Adaptive Automated Opposing Forces for Urban Operations Training

Georgiy M. Levchuk, John-Colonna Romano
Aptima Inc.
Woburn, MA
georgiy@aptima.com, jcromano@aptima.com

Krishna Pattipati
University of Connecticut
Storrs, CT
krishna@engr.uconn.edu

ABSTRACT

In this paper, we describe a model of intelligent agent that learns BLUEFOR's mission structure and develops highly effective counteractions for Opposing Forces over time. The model consists of three main components. First, the agent aggregates data about movements, actions, and interactions of BLUEFOR actors to infer the sequence and types of operations that BLUEFOR is conducting. Second, the agent develops a plan to counteract BLUEFOR's operations or adjusts its mission to improve the rate of success. Finally, the agent learns over time the effects of its actions on the BLUE operations, incorporating terrain constraints and tactical effects. The main distinction of the model from standard AI techniques is in how local and global information about multiple BLUE actors and terrain features are used to make estimates about space-time activities composing coordinated BLUE tactics and learn effects of agent's actions with every experience.

Our model has been integrated with 3D virtual world to control OPFOR avatars. It provides a unique capability to train adaptation skills in urban combat. This technology can also be utilized during intelligence analysis, Wargaming, and mission rehearsals, allowing more accurate estimation of enemy courses of action and reduction of OPFOR manning footprint.

ABOUT THE AUTHORS

Dr. Georgiy Levchuk is a Distinguished Engineer in Aptima's Analytics, Modeling, and Simulation Division. Dr. Levchuk has over 9 years experience in mathematical modeling, optimization, predictive algorithm development, and software prototype implementation, and written over 55 conference and peerreviewed publications. At Aptima, Dr. Levchuk is responsible for designing the mathematical engines for several of the company's simulation products. Currently, he is leading several projects in behavior classification and analysis domain, including using probabilistic network pattern matching to find hostile resource networks and critical enemy actors, mapping the activity networks to geospatial data under high uncertainty conditions, exploiting video surveillance data to recognize complex spatiotemporal activities and functions of interdependent moving and static entities, using probabilistic network-based text minding to find entity-relation patterns in HUMINT sources, and multi-layer behavior pattern modeling to learn, classify, and forecast normal and abnormal operator workflow. Prior to joining Aptima, Dr. Levchuk was a research assistant at the University of Connecticut Department of Electrical and Computer Engineering, where he developed algorithms for organizational structure optimization and analysis. Dr. Levchuk received a Ph.D. in Electrical Engineering from the University of Connecticut, and a B.S./M.S.M. in Mathematics (with highest honors) from the National Shevchenko University of Kiev, Ukraine.

Mr. John ("JCR") Colonna-Romano is a Software Architect and Team Lead of the Modeling, Simulation and Training Systems software team at Aptima, Inc. Mr. Colonna-Romano has over 25 years of experience in software system architecture, systems engineering and software engineering. He is interested in scenario engineering, simulation based training systems, computer game technologies and software system architecture. Mr. Colonna-Romano has co-authored two books on distributed systems middleware architecture. Prior to joining Aptima, Mr. Colonna-Romano developed wireless mobile applications for handheld devices and participated in early adopter customer projects involving Internet technologies for knowledge management, web portal, and e-commerce solutions. He has also developed software to manage product data for large manufacturing projects and interactive graphic systems.

Dr. Krishna Pattipati's is a professor at Electrical and Computer Engineering Department, Univesity of Connecticut. His current research interests are in the areas of behavior pattern reorganization, adaptive organizations for dynamic and uncertain environments, signal processing and diagnosis techniques for complex system monitoring, and multi-object tracking. Dr. Pattipati has published over 350 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems. He is a head of the CYBERLAB laboratory dedicated to research on systems theory and optimization technologies to solve industrial problems. Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He was elected a Fellow of the IEEE in 1995 for his contributions to discrete-optimization algorithms for large-scale systems and team decision making. Dr. Pattipati has served as the Editor-in-Chief of the IEEE Transactions on SMC: Part B-Cybernetics during 1998-2001, Vice-President for Technical Activities of the IEEE SMC Society (1998-1999), and as Vice-President for Conferences and Meetings of the IEEE SMC Society (2000-2001).

Adaptive Automated Opposing Forces for Urban Operations Training

Georgiy M. Levchuk, John-Colonna Romano
Aptima Inc.
Woburn, MA
georgiy@aptima.com, jcromano@aptima.com

Krishna Pattipati
University of Connecticut
Storrs, CT
krishna@engr.uconn.edu

INTRODUCTION

A key challenge for battlefield simulation is the estimation of enemy courses of action (COAs). Current adversarial COA development is a manual time-consuming process prone to errors due to limited knowledge about the adversary and its ability to adapt. Development of decision aids that can predict adversary's intent and range of possible behaviors, as well as automation of such technologies within battlefield simulations, would also greatly enhance the efficacy of training and mission rehearsal solutions.

Presently, the game-based training and mission rehearsal technologies use one of the three methods to represent opposing forces (OPFOR):

- **Scripted OPFOR** uses manually defined set of actions of OPFOR and/or tasks for friendly forces (BLUEFOR) to be executed during the simulation. These events are define by subject-matter experts (SMEs) and associated with (a range of) locations and times of execution. The event types and sequences are determined based on the knowledge of possible hostile tactics and the objectives of training sessions. This approach is challenging because it requires significant manpower in defining detailed scenarios with realistic behavior variations, and does not allow training local and global adaptive behaviors in participating team members.
- **Controlled OPFOR entities** are used when a single human player (sometimes referred to as "puckster") performing a role on OPFOR team (e.g. a leader of the enemy cell) can manipulate the actions and movements of multiple OPFOR characters (referred to as "pucks"). This method still requires presence of skilled OPFOR players and is prone to biases in their experiences.
- **Software OPFOR agents** can be used to automate the actions and movements of OPFOR characters, based on their socio-cultural background, capabilities, and training requirements of the scenario. This method is preferable to reduce the footprint of mission rehearsal and training sessions; however, previous agent implementations were able to capture only local actions and motions, and have had little success in demonstrating coordinated adaptive behaviors of the realistic enemies.

THE PROBLEM

Successful software agent(s) controlling OPFOR entities and exhibiting realistic behaviors must possess four key capabilities (Figure 1). First, the agents must be able to collect observable data about motions and actions of various actors, including people and vehicles, from the environment in a manner that is similar to ground reconnaissance. This means that the location and motion data about BLUEFOR actors only in direct line-of-site from OPFOR characters can be available to the agents.

Second, location and motion data should be converted into actions, behaviors and coordinated plans similar to human's perception processes. For example, interpreting motion of BLUE characters as checkpoint setup, reconnaissance, or patrolling should be part of the perception model of and made in automated manner by the agents.

Third, potentially ambiguous inferences about plans of BLUEFOR should be converted into the plans of actions by OPFOR characters, and those actions must be carried out in a manner similar to the actions of multiple adversaries that rely on visual cues and communication to coordinate and synchronize their activities.

Finally, the agent must be able to evaluate success of the conducted operations, learn consequences and improve success of their actions over time.

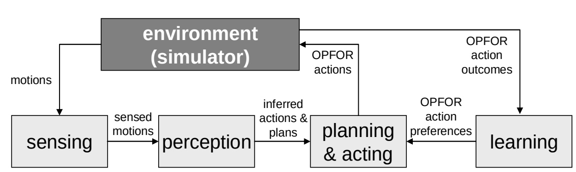


Figure 1. Main Elements of OPFOR Agent

While many researchers have studied distributed coordinated planning and execution, a significant gap exists for both perception and learning capabilities. In this paper, we describe the agents that incorporate unique plan inference and action learning based on multi-entity spatiotemporal reasoning.

RELATED RESEARCH IN PLAN RECOGNITION

Plan recognition is the process of inferring another side's plans or behaviors based on observations of its interaction with the environment. Several applications of plan recognition have been developed in the last decade. Most of the automated plan recognition models, however, have severe limitations to be used by OPFOR agents:

- **Traditional utility-based plan recognition** infers the preferences of the actors and selects the plan that achieves the highest static or expected utility. Maximum-utility plan recognition models (Mao and Gratch, 2004; Blythe, 1999) cannot track the plan evolution over time as the utility of action execution mostly does not change while the actions in the plan are executed. These models do not explicitly incorporate ambiguous observations and therefore their predictions do not change over time with incoming partial evidence.
- **Traditional probabilistic plan tracking and actor profiling** looks at patterns of activities performed by a single individual or the whole group to determine its role, threat indicator, intent, goal, or future actions. This approach does not allow tracking of coordinated and interdependent actions by multiple actors in both space and time. For example, statistical temporal event analysis techniques, such as Hidden Markov Models (Schrodt and Gerner, 2001; Singh et al., 2004), Bayesian Networks (Tu et al., 2006), Markov Decision models (Yin et a., 2004), decision tree-based models (Avrahami-Zilberbrand, and Kaminka, 2005), and conditional hierarchical plans (Geib and Harp, 2004) can reliably forecast behavior of only a single actor, dyadic relationships, or a group, where only a single action can happen at any time. Each single actor or group and its actions may look benign, and only by analyzing combined interactions can one discern the true nature of behavior to provide early predictions of future hostile activities.
- **Traditional interactions analysis models** – including differential equations (Turchin, 2003), interaction-events data analysis (Gerner et al., 2002), game-theoretic models (Brams and Kilgour, 1988), agent-based simulations (Popp et al., 2006), and others – need to be pre-populated with a large amount of data. A significant amount of noise events contribute to misleading forecasts (false alarms and false positives – the recognition of potential threats that have little or no impact) due to the sensitivity of these models to input parameters.

Instead of single actor behavior recognition, the OPFOR needs to learn the plan of BLUEFOR that consists of multiple actors performing coordinated activities constrained by the BLUE's command structure. Therefore, we need to account for the resource and organizational constraints of BLUE forces, the utility and probabilistic nature of the actions, the uncertainty in dynamic observations about BLUE's activities, and the fact that many activities might happen in parallel. The model of perception for OPFOR agent presented in this paper is similar in its ability to reason about parallel activities to the team plan recognition research (Kaminka, and Pynadath, 2002; Shi et al., 2004). Our model differs in its ability to filter the irrelevant entities and behaviors, perform data-to-model association, and find the locations of operations in the plan.

TRAINING USING VIRTUAL GAMES

Virtual simulations can be used for training various command and control (C2) skills, including situation understanding, team planning, communication, and adaptive decision making. Virtual training sessions include human players performing assigned C2 roles of the friendly forces (usually roles of unit commanders at various echelons of the organizational hierarchy). Here, the OPFOR agents can be used to control the units of opposing (RED) forces, effectively reducing the costs and human footprint of the training (Figure 2). During the game, human players receive information from the simulator using its visual interfaces, communicate with each other to assign tasks and responsibilities, schedule operation times, and control entities in the simulation by moving them and committing to execute actions. Usually, entities include people, vehicles, or military teams, and actions are either military kinetic operations (such as firing at other entities) or non-kinetic behaviors (such as greeting and communicating with locals).

Several virtual simulators have been developed in the past for military, commercial, and open-source use, including various Semi-Automated Forces (SAF) simulators such as JSAF and OTB-SAF. Recently, 3-D virtual games started to gain more ground, with many gaming engines developed with varying functionality (Wikipedia, 2010). These simulators provide overhead views for the tactical team decision making and operator views for simple-player perspectives and actions (Figure 3).

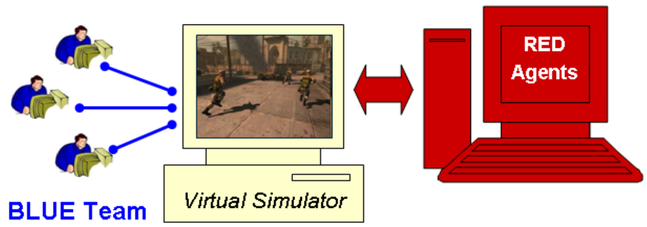
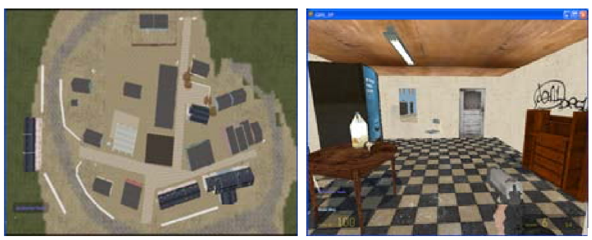


Figure 2. Using virtual simulators and OPFOR agents for C2 training



(a) Overhead view (b) Operator view

Figure 3. Example of 3D views in GDIS game system

GAMING PLATFORM AND INPUT DATA

To implement and demonstrate our OPFOR agents, we have integrated with Game DIS (GDIS) – a military training application using the 3D Gaming Engine that powers Half-Life®2 (RNI, 2010). The application, one of the winners of serious game showcase and Challenge at I/ITSEC-2006, is fully configurable and allows multiplayer participation over standard internet connection, "smart bots" and live unit animations, integration of standard DIS tools, and support for man-wearable embedded training and mission rehearsal.

The data that could be extracted from GDIS include *motion* and *state* messages at constant frequency with the following fields:

- **Id:** unique identifier of the entity in the game, which is used for aggregating all observations that relate to the same avatar; in the real world, this will be an Id of the track or an entity over which a track can be maintained.
- **TimeStamp:** this is a simulation time at which the observation is collected; it allows us to aggregate the observations in a time window and judge the temporal relations between motions and associated actions.
- **Velocity:** this is a vector of the velocity for the avatar or vehicle; it is used to reason about locomotion and interaction between avatars and other entities in the game.
- **Location:** this is latitude-longitude information for the entity; it is used to analyze spatial behaviors and aggregate events by geography.
- **Orientation:** this is the vector of the orientation of the face of an avatar; it is used to analyze the interaction between avatars and detect if a certain search or sniping is in progress.
- **Health:** this is a parameter used to judge the impact of the actions.
- **Weapon** and **ammo**: this specifies the current weapon used and remaining ammo.
- **Posture:** this specifies the posture of the avatar.

DATA PROCESSING AND EVENT RECOGNITION

Our OPFOR agent (Levchuk et al, 2008) works by consuming observations incoming from GDIS, which are low-level time-stamped locations and kinematics of the individual avatars. The agent converts location data into motion events (locomotions, interactions, and actions), aggregates those events for predefined geographic areas while fusing them with static area function information (including socio-cultural features), infers the plan that BLUEFOR is pursuing, forecasts future BLUEFOR actions, and designs OPFOR response plan.

Locomotion and Interaction Detection

Observations are aggregated for each avatar and then used to detect the next-level information: locomotions, action, and interactions between entities in the game.

The *locomotion detection* algorithm generates a stream of entity, time, and location-stamped locomotion events, from a fixed set of event types, using state, location, velocity, and orientation data from observations. The following is a list of the types of such events and the description of how they are generated in presented OPFOR agent:

- **Appear/Disappear:** this event is generated when a person appears or disappears from the area or scene; this is useful to detect in real world, but in the virtual games all characters appear at the start and disappear at the end – unless we filter out the state information

- **Stop/Start:** these events are generated when the velocity of an entity changes from positive to almost zero and vice versa
- **Walk:** this event is generated when the avatar moves with normal velocity
- **Run:** this event is generated when velocity of human avatar is high
- **Wait:** this event is generated when a person's velocity is negligible for a period of time
- **DirectionTurn:** this event is generated when an avatar's direction of movement changes significantly to left or right across several steps
- **DirectionReverse:** this event is generated when an avatar's direction of movement is reversed
- **BodyTurn:** this event is generated when a person's body skeleton is turning right/left
- **BodyReverse:** this event is generated when the body is rotated almost 180 degrees

Figure 4 illustrates how several of the locomotion types are determined.

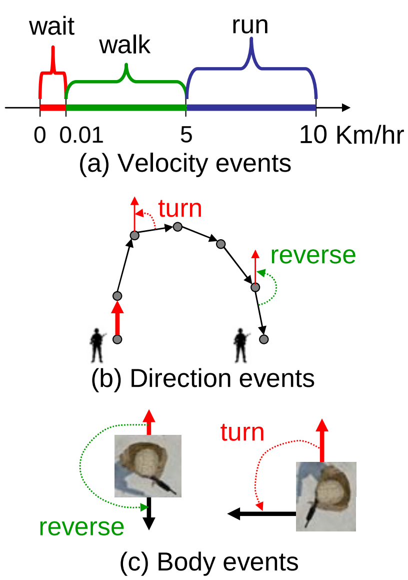


Figure 4. Examples of locomotion detection

The *interaction detection* algorithm generates events of relations between actors, from a fixed set of event types. The algorithm is rule-based (no learning is needed), similarly configured using a set of parameters. The following set of interaction event types is detected by our OPFOR agent:

- **Approach/Retreat**: a person approaches/ retreats from another entity over time
- **Vary_distance**: the distance between entities vary significantly over time
- **Keep_distance_close/far**: a close/medium distance between entities is maintained
- **Meet**: entities approach each other and meet
- **Enter/exit**: a person enters or exists the building or a car
- **Pass_by**: a person passes another entity
- **WalkTogether**: two people walking side-by-side in the same direction

Figure 5 shows how several interactions may be detected based on the distance between the avatars.

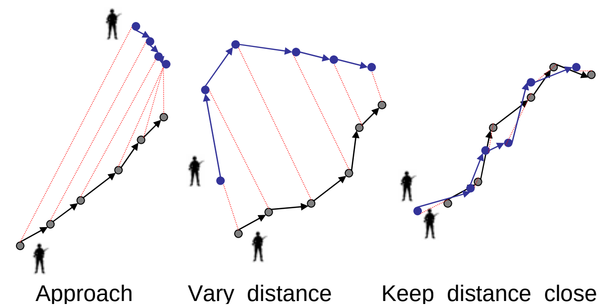


Figure 5. Examples of interaction detection

Action Recognition

Actions are higher-level behaviors of individual entities, spanning tens of seconds to minutes in the game. Actions are comprised of locomotions, and as such can be modeled and recognized. In our OPFOR agent, we use statistical classifiers for action recognition which must be first trained off-line on a set of motion data to build the model for each action type. These models are then used to recognize the actions during the actual scenario. Statistical learning capability enables the OPFOR agent to learn over time new actions that BLUEFOR avatars might perform.

We have implemented two statistical classifiers: Hidden Markov Models (HMMs) and Probabilistic Latent Semantic Indexing (PLSI). HMMs are good for modeling and detecting activities consisting of locomotions that have structure in time (i.e., strong sequential dependencies). PLSI is a good method for "bag of locomotions" modeling (i.e., when the order among locomotions does not carry significant information but the types of constituent locomotions matter). Both algorithms can be trained in a semi-supervised manner. We are currently enhancing our models, and plan on integrating Support Vector

Machines, K-nearest neighbors, n-grams, and Naïve Bayesian classifiers for use in action recognition.

PLAN RECOGNITION MODEL

The plan recognition algorithms in our OPFOR agent find the mission plan that BLUEFOR is following and the locations where the tasks are and will be executed by BLUEFOR. Plan recognition algorithms, see (Levchuk et al., 2008) for more details, take as inputs the set of action and interaction events as well as the static terrain information in the form of areas, their capabilities and relationships. The list of feasible BLUEFOR plans is defined manually and becomes hypotheses for the mission plan recognition algorithms, which test each BLUEFOR mission plan against the observations. The observations (motion events) and static terrain data are aggregated into a data network. The plan nodes (tasks) are mapped to the area nodes in the data network (Figure 6). This mapping is unknown and needs to be found. The objective function is based on the posterior probability of the plan given the observations.

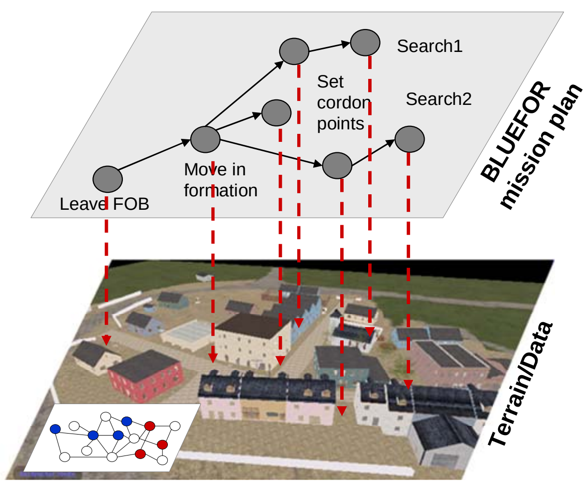


Figure 6. Plan recognition using spatiotemporal mapping

The data nodes are areas in the terrain that are predefined in advance with their polygons. Relations between areas are computed using attributes of distance from one area to another and coverage parameters (in terms of % of the area that can be used to fully observe another area, partially observe, or have no observation to another area). The coverage parameters influence how the BLUEFOR and OPFOR avatars may want to position themselves with respect to other avatars (including their own). For example, during a cordon and search mission BLUEFOR would want to set cordon positions that have overlook at the locations of searches – to enable the units at cordon points to respond quickly to a changing situation providing support to the unit that conducts the search operation.

OPFOR PLAN DESIGN AND LEARNING

From the perception component, OPFOR agent generates probabilistic estimates of the BLUEFOR plans and the mapping from those plans to the geographic areas. Each area thus gets a vector of estimates of BLUE task attributes. Using these attributes, we can select an OPFOR plan that can result in the best gain for OPFOR maximizing the reward objective function based on the operations success and entity health status.

Each plan in OPFOR's library consists of the set of nodes and links, similarly to the BLUEFORE mission hypotheses. One of the nodes is the BLUEFOR operation for which the OPFOR plan is designed. Other nodes are operations to be conducted by OPFOR. OPFOR can execute multiple plans in parallel.

In Figure 7, we show an example of the OPFOR plan and its mapping to the geographic terrain. The plan consists of 3 snipers and an action to secure egress for RED avatars. We employ a similar algorithm for mapping this plan onto the geospatial terrain, using inferred BLUE action and interaction events.

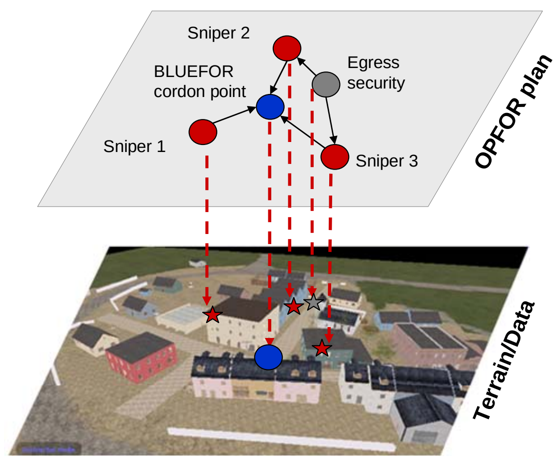


Figure 7. Planning OPFOR actions

Both BLUEFOR plan recognition and OPFOR plan design models use the same algorithms (Levchuk et al., 2008) that incorporate local static and dynamic information about the geographic area and similar information encoded in the models of the plans. Figure 8 conceptually describes how this can be achieved. We illustrate here that there are two considerations for finding the activities and their locations. First, the locations of activities are chosen based on the functions of geographic areas; such information can be encoded in the function requirements of the plan model. Second, when the task from a plan occurs, it generates events in the corresponding area. Since those events are often ambiguous, and function information alone creates too many alternatives; hence, using both event (dynamic) and function (static) attribute can improve the accuracy of activity-to-location association. In addition, information about functions of areas can be used in forecasting where future activities will take place in the future.

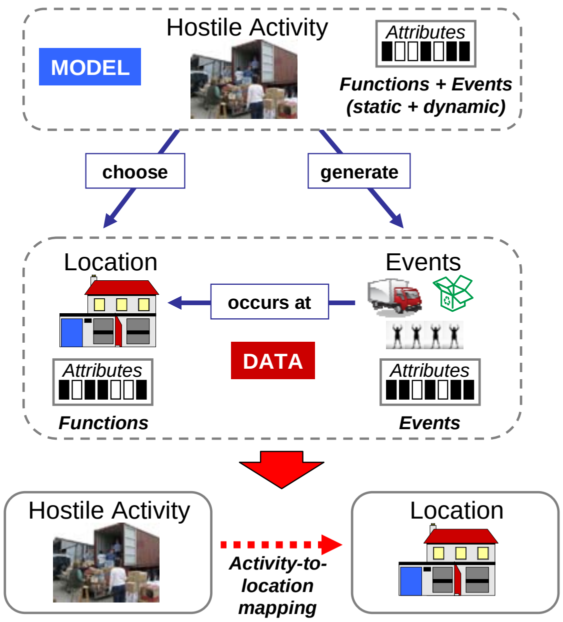


Figure 8: Considerations for activity-location mapping

Consequently, the full mapping objective function includes the mismatches between function and event components. While the event components are dynamic, they are moderately independent of the OPFOR preferences since events describe how specific operations could be observed through motion data. The function components, on the other hand, specify the preferences of selecting areas, and can change over time. These preferences constitute the primary drivers of the OPFOR adaptive behaviors, as the static function information used in the OPFOR planning corresponds to the inferences about BLUEFOR actions in the areas. Thus, these parameters correspond to the preferences of OPFOR to take their actions against BLUEFOR operations. Our OPFOR agent then learns and updates these preferences over time using temporal difference learning that incorporates the success scores of OPFOR actions. Initially, OPFOR tries to explore a set of feasible actions, over time converging to a set of most efficient preferences. The OPFOR adaptation is then equivalent to changes in OPFOR action-reaction preferences.

RESULTS AND CONCLUSIONS

Our OPFOR agent has been integrated with GDIS simulation, receiving its location and motion messages and sending back the motion and action commands for OPFOR avatars (Figure 9). The BLUE force is played by the human players or scripted avatars that execute the given plan. We then compare this plan with inferences produced by the OPFOR agent, as well as calculate the success of OPFOR and BLUEFOR operations.

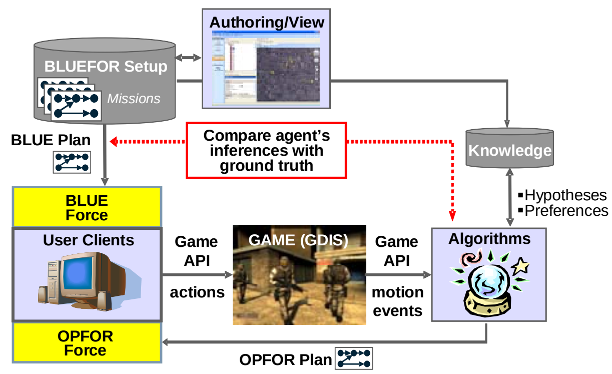


Figure 9. OPFOR agent architecture

We are currently developing and testing several tactical vignettes occurring in a small urban town which is a modification of the McKenna urban training site located at Ft. Benning. We have enhanced the terrain in our scenario level so that it contains more buildings and walls to provide more opportunities for the BLUE and RED forces to setup and execute missions. In our previous research, we assessed capabilities of the OPFOR agent to recognize the missions and task locations of the BLUE force, achieving over 75% accuracy of activity-to-location association (Levchuk et al., 2008, 2009). Our next steps are assessing the ability of OPFOR agents to adapt over time to increase rewards of their operations depending on the rate of adaptation of BLUEFORE. We expect that, in the presence of repeated actions by the BLUE force team participants, the OPFOR agents would achieve the mission success rates similar to human players in few iterations. Such functionality will provide automated technology that could improve adaptation training and increase the realism of the Wargaming and semi-automated mission rehearsal technologies.

REFERENCES

Avrahami-Zilberbrand, D., and G. Kaminka, "Fast and complete symbolic plan recognition", Proceedings of IJCAI, 2005

Blythe, J., "Decision-Theoretic Planning", AI Magazine, 20(2), 1999.

Brams, S. J., and Kilgour, D. M. (1988). Game Theory and National Security. New York: Basil Blackwell.

Geib, C.W., and Harp, S., "Empirical analysis of a probabilistic task tracking algorithm", AAMAS workshop on Modeling Other agents from Observations, (MOO-04), 2004.

Gerner, D.J., P.A. Schrodt, Ö. Yilmaz, and R. Abu-Jabr, "Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for a Post Cold War World", 2002 Annual Meeting of the American Political Science Association, 29 August - 1 September 2002

Kaminka, G., and Pynadath, D., "Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach", Journal of Artificial Intelligence Research, 17 (2002), 83-135

Levchuk, G., D. Lea, and K. Pattipati, "Recognition of Coordinated Adversarial Behaviors from Multi-Source Information", Proceedings of SPIE Defense and Security Symposium, Volume 6943, Orlando, FL, (2008).

Levchuk, G., D. Grande, K. Pattipati, Y. Levchuk, A. Kott, "Mission Plan Recognition: Developing Smart Automated Opposing Forces for Battlefield Simulations and Intelligence Analyses", Proceedings of the 13th International Command and Control Research and Technology Symposium, 2008.

Levchuk, G., O. Mendoza-Schrock, and W. Shebilske, "Adversarial Behavior Recognition from Layered and Persistent Sensing Systems", Proceedings of SPIE Defense and Security Symposium, Orlando, FL, (2009).

Mao, W., and Gratch, J., "Decision-theoretic Approach to Plan Recognition", ICT Technical Report, ICT-TR-01-2004, available at http://ict.usc.edu/files/publications/ICT-TR-01-2004.pdf

Popp, R., D. Allen and C. Cioffi-Revilla, "Utilizing Information and Social Science Technology to Understand and Counter the 21st Century Strategic Threat," in Emergent Information Technologies and Enabling Policies for Counter-Terrorism, (Eds. R. Popp, J. Yen), Wiley-IEEE Press, pp. 1-27, Jun 2006.

Schrodt, P.A., and D.J. Gerner, (2001) Analyzing International Event Data: A Handbook of Computer-Based Techniques, Draft Book; available at http://web.ku.edu/keds/papers.dir/automated.html

Singh, S., Allanach, J., Tu, H., Pattipati, K.; Willett, P., "Stochastic Modeling of a Terrorist Event via the ASAM system", Proceedings of IEEE Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, Oct. 2004

Shi, Y., Y. Huang, D. Minnen, A. Bobick, and I. Essa, "Propagation Networks for recognition of partially ordered sequential action", Proceedings of IEEE Computer Vision and Pattern Recognition, Washington, DC, June (2004).

Tu, H., Allanach, J., Singh, S., Pattipati, K.R., and Willett, P., "Information Integration via Hierarchical and Hybrid Bayesian Networks", IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 36, No. 1, 2006, pp. 19-33

Turchin, P., Historical Dynamics: Why States Rise and Fall, Princeton: Princeton University Press, 2003

Yin, M.-H., Ou, H.-J., Gu, W.-X., Lu, Y.-H., Liu, R.-X., "Fast probabilistic plan recognition without plan library", Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Vol. 1, Issue , 26-29 Aug. 2004, pp. 157 – 162

The Research Network, 2010, http://resrchnet.com/products.htm

Wikipedia 2010, List of game engines, http://en.wikipedia.org/wiki/List_of_game_engines