A META-LEVEL ARCHITECTURE FOR STRATEGIC REASONING IN NAVAL PLANNING (EXTENDED ABSTRACT)*

Mark Hoogendoorn [a] Catholijn M. Jonker [b] Peter-Paul van Maanen [a,c]
Jan Treur [a]

[a] *Department of Artificial Intelligence, Vrije Universiteit Amsterdam*
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {mhoogen, pp, treur}@few.vu.nl
[b] *NICI, Radboud University Nijmegen*
Montessorilaan 3, 6525HR Nijmegen, The Netherlands
Email: C.Jonker@nici.ru.nl
[c] *Department of Human in Command, TNO Defense, Security and Safety*
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands

1 Introduction

The management of naval organizations aims at the maximization of mission success by means of monitoring, planning, and strategic reasoning. In resource-bounded (naval) situations strategic reasoning helps in determining if a go or no go should be given, or if attention should be shifted, to a certain evaluation of possible plans after an incident. An incident is an unexpected event which results in an unmeant chain of events, if left alone. In a planning context strategic reasoning can occur both in plan generation strategies (cf. [4]) and plan selection strategies.

The above context gives rise to two important questions. Firstly, what possible plans are first to be considered? And secondly, what criteria are important for selecting a certain plan for execution? In resource-bounded situations first generated plans should have a high probability to result in a mission success, and the criteria to determine this should be as sound as possible. In this paper a generic meta-level architecture (cf. [1, 2, 3]) is presented for planning, extended with a strategic reasoning level. Expert knowledge is used to formally specify executable properties for each of the components of the architecture. These properties are used for simulation and facilitate formal verification of the simulation results.

2 A Meta-level Architecture for Naval Planning

In Figure 1 the proposed generic architecture is shown for strategic planning applicable to naval command & control organizations. The components denote processes, solid lines denote information, and the dotted lines denote a separation between (meta-)levels. First of all, plans are executed within the deliberation cycle which is the part between the dotted lines. By comparing the perceived situation with the planned situation the `Monitoring` component generates evaluation information. In case the evaluation involves an exception `PlanGeneration` determines what the possible plans are, considering the situation. The conditional rules for the possible plans given a certain event are passed by the `StrategyDetermination` component. The possible plans are forwarded to the `PlanSelection` component which evaluates the plans taking the current beliefs on the world into consideration. In case an empty list of possible plans is received `PlanSelection` informs `StrategyDetermination` that no plan could be selected. The same is done when none of the plans passed the evaluation. The `StrategyDetermination` component can, in case of such a

*The full version of this paper appeared in: B. Henderson-Sellers and M. Winikoff (eds.) *Proceedings of the Seventh International Workshop on Agent-Oriented Information Systems (AOIS'05)*, 2005, pp. 18-25, Utrecht, The Netherlands.

notification, provide `PlanGeneration` with additional conditional rules. It can also generate a new way to evaluate plans by means of different criteria rules. If a suitable plan has been found, it is passed to `PlanExecution` and becomes the current plan. The execution of the plan is done by means of the receipt of beliefs on the world and applying the plan to derive the next action. The actions that have been determined are passed to the `World`.

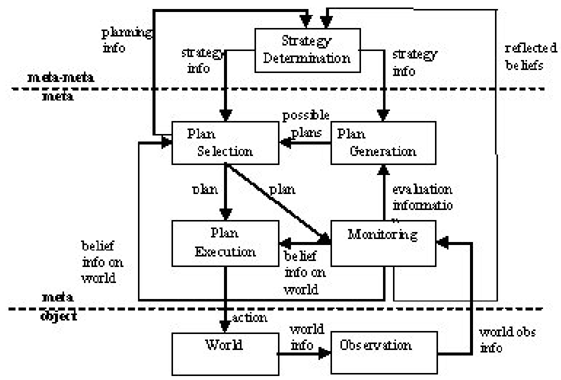

Figure 1: Strategic planning processes applicable in naval organizations.

3 Case-study in Naval Domain

The above architecture has been applied to a case-study in the naval domain. Executable properties for each of the components have been specified for this particular domain and include `PlanGeneration` and `PlanSelection` strategies based on the candidate plans and criteria passed from `StrategyDetermination`, respectively. Selection criteria strategies incorporate mission success, safety and fleet morale, over which a weighed sum is calculated, resulting in a plan evaluation formula of the following form:

$$evaluation\_value(P : PLAN) = \alpha \cdot mission\_success\_value(P : PLAN) + \beta \cdot safety\_value(P : PLAN) + \gamma \cdot fleet\_morale\_value(P : PLAN)$$

where all values and degrees are in the interval $[0, 1]$ and the weights are normalized. The degrees depend on the type of mission and the current state of the process. Furthermore, candidate generation strategy determination is based on information from `PlanSelection` and `PlanGeneration`. Three different modes of operation are defined, which are limited action demand, full preferred plan library, and exceptional action demand. Finally, the `StrategyDetermination` component also includes executable properties that establish a change in the weights of the different selection criteria in case of failure to select an appropriate plan. The above mentioned properties were used in a number of simulation runs. The results were formally verified by means of the use of a developed software tool called TTL checker. These properties include upward and downward reflection [3], verifying whether no unnecessary extreme measures are taken, plans are not changed without a proper cause, and were all satisfied for the given trace. More simulation runs and the validation of properties for the simulation traces are expected to give more insight for future complex resource-bounded naval planning support systems.

References

[1] R. Davis. Metarules: reasoning about control. *Artificial Intelligence*, 15:179–222, 1980.

[2] W. van der Hoek, J.-J.Ch. Meyer, and J. Treur. Formal semantics of meta-level architectures: Temporal epistemic reflection. *International Journal of Intelligent Systems*, 18:1293–1318, 2003.

[3] R. W. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *Artificial Intelligence*, 13:133–170, 1980.

[4] D. E. Wilkins. Domain-independent planning representation and plan generation. *Artificial Intelligence*, 22:269–301, 1984.