Developing Aspect Oriented Software Components for C2 Systems

İskender YAKIN, ASELSAN-SST
Bahar PAMUK, ASELSAN-MGEO

Outline

- Introduction
  - C2 Software Development
  - Problems
- Background
  - OO Development
  - Design Patterns
  - MDSD
  - AOSD
- Related Work
- Case Study
  - TALOS Mission Planning Software
  - C2 Concerns
    - Communication
    - Verification
    - Mission Timeline
    - Simulation
- Object Oriented Design
  - Composite Pattern
  - Observer Pattern
- Identifying Aspects
- Discussion
- Conclusion

C2 Software Development

- Development phases
  - Domain Analysis
    - Identifying system and software requirements
  - System and software design
  - Implementation
    - Adapting existing software components
    - Integration
  - Testing
    - Component tests
    - System tests
  - Maintenance
    - New requirements

Problems

- Requirements
  - Changing requirements
  - New requirements
    - Also affects existing code
- Project specific design
  - Low reusability
- Testing
  - Testing the whole system because of minor changes
- Maintenance
  - Code modification in many classes or components because of minor changes
- All these problems are faced as a result of **crosscutting concerns**.

Background

- C2 Software Development
  - OO approach
    - Crosscutting concerns
  - Design patterns
    - May slightly reduce crosscutting
  - MDSD
    - Can be used to develop core software components or concerns
  - AOSD
    - Eliminates crosscutting through accumulation of code addressing the same concern into a single component

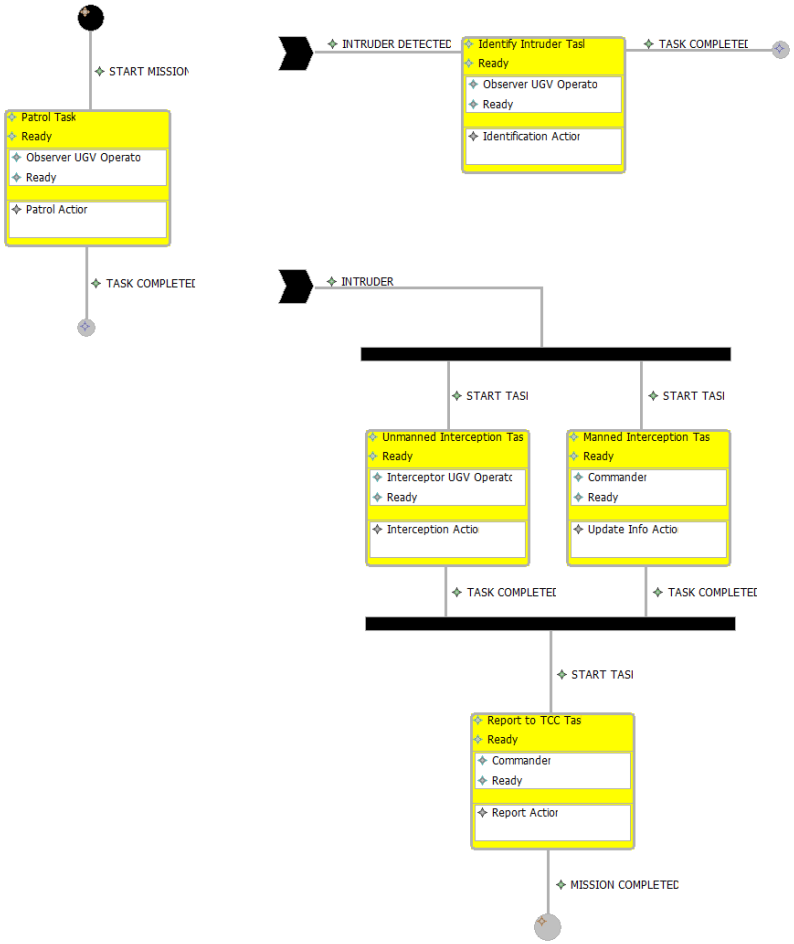
Related Work

- Previous work on C2 and C2 concerns

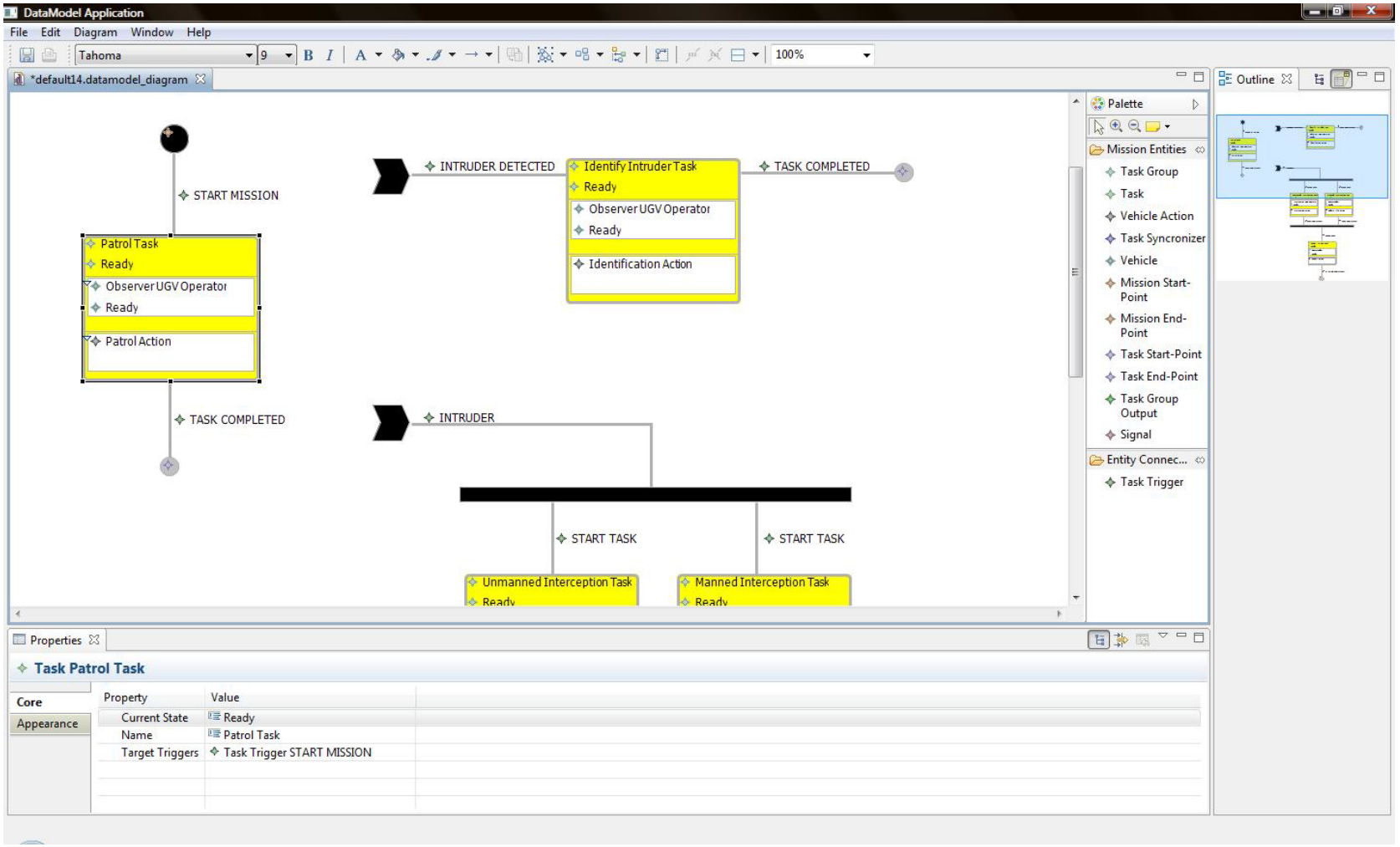
Case Study: TALOS Mission Planning

- Commander plans a mission as an activity diagram.
- A mission consists of tasks and start stop between tasks.
- Each task is assigned to an operator or to the commander.
- Operator executes a task and reports
  - Events during the task execution
  - Completion of the assigned task
- Upon reported events and completion messages mission plan is iterated and new tasks are assigned with respect to the order of tasks in the mission plan.

An Example Mission Plan



TALOS Mission Planning Software



C2 Components/Concerns

- **Communication Component**: This component encapsulates the communication part of the existing software.

C2 Components/Concerns

- **Verification Component**: This component verifies that mission plan is suitable to be executed.

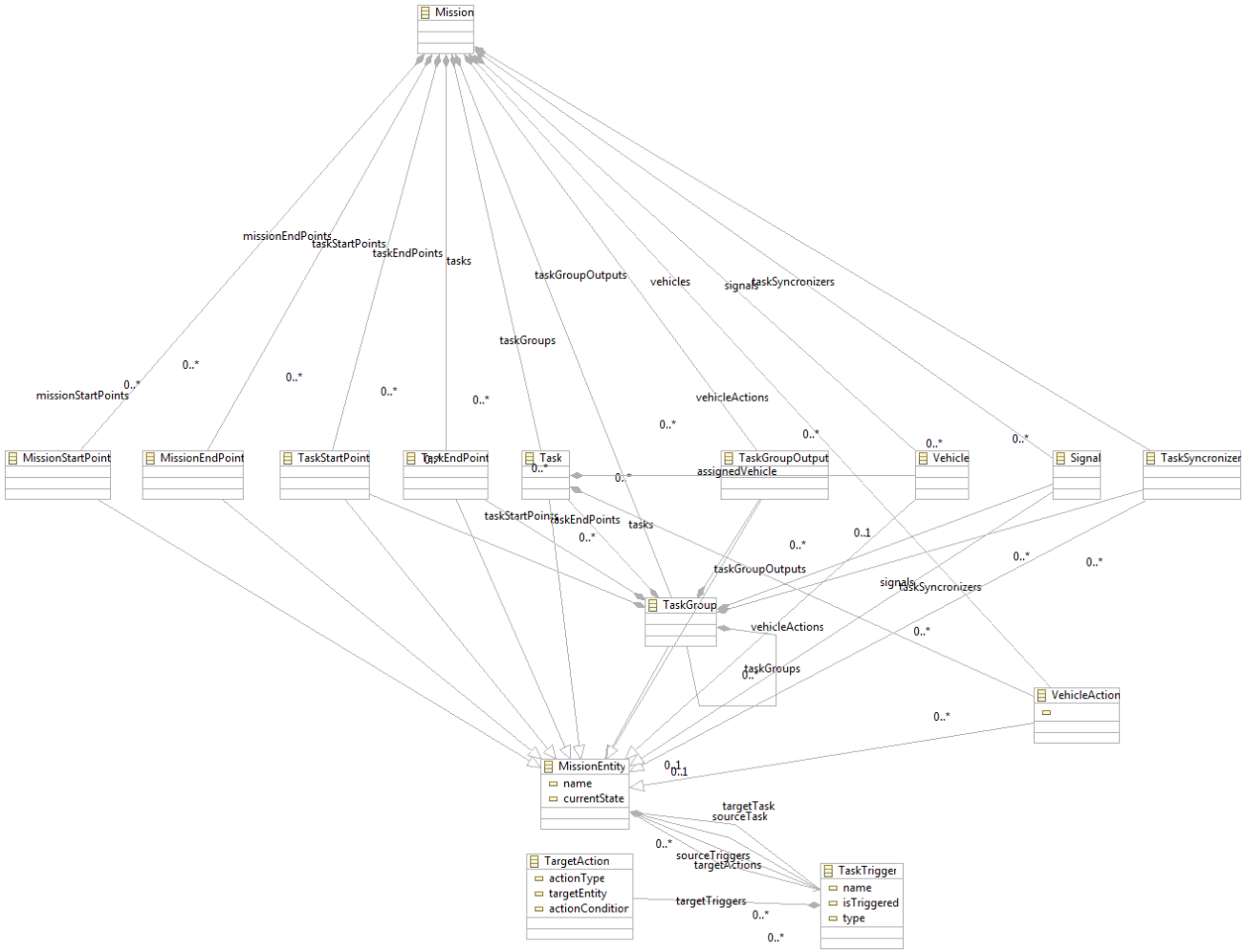
C2 Components/Concerns

- **Mission Timeline Component**: This component is responsible for displaying the details of mission in term of duration of each task along with start and finish times. TALOS Mission Planning Software is developed through MDSD, so we do not want to edit the auto-generated code manually. Since the metamodel of each task does not include start and finish times as data fields it will be required to maintain this information externally with this aspect oriented component.

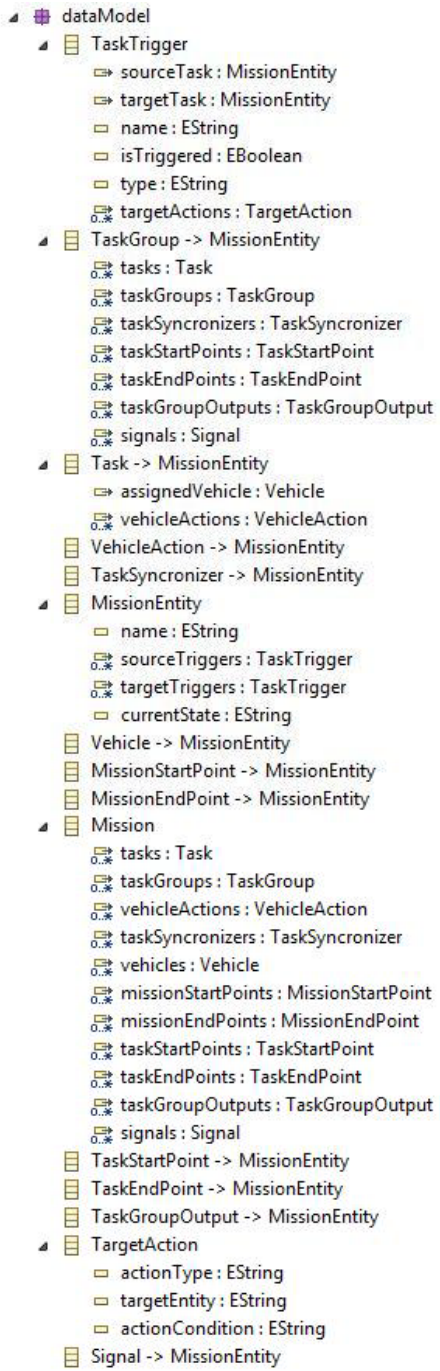
C2 Components/Concerns

- **Simulation Component:** This component simulates the existing mission plan and outputs the simulation parameters such as the order of executed tasks, their duration and the exertion time of the whole mission for each alternative task flows.

Object Oriented Design

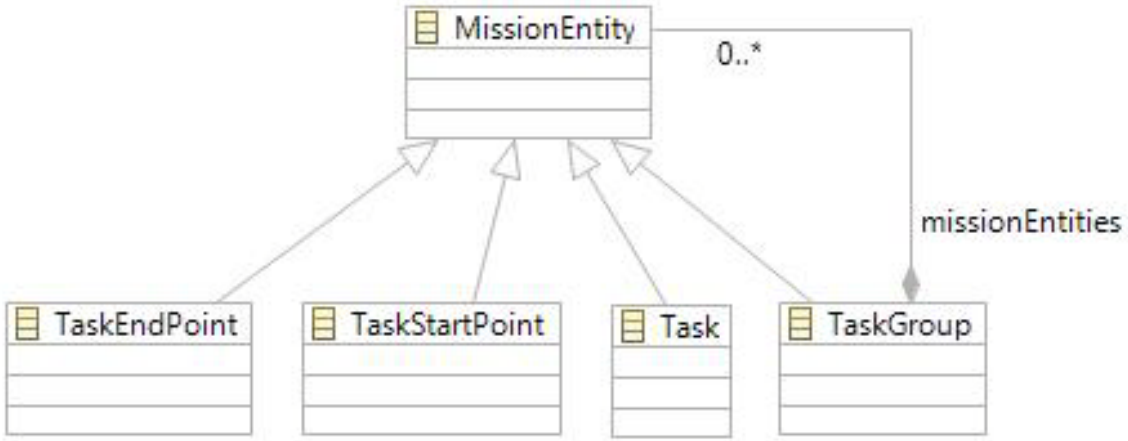


Data Model


dataModel
TaskTrigger
sourceTask : MissionEntity
targetTask : MissionEntity
name : EString
isTriggered : EBoolean
type : EString
targetActions : TargetAction
TaskGroup -> MissionEntity
tasks : Task
taskGroups : TaskGroup
taskSyncronizers : TaskSyncronizer
taskStartPoints : TaskStartPoint
taskEndPoints : TaskEndPoint
taskGroupOutputs : TaskGroupOutput
signals : Signal
Task -> MissionEntity
assignedVehicle : Vehicle
vehicleActions : VehicleAction
VehicleAction -> MissionEntity
TaskSyncronizer -> MissionEntity
MissionEntity
name : EString
sourceTriggers : TaskTrigger
targetTriggers : TaskTrigger
currentState : EString
Vehicle -> MissionEntity
MissionStartPoint -> MissionEntity
MissionEndPoint -> MissionEntity
Mission
tasks : Task
taskGroups : TaskGroup
vehicleActions : VehicleAction
taskSyncronizers : TaskSyncronizer
vehicles : Vehicle
missionStartPoints : MissionStartPoint
missionEndPoints : MissionEndPoint
taskStartPoints : TaskStartPoint
taskEndPoints : TaskEndPoint
taskGroupOutputs : TaskGroupOutput
signals : Signal
TaskStartPoint -> MissionEntity
TaskEndPoint -> MissionEntity
TaskGroupOutput -> MissionEntity
TargetAction
actionType : EString
targetEntity : EString
actionCondition : EString
Signal -> MissionEntity

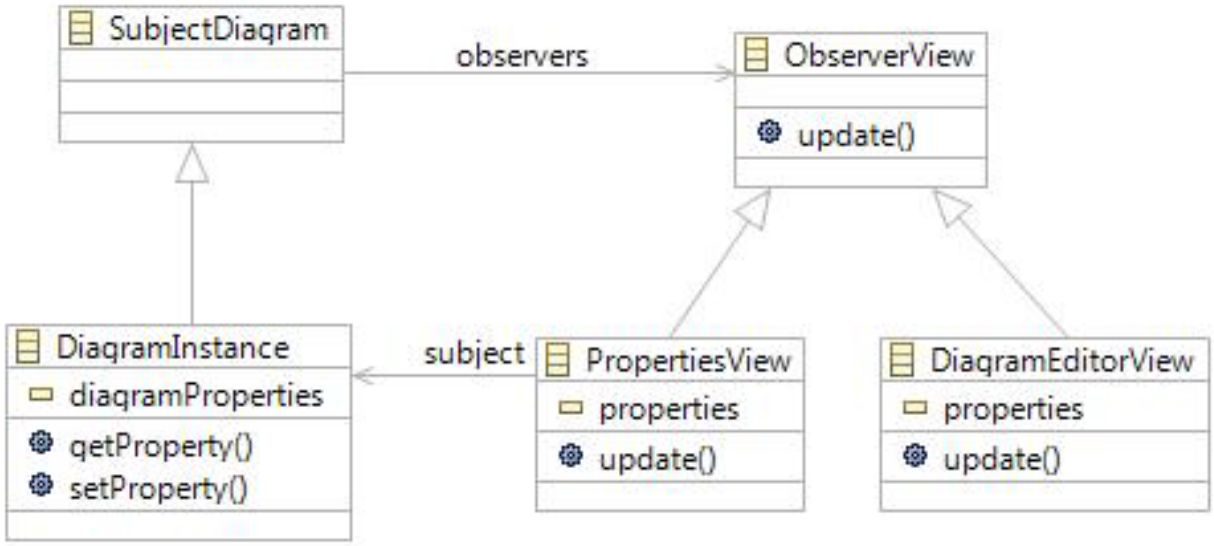
Design Patterns

- Composite Pattern



Design Patterns

- Observer Pattern



Conclusion

- AOSD applied for several C2 concerns
  - Communication concern
  - Verification concern
  - Timeline concern
  - Simulation concern
- Aspect corresponding to concerns were implemented as separate software components
- Through AOSD
  - Reduced effort in all phases of software life cycle especially in,
    - Integration
    - Maintenance
    - Testing