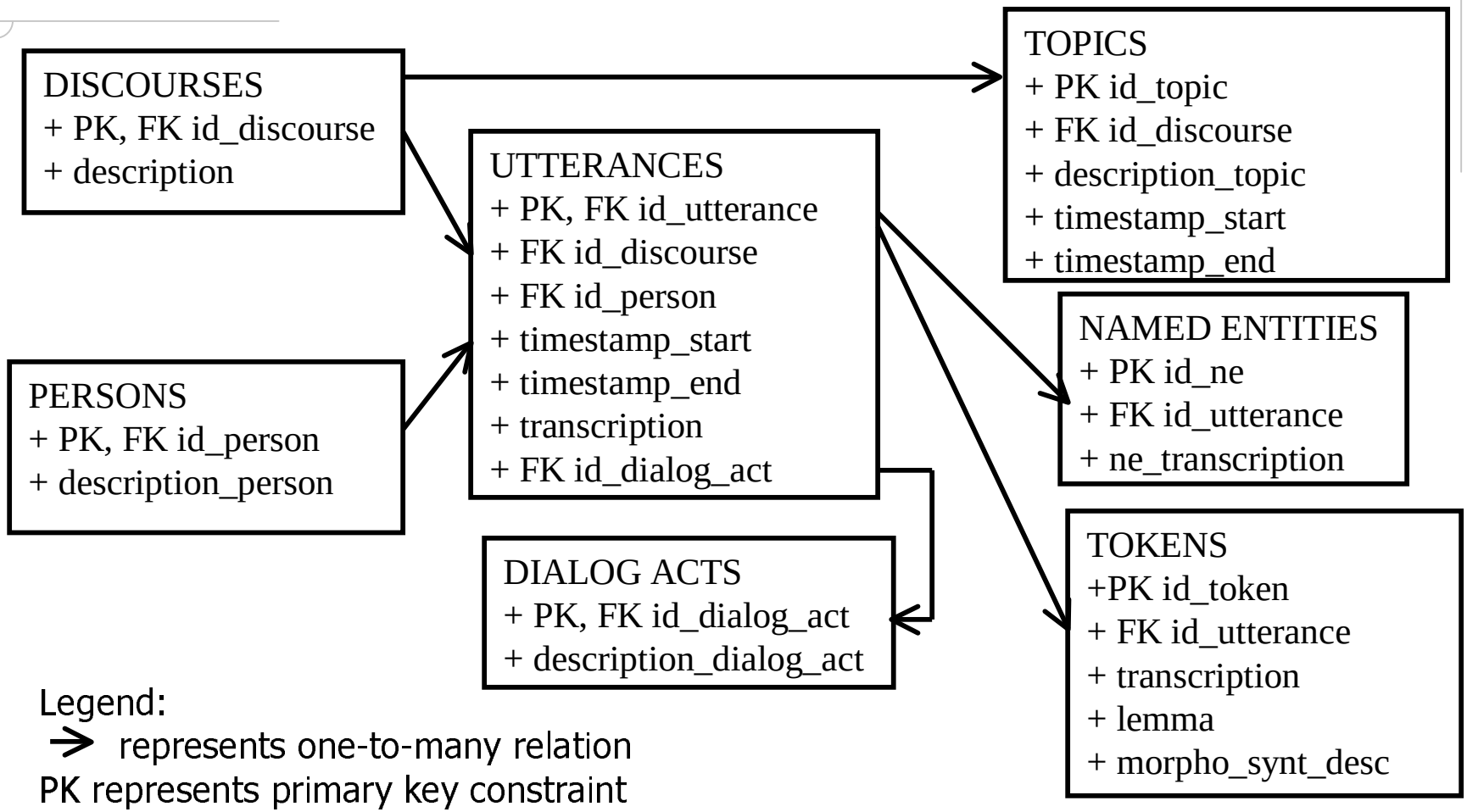
- Model adopted:
  - IBIS (Issue Based Information Systems) - model of argumentative structure, proposed by [Kunz and Rittel 1970]
  - Discussion graphs for representing a multi-agent decision making processes -> Meeting Description Schema
  - e.g.



Annotation of Argumentation Structure – Ongoing Work

- Ontology
  - Classification of different types of meeting
  - Meeting Description Schemas (DS) for different types of meeting
  - Validation of DSs on available meeting corpora
- Tools:
  - Annotation tool based on the meeting DS
  - Visualization of meetings by argumentation maps

Database Structure



Legend:
→ represents one-to-many relation
PK represents primary key constraint
FK represents foreign key constraint

Database Access

- Client-server application using "internal meaning" representation format for the communication
- Communication protocols
  - SOAP and WebServices
  - MRML (Multimedia Retrieval Markup Language) and HTTP
- Why?
  - Integration with other modalities
  - User queries do not rely on a specific database schema or a specific data model

Client-Server Application

- Client:
  - Graphical user-interface
  - Produces an “internal meaning” representation of the user query
  - Sends the user request to the server
- Server
  - Receives the client queries expressed in “internal” format
  - Identifies the DB elements that are necessary to satisfy the client queries
  - Generates a DB (SQL) query
  - Sends answer to the client in “internal” format

Example: Client Request and Server Answer in MRML

- **Query:**


```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<mrml>
    <query-step collection="2" dialogAct-id="qw" person-id="all" timeEnd=« 200"
    timeStart="0" topic-id="all" utterance-id="all" />
</mrml>
```

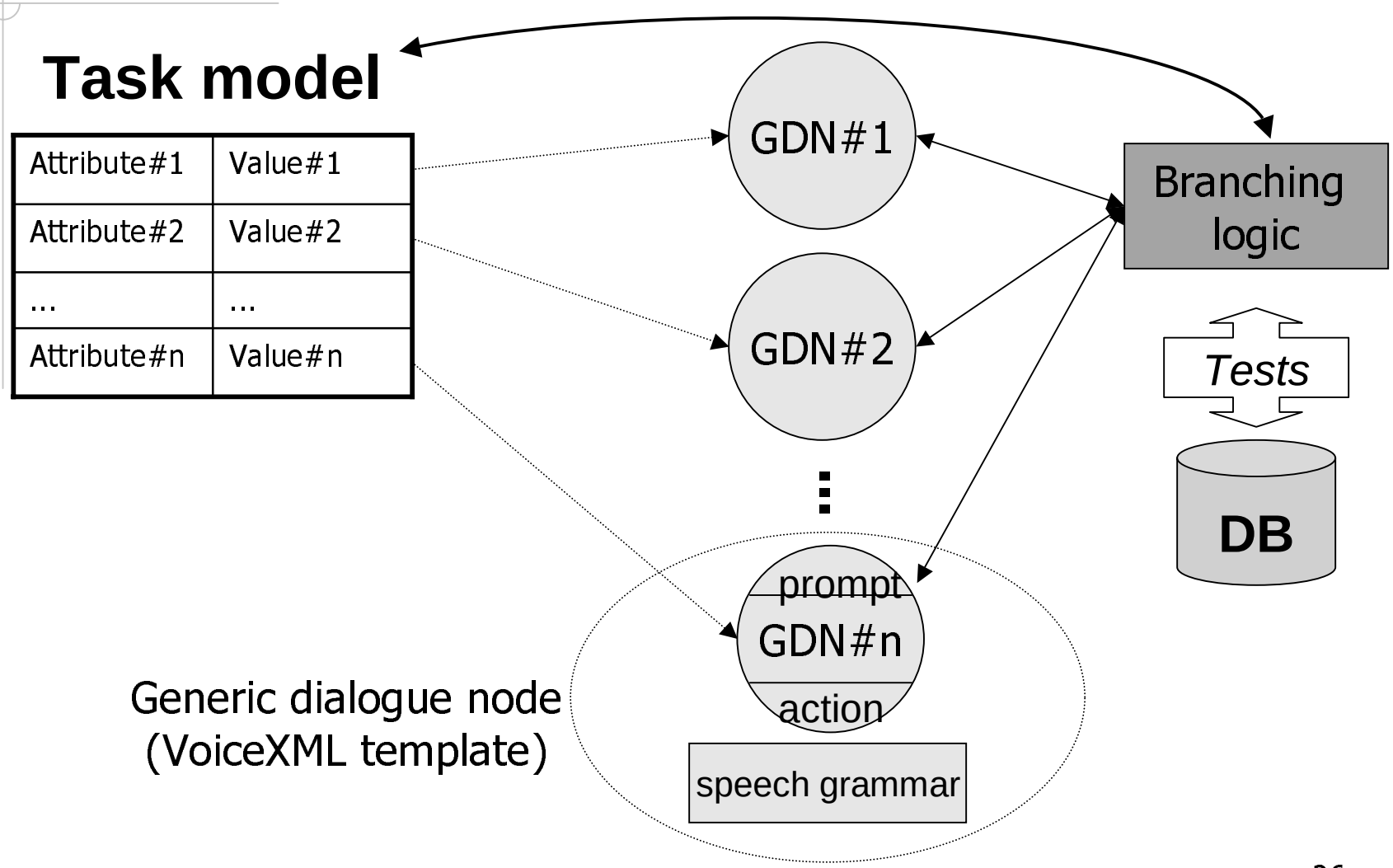

- **Answer:**


```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<mrml>-
  <query-result>
    <utterance dialogActDescription="Wh-Question" dialogActID="qw"
endIntervalTime="30.801" id="18" speakerID="Bmr001-c0"
startIntervalTime="30.241" topic="Part 1" transcription="what's this one here ?" />
    <utterance dialogActDescription="Wh-Question" dialogActID="qw^rt"
endIntervalTime="134.303" id="77" speakerID="Bmr001-c0"
startIntervalTime="133.253" topic="Part 1" transcription="what are we - why are we
writing it down ?" />
</query-result>
</mrml>
```

Multimodal Interactive Query Interface

- Goal:
  - Offer a user-interface which accepts a range of input modalities:
    - Spoken language input
    - Screen interaction by pointing
- Challenges
  - Multimodal queries plus multimedia data
  - To answer "who is <u>that</u> person?", one needs:
    - multimodal input (obviously)
    - multimedia metadata for the content of the meeting
    - cross-references between different layers of annotations (factual, thematic, rhetorical, physical)

Spoken Dialogue Manager - Abstract Structure



Example of Multimodal Query

User: “Find the video sequence where John is presenting the budget”

System: (retrieves the video file)

User: “Run it, please...”

System: (runs the video)

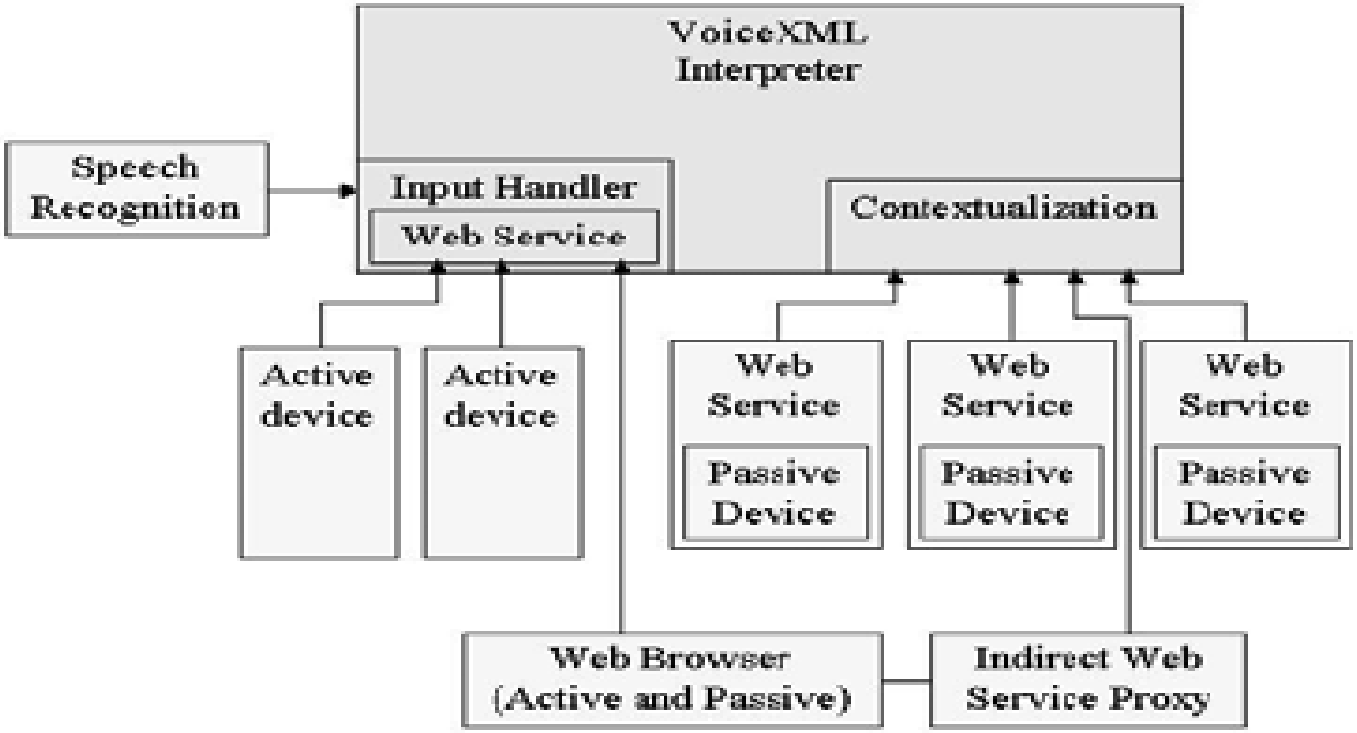
User: “Stop here!”

System: (stops the video)

User: “Show me this [pointing at the video] slide!”

System: (retrieves and runs the slideshow).

Overview of VoiceXML Interface



Legend:

- "active" devices = devices that are proactively triggered by the user
- "passive" devices = devices that need to be called in order to provide information about their state (such as a position tracking device)

Multimodal Interface - Ongoing Work

- Adapt and extend the existing voice-browsing standard VoiceXML for multimodal dialogue management
- Adapt and extend the Dialogue Manager with dynamic task selection functionalities
- Set up of a Wizard of Oz experiment for the specification and validation of the full Dialogue Model for the multimodal interaction with the meeting database

Conclusion – Work Progress

- Data preparation and study
  - annotation procedures and formats
  - *for: empirical research, evaluation, machine learning*
- Main project issues
  - shallow dialogue processing + argumentation structure annotation
  - interactive query interface
- Near future
  - develop and evaluate robust processing modules
  - user requirements + data model → evaluation metrics