MODELING AWARENESS OF AGENTS USING POLICIES

Amir Talaei-Khoei, Pradeep Ray, Nandan Parameswaran
Asia-Pacific Ubiquitous Healthcare Research Centre, University of New South Wales, Sydney, Australia

Ghassan Beydoun
School of Information Systems and Technology, University of Wollongong, Wollongong, Australia

Keywords: Awareness, Policy, Intelligent Agent, Disasters.

Abstract: In addition to cooperation, research in disaster management exposes the need for policy awareness to recognize relevant information in enhancing cooperation. Intelligent software agents have previously been employed for problem solving in disaster situations but without incorporating how the agents can create or model awareness. This paper presents an awareness based modelling method, called MAAP, to maintain awareness of software agents of a given set of policies. The paper presents preliminary results indicating that the use of policies as a source of awareness, as facilitated by MAAP, is a potentially effective method to enhance cooperation.

1 INTRODUCTION AND RELATED WORK

Disaster management may involve unknown information which may result in inadequate cooperation between agents involved. Given that many integrated standards in distributed and autonomous networks e.g. (Zimmerman 1980; Udupa 1999; Beydoun et al 2009a; Tran et al 2006; Beydoun et al 2009b) assume cooperation is already sufficient, we really need to be able to measure cooperation (Ray et al 2005). Towards this, we apply the concept of *awareness* from Computer Supported Cooperative Work (CSCW) (Daneshgar et al 2000; Sadrei et al 2007) and propose using policies as a way to recognize awareness. The classical approach in modelling semantics of agents mental attitudes (agents knowledge and states) is *possible-worlds model* e.g. (Rao et al 1991). This model provides an intuitive semantics for mental attitudes but it also commits us to *logical omniscience* and *perfect reasoning*. The assumptions (Sillari 2008) here are (1) the agent is omniscient e.g. it knows all the valid formulas. For example, while there was damage in TPS on left wing of the shuttle, NASA did not know that, because they did not take the left wing as relevant and they took the foam strike on the wing as a "turnaround" maintenance issue. (2) the agent is a perfect reasoner i.e. it knows all the consequences of its knowledge. This clearly an idealization, people just know the relevant truth and the consequences. For example, NASA policy guidelines provided for operating spacewalk rescue procedures, but NASA management team did not take it relevant. Again, they assumed that the outcome of their *reasoning is perfect*. Four different categories of approaches to address the problem of logical omniscience and perfect reasoning are (Halpern et al 2010): Algorithmic Approach, Syntactic Approach, Impossible-worlds Approach, and Awareness Approach. In our problem, there is a pragmatic interpretation for awareness, which motivates us to use awareness approach and in particular Logic of General Awareness (LGA) (Fagin 1988) underpinned by the idea relevance of knowledge. Under the possible-worlds interpretation, a valid sentence and its consequences are true in every world that the agent considers possible. The known sentence and its known consequences may or may not be relevant. LGA defines awareness of a formula as relevance of that formula to the situation. This definition is particularly applicable for cooperation enhancement process. Definition awareness, they differ explicit and implicit knowledge in a way that an agent explicitly knows a formula when it implicitly knows that and also it is aware of that.

Talaei-Khoei A., Ray P., Parameswaran N. and Beydoun G..
MODELING AWARENESS OF AGENTS USING POLICIES.
DOI: 10.5220/0003436403530358
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 353-358
ISBN: 978-989-8425-77-5

Much research proposes that policies can be used to implement the awareness in design phase of developing distributed cooperative applications. The use of given policies as guidelines to which information should be aware has not been addressed. Directory Enabled Networks (DEN) is the main policy structure used today (Sloman 1994). We borrow our policy structure from DEN-ng. In DEN-ng the main idea is that the given set of policy rules should be loaded based on current knowledge and an event. In this model, a policy consists of different policy rules while each rule defines the "event-condition-action" semantics in DEN-ng. Theses semantics are such that the rule is evaluated when an event occurs. When the condition clause satisfies then the action clause will be executed. Given modality for performing the action, Sloman (Sloman 1994) elaborates two types of policy rules: (1) authentication policy rules that permit (positive authentication) or forbid (negative authentication) to perform the defined action. (2) obligation policy rules that require (positive obligation) or deter (negative obligation) to perform the defined action.

2 AWARNESS ENHANCED AGENT MODEL

Talaei-Khoei et al (Talaei-Khoei 2011) employ intelligent agents for human roles to assist them being aware of relevant information in the situation. They propose use of policies as an alternative way to compute required awareness of these agents in disasters. They do not address how actually agents can create their awareness based on given policies, which involves technical implementation aspects. Proposing a four-step process for cooperation enhancement, Ray et al (Ray et al 2005) annotate awareness as an understanding of *relevant information* that is required for an individual to cooperate. *Modelling Awareness of Agents using Policies (MAAP)* in this paper proposes a modelling method based on Logic of General Awareness (Fagin 1988) to use policies as an alternative source of awareness. MAAP is intended to be an extension to LGA to use policy rules, DEN-ng, as an alternative source to create awareness.

In the *possible-worlds* conceptualisation of agent mental states, an agent builds different models of the world using some suitable language. To enact agent awareness, a metamodel of possible-worlds based on LGA describes the awareness of an agent. It consists of a non-empty set of variables in each world, where a variable can be differently instantiated using a

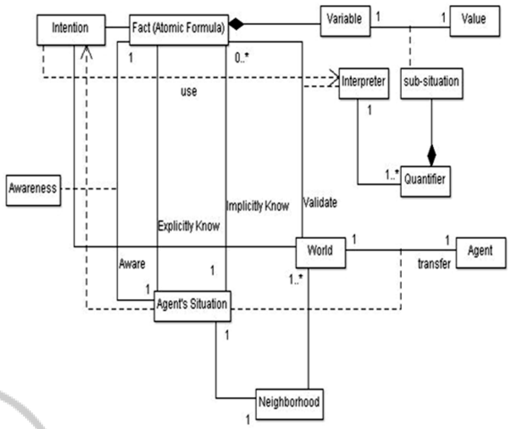

Figure 1: Possible-worlds.

domain function describing sub-situations from the possible worlds. The truth for a fact describing a given situation, presented by an atomic formula, is computed by a knowledge interpreter. Each formula has a list of variables. In each world and for each formula, the interpreter defines a set of tuples of values for variables in a formula. When, in a certain world, the atomic formula is computed as true or false in that world (see (Kinny et al 1996)). We use NetLogo agent implementation platform, which provides a knowledge interpreter according to the Logic of General Awareness. This provides a table that stores all the formulas that are true in a world for a certain list of variables i.e. quantifiers. The table has key with world and the name of the formula and it stores the list of quantifiers. Each world can also validate a formula or a fact. A formula is valid in a given world, if the quantifier list can be found in the interpreter.

Worlds in our model are connected by actions. As advised in (Rao et al 1995), a world has a single past and multiple tree-like futures, called branching-time model. If a world describes the state of affairs in the next time instant, then, we assume that, there exists an action that transfers the state of the system from the current world to the next one. We define a *path* to be a sequence of worlds. As such, the set of all paths is a reflexive transitive closure of the set of actions. We add the unary modal operators "*next*" and "*eventually*" where "*next*" of a fact is true if the fact is true at the next time instant, and "*eventually*" of the fact is true if the fact eventually becomes true. Since we do not know the future in advance, there can be more than one path for a an eventual fact. To support this, we define two more operators to represent the modality of a formula describing a statement about the world: *inevitable* and *optional*. An *inevitable* formula at a particular point in a time-tree is true if the formula is true of all paths starting from that point. An *optional* formula at a particular point in a time-tree is true if it is true of at least in one path starting from that point. We define *Done Action* to be an action that was just performed to transfer the agent from a past world to the current world. We also add a method, *Done,* that returns true if the agent has just performed that action.

Events can affect the behavior of agents. We define event as an entity that is sensed by an agent. When an agent receives an event, it logs that event in the Received Events list of the world. We implement Received Events as a table that has keys with agent and world. The value of each key shows the name of the event. Therefore, each item in this table presents events that the agent at a world has received. In order to implement the *Received* operator, that is `RE`, it returns true for an event e if there existed any world in the past that the received event table has the event e.

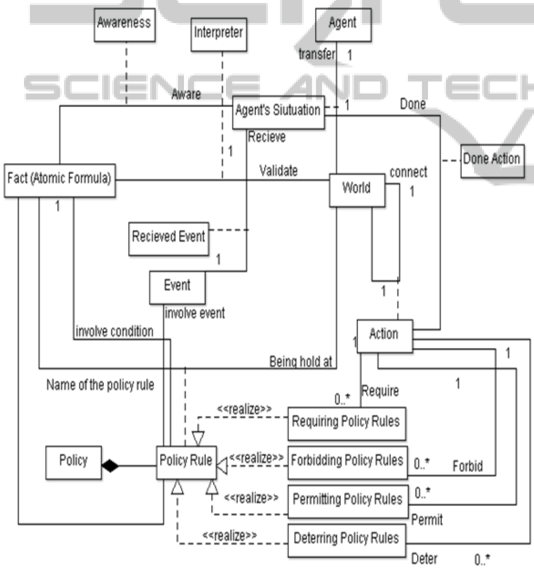

Figure 2: Branching-Time Model.

Policies are viewed as constraints between worlds. Consequently, policy rules may also be viewed as *rule-type* facts that can be true in a world. Since only *Forbidding* and *Requiring* policy rules are in force, *permitting* and *deterring* do not change the behaviour of an agent directly. Following the structure of policy rules in DEN-ng, each policy rule has an event, a condition and the modalities of an action. The implementation of policy rules is based on a table structure. Each item of the table shows a policy rule which is associated with a world and a formula describing a world state. The item besides its key consists of a condition, an event, an action and the modality of the action (i.e. "forbidding", "requiring", "deterring" and "permitting").

3 ENACTING POLICY-BASED AWARENESS OF AGENTS

A *policy-aware* agent is capable of executing the following three steps: (1) recognize relevant policy rules; (2) recognize information required to follow the relevant policy rules; (3) enact the policy through changing the behaviour of an agent based on the recognized relevant information. The first two steps implement the association of *awareness* to an agent's situation and relevant knowledge. Step 3 implements the effect of awareness on its behaviour.

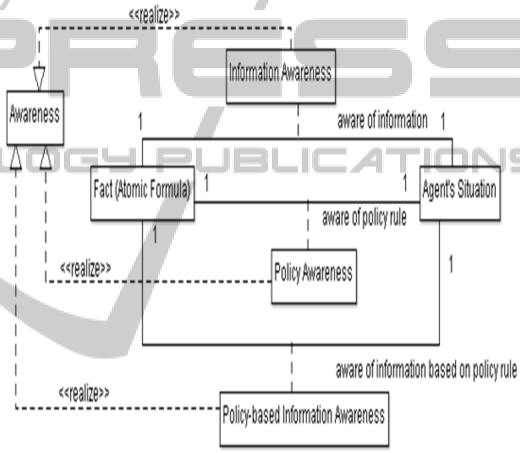

Figure 3: Awareness Association Components in MAAP.

Policy rules define an association between formulas and worlds. Agents may implicitly know a formula of a policy rule; however they may not consider the policies as relevant to the current situation in their worlds. (In the following, occasionally when no ambiguity arises, we drop the term "formula" in the phrase "formula of a policy rule" and interchangeably call it "a policy rule" or even "a policy"). A *policy-aware agent* may create its awareness to a policy rule formula when there is a possibility to break policy rules. Essentially, policy rules can function as constraints on what the agent knows and what the agent considers as relevant information. When the agent is not going to break a policy rule, there is no point in becoming aware of the policy, and agent can simply follow its normal behaviors. Based on the logic of general awareness, an agent implicitly knows the consequences of its implicit knowledge. Therefore, if an agent implicitly knows a policy rule formula and also implicitly knows that the policy conditions are satisfied while an event is already received, it creates its implicit knowledge of associated consequences of a policy rule. Depending on whether the policy rule is a *forbidding* or *requiring* rule, the agent will implicitly know that in the next time instance (that is, the next possible world) the action has been or has not been done, respectively. Taking this point into account, if the agent optionally now or in future satisfies the following three conditions, then it is said to be aware of the policy rule: (1) The agent has an implicit knowledge of the policy rule formula. (2) The agent implicitly knows that the condition of the policy rule is satisfied and it has received the event associated with the policy rule. (3) In the next time instant, the agent has done the forbidden action it has not done the required action. In MAAP, the *awareness* association between agent's situation and a fact has three different components, which realize awareness as shown in Figure 3. Based on Logic of General Awareness, being aware of a fact implies that the agent is aware of a sub-fact. Therefore, by being aware of a policy rule formula, the agent is also aware of the condition of the policy rule as well as the fact that the involved event in the policy rule is received. In fact, regardless of the truth or falsehood of these two pieces of information, they are relevant.

Being aware of a policy rule and accordingly being aware of the required fact to follow the policy rule, the agent might go for finding these information if it is possible. When an agent receives an event pertaining to a policy rule, for each world in all accessible paths, and until it is not aware of the policy rule, the agent is aware of two things: (1) conditions occurring in the policy rule, and (2) the fact that in that world, the involved event has occurred. How the agent will change its behavior after it finds relevant information is modelled.

Updates in awareness knowledge lead to changes in agent's behaviors. This may happen in two ways: (1) Awareness Deliberation; or (2) Following Policy Rules. To have *Awareness Deliberation*, following Rao and Georgeff's definitions (Rao et al 1995), we add blind and single-minded agents to our definitions. We define a blind agent to be the one who maintains its awareness about the optional truth of a formula until it implicitly knows the formula or its negation. We define a single-minded agent to be the one who maintains its awareness about the optional truth of a formula until one of following happens: (1) it implicitly knows the formula; (2) it implicitly knows the negation of the formula; (3) it does not know that the formula can be optionally true now or in future; and (4) it does not know that the formula can be optionally false now or in future. The possibility here of achieving truth or falsehood of the formula has been added to capture the ability of the agent to find out the information that it is aware of. The basic idea, here, is such that being aware of the information, a blind agent selects the paths that lead the agent to implicitly know the information or its negation. The single-minded agent checks also the possibility of acquiring such information. As there are often more than one path, we recommend the shortest-path strategy.

Figure 3 shows the model entities (1) agent's *Awareness Deliberation* and (2) *Following Policy Rules* has the relationship of perform, which means "*Does* an action". Performing an action makes the agent *transfer* from one world to another one. When the action is done and the agent is transferred to the new world, the agent will add the action to the Done Action set and the procedure *Done* will return true. An exception-list is used to include all the actions that the agent cannot perform as determined by the policy rules (as forbidden actions). Then the agent according to the short-path strategy finds an action (in the shortest path) that is not forbidden and thus ends up in the shortest path to implicitly know its awareness. Exceptions are computed by using a procedure which forces an agent to do a certain required action even if it is not encouraged by its awareness. The procedure `Does` returns the action to do and the procedure `transfer` transfers the agent from the current world to the next one. The agent will perform the actions as guided by its awareness, if there is currently no required action. The agent will not perform any forbidden actions, even if it is encouraged by its short-path awareness deliberation strategy.

4 VALIDATION

To illustrate our MAAP framework for policy awareness, we consider the following scenario: Approximately 82 seconds after the launch of The Space Shuttle Colombia Jan 16, 2003, a piece of thermal insulation foam broke off the external tank striking the Reinforced Carbon-Carbon panels of the left wing. Five days into the mission, the engineering team asked for high-resolution imaging (Wilson 2003). While the Department of Defense (DOD) had the capability for imaging of sufficient resolution to provide meaningful examination, NASA declared the debris strike as a "turnaround" issue. Therefore, it failed to recognize the relevance of possible damage in TSSC to the situation and did not ask DOD for any imaging. During re-entry to the earth atmosphere over Texas, on Feb 1, 2003, the shuttle disintegrated claiming the lives of all seven of its crew. If NASA had recognized the relevance of information about the TSSC and had requested imaging from DOD, there would have been a rescue procedure available by spacewalk for repairmen (Wilson 2003). At the time of the accident, there were the policy guidelines in NASA stating that when a possible strike is reported, if there is any TSSC damage, the spacewalk repairmen procedure must be operated. The protocol had been also established between NASA and DOD for high resolution imaging. Therefore, although the capability and the guidelines were available, NASA could not recognize the relevance of information, which led to deny image request and accordingly death of seven people as well as loss of Space Shuttle Colombia. NASA management, bombarded with irrelevant and loosely relevant information, could not recognize which policy should be applied in the situation and which information is required to be gathered. They could not realize the high possibility of debris strike. as relevant information for cooperation, NASA management should have been aware of the accrued information of TPS but they were not.

One of the policy rules in this case says that when a possible strike is reported, if there is any TPS damage, the spacewalk rescue procedure must be operated. Although NASA management team did not know the TPS damage, there existed a possibility implying that TPS might have been damaged and it was possible for NASA to recognize that the policy rule was going to be broken. In fact, if NASA had considered TPS damage as relevant information, it would have asked DOD for high resolution images to find out the possible damage. Then, recognizing the damage in TSP, NASA would have operated space walk rescue procedure. However, when the strike was reported as an event, the TPS damage as the condition that the policy rule was a relevant information, was overlooked by NASA.

NASA was not aware of TPS damage and therefore, they decided to simply classify the damage as turnaround effect rather than asking DOD for imaging and investigating if it is really turnaround effect i.e. shortest path. Applying MAAP in this situation, we see that NASA becomes aware of the TPS damage; although it does not implicitly know there is any damage or not, it just recognizes that TPS damage is a useful information. Therefore, NASA recognizes that TPS damage is useful as well as turnaround effect. Thus, NASA would choose to ask DOD for imaging because there is an option in future, which satisfies the implicitl knowledge about truth or falsehood of TPS damage and the turnaround effect.

We applied the MAAP strategy of awareness to the Space Shuttle Columbia disaster case, and implemented it using the NetLogo MAAP library. We designed four different policies and out of these policies, we made eighteen policy rules. We also designed ten different scenarios, that were similar to the real incident reported in (Wilson 2003). Some of these scenarios required a policy rule to react correctly and some of them did not. Our simulation involved eighteen steps where in step 1 it chose only one policy rule, in step 2, it chose two and so on. The program repeated each step one hundred times while each time it selected random policy rules and chose a random scenario out of the ten designed scenarios. The simulator ran each selected scenario with policy rules following MAAP and without policy rules following the standard Logic of General Awareness. The program records the total number of the failures of each of the steps with and without using MAAP. (Failure was defined as not doing a certain action and not achieving a certain situation given to the simulator for each scenario.) Taking this simulation into account, we found that the reason why the improvement had a “kink” at two points was that at these two steps, the policy rules did not match with the scenarios i.e. *received events* and *done actions.* This actually happened because of the randomized procedure taken to generate the input data. In other words, the policies taken were not related to the chosen scenario in the “kink” points. However, as the number of policy rules increased, not in all scenarios the chosen policy rules were found to be useful. In fact, although the overall improvement remains positive, in order to have better performance, the policy rules should be appropriate for a chosen scenario. The overall outcome for this evaluation is that MAAP by increasing the number of policy rules becomes a more effective methods. This is actually supported by what is proposed as a fundamental in MAAP and suggested in awareness model of DEN-ng (Strassner et al 2009).

5 CONCLUSIONS

Research in CSCW and intelligent agents demonstrated the need for a definitive method to compute awareness. This paper introduces MAAP as a modelling method based on LGA and proposes the use of policy rules as an alternative source of awareness. This can avoid bombarding an individual (agent) with irrelevant or loosely relevant information. Our approach has a three limitations: First, the design of LGA and accordingly MAAP are based on intersecting implicit knowledge and awareness to get explicit knowledge. Intersecting awareness and implicit knowledge may lose some of the relevant information. As we propose use of policies for computing awareness, this may lead to violating policies. In such situations, the agent in fact is not capable of following the policy rules. Therefore, the assumption in MAAP is that design of policies is based on the agents' capabilities, which is somewhat too ideal. A method to recognize disability of agents to follow a policy rule must be designed to enhance MAAP for future work. Second, policy rules may interact with each other and a newly added policy rule may conflict with the existing ones. Third, refining high-level policies to computational policy rules is a challenging task by itself, which consists of: (1) Determining the resources that are needed to satisfy the requirements of a policy during unexpected situations, such as disasters, (2) Transforming high-level policies into role-level DEN-ng policy rules, (3) Verifying that the lower level policy rules actually meet the requirements specified by the high-level policies. That opens a new direction for research to enhance MAAP policy refinement methods.

Finally, MAAP is specified only for DEN-ng policy rules. The reason, as it has been described, is that the awareness model of DEN-ng policy rules is strongly well cited and well equipped by supportive tools. This can be also useful to generalize the idea of MAAP. In fact, we can say that the agent will be aware of each conditional proposition, while there is a possibility now or in future to violate the proposition. As such, the agent needs to become aware of the propositions ad its associated conditions.

REFERENCES

Beydoun, G., G. Low, H. Mouratidis and B. Henderson-Sellers. "A security-aware metamodel for multi-agent systems (MAS)." Information and Software Technology 51(5): 832-845, 2009.

Beydoun, G., Low, G., Henderson-Sellers, B., Mouraditis, H., Sanz, J. J. G., Pavon, J., Gonzales-Perez, C. "FAML: A Generic Metamodel for MAS Development." *IEEE Transactions on Software Engineering* 35 (6), 841-863, 2009.

Daneshgar, F., and Ray, P.,"Awareness Modeling and Its Application in Cooperative Network Management," 2000, p. 357.

Fagin, R., and Halpern, J. "Belief, awareness, and limited reasoning," *Artificial Intelligence*, vol. 34, pp. 76, 39, 1988.

Halpern, J., and Pucella, R. "Dealing with logical omniscience: Expressiveness and pragmatics☆," *Artificial Intelligence*, 2010.

Kinny, D., and Georgeff, M. "Modelling and Design of Multi-Agent Systems," 1996.

Rao, A., and Georgeff, M. "BDI Agents: from Theory to Practice," 1995.

Rao, A. and Georgeff, M., "Modeling Rational Agents within a BDI-Architecture," 1991.

Ray, P., Shahrestani, S., and Daneshgar, F., "The Role of Fuzzy Awareness Modelling in Cooperative Management," *Information Systems Frontiers*, vol. 7, no. 3, pp. 299-316, 2005.

Sloman, M. "Policy driven management for distributed systems," *Journal of Network and Systems Management*, vol. 2, no. 4, pp. 333-360, Dec. 1994.

Sillari, G., "Quantified Logic of Awareness and Impossible Possible Worlds," *The Review of Symbolic Logic*, vol. 1, no. 04, pp. 514-529, 2008.

Strassner, G. et al., "The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking," *Journal of Network and Systems Management*, vol. 17, no. 1, pp. 5-32, 2009.

Sadrei, E., Aurum, A., Beydoun, G., and Paech, B. "A Field Study of the Requirements Engineering Practice in Australian Software Industry", *International Journal Requirements Engineering Journal* **12** (2007), pp. 145–162.

A. Talaei-Khoei, P. Ray, and N. Parameswaran, "Policy-based Awareness: Implications in Rehabilitation Video Games," presented at the In the Proceedings of Wawaii International Conference on System Sciences, Hawaii, US, 2011.

Tran, QNN, Low, GC and Beydoun, G., "A Methodological Framework for Ontology Centric Agent Oriented Software Engineering", International Journal of Computer Systems Science and Engineering, 21, 117-132, 2006.

Wilson, J., *Report of Columbia Accident Investigation Board, Volume I*. USA: NASA, 2003.

Udupa, D., *TMN: telecommunications management network*. McGraw-Hill, Inc. New York, NY, USA, 1999.

Zimmermann, H., "OSI reference model–The ISO model of architecture for open systems interconnection," *IEEE Transactions on communications*, vol. 28, no. 4, pp. 425–432, 1980.