Protocol Synthesis with Dialogue Structure Theory

Jarred McGinnis
j.p.mcginnis@sms.ed.ac.uk

David Robertson
dr@inf.ed.ac.uk

Chris Walton
cdw@inf.ed.ac.uk

Centre for Intelligent Systems and Applications
University of Edinburgh
11 Crichton Street
EH8 9LE Edinburgh, Scotland

ABSTRACT

Inspired by computational linguistic approaches to annotate the structures that occur in human dialogue, this paper describes a technique which encodes these structures as transformations applied to a protocol language. Agents can have a controlled mechanism to synthesise and communicate their interaction protocol during their participation in a multiagent system. This is in contrast to the approaches where agents must subscribe to a fixed protocol and relinquish control over an interaction that may not satisfy the agent's dialogical needs.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and coordination, Languages and structures, Multiagent systems*

General Terms

Design, Languages, Theory

Keywords

Interaction Protocols, Dialogue Structure, Distributed Protocols

Research into agent communication is producing increasingly more robust models. Much of this research has turned to other disciplines for inspiration. Philosophy and Linguistics have a several thousand year head start in reflecting upon the nature of communication. These thinkers are concerned with human communication in particular, but insights and models they have developed are applicable to the study of agent communication. BDI-logics, speech acts, social commitment and argumentation have originated in the works of philosophers and linguists.

The societal view of communication is an attractive approach for multiagent system researchers recognising its importance for creating reliable communication. A society has rules which govern the behaviour of the agents, constraining the members to perform in accordance with a set of implicit or explicit *protocols*. Participants in the society willingly sacrifice autonomy and submit to these protocols in order to gain a measure of utility or to accomplish a goal of more value than the independence lost.

Table 2: An Abridged Abstract Syntax

θ	$\in$	Agent Definition	::=	$\textbf{agent}(R,Id)$
op	$\in$	Operation	::=	θ
		(Precedence)	|	(op)
		(Send)	|	$M \Rightarrow \theta$
		(Receive)	|	$M \Leftarrow \theta$
		(Sequence)	|	$op1$ **then** $op2$
		(Choice)	|	$op1$ **or** $op2$
		(Closed)	|	$c(op2)$
		(Failed)	|	$f(op2)$

Protocols are not only created with respect to societal conventions but the act of communication itself has conventions to which speakers adhere. Relationships exist between messages regardless of the particular domain with which the messages are concerned. A question implies the anticipation of the eventual occurrence of an answer, even if the reply is a shrug of the shoulders. It is these generalised patterns which exist in human communication that we have adopted for our purposes. The result is the creation of a means to synthesise a protocol which can be reproduce the reliable communication produced by other protocolled approaches without being fixed to a static protocol.

The details of the protocol framework are described in [3]. The use of a distributed protocol framework is essential if agents are allowed to make transformations to their interaction protocol during their dialogue. Not only is it essential to provide the access to the protocol to be modified but also to ensure it is done in a transparent and verifiable way. An abridgement of the protocol language's syntax is shown in Table 2. An agent's identity is represented as θ which consists of a Role, R, and a unique identifer, *Id*. A role is a way of defining communicative activity for a group of agents rather than individuals. This agent definition is expanded by a number of operations.

Operations can be classified in three ways: actions, control flow, and conditionals. Actions are the sending or receiving of messages, or the adoption of a role. Control flow operations temporally order the individual actions. Actions can be put in sequence (one action must occur before the other),

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

Copyright 2005 ACM 1-59593-094-9/05/0007 ...$5.00.

Table 1: The Vocabulary of Transformations

Before a Message is Sent			Before a Message is Received		
$M1 \Rightarrow \theta$	$\xrightarrow{response(M1,M2)}$	$M1 \Rightarrow \theta$ **then** $M2 \Leftarrow \theta$	$M1 \Leftarrow \theta$	$\xrightarrow{continuation(M1,M2)}$	$M1 \Leftarrow \theta$ **then** $M2 \Leftarrow \theta$
$M1 \Rightarrow \theta$	$\xrightarrow{continuation(M1,R2)}$	$M1 \Rightarrow \theta$ **then** $M2 \Rightarrow \theta$	$M1 \Leftarrow \theta$	$\xrightarrow{response(M1,M2)}$	$M1 \Leftarrow \theta$ **then** $M2 \Rightarrow \theta$
$M1 \Rightarrow \theta$	$\xrightarrow{counter(M1,M2)}$	$M1 \Rightarrow \theta$ **or** $M2 \Rightarrow \theta$	$M1 \Leftarrow \theta$	$\xrightarrow{counter(M1,M2)}$	$M1 \Leftarrow \theta$ **or** $M2 \Leftarrow \theta$
Upon Failure of a Message			Upon the Reception of a Message		
$f(M1 \Rightarrow \theta)$	$\xrightarrow{correction(M1,M2)}$	$f(M1 \Rightarrow \theta)$ **then** $M2 \Rightarrow \theta$	$c(M1 \Leftarrow \theta)$	$\xrightarrow{clarification(M1,M2)}$	$c(M1 \Leftarrow \theta)$ **then** $M2 \Rightarrow \theta$
$f(M1 \Leftarrow \theta)$	$\xrightarrow{correction(M1,M2)}$	$f(M1 \Leftarrow \theta)$ **then** $M2 \Rightarrow \theta$			

or given a choice point (one and only one action should occur before any further action). The '⇒' and '⇐' denote messages, *M*, being sent and received. Besides the message the agents also communicate the state of the current dialogue. This is a marked up version of the protocol which shows the past messages communicated, the current message, and future possible paths the dialogue can take.

The engineering requirements for implementing this protocol language are relatively light-weight. Agents are required to share a dialogical framework. An agent must be able to understand the protocol, the dialogue state, and its role within the protocol. Agents need to be able to identify the agent clause which pertains to its function within the protocol and establish what actions it must take to continue the dialogue. This includes the ability to update the dialogue state to reflect any actions it chooses to perform. The are several examples of frameworks which use this protocol language.

This protocol language is well suited for our purposes. By distributing the protocol during the interaction, the agents have providence over the interaction protocol allowing agents to make transformations. The explicit transmission of the dialogue state records and communicates the choices made as the protocol is realised. It also able to catalogue the transformations made and the resulting properties which now hold because of those changes.

There are various structures which occur in human dialogue which have a different semantic interpretation but share the same syntactical shape. For example, a question followed by an answer has the same structure as a statement and a confirmation. An agent sends a message which is followed by another message being received. It is therefore necessary to restrict the vocabulary of transformations to those which can be uniquely identified by its syntactic structure. Otherwise a kind of semantic leakage occurs and ambiguity seeps into the dialogue and protocol.

Given the simplest protocol, i.e. the communication of a single message. There are a finite number of transformations which are meaningful. Table 1 shows these syntactic transformations given an atomic protocol. Parallels from dialogue structures are mapped to this set of transformations.

In dialogues, humans cue for response by a number of verbal and non-verbal cues. A message is sent and at some point later a message is received from the same agent. The messages and their content can now be said to be a *response*.

During discussions, humans will provide choices to their dialogical partners when appropriate. This same need exists in agent communication. The *counter* transformation allows agents to introduce this type of step in dialogues. A common occurrence for human dialogues is use cues to signify they wish to continue their turn in the dialogue. The *continuation* transformation enables software agents to do the same.

Clarifications and Corrections are of great interest to those studying dialogue structures [2, 1]. Corrections are usually reactions to failures in the dialogue. We have addressed outright failures such as message loss or complete misunderstanding as criteria for a *correction* transformation. Whereas, *clarifications* occur when a message received is understood but found to be wanting in detail. An example of this could be an agent providing a date but the other agent needs a year for the date as well versus an agent communicating a seemingly erroneous date such as the tenth day of the seventeenth month. The message encapsulated by a 'c' before the *clarification* transformation represents in the protocol language that the message has been sent. The 'f' encapsulation represents a message failure which is the requirement for an agent making a *correction* transformation.

The passing of the protocol allows agents to make changes and communicate those changes like actors developing a play. They suggest changes to a script that is shared amongst them as they rehearse their roles. The changes made to the protocol must be made in a controlled way or the advantages of a protocolled approach are lost. The ideas of dialogue structure taken from research in linguistics can provide the semantic buttressing need for this run time synthesis of interaction protocols. The library of transformations given a single message allows meaningful adaptations to the protocol by the agent. These transformations have correlations to dialogical phenomena in human communication. This allows agents the flexible protocols needed for dialogues which cannot be be fixed to a single protocol before execution.

1. REFERENCES

[1] N. Asher and A. Gillies. Common ground, corrections and coordination. *Argumentation*, 17(4):481–512, 2003.

[2] J. Ginzburg. Dynamics and the semantics of dialogue. In J. Seligman and D. Westerståhl, editors, *Logic, Language, and Computation*, pages 221–237. CSLI, Stanford, Ca, 1996.

[3] D. Robertson. A lightweight coordination calculus for agent social norms. In *Declarative Agent Languages and Technologies*, New York, USA, 2004. a full day workshop occuring as part of AAMAS'04.