Modeling the Broker Behavior Using a BDI Agent

LAURA FLORENTINA CACOVEAN, FLORIN STOICA
Department of Computer Science
Lucian Blaga University of Sibiu, Faculty of Sciences
Str. Dr. Ion Ratiu 5-7, 550012, Sibiu
ROMANIA
laura.cacovean@ulbsibiu.ro, florin.stoica@ulbsibiu.ro

Abstract: - The static properties of BDI systems can be considered extensions of CTL logics. This type of logic will be model the mental state of agent. The logic is extended by defining additional connectors and operators. We present in this paper the mental BDI agents whose behavior is conduct by beliefs, desires, intentions, and decisions. The case study addresses the scenario of a broker whose operation is given by market conditions and demand, offer, price and number of operations it holds.

Key-Words: - CTL, specification, BDI, agents, model checker, broker behavior

1 Introduction
In the beginning of our work, we present the methodology and supporting sets of tools, what permitted the possibility to automatically generate the algorithms for verifying the models for temporal logics from a set of algebraic specifications. We chose the algebraic description for CTL model because we use sets of states for model verification. The development of temporal logics and the algorithms for model checking can be used to verify the properties of system. In the fourth section, we present a formal description of world’s formalism for architecture of BDI [3]. There are three elements of formalism. First, intentions are treated as a par with beliefs and desires. Second the agent can choose among outcomes and in the latter case, the environment makes that determination. Third, is specifying an interrelationship between beliefs, desires, and intentions that allows avoiding many of the problems usually associated with possible-worlds formalisms. In the fifth section we present a study case for determination the broker behavior using a BDI agent.

2 Algebraic description of CTL model
CTL is a branching time temporal logic meaning that its formulas are interpreted over all paths beginning in a given state of the Kripke structure. Definition of a Kripke model [4] let \( AP \) is a set of atomic propositions. A Kripke model \( M \) over \( AP \) is a triple \( M = (S,Rel,P:S \rightarrow 2^{AP}) \) where \( S \) is a finite set of states, \( Rel \subseteq S \times S \) is a transition relation, \( P:S \rightarrow 2^{AP} \) is a function that assigns each state with a set of atomic propositions.

In an algebraic compiler \( C:L_s \rightarrow L_t \) the source and the target language used are defined using heterogeneous \( \Sigma \)-algebras and \( \Sigma \)-homeomorphisms [6]. The operator scheme of a \( \Sigma \)-algebras is a tuple \( \Sigma = (S,O,\sigma) \) where \( S \) is a finite set of states, \( O \) is a finite set of operator names, and \( \sigma:O \rightarrow S^* \times S \) is a function which defines the signature of the operators. These signatures are denoted as \( \sigma(o)=s_1 \times s_2 \times \ldots \times s_n \rightarrow s \), where \( s_i \in S \), \( 1 \leq i \leq n \). A \( \Sigma \)-algebra is a tuple defined through \( L=\langle Sem,Syn,\mathcal{L}:Sem\rightarrow Syn \rangle \), where \( Sem \) is a \( \Sigma \)-algebra called the semantic language, \( Syn \) is a \( \Sigma \)-word or a term algebra called the syntax language and \( \mathcal{L} \) is a partial mapping called the language learning function. In order to define a CTL model as a \( \Sigma \)-language, shall define an operator scheme \( \Sigma_{ctl} \) as the tuple \( \langle S_{ctl},O_{ctl},\sigma_{ctl} \rangle \) where the states \( S_{ctl}=\{F\} \) are mapping the formulas \( f \) and \( O_{ctl} = \{true, false, \neg, \land, \lor, \rightarrow, AX, EX, AU, EU, EF, AF, EG, AG\} \). The model \( M=(S,Rel,P:AP \rightarrow 2^{AP}) \) is structure as a \( \Sigma \)-language whose syntax algebra contains the sets of the expressions and whose semantics algebra contains the sets of \( 2^S \). The operator scheme for this language is \( \Sigma_{set}=(S_{set},O_{set},\sigma_{set}) \). The operators in the algebra and their signatures as defined by \( \sigma_{set} \) are shown in [5]. We retain that semantic \( Sem_{sets} \) and \( Sem_{set} \) have the carrier sets in the relation \( Sem_{F} \subset Sem_{S} \). This allows due to simplicity to show in the scheme all elements of carrier sets of \( Sem_{set} \) through their occurrences in the carrier sets \( Sem_{set} \). We don’t detail the algebraic implementation of CTL because it’s presented in paper [5], but we retain that CTL rules set are directly specified in the algebra \( Sin_{ctl} \) can be ambiguous, therefore it is necessary that the set \( F \) from \( Sin_{set} \) to be divided in the non-terminal symbols denote with CTLformula, Formula, Factor, Termen and Expresie. Thus, the defined rules deliver non-ambiguous specification in algebra \( Sin_{ctl} \). The behavior of the model checker algorithm consists of identifying the sets of states of a model \( M \) who satisfy each of a sub formula of a given CTL formula \( f \) and
constructing the set of states, from these sets, that satisfy the formula $f$.

### 3 Algebraic implementation of CTL

A CTL model checker defined as an algebraic compiler $C: L_{ctl} \rightarrow L_{sy}$ by pair of embedding morphisms $(T_C, H_C)$. $T_C: \text{Syn}_{el} \rightarrow \text{Syn}_{sets}$ maps CTL formulas from word algebra $\text{Syn}_{el}$ to set expressions in $\text{Syn}_{sets}$, which evaluate to the satisfaction of sets of the CTL formulas, $H_C : \text{Sem}_{el} \rightarrow \text{Sem}_{sets}$, maps sets in $\text{Sem}_{el}$ by the identity mapping to sets in $\text{Sem}_{sets}$[6]. The morphisms $T_C$ and $H_C$ thus constructed make the diagram commutative. Commutatively assures the fact that $T_C$ keeps the meaning of the formulas from $\text{Syn}_{el}$ when mapping them to set expressions in $\text{Syn}_{sets}$. The diagonal is $\text{Delt}: \text{Sem}_{el} \rightarrow \text{Syn}_{sets}$ [9]. Construction of $T_C$ can be entered into an algorithm which implements the CTL model checker. This algorithm is universal in the sense that being given operator scheme $\Sigma_U$ and a model $M$, the model checker $L_{ctl}$ is automatically generate from the specifications of $(\Sigma_{ctl}, D_{ctl})$. This specification is obtained by associating each operation $o \in O_{ctl}$ with an derived operation $d_{ctl}(o) \in D_{ctl}$. To define derived operations that implement the operations of $\Sigma_{ctl}$ from the $\text{Syn}_{sets}$ algebra, we use meta-variables that take as values the set expressions of the carrier sets of $\text{Syn}_{sets}$. For each operation $o \in O_{ctl}$ such that $\sigma_{ctl}(o) = s_1 \times \ldots \times s_n \rightarrow s$, $d_{ctl}(o)$ takes as the formal parameters the meta-variables denoted by $@_i$, $1 \leq i \leq 2$, where $@_i$ denotes the set expression associated with $i$-th argument of $d_{ctl}(o)$; the meta-variable $@_0$ is used to denote the resulting set expression, as example $@_0 = d_{ctl}(o)(@_1, \ldots, @_n)$ [5,6].

In following we show the specification of AU formula

\begin{verbatim}
Ctlformula -> "af" Formula;
Macro: sets Z,Z1;
Z1:=@2; Z:= S;
while(Z not_equiv Z1) do
  Z:=Z1; Z1:=Z1 union \{s in S | (succ(s)subset Z1)\};
endwhile
@0:=Z1;
\end{verbatim}

This is the behavior of the algorithm for the homeomorphism computation performed by an algebraic compilers; Thus is evaluated an expression by repeatedly identifying its generating sub expressions and replacing them with their images in the target algebra. In the case of the model checking algorithm, sub expressions are CTL sub formulas and their images are the sets of states in the model satisfies the sub formulas.

### 4 Formal description of multi-agent system

The study of multi-agent system that we present below is that the agent possesses mental attitudes as beliefs, desires and intentions representing the informational state, motivation state and intention state of an agent. These attitudes determine the system behavior. Agents with mental attitudes are called BDI agents (belief-desire-intention) and systems consisting of these agents are called BDI systems. The agent states are represented using a decision trees. Based on these the semantic models results can provide axioms of BDI system representing a table with decision procedures for BDI logic [3]. The basic components of BDI architecture are data structures representing the beliefs, desires and intentions of the agent, and functions that represent its deliberation, deciding what intentions to have, and means ends reasoning, deciding how to do them. Using decision trees we may form a series of multi-agent systems characteristics like: any time exist a several ways of environment evolution; at any time maybe exist a several actions which can be executed; at any time maybe are a several objectives that must be met. Actions leading to the objectives are dependent on environmental conditions, and are independent of the internal state of the system. It is necessary that the system to hold information about environmental conditions. This information is called belief. This component, in our case, will be implemented as a set of logical expressions. Beliefs can be seen as an informative part of the state system. It needed the system to know information about the objectives who must to be fulfilled, that is, the priorities are associated with the current objectives. This information is called desire. They represent the motivational state of the system. Since the environment system may change during the selection function or execution action is required the third component of the system state to represent the current deployment action to fulfill. The result of the latest call is the result of selection function. The agent intentions are modeling by this state. All three components of the system will be described into a propositional form. Modeling system behavior is a tree structure. For agent each transition is a primitive action performed by the system. Nodes are called nodes of selection (decision). Primitive events occurring in the environment will be called chance nodes. This structure can be seen as a decision tree. An agent with the decision tree and transformations specified may be decided according to the selection function that is optimal way to execute the desired action. Static properties of BDI systems can be considered extensions of CTL logics. This type of logic will be model the mental state of agent. The logic is extended by defining additional connectors and
operators. This will be can model the beliefs and desires of agents. These are actions performed by agents. The model considered is based on a temporal tree. The paths represent the possible environmental histories. For an agent time is linear in past and branching in future. Nodes are the states and arcs are labeled with the primitive actions. The logics can be used to reason about agents and the way in which their beliefs, desire and actions can bring about the satisfaction of desires.

Following we present a multi-modal logic and propositional temporal BDI CTL. Primitives of these languages include a set of primitive propositions \( \Phi \), logical connectors \( (\neg, \land, \lor) \), temporal operators \( (A, E, X, F, U, G) \) and modal operators \( (\text{BEL}, \text{DES}, \text{INTEND}) \). The operators \( \text{BEL}, \text{DES}, \text{INTEND} \) represent the belief, the desire and the intentions of the agent. The modal operators \( \text{BEL} \) is write like \( \text{BEL}(\text{agent believes}) \), \( \text{DES} \) is write like \( \text{DES}(\text{agent desires}) \) and \( \text{INTEND} \) is write like \( \text{INTEND}(\text{agent intends}) \). With these connectors and operators can be formed the states formulas which express the properties of states system and the path formulas express the properties of path in tree.

The state formulas for BDI_CTL are given by following rules:

s1. each atomic proposition \( f \) is state formula;

s2. if \( f \) and \( g \) are state formulas then also \( \neg f, \neg g, f \land g \) are state formulas

s3. if \( f \) is a path formula then \( Af \) and \( Ef \) are state formulas

s4. if \( f \) are state formula then \( \text{BEL}(f) \), \( \text{DES}(f) \), \( \text{INTEND}(f) \) are state formulas

The state formulas for BDI_CTL are given by following rule:

p1. if \( f \) and \( g \) are path formulas, then \( Xf \), \( Ff \) and \( fUg \) are state formulas

The language BDI_CTL usually is used to represent the mental state of an agent. For example consider that an agent intending to buy sometime an operation in the future and who wants eventually to not decrease the value of an operation, but does not believe that will eventually happened. The mental state of this agent can be represented by the following formula: \( \text{INTEND}(E(\text{buy-an-operations})) \land \text{DES}(A(\text{decrease-the-value-of-an-operation})) \land (\neg \text{BEL}(A(\text{decrease-the-value-of-an-operation}))) \). Each formula is evaluated with according to a given world and an index into series of events that define the world. BEL relation, denoted with \( \mathcal{B} \), for example, is a relation between the world at an index to a set of worlds or series of events. That means, an agent believes a formula in a world at a particular index if and only if in all its belief-accessible worlds the formula is true. Same for DES and INTEND. We consider each possible world as a temporal tree that has one last time and multiple choices for the future. Each tree structure represents options of events can be chosen by an agent in a private world. Evaluation formula will be based on world and state.

In following present a definition of Kripke structure [3]: A Kripke structure is defined to be a tuplu \( M=(W,S_w, \mathcal{R}_w, P, \mathcal{B}, \mathcal{D}, \mathcal{I}) \) where \( W \) is a set of possible worlds, \( S_w \) is the set of states in each world \( W, \mathcal{R}_w \subseteq S_w \times S_w \) is a total binary relation, \( P \) is a truth assignment to the primitive proposition of \( \Phi \) for each world \( w \in W \) at each state \( s \in S_w \) and \( \mathcal{B}, \mathcal{D}, \mathcal{I} \subseteq W \times S \times W \) are relation on the worlds, W and states S.

The satisfaction relation of formulas, denoted with \( \models \) is interpreted as:

s1. \( (M, w_s) \models f \) iff \( f \in \mathcal{P}(w, s) \) where \( f \) is a primitive proposition

s2. \( (M, w_s) \models \neg f \) iff \( (M, w_s) \not\models f \)

s3. \( (M, w_s) \models f \land g \) iff \( (M, w_s) \models f \) and \( (M, w_s) \models g \)

s4. \( (M, w_s) \models Ef \) iff there exists a fullpath \( (w_{do},w_{si},...) \) such that \( (M, (w_{do},w_{si},...)) \not\models f \)

s5. \( (M, w_{do}) \models A(f) \) iff for all fullpath \( (w_{do},w_{si},...) \) such that \( (M, (w_{do},w_{si},...)) \not\models f \)

s6. \( (M, w_s) \models \text{BEL}(f) \) iff for any \( w' \), satisfying such that \( (w,s, w') \in \mathcal{B} \) then \( (M, w') \models f \)

s7. \( (M, w_s) \models \text{DES}(f) \) iff for any \( w' \), satisfying such that \( (w,s, w') \in \mathcal{D} \) then \( (M, w') \models f \)

s8. \( (M, w_s) \models \text{INTEND}(f) \) iff for any \( w' \), satisfying such that \( (w,s, w') \in \mathcal{I} \) then \( (M, w') \models f \)

p1. \( (M, (w_{do},w_{si},...)) \not\models f \) iff \( (M, (w_{si},...)) \not\models f \)

p2. \( (M, (w_{do},w_{si},...)) \not\models f \) iff exist a \( k \geq 0 \) such that \( (M, (w_{do},w_{si},...)) \not\models g \) and any \( j \in [0,k] \), \( (M, (w_{sj},...)) \not\models f \) or for all \( j \geq 0 \), \( (M, (w_{sj},...)) \not\models f \)

An agent believes \( f \), denoted by \( \text{BEL}(f) \), in \( s \) state if and only if \( f \) is true in a world accessible through the relation \( \mathcal{B} \) with time \( t \). \( \mathcal{B} \) relation is independent by state. Similar association can do with desires and intentions. A formula \( f \) is valid in \( M \), denoted \( M \models f \) if \( (M, w_s) \models f \) for any world \( w \in W \) and any \( s \in S_w \). Difference between CTL logic and BDI logic is the occurrence constraints on beliefs, desires and intentions.

5 Study case for determination the broker behavior using a BDI agent
For start we present some elements that we need to build the CTL model beginning to purchase an operation. The operation is a financial instrument which is emitted usually for a long term and produces some finances which can receive or can convert periodic. It’s important to known as the operation is not reimbursed. The operation confers it holder permission of he explained to known as the operation is not reimbursed. The operation is a financial instrument which is emitted for a long term and produces some finances which can receive or convert periodic.
bought an operation or invested in to operation or became the shareholder means paid this price at a given moment for can obtained then a flux of future finances. The fluxes on which waits them shareholder consisted of some parts from his interest or from profits, as well as from his process or for sale a value at a given moment. The symbols on which use them in credibility operations financier [8] formula is given by: $C_0$ is the operations value to zero moment of this purchase immediately after dividends pay or exactly to the moment it purchase after emission. Its name is price of getting or price of purchase; $D_0$ is the dividend pay exactly before evaluation process or evaluation $C_0$ price; $nr$ is the number of year’s placement or investment into operations where $nr=1,2,\ldots,k,\ldots$; $per$ is the annual percent of an operation process, that is $per=100*uai$, where $uai$ is the unitary annual interest. The paradigm on the base of fundamental principle of the financial reciprocal commitments equivalence (expenditures and finance of issuer or of shareholder) accordingly the investment into operations needs to be in progress the relation: $C_0=\Sigma_{k=1, nr}(D_k/(uai+1)^k) + C_{nr}/(uai+1)^{nr}$ where $\Sigma_{k=1, nr}$ means sum from $k=1$ to $nr$ for expression $(D_k/(uai+1)^k) + C_{nr}/(uai+1)^{nr})$. This relation is referred to the equality among the of an operation process to purchase and its future values, dividends and its value of resale through the update percent or of operations profitability. The price evaluation of resale is can do if is known the purchase price and the dividend received. The relation for this case becomes: $C_{nr} = C_0 + (uai+1)^{nr} - \Sigma_{k=1, nr}(D_k/(uai+1)^k)$. In this formula appears the globally fructify factor $(uai+1)^{nr}$. Can be remarked the fact as the resale price is function of getting prices, of dividends, of the market interest and the lending course. In the construction of our model shall attend the two cases. In first case we interest the dividend. In this case we verifies to don’t diminishes the operation such in kind that when selling to don't miss from the operation value. In second case we don't interest the dividend and is due to verifies specifically the way to offer and demand from on sale. The model is builder in order to help to the decision bought or sold operations. The discussion is put from viewpoint of which person which makes speculations, broker, of the operation price. The minimum threshold which we consider is the purchase price $C_0$. The broker behavior is gives by too gain only that exist some amount of money denoted by $\alpha$ on which considers satisfactory, $C_0+\alpha$, for to determinate to continue this activity. That is, $C_0 < C_0+\alpha \leq C_1,\ldots, C_{nr}$. Operation broker is given by market conditions and demand, offer, price and number of operations it holds. The operation on market is determined by three decisions: buy, sell or wait. The time steps are:

1) he buys the operations that means he begins with $C_0$
2) he is needed to decide if keep, sell or buy the operations depending on the market and $\alpha$ is obtained:
   2') keep operations if $\alpha$ is not obtain;
   2'') if $\alpha$ is obtained then is sure to sell operations, high price and operations number is low (means high offer and low demand);
   2''') buy other operations if low price and operations number is high (means low offer and high demand) and going to (t1)
3) if is not obtain $C_0$ then out of market.

As example we present in Fig. 1 the representation of model for CTL structure. The state-transition diagram showed in Fig. 1 has four locked-state events.

![Fig. 1 Model Representation for CTL structure](image-url)

These locked-state events occur because the offer and the demand, in most instances, take one action and then await a response before moving for a new state. In fact, only two event flows Sell or Buy. CTL structure for the determination the broker behavior for capital market control is presented in the Fig. 1 and states of the system are denote with $s_0, s_1, s_2, s_3$. The Kripke model has three states and the propositional variables are from the set \{Sell, Buy, DecVal\}. Sell means when the broker decides sell the operations, Buy means when the broker decides buy other operations and DecVal means operations value decrease. The formal definition of the Kripke structure in CTL model for broker behaviour for capital market control is given by: $M = (S, Rel, P)$, where $S=\{s_0, s_1, s_2\}$, $Rel=\{(s_0,s_1), (s_1,s_0), (s_1,s_2), (s_2,s_0), (s_3,s_1)\}$, $AP=\{Buy,Sell,DecVal\}$, $P$ assigns state $s_0$ in M with not Sell, not Buy and not DecVal that is set \{¬Buy, ¬Sell, ¬DecVal\}, means in this case that the broker waits for buy the operations. Using the CTL model and beginning from the example presented in the previous section where we considered that an agent wants eventually to not decrease the value of an operation, intending to buy sometime a new operation or waiting until operation which holds don’t decrease in the future, and don’t believe that in all this time the operation don’t decrease the value. The mental state of this agent can be represented by the following formula: DES(AF (¬decrease-the-value-of-an-operation)) $\land$ INTEND(EF (buy-an-operations)$\lor$(A(¬sell-operation $U \rightarrow$ decrease-
the-value-of-an-operation))) ∧ (¬BEL(AF(¬decrease-the-value-of-an-operation))))). The state of a BDI agent at any given point of time is a triple (B, D, I). The process of practical reasoning in a BDI agent may be summarized as: a set of beliefs B, representing the information the agent has about its environment; an option generation function, which determines desires, D, on the basis of the current beliefs and its current intentions; a set of current intentions I, representing those state of affairs that it has committed to trying to bring about. The BDI_CTL model has a two-dimensional structure. A dimension is a set of possible worlds corresponding to different views of the agent representing his mental states. The other is a set of temporal states which describe the temporal evolution of the agent. A pair form of world states and temporal states is called situation. In formula presented before we have three sets representing: a fundamental desire is \( \varphi = AF(¬DecVal) \) which means the operation don’t decrease the value. In our model the set of states which satisfying formula \( \varphi \) is \( \{s_0,s_1,s_2\} \); Intentions is \( \varphi_I = (EF Buy)\lor(A(¬Sell U ¬DecVal)) \) which means intend in future buy a new operations or waiting while the operation don’t decrease its value. In our model the set of states which satisfying formula \( \varphi_1 \) is \( \{s_0,s_1,s_2\} \); Negation of beliefs is \( \varphi_2 = AF(¬DecVal) \) which means don’t believes in future the operation don’t decrease its value. In our model the set of states which satisfying formula \( \varphi_2 \) is \( \{s_0,s_1,s_2\} \); The basic BDI consistency rules [3] say: if \( BEL(f) \in P(w,s) \) and \( (w,s,w')\in B \) then \( f \in P(w',s) \); and if \( ¬BEL(f) \in P(w,s) \) then exist \( w' \) such that \( (w,s,w')\in B \) then \( ¬f \in P(w',s) \); The algorithm for constructing a pseudo BDI_CTL tableau [3] consists of five different procedure. The first expands a set of formulas to propositional CTL tableau. The second expands a propositional CTL tableau into a fully expanded propositional CTL tableau. The next two expand the elementary formulas of CTL and elementary formulas of BDI, and last checks for satisfaction of labels. Depending on the label being a fully expanded CTL tableau or not, different satisfaction conditions apply for the label. When the root nod of the pseudo BDI_CTL tableau is marked satisfied we can say the label of such a node is BDI_CTL satisfied. Returning to our example above the node \( s_2 \) will don’t be labeled as satisfied because this node contain the atomic proposition DecVal, means decrease-the-value-of-an-operation. This situation does not agree that we check the action of the agent who choose the path in which it just buy a new operations or wait a time until operations value not decrease.

6 Conclusion

The behavior of the model checker algorithm presented above consists of identifying the sets of states of a model M which satisfy each sub formula of a given CTL formula f and constructing the set of states, from these sets, that satisfy the formula f. Evaluation construct in certain condition of a general model of operation placement he permits of a broker the possibility established all the possible situations for a tackled decision for capital market. We presented in this paper the mental BDI agents whose behavior is conduct by beliefs, desires, intentions, and a decision. The case study addresses the scenario of a broker that reasons about the task avoiding behavior which doesn’t agree. In these conditions is making it clear that a real configuration with possible different situations could be transpose to a mathematic model.

References: