TYCOON: six primitive types of cooperation for observing, evaluating and specifying cooperations

Jean-Claude MARTIN
LIMSI-CNRS
BP 133, 91403 Orsay, FRANCE
martin@limsi.fr
http://www.limsi.fr/Individu/martin

Abstract

In this paper, we describe TYCOON, a model that we are developing for observing, evaluating and specifying cooperations between software and human agents. This model is based on a typology made of six primitive types of cooperations: equivalence, specialization, transfer, redundancy, complementarity and concurrency. Each of these types may be involved in several goals of cooperation such as enabling a faster interaction or improving mutual understanding. We explain how we have already applied this model to cooperation between modalities in multimodal human-computer interaction both with a software development and some experimental metrics for the analysis of the multimodal behavior of subjects. We also provide insights on how we think this model might be useful to model more generally cooperation and communication between agents. We also show some of its limitations when considering psychological approaches to communication.

Introduction

In the current information Society, people have more and more opportunities to interact with and through computers. Furthermore, software is no longer seen as monolithic applications but as distributed components. Thus, it is not surprising that several federative projects address communication and cooperation between both human and software agents at several levels:

- Integration between information detected on several communication modalities such as speech and gestures in the area of multimodal Human-Computer Interaction,
- Integration between contributions of several users in Computer Supported Collaborative Work and Computer Mediated Communication,
- Integration between software agents in multi-agents systems.

Examples of such federative projects are FIPA's specifications (FIPA 1999), Open Agent Architecture (Martin, Cheyer and Moran 1999) and Magic Lounge (Bernsen et al. 98). Unified frameworks for modeling cooperation at all these levels in a coherent fashion might be useful.

In this paper, we describe TYCOON (Martin and Béroule 1993) a theoretical framework aiming at enabling the observation, the evaluating and the specification of cooperations. We explain how we have already applied this framework to multimodal Human-Computer Interaction. Then, we provide first ideas on how we think it could be related to models of cooperation in multi-agent architectures and in psychological models of communication.

The TYCOON theoretical framework

Definitions[1]

Global overview. Several agents communicate and cooperate within an application domain.

Cooperative environment. A cooperative environment is a global framework composed of: an application domain, a set of referencable objects, a set of agents and a set of types of possible cooperation between these agents.

Application domain. An application domain is defined by a set of message templates (including command names and associated parameters in the case of Human-Computer Interaction) and a set of object.

Referencable object. A referencable object embeds an object of the application with knowledge on how to refer to this object (with linguistic or non-linguistic means).

Information chunk. An information chunk is represented by a set of features. A feature provides the value of only one attribute of an information chunk (i.e. the date, at which it was detected, or a word that was recognized, or a request that was understood). A name, a content and a

[1] In this section, we will limit ourselves to our own restricted and general definitions, which will be illustrated and discussed later. We are well aware of the fact that the reader might have different definitions.

confidence factor define a feature.

Agent. An agent is a computational process represented by: its name, a set of input information chunks it may process, a set of output information chunks it may produce and a confidence factor associated with this process. An agent may be human or software.

Cooperation. A cooperation requires the exchange of information in order to achieve a common goal. In Tycoon, we have distinguished six possible types cooperation between agents (table 1).

A typology of types of cooperation

Equivalence. A cooperation by equivalence is defined by a set of agents, a set of chunks of information, which can be produced by either of the agents and a criterion, which is used to select one of the agents. When several agents cooperate by equivalence, this means that a chunk of information may be produced as an alternative, by either of them.

Transfer. A cooperation by transfer is defined by two agents and a function mapping the output of the first agent into the input of the second agent. When several agents cooperate by transfer, this means that a chunk of information produced by one agent is used by another agent.

Specialization. A cooperation by specialization is defined by an agent, a set of agents *A* and a set of chunks of information this agent is specialized in when compared to the agents of the set *A*. When agents cooperate by specialization, this means that a specific kind of information is always produced by the same agent.

Redundancy. Several agents, a set of chunks of information and three functions define a cooperation by redundancy. The first function checks that there are some common attributes in chunks produced by the agents, the second function computes a new chunk out of them, and the third function is used as a fusion criterion. If agents cooperate by redundancy, this means that these agents produce the same information.

Complementarity. A cooperation by complementarity is defined similarly as a cooperation by redundancy except that there are several non-common attributes between the chunks produced by the two processes. The common value of some attributes might be used to drive the fusion process. When modalities cooperate by complementarity, different chunks of information are produced by each agent and have to be merged.

Concurrency. A cooperation by concurrency means that several agents produce independent chunks of information at the same time. These chunks must not be merged.

Goals of cooperation. Several agents may exchange information and cooperate for several reasons such as enabling a fast interaction between agents or improving mutual understanding of the agents.

Composing cooperations. In order to model sequence of cooperations during interaction, each types of cooperation can be composed with other types of cooperation.

Limitations

These definitions do not include some classical features of agents and collaboration such as shared plan, intention, interactive dialogue, beliefs, desire, and autonomy. The reason is that in our work we wanted to focus on the exchange of information as a required feature of cooperation. Yet, we do not make any assumptions about the linguistic complexity of exchanged chunks of information (i.e. lexical, semantic, or pragmatic).

	Improving Recognition	Fast interaction	Adaptation to changes	...
Equivalence				
Transfer				
Specialization				
Redundancy				
Complementarity				
Concurrency				

Table 1: The proposed typology for classifying exchanges and cooperations. Six "types of cooperation" between agents (lines) may be involved in several "goals of cooperation (columns).

Computational model

In order to process cooperation during communication, our model involves a cooperation network (Martin and Néel 1998), which is a symbolico-connectionist model adapted from Guided Propagation Networks (Béroule 1985). A cooperation network is made of interconnected processing nodes (Figure 1). In order to cooperate with other agents, a software agent requires such a cooperation network. The next section will illustrate this cooperation network model in the case of the multimodal application that we have developed.

Input nodes: an input node can be "activated" by an external information chunk sent by other agents. When it becomes activated, it sends a symbolic structure representing the detected information chunks to one or several cooperation nodes.

Cooperation nodes: a cooperation node can be activated by one or several such symbolic structures representing hypotheses sent either by input nodes or by other cooperation nodes. The topology of such a network as well as the behavior of each cooperation node has to be specified for each new agent willing to integrate cooperations from other agents.

Terminal nodes: the activation of a terminal node represents the recognition of a message template and triggers a reference resolution process before executing an associated command by the agent.

Salience value: the reference resolution process is based on the computation of salience values such as in (Huls, Claassen and Bos 95). The salience value of a referencable object in a chunk of information gives an idea of how much this object is explicitly referred to in this chunk. A global salience value is computed across several information chunks in order to find the best candidate for the reference resolution. In case of ambiguity, two referencable objects may have the same salience in the same chunk of information. Yet, this ambiguity might be removed when considering the salience of these objects in other chunks of information.

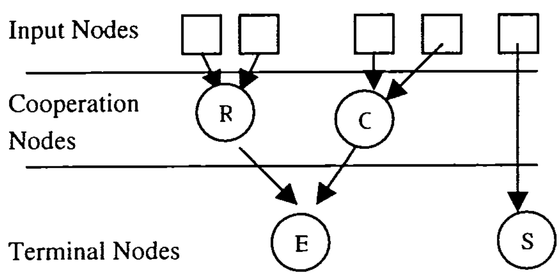

Figure 1: A simple example of a cooperation network. Input nodes (squares) are connected to cooperation nodes (circles). Each cooperation node can be of six type: Equivalence (E), Redundancy (R), Complementarity (C), Specialization (S), Transfer (T). Concurrency (//).

Fusion criteria: three fusion criteria can be used for complementarity: time coincidence (the chunks of information are produced within the same temporal window), temporal sequence, and structural completion (all specified chunks have been detected).

A command language for specifying types of cooperation: instead of hard-coding the cooperations between agents in our software for a specific application, we have developed a command language for specifying cooperations between agents.

Algorithm: the algorithm for processing chunks of information coming from other agents is provided in table 2.

```
Parse the specification file.
Create the cooperation network.
Create the set of referencable objects.
While not exit
    // Process incoming information
    IF (information chunk has been received)
    THEN
        Update salience of referencable objects.
        Create an information object.
        Put it into the output of the information
        node  managing this information.

    // Propagate information between nodes
    FOR each cooperation node in the network
      Test if this node should be activated
      (this depends on the type of cooperation of
      this node, and on the output of its input
      nodes).

      IF this node needs to be activated
      THEN
        Build a new hypothesis object.
        Compute its score.
        Put this hypothesis object into the
        output of this node.

        IF this node is terminal
        THEN
            Solve references and execute command
```

Table 2: Algorithm of propagation in the network.

Application to Multimodal Interfaces

Software implementation

In the case of the multimodal application, two modality agents (a speech recogniser and a module processing simple 2D pointing gestures) cooperate with a multimodal agent. This multimodal agent makes use of a cooperation network and a file specifying cooperations between speech and gestures (table 3). When events are detected by modalities, hypotheses on the detected cooperation between modalities propagate in the network. The salience of the referencable objects is also updated as a function of the detected events. For instance, the recognition of the word "hotel" by the speech recognizer will increase the salience of all the displayed hotels. Table 4 provides the informal definition of some heuristic rules for updating the salience of referencable objects. After several propagation steps, when a command is recognized, the most salient object is selected as a candidate for parameter's value, and the command is executed. Multimodal recognition scores are computed as a weighted sum of modality confidence factors and the score of events (such as the one provided by the speech recognizer).

```
// A variable V3 is defined as the beginning of a
// sequence
start_sequence   V3

// V3 may be activated by one event among several
equivalence V3
        Speech what_is_the_name_of
        Speech what_is_that

// This V3 variable is linked sequentially to
// a second variable V4
complementarity_sequence   V3   V4

// V4 may only be activated by a gesture event
specialization V4 Gesture *

// V4 is bound to a parameter of an application module
// which is involved in the execution process
bind_application Parameter1NameOf V4
```

Table 3: Example of the specification of a command in the multimodal map application. This specification enables the cooperation between speech and gesture in order to get the name of an object of the map (hotel, museum...).

Speech	If the recognized sentence contains the unique name of an object (i.e. "the Orsay museum"), set the salience of this object in the speech modality to the score provided by the speech recogniser. If the recognized sentence contains only the value of a property of an object (i.e. "the museum"), increase the salience in the speech modality of all referencable object having the same property value (i.e. all the museums) taking into account the score provided by the speech recogniser.
Gesture	Set the salience in the gesture modality as a function of the distance between the location of the object and the focus point of the recognized gesture.
Graphics	Set the salience in the graphics modality as a function of the distance between the location of the object and the center of the screen.
History	After the recognition of a command, the salience of objects referred to in this command is decreased by a forgetting factor.

Table 4: Informal definition of some of the rules used for updating the salience of objects when an event is detected on a modality.

Analyzing the multimodal behavior of subjects

Several user studies have been done by researchers to analyze the multimodal behavior of subjects when interacting with simulated or implemented multimodal systems; see (Martin, Julia and Cheyer 1998) for a survey of multimodal user studies. We have applied the TYCOON model to the analysis of the multimodal behavior of subjects in a Wizard of Oz experiment at the Stanford Research Institute (Cheyer, Julia and Martin 1998). During this experiment, subjects were asked to interact with a simulated system using speech and pen. Sessions were videotaped. During the analysis of the video corpus, salience of the reference to objects was computed similarly to the rules described in table 4.

The rate at which a subject makes use of equivalence (i.e. switches between several modalities for the same command) is computed with the following formula: the number of commands C_j expressed via different modalities is divided by the total number of commands expressed by the subject during the experiment.

$$\tau_{equivalence} = \frac{\left|\{Cj / equivalent(Cj)\}\right|}{\sum_{Cj} |Cj|}$$

The rate at which the subject's behavior is either redundant or complementary is computed with the following formula: a global salience value is computed over all referents r_k of all the commands C_j expressed by the subject; then this number is divided by the number of referents expressed by the subject during the experiment.

$$\tau_{compl.\,/\,redund.} = \frac{\sum_{Cj} \sum_{rk \in R(Cj)} salience(Cj, rk)}{\sum_{Cj} |R(Cj)|}$$

Some results of the computation of such statistics are described in (Kehler et al. 98).

Discussion

Multi-agent Communication

Our framework is focussing on the way an agent integrates messages coming from several other agents. Multi-agent software architectures often feature a so-called *facilitator* agent brokering the messages emitted by agents thanks to services declared by these agents. Within our model, several agents a_i may cooperate according to several types of cooperation to process a message *m* received by the facilitator (Figure 2):

- equivalence: each agent *ai* can process the message *m* but with different response time or confidence which will lead the facilitator to send the message to only one of these agents,

- redundancy: each agent a_i can process the message m but with different response time or confidence which will lead the facilitator to send the same message m to all the agents a_i, to wait for the results and to merge them,
- complementarity: each agent a_i can process only part of the message m which will lead the facilitator to send parts of the message m to all agents, to wait for the results and to merge them,
- specialization: the facilitator will send the message m to the only agent who can process it.

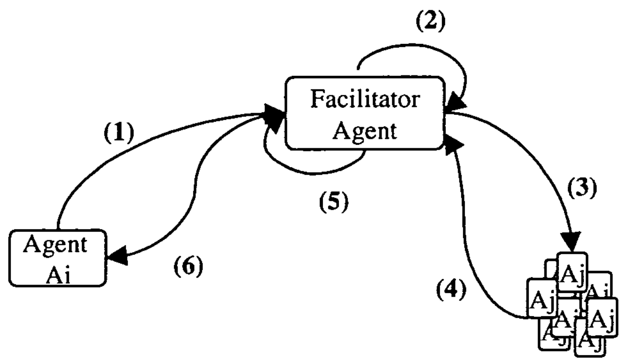

Figure 2: Suggestions on how the types of cooperation could be used by a facilitator agent in the case of multi-agent architecture:

1. An agent Ai sends a message to the facilitator agent
2. Considering the services declared by a set of agents {Aj}, the facilitator selects one or several agents Aj as well as the type of their cooperation.
3. The facilitator builds some messages and sends them to the selected agents Aj.
4. One or several agents report to the facilitator
5. In the case of redundancy and complementarity these messages are integrated.
6. The facilitator sends a reply to the agent Ai.

TYCOON could also be used for observing the cooperations between agents and modifying the selection strategy used by the facilitator in the light of the real behavior of agents compared to the services they have declared.

Psychological Models of Communication

Notions like equivalence, transfer, specialization, redundancy are often mentioned in psychological and neurobiological studies of multi-sensory fusion processes (Hatwell 1994, Stein and Meredith 1993, Hartline 1985). Yet we have not found them grouped into a coherent typology of types of information exchange or cooperation as the one we suggest.

(Decortis and Pavard 1994) consider several approaches to the study of communication and cooperation: those based on communicative acts, the cognitive approach willing to integrate the inferential nature of communication and the ethnomethodological approach which try to do without plan. Cooperation requires communication in order to exchange information, to show to others our intention, to interpret and understand the others intentions in terms of present and future actions. Communication also requires cooperation in order to ensure the success of communicative acts. In the case of human-human communication, our framework can be used for analyzing the way someone integrates knowledge coming from several people. When considering communication between only two people, it could be also used to analyze the way linguistic and non-linguistic signals (facial expression, gesture, and shared awareness of actions) are produced.

Herbert Clark considers language use as a form of joint action (Clark 1996). As a joint activity, a conversation consists of a joint action and the individual actions by the conversational participants that constitute the joint action. We believe that the types of cooperation that we have proposed could be useful to analyze the cooperations between these individual actions that make up a conversation. Individual actions could be observed to cooperate by equivalence, redundancy...in the building of the joint conversation.

(Grosz and Sidner 1986) proposed a theory of discourse structure comprising three components dealing with different aspects of the utterances in a discourse: a linguistic structure, an intentional structure and an attentional state. The basic elements of the linguistic structure are the utterances. The intentions and the relations of domination and satisfaction precedence are the basic elements of the intentional structure. Finally, the attentional state contains information about the objects, properties, relations and discourse intentions that are most salient at any given point. What we call referencable objects are related to such an attentional state component. Yet, our approach does not deal with the linguistic and intentional components.

Conclusion

In this paper we have presented a model made of six types of cooperation and a cooperation network. We described how we applied it to multimodal human-computer interaction. In the future, we would like to study how our types of cooperation could be used to model cooperation between actions of the agents having a shared plan (i.e. cooperation by complementarity between two actions could be used to achieve part of a plan).

References

Bernsen, N.O.; Rist, T.; Martin, J.C.; Boullier, D.; Briffault, X.; Dybkjaer, L.; Hauck, C.; Henry, C.; Masoodian, M.; Néel, F.; Profitlich, H.J.; André, E.; Schweitzer, J.; Vapillon, J. 1998. Magic Lounge: A Thematic Inhabited Information Space with "Intelligent" Communication Services. *La Lettre de l'Intelligence Artificielle*, May 1998.

Béroule, D. 1985. Un Modèle de Mémoire Adaptative, Dynamique et Associative pour le Traitement Automatique de la Parole. Ph.D. diss., Limsi-CNRS, Orsay University.

Cheyer, A.; Julia, L. and Martin, J.C. 1998. A Unified Framework for Constructing Multimodal Experiments and Applications. In Proceedings of the Second International Conference on Cooperative Multimodal Communication, Theory and Applications (CMC'98), 28-30 January 1998, Tilburg, The Netherlands

Clark, H.H. 1996. Using Language. Cambridge University Press.

Decortis, F. and Pavard, B. 1994. Communication et coopération : de la théorie des actes de langage à l'approche ethnométhodologique. In *Systèmes coopératifs : de la modélisation à la conception*. Bernard Pavard (Ed.) ISBN 2-906769-18-5.

FIPA 1999. http://www.fipa.org/

Grosz, B. J. and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3):175-204.

Hartline P. H. 1985. Multimodal integration in the brain. Combining dissimilar views of the world. In M.J. Cohen (ed.), *Comparative neurobiology: modes of communication in the nervous system*, chap. 18, 309-334, John Wiley and Sons.

Hatwell, Y 1994. Transferts intermodaux et intégration intermodale. In *Traité de Psychologie Expérimentale,* (eds.) M. Richelle, J. Requin & M. Robert. Paris : Presses Universitaires de France.

Huls, C.; Claassen, W. and Bos, E. 1995. Automatic Referent Resolution of Deictic and Anaphoric Expressions. *Computational Linguistics* 21(1):59-79.

Kehler, A.; Martin, J.C.; Cheyer, A.; Julia, L.; Hobbs, J. and Bear, J. 1998. On Representing Salience and Reference in Multimodal Human-Computer Interaction In Proceedings of the AAAI'98 Workshop on Representations for Multi-modal Human-Computer Interaction. July 26-27, 1998, Madison, Wisconsin.

Martin, D.L.; Cheyer, A.J. and Moran, D. B. 1999. The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence* 13:91-128

Martin, J.C. and Béroule, D. 1993. Types et Buts de Coopération entre Modalités In Actes des cinquièmes journées sur l'ingénierie des Interfaces Homme-Machine (IHM'93), 19-20 octobre, Lyon, France.

Martin, J.C. and Néel, F. 1998. Speech and gesture interaction for graphical representations: theoretical and software issues. In Proceedings of the Workshop on "Combining AI and Graphics for the Interface of the Future", European Conference on Artificial Intelligence (ECAI'98). August 24, Brighton.

Martin, J.C.; Julia, L. and Cheyer, A. 1998. A Theoretical Framework for Multimodal User Studies. In Proceedings of the Second International Conference on Cooperative Multimodal Communication, Theory and Applications (CMC'98), 28-30 January 1998, Tilburg, The Netherlands.

Stein, B.E. and Meredith, M.A. 1993. The merging of the senses. Cognitive Neuroscience series. A Bradford book. MIT Press.

Acknowledgments

The author wishes to thank at Limsi: F. Néel and J. Vapillon; and at SRI: J. Hobbs, A. Kehler, A. Cheyer, L. Julia and J.Bear.