Using Decision-Theoretic Models to Enhance Agent System Survivability

Anthony Cassandra, Marian Nodine, Shilpa Bondale
Telcordia Technologies, 106 E 6th St. Suite 415, Austin, TX 78733 USA
{arc,nodine,bondale}@research.telcordia.com

Steve Ford, David Wells
Object Services and Consulting, 6111 Baywood Ave., Baltimore, MD 21209 USA
{ford,wells}@objs.com

ABSTRACT

A survivable agent system depends on the incorporation of many recovery features. However, the optimal use of these features requires the ability to assess the actual state of the agent system accurately at a given time. This paper describes an approach for the estimation of the state of an agent system using Partially-Observable Markov Decision Processes (POMDPs). POMDPs are dependent on a model of the agent system – components, environment, sensors, and the actuators that can correct problems. Based on this model, we define a state estimation for each component (asset) in the agent system. We model a survivable agent system as a POMDP that takes into account both environmental threats and observations from sensors. We describe the process of updating the state estimation as time passes, as sensor inputs are received, and as actuators affect changes. This state estimation process has been deployed within the agent system that runs the Ultralog application and tested using Ultralog's survivability tests on a full-scale (1000+) agent system. This test successfully ran a long-running logistics application in an unstable environment with high failure rates.

1. INTRODUCTION

This paper describes the approach for estimating the state of an agent system, i.e., the composite state of its agents and other components, as implemented within the Ultralog [5] Adaptive Defense Coordinator (ADC) [6]. A key aspect of the Ultralog system is its resilience to failures and attacks. The Ultralog application runs over an agent society that is assumed to exist in an unstable environment, both with respect to natural and externally-induced failures. Resilience to failure is accomplished using a variety of independently developed security- and robustness- oriented defense mechanisms, some of which operate in a proactive and some in a reactive manner. Within the defenses, sensors detect problems within the agent system, and actuators respond with some remediation action. For example, an agent health sensor may detect that an agent has failed, and a restart action will restart that agent. Alternatively, if there is an elevated security threat, an action may be taken to raise encryption levels for messages between certain agents.

While it is desirable for numbers of independently-developed sensors and actuators to be able to effect changes in an agent system, this approach has the effect that the sensors and actuators may interfere with each other, causing the outcome of their operation to be destructive rather than constructive. For instance, an agent may deliberately disconnect itself temporarily, but a "liveness" sensor may report the agent "down", possibly causing an inappropriate decision to restart the agent on another machine.

The Adaptive Defense Coordinator (ADC) presides over the sensors and actuators. It considers the entire set of sensor outputs that it receives, and attempts to determine an overall picture of the state of the agent system and its components. This paper examines this *state estimation* component of the ADC. Based on this overall picture of the agent system state, the ADC also may enable one or more nonconflicting actions at any given time, aiming to improve the overall health of the agent system.

2. MODELING THE AGENT SYSTEM

State estimation operates over a domain model representing the agent-based system, its operational environment, and its applications. At the heart of the domain model is a notion of an *asset* – a distinguishable, monitored entity within the agent-based system. An asset can be an agent, a node that agents reside on, or an elements of the agent's infrastructure. Each asset has a specific type that provides the modeling information necessary to describe the state of an asset. Each asset type model is composed of a set of *state dimensions* that capture the salient attributes of the asset.

The state of an asset is monitored using one or more *sensors*, each which observes a specific asset, and provides *diagnoses*. The state estimation process examines the collective set of diagnoses for a given asset to obtain an overall, composite view of the state of that asset. An *actuator* can affect the state of an asset. If an asset is not in a "good" state, then one or more actuators may be invoked by the ADC in an attempt to rectify the problem.

The environment in which the agent-based system is running is modeled using *threat models*. Threats are environmental factors that may affect assets, such as natural computer failures or deliberate attempts at security compromise. Each threat may (with some probability) cause one or more *events*, the effects of which may correspond to an actual change in the state of some asset. Effects may be *transitive* – for instance, if a node fails, then all of the agents

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.
Copyright 2005 ACM 1-59593-094-9/05/0007 ...$5.00.

that reside on that node also will fail. An event could also trigger other events that would in turn affect other assets.

The individual models for the assets, sensors, environment and actuators are the components of an overall model of the assets in the Ultralog system. The discrete, probabilistic nature of these components and the inherent hidden state of the assets, make the partially observable Markov decision process (POMDP) model [2] a very good way to formally capture all these components. We adapt the POMDP model framework to the specifics of the ADC requirements, allowing us to leverage existing state estimation techniques.

Generally, POMDP models are created with the intent of using automated techniques for finding optimal or near optimal policies of action. In the ADC, we use the POMDP model as a framework for calculating state estimations in discrete models with hidden state. However, we use heuristic techniques, rather than computing the optimal policy, to determine the ADC actions. To keep the complexity of the model to a manageable size, we use a series of independent POMDP models, one for each state dimension of each asset type. This makes a relatively strong assumption about the independence of asset state dimensions, sensors, events and actuators; however, experiments have proven the effectiveness of the ADC nonetheless.

3. STATE ESTIMATION

Due to uncertainties related to the environment, the sensors, and the agent system, it is rarely possible to get a definitive understanding of the true state of a given agent at a specific time. Instead, we use a state estimation for an agent defined as a probability distribution across the states in each dimension. These probabilities reflect the uncertainties associated with the asset's environment, including uncertainties about diagnoses, threats, events, and actuator effects.

The POMDP model parameters for the state transition function are: $P(s'|s,a)$ where s is the current state, s' the ending state and a the action. In the Ultralog model, there are two separate conditions under which assets can undergo state transitions: threat-based events and actuator usage. Actuator-based state transitions are directly modeled so that each action corresponds to one of the actuators. Most of the currently implemented Ultralog models use deterministic state transitions for the actuators, though our development of the state estimation computations makes no such assumption.

The probabilities for threat-based state transitions are kept separately from the actuator-based state transitions. They are also time-dependent, something not normally present in a POMDP model. This means that the POMDP model parameters for threat-based state transitions must be computed dynamically based on the elapsed time and all of the events that could have affected the asset. The formulas for computing these probabilities are complex, as multiple events (both direct and transitive) may have been involved.

The observation function parameters of a POMDP are $P(z|s,a)$ where z an observation, s is a state and a the action. Each observation corresponds to a sensor diagnosis value. We assume that these are not dependent on the action chosen, amd therefore model these conditional probabilities directly within the sensor models.

In the state estimation process, the ADC keeps a state estimation for each asset and continually updates this as it receives new information and as time passes. The initial state estimation for an asset is explicitly given in the model and usually reflects a "good" initial state. There are two system situations which trigger the ADC to update the state estimation for an asset: sensor diagnosis arrival and actuator completion. Both these state estimation calculations must also account for the possible effects (both direct and transitive) of these threats over time. Therefore, the state estimation computation consists of two phases. The first phase updates the state estimation to account for the passage of time since the last state estimation, while the second phase incorporate the newly arriving information, either the diagnosis or the actuator result.

4. RELATED WORK AND CONCLUSIONS

Defense coordination has been addressed by Ultralog for several years. Both Brinn and Greaves [1], and Helsinger et.al [4], have described the overall Ultralog approach to survivability. The description in Helsinger et.al. used an earlier, more narrowly-scoped version of the technology described in this paper. The basic architecture and function of the Adaptive Defense Coordinator as it exists today is described in [6]. An extended version of this paper, covering both theoretic and implementation issues, will be published in [3].

The goal of the work described in this paper is to provide a means to accurately assess the state of an agent-based system, in terms of the states of each of its components, and to use this to enhance system survivability in the face of significant levels of failure and uncertainty in its environment. Our approach uses POMDP models to estimate state in a large agent system. Though system-specific constraints required adapting the model and making independence assumptions, the existing theory for these models proved applicable towards the survivability goal. To test our theories, we implemented this approach within the agent system supporting the Ultralog application. This implementation was tested successfully as a part of the Ultralog survivability testing on a system of 1000+ agents.

5. ACKNOWLEDGMENTS

This research is sponsored by the Defense Advanced Research Projects Agency Information Exploitation Office (IXO) under contracts MDA972-01-C-0025 and NBCHC010011. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government. The authors also would like to acknowledge additional contributors, including Mark Greaves, Marshall Brinn, Michael Thome and Aaron Helsinger.

6. REFERENCES

[1] M. Brinn and M. Greaves. Leveraging agent properties to assure survivability of distributed multi-agent systems. In *Proc. Int'l Conference on Autonomous Agents and Multi-Agent Systems*, July 2003.

[2] A. Cassandra, L. Kaelbling, and M. Littman. Acting optimally in partially observable stochastic domains. In *Proc. 12th National Conference on Artificial Intelligence*, pages 1023–1028, August 1994.

[3] A. Cassandra, M. Nodine, S. Bondale, S. Ford, and D. Wells. Using pomdp-based state estimation to enhance agent system survivability. In *Proc. IEEE Symposium on Multi-Agent Security and Scalability*, August 2005. (to appear).

[4] A. Helsinger, K. Kleinmann, and M. Brinn. A framework to control emergent survivability of multi-agent systems. In *Proc. Int'l Conference on Autonomous Agents and Multi-Agent Systems*, pages 28–35, July 2004.

[5] DARPA ultralog web site: `http://www.ultralog.net`.

[6] D. Wells, P. Pazandak, M. Nodine, and A. Cassandra. Adaptive defense coordination for multi-agent systems. In *Proc. IEEE Symposium on Multi-Agent Security and Scalability*, August 2004.