The 1997 IEEE International Conference on Systems, Man, and Cybernetics
Hyatt Orlando, Orlando, Florida, USA • October 12-15, 1997
THEME: Computational Cybernetics and Simulation

Petri Nets in the Evaluation of Collaborative Systems

AnneMarie Ferraro
MicroStrategy, Incorporated
8000 Towers Crescent Drive
Vienna, Virginia 22180. USA

Edwin H. Rogers
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180-3590, USA

Abstract

Petri nets were conceived by Carl Petri as a mathematical means of describing activities, resources, and states of a system. They have been used to model, analyze and evaluate control system behavior. They have also been used in software engineering.

More recently, Petri nets have been used to describe the behavior of computer supported workflows and extended to apply to protocols appearing in collaborative systems comprised of human participants and multiple resources including oral communication and electronic interaction. In every case, the Petri nets are modeling protocols. These interaction management rules include interfaces among real machines, virtual machines and humans.

This paper describes Petri net modeling in the context of a study of team interactions in a mixed media lab setting. Analysis of extensive records of team speech and use of computer tools available to each member, led to empirically derived Petri nets which capture actual behavior, including the evolution of team preferred protocols. Among these were synergies between member communications via distinct media.

Specifically, throughout the study, the participants, unbeknownst to them, were learning to use oral communication to change the protocols which were assigned for the electronic medium. The change in behavior was so subtle that only through evaluating the transcripts and creating Petri net models to capture observed behavior was this made clear.

1. Collaboration Protocols

This section discusses how collaboration and protocols are related in anticipation of formal modeling and a formal empirical study of collaboration protocols in succeeding sections.

Common collaborations & protocols

People collaborate among themselves in almost every activity. Moreover, people have invented tools to make themselves more productive and thereby have developed interactions with their tools. In the broadest sense people work in collaborative environments which engage them with one another and with a variety of tools. Among the most notable of the latter are computer-based tools; indeed, “interactivity” in common parlance is associated almost automatically with computers.

Protocols are the rules by which we manage interactions with one another and with our tools. For example, Roberts’ Rules of Order are protocols for the management of meetings. They do not directly involve participants other than people but they have a notable complexity in the way they allot control of the floor and limit the topics which may be discussed in various situations.

Board games provide other examples. In checkers and chess two players take turns and must observe particular rules regarding allowable moves and when a piece may be captured. Monopoly is turn taking as well but may involve many players and allows them to barter properties and to build houses and hotels and charge rent in appropriate circumstances.

Computer supported collaborations & protocols

Other examples lie in the protocols governing access to computer system resources built into a computer operating system. A multi-tasking system will maintain a priority queue of tasks waiting for service and apply some form of round-robin, time-out, best-fit scheduling.

Protocols also govern how people interact via computer systems in the growing domain of computer supported collaborative work. In such systems, several people undertake the accomplishment of a task or set of tasks with the aid of networked computers. Special software (often called groupware) will apply protocols to manage individual access to shared computer resources and in many cases, to direct group activity toward task completion. Simple examples include the way newsgroups and on-line chat rooms are managed. Another may be found in group decision support systems that take groups through stages of problem formulation, brainstorming, commenting, voting, and other consensus building activities during which each participant has limited access to the attention of others via the computer network and limits on the nature of the information which may be seen and entered.

Some groupware limits its role to managing access to sharable computer resources without placing constraints on what the participants may do with their access (e.g., which applications to run and which data to access). This is called meta-groupware.

2. Petri Net Models

Use of Petri Nets to model protocols is illustrated in this section, and particular models are put forward to capture synergies which might be found during face-to-face, computer supported meetings. These are incorporated in a lab study of team behavior described in subsequent sections.

Simple protocol examples in PN

Not surprisingly Petri Net models can capture control protocols in collaborative situations as well as in more traditional engineered systems. For example, Figure 1 shows a simple Petri Net in which participants in a meeting are either listening or talking. Figure 1(a) places no limits on how many may speak at once, while Figure 1(b) uses a mutual exclusion token (the "mutex" asterisk) to limit speech to one person at a time. Figure 1(c) shows a further elaboration in which a listener may interrupt the speaker.

In Figure 2 is a model in which the **Invite** transition is controlled by a "guard" *[Rsvp(t,L,p)]*. The boolean function *Rsvp* must evaluate to True in order for the **Invite** transition to fire. *Rsvp* determines whether a person *p* in the listen place *L* responds affirmatively to an offer from the speaker *s* to take the floor.

Priority queuing is commonly used to manage access to a resource. Figure 3 is a Petri Net model which combines elements from the previous examples to define a first come, first served priority queue. Persons in the **Participant Pool** may request control of the resource (perhaps the oral floor or something else) at any time. When they do so, a priority number *y* is assigned and the next available *y* value is incremented. The compound token *(p,y)* enters a priority queue *Q*. When either transition **Take_control** or transition **pass_control** fires, the function *MinPr* selects the participant carrying the lowest priority number *y* from the queue *Q*.

A set of CSCW protocol models

For computer supported collaborations there are, among others, computer resources to share. For our immediate purposes, we imagine participants engaged in a face-to-face meeting. There is a computer which each may control via a personal keyboard and pointing device. All participants can see the screen of the shared computer. Because this is a face-to-face meeting, there is an oral floor to be controlled as well.

Three basic protocol models for control of the "computer floor" are described next. These are the protocols subjected to an empirical study described in the next section.

Direct Capture. This is the protocol described by Figure 1(c) applied to control of the shared computer rather than an oral conversation. Briefly, if no one has control, then any participant in the session may take control (using the sole "mutex" token). If someone is in control, then another may take control only be executing an "interrupt" which summarily replaces the controller with the requester.

Request & Grant. This protocol is described by Figure 3 applied to control of the shared computer. Briefly, if no one has control, then any participant may take control using the sole "mutex" token. If someone is in control, then others wishing control must request control by entering a first come, first served queue and wait there until the person in control, voluntarily releases control in order to move forward in the queue. The queue is not visible to participants.

Visible Queue. This is identical to Request & Grant except that (a) the queue is on display for all participants to see at all times and (b) anyone on the queue may remove themselves at any time.

There are many, many other possibilities, of course, but these are among the simplest and serve to illustrate the roles of Petri Net modeling in assessing protocols. The first two have been used in an earlier study [5].

3. A Lab Study

At Rensselaer we have a facility called the Design Conference Room http://dcr.rpi.edu/ in which face-to-face, computer supported meetings can be held. Software capable of offering a variety of protocols for controlling a shared personal computer has been implemented. It also permits each participant to run a "ghost" cursor on the shared computer screen for the purpose of pointing and highlighting during team discussions when another team member controls the public machine.

Design of a study

In order to compare the behavior and effectiveness of teams using these and other protocols, four team activities requiring collaboration were selected, student volunteers recruited and teams formed among them using background surveys and interviews. Teams were given basic training in the use of the room and its special collaboration software.

Each session for each team was assigned a different combination of protocol and task. These assignment combinations were rotated among teams in order to isolate effects of experience to some degree.

Data collection & summary of observations

Each team took part in four hour long sessions during which audio and video monitoring was accompanied by direct observation and the screen of the shared computer was saved at regular intervals. Software recorded and time stamped each request for control, each transfer of control of the electronic (computer) medium and each use of a ghost cursor. Post session surveys and a wrap-up survey were filled out by each participant. This data was subjected to careful analysis.

The data collected enabled observations on user preferences among protocols, team productivity, quality of team discussions and the manner in which teams managed control of the public computer. Details are found in [1].

Of particular interest in this paper are the synergies which became evident between interactions in the oral medium (conversation) and the computer medium. Each team member has a presence with respect to each medium. There is a corresponding "floor" which participants may control. Clearly, one-at-a-time protocols such as Direct Capture, Request & Grant and Visible Queue applied to the computer floor limit control to one person at a time. The oral floor is subject to two or more people talking at once in such face-to-face meetings as were held in this study (Figure 1(a)).

At certain points in their work under the Direct Capture protocol for computer control, the speaker would suggest that they take turns controlling the computer in some meaningful order, and then the interrupt mechanism and conversation would be used by the team members to do just this. In effect, the speaker had built a queue Q which was then used to determine control of the shared computer.

Such orally mediated passing of computer control was observed when the two queue protocols were being enforced. A team would consciously keep the queue empty until there was some oral agreement that computer control should pass to a particular team member. Then that team member would request control and the person in control would release control. This coordinated effort had the effect of an agreed upon interrupt under Direct Capture.

4. Revising the Models

Actual behavior went beyond the simple activities suggested by Figures 1,2,3. A first step in analysis was to make suitable extensions of these to reflect repeated behavior by teams. Then these were interpreted to better understand why new behaviors took place.

Extending the PN models to accommodate observed behavior

The observation of team "conspiracy" to make a protocol fit another model of interaction was not always obvious and was made clearer by the very exercise of "encoding" direct observations of team behavior in extended versions of the Petri Nets in Figures 1,2,3.

For instance, the first extension had to recognize the presence of each participant with respect to each medium. Figure 4 is a simple combination of Figures 1(a) and 2 to achieve this. Note that the synergy is expressed in the guard on the Transfer transition.

Adaptation of protocols by participants

In fact, the oral construction of a queue when there was none built into the available computer protocol made a significant change in the simple placing side-by-side of Figure 1(a) for the oral floor and Figure 1(c) for the computer floor for Direct Capture. Figure 5 shows the result in which activity on the oral floor builds a queue Q=buildQ(t,CP) and the use of this queue to govern subsequent transfers of computer control.

Similar modifications were required to accommodate behavior under the other enforced protocols [1].

5. Implications for Collaborative System Design

Several studies of meeting protocols have been reported. A notably relevant one is that of McKinlay [5]. This and a number of other studies have been subjected to meta-analysis by Hollingshead and McGrath [3], Walther [6], and McLeod [4]. Each meta-analysis summarizes findings on the effectiveness and perceptions of collaborative systems. The work by Furuta and Stotts [2] used Petri nets to drive protocol selection in the Trellis meeting management system.

The study reported in this paper has used Petri nets as the analytical tool for CSCW systems to examine all aspects of the collaborative process. This evaluation of the protocols used in the Rensselaer Design Conference Room has led to a more informed choice to be offered in successor systems and facilities, such as the Rensselaer Collaborative Classroom http://dcr.rpi.edu/colclass.html.

This work has strongly suggests that in designing future collaborative design systems (and perhaps CSCW systems in general) Petri Net models and empirical studies using well chosen collaborative activities and representative user teams will be very important.

6. References

[1] Ferraro, AnneMarie. "An Examination of the Collaborative Design Process Using Multiple Media Resources and Sharing Protocols," PhD Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1996.

[2] Furuta, Richard and P. David Stotts. "Interpreted Collaboration Protocols and their use in Groupware Prototyping," *Computer Supported Cooperative Work '94 Conference Proceedings*, pp 121-131.

[3] Hollingshead, Andrea B. and Joseph E. McGrath. "Computer-Assisted Groups: A Critical Review of the Empirical Research" in *Team Effectiveness and Decision Making in Organizations*, Jossey-Bass, San Francisco, CA, 1995, pp 46-78.

[4] McLeod, Poppy Lauretta. "An Assessment of the Experimental Literature on Electronic Support of Group Work: Results of a Meta-Analysis," *Human-Computer Interaction*, Vol 7, 1992, pp 257-280.

[5] McKinlay, Amy, Rob Proctor, Oliver Masting, Robin Woodburn, and John Arnott. "Studies of Turn-taking in Computer-Mediated Communications," *Interacting with Computers*, vol 6, No 2, 1994, pp 151-171.

[6] Walther, Joseph B., J.F. Anderson, and D. Park. "Interpersonal Effects in Computer -Mediated Interaction: A Meta-Analysis of Social and Anti-social Communication," *Communication Research*, Vol 21, 1994, pp 460-487.

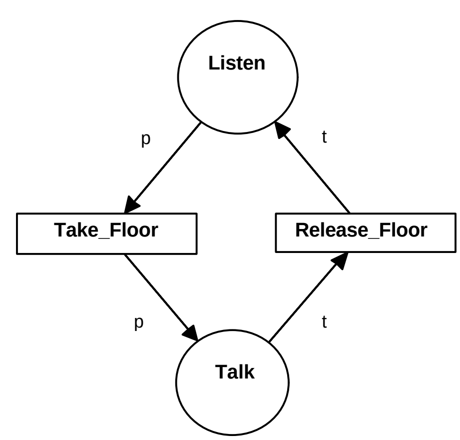

Figure 1(a). An oral meeting model in which a participant p may take the floor at any time and a speaker t may release the floor at any time. There is no limit on the number of simultaneous speakers.

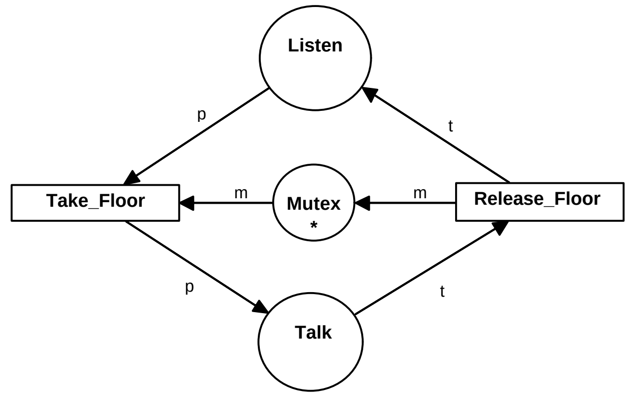

Figure 1 (b). An oral meeting model in which there can be at most one speaker at a time governed by which participant has captured a solitary "mutex" token.

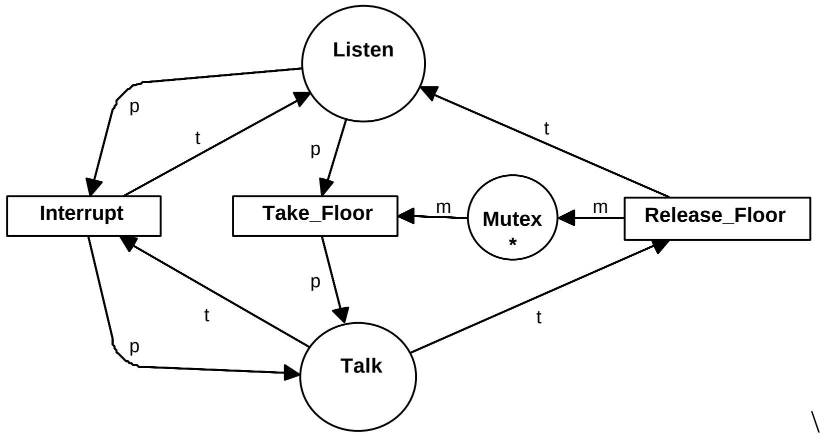

Figure 1(c). An oral meeting model in which there can be at most one speaker at a time and interruption is allowed.

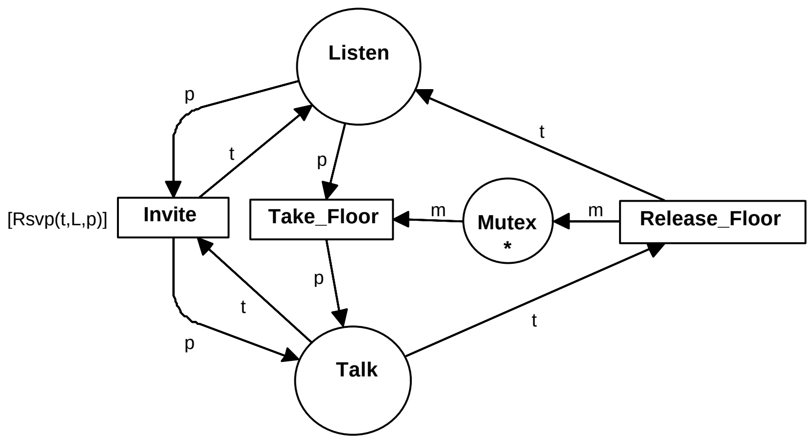

Figure 2. An oral meeting model in which the speaker t may invite another participant p from the listener place L to take the floor. The transition is guarded so as to be enabled to fire only if there is an affirmative Rsvp.

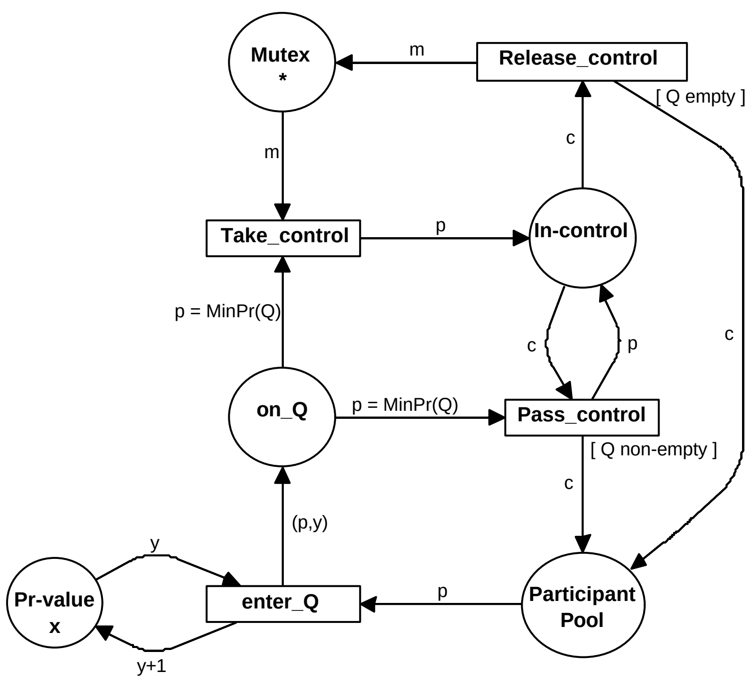

Figure 3. A floor control model incorporating a first come, first served protocol to be employed when the controller releases control and the queue of those wishing control is non-empty.

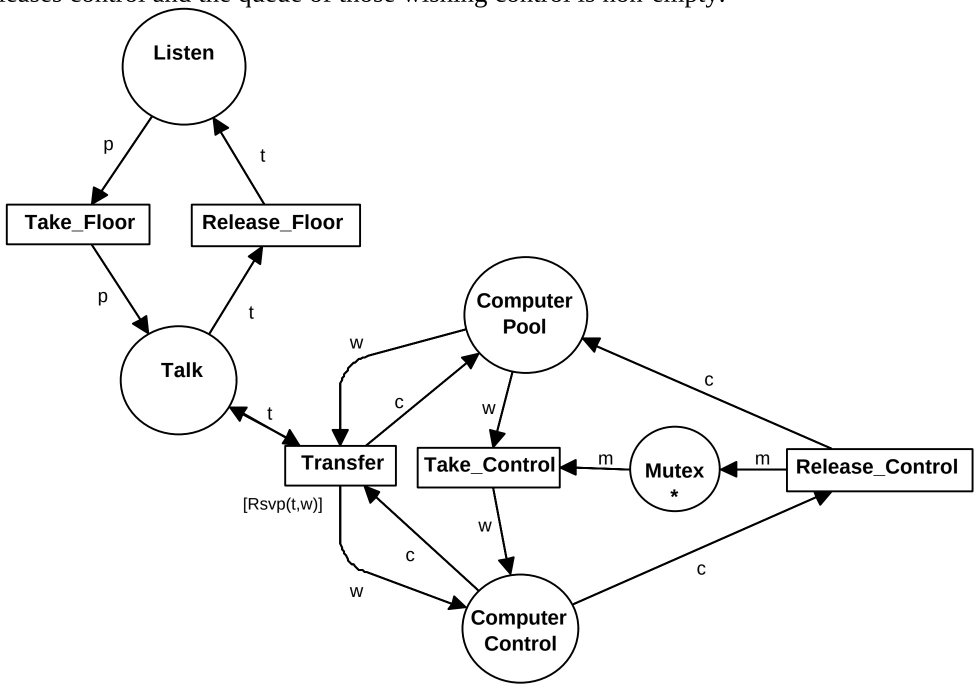

Figure 4. A session in which each participant has a presence in both the oral medium as a listener p or talker t and in the computer medium as a waiter w or controller c. Synergy is represented by the stipulation that a Transfer of computer control can occur only if the talker t receives an affirmative Rsvp from a participant w who is waiting to take control.

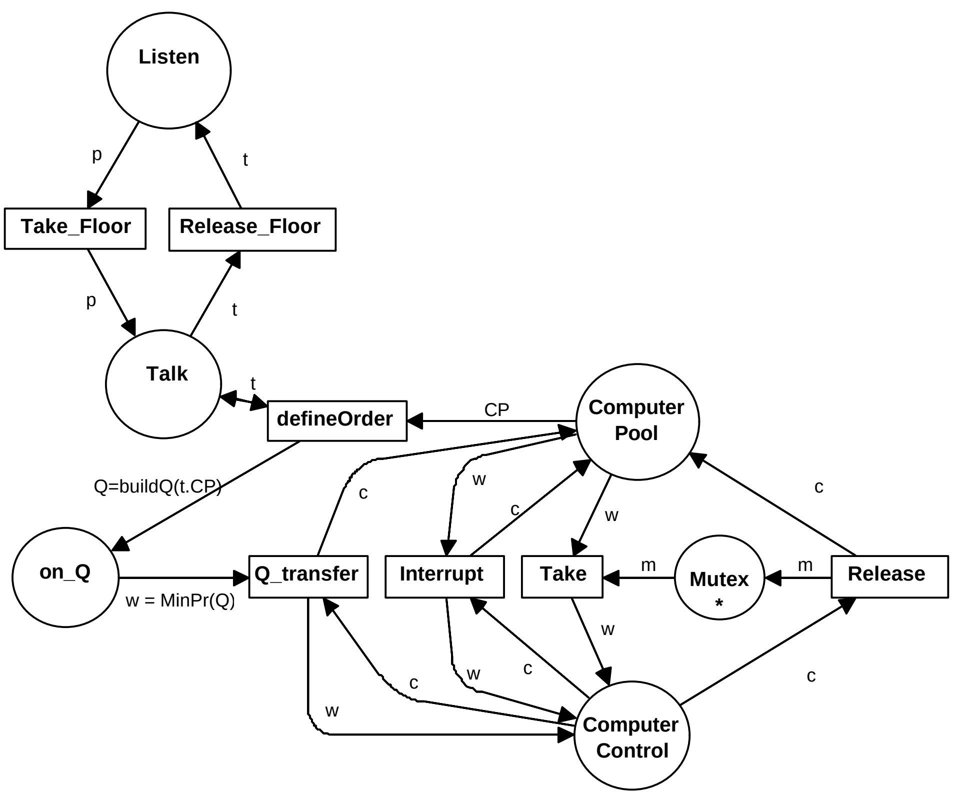

Figure 5. Discussion by t on the oral floor builds a queue Q of those waiting in the Computer Pool CP so that subsequent transfers of computer control may occur by the firing of the Q_transfer transition when the controlling participant elects to make such a transfer.