How do we know that a scenario is appropriate?

Ian Whitworth

Steve Smith, Geoff Hone, Iain Macleod

All of the:

Department of Information Systems
Defence College of Management and Technology
Cranfield University at the
Defence Academy of the United Kingdom

Shrivenham SN6 8LA UK

i.r.whitworth; s.j.smith; g.n.hone; i.macleod@ cranfield.ac.uk

Is That Scenario Appropriate?

First, the definitions:

Appropriateness suggests “fitness for purpose”

but before that comes:

Just what is a scenario?

Scenario Definitions:

Generic but incomplete:
An outline or model of an expected or supposed sequence of events.

Taken from www.dictionary.com

Domain Specific:
An initial set of conditions and time line of significant events imposed on trainees or systems to achieve exercise objectives.

DMSO – 2006

Proposed:
A representation of the state, and present actions, of a set of animate and/or inanimate objects, so as to permit the exploration of, or reasoning about, their future state and the events that lead to it.

Why have Scenarios?

MILITARY

- Training
- Tactical Development
- Mission Rehearsal
- War-games
- Geo-Political Games

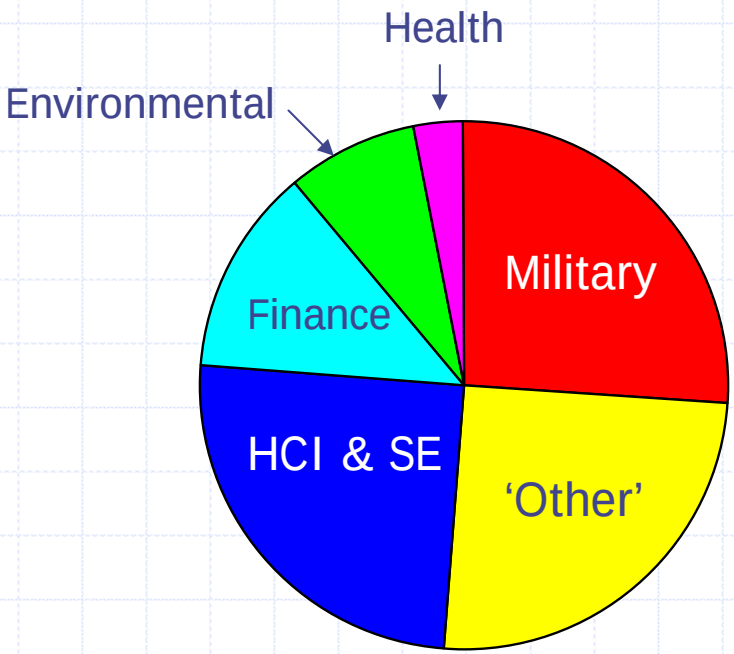
CIVILIAN

- Training
- Disaster Preparedness
- Corporate Planning
- Countering Epidemics

Some of these need 'one-off' or custom-written scenarios but other applications need the facility for scenario re-use

Current work on Scenarios

Two literature searches and one bibliography search:



Everything was domain specific!
Nothing was concerned with the generic basis of scenarios!

A Systems Approach to Scenarios

Take a small set of scenarios ...

Analyse their structure, categorise their components ...

Assess each component for individual items ...

Tabulate the results ...

Test this against another small set of scenarios ...

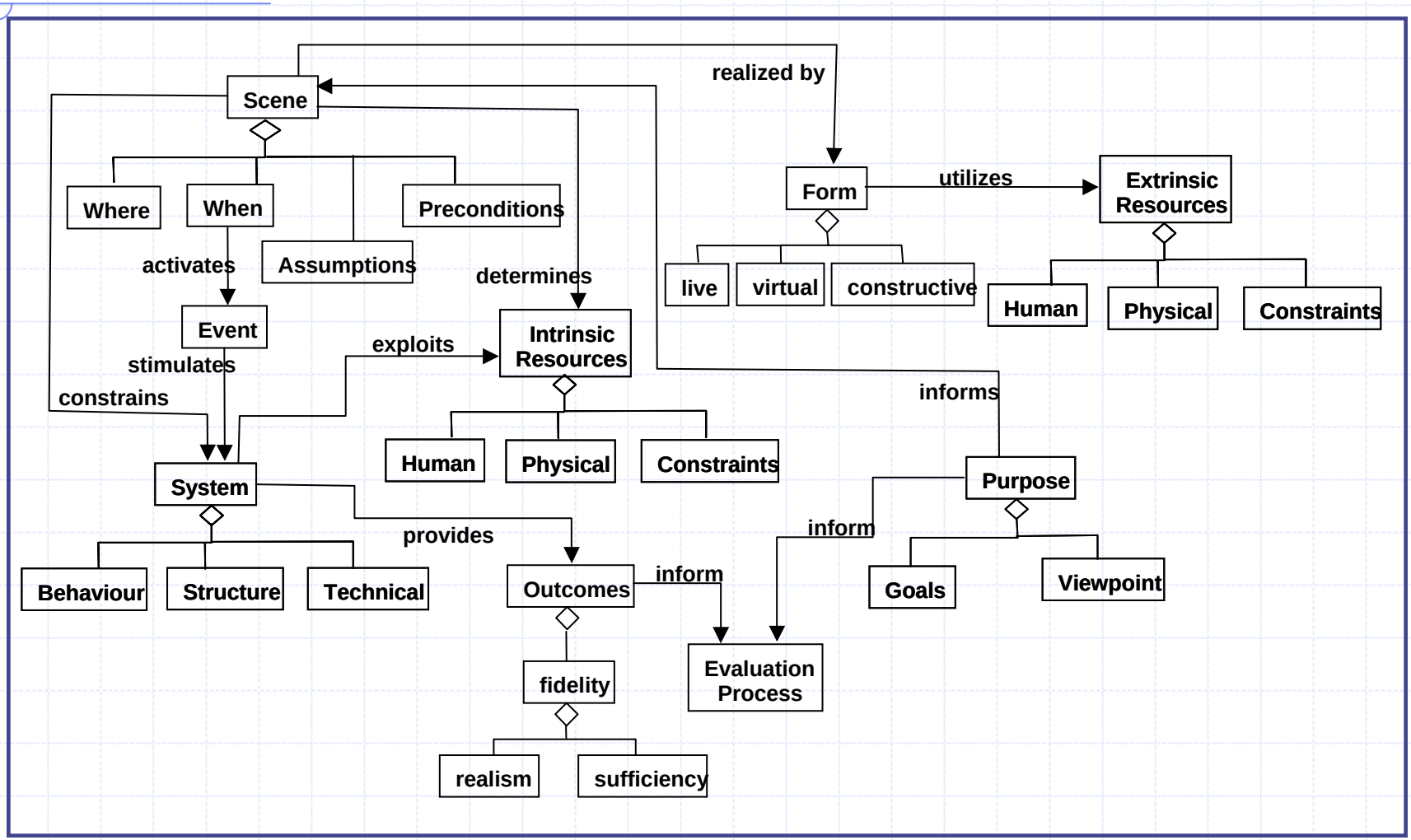
Prototype a scenario architecture ...

Produce definitions for components and items ...

AND FINALLY:

Test the architecture against two examples of good scenarios

Scenarios: the proposed architecture



Scenarios: test procedure ...

When we had produced definitions for components and items, it was possible to build a checklist.

This defined each component and the items making it up

For example:

Purpose: the reason the scenario is required.

Goal	what is intended to be achieved.
Viewpoint	the position that values the goal

Each definition was followed by a box that could hold a description of how that component or item was identified, and a second box to hold a reference.

Each paragraph in the scenario could be numbered. We could now tabulate each identification and an appropriate reference.

Scenarios: the big test

The two scenarios chosen to test the prototype architecture were:

"IVY LEAGUE"
A classic strategic-level war game from the era of President Reagan. This was based on a set of global events that escalated into war, culminating with a nuclear strike on Washington DC.
(It is covered in some detail in TB Allen's book "War Games" , 1987)

"FOMBLERS FORD"
This is the DARPA rewrite of EW Swinton's classic "Duffers Drift", taking Boer War conflict into the era of NEC, and similar technological advances (set in the Balkans of the 1990s)

Scenarios: What did we find?

The results confirmed that our architecture was valid.

In each scenario, we could:

- identify those major components present

-identify the items present for each component

And more importantly, we were able to establish that the two scenarios used two slightly different sets of components, but both had a specified purpose.

We later found that several 'alleged' scenarios were only text or spreadsheet descriptions of an event; fitting the DMSO definition but lacking any identifiable purpose.

Mapping Scenario to Architecture

Most scenarios are constructed in a spreadsheet format …

Timelines and events are important …

We followed this approach

We took a major disaster (Buncefield Oil Terminal Fire)

1. A scenario was crafted in the normal (chronological) form

This met very little of the architecture

2. A textual version was prepared

This had no stated purpose

3. A (short) requirements document was written

This could be related directly to the architecture
A link to 2. & thence to 1. was established

Three propositions

1. Any scenario must have a declared purpose
2. Any scenario must contain all those components necessary to meet the declared purpose
3. These do not have to be in one document

   (provided that there is full traceability)

And, yes, this does mean that a spreadsheet chronology may only represent just a few components in our architecture.

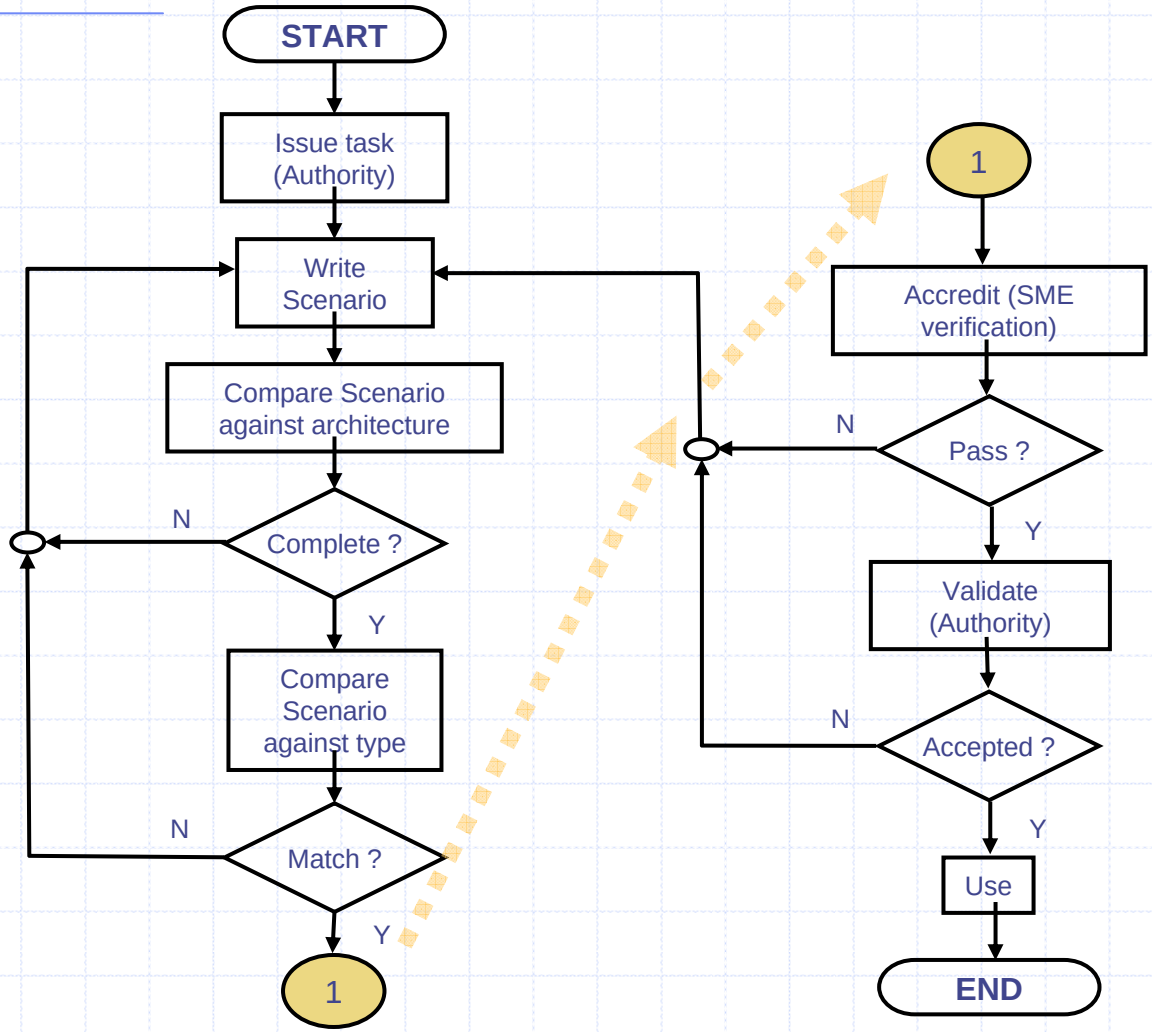
Using Scenarios

Purpose indicates a requirement.
Any scenario must have been:
- required to meet a need
- authorised for generation
- written
- assessed for suitability
- accepted

This means some measure of VV&A

To facilitate this we propose a generic process for scenario employment. This works equally well with new requirements and for the reuse of existing scenarios

Scenario Writing Process



The Writing Process in Action

As the process flowchart shows, the full VV&A is implemented:

- Completeness check (Scenario against Architecture)
- Check for correct type
- SME Accreditation Check
- Validity check (are concepts still appropriate)

And at each stage, a negative returns the scenario for re-writing.

An existing scenario can be processed (insert at the 'write' stage)

In Conclusion ...

We have:

- **Prototyped a Scenario Architecture**
- **Tested the Architecture**
- **Prototyped a Scenario Assessment Tool (SAT)**
- **Produced a draft Scenario Generation process, enabling**
- **Scenario VV&A and Scenario Reuse to be formalised**

The next stage of the work will be the development of the SAT in both hard-copy and computer-based form

The work of which this product is part was initiated by the Research Director, CBD/Human Sciences. This support is gratefully acknowledged