26th annual INCOSE international symposium

Edinburgh, UK
July 18 - 21, 2016

Applying Model-based SE Techniques for Dependable Land Systems

Richard Payne, John Fitzgerald
Newcastle University, UK
richard.payne@newcastle.ac.uk
john.fitzgerald@newcastle.ac.uk

Jeremy Bryans
Coventry University, UK
jeremy.bryans@coventry.ac.uk

Elspeth Winthorpe
MOD, UK
elspeth.winthorpe100@mod.uk

Overview

- **Dependability in Systems of Systems**
- LOSA and the Study
- Model-based SE Techniques
- Conclusions
- Wrap Up

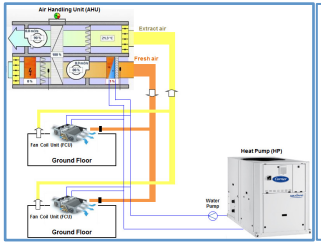
Our Work

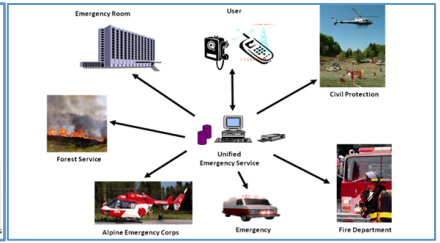
Design technology (foundations, methods, tools) for:

- Systems of Systems (SoS)
- Cyber-Physical Systems (CPS)

We focus on *model-based design:*

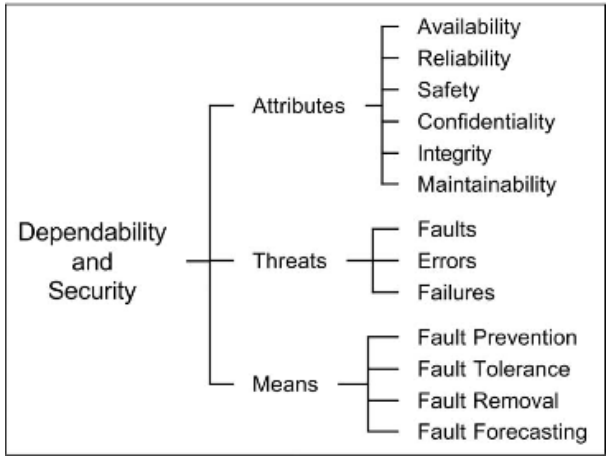
- Models as a basis for collaborative development
- Machine-assisted analysis of models as a means of assessing system dependability





Dependability

The dependability of a system is its ability to deliver service that can *justifiably be trusted*



A. Avižienis, et al., "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Trans. *Dependable and Secure Computing* 1(1):11-33, Jan.-Mar 2004.

Dependability and SoSs

- Operational and Managerial Independence
  - The relationship between CS behaviours or services may not be known to the engineer
- Distribution
  - Communication failures and trustworthiness of networks
  - Time-lags, bandwidth and synchronisation issues
- Evolution
  - Need to re-evaluate conformance properties after evolutions
- Emergence
  - SoS-level properties must be verifiable from SoS composition

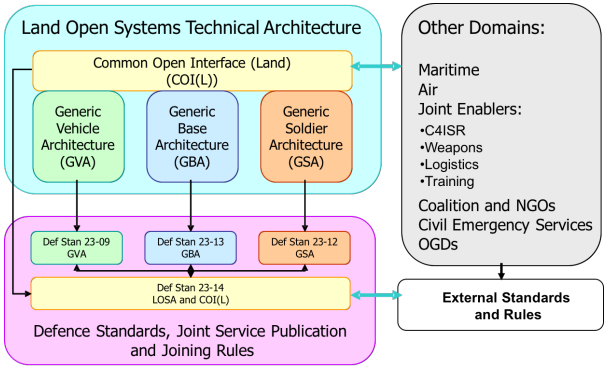
Overview

- Dependability in Systems of Systems
- **LOSA and the Study**
- Model-based SE Techniques
- Conclusions
- Wrap Up

Land Open Systems Architecture

- Ops rooms that are not integrated.
- 24 different types of battery on the dismounted soldier.
- Electro-Magnetic incompatibilities.
- Bases that are not designed for simple facilities management.

Land Open System Architecture (LOSA) – an approach to ensure the delivery of integrated, interoperable, agile force elements across the Land Domain in UK MOD.



Purpose of the Study

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

To establish the feasibility of a pragmatic method of enabling the assessment of security, safety and reliability dependencies within a given SoS within the LOSA context.

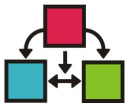
The method should

- enable an understanding stakeholder roles
- allow modelling of power, not just digital phenomena
- demonstrate ability to analyse properties
- allow modelling of “what-if” scenarios

The Approach

- Use design time model-based techniques
- Use state of the art in SoS and CPS engineering
  - Completed EU projects on SoS and embedded systems
  - New EU project on CPS

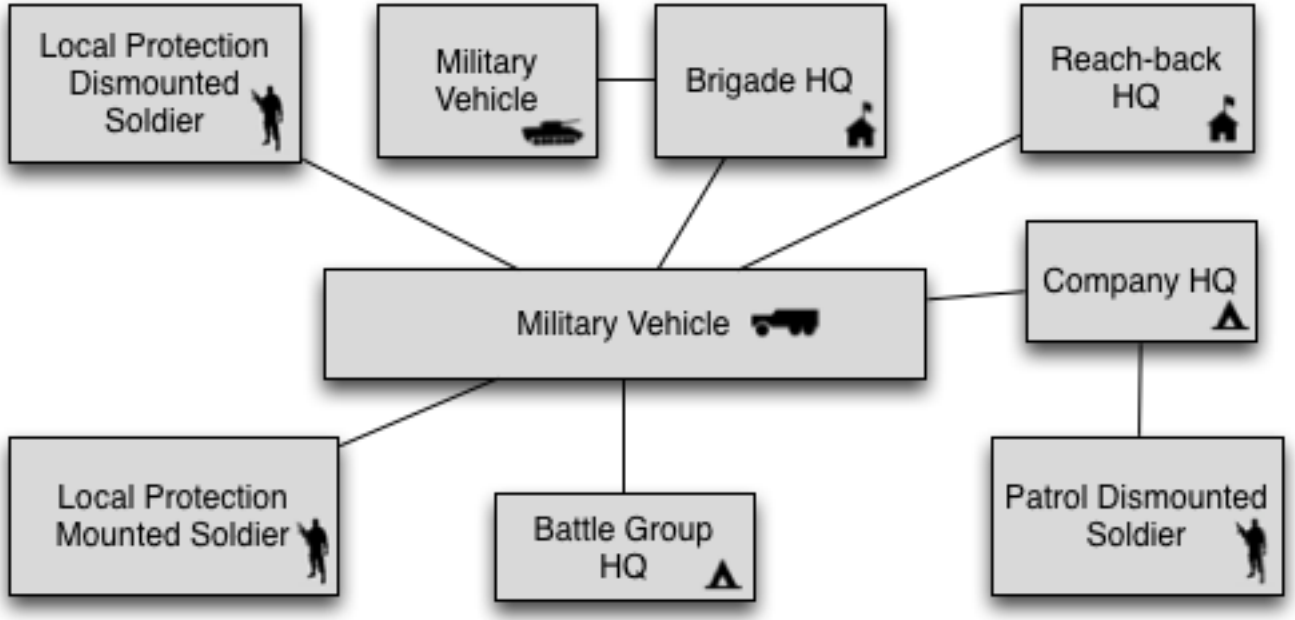
COMPASS



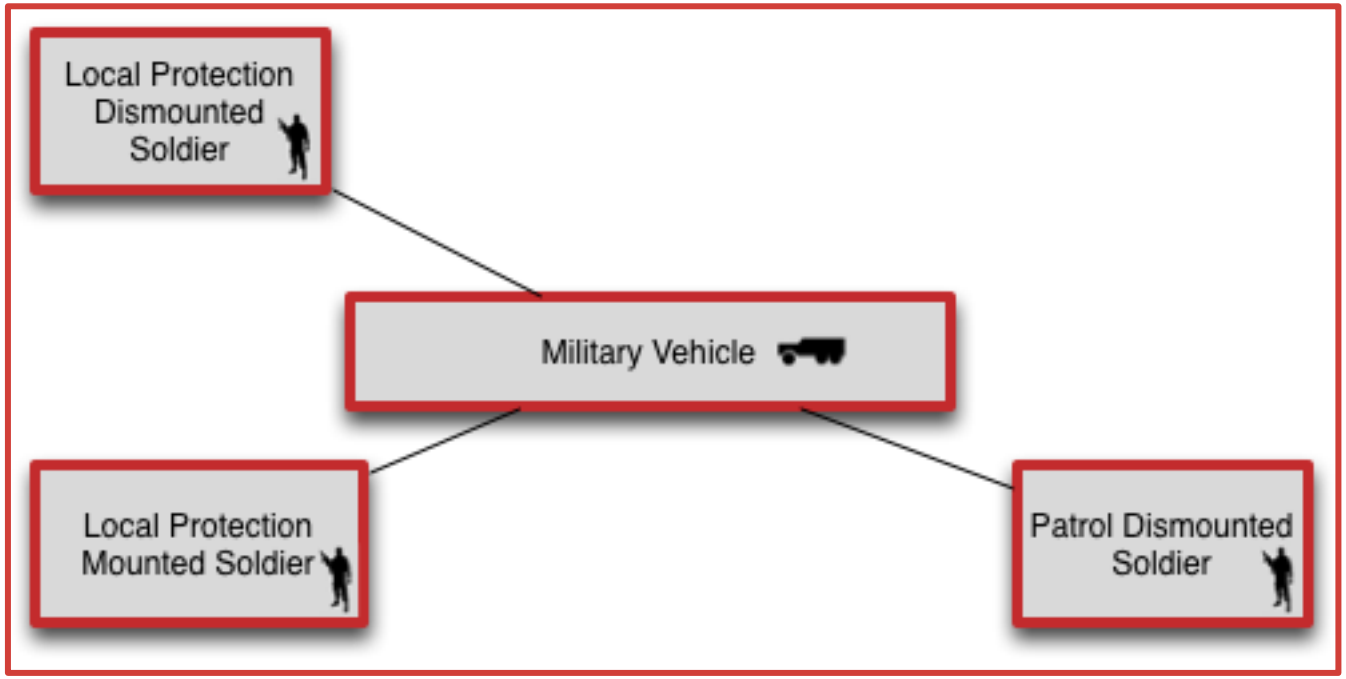
DESTECS

INTO-CPS

LOSA SoS Feasibility Example



LOSA SoS Feasibility Example



SoS boundary defined with simplified constituents

Representative Dependability Properties

- Simple scenario is defined to analyse example behaviour
- Representative dependability properties identified

Property	Comments
Availability	What should we expect a function is not available?
Availability	Functionality available if power below a threshold?
Reliability	What happens if a message is lost?
Reliability	What happens if there are faults with power supplies?
Safety	Can we distinguish between safe and unsafe states?

Overview

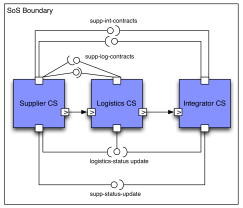
- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
  - **SoS Technologies**
    - **Architectural Modelling: *Interface Contracts***
    - Formal Modelling: *COMPASS Modelling Language (CML)*
    - Fault Modelling: *Fault Modelling Architectural Framework*
  - CPS Technologies
- Conclusions
- Wrap Up

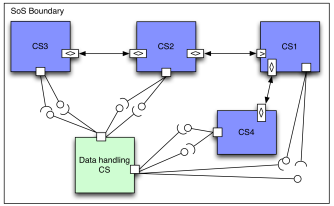
Architectural Modelling

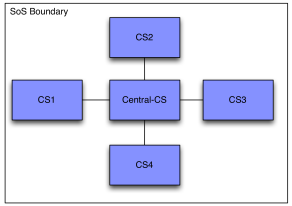
26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

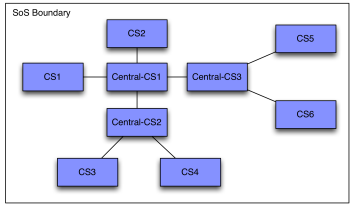
- SoSs present significant engineering challenges
  - Can we justifiably rely on CS behaviour?
  - Bound behaviours that can be relied upon *without over-constraining*
  - *Promote desirable* and *limit undesirable* emergent behaviours

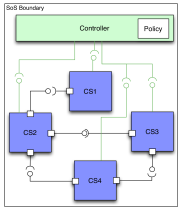
- Modelling patterns used to define SoS structure and behaviour
  - Part of SoS Patterns work ongoing in INCOSE SoS Working Group

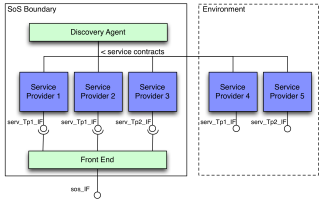






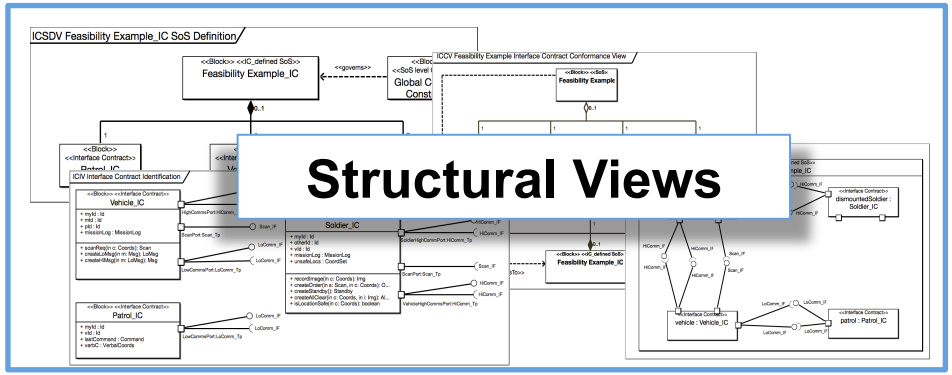


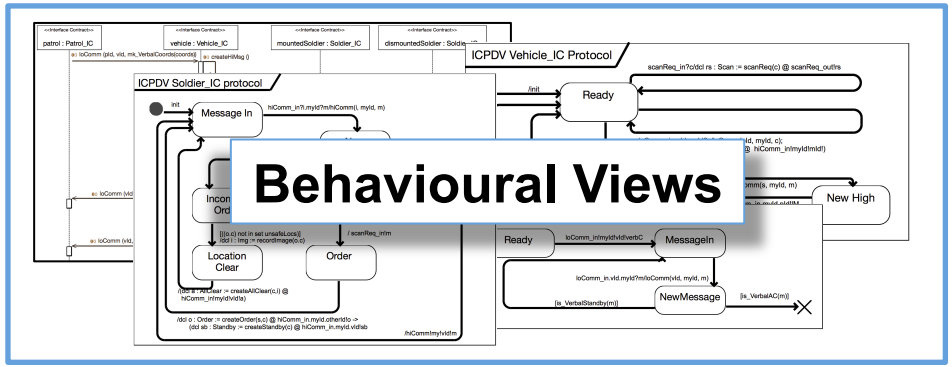




The Interface Contract Pattern

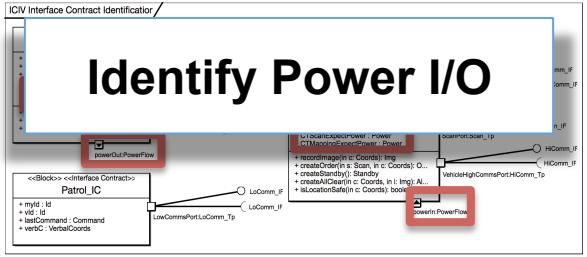
- Use *interface contract* pattern to define the behavior provided by and required by CSs
- Collection of viewpoints for modelling and defining the contracts of a SoS
  - Structure: contract definition, composition and conformance
  - Behaviour: contract protocols and scenarios
- Defined and implemented as a SysML profile

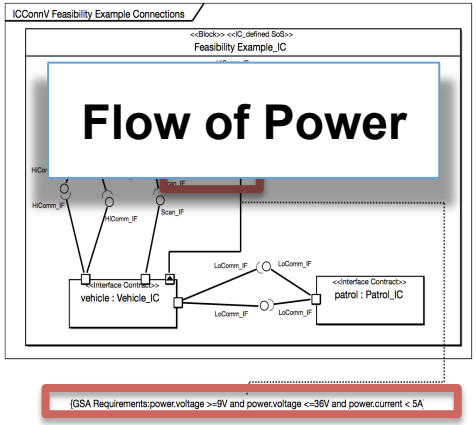


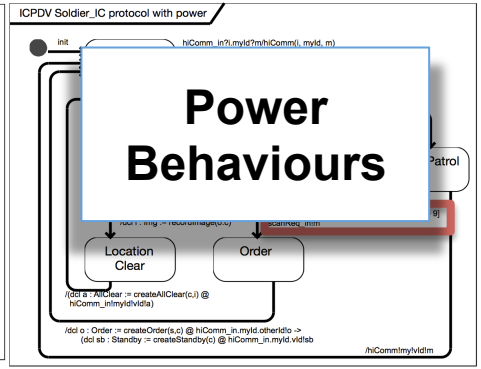


Modelling Power in ICs

- Existing pattern allows only representation of digital phenomena
  - Stakeholder meetings highlighted need for modelling power (and in future water, and other physical flows)
- Extend IC pattern to model:
  - Identify power *inputs/outputs*
  - *Flow of power* between CSs
  - *Behavioural constraints* based on physical properties







Dependability Properties

Property	Comments
Availability	Can model the behaviour when a function is not available, and analyse the outcome using behavioural views
Reliability	Can model messages being lost and analyse the outcome behavioural views
Safety	We distinguish transitions that lead to unsafe states using protocol definition views
Continuous properties	Can model power as a continuous variable and include in transition guards

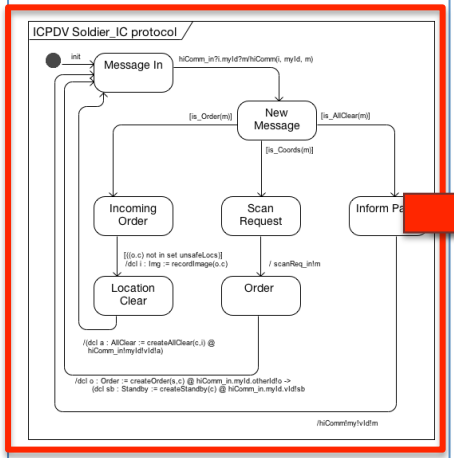
Overview

- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
  - **SoS Technologies**
    - Architectural Modelling: *Interface Contracts*
    - **Formal Modelling: *COMPASS Modelling Language (CML)***
    - Fault Modelling: *Fault Modelling Architectural Framework*
  - CPS Technologies
- Conclusions
- Wrap Up

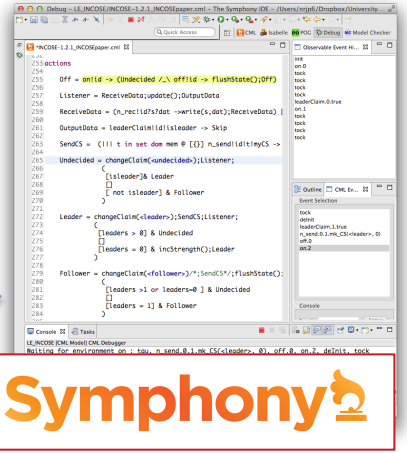
Formal Modelling

- CML (COMPASS Modelling Language) developed for modelling SoSs
  - Based on well-established formal languages with mathematical semantics
- Can model data, functionality, event ordering and communication
- Range of formal analysis techniques
- Proof of concept tools developed for translating models from SysML into CML

Analysing the Model




```
process Soldier_IC = mid,otherid,vid :
Id @
begin …
 actions
 MessageIn =
   hiComm_in?i.myId?m ->
      (hiComm(i, myId, m);
NewMessage(m))

 NewMessage = m : Msg @
  [is_Order(m)] & IncomingOrder(m)
  []
  [is_Coords(m)] & ScanRequest(m)
  []
  [is_AllClear(m)] & InformPatrol(m)

…
@
 init ->  MessageIn
end

process Patrol = …
process DismountedSoldier = …
process Vehicle = …
process MountedSoldier = …

process FeasibilityExample_IC = …
```



Symphony

Symphony Tool Platform

- Analyse scenario emergent behaviour
- Simulate execution of model
- Model checking and theorem proving available

Dependability Properties

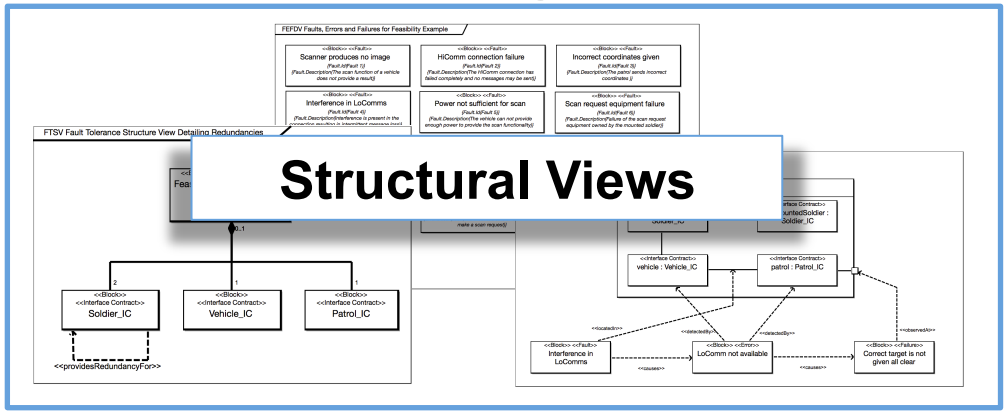
Property	Comments
Availability	We can explore the consequences of functional interfaces not being available using simulations
Reliability	We can simulate lost messages and explore consequences
Safety	We can use invariants to describe safe and unsafe states, and use analysis tools to find out whether the system ever enters the unsafe states
Continuous properties	We can model power fluctuations as discrete state changes, but not as continuous variables

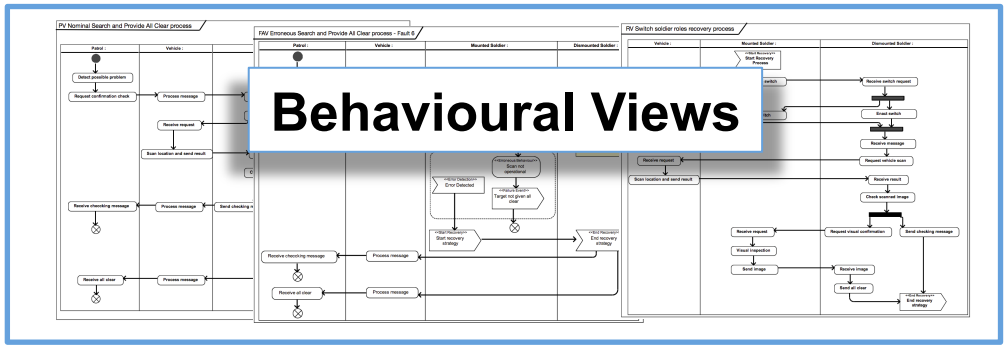
Overview

- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
  - **SoS Technologies**
    - Architectural Modelling: *Interface Contracts*
    - Formal Modelling: *COMPASS Modelling Language (CML)*
    - **Fault Modelling: *Fault Modelling Architectural Framework (FMAF)***
  - CPS Technologies
- Conclusions
- Wrap Up

Fault Modelling

- Use the *Fault Modelling Architectural Framework (FMAF)*
- Prompts SoS engineer to consider impact of faults at early design stages
- Views & concepts for designing fault-tolerant SoSs
  - Structure: faults and failure modes; fault tolerance structures; recovery procedures
  - Behaviour: fault activation; erroneous behaviour; recovery strategies





Dependability Properties

Property	Comments
Availability	We can investigate causes and consequences of functional interfaces not being available
Reliability	We can investigate causes and consequences of lost messages
Safety	We can compare the effects of actions in safe and unsafe states
Continuous properties (e.g. power)	We can model power fluctuations as discrete state changes, and investigate consequences of these fluctuations

Overview

- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
  - SoS Technologies
  - **CPS Technologies**
    - **Co-modelling and Co-simulation: Crescendo**
- Conclusions
- Wrap Up

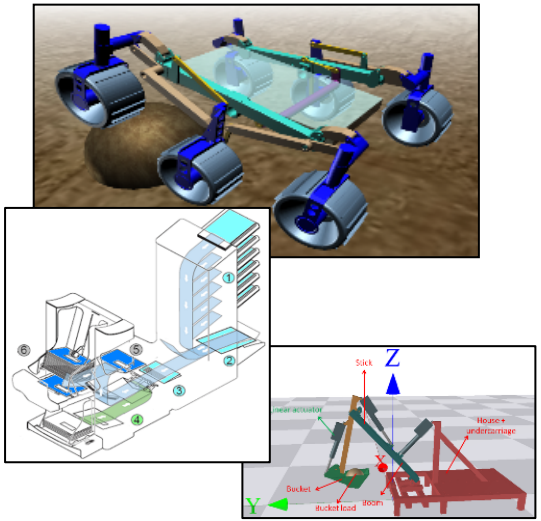
Co-modelling Software and Physics

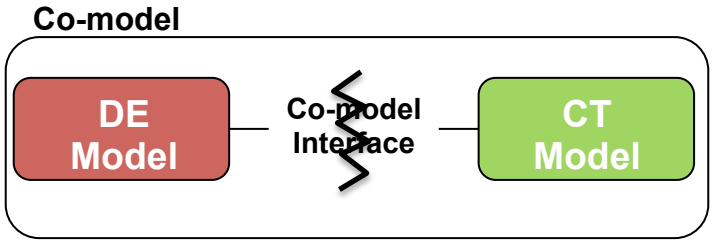
- **Discrete-event (DE)**, e.g. VDM-RT

- In simulation, only represent points in time at which the state changes
- Good abstractions for software
- Less suited for physical system modelling

- **Continuous-time (CT)**, e.g. differential equations

- In simulation, the state changes continuously through time
- Good abstractions for physical system disciplines
- Poor software modelling support





Co-modelling the LOSA SoS

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

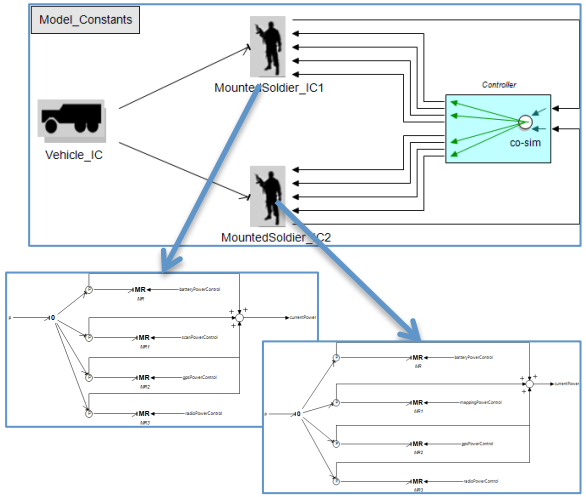
DE Model in VDM-RT

```
class MountedSoldier


types
    public SoldierType = <Scan>|<Mapping>

instance variables
    public battery: [BatteryChargeController] := nil;
    public scanFunction: [ScanFunctionController] := nil;
    public mappingFunction: [MapFunctionController] := nil;
    public gpsFunction: [GPSUnitController] := nil;
    public radioFunction: [RadioUnitController] := nil;
    public msType : SoldierType;
    public totalPower : real := 0.0;


operations

    public MountedSoldier: SoldierType ==> MountedSoldier
    MountedSoldier(tp) ==
    (
        msType := tp;
        battery := new BatteryChargeController();
        gpsFunction := new GPSUnitController();
        radioFunction := new RadioUnitController();
        cases msType:
         <Scan>    -> scanFunction := new ScanFunctionController(),
         <Mapping> -> mappingFunction := new MapFunctionController()
        end;
    );


    public updatePowerUsage : () ==> ()
    updatePowerUsage() == let now = time/1e9 in
    (
        cases msType:
         <Scan>    -> scanSoldierScenario(now),
         <Mapping> -> mapSoldierScenario(now)
        end;
    );


    --Scenario for Soldier_IC
    --
    private scanSoldierScenario: real ==> ()
    scanSoldierScenario(t) ==
    (
        if(t > 0 and t <= 1) then
        (
```

Co-model Interface

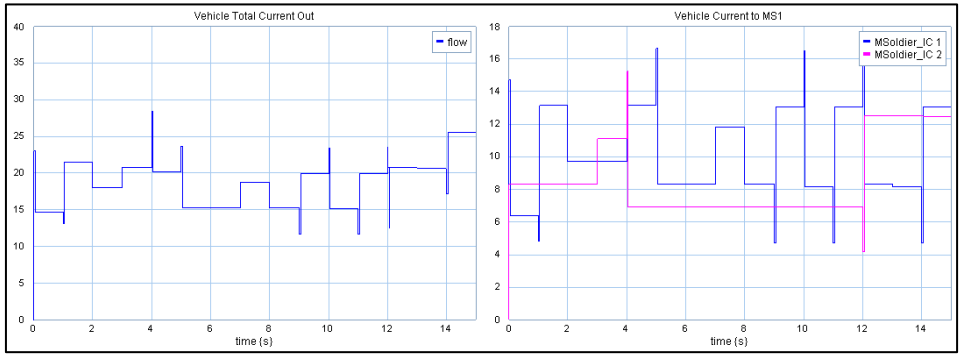
```
-- Monitored variables
monitored real ms1CurrentPower;
monitored real ms2CurrentPower;

-- Controlled variables
controlled real ms1BatteryExpectedPower;
controlled real ms1ScanExpectedPower;
controlled real ms1GPSExpectedPower;
controlled real ms1RadioExpectedPower;

controlled real ms2BatteryExpectedPower;
controlled real ms2MappingExpectedPower;
controlled real ms2GPSExpectedPower;
controlled real ms2RadioExpectedPower;
```

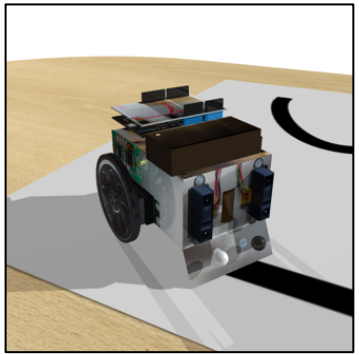
CT Model in 20-sim

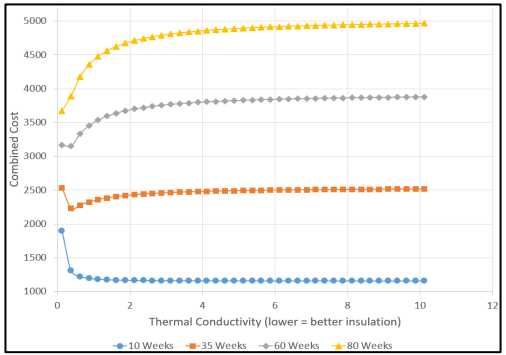


Co-simulation

- Co-simulation results
  - Can analyse the flow of current from the vehicle depending upon the behaviour of soldiers
  - Use as basis for decision making







John Fitzgerald J., Gamble C., Payne R., Larsen P.G., Basagiannis S., Mady A.E. "**Collaborative Model-based Systems Engineering for Cyber-Physical Systems, with a Building Automation Case Study**" In INCOSE International Symposium (IS 2016)

Dependability Properties

Property	Comments
Continuous properties	We can model power as a continuously changing variable, and observe fluctuations over time
Discrete-Continuous Interaction	We can explore the interaction and dependencies between discrete and continuous aspects of combined models

Overview

- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
- **Conclusions**
- Wrap Up

Conclusions and Assessment

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Project purpose was "… to [establish the feasibility of] a pragmatic method of enabling the assessment of security, safety and reliability dependencies within a given system of systems within the LOSA context."

Assessment: a pragmatic method of assessing security, safety and reliability dependences of an SoS in the LOSA context ***is*** feasible, *although the different elements are at different levels of maturity*.

Conclusions and Assessment 2

- **Interface Contracts**
  - Useful for model consistency checking and as a communication tool
- **Formal modelling**
  - Provide the most confidence of satisfaction of properties, but integration with some established modelling techniques (e.g., SysML) is vital
- **Fault Modelling**
  - May have value in identifying and managing causal chains leading to potential system and SoS failures
- **Co-modelling**
  - Potential to aid analysis and assessment of cross-domain dependability properties (integration of continuous domains).

Overview

- Dependability in Systems of Systems
- LOSA and the Study
- Model-based SE Techniques
- Conclusions
- **Wrap Up**

Recommendations for Further Work

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

1. Compare methods with potential alternatives along cost, cost-effectiveness and usability dimensions
2. Assess potential to integrate with relevant engineering processes and to input to future standards
3. SoS and CPS requirements modelling
   - including requirements that span DE and CT models
4. Place results obtained from analyses (formal, semi-formal and co-simulation) w.r.t. safety cases

Further Information

- Bryans J, Fitzgerald J, Payne R, Winthorpe E. **Applying Model-based SE Techniques for Dependable Land Systems.** In INCOSE International Symposium (IS 2016)
- John Fitzgerald J., Gamble C., Payne R., Larsen P.G., Basagiannis S., Mady A.E. "**Collaborative Model-based Systems Engineering for Cyber-Physical Systems, with a Building Automation Case Study"** In INCOSE International Symposium (IS 2016)

http://www.into-cps.au.dk
http://thecompassclub.org

richard.payne@newcastle.ac.uk ✉
@riffio, @Ncl_CPLab