Social Comparison for Failure Detection and Recovery in Multi-Agent Settings

Gal A. Kaminka and Milind Tambe

Computer Science Department and Information Sciences Institute
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
(310) 822-1511 {galk, tambe}@isi.edu

In dynamic multi-agent environments, agents, often based on hand-crafted reactive plans (operators), form teams to collaborate in achieving joint goals[1]. The complexity and unpredictability of such environments present the agents with countless opportunities for failures, which could not have been foreseen by the designer (e.g., due to incomplete task descriptions, unreliable sensors, etc.). Without additional information about their situation, and the ideal behavior expected of them, agents are unable to detect these failures or take corrective action. Previous approaches have focused on specifying explicit execution-monitoring conditions, but this is difficult in such environments, since it is like specifying the correct behavior -- in the type of environment where the latter is problematic, the former may be just as hard. Instead, we propose a novel approach to failure detection, based on ideas from *Social Comparison Theory.* Here, agents use their team-mates as sources of information on the situation and the ideal behavior. The agents compare their own goals and plans to those of the other team members, in order to detect failures and correct their behavior. Since continuous communications from team-mates is usually costly, impractical, or erroneous, an agent can utilize its team-tracking capabilities to infer its comrades' goals and plans.

Our application domain involves developing pilot agents for participation in synthetic environments that are highly dynamic and rich in detail. While it is easy for a human designer to detect and correct failures once they occur, it is generally hard for her to anticipate them in advance. For example, a company of pilot agents may leave the home-base, leaving a single comrade hovering in place due to incorrect mission specification. Or, a pilot agent may miss detecting a landmark, failing to land even though other team-members have landed correctly. Despite significant development efforts, many failures have occurred in actual synthetic exercises, all of them in team settings. The agents should be able to automatically detect these failures using their team-mates, and take corrective action.

Our agents' design is based on *the Joint-Intentions Framework*, which allows the agents to work in the context of teams based on team-members' hierarchical joint goals. We use agent modeling techniques to allow the agents to infer team-mates' goals and plans from their observable behavior. The modeling process uses the agent's own reactive operators to also describe the goals of the other agents. The operators may be individual (specific to one agent) or joint (shared by a team). By contrasting its own operator hierarchy with those of the team members, the agent can identify discrepancies between the hierarchical reactive operators which it is executing and those of its team-mates. These differences raise alarm levels to indicate possible failures. These, in turn, are reasoned about explicitly to verify that indeed a failure has occurred, and then take corrective action. In the earlier example, our agent discovers it is stuck in place while the others are flying away by comparing its chosen method of flight to that of its comrades. In the second example, it discovers that while it is executing the joint reactive operator for formation-flying, other team-members are executing a different joint operator, which involves landing.

Joint goals must be the same for team-members by definition, and so any discrepancy must mean a failure has occurred. Individual goals, however, may differ: agents may differ in social status (e.g., military rank), or in social role (e.g., forward vs. goalie in soccer) that justify differences in behavior and goals for the team task. Our agent reasons explicitly about the social roles and status of its team-mates, filtering out differences with agents which are irrelevant for failure detection purposes. The agent also reasons explicitly about the level of evidence for a failure and its confidence in itself. For instance, a majority of differing team members provides more evidence of a failure. Higher confidence level of an agent offsets other agents' having different plans and goals.

The process described above has been implemented in Soar. We are now looking into expanding the reasoning process regarding the evidence of failure. An additional task is to integrate communications (when possible) for verification of failures and alerting team members.

References

Newell, A.. 1990. *Unified Theories of Cognition.* Harvard University Press.

Tambe, M. 1996. Tracking Dynamic Team Activity. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 80-87. Portland, Oregon.