Multilingual Restricted Domain QA System with Dialogue Management

Prof. Sivaji Bandyopadhyay
Srinivasa Rao Godavarthy
Partha Pakray

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JADAVPUR UNIVERSITY
KOLKATA-700032, INDIA
2007

Email : sivaji_cse_ju@yahoo.com , srinu_vasu504@hotmail.com , pathapakray@gmail.com

Introduction

- Restricted domain question answering (RDQA) deals with questions under a specific domain like railways or medicine.
- The possible questions are limited by the domain, therefore all the domain knowledge in the system are encoded to analyze questions and the system attempts to provide exact answer instead of listing a set of related documents.

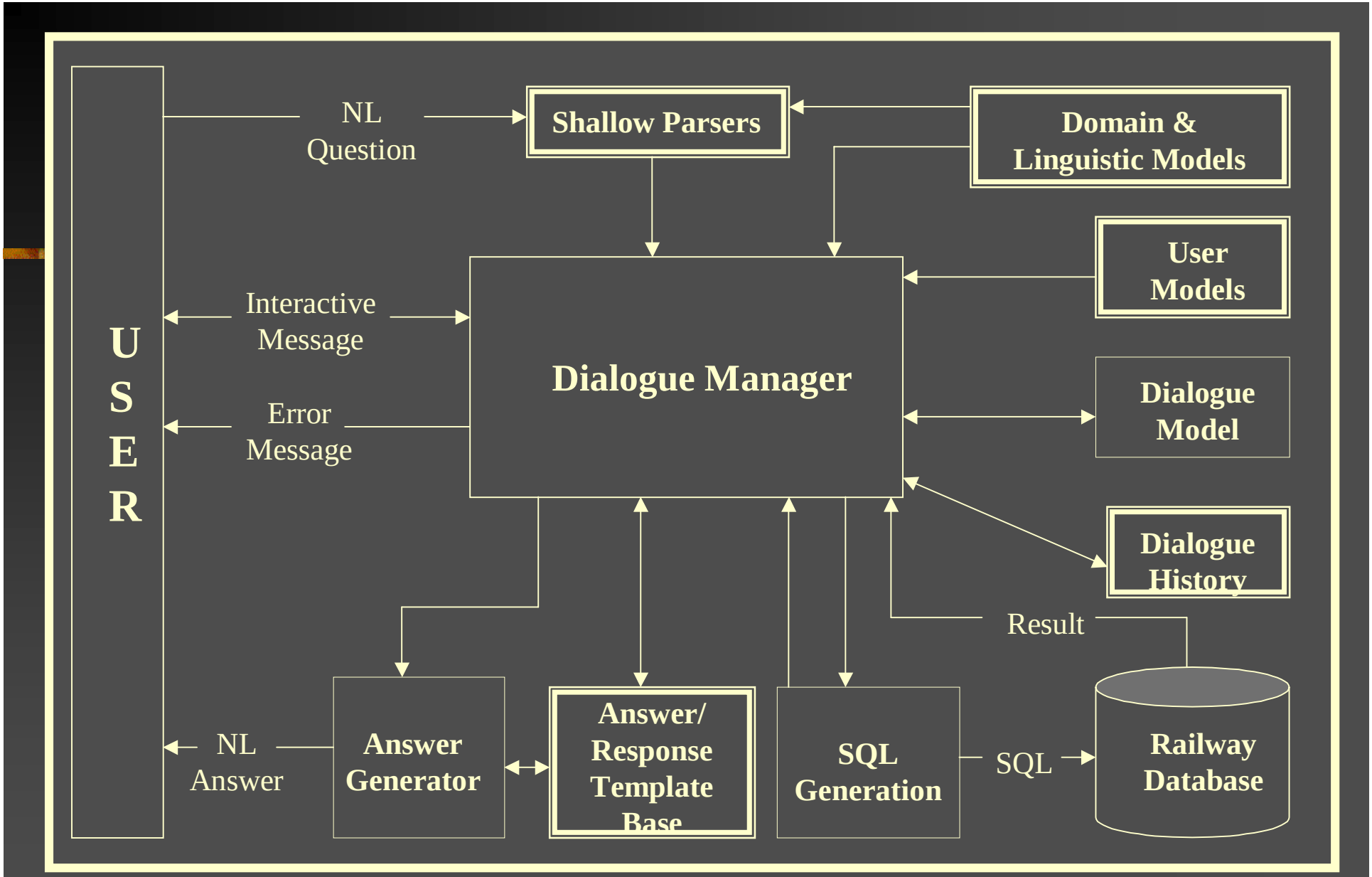
Introduction

System	Domain
Jupiter	Weather information
RailTel and ARISE	Rail travel and ticket reservation services

Examples Of RDQA Systems

Introduction

- The present QA system works in the Indian railway domain that accepts typed text inputs in Telugu and Bengali and provides text output also in the query language.
- It provides railway information service such as train arrival and departure time information, fare calculations, trains between important stations etc.
- The system generates SQL statement(s) from the input natural language (NL) question, executes the SQL query over a relational database and then provides user friendly NL answer.



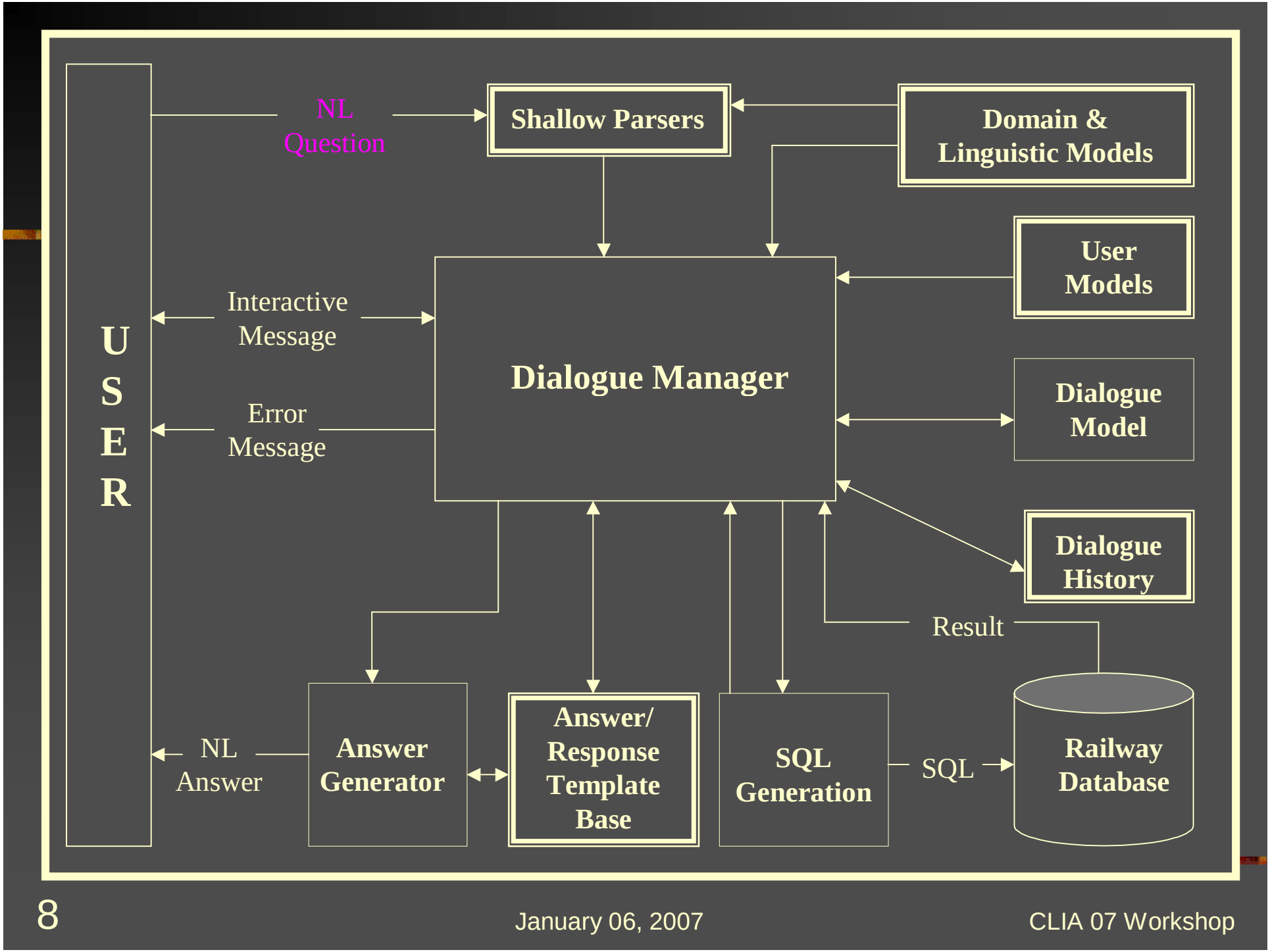
Multilingual Restricted Domain QA System

System Overview

- The language-specific components have been shown as double line boxes and language-neutral components have been shown in single line boxes.
- Initially our input NL query is analyzed by the shallow parser, which tags it morphosemantically using the domain ontology present in the domain and linguistic model.
- The tagged words are in the form of chunks like station name chunk, train name chunk, keyword chunk etc.
- The keyword chunks and sometimes information chunks are used to identify the query topic, i.e., query frame correctly without any ambiguity.

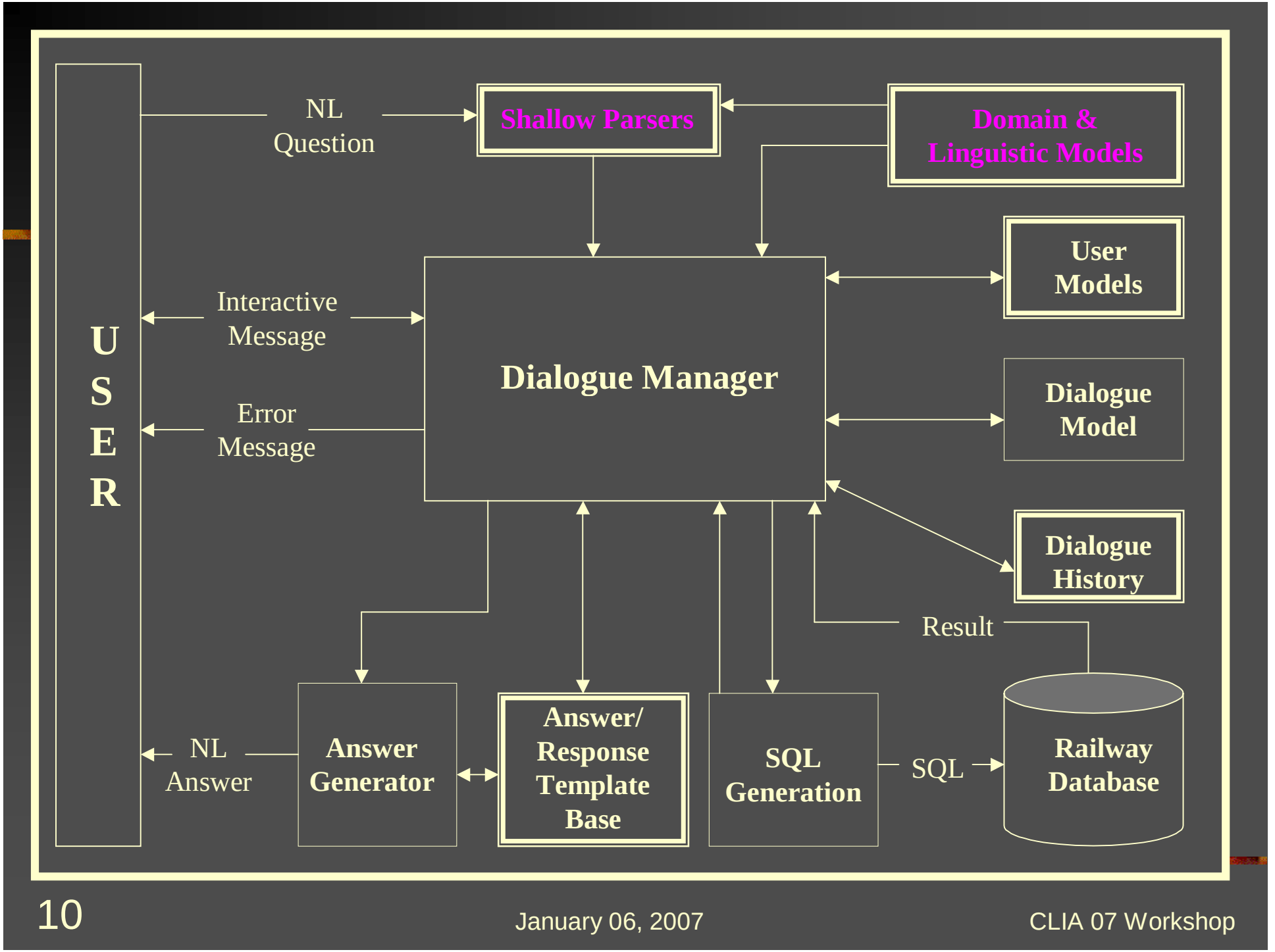
System Overview

- Dialogue Manager (DM) is used to obtain relevant missing information in the user query from the user model or the dialogue history or to generate a sub dialogue with the user or to generate a clarification request to the user in case of ambiguity in the user input.
- System gets all the required information directly or through sub dialogues from the user
- SQL generation procedure to generate all necessary SQL statement(s) using the information chunks.
- The SQL statements are used to retrieve the correct answer from the database. This result is forwarded to answer generator via DM to generate a natural language answer.


USER
NL Question
Shallow Parsers
Domain & Linguistic Models
User Models
Interactive Message
Error Message
Dialogue Manager
Dialogue Model
Dialogue History
Result
NL Answer
Answer Generator
Answer/ Response Template Base
SQL Generation
SQL
Railway Database

Natural Language Question

- *Secundrabad nundi tirupatiki tikkettu dhara enta*
- *Secundrabad theke tirupati porjonto bhara koto*
- [What is the fare from Secunderabad to Tirupati].


USER
NL Question
Shallow Parsers
Domain & Linguistic Models
User Models
Interactive Message
Error Message
Dialogue Manager
Dialogue Model
Dialogue History
Result
NL Answer
Answer Generator
Answer/ Response Template Base
SQL Generation
SQL
Railway Database

Domain Model

- The words that occur in the input query for Railway information system includes words describing train name, station name, reservation class, and date and/or period of journey or keywords etc.
- Hence the domain model contains look up tables with these words and the corresponding train name, station name, alias station names etc.

Domain Model

Query_Frame	Telugu Keyword	Bengali keyword
Fare	dhara	bhara
Seats	khaleelu	seat

Keyword Table

Ename	Telugu Name	Bengali Name
TIRUPATI	*tirupati*	*tirupati*

Station Table

Domain Model

Stations	City in Telugu	City in Bengali
Howrah, Sealdah, Calcutta	Kolkata	Kolkata

Alias Station Table

Train NO	Train Name In Telugu	Train Name In Bengali
2734	Narayanadri	Narayanadri

Train Name Table

Domain Model

- The domain knowledge consists of two kinds of rules :
- *Default value rules* supply default values for information not specified by the user.
- For ex : “I would like to go on the 6th”, the current month is taken as the default (or the next month if the 6th has already passed).
- *Interpretive rules* transform vague qualitative values given by the user into more precise quantitative values used by the system.
- For ex : “I want to go this morning” is transformed into “I want to go today between 6 am and 12 noon” .

Linguistic Model

- The linguistic model consists of a Telugu inflections table which contains both noun and verb inflections related to the domain to handle the inflected words in the query, and post-positions or route words table to identify station name under correct category.
- This model also maintains some set of *orthographic rules* (spelling rules) to identify the root word(s) from the surface level word(s).

Linguistic Model

Telugu Inflection	Bengali Inflection
ku	se
tundi	te
loo	ti

Inflection Table

Day	Name in Telugu	Name in Benali
1	Okati	Ekta
2	Rendu	Duto

Date Table

Number	Month in Telugu	Month in Bengali
1	January	January
2	February	February

Month Table

Shallow Parser

- Our shallow parsing task is a combination of
  - Morphosemantic tagging.
  - Chunking (Finding the fragments from the morphosemantic tags).
  - Query frame decision (Identify the query frame based on the keywords).

Morphosemantic Tagging

- The root words are extracted from the inflected words using the inflections tables and orthographic rules during morphological analysis of the input.
- The tagger uses morphosemantic tags like train name, station name, reservation class name etc. as well as the associated inflections and / or post-positions to tag the input query using the domain and linguistic model. For example

	Inflected word / Post-Position	Root Word
Telugu Word	tirupatiki	tirupati
Bengali Word	tirupati porjonto	tirupati

Morphosemantic Tagging

Secundrabad nundi tirupatiki tikkettu dhara enta
Secundrabad theke tirupati porjonto bhara koto

- [<Station_Name> Secunderabad </Station_Name> + <Post_Position> nundi </Post_Position>]
- [<Station_Name> Secunderabad </Station_Name> + <Post_Postion>theke</Post_Position>]
- [<Station_Name> Tirupati </Station_Name> + <Inflection> ki </Inflection>]
- [<Station_Name> Tirupati </Station_Name> + <Post_Postion>porjonto</Post_Position>]
- [<Keyword>*dhara*</ Keyword>]
- [<Keyword>b*hara*</ Keyword>]

Chunking

- Chunking is a process of finding semantically related non-overlapping groups of words from the morphosemantically tagged output.
- For example, <Station_Name> chunk is morphosemantically tagged like <Station_Name>/ <City_Name> + <Inflection> /<Post_Position> / <Route_Word>.
- The <Inflection> / <Post_Position> / <Route_Word> information helps to identify the source or the destination station name.

Chunking

Secundrabad nundi tirupatiki tikkettu dhara enta

Secundrabad theke tirupati porjonto bhara koto

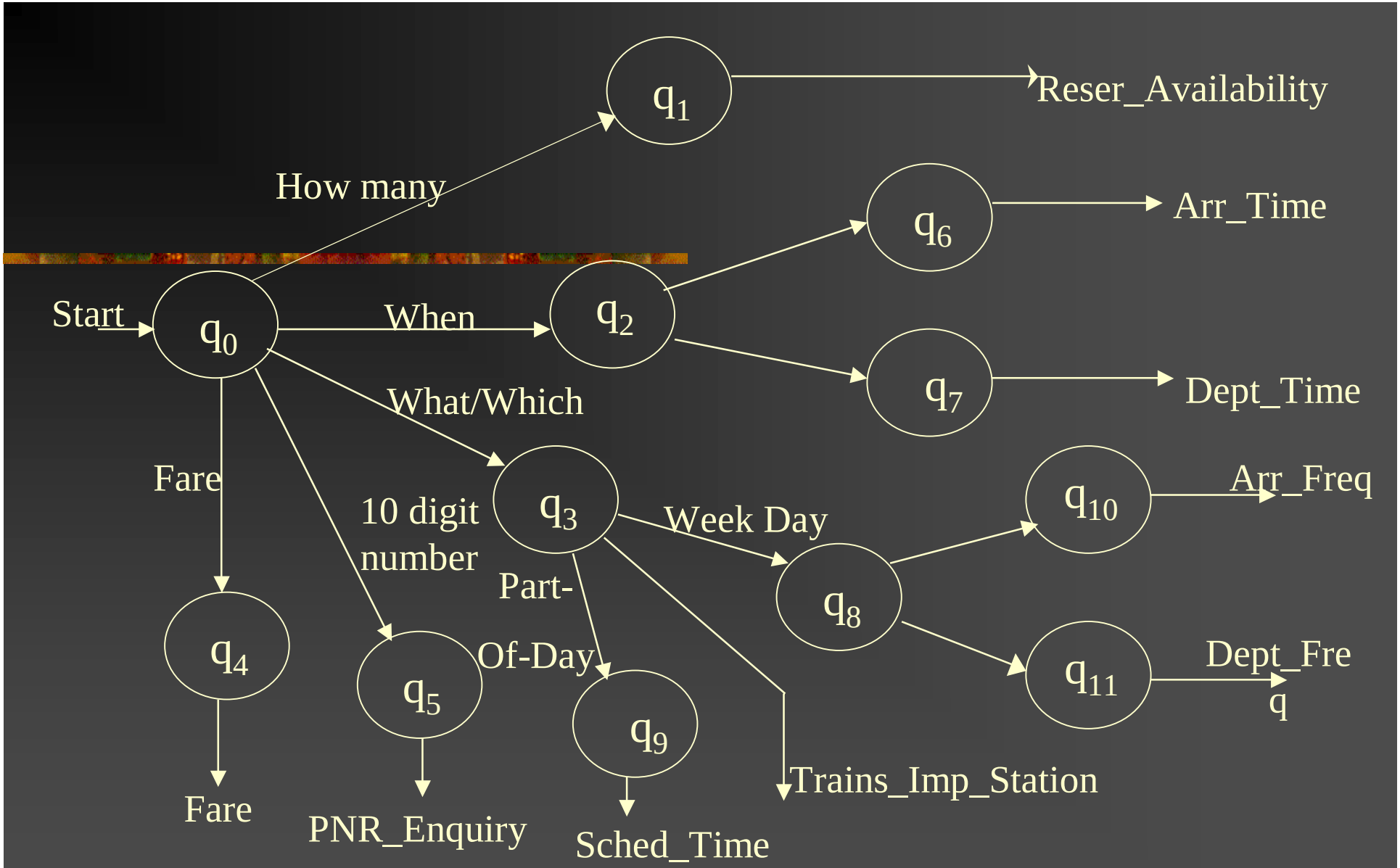
- (Station Name)[<Station_Name> Secunderabad </Station_Name> +<Post_Position> nundi </Post_Position>](/Station Name)
- (Station Name) [<Station_Name> Secunderabad </Station_Name> +<Post_Postion>theke</Post_Position>] (/Station Name)
- (Station Name)[<Station_Name> Tirupati </Station_Name> +<Inflection> ki </Inflection>] (/Station Name)
- (Station Name) [<Station_Name> Tirupati </Station_Name> + <Post_Postion>porjonto</Post_Position>] (/Station Name)
- (Keyword) [<Keyword>*dhara*</ Keyword>](/Keyword)
- (Keyword) [<Keyword>b*hara*</ Keyword>](/Keyword)

Chunking

- Secunderabad is Source Station name
- Tirupati is Destination Station name
- Secunderabad is Source Station name
- Tirupati is Destination Station name

Query Frame Decision

- A finite state automata (FSA) has been designed to identify the query frame without any ambiguity.
- There are some keywords that may identify either [Arr_Time] or [Dept_Time] query frames.
- To resolve this ambiguity, if it is a source station for the specific train, then the query is identified under [Dept_Time] query frame. Otherwise query will be under [Arr_Time] query frame.



FSA For The Identification Of The Query Frame

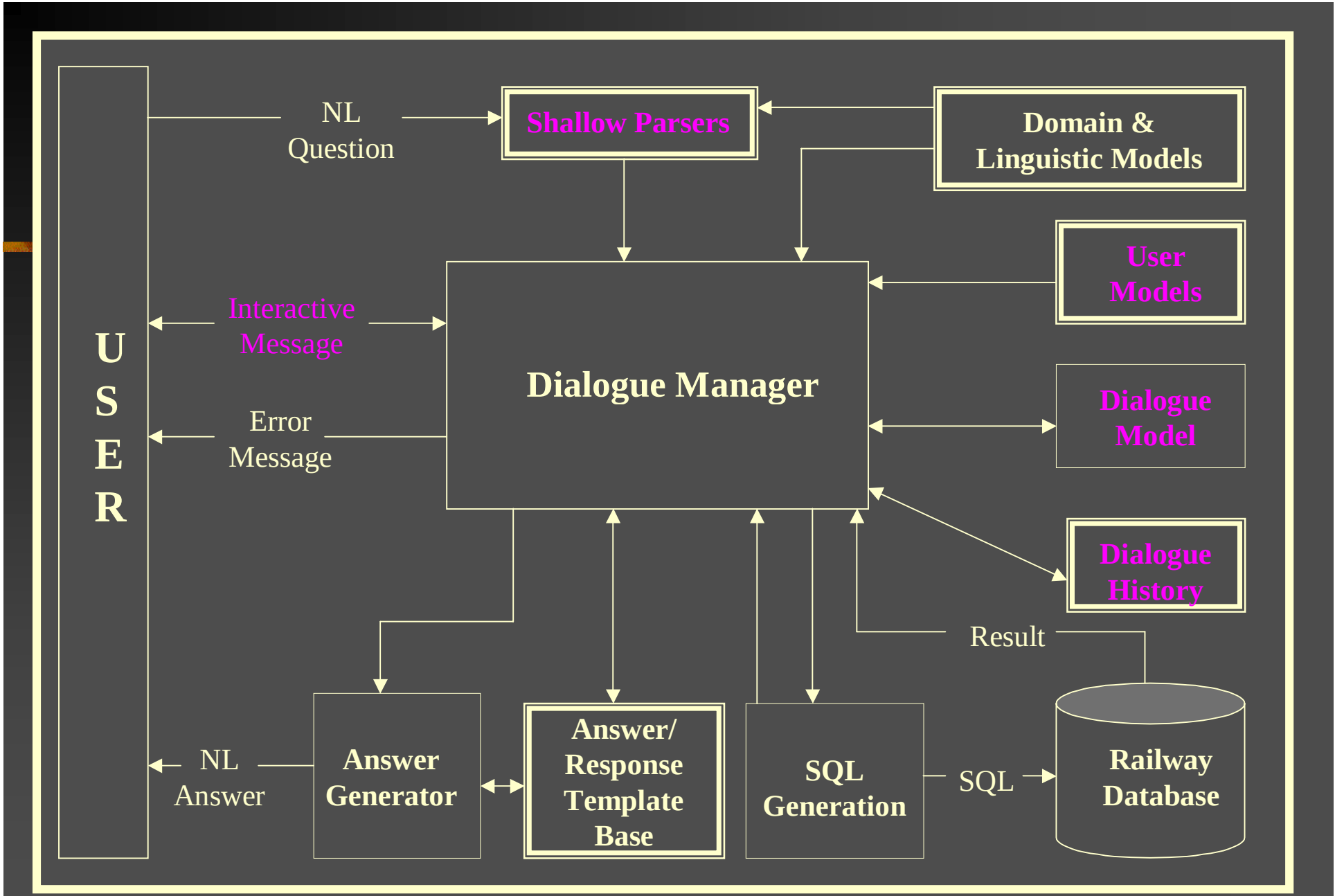
Query Frame Decision

Secundrabad nundi tirupatiki tikkettu dhara enta
Secundrabad theke tirupati porjonto bhara koto

- The Query Frame Belongs to Fare, the needed Chunks are
  - Source Station: Secunderabad
  - Destination Station: Tirupati
  - Train no/name: Naaraayanaadri Express (Guess by using user model)
  - Reservation Class: 3-tier A.C (Guess by using user model)

Query Frame	Needed Chunks
[Arr_Time]	Destination station, Train name/ number
[Dep_Time]	Source station, Train name/number
[Fare]	Source and Destination station/ Distance b/w stations, Train name/no & Class of journey
[Trains_Imp_Stations]	Source and Destination station
[Arr_Freq]	Weekday of journey, Source and Destination station
[Dep_Freq]	Weekday of journey, Source and Destination station
[Sched_Time]	Period/ time user wants to travel, Source and Destination station
[Reser_Availability]	Date of journey, Train name/number, Reservation class name
[PNR_Enquiry]	10 digit PNR number

Needed Chunks for each Query Frame


USER
NL Question
Shallow Parsers
Domain & Linguistic Models
User Models
Interactive Message
Error Message
Dialogue Manager
Dialogue Model
Dialogue History
Result
NL Answer
Answer Generator
Answer/ Response Template Base
SQL Generation
SQL
Railway Database

Dialogue Manager

- The main task of the dialogue manager (DM) component is to decide on the appropriate way to react to the user input.
- DM initially checks the dialogue history to obtain values for the relevant information chunks.
- If the values for the relevant information chunks are not available, the DM searches the user model to recommend to the user possible source or destination stations or the class of journey.

Dialogue Model

- Dialogue model consists of a Finite State Machine (FSM), in which each dialogue state is defined with the parameters:
  - Clarification Requests(CR),
  - Intended Dialogue Acts(IDACT),
  - Discourse Goals(DG).
- Each action results in a new dialogue state.

Dialogue Model

- The CR are generated by the system in case of ambiguity, in order to achieve grounding with the user.
- The IDACT are generated by the system to get the missing information for a particular query frame from the user.
- The DG keep track of the underlying activity related tasks still necessary to be fulfilled, which are identified during user interaction with system.

U1: secunderabad nundi tirupatiki tikkettu dhara enta

secundrabad theke tirupati porjonto bhara koto

[What is the fare from Secunderabad to Tirupati].

State 1 (After U1, but before S1):

CR: None.

IDACT: Train name and Reservation class [User didn't mention the train name and class name in which, he needs fare details].

DG: Fare calculation; [DM generates a sub dialogue, do you want in the Naaraayanaadri Express i.e guess by using user model

S1: [Meeku <Naaraayanaadri Express>loo kavaleenaa]

[Apni <Naaraayanaadri Express> e jete chan]

Do You Want In Narayanadri Express ?

Fare Query Sample Dialogue

U2: Yes

State 2 (After U2, but before S2):

CR: None.

IDACT: Reservation class.

DG: Fare calculation; [System asks for the reservation class is three-tier a.c in the Naaraayanaadri Express i.e guess by using user model].

S2: [Meeku Naaraayanaadri Expressloo <3-tier A.C> loo kavaleenaa]
[Apni Naaraayanaadri Express e <3-tier A.C> te jete chan]
Do You Want 3-Tier A.C in Narayandri Express?

U3 :Yes

Fare Query Sample Dialogue

State 3 (After U3, but before S3):

CR: None

IDACT: None

DG: Fare calculation [System Generates all possible SQL queries and retrieves specified answer in database].

State 4 (At S3): The Answer generator generates NL answer.

CR: None IDACT: None DG: None

S3: Naaraayanaadri expressloo secunderabad nundi tirupatiki tikkettu dhara three tier acloo 663 roopaayulu mariyuu tatkal loo 963 roopaayulu. Naaraayanaadri Expresse Secundrabad theke Tirupati jete tikiter bhara three tiere ac te 663 taka ebong tatkale 963 taka.

[Fare from Secundrabad to Tirupati by Narayanadri Express in Three Tier AC is Rs.663 and in Tatkal is Rs.963]

Fare Query Sample Dialogue

Dialogue History

- It represents the state of dialogue, which records focal information, i.e. what has been talked in the past and what is being talked in the present.
- It contains information about previous chunks and their types as well as other dialogue information like answers retrieved by the current SQL statement(s) and the answers for the previous queries in the dialogue.

Dialogue History

Source Station	Secunderabad	Train No	2734
Destination Station	Tirupati	Train Name	Narayanadri
Source City		Date	
Destination City		Period	
Distance	279	Time	
Reservation Class	SL	Day	
Query Frame	Fare	Answer	124

Language Dependent Dialogue History Table

User Model

- The Dialogue Manager identifies the incomplete data in the query and consults the User model to reason about the incomplete data.
- The system generates precision sub dialogues to authenticate these guesses with the user. If the user responds positively, the system generates the answer.
- In case of negative response, the system displays possible alternatives and the user selects the appropriate information.

User Model

From	Howrah	PNR NO	636-6081175
To	Secunderabad	Distance	1545
Train No	2703	Concession	
Train Name	Falaknama	Amount	449
Date	20-12-06	Dept Time	07:20
Boarding	Howrah	Coach	S14
Reservation Class	SL	No of Members	1

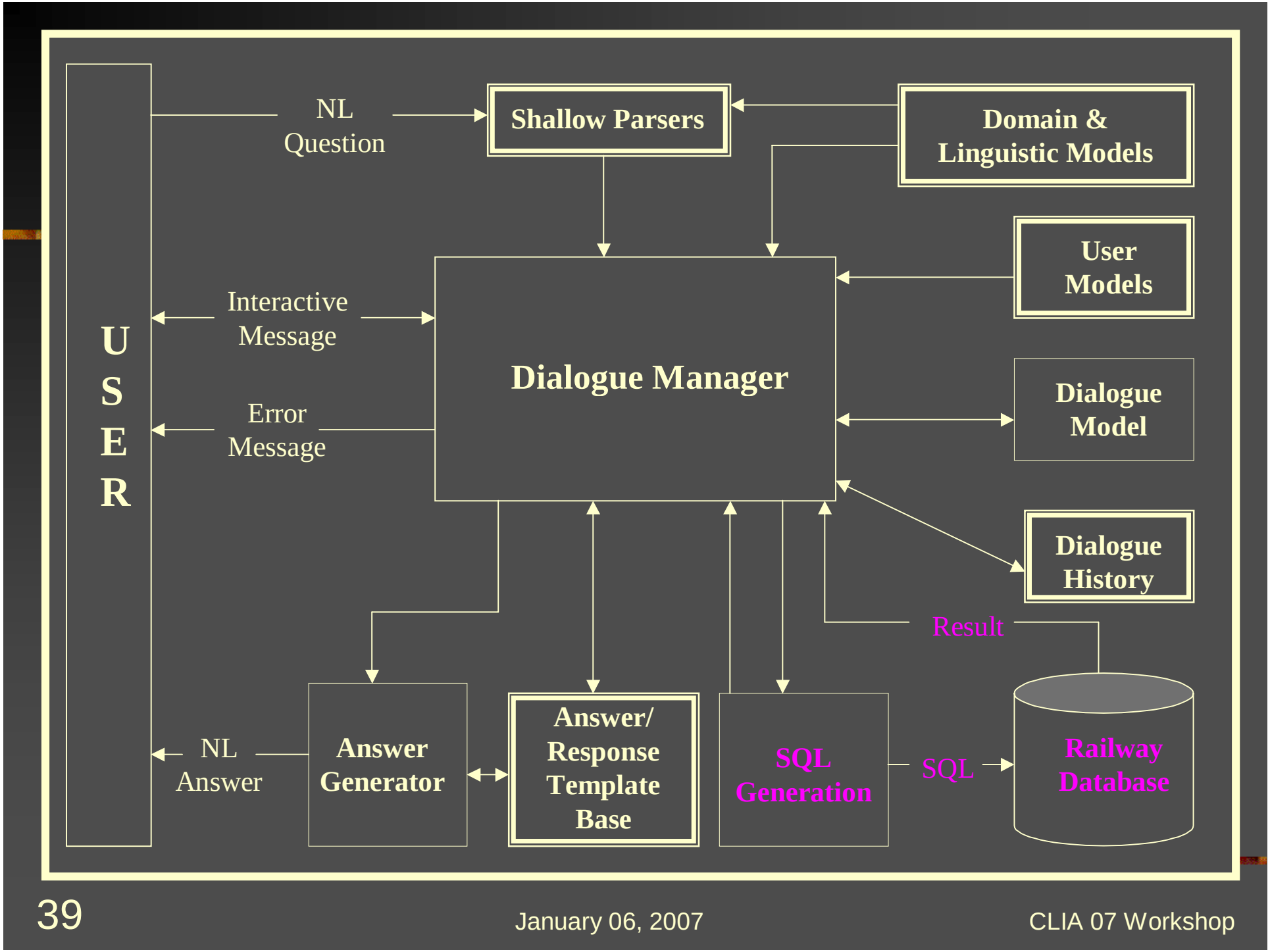
Language Dependent Booking Ticket Details Table

User Model

Secundrabad nundi tirupatiki tikkettu dhara enta
Secundrabad theke tirupati porjonto bhara koto

- [Meeku <Naaraayanaadri Express>loo kavaleenaa]
- [Apni <Naaraayanaadri Express> e jete chan]
- Do You Want in Naaraayanaadri Express ?

- [Meeku Naaraayanaadri Expressloo <3-tier A.C> loo kavaleenaa]
- [Apni Naaraayanaadri Express e <3-tier A.C> te jete chan]
- Do You Want 3-Tier A.C in Narayanadri Express?

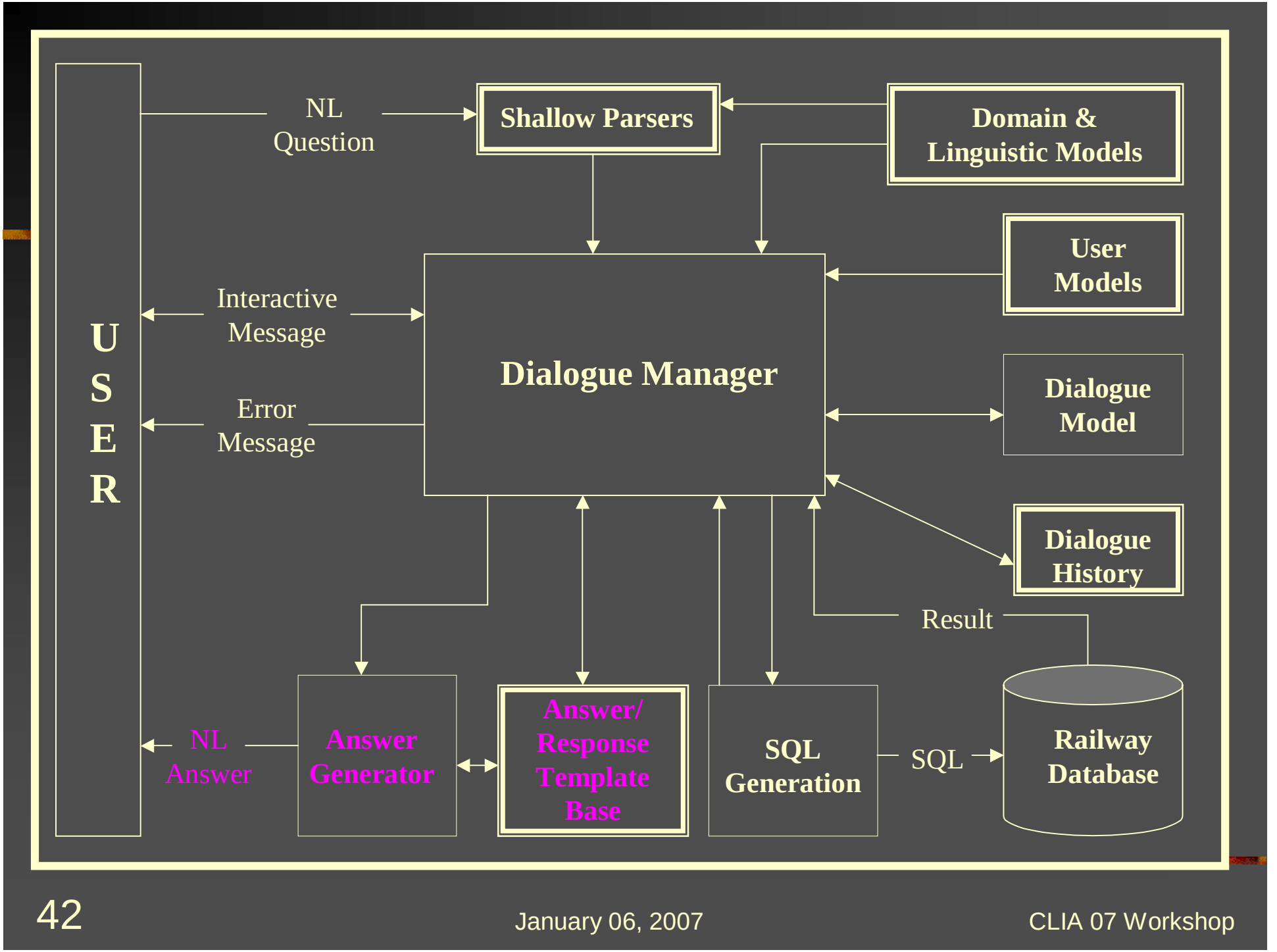

USER
NL Question
Shallow Parsers
Domain & Linguistic Models
User Models
Interactive Message
Dialogue Manager
Error Message
Dialogue Model
Dialogue History
Result
NL Answer
Answer Generator
Answer/ Response Template Base
SQL Generation
SQL
Railway Database

SQL Generation

- This module acquires the information chunks identified by the shallow parser via DM in the input query.
- If all the necessary chunks are present in the current query, this procedure generates all necessary SQL statement(s).
- This SQL statement(s) are used to retrieve the correct answer from the database.
- This result is forwarded to answer generator via DM to generate a natural language answer.

SQL Generation

- Select Route_No From GetRoute where Source_Station='Secunderabad Jn' and Destination_Staton='Tirupati'
- Select Distance from Schedule2764 where Station-name='Secunderabad Jn'
- Select Distance from Schedule2764 where Station-name='Tirupati'
- Select Fare-sl from Fare where dist_from<=737 and dist_to>=737


USER
NL Question
Shallow Parsers
Domain & Linguistic Models
User Models
Interactive Message
Error Message
Dialogue Manager
Dialogue Model
Dialogue History
Result
NL Answer
Answer Generator
Answer/ Response Template Base
SQL Generation
SQL
Railway Database

Answer/Response Template Base

- Each query frame for each language has its corresponding answer templates stored in the Answer/Response Template base. Each template consists of several slots.
- These slots are filled by the retrieved data from the database and the semantic frames generated from the query.
- In case of numeric data retrieved from the database, it is shown in English. Non-numeric data like train name, station name etc. are retrieved from the database in the query language and shown to the user.
- The filled in answer templates are sent to Dialogue Manager for forwarding it to the user.

Answer/Response Template Base

- <Train-Name>+loo <Source-Station> nundi <Destination-Station>+ki tikkettu dhara <Reservation-Class>+loo <Fare> roopaayulu +mariyuu tatkal loo + <Fare> roopaayulu
- <Train-Name> +e <Source-Station> theke <Destination-Station> + jete tikiter bhara <Reservation-Class>+ te <Fare> + taka ebong tatkale <Fare> taka.

Natural Language Answer

- *Naaraayanaadri expressloo secunderabad nundi tirupatiki tikkettu dhara three tier acloo 663 roopaayulu mariyuu tatkal loo 963 roopaayulu.*
- Naaraayanaadri Expresse Secundrabad theke Tirupati jete tikiter bhara three tiere ac te 663 taka ebong tatkale 963 taka.
- [Fare from Secundrabad to Tirupati by Naaraayanaadri Express in Three Tier AC is Rs.663 and in Tatkal is Rs.963]

U1: eppudu coromondal express vijayawadaku veltundi

kakhon coromodal express vijayawada pouchhabe

[When coromondal express goes to vijayawada]

This Query Frame belongs to arrival Query Frame, the needed chunks are

Train Name/Train No : Coromondal Express

Station Name : Vijayawada

S1: The system confirms about up/down journey of the train

U2: The user requested for upward train, running from Howrah to chennai via DM

S2: 10:30 gantalaku coromondal express vijayawadaku veltundi.

10:30 samaye coromondal express vijayawada pouchhabe

[At 10:30hrs coromondal express goes to vijayawada]

Example 1 : Arrival Query Frame

U1: Yeswanthpurhowrah Express eppudu bayuluterutundi

Kakhan Yeswanthpurhowrah Express charbe

[When the yaswanthpurhowrah express is starting]

This Query Frame belongs to Departure time Query Frame, the needed Chunks are

- Train name/Trainno : Yeswanthpurhowrah Express
- Station name : Yeswanthpur (By using Default value)

S1: Meeku yeswanthpur nundi kaavalenaa

Apni ki yeswanthpur theke aste chan

[Do you want from yeswanthpur]

U2: Yes [If no, System asks which station do you want]

S2: 19:30 gantalaku yeswanthpur nundi bayuluterutundi

19:30 samaye yeswanthpur theke chharbe

[It will leave yeswanthpur at 19:30 hrs]

Example 2 :Departure Time Query Frame

U1: naaku falaknama expressloo howrah nundi secundrabadku vachaa somavaramu reservation kaavalenu
Ami falaknama expresse howrah theke secundrabad porjonto porer sombar reservation chai [I want Reservation From Howrah to Secundrabad on next Monday]

This query frame belongs to reservation-availability, the needed chunks are

- Source Station: Howrah
- Destination Station: Secundrabad
- Date of Journey: Next Monday (By Using Interpretive rules)
- Train no/Train name: Falaknama Express (Guess By using user model)
- Reservation Class: 3-tier A.C (Guess By using user model)

Example 3: Reservation-Availability Query Frame

S1: Meeku falaknama expresslo kaavalenaa

Apni ki falaknama expresse jete chan

[Do you want in Falaknama express]

U2: Yes [If no, the system shows all possible trains & user select one]

S2: Meeku falaknama expresslo 3-tier A.Clo kaavalenaa

Apni ki falaknama expresse 3-tier A.Cte jete chan

[Do you want 3-tier A.C in Falaknama Express]

U3: Yes [If no, the system shows all possible journey classes in the selected train and user selects the appropriate one]

S3: Displays number of seats available in selected train and selected class of journey for next Monday.

Example 3: Reservation-Availability Query Frame

U1: Falaknama expressloo tikkettu dhara enta
Falaknama expresse tikiter bhara koto
[What is the fare in the falaknama express]

This Query Frame belongs to fare query, the needed chunks are

- Train name/Train no : Falaknama Express
- Source Station: Howrah (Guess by using user model)
- Destination Station: Secunderabad(Guess by user model)
- Reservation Class: 3-tier A.C (Guess by using user model)

S1: Meeku howrah nundi kaavalenaa
Apni ki howrah theke jete chan
[Do you want from howrah]
U2: Yes[If no system asks which station do you want]

Example 4: Fare Query Frame

S2: Meeku secunderabad ku kaavalenaa

Apni ki secunderabad jete chan

[Do you want to go to secunderabad]

U3: Yes [If no system asks which station do you want]

S3: Meeku falaknama expressloo three-tier a.cloo kaavalenaa

Apni ki falaknama expresse three-tier ac te jete chan

[Do you want 3-tier a.c in falaknama express]

U4: Yes [If no system asks which class of journey do you want]

S4: Howrah nundi secunderabadku tikkettu dhara three-tier a.cloo 1145 roopaayulu mariyu tatkalloo 1445 roopaayulu

Howrah theke secunderabad porjonto falaknamaexpreeese tikitter bhara holo three tire AC TE RS.1145 ebong Tatkale 1445

[Fare from Howrah to Secundrabad by Falaknama Express in Three Tier AC is Rs.1145 and in Tatkal is Rs.1445]

Example 4: Fare Query Frame

U1: Neenu Secunderabad nundi chennai vellalanukuntunnanu
Ami Secunderabad theke chennai jete chai.
[I want to travel from secunderabad to chennai]

This Query Frame belongs to departure frequency query, the needed chunks are

- Source Station: Secunderabad
- Destination Station: Chennai
- Week Day Of Journey: Next Monday (System ask Question)

Example 5: Departure Frequency Query Frame

S1: Meeru Secunderabad nundi chennaiku e roojuna vellalanukuntunnaruu

kon din apni Secunderabad theke chennai porjonto jete chan

[Which Day you want travel from secunderabad to chennai]

U2: vaccha somavaaramu porer sombar [Next Monday]

S2: Meeru Secunderabad nundi chennaiku decemburu eeraviiduna e samayamunaa vellalanukuntunnaru

kon somoy pochise decemember jete chan

[what time do you go to on decemember 25th]

U3: sayamtram sondhar somoy [Evening]

S3: It displays all possible trains from secunderabad to chennai on next Monday at evening

Example 5: Departure Frequency Query Frame

Conclusion and Future work

- The question answering system separates dialogue control from the application logic, provides a portable dialogue manger and a user-friendly interface.
- We are developing more robust shallow parser and the modules for the remaining query frames with dialogue management.
- The system needs to be upgraded so that a user can query for railway information over phone. The speech input can be converted to textual query. This textual query can be input of our system and the textual output can be converted to speech again to answer the user.

THANK YOU