

Extending BDI Multi-Agent Systems with Situation Management

J. Buford
Altusys Corp.
Princeton, NJ, USA
buford@altusystems.com

G. Jakobson
Altusys Corp.
Princeton, NJ, USA
jakobson@altusystems.com

L. Lewis
Southern New Hampshire U.
Manchester, NH, USA
l.lewis@snhu.edu

***Abstract** – We extend the BDI (Belief, Desire, Intention) agent model by enabling agent beliefs to be based on real-time situations that are generated by a situation management (SM) system. This has several advantages for multi-agent systems using BDI agents. First, because of the use of event correlation and data fusion techniques in situation management, agent platforms can support highly reactive distributed applications. Second, the situation manager provides a semantically rich representation of the world and can dynamically adapt its representation for situations over time. From the system architecture perspective, we discuss several alternatives for how existing BDI-capable agent platforms can incorporate this extension. These alternatives range from complete SM functionality in each agent to having shared SM functionality among multiple agents. We also consider environments where different agent platforms use our Situation-Based BDI (SBBDI) Agent method and must interoperate. We include an example of an SBBDI agent system for homeland security threat assessment.*

Keywords: Situation management, multi-agent systems, BDI, SBBDI agent, event correlation, case-based reasoning, FIPA

1 Introduction

1.1 Problem landscape

Multi-agent systems are playing an increasingly important role in monitoring and control of distributed large-scale systems whose behavior is characterized by complex operational situations. Such monitoring and control tasks are needed in modern battlespace operations, homeland security applications, and in disaster response management. Consider a military base or a large urban facility such as a sports stadium. In order for such facilities to carry out normal functions and activities, numerous types of infrastructure are individually and collectively critical, including gas pipelines, electrical stations, communications networks, water supply, transportation facilities, physical storage areas, emergency and law enforcement services, and the like. An assessment of possible threat and vulnerability situations should include knowledge of the interdependencies among these

infrastructures where such knowledge may uncover weaknesses that are especially attractive and exploitable by terrorists. Further, such situation assessments should include observations of events such as object recognition and intelligence data in near real-time.

A system is needed that allows representation and reasoning over multiple types of infrastructure vis-à-vis situation exemplars and an evolving store of static and dynamic data. As a pre-processing step before run-time, the system should be able to generalize from historical and human-created threat and vulnerability situations to form situation exemplars. As a condition of operation, the system should provide a common base format by which to represent multiple classes of infrastructure data, situation exemplars, geo-spatial data, and intelligence. In near real-time, the system should then be able to compare situation exemplars to representations of known information and detect similarities.

1.2 A multi-agent system approach to situation management

This paper describes a Multi-Agent System (MAS) [1] approach to the problem of situation management (SM) in distributed large-scale systems as illustrated in the previous section.

We see SM as a closed-loop process, where primary information is sensed and collected from the managed operations space. This raw information from the World is then fused and correlated in order to provide required inputs for recognizing operational situations happening in the World. The operational situations provide the basis for reasoning about world entities and planning actions. Finally, actions are executed to affect the World. As the World gets affected, new information about the World is sensed and the process is repeated. Having this iterative control loop cycle is an important element of our multi-agent systems approach to SM.

1.3 Situation-based BDI agents

The need to model the intelligent acts of perception, goal-directed behavior, and reasoning about an environment has resulted in a well-known conceptual architecture called the Belief-Desire-Intention (BDI) agent model [2]. Since its introduction, the BDI model has experienced several functional advancements and software

implementations. However recent attention to large-scale disaster relief operations, homeland security tasks, and the management of asymmetric network-centric battlespaces have revealed weaknesses in the BDI model, namely the problem of coping with fast moving, unpredictable, and complex operational situations. One of the reasons for this weakness is the use of a relatively simple Event-Plan paradigm of invoking plans by a single triggering event.

In this work we extend the Event-Plan paradigm to capture multiple events; particularly, we introduce the Event-Situation-Plan (ESP) paradigm, where plans are invoked in response to correlation of multiple events and recognition of complex operational situations happening in the managed applications. In addition, we describe possible architectures of agents and agent platforms that support the ESP paradigm and discuss issues related to extending existing BDI-based MAS implementations to support ESP.

The paper is organized as follows. The next section summarizes previous research in BDI agents. The following section describes the new situation-based BDI (SBBDI) agent model. Section 4 presents an example SBBDI agent system for threat analysis of a building complex. Section 5 describes several integration architectures and section 6 concludes the paper.

2 Related work

Since its inception, the BDI model has experienced extensive research and development activity. The mathematical formalism based on First Order Logic (FOL) initially proposed by [2] and presented in a full form in [1] provides an important base for future extension of the BDI model. Emotions were integrated into the BDI architecture in the TABASCO system [3], learning mechanisms were studied in [4], and an extensible BDI interpreter architecture was introduced in [5]. A specific solution for the reasoning process based on Case-Based Reasoning (CBR) was studied in [6].

The BDI agent architecture has been implemented in variety of systems, such as UM-PRS [7], JACK [8], JAM [9], and JADEX [10]. The dMARS system [1], which is based on PRS [11], has been successfully used in several applications including air traffic management, combat air mission reasoning and control and automation of customer service applications. The use of the BDI agent model for e-Health applications was described in [12]. In [13] the BDI model was applied to network management.

Our work extends the BDI model with capabilities for event correlation and situation management, features we believe are important for many large-scale applications involving increased cognitive automation.

3 The SBBDI agent model

3.1 Basic BDI agent model

The BDI model operates with three basic cognitive attitudes: beliefs, desires and intentions. On the BDI model, rational behaviour is motivated by achieving desires via intentions, given the truthfulness of the beliefs.

In the BDI model beliefs are facts about the world that the agent possesses and believes to be true, desires are an agent's general motivations for actions, and intentions are a subset of desires that in specific circumstances the agent has decided to achieve.

An agent may react to external events from the world and to its own internal events, i.e. the events associated with the changes in the agent's beliefs, desires and intentions. The overall functioning of the BDI-agent (Figure 1) is described in [2].

Rao and Georgeff [2] added two features to the original BDI model which were motivated by their goal of building computationally effective agents. First, they replaced the declarative notion of intentions with a procedural specification of instantiated and executable plans, where plans are sequences of actions that an agent stores and is able to select. Second, a plan is invoked by a single event.

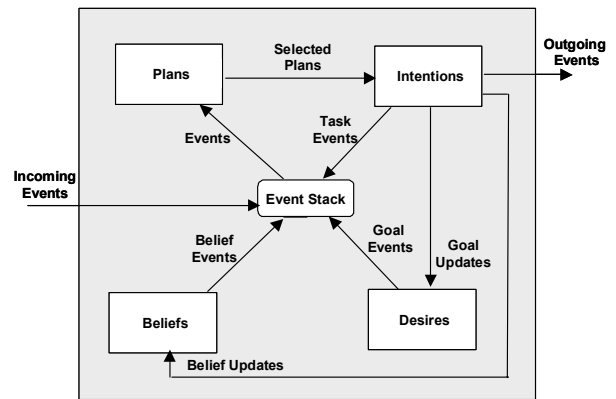


Figure 1: Architecture of the Basic BDI Agent

3.2 ESP paradigm

As mentioned in the Introduction, triggering plans by a single event might significantly limit the reasoning power of an agent. In many applications found in telecommunications, military battlefields, homeland security, and other domains, decisions are often made on the basis of multiple events. Thus, we have extended the existing model with the capability to respond to multiple events.

On a conceptual level, this means that instead of invoking plans by a single event, the agent is primed with the additional behaviors of analyzing multiple events simultaneously, performing event correlation, and inferring situations. Plans are then invoked as a result of situation interpretation, which we introduced as the ESP paradigm in contrast to the existing EP paradigm. During this process an agent considers temporal, causal and other relations between the events to recognize a high-level operational situation which describes the state of the world at a specific time. The act of recognition of such operational situations invokes a plan or a specification for automatic generation of a plan which is embedded in the situation. The new features of the BDI agent model are as follows:

1. Event correlation [14] is used to analyze complex sets of events and to capture an operational context of the world in the form of a situation, including structural,

- logical, spatial, connectivity and other domain-specific relations between the entities of the world
2. Feedback from recognized situations is used as contextual information to control the event correlation process.
3. The use of situations allows the implementation of situation prediction capabilities and learning new situations. We note that the situation learning capability is not discussed in this paper.

3.3 ESP-based agent interpreter

On the conceptual level the architecture of the interpreter of a BDI agent extended with the ESP paradigm is illustrated in Figure 2. The architecture depicts the following main steps of the interpretation process:

- a) Event Correlation
- b) Situation Recognition
- c) Plan Deliberation
- d) Plan Instantiation
- e) Intention Execution

We consider situations as aggregated states of domain entities and states of the relations between the entities at a particular moment of time or during a time interval. The relations could be very different in nature, including basic class, structural, connectivity, temporal, and other relations specific to the application domain. We distinguish between abstract situations, i.e. situations which describe certain behavioral stereotypes of configurations of entities of the world, and operational situations which are instantiations of abstract situations based on FOL existential and universal instantiation. Abstract situations are described as formulae with

The event correlation process uses spatio-temporal correlation rules, domain ontologies and facts available to the agent, and may infer a synthetic event. A synthetic event is a high-level abstraction of the meaning of the events analyzed by the event correlation process. The situation recognition process constructs an operational situation model which corresponds to the actual situation happening in the external world. The process of construction of the operational situation is based on recognition of typical situations happening in the external world, and selection and instantiation of typical abstract situation, and is used to update the overall operational situation model. The invocation of a plan from a situation allows one to model complex plan triggering conditions, including triggering pre-conditions and maintenance conditions. Since operational situations are the result of event correlation and situation recognition processes, the plan invocation conditions ultimately depend on the temporal, structural, causal and other domain specific relations.

The rest of the interpretation loop corresponds to the steps described in [2].

In the Figure 2 we also depict the agent's Semantic Memory, which comprises the agent's rules, ontologies, abstract situations, and abstract plans. The Agent's Semantic Memory is updated by an external entity, a human or another agent.

3.4 ESP and BDI agent integration

Figure 2 describes only a conceptual solution for extending the BDI model with the ESP capability, while a specific implementation of the ESP paradigm and

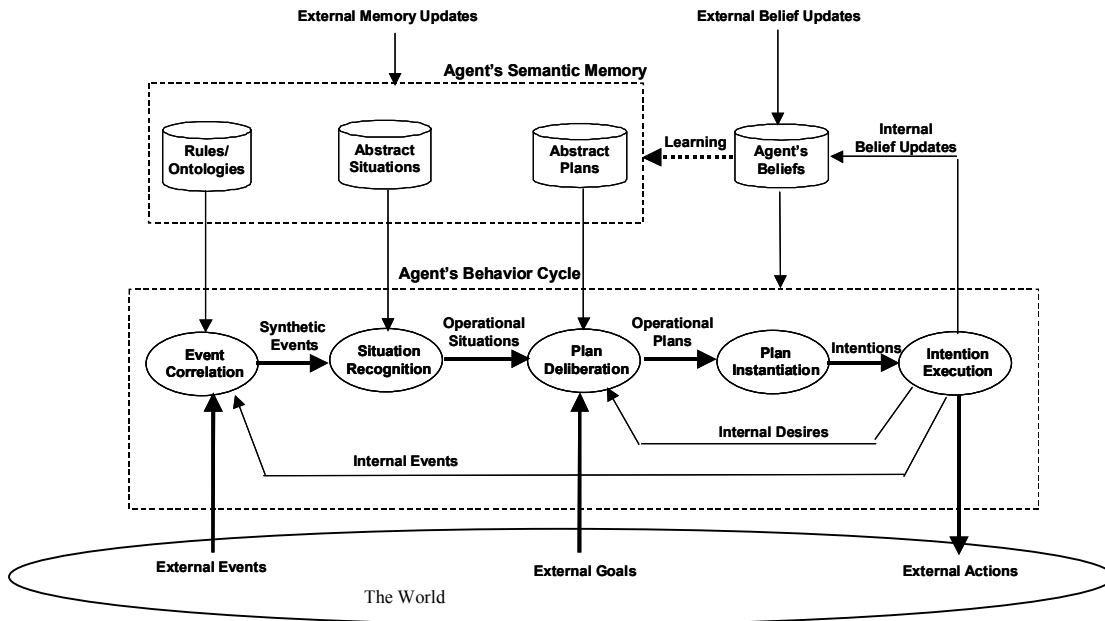


Figure 2: Conceptual Architecture of the ESP-Based Agent

variables in FOL and are stored in the Agent Semantic Memory, while the operational situations are ground formulae in FOL and are handled as facts stored as agent's beliefs. An important aspect of our approach is that a situation may contain references to the plans associated with the situations.

integration with the basic BDI-agent framework could be done in several ways. Figure 3 depicts one of the possible integration solutions.

In Figure 3, all incoming events are passed to the event correlation component which correlates those events along with the internal events from the BDI agents. Synthetic events represent the complex multi-event patterns

recognized by the event correlation process from the stream of raw and fused events. They are used by the situation recognition process to create or update the operational situation model. This operational situation model is available to the agent's belief system to update its beliefs. Since the operational situations are handled as beliefs, a corresponding belief event is generated when an agent's belief update has occurred. Then the process follows the basic BDI interpretation path.

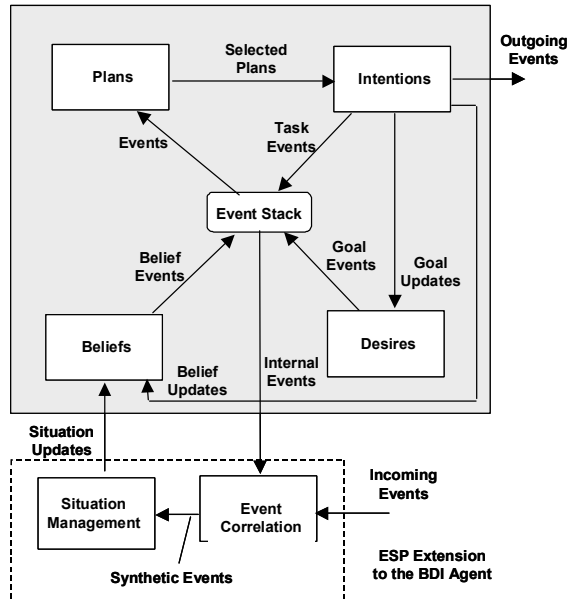


Figure 3: Situation Management and BDI Agent Integration

Since the situation recognition depends on the event correlation process, the multi-event triggering of plans using ESP paradigm is implemented not directly, but via situation recognition, situation update and generation of the corresponding belief update event. This integration preserves the general BDI agent framework.

In general, operational situations represent a subset of an agent's beliefs. Agents are free to change any belief except for those beliefs which are operational situations. These are changed only by the situation recognition process.

4 Threat assessment example

Situation management is intended for application domains characterized by large volumes of real-time events and complex domain models which require a combination of data fusion, event correlation and semantic reasoning in order to identify and assess the current context and recommend actions. As stated earlier, such application domains include homeland security, disaster recovery, and battlefield management. Figure 4 illustrates the coupling of a BDI agent system with a situation management system, using real-time threat assessment for a building complex (e.g., military base, sports stadium, industrial site, large office building) as the application problem. Specifically, we want the SBBDI agent system to continually assess vulnerabilities of the building complex

to various threats, based on changing conditions both within and external to the complex.

We decompose the threat assessment problem into specialist agents, each addressing some dimension of the infrastructure, processes, staff, or external events. We are interested in how the situation management system creates and maintains the dynamic state of the building complex for use by the SBBDI threat agents.

Each building consists of a number of inter-related infrastructures such as structural, architectural, mechanical, electrical, and communications infrastructure. The site is managed and has established operational processes, policies, and staff. Assume each building is instrumented with sensors which monitor the infrastructure. Additionally, processes and staff for operations, business functions, and security are actively monitored using workflow mechanisms. Together with events related to external conditions, these varied event streams are fused as needed and input to the event correlation function. The event correlation function processes all events and generates high level synthetic events for the situation manager. The details of the processing performed by the EC and SM have been described previously [15].

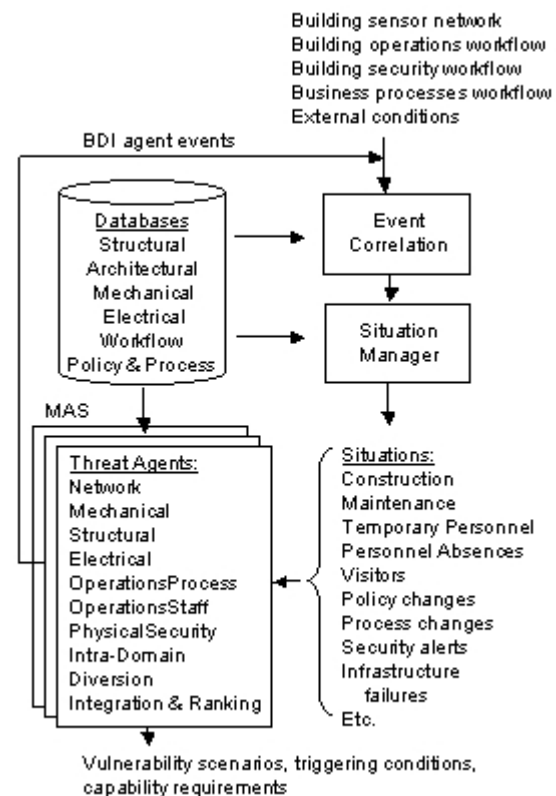


Figure 4: SBBDI threat assessment agents using situations from the situation management system

Agents access static information about building structure, workflows, policies and processes from one or more databases (Figure 4). The situation manager uses events to create and update dynamic state and produces a set of situations which describe the dynamic conditions of interest to threat analysis. For example, situations which concern unusual conditions such as changes in personnel, processes, policies, or facilities are important to threat assessment.

These situations are used by agents to manage their beliefs. Situations are updated from new events, and agent beliefs may be affected. In Figure 5, agent beliefs, plans and goals are inter-related according to the BDI model. A threat agent has a set of plans such as find-vulnerability, predict-response, and assess-capability. These plans are specialized to the infrastructure, process or staff dimension which the agent is designed to analyze. The threat agent maintains beliefs about different areas of weakness of the infrastructure it is assessing. Changes in beliefs and goals trigger new plan steps. The beliefs are derived from the real-time situations produced by the situation manager.

In addition to the specialized Threat Agents, Figure 4 also shows other agents such as:

- Inter-domain: identifies threats that cross domains
- Diversion: identifies threat-enabling activities
- Integration and ranking: performs global assessment of the most significant threats

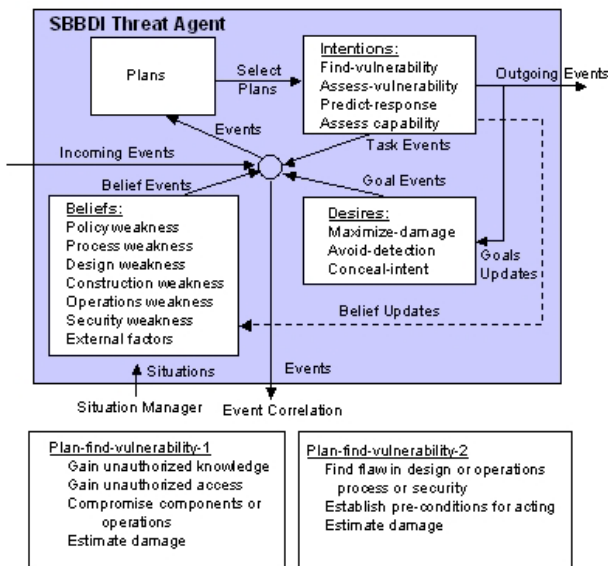


Figure 5: SBBDI threat agent

The output of the threat assessment system is a ranked set of vulnerability scenarios, triggering conditions, and capability requirements. Triggering conditions are pre-conditions which must be satisfied in order to implement the threat. This output can be represented as situations and used by upstream situation management systems which perform threat monitoring and prevention.

5 Platform and agent architecture

5.1 Integration considerations

Many agent platforms have supported the BDI model, including PRS [11], UM-PRS [7], JACK [8], JAM [9], and JADE [16]. The integration of situation management with a BDI agent platform should enhance scalability of the event processing capability of the platform and should provide a representation match between the belief representation used by the BDI agents and the situation representation provided by the situation manager.

For example, Figure 6 shows a possible integration of the SM (situation manager) and EC (event correlation) components with two agent platforms, JADEX [10] and AgentTalk [17], each of which provides BDI agents. The SM and EC components are built into the platform and provided as services to agents on the platform. This placement can reduce the external event flow and resulting processing load to the agents. Another approach is to integrate the SM component into the agent while keeping the EC separate. The choice of placement of these components in the platform or agent depends on the expected load due to message flow and situation access.

5.2 Multiple agent platforms

In the application domains of interest, various agent platforms will be deployed and should interoperate. As an example, in disaster recovery multiple teams from various jurisdictions and agencies will enter the disaster area with their staff, equipment, and supplies. These teams would typically have sensing and analysis equipment with semantic level functions enabled by agent platforms. However it is unlikely that all jurisdictions and agencies will use a common agent platform. Instead, there should be mechanisms by which different agent platforms interoperate.

FIPA (Foundation for Intelligent Agents) [18] provides interoperability between agent platforms and a directory mechanism by which agent platforms can discover other agent services. Important features of FIPA specifications include 1) a generic message transport by which FIPA-compliant agent platforms (AP) connect, 2) ability for nomadic agents to adapt to changing network conditions using monitor and control agents, and 3) a formal agent communication language based on a set of defined communicative acts and a method by which agents can establish a common ontology for communication. In addition, FIPA defines an experimental specification for an ontology service by which agents can share an ontology and translate between different ontologies. Note that FIPA does not currently specify ontologies for application domains.

In addition to the FIPA interoperability mechanisms, it is necessary that agents in different agent platforms be able to share situations and events. Continuing the example from disaster recovery, mobile teams in the field may have separate SBBDI agent platforms. Each team may want to coordinate with other teams, and also communicate to control centers which provide overall direction. To meet this requirement, the situation management function can chain to other situation management instances on other platforms (Figure 6). Message interoperability between different SBBDI agent platforms can be enabled by the FIPA mechanisms described previously. The specific types of situations and events that are propagated could be determined through negotiation between the respective sets of agents on each platform.

6 Conclusions

Situation management is intended for application domains characterized by large volumes of real-time events and

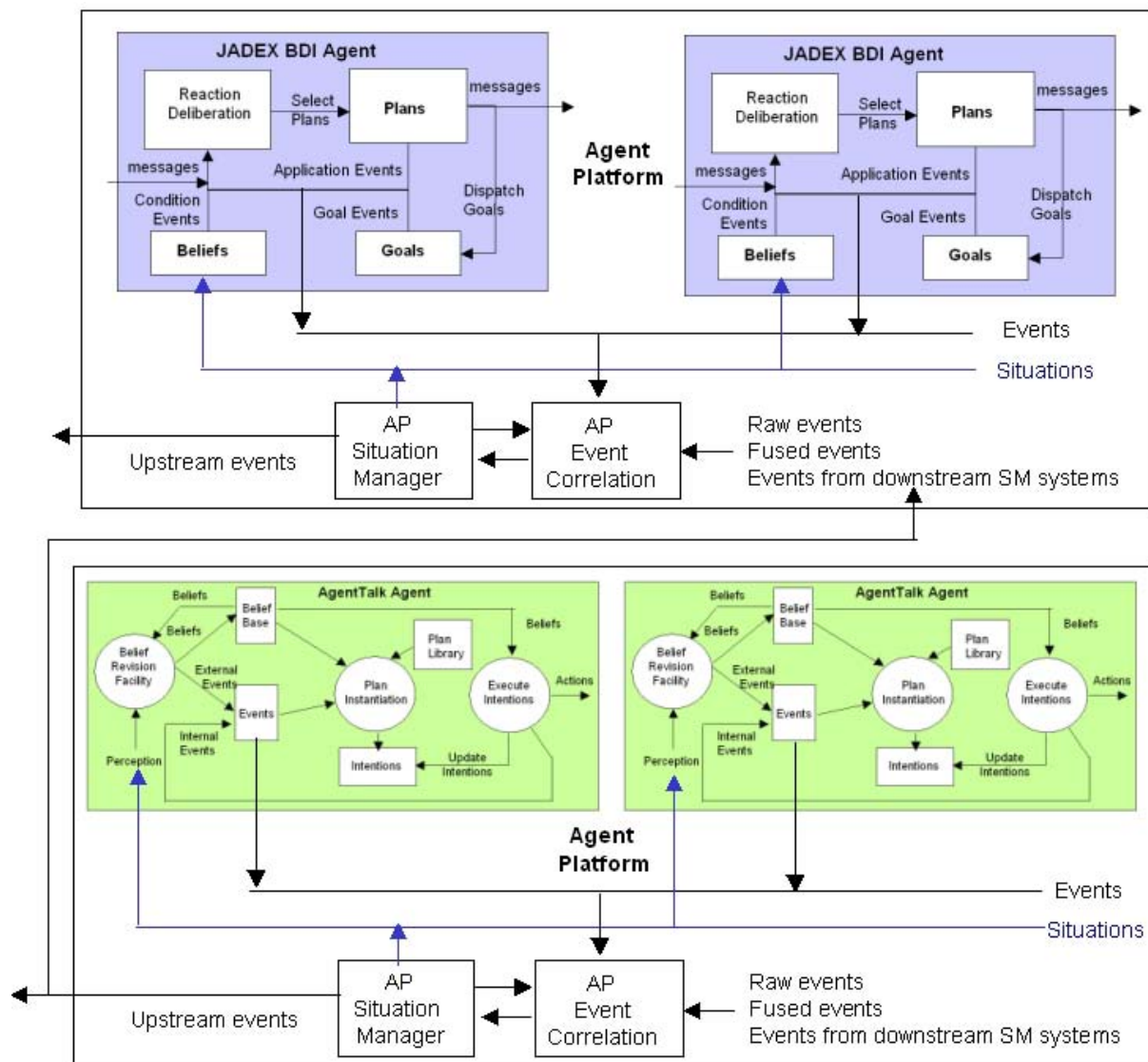


Figure 6 Agent platform level situation manager and event correlator integrated with BDI agent platforms based on JADEX design [10] (top) and AgentTalk design [17] (bottom)

complex domain models which require a combination of data fusion, event correlation and semantic reasoning in order to identify and assess the current context and recommend actions. Such application domains include homeland security, disaster recovery, and battlefield management. Due to the highly distributed and multi-disciplinary nature of these applications, MAS is a convenient development model. Further the BDI agent model is a well-established approach to designing deliberative agent systems.

Here we have extended the BDI (Belief, Desire, Intention) agent model by including agent beliefs that are real-time complex situations generated by a situation management system. This integration produces SBBDI agents which can support highly reactive applications and an enhanced representation of agent beliefs. From a system design perspective, the data fusion, event correlation and situation management technologies offer significant scalability for real-time event processing and state analysis. In addition, the large scale of these application domains suggests that multiple agent platforms will have to cooperate. We briefly described a multi-agent platform integration.

These results are illustrated by a homeland security threat assessment example.

There are several future research issues of SBBDI agents that need to be addressed:

- Effective situation specification languages and methods that preserve the completeness and correctness of situations
- Synergistic two-way communication between the basic BDI agent model functions and the situation management functions to increase the overall effectiveness of the SBBDI agent
- Learning situations by the SBBDI agent system

References

- [1] Wooldridge, M. *An Introduction to Multi-Agent Systems*. John Wiley and Sons, 2002.
- [2] A. Rao, and M. Georgeff. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, 1995.

- [3] A. Staller and P. Petta. Introducing Emotions into the Computational Study of Social Norms: A First Evaluation. *Artificial Societies and Social Simulation*, 491, 2001.
- [4] A. Guerra-Hernandez, A. El Fallah-Seghrouchini, and H. Soldano. Learning in BDI Multi-Agent Systems. In J. Dix and J. Leite, editors, *Proceedings of CLIMA IV*, Springer, 2004.
- [5] A. Pokahr, L. Braubach, W. Lamersdorf. A Flexible BDI Architecture Supporting Extensibility, *The 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-2005)*.
- [6] Pavón, J., Corchado E. and Castillo L. F. (2004) Development of CBR-BDI Agents: A Tourist Guide Application. in *7th European Conference on Case-based Reasoning*, Funk P. y González Calero P. A. (Eds.) Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence (LNAI 3155), Springer Verlag. 2004, 547-555.
- [7] J. Lee, M. Huber, E. Durfee, and K. Kenny, UM-PRS: An Implementation of the Procedural Reasoning System for Multi-Robot Applications, in *Conference on Intelligent Robotics in Field, Factory, Service and Space*, MIT Press, 1994, pp.842-849.
- [8] N. Howden, R. Rönquist, A. Hodgson and A. Lucas. JACK - Summary of an Agent Infrastructure. *5th International Conference on Autonomous Agents*, 2001.
- [9] M. J. Huber. JAM: A BDI-theoretic Mobile Agent Architecture. *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*. Seattle, WA. May, 1999. pgs 236-243.
- [10] A. Pokahr, L. Braubach, W. Lamersdorf. Jadex: A BDI Reasoning Engine, in *Multi-Agent Programming*, Kluwer Book, Editors: R. Bordini, M. Dastani, J. Dix and A. Seghrouchini, 2005.
- [11] F. F. Ingrand, R. Chatila, R. Alami and F. Robert. PRS: A High Level Supervision and Control Language for Autonomous Mobile Robots. *IEEE ICRA 96*, Minneapolis, USA.
- [12] L. Braubach, W. Lamersdorf, Z. Milosevic, A. Pokahr. Policy-Rich Multi-Agent Support for E-Health Applications, *5th IFIP Conference on e-Commerce, e-Business, and e-Government (I3E 2005)*, 2005.
- [13] M. Cheikhrouhou. BDI-oriented agents for network management, *Global Telecommunications Conference. GLOBECOM '99*, Volume 3, 1999, pp.1964 – 1968.
- [14] G. Jakobson, M. Weissman. Real-Time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints. *Integrated Network Management IV*, IEEE Press, 1995.
- [15] G. Jakobson, J. Buford, and L. Lewis. Towards an Architecture for Reasoning about Complex Event-Based Dynamic Situations, *International Workshop on Distributed Event-Based Systems DEBS'04*, Edinburgh, UK, 2004.
- [16] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa. Jade: A Whitepaper. *Exp* – v. 3, n. 3, September 2003.
- [17] Jason Manual, <http://jason.sourceforge.net/Jason.pdf>, 2005.
- [18] FIPA. FIPA Abstract Architecture Specification. SC00001L, Dec. 2003.