

Modelling Tactical Decision-Making with TDF (Tactics Development Framework)

Rick Evertsz

rick.evertsz@rmit.edu.au



Overview

- ❖ Applications of tactics modelling
- ❖ Current approaches to tactics modelling
- ❖ The TDF Methodology
- ❖ TDF Example
- ❖ Evaluation of TDF
- ❖ Future developments

Applications of Tactics Modelling



Capability Analysis

e.g., sonar performance, hull sonar signature...

Applications of Tactics Modelling



Live Fire Training

e.g., robots reacting and moving realistically, ...

Applications of Tactics Modelling



UAV Tactical Decision-Making

e.g., handling incoming missile, detecting and dealing with icing...

Applications of Tactics Modelling



Training, Mission Rehearsal

e.g., non-player characters using realistic tactics...

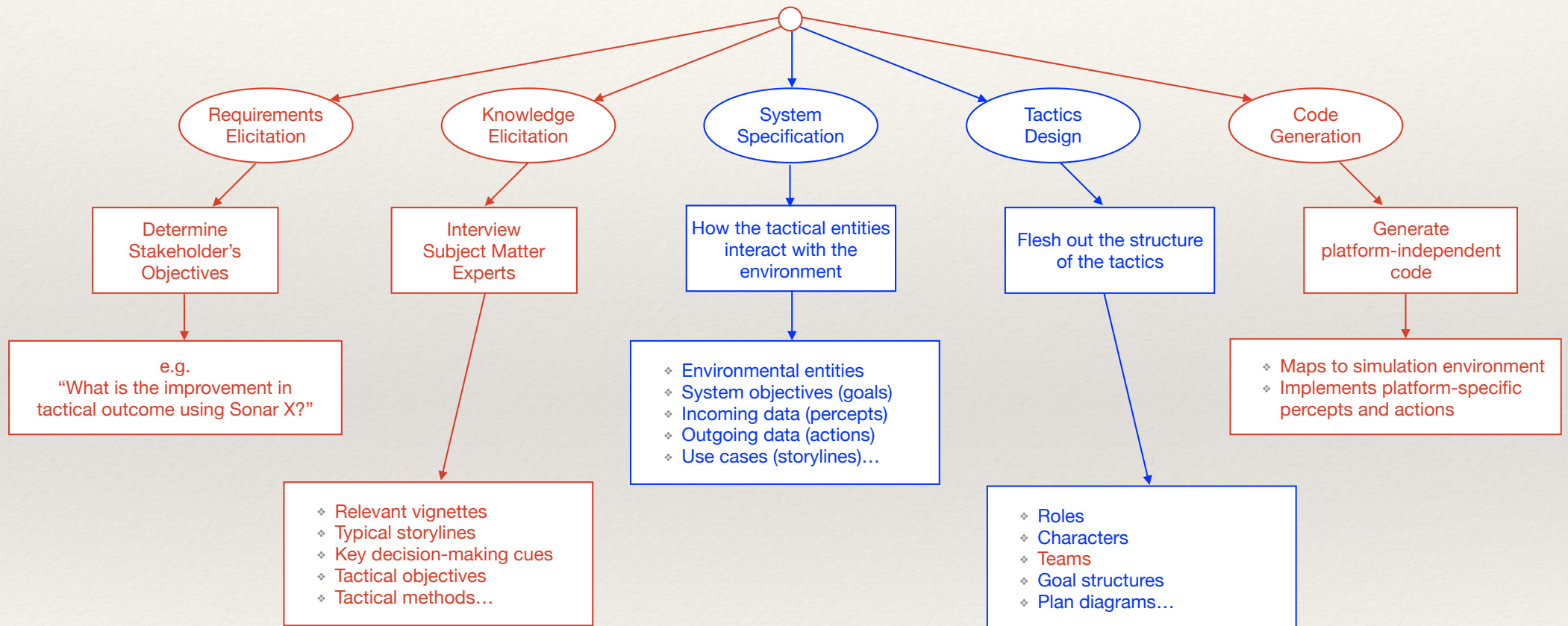
Current Approaches to Tactics Modelling

- ❖ Scripting
 - ❖ e.g. VBS Application Scripting Interface (ASI)
 - ❖ tied to virtual environment
 - ❖ must explicitly encode low-level details of decision-making
 - ❖ results in inflexible and fairly rudimentary behaviour
- ❖ Sophisticated AI models, e.g. TacAir-Soar
 - ❖ difficult to write, debug and maintain
- ❖ All models represented at the implementation level

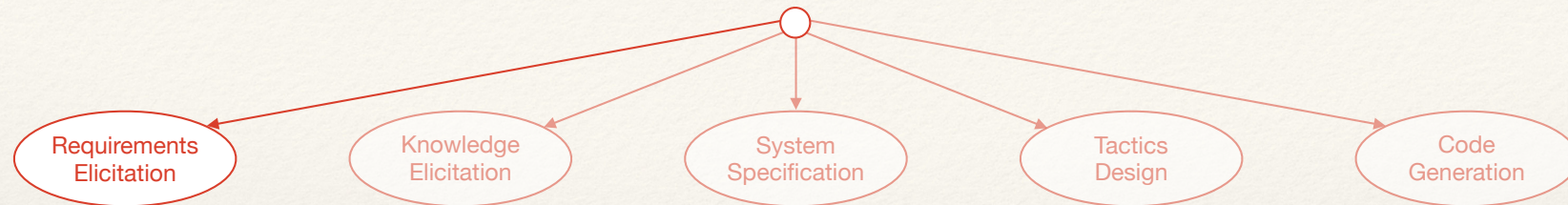
What is Needed?

- ❖ Support for the whole model development life cycle
 - ❖ requirements & knowledge elicitation
 - ❖ model specification and design
 - ❖ verification & validation
 - ❖ code generation
- ❖ Easier to design, build, understand, modify, validate
- ❖ SME can critique models
- ❖ Traceability from requirements through implementation
- ❖ Libraries of reusable models
- ❖ Independent of virtual environment

The TDF Methodology



TDF Example - Undersea Warfare



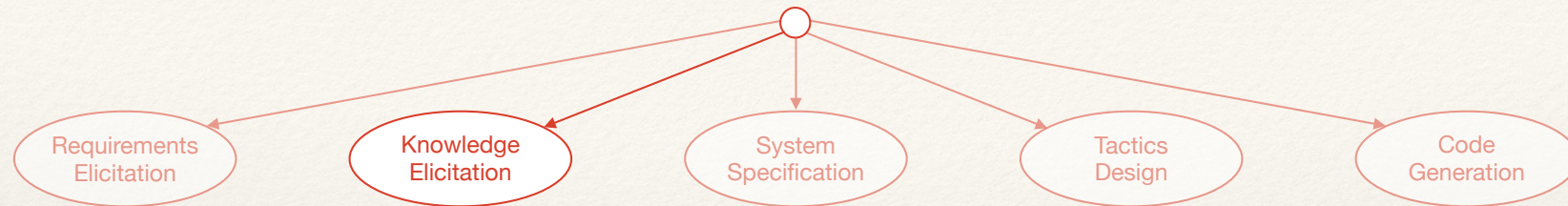
❖ Stakeholder Objective

- ❖ Purchase sonar X or Y?

❖ Modelling Objectives

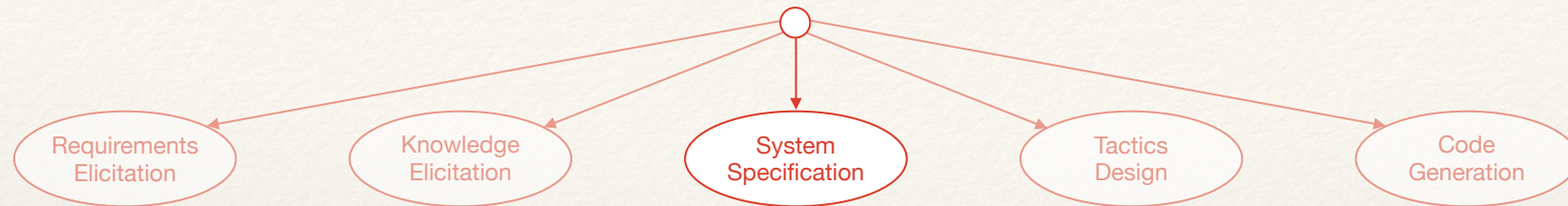
- ❖ Which sonar, X or Y, provides earliest detection in typical scenarios?
- ❖ Does the difference affect tactical outcome in typical scenarios?

TDF Example - Undersea Warfare



- ❖ Interviews with SME
- ❖ Vignettes
 - ❖ Support STG, detect/destroy RED submarine in strait
- ❖ Concept Map (SO/MO root)
- ❖ RPD (Recognition-Primed Decision-Making)
 - ❖ 80-95% correspondence to expert commanders
 - ❖ uncertainty, time pressure
 - ❖ cues, expectancies, goals, actions

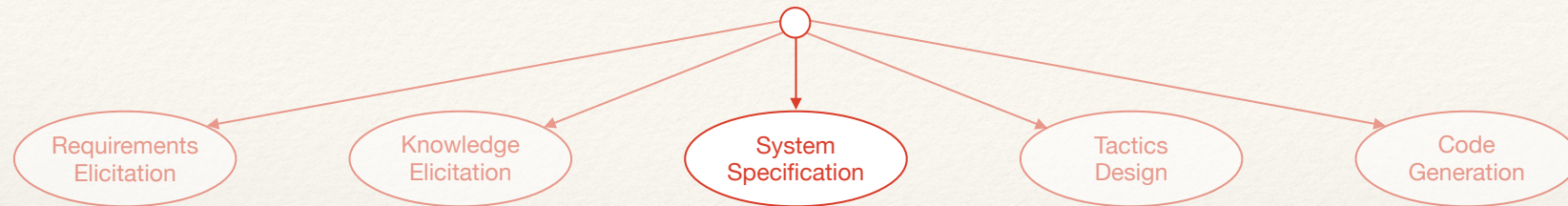
TDF Example - Undersea Warfare



❖ Identify system-level artefacts

- ❖ missions
- ❖ storylines
- ❖ goals
- ❖ percepts (cues) and actions
- ❖ actors (external entities)
- ❖ roles

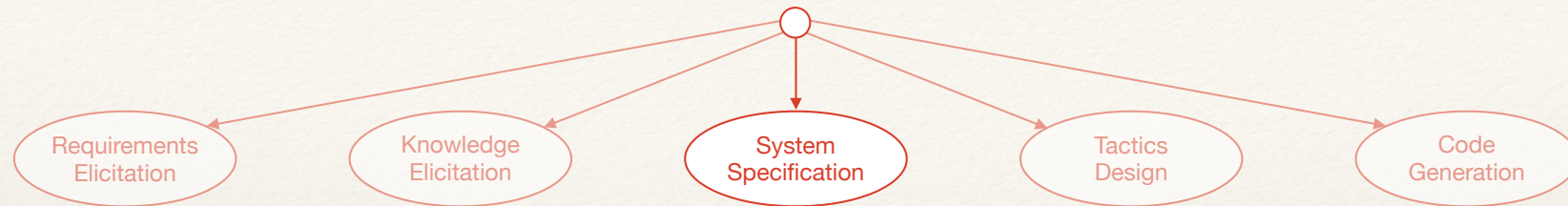
TDF Example - Undersea Warfare



❖ Mission

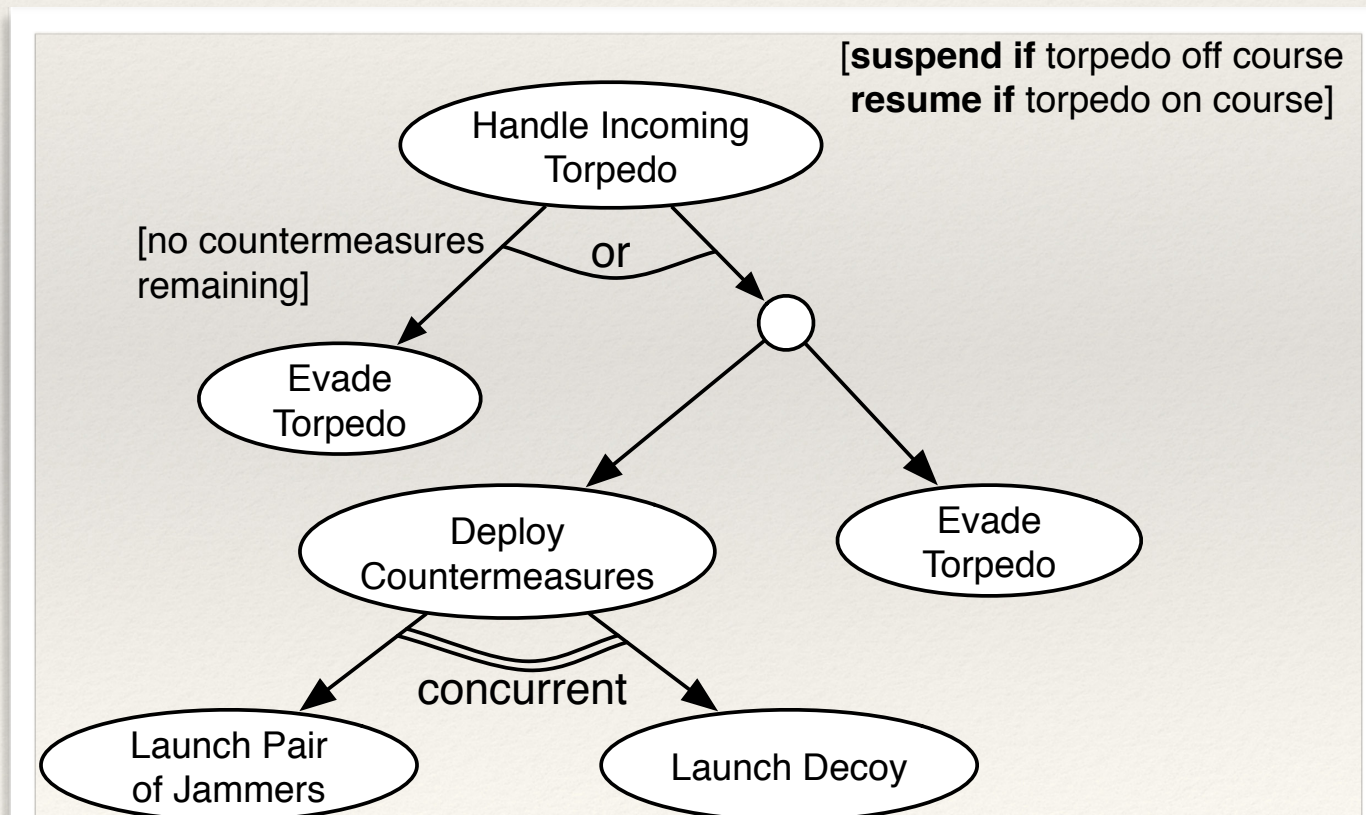
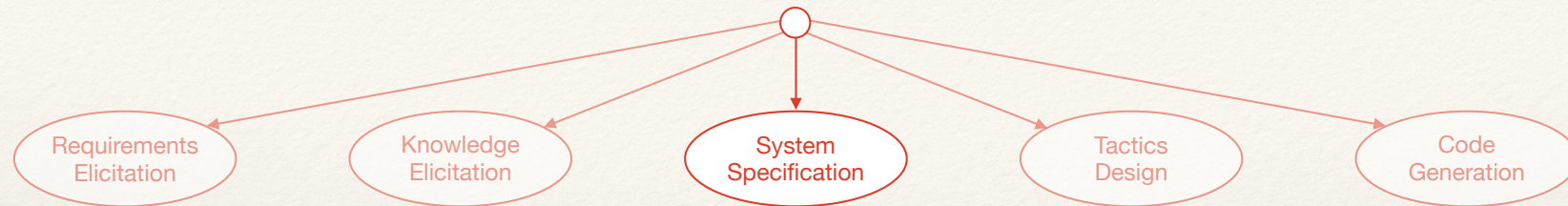
- ❖ Objective: destroy RED submarine
- ❖ Secondary Objectives: identify other maritime vessels
- ❖ Mission Statement: protect STG, expect RED in strait, intercept and destroy RED submarine
- ❖ Operational Constraints: stealth, do not engage others
- ❖ Risks: number of RED submarines is unknown
- ❖ Opportunities: RED moving fast, so more detectable
- ❖ Storylines: Navigate to AO, Search for Target, Classify Contact, Attack Target, Incoming Torpedo...
- ❖ Data: mission route, map, undersea contour map...

TDF Example - Undersea Warfare

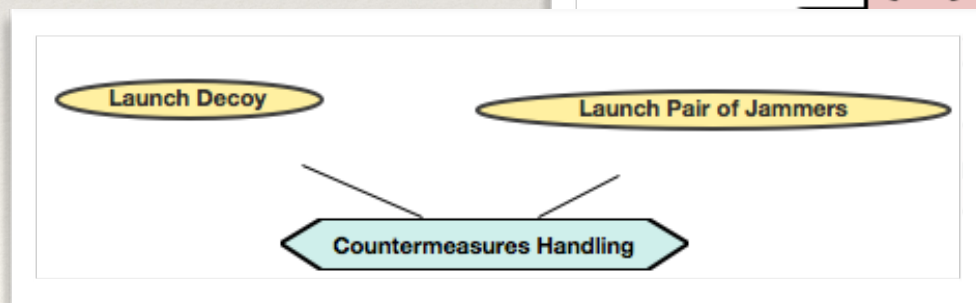
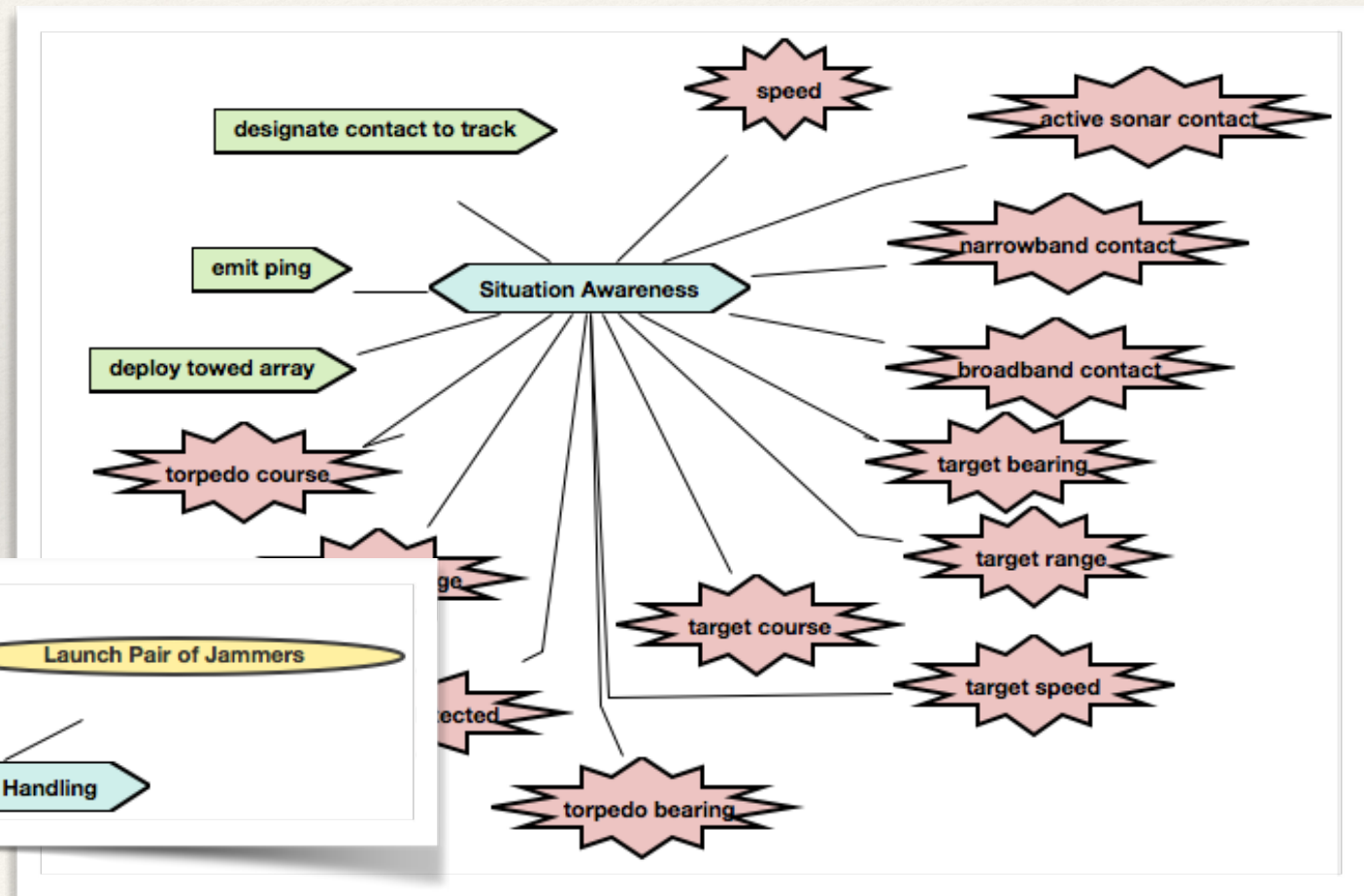
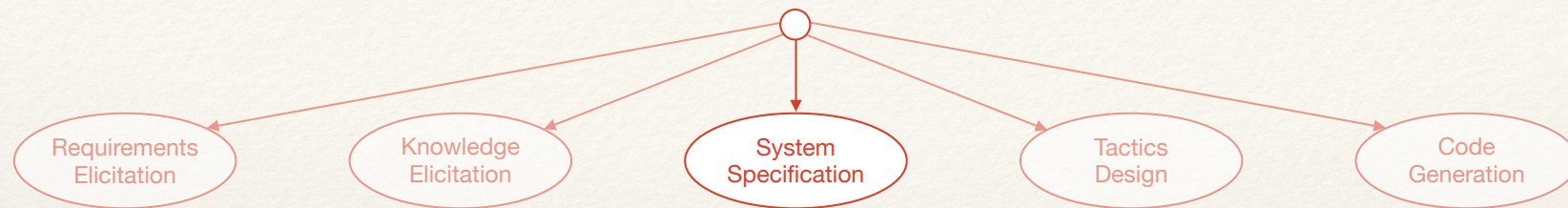


- ❖ “Handle Incoming Torpedo” Storyline
 - ❖ Torpedo detected (percept)
 - ❖ Estimate Bearing and Distance (goal)
 - ❖ Deploy Countermeasures (goal)
 - ❖ Evade Torpedo (goal)

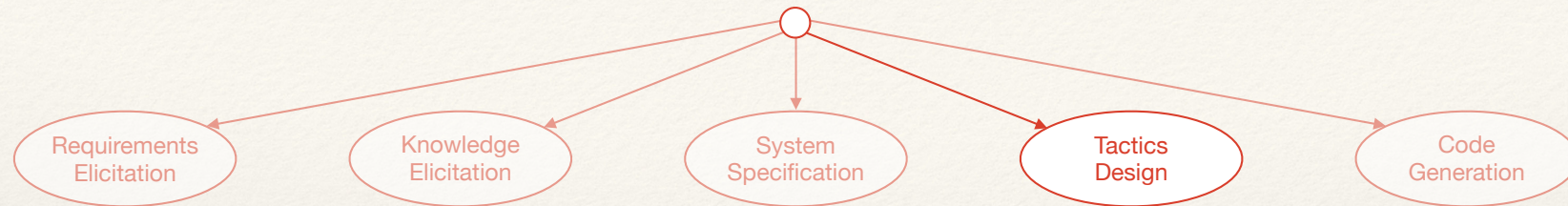
TDF Example - Undersea Warfare



TDF Example - Undersea Warfare

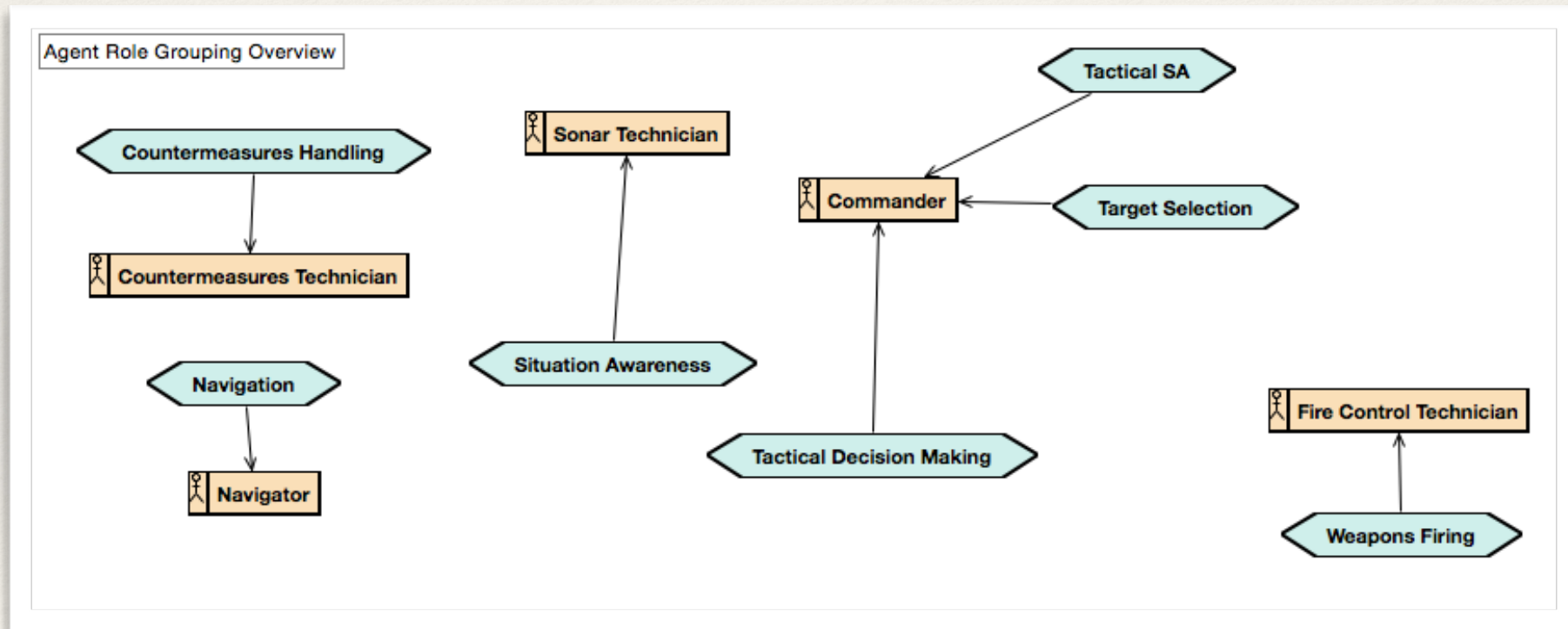
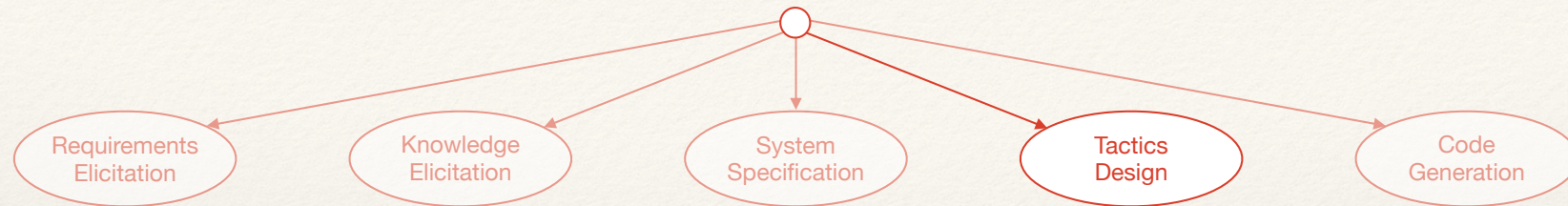


TDF Example - Undersea Warfare

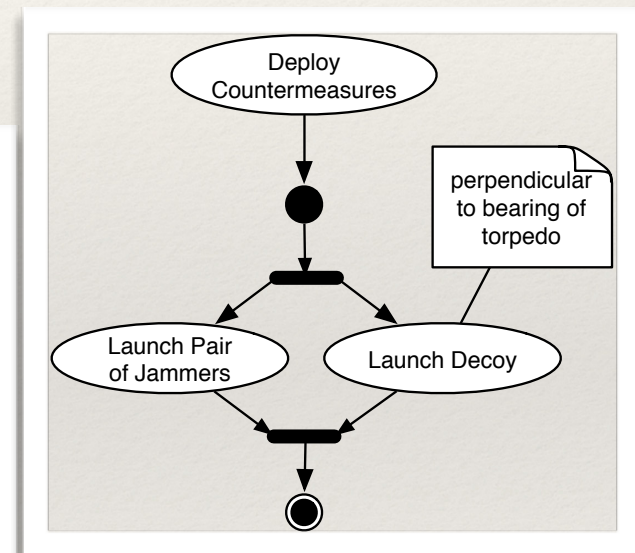
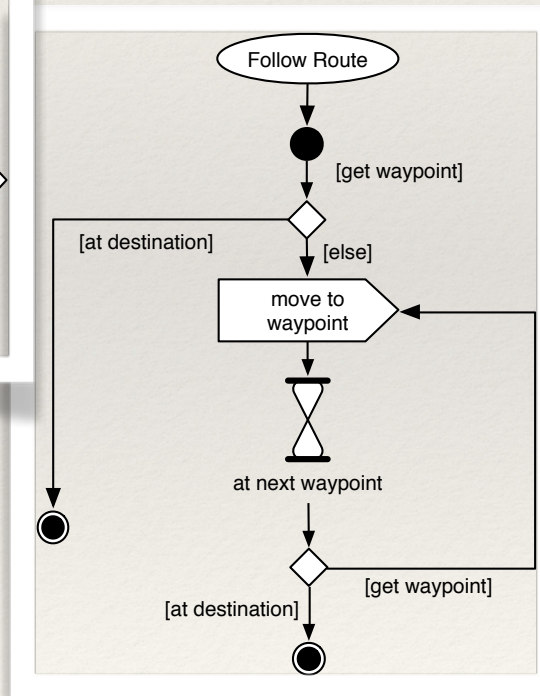
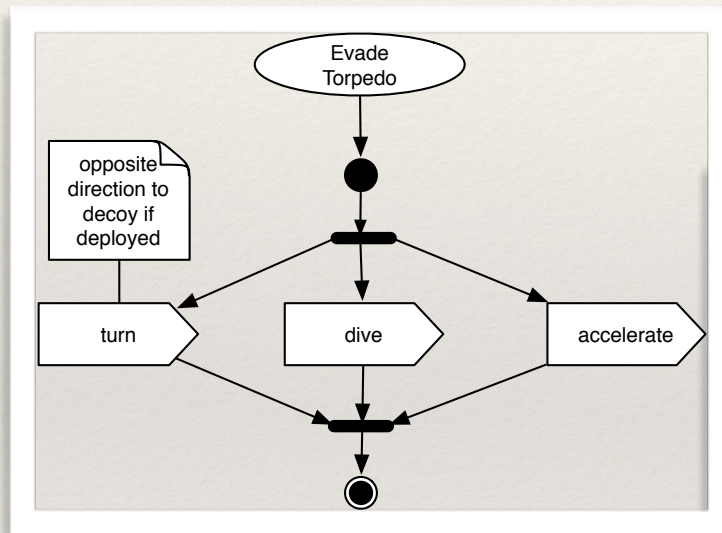
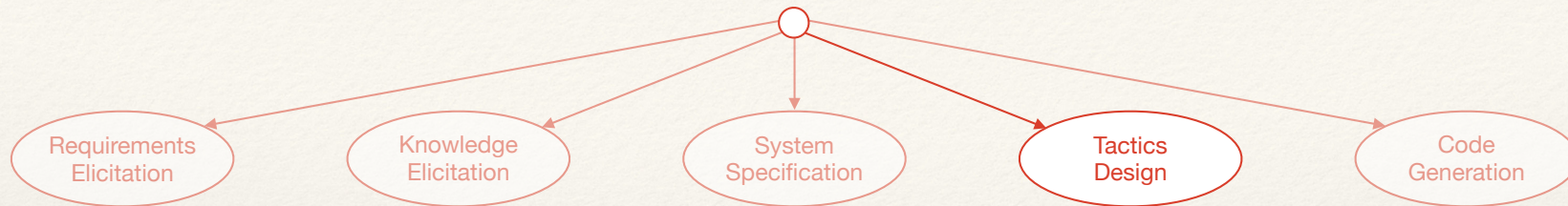


- ❖ Assign roles to characters and teams
- ❖ Tactics Design Patterns
 - ❖ objective (Escape Torpedo)
 - ❖ trigger (Incoming Torpedo Detected)
 - ❖ problem and solution descriptions
 - ❖ context (No Countermeasures Remaining)
 - ❖ goal structures
 - ❖ plan diagrams
 - ❖ outcomes, restrictions, information updated, source

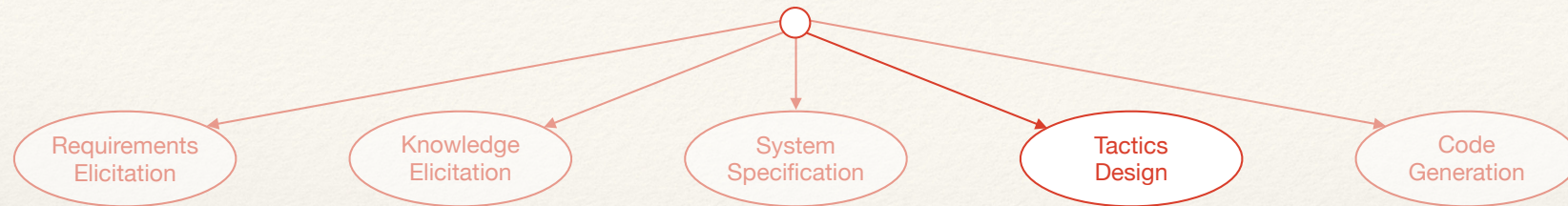
TDF Example - Undersea Warfare



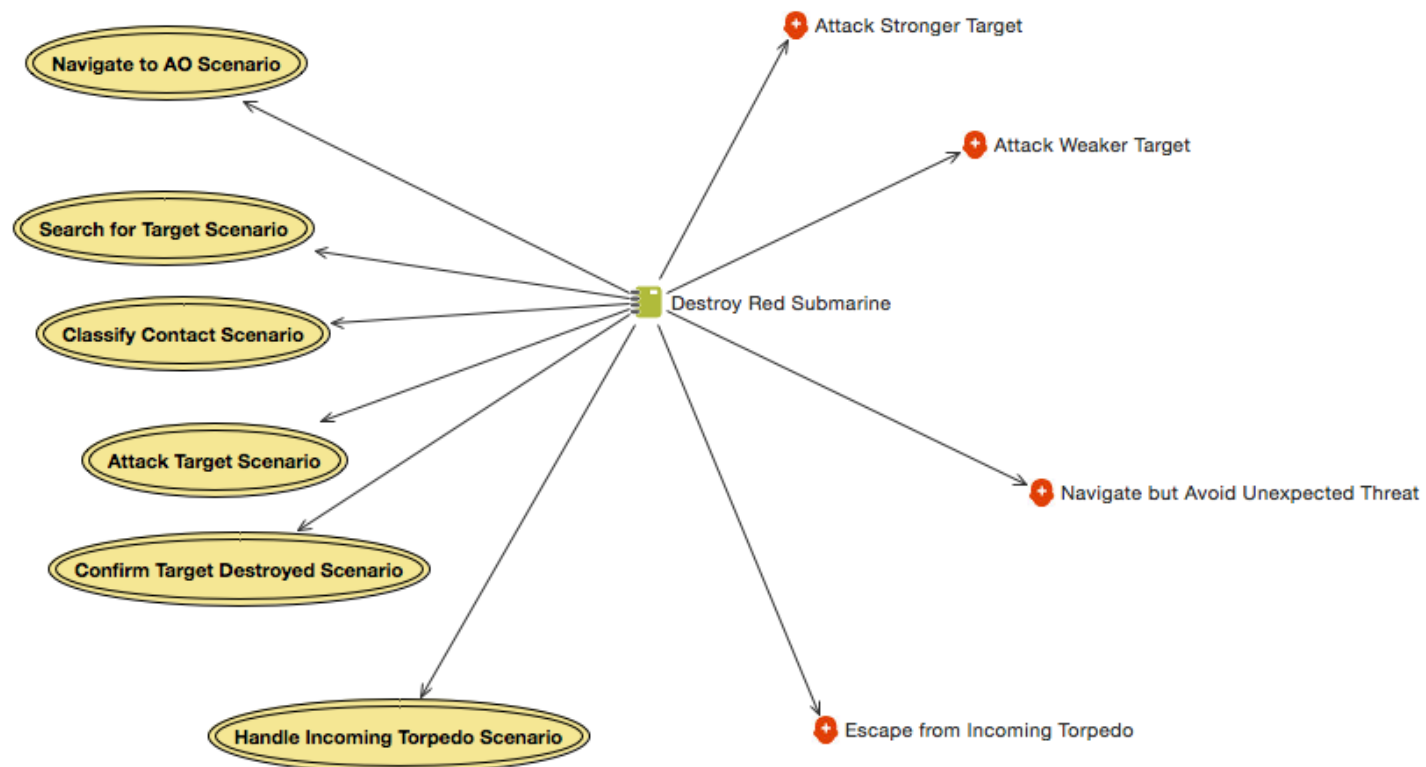
TDF Example - Undersea Warfare



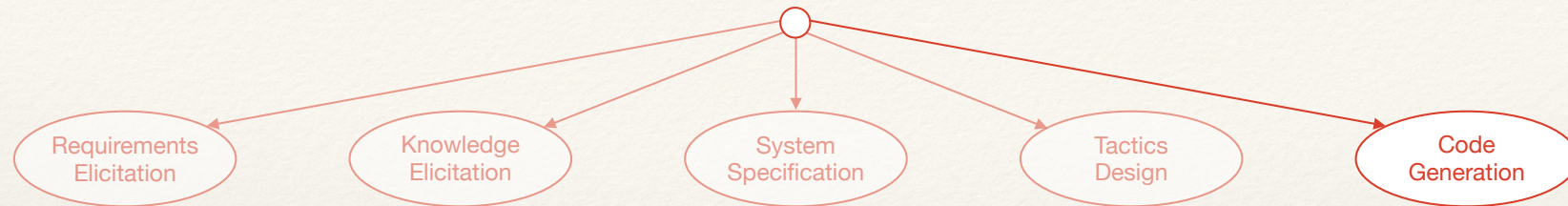
TDF Example - Undersea Warfare



Mission Diagram



TDF Example - Undersea Warfare



- ❖ Generates JACK and GORITE code stubs
- ❖ TDF code is demarcated
- ❖ Any changes outside the demarcated areas are preserved

Evaluation of TDF

- ❖ USW analysts have applied to their tactical simulations
 - ❖ High-level tactics view facilitates design
 - ❖ Helps predict how the characters will behave at runtime
 - ❖ Tens of thousands of Monte Carlo runs
 - ❖ Facilitates reuse and building of tactics libraries
- ❖ TDF vs. UML
 - ❖ UAV photoreconnaissance scenario
 - ❖ TDF 82.4% vs. 66.4% for UML ($p < 0.025$)

Benefits of TDF

- ❖ Supports the whole model development life cycle
- ❖ Will facilitate requirements and knowledge elicitation
 - ❖ Subject Matter Expert can critique models
- ❖ Diagrammatic specification of tactics
- ❖ Promotes reuse of tactics across implementation platforms
- ❖ Traceability from requirements through to implementation

Future Developments

- ❖ Design of team structures
- ❖ Formal verification of designs
- ❖ Extensions to TDF tool
- ❖ Code generation for other implementation languages
- ❖ Integration with major simulation platforms, e.g. VBS3



with support from



Australian Government
Department of Defence
Defence Science and
Technology Organisation



defence science institute
Creating Defence Science Research Networks for Australia