



UNIVERSITY OF
Southampton

Experience of Applying Rodin in an Industrial Environment

James Sharp, Neil Evans and Helen Marshall

Context

- AWE has been using Formal Methods (in various forms) for over a decade.
- Our application of formal methods encompasses:
 - analysis of existing electrical/software systems,
 - analysis of Safety Themes,
 - and most recently, in applying mathematical rigour to the design of electrical systems.

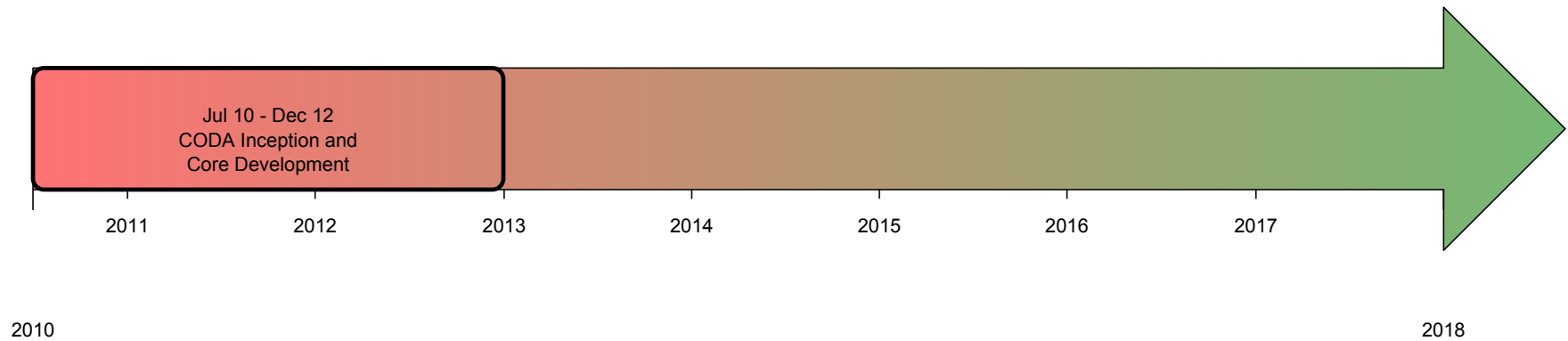
Presentation Aims

- Introduce the Co-Design Architecture (CODA).
- Highlight how Rodin is currently being used within AWE.
- Suggest how to introduce Formal Methods into the systems development process.

What is CODA?

- CODA provides a graphical interface and methodology to develop, analyse, and formally verify the interactions between, and the behaviour of, the components of systems comprising both software and digital electronic hardware.
- CODA guides the designer to embrace modelling the entire system:
 - This includes modelling the interactions with its environment.
- CODA constrains and specialises Event-B, tailoring it for our specific applications.
 - Refinement in CODA can be targeted at individual components in a clean, visual manner.

CODA Timeline: Past



CODA IDE v1

Event-B – platform:/resource/SL3/CP.cpd#_jFR9QDpEEeG8jejB4yGA_w – Rodin Platform – /Volumes/Data/CODAdemoWorkspace

Tahoma 9

Event-B Explorer

- aCC
- aCM
- aCM2
- NewProject
- SL3
 - ComponentCtx
 - X1
 - X2
 - m0
 - m1
 - m2
 - Variables
 - Component CP
 - Statemachine SM
 - Invariants
 - Events
 - Proof Obligations

CP.cpd#1 m2

Environment

- PowerUp
- Reset

Controller

- IO
- sendunlock
- RecvPowerUp
- RecvReset
- StartIO
- SetA
- SetB
- ResetB
- ResetA

chan DATA

StrongLink

- Unlock
- RcvBit

Palette

- Component
- Asynchrono...
- Synchronous
- Port Wake
- Self Wake
- Method
- Initiator
- Connector
- Sender
- Receiver

Properties Tasks Search Error Log

P PortWake

Label	Connector	Value	Comment
recvUnlock	chan	unlock	

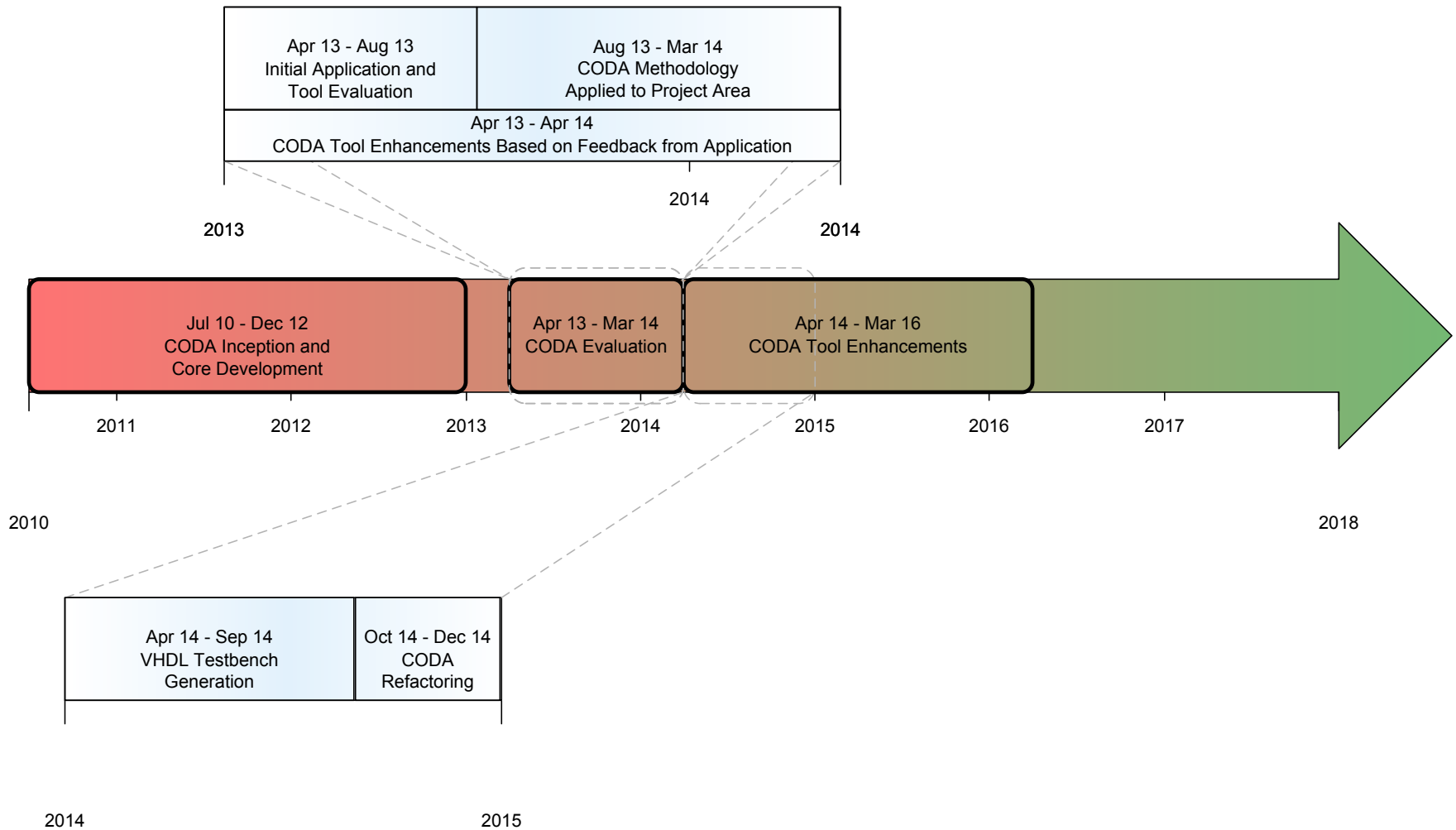
Overview
Core
Parameters
Port-Wakes
Port-Sends
Method-Calls
Wake-Events

Add Receive Delete Receive Move Up Move Down

Utilising the Rodin IDE

- Extensive use is made of the underlying Rodin engine and other Rodin plug-ins.
 - iUML-B State machines are used to model behaviour of individual CODA components.
 - ProB is used to model check and animate translated Event-B (from iUML-B and CODA graphical models).
 - The Proof Obligation Generator, SMT and Atelier-B provers are used to prove correctness of the translated Event-B.
- Additional analysis at the CODA-level has been introduced:
 - A CODA validation check prior to translation.
 - The CODA Simulator for CODA-level animation.

CODA Timeline: Past and Present



Recent Application at AWE

- Case Study looking at a complex ‘slice’ of a system’s functional behaviour.
 - 9 month period, with 20 on-site contractor days.
- Applying Event-B refinement within the CODA methodology
 - Looked at correctly capturing the top level requirements: what is the aim of the system?
 - Forced resolution of ambiguities in the informal system definition
 - Highlighted a disconnect between the requirements levels
 - Ensured problem was completely understood

Key Findings of the Methodology

- Refinement chains are not a one *'shot approach'*.
 - Iteration through the refinement chain, sometimes requiring significant refactoring is necessary.
 - Recent application required 6 iterations through various levels of refinement.
- SMT solvers are key:
 - An increase of 31% automatically discharged proofs
 - 98% (with SMT) vs 67% (without SMT) of a total 1173 proofs.
- Analysis performed using Event-B and the CODA methodology is bearing fruit:
 - Prominent in illustrating (lack of) understanding of the problem.
 - Can be used to provide confidence to the customer through animation in ProB.

Verification and Validation

- How does Rodin enhance V&V?
- Verification (safety) of the design:
 - Formal proof through the use of Invariants.
 - LTL statements asserted by ProB over the model.
- Validation of the design:
 - Animation in ProB can and has been used to walk through scenarios at different refinement levels during the development process

Further V&V: Responsiveness

- We define the responsiveness of a system as “*a system’s ability to always respond in the correct and expected way to external stimuli*”.
- First attempt used a composite approach from a common specification:
 - Manually identify valid paths through the specification and define invalid path rules.
 - Automated generation (using MALPAS) of **all** paths through the specification.
 - Number of paths then reduced by applying the invalid path rules.
 - Iterative comparison of the results of these two approaches produce a common, complete, set of valid paths of the system.

CODA IDE v4

The screenshot displays the CODA IDE v4 interface with a component diagram for 'LowerLevelComponent'. The diagram is structured as follows:

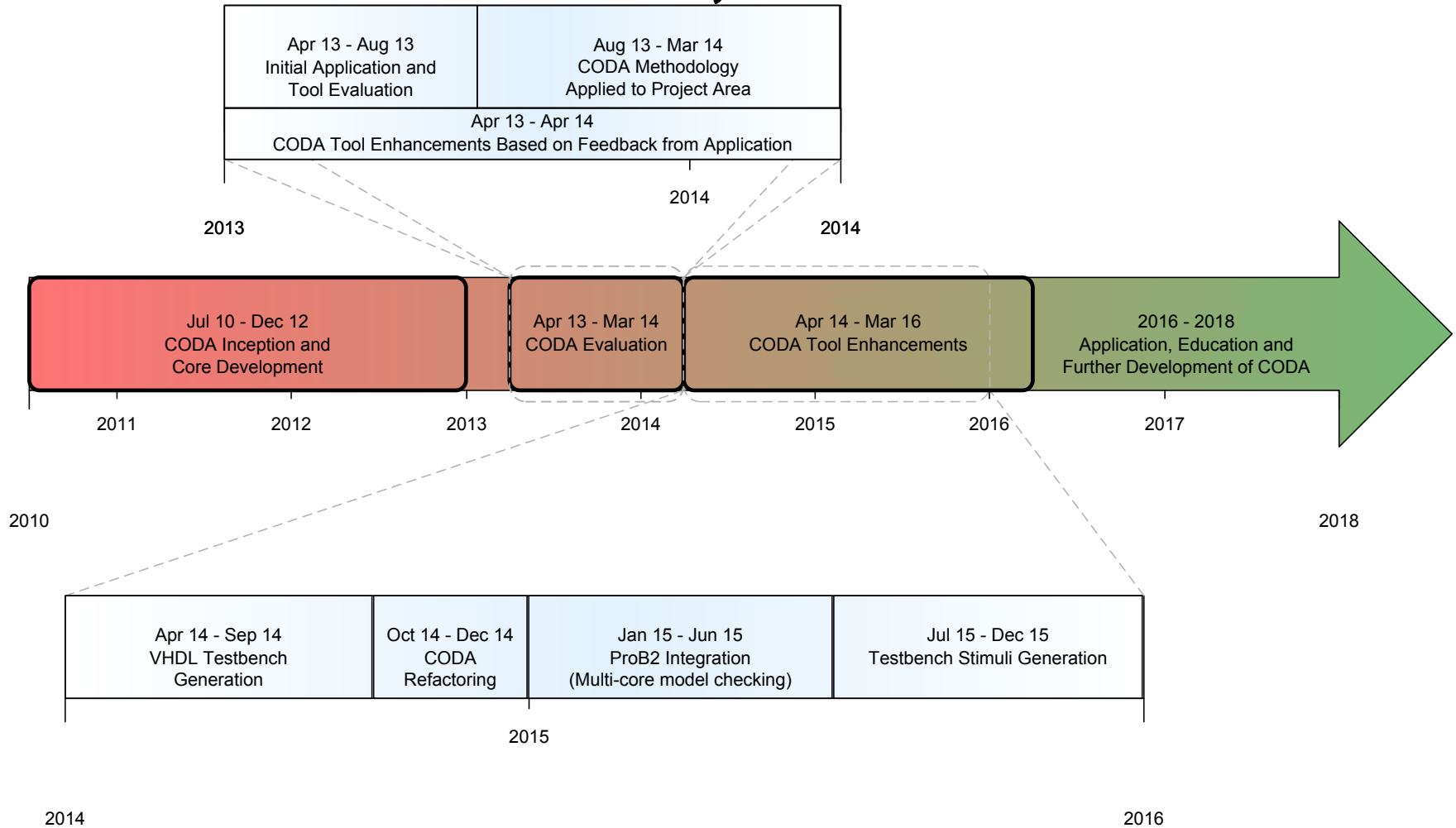
- ENVIRONMENT** (outermost container):
 - Operations: `apply_alt_energy`, `apply_intended_energy`
 - EnergyTypes: `alternate_env_path`, `intended_env_path`
- EXCLUSION_REGION** (inner container):
 - Operation: `apply_er_energy`
 - EnergyTypes: `ER_energy`
- PORTAL** (component):
 - Operations: `Portal_SM`, `block`, `allow`
 - EnergyTypes: `portal_energy`
- TRANSFORMER** (component):
 - Operation: `step_up`
 - EnergyTypes: `TF_to_OBJ`
- OBJECT** (component):
 - Operations: `absorb`, `engage`, `release`, `break`
 - EnergyTypes: `Object_Behaviour`

Relationships and connections in the diagram include:

- `ENVIRONMENT` contains `EXCLUSION_REGION`.
- `ENVIRONMENT` contains `PORTAL`.
- `ENVIRONMENT` contains `TRANSFORMER`.
- `ENVIRONMENT` contains `OBJECT`.
- `EXCLUSION_REGION` contains `TRANSFORMER`.
- `EXCLUSION_REGION` contains `OBJECT`.
- `PORTAL` is connected to `TRANSFORMER` via `portal_energy`.
- `TRANSFORMER` is connected to `OBJECT` via `TF_to_OBJ`.
- `TRANSFORMER` is connected to `OBJECT` via `ER_energy`.

The interface also shows a left-hand 'Event-B Explorer' tree, a 'Rodin Problems' panel at the bottom, and a 'Symbols' keyboard layout on the right.

CODA Timeline: Past, Present and Future



Concluding Remarks

- Enhancement of current engineering practices by adding mathematical rigour.
- Introduction via the V&V route met with less resistance.
 - Not too intrusive
 - But able to illustrate real benefit to the systems development process in an incremental manner
- Scale-up activities over time as they becomes more accepted.
- Continued application is key to reaffirming the benefits.