

A Unified Test Assertion Model?

Comparisons with UML

DRAFT 0-1

Stephen Green
SystML
www.systml.co.uk

Parallels with UML

⇒ Predicative

- Artefacts
- Properties

⇒ Event Behaviour

⇒ Modules

- Profiles
- Pre-conditions

⇒ Prose sections

⇒ Class Diagram

- Classes
- Properties

⇒ Activity Diagrams

⇒ Packages

- Name-spaces
- Notes

⇒ Prose sections

Predicative

⇒ Artefacts

- How similar is an artefact to a class?
 - They both can consist of further artefacts/classes via associations, containment, etc.
 - They both are expected to exist in the implementation without necessarily doing something
 - Artefact could be a class or object when implemented

⇒ Properties

- An artefact can be required to possess properties and these could be the properties of a class

Event Behaviour

- ➔ Maybe an predicate is a special form of an event-behaviour
- ➔ Or the event-behaviour could include a predicate just as an activity diagram can contain one or more classes or sets of classes
- ➔ Events and behaviours can reference and describe artefacts and their properties

Contrasts

- ⇒ The parallels are there but UML relates to OO whereas the TA might be a lot broader than that
- ⇒ The final use of the TA might be different to that of a UML diagram
 - If a TA expressed as many points in one statement as a UML diagram contains classes, actors or actions then it might be less useful
 - References to and from each point in TA
 - A test may need to point to a single TA
- ⇒ Groups of TAs under common pre-conditions
 - Groupings at various levels may be needed

Prose

Both UML and TA models have in common the need to be supplemented with prose expressions

Lessons from UML Comparison

- ⇒ Simplicity
- ⇒ Extensibility
- ⇒ Grey areas
- ⇒ Allow prose
- ⇒ Linked

Simplicity

- ➔ a suspicion that test assertions need to be far simpler than UML in order to be usable

Extensibility

- ➔ a suspicion that there are likely to be design features in specifications which require more than just the two types of model

Grey Areas / Overlaps

- ⇒ One model type can include another
 - Example: An event may require that there exist an artefact with certain properties
 - Example: A property might change during an event in a certain way
- ⇒ One type might make reference to elements of another
 - Example: An event might involve a change to a property of TA expressed as a predicative
- ⇒ A TA of one type might share in common pre-conditions with a TA of another type

Prose may suffice for some TA requirements

⇒ Prose

- Clear
- Succinct
- May be better when lots of artefacts have to be linked
- Sometimes require very little explanation in prose format even though complex

⇒ Models

- Sometimes cloud the meaning
- Verbose at times
- Structure of models sometimes inadequate or inappropriate
- Require expertise not always available

Links between TA and UML

Maybe a TA will sometimes
include UML elements

But: there needs to be
provision equally for other design disciplines for
which UML may not itself have to cater