A Synergy of Agent Components: Social Comparison for Failure Detection

Gal A. Kaminka Milind Tambe
Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
{galk, tambe}@isi.edu

1 Overview

Recently, encouraging progress has been made in integrating independent components in complete agents for real-world environments. While such systems demonstrate component *integration*, they often do not explicitly utilize *synergistic* interactions, which allow each component to function beyond its original capabilities because of the presence of other components. This abstract presents an implemented illustration of such explicit component synergy and its usefulness in dynamic multi-agent environments. In such environments, agents often have three important abilities: (a) collaboration with other agents (teamwork), (b) monitoring the agent's own progress (execution monitoring), and (c) modeling other agents' beliefs/goals (agent-modeling). Generally, these capabilities are independently developed, and are integrated in a single system such that each component operates independently of the others, e.g., monitoring techniques do not take into account the modeled plans of other agents, etc.

In contrast, we highlight a synergy between these three agent components that results in significant improvement in capabilities of each component: (a) The collaboration component constrains the search space for the agent-modeling component via maintenance of mutual beliefs and facilitates better modeling, (b) the modeling and collaboration components enable SOCFAD (Social Comparison for Failure Detection), a novel execution monitoring technique which uses other agents to detect and diagnose failures (the focus of this abstract), and (c) the monitoring component, using SOCFAD, detects failures in individual performance that affect coordination, and allows the collaboration component to replan.

SOCFAD addresses the well known problem of agent execution monitoring in complex dynamic environments, e.g., [4]. This problem is exacerbated in multi-agent environments due to the added requirements for coordination. The complexity and unpredictability of these environments causes an explosion of state space complexity, which inhibits the ability of any designer to enumerate the correct response in each possible state in advance. For instance, it is generally difficult to predict when communication message will get lost, sensors return unreliable answers, etc. The agents are therefore presented with countless opportunities for failure, and must autonomously detect them and recover.

To detect failures, an agent must have information about the ideal behavior expected of it. This ideal is compared to the agent's actual behavior to detect discrepancies indicating possible failure. Previous approaches to this problem (e.g., [4]) have focused on the designer or planner supplying the agent with redundant information, either in the form of explicitly specified execution-monitoring conditions, or a model of the agent itself which may be used for comparison. While powerful in themselves, these approaches have limitations which render them insufficient in dynamic multi-agent environments: (a) They fail to take into account information from sensors that monitor other agents, and are thus less robust. For example, a driver may not see an obstacle on the road, but if she sees another car swerve, she can infer the presence of the obstacle; (b) Monitoring conditions on agent behavior can be too rigid in highly dynamic environments, as agents must often adjust their behavior flexibly to respond to actual circumstances; and (c) Both approaches require the designer to supply redundant information, which entails further work for the designer, and encounters difficulties in scaling up to more complex domains.

We propose a novel complementary approach to failure detection and recovery, which is unique to multi-agent settings. This approach, SOCFAD, is inspired by ideas from Social Comparison Theory [1], a theory from social psychology. The key idea in SOCFAD is that agents use other agents as information sources on the situation and the ideal behavior. The agents compare their own behavior, beliefs, goals, and plans to those of other agents, in order to detect failures and correct their behavior. The agents do not necessarily adapt the other agents' beliefs, but can reason about the differences in belief and behavior, and draw useful conclusions regarding the correctness of their own actions. This approach alleviates the problems described above: (a) It allows relevant information to be inferred from other agents' behavior and used to complement the agent's own erroneous perceptions, (b) It allows for flexibility in monitoring, since the flexible behavior of other agents is used as an ideal, and (c) It doesn't require the designer to provide the agent with redundant information, utilizing instead other agents as information sources.

Teamwork or collaboration is ubiquitous in multi-agent domains. An important issue in SOCFAD is that the agents being compared should be socially similar to yield meaningful differences. By exploiting the synergy with the collaboration component, SOCFAD constrains the search for socially-similar agents to team-members only. Furthermore, the collaboration component is able to provide SOCFAD with guarantees on other agents' behaviors (through mutual beliefs) which are exploited to generate confidence in any detected failures. By exploiting the agent-modeling component's capacity to infer team members' goals, SOCFAD enables efficient comparison without significant communication overhead.

Knowledge of other agents can be communicated. However, such communication is often impractical given costs, risk in hostile territories, and unreliability in uncertain settings. Our implementation of SOCFAD relies instead on the agent modeling component that infers an agent's beliefs, goals, and plans from its observable behavior and surroundings for comparison.

2 Implementation

Our agents' design is based on reactive plans (operators) [1], which form hierarchies that control each agent. The design implements an domain-independent explicit model of teamwork [3]. Operators may be team operators (shared by the team) or individual (specific to one agent). Team operators achieve and maintain joint goals, and require coordination with the other members of the team as part of their application.

We use the $RESC_{team}$ [2] agent-modeling technique to *infer* the operator-hierarchies of other agents in the team from their observable actions. The agent therefore has unified representation of its own plans and those of its team-mates. The comparison process is simply comparing the operators in equal depths of the hierarchies belonging to the agent and its social role models.

Explicit team operators form the basis for teamwork, requiring mutual belief on the part of the team members as a condition for the establishment, and termination of team operators. At the team level, members are maximally socially similar, requiring that identical plans be executing. Any difference in team operators between agents in a team is therefore a certain sign of failure, regardless of its cause.

In service of team operators, different agents may work on different individual operators. These do not carry with them the responsibilities for mutual belief that team operators do, and so differences in individual operators are not sure signs of failure, but at best indications of the possibility. We therefore require additional information about the agent's role and status which can help in determining whether the difference is justified or not.. Differences with agents of similar role or status have greater weight in our confidence that a failure has occurred.

3 An Example: SOCFAD at Work

Our application domain involves developing pilot agents in a multi-agent battlefield simulation--dynamic, complex and rich in detail. Here, agents encounter never-ending opportunities for failure. For example, a team of three helicopters arrives at a specified landmark position. Upon detection of the landmark, they are to jointly switch from a "fly-flight-plan" plan to a "wait-at-point" plan, in which one of the team-members, whose role is that of a *scout*, is to continue forward towards the enemy, while its teammates (*attackers*) wait for its return. Due to unanticipated sensory failure, one attacker does not detect the landmark at the waiting point. Without SOCFAD running, instead of waiting behind, the miscoordinating agent would continue to fly forward with the scout, leaving the other attacker behind. However, with SOCFAD running, the miscoordinating agent infers (through agent modeling) that the other agents are executing the "wait-at-point" plan and detects a discrepancy with its own team plan of "fly flight plan". It then infers (by abduction) that the other agents have detected the landmark, even though its own sensors didn't. By adopting this belief, it recovers and re-establishes coordination with the team.

4 References

[1] Newell A., 1990. Unified Theories of Cognition. Harvard University Press.

[2] Tambe, M. 1996. Tracking Dynamic Team Activity, in Proceedings of the National Conference on Artificial Intelligence (AAAI-96), Portland, Oregon.

[3] Tambe, M. 1997. Agent Architectures for Flexible, Practical Teamwork, in Proceedings of the National Conference on Artificial Intelligence, Providence, Rhode Island.

[4] Williams, B. C.; and Nayak, P. P. 1996. A Model-Based Approach to Reactive Self-Configuring Systems. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, Oregon.