Rapid Dialogue Prototyping

- Rapid dialog prototyping is a significant part of the dialog model design.
- It is used to identify and model specificities of human-machine communication in real-life conditions.
- It is an iterative process.
- It must be flexible, allowing us to rapidly experiment with different dialog strategies.
- The result of the dialog prototyping process is a stabilized dialog model represented as a finite state automaton.

Dialog-repair sub-system

- The objective of the practical session was to design a simple dialog model to interact with the user.
- The interaction is reduced to the processing of the following simple question to the user:

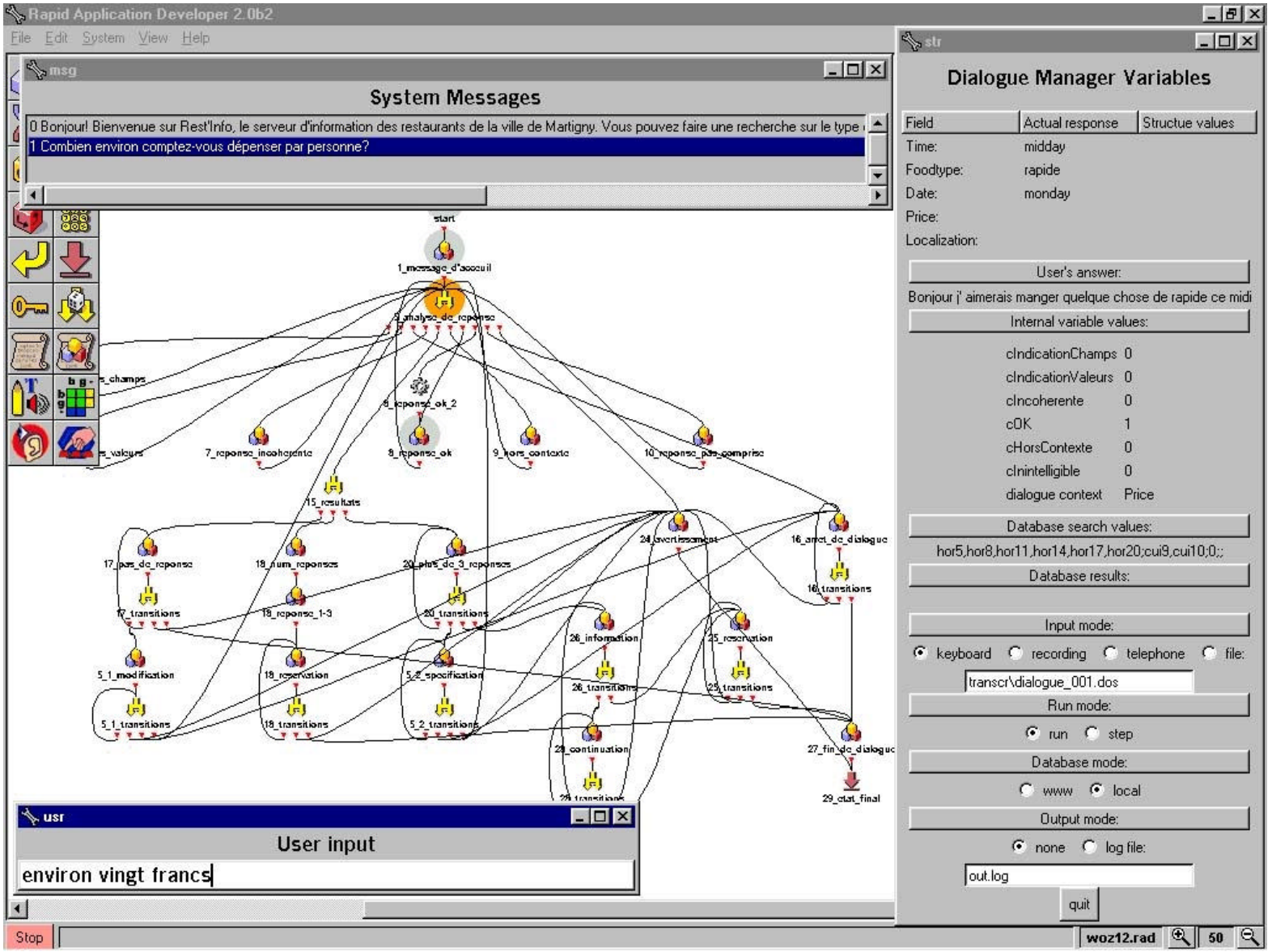
  What is your favorite color?

- Interestingly enough, the difficulties arising in such a simple situation are quite representative of the ones occurring in more complex dialog tasks.

What is your favorite color?

- Yellow!
- Well, I like violet blue purple.
- ahh hem [break] me I like I like rather colors hem hot colors.
- What! I haven’t understood, what did you say?
- Sorry! Why you want to know that?
- Yeah, ok, what are the choice?
- Oh! That! But me, I wanted to go to the movies.
- ahhh [break] I never eat oranges!
- humm.

Rapid dialog prototyping: the CSLU toolkit



Demo 1:
Simple model

- **Features**: User's answers are categorized into 5 groups: OK, Repeat, Explain, Out-of-scope, Misunderstood.
- For each category, the system carry out some predefined action (message) in order to help the user.

- **Limitation**: There is no account for dialog history; in a more sophisticated approach, the system should be able to adapt its behavior according to the user mental representation.

Demo 2:
Enhanced model, improvement:

- Bring variability to the dialog control.
- If the same repair-state is consecutively reached twice, reformulation is used in order to bring some new information to he user.
- User’s responses are not limited to strict commands – a garbage grammar model is used.
- Dialog repair is integrated as a subdialog in order to become a reusable component and to overcome some of the weaknesses of the finite-state model.

Conclusion

- Designing an efficient dialogue prototyping tool
- Architecture: finite state automaton
- Implementing this model in a example
- Learning to handle the typical problems in such a dialogue model