Abstract Models for Dialogue Protocols (Extended Abstract)

Raquel Fernández
Department of Computer Science
King's College London
`raquel@dcs.kcl.ac.uk`

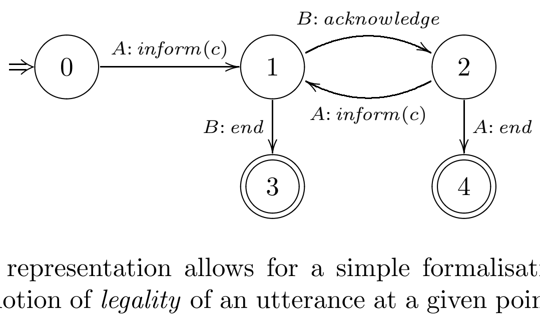
Ulle Endriss
Department of Computing
Imperial College London
`ue@doc.ic.ac.uk`

Introduction. If we look at a corpus of real human-human dialogues, we find evidence of frequently reoccurring sequences of utterance types. For instance, questions are followed by answers and proposals are usually either accepted, rejected, or countered. These *interaction patterns* have inspired a line of research whose object of description is, broadly speaking, the rule-governed behaviour exhibited by dialogue interaction. Communication patterns are usually modelled by means of *conventional* protocols, i.e. public conventions which specify the range of possible follow-ups available to the participating agents.

In this paper, we examine a variety of dialogue protocols, taking inspiration from two fields: natural language dialogue modelling and multiagent systems. In communicative interaction, one can identify different features that may increase the complexity of the dialogue structure. This motivates a hierarchy of abstract models for protocols that takes as a starting point protocols based on deterministic finite automata. From there, we proceed by looking at particular examples that justify either an enrichment or a restriction of the initial model.

An extended version of this paper is available as a technical report [5].

Protocols as finite automata. Deterministic finite automata (DFAs) have been widely used to represent communication protocols, in particular in the area of multiagent systems. Pitt and Mamdani [8] give several examples for such automata-based protocols. One of them, the *continuous update protocol*, specifies a class of dialogues between two agents A and B where A continuously updates B on the value of some proposition. The following diagram provides an intuitive description of this protocol:



This representation allows for a simple formalisation of the notion of *legality* of an utterance at a given point in a dialogue. Given the current dialogue state q, an utterance u constitutes a legal continuation of the dialogue iff there exists a state q' such that the automaton's transition function maps the pair (q, u) to q'. This type of protocols will provide the starting point for our proposed classification of communication protocols.

Protocols that allow for subdialogues. DFA-based protocols have also been successfully used in natural language interaction. However, some very common features of natural language dialogue cannot be captured by a DFA. The following example shows a dialogue where several question-answer sequences are embedded:

A :	(1) Who should we invite?	$[Q_1]$
B :	(2) Should we invite Bill?	$[Q_2]$
A :	(3) Which Bill?	$[Q_3]$
B :	(4) Jack's brother.	$[A_3]$
A :	(5) Oh, yes.	$[A_2]$
B :	(6) OK, then we should invite Gill as well.	$[A_1]$

The presence of embedded subdialogues creates a structure that calls for an enrichment of the DFA-based model. This can be modelled by adding a stack to a DFA. In the example above, questions would get pushed onto the stack, to be then popped by their respective answers. The machine model of a DFA together with a stack corresponds to a *pushdown automaton*. An example of a structuring mechanism able to handle this kind of phenomena is Ginzburg's QUD (*questions under discussion*) [6].

Protocols with memory. DFAs are abstract machines with a limited amount of memory (encoded by the states of the automaton). Adding a (finite) stack amounts to enriching the automaton with an unlimited memory component. Modelling this memory as a stack is just one of many options. Besides stacks, we may consider a variety of *abstract data types* (ADTs) such as, for instance, sets or lists. Every ADT comes with a set of basic operations ($push(x)$ and pop in the case of a stack) and functions (top to return the top element on a stack, for example).[1] In the sequel, we are going to discuss several examples that motivate different choices for ADTs as memory components on top of our basic DFA-based model.

Protocols with a stack of sets. As some authors have noticed [6, 2], when successive queries are asked within a single turn, a protocol with a simple stack is not always correct. This is illustrated by the following example (adapted from [2]):

A :	(1) Where were you on the 15th?	$[Q_1]$
A :	(2) Did you talk to him after the incident?	$[Q_2]$
B :	(3) I didn't talk to anyone.	$[A_2]$
B :	(4) I was at home.	$[A_1]$
B :	(3') I was at home.	$[A_1]$
B :	(4') I didn't talk to anyone.	$[A_2]$

Dialogues as the one above show that when two or more questions are uttered in succession by the same speaker,

[1] A recent formalisation of a complex protocol with memory in dynamic logic (DL), namely a protocol for inquiry-oriented dialogues based on Ginzburg's *dialogue gameboard* [4], suggests that the usual DL program constructors provide an adequate language to combine these basic operations.

the respondent is sometimes allowed to answer them in any order. In terms of our hierarchy of protocols with memory, such an architecture can be modelled using a DFA together with a finite *stack of sets*. The questions under discussion that currently have maximal conversational precedence are those in the top set on the stack.

An interesting issue is what causes a question to be either inserted into the maximal set of the stack or, alternatively, to be pushed on top of the stack of sets. As Ginzburg [6] and Asher [2] argue, this depends on the relation of that question to those that are already in the maximal set on the stack. For instance, in the previous example the discourse relation of *Coordination* is said to hold between questions (1) and (2), while in the earlier example the relation that holds between the questions is that of *Query-elaboration*. This shows that, in order to determine the legality of a dialogue move with respect to a given protocol, one also has to take into account complex relations between the utterances occurring in a dialogue. Integrating this kind of conditions with our abstract model of protocols with memory is one of issues we are currently investigating further.

Protocols with a blackboard. Our next example is inspired by work on argumentation in discourse modelling. Argumentation-based protocols have recently been used to model different types of dialogues between software agents [1]. Central to this approach is the notion of a so-called *commitment store* [7]. For example, *asserting* an argument amounts to placing that argument into one's commitment store. A *retract* move would then be considered legal only if the corresponding argument can be found in the agent's commitment store (and would itself cause the respective argument to be deleted again).

This kind of "blackboard architecture" may be modelled in terms of a DFA-based protocol together with a (finite) *set* (or possibly one set for each agent). Any utterances that may affect the legality of utterances later on in a dialogue would be stored in this set.

Protocols with a list. Systems providing access to the dialogue history explicitly in order to check the legality of an utterance may be modelled as DFA-based protocols together with a finite *list* (by appending utterances to the end of the list as they occur in the dialogue).

This is the most powerful protocol model we have discussed, because, given the (full) dialogue history, it should—in principle—always be possible to specify *any* conditions pertaining to the legality of an utterance. In fact, this is precisely the thesis underlying the conventionalist approach to communication protocols (in multiagent systems research): what is legal may only depend on publicly observable facts.

Shallow protocols. So far we have concentrated on enriching the basic model of DFA-based protocols to cater for a variety of complex dialogue phenomena. Where such phenomena are not present, we may usefully restrict the model rather than extend it. Recently, a class of so-called *shallow protocols* has been introduced in the context of multiagent systems [3]. A shallow protocol is a protocol where the legality of an utterance can be determined on the sole basis of the previous utterance in the dialogue. For example, to express that any proposal by agent A must be followed by either an acceptance, a rejection, or a counter proposal by agent B, we may use the following shallow rule:[2]

$$\textit{A: propose} \;\rightarrow\; \bigcirc\,(\textit{B: accept} \;\vee\; \textit{B: reject} \;\vee\; \textit{B: counter})$$

While such a simplistic approach will have little relevance to natural language dialogue modelling, it can be of great interest in the area of multiagent systems. As shown in [3], it is possible to check *a priori* whether a given agent will behave in conformance to a given shallow protocol by inspecting the agent's specification, rather than just observing an actual dialogue at runtime.

Conclusion. We have reviewed a variety of interesting features of dialogue as they occur either in natural language interaction or in the context of multiagent systems. These features have given rise to a number of different abstract models for dialogue protocols. These protocols are based on the machine model of a deterministic finite automaton, which we have further enriched with memory components modelled as different abstract data types. In one case, we have also seen that a restriction of the basic model can have useful applications.

We hope to have been able to point out interesting connections between issues in dialogue modelling on the one hand, and well-known machine models from the theory of computation on the other.

Acknowledgements. The work of the first and the second author is supported by UK ESRC grant RES-000-23-0065 and EC grant IST-2001-32530, respectively. Thanks to Jonathan Ginzburg and Nicolas Maudet for useful comments and encouragement, and to the referees of the Tbilisi Symposium 2003 for their helpful suggestions.

References

[1] L. Amgoud, N. Maudet, and S. Parsons. Modelling Dialogues using Argumentation. In *Proceedings of 4th International Conference on MultiAgent Systems (ICMAS-2000)*. IEEE, 2000.

[2] N. Asher. Varieties of Discourse Structure in Dialogue. In *Proceedings of the Second International Workshop on Dialogue (Twendial)*, 1998.

[3] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Protocol Conformance for Logic-based Agents. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, 2003. To appear.

[4] R. Fernández. A Dynamic Logic Formalisation of Inquiry-Oriented Dialogues. In *Proceedings of the 6th CLUK Colloquium*, Edinburgh, 2003.

[5] R. Fernández and U. Endriss. Abstract Models for Dialogue Protocols: A Preliminary Report. Technical Report TR-03-03, Department of Computer Science, King's College London, July 2003.

[6] J. Ginzburg. A Semantics for Interaction in Dialogue. Forthcoming for CSLI Publications. Draft chapters are available from http://www.dcs.kcl.ac.uk/staff/ginzburg/.

[7] C. L. Hamblin. *Fallacies*. Methuen, London, 1970.

[8] J. Pitt and A. Mamdani. A Protocol-based Semantics for an Agent Communication Language. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)*. Morgan Kaufmann Publishers, 1999.

[2]The *next-operator* $\bigcirc$ (familiar from linear temporal logic) refers to the next turn in the dialogue.