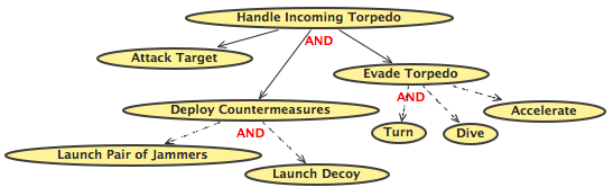

Figure 1: Goal Structures in TDF

from left to right. The submarine will first fire back at the attacker by trying to achieve the goal **Attack Target**. It will then **Deploy Countermeasures** and **Evade Torpedo**. The goals at the next level are unordered (signified by dotted lines), so for example, to **Evade Torpedo** the submarine can **Turn**, **Dive** and **Accelerate** in any order.

2.3 Tactics Design Patterns

TDF tactics design patterns are an idiomatic, BDI-influenced method of representing the intent of a tactic. They outline a tactic's main objective, triggering condition, problem description, solution description, invocation restrictions (**context**), outcomes, maintenance conditions, information required, information updated, goal structure, plan diagrams and the source of the information the tactic is based upon.

2.4 Plan Diagrams

Each plan diagram is diagrammatic pseudo-code, based upon UML Activity Diagrams, and represents part of an overall tactic. A typical tactic will include a set of plan diagrams. Iteration can be expressed as shown in Figure 2. The plan performs a **move to waypoint** action and waits until the next waypoint has been reached. It loops until the destination has been reached.

2.5 Scenarios, Roles, Percepts, Actions, Actors, Agents

Each scenario outlines part of how a mission could play out. Roles express functional groupings and are ultimately assigned to agents. The system interacts with actors, receiving percepts and performing actions.

2.6 Capabilities, Data, Events, Messages

Capabilities allow tactics to be structured into meaningful units that can be combined into agents. Tactics can involve the use and modification of data sources, the handling of events, and messaging between agents involved in executing a group tactic.

3. RELATED WORK

There are a number of AOSE tools including those based on the Tropos [2] and O-MaSE [3] methodologies. These tools focus on the decomposition of the system into functionally distinct components and how those components relate to one another. These tools are analogous to PDT, which TDF extends in a number of ways, most notably with tactics design patterns. Design patterns have been used for mobile agents [1], but they are quite similar to those found in object oriented design. An initial investigation of design patterns for human behaviour models has not progressed to the point of implementation [6].

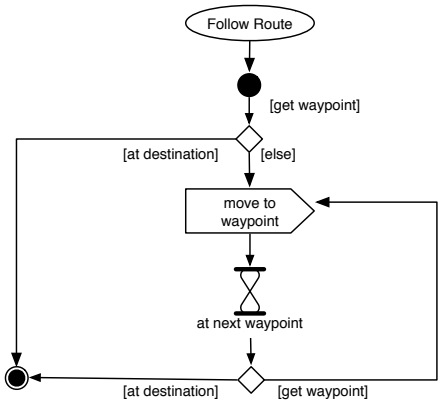

Figure 2: Waypoint Following Plan Diagram

4. CONCLUSIONS AND FUTURE WORK

TDF provides much needed design support for tactics modelling, and helps with reuse and sharing of models between developers. This is facilitated by the maintenance of the link between missions, goals, plan diagrams and implementation. This kind of support is essential for large, reusable tactics libraries.

TDF will benefit from extensions that perform consistency checking, for example, to identify goals that are not supported by a corresponding collection of plan diagrams. Management of large tactics libraries will benefit from the development of tactics ontologies with tool support within TDF. Another potential area for extension would be to address the knowledge acquisition phase, including tool support for interviewing Subject Matter Experts and linking such sources to existing or yet-to-be-defined design artefacts.

We acknowledge the support of the Defence Science and Technology Organisation, and the Defence Science Institute.

5. REFERENCES

[1] Yariv Aridor and Danny B. Lange. Agent design patterns: elements of agent application design. In *Proceedings of the second international conference on Autonomous agents*, pages 108–115. ACM, 1998.

[2] P. Bresciani, A. Perini, p. Giorgini, F. Giunchiglia, and Mylopoulos.J. Tropos: An agent oriented software development methodology. *AAMAS*, 8(3):203–236, 2004.

[3] S.A. DeLoach, W.H. Oyenan, J.C. Garcia-Ojeda, and J. Valenzuela. O-MaSE: A customizable approach to developing multiagent development processes, 2007.

[4] L. Padgham and M. Winikoff. *Developing intelligent agent systems: a practical guide*, volume 1. Wiley, 2004.

[5] A.S. Rao, M.P. Georgeff, et al. BDI agents: From theory to practice. In *Proceedings of the first ICMAS (95)*, pages 312–319. San Francisco, 1995.

[6] Glenn Taylor and Robert E. Wray. Behavior design patterns: Engineering human behavior models. In *Proceedings of the Behavior Representation in Modeling and Simulation Conference*, 2004.

[7] M. Winikoff. JACK intelligent agents: An industrial strength platform. *Multi-Agent Programming*, pages 175–193, 2005.