

A General Sentinel Model: Towards a Consensus for Plan Monitoring Agents

Brian Kettler

ISX Corporation
2000 North 15th St., Suite 1000
Arlington, VA 22201
bkettler@isx.com

Joseph C. Pemberton

Pacific-Sierra Research Corporation
1400 Key Boulevard, Suite 700
Arlington, VA 22209
pemberto@psrw.com

From: AAAI Technical Report WS-98-02. Compilation copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Abstract

In this paper, we present a general model for plan sentinel agents. We raise some of the important design issues for sentinel systems, and discuss some of the implications for several DARPA Programs with which we have recent experience. We conclude that sentinel agents will provide both human and automated planners with relevant feedback on plan execution

Introduction

In complex, real-world domains such as military command and control, the plan execution environment is dynamic and inaccessible, making the outcome of actions uncertain. Conventional planning techniques cannot cope with the pace of change inherent in a battlespace or similar plan execution environments. Classical AI planning techniques have often assumed relatively static worlds with limited mechanisms for change. Human planning processes with today's systems also do not typically plan for change. For example, in an air operations planning domain, few alternative plans (Air Tasking Orders) are generated, due to the production cost. Furthermore, Air Tasking Orders are typically generated on a rigid cycle of one every 24 hours. Unfortunately, the bottom line is that the real world typically changes in unpredictable ways, and planning systems need to address these issues in order to ensure success.

Planning systems must accommodate change both by allowing the rapid generation of alternative or contingency plans and by detecting events in the world that necessitate the revision of a plan. The former is the focus of automation efforts such as the DARPA/Rome Lab Planning Initiative (ARPI) 1997 Technology Integration Experiment.¹ The latter task is the challenge for automated plan sentinels.

We view plan sentinels as agents that monitor plan assumptions, dependencies, and constraints to detect when a plan might need to be changed (e.g., an alternative action chosen, etc.), either by a human or an automated planner.² We envision hundreds or thousands of sentinels of varying complexity - ranging from simple triggers to intelligent agents - which monitor key plan assumptions in a dynamic data environment. Changes to the situation would cause changes in situation data that would be detected by sentinels. Human or machine planners would be informed of relevant changes. Planners could then decide if and how to revise the plan or conversely the output of sentinels could be used to select between different contingency plans.

In the rest of this paper, we present a conceptual model for plan sentinels. This work was first presented at the DARPA ISO Sentinels Workshop³ in May of 1997 to stimulate discussion and further refinement of what plan sentinels are, what is required to support them, which DARPA-funded programs could benefit from them, etc. This workshop included representatives from several DARPA planning and C2 (command and control) programs such as the Advanced Logistics Program (ALP), Joint Forces Air Component Commander (JFACC), DARPA/Rome Lab Planning Initiative (ARPI), and Project Genoa (a DARPA program to build a prototype decision support system for the National Security Council). ISX is involved in all of these programs, as well as the new DARPA program on Agent-Based Systems (ABS), in which we will further investigate the concept of sentinels as agents. We hope that this paper will stimulate further discussion and research about the issues relating to sentinels and provide a useful starting point for developing a (conceptual) taxonomy and set of basic requirements for sentinels. Application developers producing software for complex planning or command and control domains and researchers studying the underlying principles of plan monitoring need to be cognizant of these requirements and issues.

¹ ISX participated with several contractors in this demonstration that showcased rapid generation of alternative plans using SRI's Multiagent Planning Architecture (MPA), featuring the SIPE generative planner.

² For example, constraints could include structures such as plan validations captured by generative planners (Kambhampati, 1992) and also human-specified constraints.

³ <http://yorktown.dc.isx.com/sentinels/>

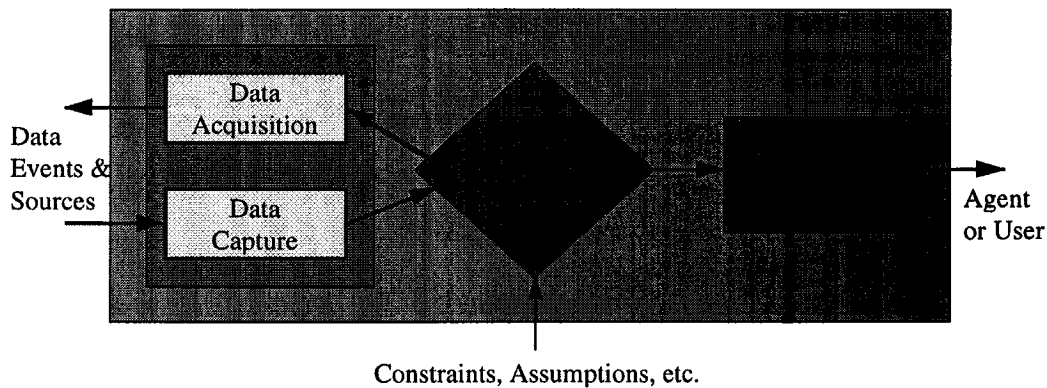


Figure 1: A General Sentinel Model

A Sentinel Model

Our high-level view of a sentinel (shown in Figure 1) breaks down an individual sentinel into three components: Data Acquisition and Capture, Processing, and Notification. Data can come to a sentinel from a variety of sources, including databases (responding to queries), live data feeds, and human input. The processing component evaluates the incoming data to determine if a monitored event has occurred. Notification can be to other sentinels, applications, or to human operators. Notice that this sentinel model subsumes triggers and monitors. A trigger is simply a sentinel with minimal data acquisition and minimal processing. A monitor is a sentinel with minimal data acquisition and some degree of processing based on the condition being monitored (alternatively, a monitor is a trigger with some additional processing capability).

A sentinel monitors information by either watching information sources (data capture), or through active data collection (data acquisition). These two sub-modules provide the sentinel with the data it needs to detect events that are relevant to the execution of the plan. The complexity of the capture and acquisition functionality will depend on the services provided by the data sources. For example, data acquisition may as simple as subscribing to one data entry in a single database. Conversely, data acquisition could require continuous polling of a variety of data sources, each of which has a different query and data format. In even more complicated situations, data acquisition might require several successive interactions with a variety of data sources before the relevant data is obtained.

The processing component provides a general purpose reasoning capability that allows the sentinel to determine the relevance of a given data event. It directs data

acquisition to gather information from various data sources and receives information from data capture. The information is processed (using some type of inference method) to determine if the event that it was created to detect has occurred. If the event has been detected, then a signal is sent to the notification component. The sentinel agent processing component needs to have some level of temporal logic in order to reason about time, for example when it is required to request information periodically or if it needs to detect a trend in data or average some data over a specified period of time (e.g., airplane landings per hour greater than 10). In addition, the processing component may in fact need some way to represent the part of the plan that it is supporting in order to assess the impact of the notification on the replanning process.

The notification component is responsible for notifying all interested parties about the observed event. This is likely done by keeping a list of whom to notify when an event is detected. When the event detection signal is received, the notification component then sends an appropriate message to each entry on the list. For example, a human user might get an email message, whereas another sentinel agent might be sent an object describing the event. In a fully automated planning system, the output of the sentinel could be used to directly impact the choice of future plan actions.

This proposed sentinel model raises several issues about the design and implementation of sentinels. These are centered around the three key requirements for complex plan sentinels:

1. **complex reasoning ability:** the ability of a sentinel to infer a complex event or situation change from data.
2. **agency:** the ability of a complex sentinel to proactively establish itself through specialization to the task/data environment, to seek and process

data, and to pass relevant information to other agents (human or machine).

3. **scaleability**: how the functionality of a sentinel (i.e., its reasoning capability, etc.) scales to more complex data environments and events and how the techniques for managing/coordinating sentinels scales to networks of hundreds of sentinels or more.

Each of these requirements will be addressed in turn.

Complex Reasoning Ability

A sentinel receives input from its data environment. These inputs are received from other sentinels (from triggers to more complex sentinels) or in response to queries issued by the sentinel to the data sources. A sentinel thus has the ability to establish triggers and form queries on the data sources, probably through the aid of data mediation tools that are either embedded in the sentinel or in the front end to the data sources. This incoming data may be aggregated by the sentinel.

A complex reasoning ability is required in order for the sentinel to be able to infer from the data whether or not the complex event that it is monitoring has occurred. A complex event is an event that requires more than simple processing to detect. For example, a simple event might be when the number of missiles at a forward airbase has dropped below a pre-specified threshold. Conversely, a complex event might be when the likelihood that the number of missiles at a forward airbase will drop below a given threshold over the next 10 days has exceeded 40 percent. In either case, the reasoning component allows the sentinel to filter situation data in order to detect an event that is relevant to some aspect of the plan execution.

The reasoning component can be implemented using a variety of techniques including rule-based reasoning, case-based reasoning, or Bayesian reasoning. For example, a belief net might be used by the sentinel to determine the belief that there will be a shortage of a given type of ammunition. Alternatively, temporal reasoning might be performed to reason about the impact of a projected bridge repair delay on the delivery of relief supplies at some later point in the plan. This sort of reasoning requires a knowledge base (rules, cases, conditional probabilities, etc.), a representation of relevant parts of the plan, and potentially significant computational resources (CPU, memory, communications bandwidth, etc.).

In addition, more sophisticated sentinels, which embody greater degrees of agency, need to reason about their mission and environment. An "environmentally aware" sentinel would understand its data environment and adapt to changes in data sources (e.g., new sources being added,

old ones being replaced/deleted, corrupted or missing data, etc.). To detect such changes and adapt accordingly, the sentinel requires a complex reasoning ability. This could range from the sentinel notifying the sentinel server when the information it needs to monitor a given event is no longer available all the way to the sentinel actively locating new sources for the missing information. If the sentinel is powerful enough, it may be able to "replan" to handle the change on its own, provided there were some planning functionality inside the sentinel.

A highly proactive, autonomous sentinel might simply be given general tasking about which event to monitor. This tasking may be fairly specific - "look for enemy SAM (surface-to-air missile) activity within range of the destination airport" - or more general "look for enemy activity that would delay the construction of a given bridge". In both cases, the sentinel must determine which data sources it should access in order to obtain the relevant information. In addition, this tasking can change. For example, the region of interest to monitor might shift during the course of a campaign. The sentinel needs to adapt to this change.

Agency

Sentinels for monitoring complex plan assumptions require some qualities of autonomous intelligent agents. These qualities include:

- **autonomy**: the sentinel can proactively establish itself to monitor relevant information and perform the requisite complex reasoning to determine if a complex event has occurred
- **articulation**: the sentinel communicates relevant information to other agents
- **specializability**: instantiated from a general class, the sentinel can be adapted to the specifics of its data environment and tasking.

Autonomy in sentinels is important for three reasons. First, it allows an application (e.g., a logistics planner) to off load some of its information processing tasks as separate processes that can operate in parallel with the application and potentially be shifted to other less burdened processors. The second advantage of autonomy is that it can allow a sentinel to migrate closer to the data that it is monitoring, thereby reducing the load on the communication channels (i.e., through transmission of a high-level notification rather than large amounts of low-level data). A third advantage is that a sentinel agent is a compact encapsulation of the information-processing task that the application would otherwise perform by itself. This encapsulated form makes it possible for different applications, and different tasks handled by the same application, to make use of the same sentinel. Eliminating redundant creation of sentinels that monitor the same events will reduce the overall number of sentinels and their overall computational burden.

In a highly autonomous sentinel system, individual sentinels might seek out opportunities for “work”. In such a system, either it or some other agent could be watching the operations of a machine or human planner. Whenever the planner makes an implicit or explicit assumption, the sentinel can go to work monitoring that assumption. Implicit assumptions are often made, particularly in mixed-initiative environments. If the planner connects a data element representing a situation to an action in the plan, for example, that operation could be interpreted as specifying a dependency of the plan on the situation element - an assumption that must then be monitored by a sentinel. When the need for monitoring that assumption is gone, such as after the action has been executed, then the sentinel needs the ability to terminate and clean up after itself.

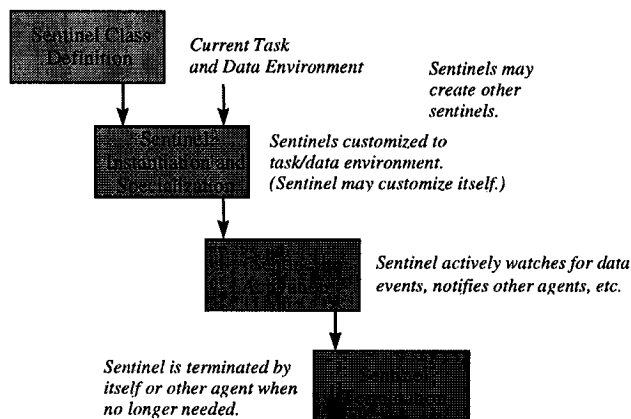


Figure 2: Sentinel Lifecycle.

A sentinel needs to be articulate: it must be able to notify one or more agents when the event being monitored has occurred. To do this, it must know whom to notify (and how to notify them). The agent being notified may want additional information besides the fact that the event has occurred. For example, a sentinel monitoring a plan assumption might signal that the assumption no longer holds. The agent being notified might want the rationale for the sentinel’s decision that the event has occurred. For example, an air-strike action in the plan might depend on the assumption that a load of precision-guided munitions arrives two hours before takeoff. This in turn depends on the assumption that a transport plane carrying these munitions arrives at the airbase prior to that time. The transport plane’s on-time arrival is predicated on clear weather conditions at the landing site. A sentinel is thus created to monitor each of these assumptions, including the weather assumption. The weather-monitoring sentinel watches the relevant data sources, applying a rule-based system, for example, to determine if snow at the landing site is likely. If the sentinel determines that snow is likely, it then notifies the planner, which may also want to know the data and reasoning upon which the sentinel’s decision was based.

Specializability is the ability to create a new sentinel or modify its behavior. A new sentinel can be created from a set of generic classes that each model a default behavior for monitoring a particular type of event. Some event types include:

- Monitor geographic region X for Enemy units of type Y,
- Monitor a transportation asset (road, bridge, etc.) to see that it remains open,
- Monitor a port for the arrival of item Y,

When a particular need for a sentinel is identified either by the planner (human or machine) or an agent monitoring the planner’s actions (such as an autonomous sentinel), a new sentinel is spawned by instantiating the generic class. For example, part of the air operations plan may depend on their being no SAMs (surface-to-air missile sites) within 10 miles of a particular target. Thus an instance of the generic sentinel class “Monitor geographic region X for Enemy units of type Y” would be instantiated with the parameters “SAM” for Y and the area within a 10 mile radius of the target for X. The advantage of specializing a general sentinel class is that the class can have mechanisms common to a variety of particular tasks. Monitoring a geographic region, for example, may require the sentinel to access imagery, situation representations, etc. These mechanisms would specify relevant data sources and how to access them and also general reasoning techniques for inferring events from data sources.

Related to specializability is the ability of a sentinel to adapt to changes in tasking and its data environment. This ability was discussed in the section on complex reasoning. The sentinel’s tasking could change, for example, if the region of interest changes. The data environment could change, for example, if the situation representation database becomes inaccessible to the sentinel and an alternative data source, such as imagery analyses, needs to be found and accessed instead.

Specializability to a particular task/data environment is one phase in a sentinel’s lifecycle. Other potential phases of the lifecycle are shown in Figure 2, above. Lifecycle issues include when to create a sentinel, how it is created, whether or not a sentinel can cause other sentinels to be created, and when a sentinel should be terminated. How to optimally manage the lifecycle of sentinels is an open question.

Scaleability

The scaleability of a system of sentinels in terms of computational resources is a function of the number of sentinels, the complexity of reasoning ability in each sentinel, and the distribution and communication patterns of the system. The complexity of reasoning includes both the computation needed to access the data sources

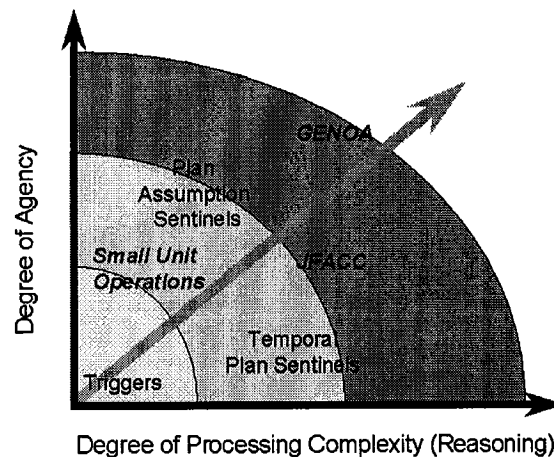


Figure 3: Sentinel Agency versus Processing Complexity for several DARPA programs.

(query processing and frequency) and the computation needed to process the inputs from the data sources. Some of this computation may be parallelizable both within a sentinel and within the system of sentinels distributed across a network of CPUs, perhaps dynamically with load balancing. There are at least two dimensions on which the scalability of sentinels can be plotted (see Figure 3). On top of this graph, we have added the name of several DARPA programs based on our perception of the types of sentinels that they would require.

The dimensions of agency and processing complexity were discussed in the previous section. The sentinels shown range from triggers (low agency, processing is a threshold or change detector on one particular data element) to complex plan sentinels (high degree of agency, processing complexity is task dependent, multiple inputs).

Some DARPA programs may require a large number of sentinels. For example, a logistics plan may have sentinels associated with materiel in transit, facilities (ports, depots, etc.), transport vehicles, etc. Other programs may require fewer, more complex sentinels such as in a crisis understanding domain where the general course-of-action may hinge on a few key assumptions, which may be monitored by sentinels using knowledge-based reasoning (Bayesian belief nets, etc.).

The management of a large number of sentinels is challenging. To the extent that sentinels can manage their own lifecycles (as autonomous, specializable sentinel agents, for example), it will obviate the need for centralized sentinel managers, which is a potential bottleneck in the system of sentinels. A more centralized manager, however, might be useful to coordinate among multiple sentinels, look for redundant sentinels, etc. Coordination among sentinel agents, including the

organization of a group of sentinels in a notification hierarchy, might also be done sentinel-to-sentinel using strategies evolved from intelligent agent research and development. Another alternative is to use an advertise-and-subscribe approach to match needs for sentinel functionality with existing sentinels.

The scalability is also influenced by the communications requirements of the system of sentinels. For example, a large number of simple sentinels may result in an overloaded communication network. Alternatively, highly complex and specialized sentinels reduce the likelihood that a sentinel's output can be used by more than one part of the planning process. The strain sentinels place on the communication resources is further compounded by the fact that applications, agents (including sentinel agents), and data sources may be distributed over a WAN, intranet, or internet.

Summary

In this paper, we have presented a general model of sentinels to support planning in dynamic environments. Our sentinel model subsumes monitors, triggers and sentinels. This work grew out of a DARPA workshop that was aimed at addressing the varied use of the work "sentinel" and its application to different DARPA programs. Our future plans are to continue developing and implementing this model and related framework and to apply it to different planning problem domains. In particular, we are currently developing an agent-based sentinel system to provide plan execution assessment for air-campaign planning under the DARPA funded Joint Forces Air Component Commander (JFACC) program. Plan sentinels fill an important role in plan execution monitoring, and we them to become an integral part of all future real-world planning systems.

References

- Cohen, P. and St. Amant, R. 1997. Mixed Initiative Plan Evaluation and Repair:
<http://earhart.cs.umass.edu/research/miper/miper.html>.
- Mattox, D.; Smith, K.; and Seligman, L. 1997. Software Agents for Data Management, MITRE Technical Report:
<http://yorktown.dc.isx.com/sentinels/readme/Agents.html>.
- Pemberton, J. 1997. Final Report: DARPA/ISO Sentinels/Triggers Workshop:
<http://yorktown.dc.isx.com/sentinels/report/index.html>.
- Russell, S and Norvig, N. 1995. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Prentice Hall.
- Valente, A.; Gill, Y.; and Swartout, W. 1997. INSPECT: An Intelligent Tool to Evaluate Air Campaign Plans:
<http://www.isi.edu/expect/inspect.html>.
- Whittaker, G. 1997. Plan Sentinels: Pro-Active Plan Monitoring, MITRE Technical Report:
<http://yorktown.dc.isx.com/sentinels/readme/PlanSentinels.html>.

Acknowledgments

This work was funded by DARPA. The authors would like to thank Tip Clifton, Ray Kolbe and Mark Klopfenstein for their help, Gary Edwards and John Schill for their input and direction, and Bob Neches for providing comments on an earlier version of this paper. This work was performed while the second author was employed by ISX Corporation.