# An Influence Diagram Approach for Multiagent Time-Critical Dynamic Decision Modeling

Le Sun[1], Yifeng Zeng[2] and Yanping Xiang[1]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China
{sunle2009, xiangyanping}@gmail.com
[2] Dept. of Computer Science, Aalborg University, DK-9220 Aalborg, Denmark
yfzeng@cs.aau.dk

**Abstract.** Recent interests in multiagent dynamic decision modeling in partially observable multiagent environments have led to the development of several representation and inference methods. However, these methods have limited application under time-critical conditions where a trade-off between model quality and computational tractability is essential. We present a formal representation for modeling time-critical multiagent dynamic decision problems through interactive dynamic influence diagrams. The proposed model, called interactive time-critical dynamic influence diagrams, has the ability to represent space-temporal abstraction in multiagent dynamic decision models. More importantly, we take the notion of object-orientation design and make the representation flexible and reusable. The new design facilitates the modeling and implementation of models' self-expansion and self-compression.

**Keywords:** Time-Critical Decision Making, Multiagent Systems, Model Construction.

## 1 Introduction

Timely action is often critical in facing rapid changes in the real world. The time-critical dynamic decision problem is to decide or select a course of actions that shall achieve a set of goals while they must be executed under time constraints. It may be considered as a real-time decision problem[1] that seeks an optimal trade-off between solution quality and solution time via the use of the most appropriate model and solution algorithm. There is a growing line of interest, mainly on a single-agent setting, for addressing time-critical dynamic decision problems [2], [3], [4]. Most of previous work adopts a type of normative systems, e.g. Bayesian networks and influence diagrams [2]. Recently, Xiang and Poh [3], [5] proposed a formal representation of time-critical dynamic influence diagrams that provide explicit support for the modeling temporal processes and dealing with time-critical situations. Their work is applicable in a single-agent decision domain.

Time-critical decision modeling is more significant for multiagent applications due to the complicated decision process and solutions. The modeling of time plays an important role in a multiagent domain since multiple agents may interact with each other over time. Our interest in time-critical multiagent systems is motivated by the emergence of several applications including anti-air defense domain[6], Robocup[7] and multi-player online games[8]. Additionally, a suitable set of time-critical decision making techniques would allow multiple agents to coordinate their actions within a time limit so that individual rational actions do not adversely affect the overall system efficiency [9].

The purpose of this paper is to present a form technique for modeling multiagent time-critical dynamic decision problems. We rest on the representation of *interactive dynamic influence diagrams* (I-DIDs) [10], and propose a formalism called *Interactive time-critical dynamic influence diagrams (I-TCDIDs)* that provide explicit support for the modeling of time in the representation. I-TCDIDs focus on both data abstraction and model abstraction. For a given domain, a suite of decision models at different levels of space-temporal abstraction may be specified by either domain experts or knowledge engineers, and then organized in a knowledge base. In general, each abstract model may be solved by a number of algorithms and the choice of abstract model and algorithm will affect the final decision quality and computational cost.

Similar to I-DIDs, the construction of I-TCDIDs is a tedious job. The models can only be used in the specific domain for which it was created because each node corresponds to some domain attributes and the set of nodes and the network structures are fixed in advance [11]. The thing becomes more complicated when we need to expand models over time[1]. The construction becomes intractable in a complex problem domain. We therefore take the notion of object-orientation to design an efficient representation scheme for I-TCDID*s*. The proposed design reduces the implementation complexity of the problem and makes possible the models' self-expansion and self-compression. We illustrate the design through an anti-air example and show the utilization of I-TCDIDs.

## 2 Background

I-DID provides a relatively efficient method for representing multiagent sequential decision problems [10]. Its static model, called interactive influence diagram (I-ID), extends influence diagrams by introducing a new model node. We show one example of I-ID in Fig. 1(a). The I-ID model is constructed from the viewpoint of agent *i* that interacts with agent *j*. The model node, $M_{j,l-1}$, contains possible computable models of other agent like $m_{j,l-1}^1, \ldots ,m_{j,l-1}^n$ in the low level *l-1*. Solutions of all models are weighted by agent *i*'s beliefs on *j*'s models and aggregated into chance node $A_j$ (via *the policy link*). The issue becomes complicated when I-ID is expanded into I-DID over time. As agent *j* may act and receive observations, its models need to be updated to reflect their new beliefs. We assume the model node at time *t*, $M_{j,l-1}^t$, contains two

[1] To the best of our knowledge, one of the most efficient decision tool HUGIN (www.hugin.com) does not implement the expansion functionality that would automatically expand Bayesian Networks/influence diagrams to dynamic diagrams. I-DIDs are heavily built on HUGIN API.

$j$'s models ($m_{j,l-1}^{t,1}$ and $m_{j,l-1}^{t,2}$), and show the model update in Fig. 1(b). Since agent $j$ may receive any of $|O_j|$(=2) possible observations the updated set at time $t+1$ will become 4 models ($m_{j,l-1}^{t+1,1}$ ,…, $m_{j,l-1}^{t+1,4}$). The four models differ in their initial beliefs. The distribution over the updated set of models in the chance node $Mod[M_j^{t+1}]$ depends on the distributions over $j$'s action and observation that led to these models, and the prior distribution over the models at time step $t$. More details about I-DID refers to [10] due to the limited space here.

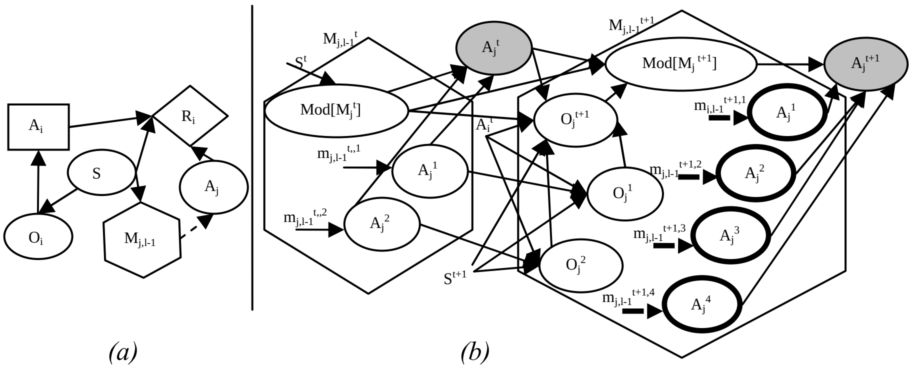

**Fig. 1.** (a) A generic level $l > 0$ I-ID for agent $i$ with a model node ($M_{j,l-1}$) and the policy link represented by the dashed arrow. (b) Model update from $t$ to $t+1$. $Mod[M_j^t]$ has the number of $j$'s models as its values. Notice the growth of models in the model node at $t+1$ in bold.

Current design of I-DIDs or most probabilistic graphical models are not essentially rooted in the object-oriented paradigm. We perceive that object-orientation conception would improve the current design and implementation. Here it is necessary to cover some of basic concepts. In the object-oriented paradigm the basic component is an object, an instance of a class. A class is a description of objects with common structures, behaviors and attributes, and has an associated set of nodes, connected by links. In addition to usual nodes in probabilistic graphical models, a class may also contain special nodes, called instance nodes, representing instances of other classes. A class instance represents a network containing three sets of nodes as defined in HUGIN: input nodes, output nodes and protected nodes. Input nodes and output nodes are the class interfaces and used to link the class instances to other network fragments. They must only be decision or chance nodes. Protected node is the node that only has parents and children inside the class. It can be all kinds of nodes.

## 3 Knowledge-Based Interactive Time-Critical Dynamic Influence Diagrams(I-TCDIDs)

I-TCDID is a formalism designed to facilitate the modeling and solution of multiagent time-critical dynamic decision problems. It extends I-DID by including the concepts of temporal arcs and time sequences. Furthermore, I-TCDID incorporates the object-oriented conception to realize models' self-expansion and self-compression. Repetition of identical structures is avoided for dynamic modeling.

An I-TCDID is defined as an instance of inference class which is composed of instances of time-slice class (i.e. time-slice instances). Given the initial information (specified by domain experts), the instance of inference class (inference instance) realizes dynamic decision modeling, and gets the optimal policy. As mentioned in Section 2, I-DIDs introduce a specific model node representing other agents' models and the models are expanded over time. This would become inflexible and redundant because other agents' may be abstracted in I-TCDID and we do not need to consider other agents' actions in every time step. I-TCDIDs address this gap by allowing the representation of other agents' models as the values of instances of agent class.

Each node in an I-TCDID represents a set of time-indexed variables (including instance node). The set of time indices may be different from one node to another, but they must be a subset of a master time sequence. The arcs in an I-TCDID are called temporal arcs and they denote both probabilistic and temporal (time-lag) relations among the variables. I-TCDID allows for the coexistence of nodes with different temporal information in the same model.

A formal definition of interactive time-critical dynamic influence diagram is given below.

**Definition 1.** An Interactive time-critical dynamic influence diagram I-TCDID=<**D**, **C**, $V$, **Ai**,**At**, **P**, **AC**, **TC**, $T_m$> where:

**D** is a set of *temporal decision variables*. Each $D \in \mathbf{D}$ is a sequence of decision variables indexed by a time sequence $T_D$ and is represented in the graph by a square node.

**C** is a set of *temporal chance variables*. Each $C \in \mathbf{C}$ is a sequence of chance variables indexed by a time sequence $T_C$ and is represented in the graph by an oval node.

$V$ is a *temporal utility variable*. It is a sequence of utility functions indexed by a time sequence $T_V$ . $V$ is represented in the graph by a diamond node.

$\mathbf{Ai} \subseteq (\mathbf{D} \cup \mathbf{C}) \times (\mathbf{D} \cup \mathbf{C} \cup \{V\})$ is a set of *instantaneous arcs* such that $(X,Y) \in \mathbf{Ai}$ if and only if there exists an instantaneous arc from node $X \in (\mathbf{D} \cup \mathbf{C})$to node $Y \in (\mathbf{D} \cup \mathbf{C} \cup \{V\})$. An *instantaneous arc* is represented in the graph by a solid directed arc.

$\mathbf{At} \subseteq (\mathbf{D} \cup \mathbf{C}) \times (\mathbf{D} \cup \mathbf{C} \cup \{V\})$is a set of *time-lag arcs* such that $(X,Y) \in \mathbf{At}$ if and only if there exists a time-lag arc from node $X \in (\mathbf{D} \cup \mathbf{C})$ to node $Y \in (\mathbf{D} \cup \mathbf{C} \cup \{V\})$. A *time-lag arc* is represented in the graph by a directed dashed arc.

**P** is a set of *conditional probability distributions*. For each chance node $X \in \mathbf{C}$, we assess a sequence of conditional probability distributions $p(X_i|\pi(X_i))$ where $i \in T_x$, $\pi(X_i)=\{Y_j|(Y,X) \in \mathbf{At}, j=\max\{k|k \in T_Y, k<i\}\} \cup \{Y_j|(Y,X) \in \mathbf{Ai}, j=i\}$.

**AC** is a set of *agent instance* (i.e. instances of *agent class*). Each $AC \in \mathbf{AC}$ is an agent instance indexed by a time sequence $T_{Ac}$.

**TC** is a set of *time-slice instance* (i.e. instances of *time-slice class*). It is defined as $TC$=<**D**,**C**,$V$,**AC**,**Ai**,**At**,**P**>. Each $TC \in \mathbf{TC}$ is a time-slice instance indexed by a time sequence $T_{TC}$.

$T_m$ is the *master time sequence*. A time sequence is a set of time indices represented by $<t_1,t_2,\ldots,t_n>$, where $t_1$ is the initial time point of interest and $t_n$ is the last time point of interest. Let $\mathbf{T_D}=\{T_D|D\in\mathbf{D}\}$,$\mathbf{T_C}=\{T_C|C\in\mathbf{C}\}$, $\mathbf{T_{AC}}=\{T_{AC}|AC\in\mathbf{AC}\}$, $\mathbf{T_{TC}}=\{T_{TC}|TC\in\mathbf{TC}\}$ and $T_V=T_m$. Each time sequence must be a subsequence of the master time sequence.

**Definition 2.** Given a time-slice class *TC,* we define an *inference class IC*=<INF, *TC*>, where INF is the *initial information.*

I-TCDIDs are graphical models of instances of inference class. For a given domain, an inference class and a suite of decision models at different levels of space-temporal abstraction of inference class may be specified by either domain experts or knowledge engineers, and then organized in a knowledge base. The abstraction is organized in the form of initial information which can be used to initialize an inference instance. Initial information contains time index of both nodes, state variables and embedded models. The selected information is used to index the nodes in a time-slice class and time-slice instance nodes.

### 3.1 Agent Class

Agent class models common domain structures, behaviors and attributes in the domain. It is an inference model with input nodes, output nodes and protected nodes.

**Interactive Object-Oriented Influence Diagrams（I-OOIDs）.**I-OOIDs generalize I-IDs to abstract agents' models as an agent class. Each agent's inference model is an agent instance that includes a set of its intentional and subintentional models and has the agent's optimal policy as the output. In I-OOIDs, an agent instance corresponds to an agent instance node and is represented as a dashed rectangle. For the simplicity of presentation, we consider two agents, *i* and *j*, which are interacting in a common environment. Fig.2 shows an example of one agent, say *i*'s, I-OOID with an instance node $C_{j,l-1}$ of another agent *j*. This is agent *i*'s inference model in the strategy level *l*. For more than two agents, we could have an agent instance node for each other agent. The new instance nodes are conditioned on the physical state and possibly instance nodes of other agents while they are linked to the utility node.

**Agent Class.** As shown in Fig. 2, the dashed rectangle represents agent *j*'s instance node. The input node *S*, which is represented by an oval with heavy grey border, is a set of current state variables. Input nodes represent nodes that are actually not in the class; they act as place-holders for parents of nodes inside instances of the class. The output node $A_{j,l-1}$, which is represented by a gray (fill-in) oval node, is a set of optimal actions of agent *j*.

We can see from Fig. 2, the input and output nodes constitute the interface which interacts with the surroundings of the instance. The protected nodes could not interact with other nodes outside of the instance node. The state node *S*, connected to the input node *S* of agent instance, is the node that is the actual parent of the children of the input node *S*. *Default potential*[12] is provided if an input node do not have a parent node.

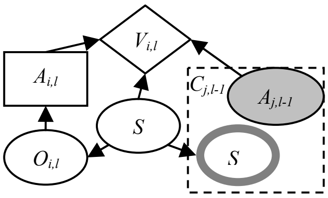


**Fig. 2.** Agent *i*'s I-OOID including agent *j*'s instance node $C_{j,l-1}$.

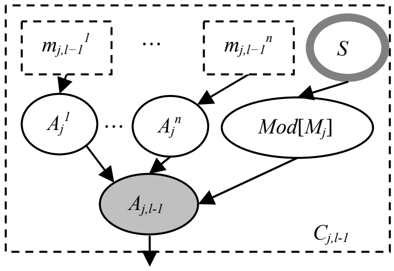


**Fig. 3.** The deployed form of agent *j*'s instance node with several computational models($m_{j,l-1}^{1}$,…, $m_{j,l-1}^{n}$) instantiated from *j*'s agent class.

The agent instance node contains as its values the alternative computational models ascribed by *i* to the other agent. The detailed agent instance node is shown in Fig. 3. The nodes surrounded by dashed rectangles, $m_{j,l-1}^{1}$,…, $m_{j,l-1}^{n}$, are computational models in level *l-1*. Each computational model is an instance of *j*'s agent class in the low level. Hence agent class is defined in a recursive way.

### 3.2 Time-Slice Class

The basic building block of I-TCDID is a one time-interval model of a specific domain. The one time-interval network fragment constitutes a class called time-slice class and each specific time-interval model is an instance of time-slice class. The arcs are called temporal arcs and they denote both probabilistic and temporal (time-lag) relations among the variables. The framework of time-slice class is shown in Fig. 4. The nodes with heavy grey border in Fig. 4 are input nodes. The input nodes, $S^{t-1}$, $A_{i,l}^{t-1}$, $C_{j,l-1}^{t-1}$, represent the belief states, actions of agent *i*, and the instance model of agent *j* in the previous time step respectively. The grey nodes, $S^t$, $A_{i,l}^{t}$, $C_{j,l-1}^{t}$, are output nodes which represent a set of corresponding variables at the current time step. Solid arcs are instantaneous arcs and dashed arcs are time-lag arcs that model relationships between nodes in continous time-slices. For instance, the dashed arc between $S^t$ and $S^{t-1}$ represents the physical states in current time-slice influencing that of next time-slice.

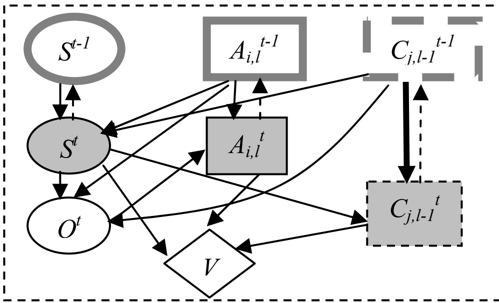


**Fig. 4.** A generic level $l$ Time-slice class for agent $i$. Notice the model update arc represented by solid bold arc denotes the update of the models of $j$ and of the distribution over the models, over time.

The nodes $C_{j,l-1}^{t-1}$ and $C_{j,l-1}^{t}$ are not the actual input and output nodes. The arcs, coming from and going to the agent instance node, are called *influential arcs.* For instance, the arcs from $C_{j,l-1}^{t-1}$ to $S^t$ and $O^t$ only represent the influencial relationships in Fig. 4. The solid bold arc from $C_{j,l-1}^{t-1}$ to $C_{j,l-1}^{t}$ is a new arc called *model update arc* (the time-indexed model update link [10]) reflecting updates of models in agent instance node between two continouse time-slices. The updated model node demands only the place-holders $S$ and instances of agent $j$'s classes e.g. $m_{j,l-1}^{t}$ and so on. The model update arc may be replaced by the dependency arcs between the chance nodes that constitute the agent instance nodes in the two time-slices. The influential arcs coming from the rest fragment of the time-slice class and connecting to the agent instance node can be seen as the arcs connecting to the input node $S$ of agent instance node. This is the same as output arcs of agent instance node.

### 3.3 Inference Class

The inference process of interactive time-critical dynamic modeling is defined as an inference class that is index by a time sequence. We show an instance of agent $i$'s top-level inference class in Fig. 5. Four time-slices ( $t_1$, $t_2$, $t_3$, $t_4$) are considered and the master time sequence is $<1,2,3,4>$. $S_1$ and $S_2$ are a set of input state variables that are indexed in a different way. For example, the node $S_1$ is indexed by the time sequence $<1,2>$. Consequently, the state variables in node $S_1$ are only considered in time-slices *t1* and *t2* and are referenced by the input nodes in time-slice instance *t1* and *t2*. The output is agent $i$'s policy, grey color fill-in rectangle node $policy_i$.

$M_j^0$ contains agent $j$'s models ascribed by agent $i$. Recall that the agent instance node contains all candidate models of other agents. These models may themselves be inference instances (the static models, I-OOIDs, IDs etc., initialized by initial information) leading to recursive modeling. They may be abstracted in a different way. This requires to index each model with a unique time sequence in the agent instance node. Assume that agent $j$ has two candidate models, $m_j^1$ and $m_j^2$. we show one example of initial information about agent instance node with different time-indexed models in Fig. 6.

In this case, model $m_j^1$ is indexed by the subtime sequence <*1,3*> while $m_j^2$ is indexed by <*1,2*>. We may also index the instance node using a single time sequence if all models share the same sequence. This is exactly the case for the initial information in Fig. 5 where $M_j^0$ is time-indexed by <*1,3*> and all models have the same time sequence <*1,3*>. In this case, Agent *j* may not be considered in time sequence <*2,4*> for its negligible influence. Agent *j* may take actions for fewer time steps and play an intervention only at the indexed times. This means that agent *j* has been temporally abstracted by omitting its value at some intermediate time indices.

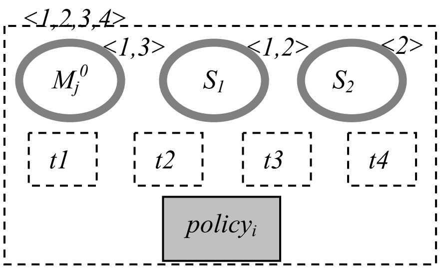


**Fig. 5.** An instance of Agent *i*'s inference class with four time-slice instances(*t1, t2, t3, t4*).

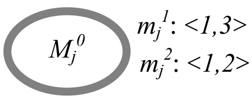


**Fig.6.** Initial information about agent instance node in which two models, $m_j^1$ and $m_j^2$, have different sub-time sequences < *1, 3* > and < *1, 2* > respectively.

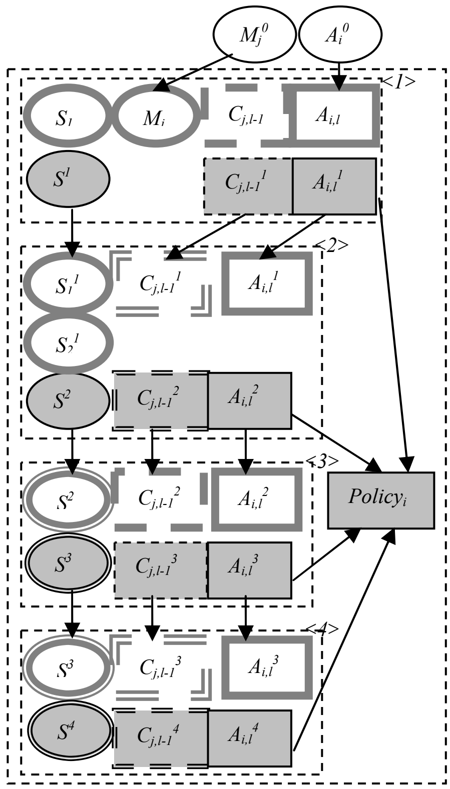


**Fig.7.** The deployed form of inference instance of agent *i.* Nodes with double circles are deterministic nodes.

The deployed process of inference instance is shown in Fig. 7. We repeat a normal node (expect the model node) only if its time sequence is equivalent to the master time sequence; otherwise, it will be casted into a deterministic node (which is deterministically dependent on its parent nodes) for the time step where the index value is omitted from the time sequence. For the agent instance node, we update the model only at the time step if the time is indexed in the time sequence to the model inside the model node. Otherwise, we retain all models from the previous time step and do not perform any model update - we also mark the instance node using the type of deterministic nodes. There is no solutions (actions performed by agents) from the model at a particular time step which is not indexed in the time sequence. For facilitating the CPT setting of action node $A_{j,l-1}$ (in Fig. 2), we assume a uniform distribution of actions from the model, e.g. assigning the probability $1/|A_j|$ to the columns corresponding to the model.

The policy node as the output node of inference class represents agent *i*'s optimal policy. Its parents are the output nodes of time-slice instances included in inference class. Time-slice instances are related to each other: the current time-slice instance influences the next time-slice instance. This is evident in the deployed form in Fig. 7 There are only two agent *j*'s instance nodes $C_{j,l-1}^{1}$ and $C_{j,l-1}^{3}$, and nodes $C_{j,l-1}^{2}$ and $C_{j,l-1}^{4}$ are assumed to be deterministically dependent on (possibly equal to) nodes $C_{j,l-1}^{1}$ and $C_{j,l-1}^{3}$ respectively. In addition, CPTs of node $A_{j,l-1}^{2}$ in the agent instance node $C_{j,l-1}^{2}$ have a uniform distribution.

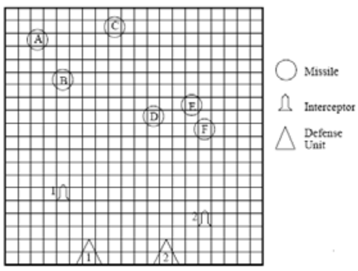


**Fig. 8.** The anti-air defense scenario.

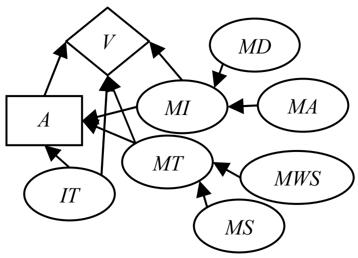


**Fig. 9.** Level *0* model of agent *j*. Note that agent *i*'s instance node is not included.

## 4 Case Study

We illustrate the usefulness of I-TCDIDs on an anti-air defense domain. The problem is a slightly modified version of the anti-air denfense game[6]. The game has six incoming missiles and two defenders, say agents *i* and *j*, to intercept the missiles in a 20 by 20 grid world. We show the game scenario in Fig. 8. Each of the agents decides to intercept the coming missiles to minimize the damages. Since no communication exists between the two agents they need to model each other to coordinate their interception decisions. This would avoid a redundant target at the same threat.

Let us consider a particular setting of this problem in which agent *i* considers two distinct level *0* models of *j*. The two models may differ in the probability that agent *j* owns interceptor types. Agent *i* makes decisions in a master time sequence <*1,2,3,4*> considering agent *j*'s actions only in time-slice <*1,3*>. Agent *j*'s two models are all time-indexed by the sequence <*1,3*>. Agent *j* is assumed to act randomly except for observing in time-slice <*2,4*> from the eyes of agent *i*. Agent *j*'s model is constructed as shown in Fig. 9. The missile's warhead size (*MWS*) and speed (*MS*) and the distance (*MD*) and angle (*MA*) between the missile and the agent are the relevant features influencing the agents' decisions. As agent *j*'s observations, the missiles' threat(*MT*) and interceptorbility(*MI*, the likelihood of intercepting a missile) and the type of interceptor(*IT*) the agent has (long range or short range), directly impact the agent's utilities.

We may proceed to deploy and compute the inference instance after the initialization of I-OOID. The detailed structure of a time-slice instance and agent instance is shown in Fig. 10. Two models of agent *j*, $m_j^1$ and $m_j^2$ , are included in agent *j*'s instance node. Fig. 9 shows the model structure. They model two cases respectively: Agent *j* has both short and long range interceptors to shoot all the missiles; agent *j* has only long range interceptors that can only shoot down one of A, B, or C. An I-TCDID with two time-slice instances is shown in Fig. 11. $Missile^0$ is the missile class node that could be further instantiated into four missiles' variables (*MD, MA, MWS, MS*) coming from Radar-data. Since the $M_j^0$ node is time-indexed by<*1,3*>, the agent instance node $C_j^2$ is respresented using double-dashed rectangle which means $C_j^2$ is assumed to be deterministically dependent on (possibly equal to) nodes $C_j^1$. The models are not updated or expanded into $C_j^2$.

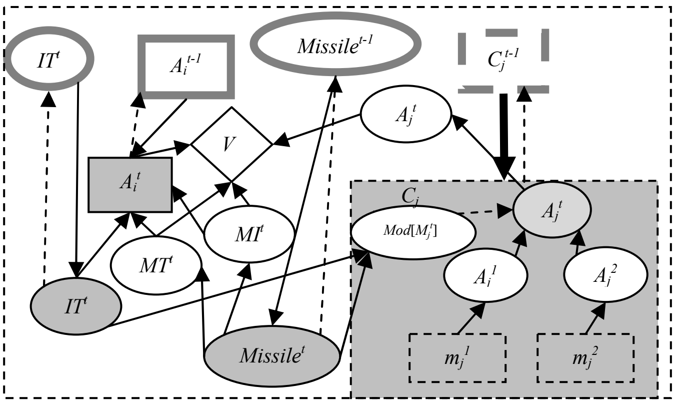


**Fig. 10.** The deployed time-slice instance node and agent *j*'s instance node with two computational models($m_j^1$ , $m_j^2$).

In Fig. 10, the agent instance node contains different inference instance models that are expanded from the level *0* IDs where nodes are time-indexed by the initial information. We get the probability distribution of *j*'s actions in chance node $A_j^1$ by solving the level *0* models of *j* in Fig. 11. On performing the optimal action(s) at time step *1*, *i* may receive observations of the missiles' information to speculate *j*'s information of interceptors. This is reflected in updated models and new beliefs corresponding to these models. Consequently,agent *j*'s instance node in the next time-slice contains more models of *j* and *i*'s updated belief on *j*'s possible inference models.

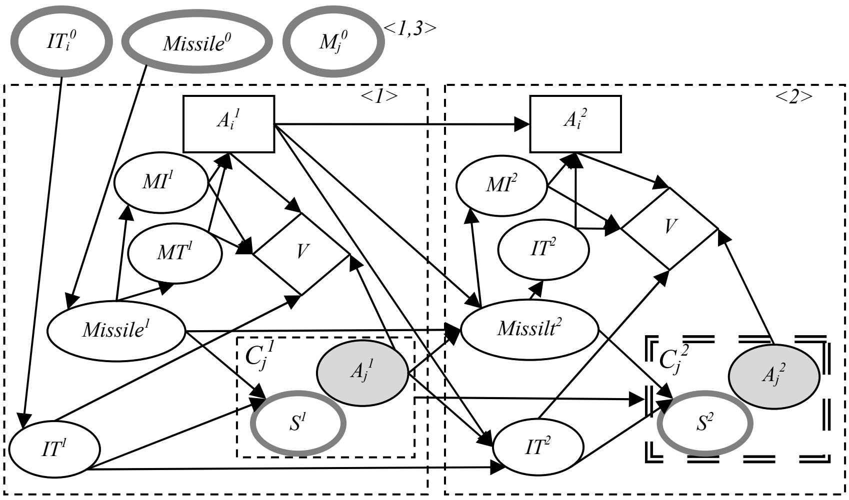


**Fig. 11.** An I-TCDID with two time-slice instances. The node $M_j^0$ is time-indexed by time sequence *<1,3>*, so $C_j^2$ is assumed to be deterministically dependent on (possibly equal to) nodes $C_j^1$.

## 5 Related Works

One of most relevant works is the representation of time-critical dynamic influence diagrams [3], [5]. Xiang and Poh [3] proposed two forms of time-critical dynamic influence diagrams: the condensed form and the deployed form. The condensed form is used in modeling process. The deployed form is the unfolded form and be mainly used for the inference purpose. The approach has successfully tackled some medical decision problems e.g. the treatment of cardiac arrest.

For the study of multiagent dynamic decision making, Noh and Gmytrasiewicz [6] investigated multiagent coordination in an anti-air defense domain and took the recursive modeling framework. Their method does not enjoy an explicit representation of problem domain and is difficult for a generalization. I-DIDs [10] have become a popular tool for modeling sequential multiagent decision problems. The modeling takes the viewpoint of individual agents and explicitly describes how other agents behave in agents' interaction. Extension to modeling time-critical decision problems would be a natural way as time-critical dynamic influence diagrams extend influence diagrams. However, the complicated representation needs to be ironed in a modular manner.

Object-oriented Bayesian networks [11] become a powerful representation technique that allows the modeling of complex domain. Bangso and Olesen [12] used the similar concept to solve a large medical decision problem. Its extension to influence diagrams sounds no straightforward. Currently, HUGIN takes the initiative and proceeds to the implementation.

## 6 Conclusion

We propose a formal model of I-TCDIDs to represent multiagent time-critical dynamic decision problems. The new technique uses an object-orientation concept to abstract the representation especially on the model expansion over time. It defines an instance of inference and time-slice class based on the concept of agent class. We show the space-temporal abstraction and agent abstraction in I-TCDIDs, and demonstrate one application in the anti-air defense game. Future work would be interesting to study the impact of initialization on the inference instance in I-TCDIDs.

## References


1. Séguin, R., Potvin, J.YJ., Gendreau, M., Crainic, T.G., Marcotte, P.: Realtime Decision Problems: An Operational Research Perspective. Journal of the Operational Research Society, 48, 162-174 (1997)
2. Horvitz, E., Seiver, A.: Time-critical action: Representations and application. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, pp. 250–257 (1997).
3. Xiang, Y.P., Poh, K.L.: Time-Critical Dynamic Decision Making. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 688-695(1999).
4. Cohen, I.: Improving time-critical decision making in lifethreatening situations: Observations and insights. Decision Analysis, 5, 100–110(2008)
5. Xiang, Y.P., Poh, K.L.: Knowledge-based Time-Critical Dynamic Decision Modeling. Journal of the Operational Research Society, 53, 79-87(2002)
6. Noh, S., Gmytrasiewicz, P. J.: Agent modeling in anti-air defense: A case study. In: Proceedings of the Sixth International Conference in User Modeling, pp. 389–400(1997).
7. Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, H., Osawa, E.: RoboCup: A challenge problem for AI. AI Magazine, 18, 73–85(1997).
8. Yee, N.: The Demographics, Motivations, and Derived Experiences of Users of Massively Multi-User Online Graphical Environments. Presence: Teleoperators and Virtual Environments, 15, 309-329(2006)
9. Bond, A.H., Gasser, L.(Eds.): Readings in Distributed Artificial Intelligence. Morgan Kaufmann, San Mateo, CA(1988)
10. Doshi, P., Zeng, Y., Chen, Q.: Graphical models for interactive pomdps: Representations and solutions. Journal of Autonomous Agents and Multiagent Systems (JAAMAS), 18, 376–416(2009)
11. Koller, D., Pfeffer, A.: Object-Oriented Bayesian Networks. In: Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 302-313(1997).
12. Bangsø, O., Olesen, K.G.: Applying Object Oriented Bayesian Networks to Large (Medical) Decision Support Systems. In: Proceedings of 8th Scandinavian Conference on Artificial Intelligence, Bergen, Norway(2003).