Validating a Scenario Assessment Tool

Geoff Hone
Ian Whitworth
Andy Farmilo

Department of Informatics and Sensors
Cranfield University at the Defence Academy of the United Kingdom
ghone.cu@defenceacademy.mod.uk   i.r.whitworth@cranfield.ac.uk   a.farmilo@cranfield.ac.uk

David Swift

Directorate of Training (Army)
Trenchard Lines, Upavon, Wiltshire
prinpsy@ditraining.mod.uk

Keywords: VV&A Scenarios MSDL Consistency Checks

Abstract

At the 11th ICCRTS we reported on work directed at producing a generic scenario architecture for a broad range of simulation activity. Since that time, the architecture has been revised, and offered as an intermediate step between requirement and detailed scenario generation. This is seen as part of the whole validation and verification process which is considered essential for modern simulation. Based on this architectural framework, tools have been developed to enable the rapid assessment of scenarios at both high level (a spreadsheet based tool) and at a more detailed level. These will enable scenarios to be assessed for validity as fast as an agile force can call for them. However, it is essential that the tools themselves be validated, and we can report on a validation of the developed Scenario Assessment Tool against a number of developed scenarios of acknowledged military relevance. This will facilitate rapid scenario development, and assessment of existing scenarios for re-use for both training / mission rehearsal, and for experimentation. A web-based tool for the fast, secure assessment of scenarios written using the Military Scenario Description Language (MSDL) is currently in development.

1. Introduction

Scenarios (whatever they might be, since the noun has multiple definitions) are the basis for a wide range of military-related activities such as training, planning, war-gaming, procurement, and are widely used by civil administrations and for commercial activities. If, however, a scenario is to be of value for any given application, it must be "appropriate", or "fit for purpose". Our concentration was on the identification of the components of an appropriate scenario, any relationships that might exist between them, and on the individual items forming part of each component. From this, we were able to propose a provisional scenario architecture – effectively the base-line for any scenario assessment – and which we considered as the basis for a formal generic scenario architecture.

We have already mentioned the uncertainty surrounding the term "scenario", and in the course of several literature searches were unable to identify a single definition that was both generic, and of universal applicability. Of the many to be found, the nearest definition to our target was that offered by NATO (2002):

> *A description of the environment, means, objectives, and events related to a conflict or a crisis during a specified time frame suited for satisfactory study objectives and the problem analysis directives.*
>
> *NATO 2002*

This is specifically directed at C2 Assessment, and would seem to be both too specific in direction, and at the same time lacking in precision for our purposes. Accordingly, we proposed the following definition (Whitworth, Hone, and Farmilo, 2007) as being more generic in nature.

> ***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.***

It will be seen that our proposal that it is the projection of current states into the future that is at the heart of all scenarios, as can be seen in the use of scenarios for training, mission rehearsal (as a particular form of training), and for war-gaming. Even scenarios for routine training will often include one or more points at which the trainees can make critical errors, as a means of enhancing the learning experience.

It is generally held that war-games, as we currently know them, were started at the US Navy Academy following the War of 1812. The Prussian General Staff referred to the concept as "Kreigsspiel", and the concept was taken up by many General Staffs during the 20th Century. During the Cold War era, there was much attention to the geo-political games, as used by all major world powers to explore the effect of actual and potential policy changes and events; Mandel (1977) holding that this was the case as early as 1961. Throughout all of these cases, we argue that there has been a scenario (as in our definition above) that forms the base from which the game has been developed, and the basis from which the game was played. In almost every case, gaming and simulation requires a scenario as a start point. Indeed, it can be argued that the initial set-up for a sand-table exercise in the 1950s and '60s was a scenario.

Some scenarios are intended to be re-used. A scenario intended for (say) training Mounted/Armoured Infantry in Platoon tactics will be crafted to meet a specific set of training objectives, and will then be re-used for every successive Platoon until the training objectives change. Changes to training objectives are usually co-incident with changes to doctrine and/or equipment. Alternatively, a scenario written to train a force for a specific mission will be used repeatedly until all elements of the force are clear on roles, procedures, timings etc.

Development of the principles of Network-Centric Operations (NCO) and Network-Enabled Capability (NEC) has been carried out, to a large extent, through scenario-based simulation and experimentation. Alberts *et al* (2002) have emphasized the need for 'Campaigns of Experimentation' as a way forward, and this approach can often require multiple consistent scenarios, and scenario re-use. Hence, the creation of a framework for scenario development will help to ensure that scenarios are created in a coherent manner.

Not all scenarios are intended for multiple or sequential use, some scenarios are intended to be used once only. A disaster scenario (as in one designed to test the integration and response of the emergency services) in a single bounded area or locality is unlikely to be repeated. Lessons learned from one single use will be applied to each of the services, and a fresh scenario will be subsequently employed to see if any identified problems have been resolved.

In our earlier work, the "goodness" of a scenario was used to refer to the suitability of a scenario for its intended use, combined with the suitability of the intended use – thus encompassing the range of Validation, Verification and Accreditation (VV&A) functions. Thus, a scenario written to exercise NATO Staff in their response to an invasion of Western Europe, by Eastern Bloc forces, through the Fulda Gap will remain suitable (good) at the Staff level, but will not be appropriate (lacking "goodness") with regard to the current European and NATO geo-political structure.

Current work on scenario architectures:

We have previously reported (Whitworth et al, 2007; Hone, Whitworth and Farmilo, 2008) the detailed results of two internet-based literature searches into the extent and nature of current research activity in respect of scenarios. They can be summarised here as falling into the following broad areas:

Health, Environment, Military, Synthetic Environments, Finance, Other

with **Military** (103) and **Other** (100) accounting for about half of the hits. It was observed that many of the **Other** hits were related to hobby-type war-gaming.

The "Military" category was then investigated further by searching the US DoD "Stinet" website (considered by the DoD to be the major repository of military related documents). The small number of documents found did cause some surprise, and each "hit" was then individually reviewed for military relevance. A high proportion of the documents were (in part or in whole) from PhD Theses on programming or software design. It seems clear from this~~,~~ that published activity in the specific area of scenario architectures is minimal. A separate search, targeted on the production of a general-purpose bibliography, produced material that was over 90% related to Software Engineering, or Systems Engineering.

2. Systems Approaches

2.1 Scenarios and the UK Defence Systems Approach to Training (DSAT).

Scenarios, construed in the broad terms above, play a key role in articulating both UK *operational* requirements and the *training* requirements designed to enable them. The former begin with a small set of overarching military tasks comprising military (Defence) capability; these tasks are given further elaboration by military policy and doctrine, including Defence Strategic Guidance. Defence Strategic Guidance defines an environment in which non-state actors, increasing domestic and international scrutiny and accountability, integration of military into coherent cross-governmental response, terrorism and both high and low technology asymmetry are prevalent. For present purposes, there are three key underpinning concepts informing the development of scenarios.

- Firstly, the distinction between '*a*' war (generic war-fighting) versus '*the*' war (mission and theatre specific).
- Secondly, *Fighting Power*, comprising three components (see Figure 1 below):
- Thirdly, the Contemporary Operating Environment (COE)

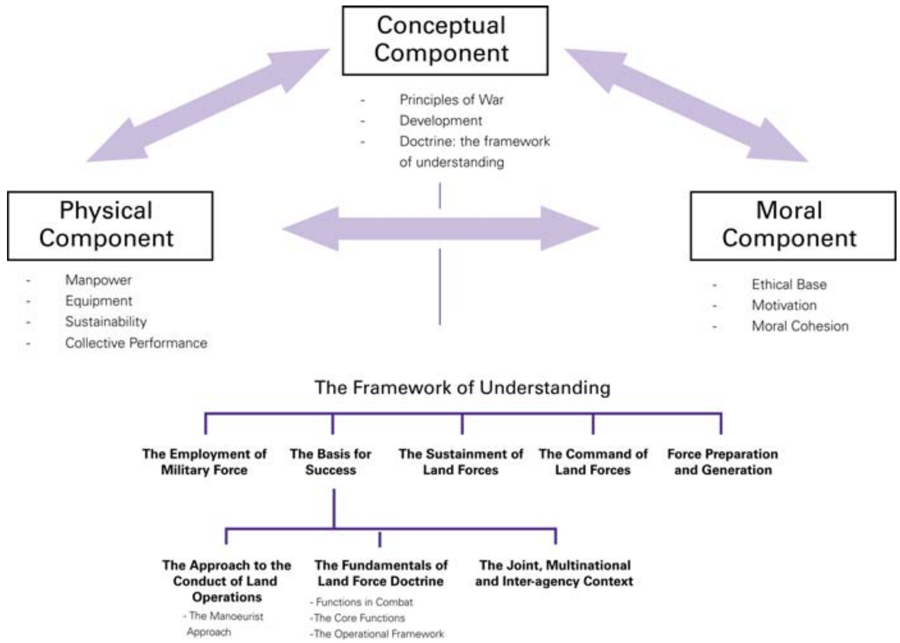

Figure 1. Fighting Power.

Fighting Power primarily informs the articulation of operational and training requirements for **'a'** war.

The COE identifies 6 dimensions:

Physical.	Human.
Political.	Military – The Threat
Economic.	Information.

The COE primarily informs the articulation of operational and training requirements for **'the'** war.

The UK Land Warfare Centre (LWC), which deals primarily with *Collective* levels of training, has developed a general scenario, GENSCEN, based upon the COE. While *Individual* levels of training similarly make use of scenarios, they are currently subject to the requirements and techniques of the Defence Systems Approach to Training (DSAT) to a greater degree than collective training.

The UK DSAT originated in the mid 1960s by the Army's Royal Electrical & Mechanical Engineers (REME), was rapidly adopted tri-Service in different variants and unified during the current decade. DSAT is a training quality cycle which is both systemic and systematic and has the following cardinal features (see Figure 2, below):

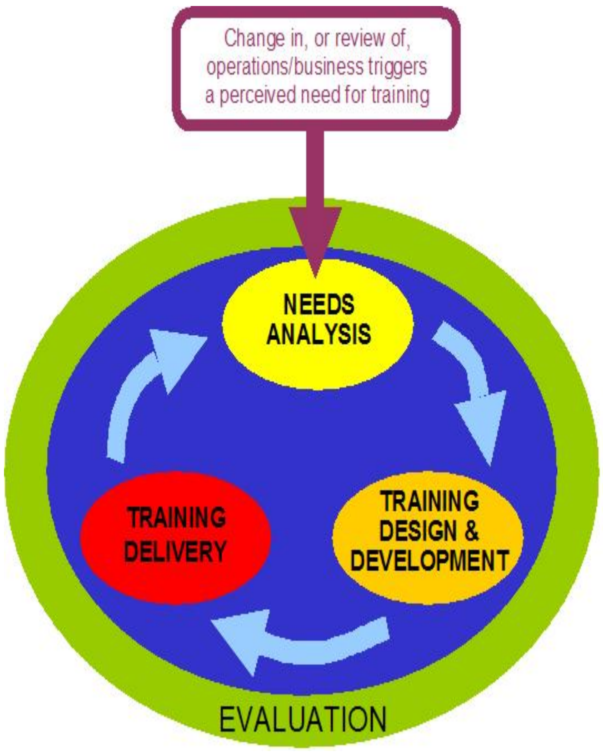

Figure 2: UK Systems Approach to Training (DSAT)

Effective though DSAT is at identifying and resolving training needs as they arise in many contexts, it assumes essentially a steady state resource allocation (costs adjusted only for inflation). By the late 1980s it was recognised that a "handshake" between the training quality cycle and the procurement cycle was needed in order to ensure that training equipments, such as simulators, were properly specified – in all regards - for training purposes. This was achieved by means of Training Needs Analysis (TNA) methodology, whose main features are summarised at Figure 3, below.

TNA methodology followed a similar history to DSAT, having been unified tri-Service in 2001, and may be considered to be a "virtual" application of DSAT. At the nub of the problem of training equipment specification prior to TNA was a failure to inject sufficient rigour into the requirement: training simulators tended to be procured against "scenarios", then understood to be brief, descriptive, account of different roles and missions of the weapon platform. Contractors could - and did – take advantage of the looseness of such requirement stated in this way. Following a National Audit Office report and an enquiry, TNA was mandated as the solution. Part of this problem was solved by formalising a requirement for Fidelity Analysis within TNA, but even more fundamental was the insisting that the training requirement had *military training objectives*, specified as per DSAT, at its core.

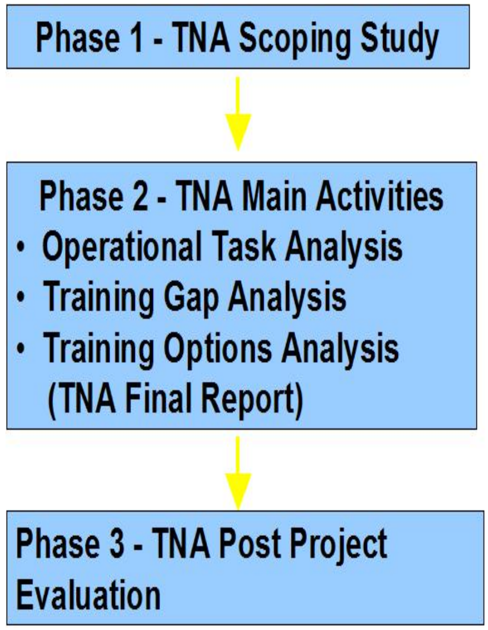

Figure 3: TNA Methodology

TNA methodology followed a similar history to DSAT, having been unified tri-Service in 2001, and may be considered to be a "virtual" application of DSAT. At the nub of the problem of training equipment specification prior to TNA was a failure to inject sufficient rigour into the requirement: training simulators tended to be procured against "scenarios", then understood to be brief, descriptive, account of different roles and missions of the weapon platform. Contractors could - and did – take advantage of the looseness of such requirement stated in this way. Following a National Audit Office report and an enquiry, TNA was mandated as the solution. Part of this problem was solved by formalising a requirement for Fidelity Analysis within TNA, but even more fundamental was the insisting that the training requirement had *military training objectives*, specified as per DSAT, at its core.

Military training objectives are formally stated in terms of *Performance*, *Conditions* and *Standards*, but it should be noted that a set of military training objectives do not comprise, nor are they a substitute for, a scenario. In DSAT/TNA terms, a scenario may be said to provide a plausible ("realistic") *context* for training objectives – it is an elaboration of the *Conditions* element of a military training objective. Whilst the requirements for individually trained skills will always be with us, the changing nature of operations, and the facilities offered by modern simulators, now place a much greater emphasis on collective training. The orthodox DSAT is very difficult to apply at the collective level, although a Training Matrix can be applied to networked simulators (see for example Hone, 1995), a more rigorous understanding of 'scenario' will help to enhance the value of any training given. One view of scenarios, exemplified by the MSDL, is that they provide a finite, but not necessarily bounded, set of initial conditions within which the feasibility and effects of military capability may be explored at different levels.

2.1 A Systems Approach to Scenarios

As an approach to identification of those items that could be said to constitute a scenario, a number of existing scenarios were collected, and were then analysed to determine their individual components. From this, the provisional architecture was developed and refined. It was then possible to identify the relationships between components, the dependencies between components, and the items which could be part of some of the components.– This approach then enabled the provisional architecture to be cast as a checklist of components and items.

Given this prototype checklist, it was possible to review a sample set of scenarios, and to determine which of the components a given scenario actually contained. Using this approach, it was found that (for example) an apparently detailed and complete scenario contained only one component, but omitted all others that had been identified as potentially necessary. This can be seen as starting to validate the architecture rather than a scenario. The developed architecture and checklist were then applied to two detailed, well-known, and publicly available scenarios.

- The first was a geo-political game from the Reagan Presidency, known as Ivy League, described in detail by Allen (1987) for his comprehensive text on War Games.
- The second was “Fomblers Ford”, the DARPA re-write of Swinton’s classic from the Boer War era “The Defence of Duffers Drift”. The DARPA version is set in the Balkans of the early 1990s, with equipment appropriate to the electronic age.

These tests confirmed some components will be found in all scenarios, but that not every component – and certainly not every item – will be found in all scenarios.

3. The proposed Architecture

The revised top-level architecture is shown below. This is not a flow chart or a process, but purely a representation of those components, and items, which we have found in scenarios, and with some indication of the relationships and dependencies between them. Hence, in Figure 4, the components “Purpose” and “Scene” are shown as having a relationship, and the direction of that relationship is indicated.

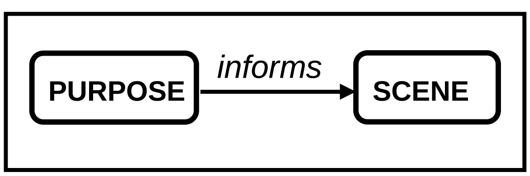

Figure 4: Inter-component Relationship

Figure 5 shows a component decomposed into items. Multiple relationships are not shown, but may potentially exist. Hence, the architecture contains no information on whether there is – for example - more than one purpose, or more than one scene, but this can be shown in the checklist, or using the paper version of the architecture.

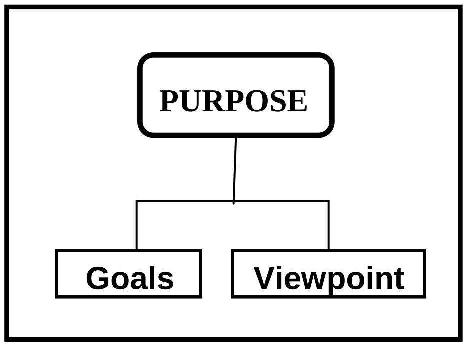

Figure 5: Component Decomposed

The full top-level architecture model is shown in Figure 6, and aims to show the generic structure of a scenario. This is intended to provide a representation of all possible components, their items and relevant component-item relationships. Definitions have been offered for each component and item.

The initial view formed from this work was that there were three levels to be considered for a scenario:

1. **The initial requirement**
   This is not, in itself, amenable to any manual or automated checking process, in that if an authority (individual or group) has called for a scenario, then the requirement is sufficient in itself, and validation is not an issue.

2. **The architectural level**
   We have offered an approach to checking that a required scenario will meet a standardised architecture (we refer to this as the primary architecture). Since this starts by considering a statement of purpose for the scenario, this does act to check that the initial requirement is properly documented.

3. **The detailed scenario level**
   Having established that an outline scenario conforms to the primary architecture, it is now appropriate to move toward describing the scenario in detail. The MSDL (SISO, 2008) looks as if it will become the established standard for such a description (indeed, we are not aware of any alternatives currently on offer). A process to check for detail conformity to the MSDL is described later (section 6).

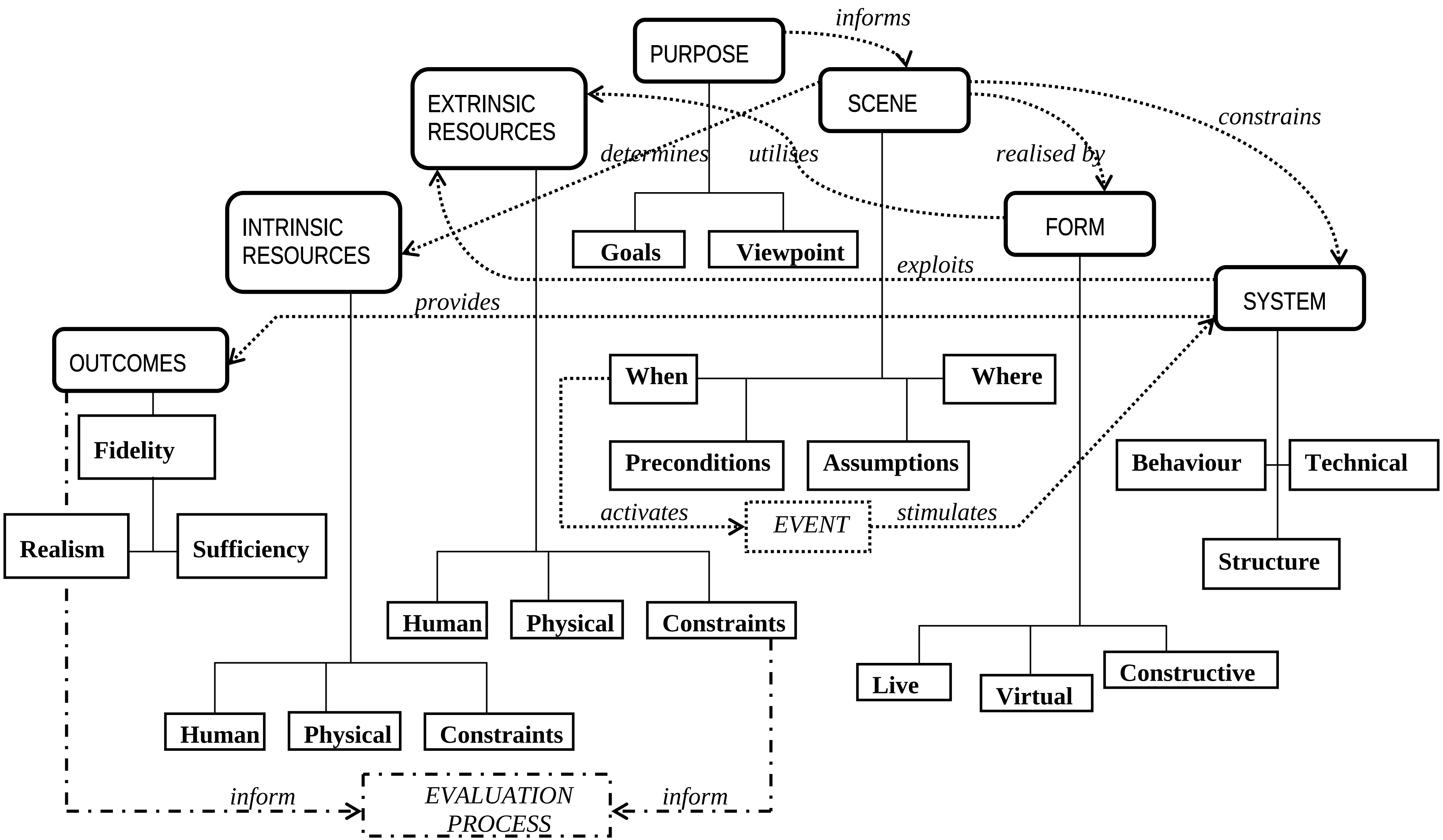

Figure 6: Scenario Architecture

As we started to consider the level of detail required by the MSDL, it became difficult to see direct links between the Architecture, and the MSDL. This suggested that there is a need for an intermediate stage between the two. We have explored this by means of a functional decomposition of the Architecture. A decomposition to the third or fourth level has identified the scope for sufficient detail as to enable a mapping onto the MSDL. As an example, one component of "Purpose" in the original architecture is "Goals"; but each level of command in a training exercise may have a different goal. In the example (below) of a Scenario to train a Battle-group, both Platoon and Company Commanders will have to consider training for their own sub-unit as an entity, and also as part of the larger formation to which they belong. The intermediate decomposition allows for the following alternative Goals:

Training	Experimentation
Mission planning	Mission Rehearsal

and the Views component accepts the multiple interests shown in the example in this paragraph.

4. Testing Scenario against Architecture

4.1 The Primary Architecture

The Architecture, as a list of components and items, was set down in the form of a checklist (with both flat paper and spreadsheet variants). The procedure was to take a scenario, number the paragraphs, and then to record on the checklist the location of each component. A column for notes allowed the analyst to indicate which part of a paragraph was considered to be the relevant item (if that was considered to be necessary). It was found that, whilst any specific scenario did not have to contain all the identified components, some components were essential to any scenario. One particular problem, found with many alleged scenarios, was the lack of a stated purpose.

A number of scenarios were examined, and it was concluded that a purpose was always necessary. It was usually obvious, but the purpose was not invariably stated as part of the scenario itself. Attention has not yet been given to the extent to which a purpose may be removed from the actual scenario, but for the time being it is considered that this should not exceed two removes. On this basis, Document 1 may state the purpose of a scenario, and Document 2 may both refer to Document 1, but also authorise the generation of a scenario. The scenario then becomes Document 3. This could (typically) arise in the case of the Infantry Platoon training cited earlier. This gave a level above that of the architecture for the Purpose (or the formal requirement), and it seems clear that a scenario requires – as a minimum - those components that will allow it to meet its purpose. A scenario must therefore be purpose-oriented, and from this the following proposition was offered in Whitworth Hone & Farmilo, 2007:

1. Any scenario must have a declared purpose

If the purpose-oriented nature of a scenario is accepted, then it follows that any scenario must also have an architecture that will allow it to meet its purpose. Hence, while any two or more scenarios may have differing sets of components (detail architectures), they will all require those components necessary to achieve those purposes. This lead to the second proposition:

2. Any scenario must contain all components necessary to meet the declared purpose

4.2 An Example of Testing.

Some of our work on Task and Work Analysis (not reported here) has been based on a UK Training Video depicting one activity practiced by an Armoured Battlegroup (BG). A BG is the key combat unit in the British Army, and is comprised of Armour and Armoured Infantry, plus some support units, which combine into a coherent combat unit. A scenario was based on that training video (it ran to only six paragraphs), the paragraphs were numbered, and the scenario was then compared to the architecture using the process set out in Section 4.1 above. The first paragraph is shown here (Figure 7a), together with a matching extract from the assessment form (Figure 7b).

Scenario: Armoured Battle-group in Quick Attack

Para. 1
One outcome of the change from the attrition warfare of the 60's and the 70's, to modern manoeuvre warfare, is that that armoured formations need substantially more space in which to train. The training requirement is still that all levels of command are able to practice their planning and operational skills, both individually, and collectively, without any reasonable hindrance, and in a realistic manner. For an armoured battle-group, this has the effect that training areas such as that of Salisbury Plain are now no longer big enough (crossed as the area is by a number of public roads) and the time allocation on larger areas such as the British Army Training Unit - Suffield (BATUS) mitigates against conducting the early stages of training in battle-group operations. With this in mind, most battle-group training has to be done in a synthetic environment.

Figure 7a: Extract from test scenario

Scenario	**Armoured Battle-Group in Quick Attack**

Purpose ... The reason the scenario is required	Training a BG in all command levels	1
Goal ... What is intended to be achieved	Practice in Planning and Operational Skills	1
Viewpoint ... The position that values the goal		

Figure 7b: Extract from assessment form

4.3 Getting the whole thing right.

No scenario is developed in a vacuum. Some authority has identified the need for a scenario and authorised its generation, someone must have written it, someone else must have assessed it. Considered as a process, it is then necessary that the finished scenario satisfies the basic requirements of Validation, Verification and Accreditation (VV&A) that would be applied to any simulation.

A simplified view of VV&A is that it asks three questions:

Have we built the right product?
Have we built the product right?
Does it do what we wanted it to do?

When these are satisfied (affirmative answers), the scenario can be accredited for use against the current requirement. A similar process allows for the use of standard models across a number of simulations, once the VV&A process has been completed.

The process above is independent of the detailed task of writing the scenario. While there are several tools which will take a set of specifications and write a detailed scenario from them, these may now need reconsideration. The Military Scenario Definition Language (MSDL) offers a standard format for developing scenarios, is already a SISO standard, and is expected to become an IEEE Standard during 2009. A present need is to confirm that any scenarios developed:

1. **meet** the basic requirements for a scenario as described above.
2. **include** all other critical information.
3. **conform** in their internal details to the prescribed formatting.

Further, since the greatest single use for Scenarios is for training, and given the importance of Training as outlined in Section 2, we can take another view of VV&A and ask:

Are we training the right task?
Are we training the task right?

It now becomes important that there is a traceable route from the initial requirement right through to the MSDL. This will require the intermediate step suggested above at the end of section 3.

As with the check against architecture, this can be done using a "flat-paper" or computerised approach. Our preference is for the computerised approach, as this does facilitate the maintenance of an audit trail. To facilitate this part of the work, we used our HTA TOOL (see ref) to decompose the primary architecture from its existing two levels down to three or four levels (the secondary architecture). This was sufficient to enable a direct mapping with some parts of the MSDL. The decomposition was effected using the HTA TOOL's "Indented List" option, and then the view was changed to the "Tabulated List" option which permits the making of notes (as in Figure 7b), and even the incorporation of illustrations. Examples of the two forms are given below in Figure 8a and b. The greyed areas show the items that were covered in the Primary Architecture. It will be seen that the further decomposition is easier to follow in the Indented form; against this, the Tabular form offers the facility to add notes, unit designations, etc. As before, there is a trade-off between "ease-of-use" and "ability-to-document". We consider that the Indented form should be considered as a check-list, while the Tabular form can constitute a more detailed check.

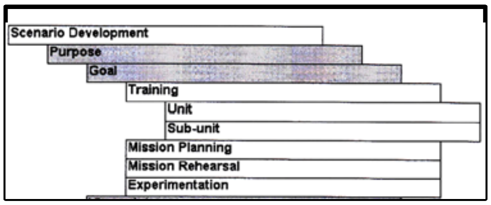

Figure 8a: Extract from Indented list.

Super-ordinate	Task	Notes
0	**Scenario Development** 1 Purpose 2 Scene 3 Form 4 Extrinsic Resources 5 System 6 Intrinsic Resources 7 Outcomes 8 Evaluations	
1	**Purpose** 1.1 Goal 1.2 Viewpoint	
1.1	**Goal** 1.1.1 Training 1.1.2 Mission Planning 1.1.3 Mission Rehearsal 1.1.4 Experimentation	
1.1.1	**Training** 1.1.1.1 Unit 1.1.1.2 Sub-unit	
1.1.1.1	**Unit**	
1.1.1.2	**Sub-unit**	

Figure 8b: Extract from Tabular list.

The “Notes” column of the Tabular list does also permit the analyst or training planner to cross-reference the training requirements that have previously been identified (enabling a check on the validity of the training planned by comparison with the desired training task), and will permit similar referencing of scenarios for mission rehearsal and mission planning. We believe that this enables the scenario developer to pursue the validation process in an incremental manner, as many times as necessary, whilst maintaining both an audit trail, and contact with the end-users.

5. Checking the scenario in detail

The Federation Object Model (FOM) Checker, developed at RMCS Shrivenham almost a decade ago, relied on placing data into a database which could then be interrogated. Software engineering approaches since that time would argue for a new method. Our proposal is that the tool would utilise the latest version of the MSDL (currently the Standard V1) as a baseline for interrogating a new scenario file, written using that schema. This will take the form of its own MSDL XML file which can be loaded into the tool before comparison with the new scenario. By using an XML file separate to the tool itself, the tool will be able to use the latest version of the MSDL without requiring regular updates. As the MSDL has only just been accepted as a SISO Standard, and regular modifications are anticipated, this is seen as a vital feature. The checking loop will resemble that in Figure 8.

In this arrangement, the scenario is developed on a PC or laptop. The developer then accesses the Scenario Checker web application, where they can select to upload the MSDL file for checking. The file is first compared against the latest standard MSDL file for initial compatibility, before being analysed by the tool for completeness. A report is then sent back to the user and displayed within the web application, facilitating revision and resubmission if needed.

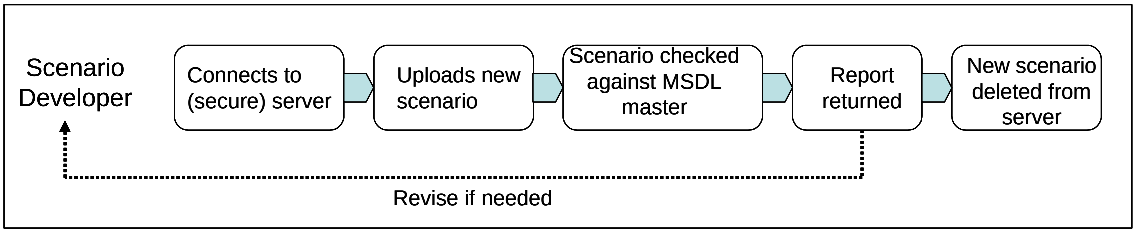

Figure 9: Scenario Checker design

A web-based tool is recommended as the best format for the scenario checker because it could provide easy, cross-platform access for the majority of potential users. They would not need to install any additional files or applications, nor keep them up to date, as the one latest version of the tool would be that provided by the web service. Security would also benefit from this approach. A tool intended for web-based use could be run on a stand-alone server (PC even) when highly sensitive scenarios were under development. For other sensitive scenarios, a secure web-server (i.e. password protected) would offer adequate security. Where only commercial considerations need to be considered, the sole problem would be that of finding a willing host. Such a host would not need to keep a copy of any new scenarios on its server as the only file(s) required would be for the MSDL Schema.

The Schema defines every possible element allowed within an MSDL XML file, including attributes such as minimum occurrences and data type. It even defines the actual pattern of strings, a Country Code for example, whose value must be `“[A-Z*]{2}”`. This notation indicates that the string should be 2 characters, both upper case letters, such as “UK” or “US”. These XML Schema files can be used directly within the tool for checking against new scenarios.

In the development version, the first check pass verifies that the MSDL Schema is complete in the sense of meeting basic MSDL requirements, and the second pass confirms that all items identified in the second level architecture decomposition are present.

An alternative approach is envisaged where the first pass of the checker would confirm or deny the existence of all MSDL defined elements, i.e. a basic check against the MSDL Schema, highlighting any omissions, particularly those which are key components of a Scenario such as the Purpose (as in our second proposition above: **All scenarios must have a declared purpose**). This should be included within the Objectives of the Plan component of MSDL. If this was not the case then such an omission would be highlighted. If highlighting were to follow a “traffic light” convention then such a position would indicate immediately that the scenario required some attention before use.

The second pass would check those elements that do exist against the format required by the MSDL. This would be appropriate for the Country Code described above. An alternative example is the “ScenarioTime” attribute of the Environment component which should be in the format `DDHHMMSSZMONYY`, and such checks could be easily performed. Any deviation from the required format would be highlighted.

The resultant report could have the format shown in Table 1 (some sub-elements are omitted for the example). It is envisaged that such a tool could eventually be built-in to a scenario creation application, checking in real-time as the scenario is created to avoid the need to resubmit to a scenario checker. Also in development is a modification to the MSDL Checker to generate a report that can be passed back to the HTA TOOL for direct comparison with the extended scenario decomposition, and also exported to Excel for use in reporting and general documentation.

No	Name	Exists	Correct Format
1	Options	Yes	N/A
2	Plan	Yes	N/A
3	Environment	Yes	N/A
3.1	ScenarioTime	Yes	No
3.2	Terrain	Yes	N/A
3.2.1	TerrainSource	No	
3.2.2	UpperRight	Yes	Yes
3.2.3	LowerLeft	Yes	Yes
3.3	Weather	Yes	N/A
3.3.1	Temperature	Yes	Not within expected range

Table 1: Sample Scenario Checker report

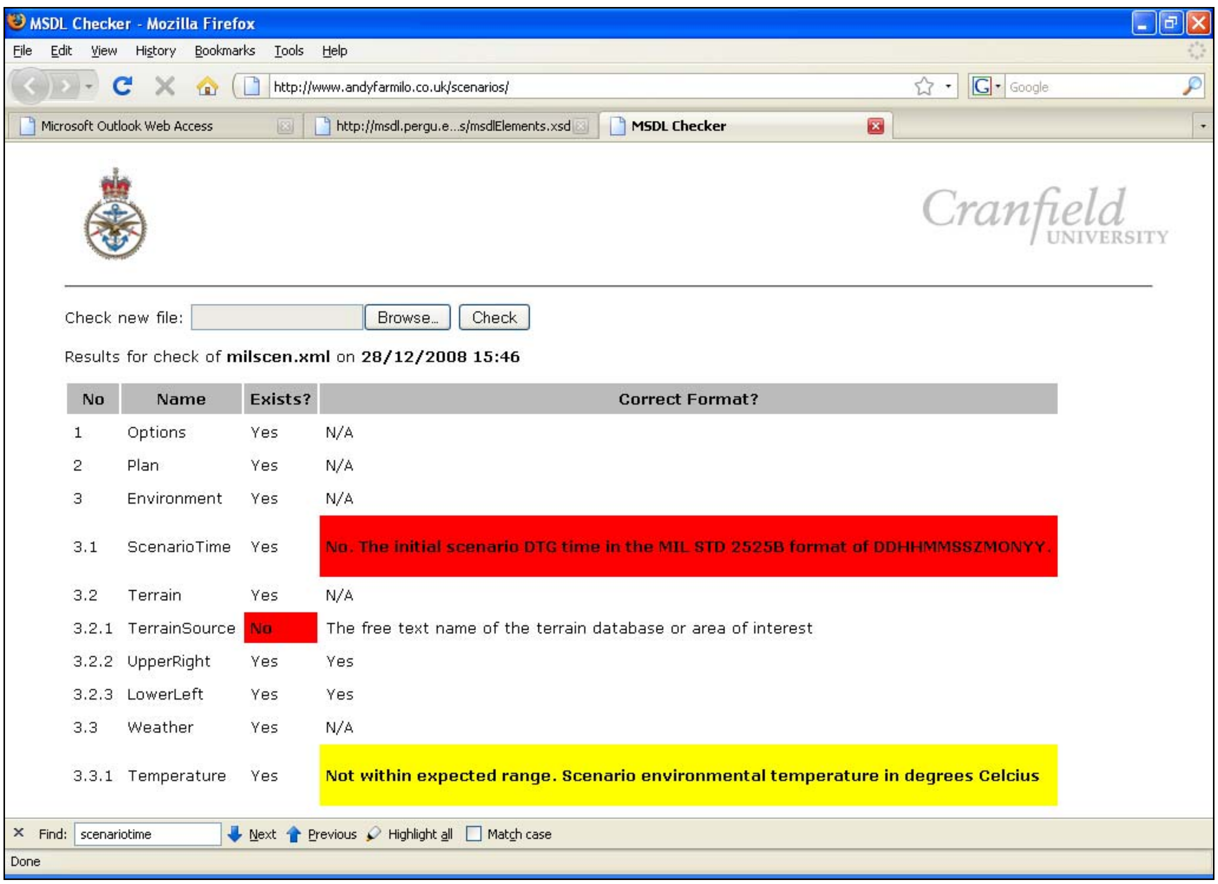

Figure 9: Screenshot of potential Scenario Checker report

The web-based tool allows a basic but intuitive interface, enabling the user to browse for the XML file to check against the MSDL Schema. The user then initiates the background comparison and analysis (with progress bar), the results of which are then displayed in an HTML table. However, as Scenario files are likely to be very large (probably hundreds if not thousands of elements), a summary of the most critical omissions from the XML table (as shown above in Figure 9) may prove to be more immediately useful. The screenshot also shows how the description of an element from the Schema should be shown - where there is a problem - to provide the user, or reviewer, with a more detailed and human response on how to correct the issue.

6. Conclusions

6.1 Future Work

We see several interwoven strands to the future of this work:

- **The tool for scenario primary architecture assessment**.

This currently exists in flat-paper form and in a spread-sheet version (both freely available to anyone wishing to experiment). Rather than trading speed of use against ease of reporting, it is for users to decide on their need for either or both. There seems no be no need to develop this further, in that its function is to act as an early error-trap.

- **Scenario type identification**

We had originally envisaged the possibility of there being several sub-sets of the architecture, which could be described as “Type 1”, “Type 2”, etc. Investigating this would require a substantial number of assessments made using the original tool, followed by some data collation, and it may well be that either the MSDL Schemas, and/or the expanded architecture, can offer a better way of handling such variations. Certainly, the expanded architecture offers the potential for a finer-grained assessment.

- **Consideration of the level-of-detail between the Architecture and the MSDL**

This is currently discussed in Section 4.3, and forms the bridge between the Primary Architecture, and the MSDL. We are currently using a Task Analysis tool to assist with the decomposition, and this gives us the option of hierarchical or tabular output (again, with a trade-off between speed, and the ability to document the development of the scenario requirements). This may also facilitate the integration of the military requirement, and the scenario under development, as the particular Task Analysis tool used has been optimised for use in military training, and is currently in service for this purpose.

- **Further development of the tool for checking a scenario written in the MDSL for completeness and correctness and a possible extension into realism**

This may also have to relate to the MSDL Schemas which form part of the whole MSDL approach. Goodwin (2006) has stressed the need for realistic scenarios for training, and the MSDL relies on the use of MIL-STD 2525B references, which must provide a measure of realism. Note, however, that realism should not be confused with fidelity.

6.2 Summary

There is substantial evidence of tools to assist in the writing of scenarios, but all those readily available appear to be domain dependent. There is no readily available evidence relating to work on the generic basis of scenarios, or of work on tools and methods for the assessment of a scenario on a generic basis. It is believed that the approaches outlined here will, to some degree, overcome this. There does not, as yet, appear to be any standardisation in the terms used to describe, or to discuss, a scenario, and the MSDL has the promise to overcome this. Whilst the component definitions offered here may help in this direction, we feel that the matter should be addressed more formally.

It is believed that the tools and approaches discussed here can facilitate the V&V of a scenario, prior to its employment in a simulation, and having full regard to the intended purpose of that simulation. Further, it does so in an incremental matter.

There is some limited evidence that scenarios can be categorised into a relatively small number of domain-independent, and application-independent types. This is referred to in the preceding sub-section. It would, logically require a separate research effort, and with scenario detail design formalised at the MSDL level, it may not now be of value.

7. References

Alberts D.S., Hayes R.E., Leedom D.K., Kirzl J.E. and Maxwell D.T. (2002) **Code of Best Practice for Experimentation**. Washington, DC: CCRP (see also NATO, 2002)

Allen, T. B., (1987) **War Games**. London, William Heinemann Ltd

DMSO (2006) **The DMSO Modelling and Simulation Glossary**. Downloaded on 6-Mar-2006 from : https://www.dmso.mil/public/resources/glossary

Goodwin, G.A., (2006) The Training, Retention and Assessment of Digital Skills: A Review and Integration of the Literature. Research Report 1864, Washington DC, the US-ARI.

Gorman, P., (2000) **Fomblers Ford**. Washington DARPA

Hone, G.N., (1995) **Combined Arms Training: A Workshop on Training Matrix Design.** DRA/CHS/HS3/TR95027/01. Farnborough, Hampshire; Defence Research Agency.

Hone, G., Whitworth, I.R. and Farmilo, A. (2008) How good is that scenario: A Levels-of-description Approach. **Proceedings of the Fall ’08 Simulation Interoperability Workshop**. Orlando, FL. Simulation Interoperability Standards Organization (SISO)

Mandel, R., (1977) Political Gaming and Foreign Policy Making During Crises. **World Politics**. July, 1977.

Morales, M.C., Moulding, M.R. and Newton, A.R., (1999) **Improving the Specification of Synthetic Environments: Towards a More Rigorous Approach for the Automated Checking of HLA Specifications**. SISO Spring ’99, Paper #171.

NATO (2002) **The NATO Code of Best Practice for C2 Assessment – Revised 2002**. NATO SAS-026, 039. Reprinted by the CCRP (see also Alberts, Hayes, et al, 2002)

SISO (2008) **Standard for: Military Scenario Definition Language (MSDL).** Simulation Interoperability Standards Organization (SISO), Inc; Orlando, FL

Whitworth, I.R, Hone, G. and Farmilo, A., (2007) **Developing the Scenario Assessment Tool.** Proceedings of the NATO M&S Conference, Prague, Czech Republic Oct 2007.

Whitworth, I.R, Smith, S., Hone, G. and Macleod, I. S., (2005) **How do we know that a scenario is appropriate?** Proceedings of the 11th ICCRTS, Maclean, VA.

Notes:
1. Duffer's Drift was originally published in 1907. The copyright lapsed and for about 12 months, in 1992, Swinton's original text was available for public download from the US Army Training website, using the URL:

http://www.adtdl.army.mil/cgi-bin/atdl.dll/misc/duffers-drift/duffers-drift.htm

The work has since been reprinted, acquired a new copyright, and is now only generally available commercially (the US Army website above is now "access by account only").

DARPA continue to make "Fomblers Ford" freely available.

DMSO became the Modeling and Simulation Coordination Office in 2007, and their glossary now contains four definitions of a "Scenario".