The Validation of an Infrared Simulation System

Alta de Waal, Cornelius J. Willers, J.H.S Roodt and Azwitamisi E. Mudau
Defence, Peace, Safety and Security (DPSS), CSIR, South Africa,
Waikato Institute of Technology – New Zealand.
adewaal@csir.co.za, nwillers@csir.co.za, henk.roodt@wintec.ac.za,
amudau@csir.co.za

Abstract. A commonly-used term in the simulation domain is 'validation, verification and accreditation' (VVA). When analysing simulation predictions for the purpose of system solution development and decision-making, one key question persist: "What confidence can I have in the simulation and its results? " Knowing the validation status of a simulation system is critical to express confidence in the simulation. A practical validation procedure must be simple and done in the regular course of work. A well-known and acknowledged validation model by Schlesinger depicts the interaction between three entities: Reality, Conceptual Model and Computer Model, and three processes: Analysis & Modelling, Programming and Verification, and Evaluation and Validation. We developed a systematic procedure where each of these six elements is evaluated, investigated and then quantified in terms of a set of criteria (or model properties). Many techniques exist to perform the validation procedure. They include: comparison with other models, face validity, extreme condition testing, historical data validation and predictive validation - to mention a few. The result is a two-dimensional matrix representing the confidence in validation of each of the criteria (model properties) along each of the verification and validation elements. Depending on the nature of the element, the quantification of each cell in this matrix is done numerically or heuristically. Most often literature on validation for simulation systems only provides guidance by means of a theoretical validation framework. This paper briefly describes the procedure used to validate software models in an infrared system simulation, and provides application examples of this process. The discussion includes practical validation techniques, quantification, visualisation, summary reports, and lessons learned during the course of a validation process. The framework presented in this paper is sufficiently general, so that the concepts could be applied to other simulation environments as well.

1 Introduction

Software simulation of hardware systems plays in integral part in the development and evaluation of complex systems.

Figure 1 depicts typical steps in the development process of a complex system: The characterisation of the system of interest provides inputs for modelling. The objective of characterisation is to build conceptual and computer models. Through repeated design and testing in the simulation environment, a design solution is synthesised. These designs are implemented in the real world (as hardware and software, process and procedures) and deployed for evaluation and operational use. Key to this process is continuous re-evaluation and improvement of the solution. Simulation is the only way to support the evaluation of hardware solutions cost-effectively. Ultimately, in order for the simulation to contribute and support the design synthesis and evaluation flow (Figure 1), one must have confidence in the simulation and its results. Some measure of confidence can be achieved by validating subsets of the simulation (Willers & Roodt 2011). The applicability of a simulation system was shown to have a high confidence only in subsets of the application domain where the simulation was validated against real-world scenarios. However, (Willers & Roodt 2011) also showed that with careful model design and extrapolation of results, the number of points to test for validity can be minimised, whilst maintaining a high confidence in the simulation. This is done by choosing critical points across the operational domain of the system to be validated. This led to the conclusion that validation of simulation models can contribute to the objective confidence in a simulation. Estimating simulation confidence requires careful consideration of three distinct elements of the system under investigation: (1) the underlying real world phenomena, (2) the conceptual models of the phenomena, and (3) the computer models and the scenario descriptions used in the simulation. Schlesinger defined a model for the validation and verification process (Sargent 1999, Schlesinger 1979) as described in Section 4. A practical validation procedure must be simple and done in the regular course of work, otherwise it becomes too expensive and impractical. It is worthy to note that the process as depicted in Figure 3 later in this paper can be translated into a similar process for the development of hardware solutions, and we will use this fact to link the work done already in modelling and simulation in 1979 to the work done in project management in the 1990s and adopted in systems engineering during that time (see Figure 2).

This article refines a validation procedure already introduced in previous papers (Willers & Roodt 2011, Willers and Wheeler 2007) and applies it to countermeasure and aircraft models in an infrared simulation system. The objective is to illustrate the practical implications of the validation procedure - both in application and representation of results. The article concludes with lessons learned and recommendations.

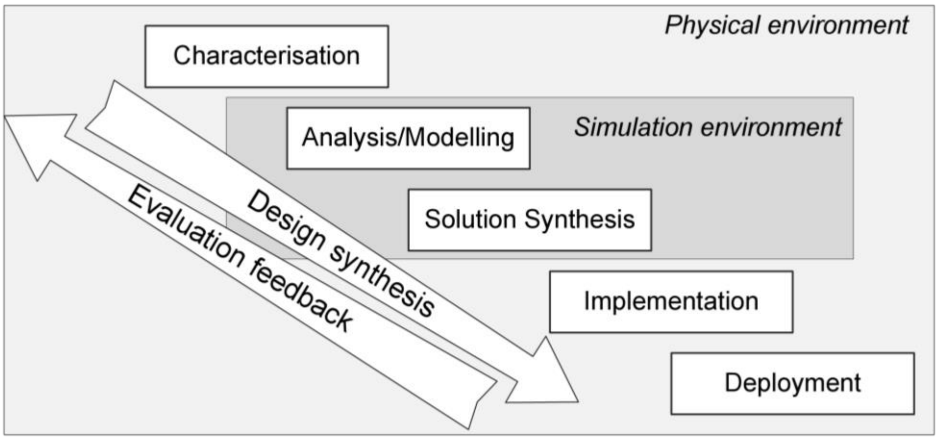


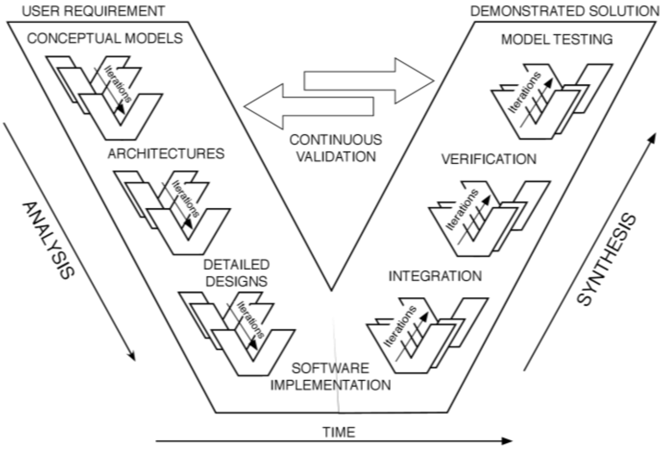
Figure 1: Design synthesis and evaluation flow

2 The System Engineering Context

Building on the thinking in management of large defence projects, software development, technical product development methods and defence acquisition approaches, Forsberg and Mooz proposed the "Vee" process model starting with user needs on the left top of the V and the validated system on the top right (see Figure 2 adapted to reflect the development of a software simulation system and models for it). The left side of the V follows the classical analysis/decomposition approach used in the waterfall model and the right hand side follows a classic synthesis and assembly approach, verified and validated against the elements in the descending analysis arm (Forsberg and Mooz 2007). In principle, the Vee Process Model (as it is formally called) allows for tried and tested baseline management during system development. By using several superimposed Vee models, it is possible to consider a system-of-systems.

The International Council On System Engineering (INCOSE) reflected on the future of System Engineering (SE) back in 2007 and highlighted the trend toward model-centric approaches in several of the engineering disciplines. A significant part of the move towards model-based system engineering (MBSE) would be the development of interoperable tools and approaches to support the widespread adoption of the methodology (INCOSE 2007). In the six years since, several system modelling standards have emerged, including SysML^{TM}. Modelling and simulation systems are now embedded in all areas of the system life cycle and this trend is expected to continue.

Models start off as conceptual artefacts that relate to physical or theoretical phenomena. Sometimes these models are described in formal languages (mathematics, for example) and software coded versions are developed. The computerised model can be simulated in a simulator environment and the behaviour checked. This behaviour should correspond to the behaviour of a physical system (on which the model was based) under strict bounding conditions. In addition and similarly to any product developed using the Vee process model, a mathematical or software model is developed in response to a user requirement or need and it must finally be validated to ensure that the developed solution (model) fits with the requirement (this is the reason for insisting on the continuous validation steps between the arms of the Vee). Formally: *Model Validation consists of delivering substantive evidence that a computerised model possesses a satisfactory range of accuracy consistent with the intended* *application and within the application domain* (Schlesinger 1979). These models can then be used to experiment with in simulation environments that must be robustly verified too.



The "Vee" process model cast within the context of model development. Adapted from Kevin Forsberg and Harold Mooz (1995) by JHS Roodt 2013.

Figure 2: The Vee Process Model as Applied to Simulation Systems

Modelling standards and appropriate modelling methodologies (combined with deep insight into the phenomena being modelled to ensure the integrity of the conceptual models) play a critical role during the analysis phase, while adhering to the best practices in software development standards and approaches (Agile, Lean Enablers for System Engineering, etc.) is non-negotiable during the model synthesis phases. This understanding forms the backdrop to the approach followed for the verification of a system of models that together make up a specific sub-set of an infrared electronic warfare simulation environment. *Model Verification consists of delivering substantive evidence that a computerised model (software design and coded/programmed) represents the conceptual model to specified accuracy* (Schlesinger 1979). In this way the derived 'solution' is checked that it is indeed sound and robust and at the same time it is continuously validated against the user requirement. We believe that this *combined* approach is sufficiently generic to be applicable to similar physics based, deterministic models.

3 Terminology and Applicability

This paper starts with the terms 'validation, verification and accreditation' (VVA), and proceeds below with the terms Qualification, Verification and Validation. The intent with all these terms is to ensure, measure or quantify confidence in a system – a simulation system in this case. Different laboratories tend to use different terminology. Qualification, Verification and Validation are all defined in the next section. The term 'validation' is very often (also in this paper) used with two different meanings: (1) a specific, well defined lower-level step in the process and (2) in a broader more generic context to describe the whole process, including qualification, verification and low-level validation. Adding to the confusion, the process is often referred to as the 'VVA' process, but most often *not* including a formal accreditation step.

Accreditation is the fourth term not considered in this paper: it is the act of granting credibility, recognition, acceptance or formal approval by a recognised body of a product such as a simulation system. Accreditation is handled by an institution outside the environment where the work is done and hence, is beyond the scope of this paper.

The process is described in this paper in the context of software and software simulation development. The same principles can be applied with suitable changes to the process in other disciplines as well.

4 Validation Procedure

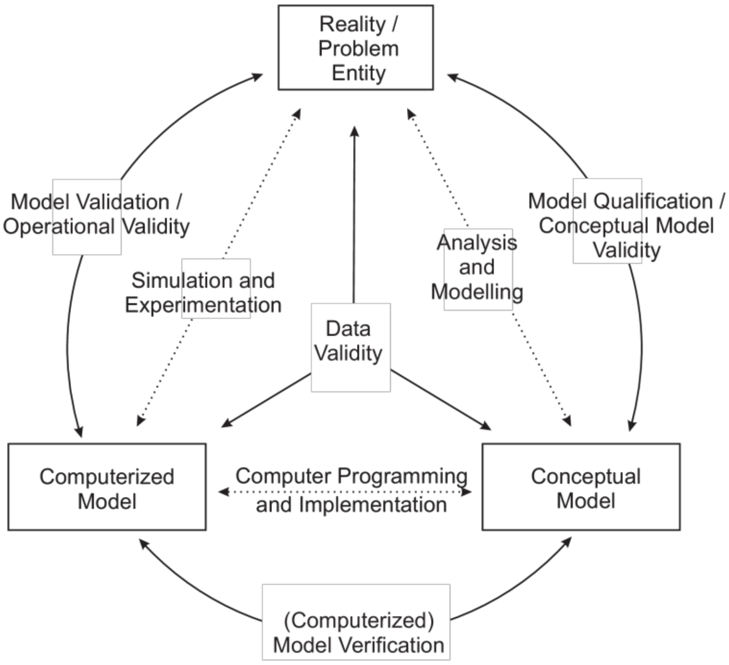


Figure 3: Simplified version of the Verification and Validation Process [extended from Sargent 1999, Schlesinger 1979]

4.1 Validation Elements

Based on Schlesinger's model (Schlesinger 1979), depicted in Figure 3, (Willers and Wheeler 2007) formulated a process quantifying model properties in terms of the *three entities*: Reality, Conceptual Model, Computer Model, as well as the *three transitioning processes*: Analysis and Modelling, Software model design and programming and Verification, and Experimentation and Validation.

1. *Reality* is an entity, situation, or a system which has been selected for analysis.
2. A *Conceptual Model* is a description (verbal, mathematical, governing relationships or natural laws) that purport to describe Reality.
3. A *Computer Model* is an operational computer programme which implements a Conceptual Model.
4. *Model Qualification* consists of delivering substantive evidence of the adequacy of the Conceptual Model to provide an acceptable level of agreement within the application domain of the model.
5. *Model Verification* consists of delivering substantive evidence that a computerised model (software design and coded/programmed) represents the conceptual model to specified accuracy (Schlesinger 1979).
6. *Model Validation* consists of delivering substantive evidence that a computerised model possesses a satisfactory range of accuracy consistent with the intended application and within the application domain (Schlesinger 1979).

In 1979 the context of software design was often restricted to how the different subroutines were programmed to interact with data files. In that context the problem often was deciding whether the software program using the appropriate routines was delivering outputs to within the expected accuracy as required by a conceptual model. For example: was the second-order derivative numerically calculated to within the required accuracy of a range of interest. Today it is assumed that -

the fundamentals are sufficiently in place, and the focus shifts toward the application: the total object being modelled is considered for accuracy in behaviour. The fact remains that we still need to do a proper design of the software before implementation, including the use of frameworks like UML or SysML. Programming is just one step in the process, and it is this combination of model design, software design, software writing (coding) and transforming it into executable code that must be considered during verification. The final product, as implemented as an executing software is called the simulator. The simulator may be purely software based, or may include interfacing to external hardware and may even employ humans interacting with the software.

The validation/verification is achieved by considering the following elements:

1. *Reality* is evaluated in terms of the scope, quality, relevancy and quantity of the available/measured information. The theoretical knowledge base is also considered - how well is the object understood?
2. *Analysis and Modelling* is evaluated in terms of the quality of information extracted, the internal and historical consistency of the results, and especially important, results matching with theoretical model predictions. Also important is the quality of the tools and means of analysis.
3. *Conceptual Model* is evaluated in terms of the scope, quality and strength of the model, and the degree to which static and dynamic properties are understood, modelled and quantified. The conceptual model is the mathematical representation of the problem entity developed for the particular study.
4. *Software design and programming and Verification* considers the degree to which the conceptual model is captured, and the scope of coverage of static and dynamic properties in the data and software code.
5. *Computer Model* is evaluated in terms of how well the computer model matches the conceptual model, using the same evaluation as for the conceptual model.
6. *Experimentation and Validation* is evaluated in terms of matching the simulation output (in the case of the infrared simulation, the images) with the observed reality (real world measured images). How well does the simulation output recreate the physical representation of the object, in the specific scenario/environment.

The above evaluations are performed using a combination of the techniques defined in Section 4.2. Some of these are quantifiable while others are heuristic and even subjective (but traceable and verifiable).

Expanding on (Sargent 1999), the qualification-verification-validation-based simulation procedure is summarised as follows: (1) Compile a simulation model specification, covering all levels of simulation, from the lower physical object levels, through to abstract integration levels. Ideally, the specification must provide a **set of parameters** or capabilities, with detailed requirements (including parametric accuracy). For each parameter in the list there must be a **set of criteria** for scoring the specific parameter. (2) Compile a value system for a particular simulation task. Not all simulation tasks require exactly the same model scope and detail. The value system serves to point out the relative importance of certain requirements. (3) Develop, test, review and document all (conceptual and computer) model construction activities. Collect, document and archive the information, analyses, and data used to develop the model. (4) Maintain a continued repetitive cycle of qualifying, verifying and validating throughout development (using the techniques in Section 4.2). (5) Maintain a continued internal consistency check throughout development. Extensive regression testing confirms that stable performance against previous versions (temporal consistency). Perform internal cross-model consistency checks (lateral consistency). (6) Maintain a structured and consistent review policy. Figure 4 shows the link between the model specification and the model validation.

4.2 Qualification, Verification and Validation Techniques

Verification and validation are generally done by a combination of objective (statistical or mathematical procedures) and subjective evaluations. A number of tests are described in (Sargent 1999, Irobi et al. 2001 and Martis, 2006) which include (1) Animation, video and operational graphics. (2) Comparison with other models. (3) Degenerative stress testing (how does it break?). (4) Extreme condition testing. (5) “Face validity,” asking people knowledgeable about the system whether the model and/or its behaviour are reasonable. (6) Testing against historical data. (7) Internal validity across related experiments. (8) Parameter sensitivity analysis for perturbations from validated test points. (9) Comparison of predictions and observations. (10) Tracing entity behaviour propagating through the model. (11) Expert opinion and Turing tests. (12) Regression testing. (13) Validating of predictions versus reality.

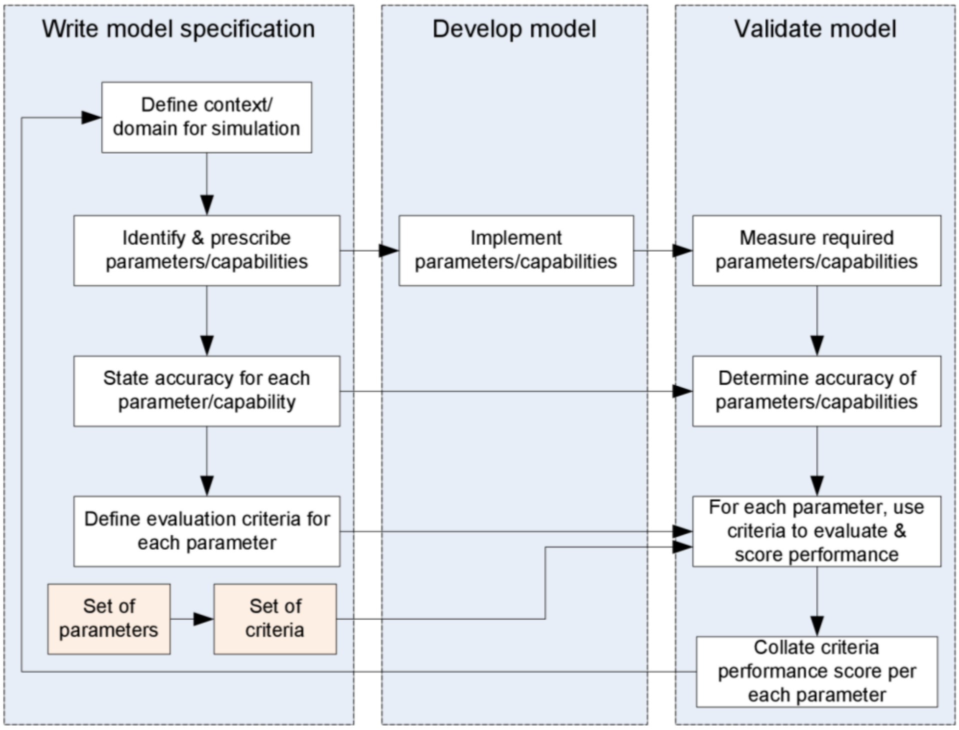


Figure 4: Model Development Process; Specification to Validation

4.2.1 Validation across the Hierarchy

Simulation systems with hierarchical structure of dissimilar building blocks present a particularly difficult validation challenge. As shown in Figure 5, the lower level verification/validation is typically performed at component level with functional unit testing. At higher levels the (dynamic) interaction between functional blocks becomes the dominant factor determining behaviour. The interaction is initially validated by comparing the simulated behaviour with the required behaviour, using carefully designed scenarios. These scenarios are designed to create a specific interaction patterns with known outcomes. Once the interaction and its outcomes are validated, the long term stability of the system is ensured by scenario-based regression testing. Regression testing verifies that changing one component of the system does not affect the behaviour of the system in other respects. Regression testing performs the critically important task of ensuring the stability of a large and complex simulation system under continual development and improvement.

One may ask whether the lower level verification is of interest to the user. While it is clear that the user may primarily want to know that the solution fits the requirement, the user (but also a responsible simulation system developer) may also require that the underlying components adhere to required standards, or combine in behaviour and mathematical outcome in a predictable manner within a specific domain of interest. This is similar to expecting to know that the gearbox will be able to withstand a specific torque once the engine runs in a certain mode, where the mode was the only requirement from the user. This adds to the value of the validated result. The verification (as seen on the right arm of the Vee in Figure 2) is crucial to the on-going validation. It is exactly this insight that brings the system engineering process and the system modelling process into harmony.

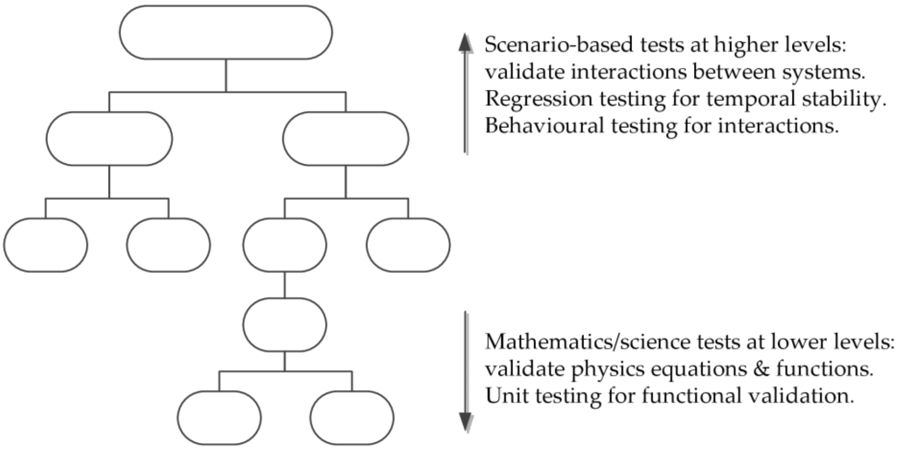


Figure 5: Validation of Hierarchical Systems.

4.3 The validation process in practice

The validation of a simulation (as a numerical model of a natural system) is near impossible as pointed out by (Oreskes, 1994), because natural system are never closed and therefore model results are always non-unique. We can, however focus on the validation of the simulation models as individual entities. In this section, we discuss the practicalities of the validation process as applied to simulation models. In practice, we work towards a validation framework for each model - the framework is the interaction between validation elements (as described in Section 4.1), criteria and techniques. However, we first need to define the specification and value system and then the three entities. To summarise, define the following:

1. Specification and Value System
2. Validation Criteria
3. Validation Techniques

5 Validation of an infrared simulation system - Application

5.1 The Simulation System

This section addresses the application of the validation procedure models in a physics-based C++ object-oriented image generating simulation system. The simulation system under consideration models the radiometric, geometric and kinematic characteristics of objects such that images can be compiled showing the objects from any arbitrarily chosen view point. The objects in the simulation can move in a three-dimensional world in six degrees of freedom. Some of the objects (observers) can form images of other objects in the simulation, and some objects are moving objects (the two are not mutually exclusive). The simulation calculates detailed infrared images, in the requisite spectral bands, from first-principle theory and models. An example of the images created by the simulation, showing an aircraft against a cloud background in three different spectral bands, is presented in Figure 6. One of the applications of the simulation system is to develop aircraft self-protection measures to counter an infrared missile threat. This makes it imperative to validate the radiometry, geometry and kinematic behaviour of all objects, including the countermeasure objects (flares, DIRCM) and the aircraft.

An integrated approach is to use simulation models, in conjunction with field trails, to predict and verify countermeasure effectiveness.

Figure 6: Calculated signature images in the near infrared, the 3-5 µm and the 8-12 µm spectral bands.

5.2 Specification and Value System

The intended purpose and therefore focus areas of the imaging simulation is:

1. High quality and radiometric accurate simulated imagery of target and countermeasure signatures:
    (a) Accurate atmospheric spectral transmittance and background modelling.
    (b) Accurate environment modelling.
2. These target and countermeasure models must generate accurate stimili for the missile to react to.
3. Accurate missile behaviour towards these stimili.

The focus areas above are all used in a comprehensive simulation environment where thousands of missile flights can be simulated, covering a wide variety of scenarios and signature conditions. The intended purpose hereof is to set up effective safety profiles for aircrafts.

5.3 Validation Criteria

The simulation models comprise radiometric signatures, geometric shapes, kinematic behaviour, and combinations thereof, and in some cases, imagers (cameras). The modelling strategy is to model at the lowest practical detail level, using first-principle physics models. This approach yields the best accuracy and wide application, but requires a significant effort in model design, computer programming and testing. The key considerations identified for such an imaging infrared simulation system are: a) Radiometric accuracy in all spectral bands, i.e. sunlight and thermal radiance to provide correct colour ratios; b) Accurate emitting source surface temperature behaviour (aerodynamic or thermodynamic heating); c) High fidelity geometric and spatial texture modelling to provide shape of targets and countermeasures; d) True dynamic and kinematic behaviour in six degrees of freedom; e) Detailed modelling of signatures and backgrounds; f) Accurate atmospheric transmittance and path radiance models; g) Realistic rendering of the scene image in radiometric, spatial and temporal terms; and h) Comprehensive sensor modelling to account for primary and second order imaging effects.

Evaluations should be done for all six elements (processes and models) in Figure 3 using a set of criteria. For the purpose of the countermeasure and aircraft model validation, the following specific criteria are used (relevant only to the infrared imaging simulation):

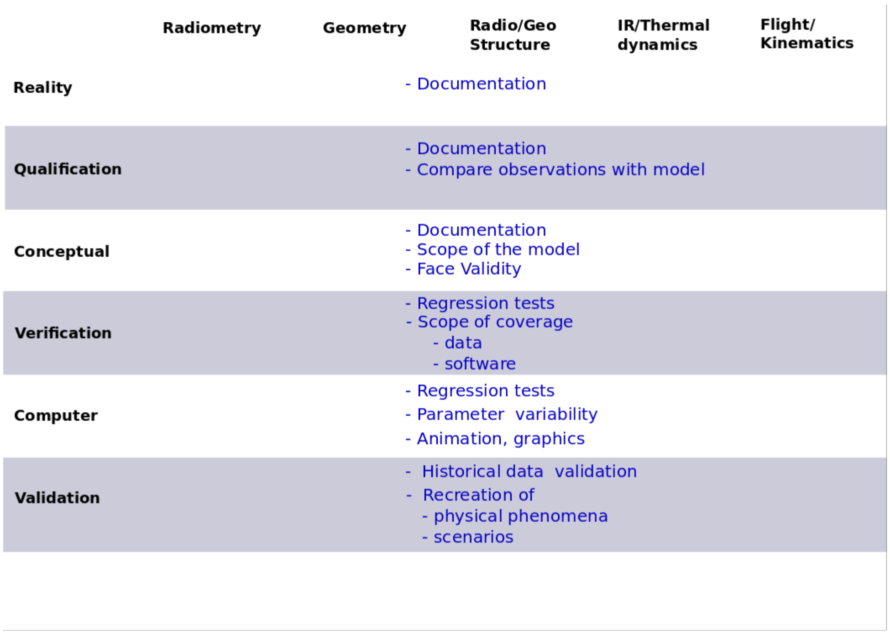
- radiometrc signature properties,
- geometric properties,
- radiometric-geometric properties,
- material thermal properties, and
- kinematic, aerodynamic.

5.4 Validation Techniques

The validation techniques to be used for the countermeasure and flare validation are listed below.

1. Traceability of information and documentation.
2. Face Validity - asking knowledgeable people/experts about the system whether the model and/or its behavour are reasonable.
3. Match observations to model.
4. Scope of the model.
5. Scope of data and software coverage.
6. Animation and operational graphics - Display the model's operational behaviour graphically.
7. Regression testing - Perform repeated tests against previously validated test results in order to verify internal consistency of the models after making changes to code and/or models.
8. Parameter variablility - Change the values of input and internal parameters of a model to determine the effect upon the model's behaviour and its output.
9. Historical data validation - Simulation results will be compared to measured results.

The validation elements, criteria and techniques are the three dimensions around which the validation process evolves. Each relevant interception of these dimensions is investigated in this report. Figure 7 displays the validation framework. Not all validation techniques are relevant for all validation elements as is shown in the grid-like framework.



	Radiometry	Geometry	Radio/Geo Structure	IR/Thermal dynamics	Flight/ Kinematics
Reality	- Documentation				
Qualification	- Documentation - Compare observations with model				
Conceptual	- Documentation - Scope of the model - Face Validity				
Verification	- Regression tests - Scope of coverage - data - software				
Computer	- Regression tests - Parameter variability - Animation, graphics				
Validation	- Historical data validation - Recreation of - physical phenomena - scenarios				

Figure 7: Validation framework for infrared scene simulation.

5.5 Concept Maps

A concept map is a diagram showing the relationships between concepts. It is a graphical tool that organises and represents the knowledge about a concept. The relationships are linking phrases such as "has", "is a function of", and "comprises". Figure 9 is a concept maps about concept maps. Concept maps are used in the validation process to visually compare the conceptual model and the computer model. During the process, two concept maps (one for the conceptual model and the other for the computer model) are constructed and then compared to evaluate if all the conceptual elements are represented in the computer model. Not only is the output of this process (the actual concept map) of value, but also the facilitation process where a workshop of people participates to come to an agreement about the content and form of the concept map. It is therefore both the process and the final result that is of value. In this context it can be seen as a very powerful knowledge management tool.

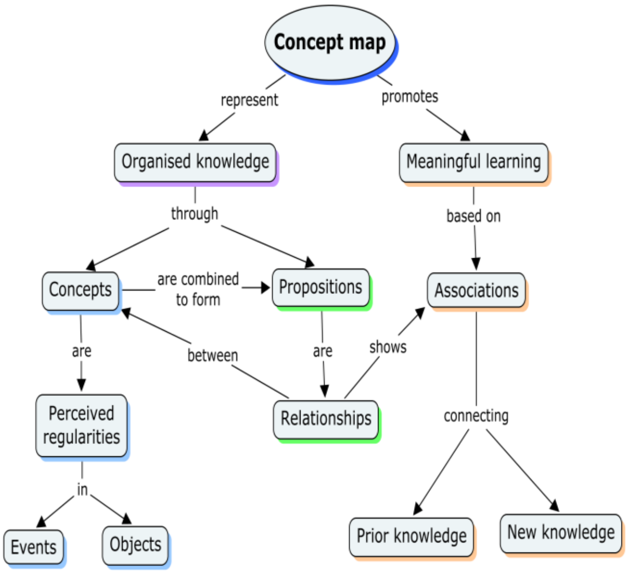


Figure 9: Concept map about concept maps (`http://books.kmi.open.ac.uk/knowledge-cartography/preface/`).

5.6 Visualisation

Intensity is a very important output from the infrared simulation system. Polar plots [1] are a good way to represent the data, but they only give an indication of values across planar sections of the three-dimensional data volume, e.g., the x, y or z cut. Intensity values for each combination of yaw and pitch angles were generated and a spherical plot produced a 3-D visualisation of the flare intensity (using the *pyradi* toolbox (Willers et al. 2012)). This technique provides insight into three-dimensional shapes that are not visible using a 2-D intersect (Figure 10). As can be seen, there is no intuitive way in which the 2-D graphs could be extrapolated to the 3-graphs. In many cases the results were different than expected and pointed out some conceptual errors in the model design phase.

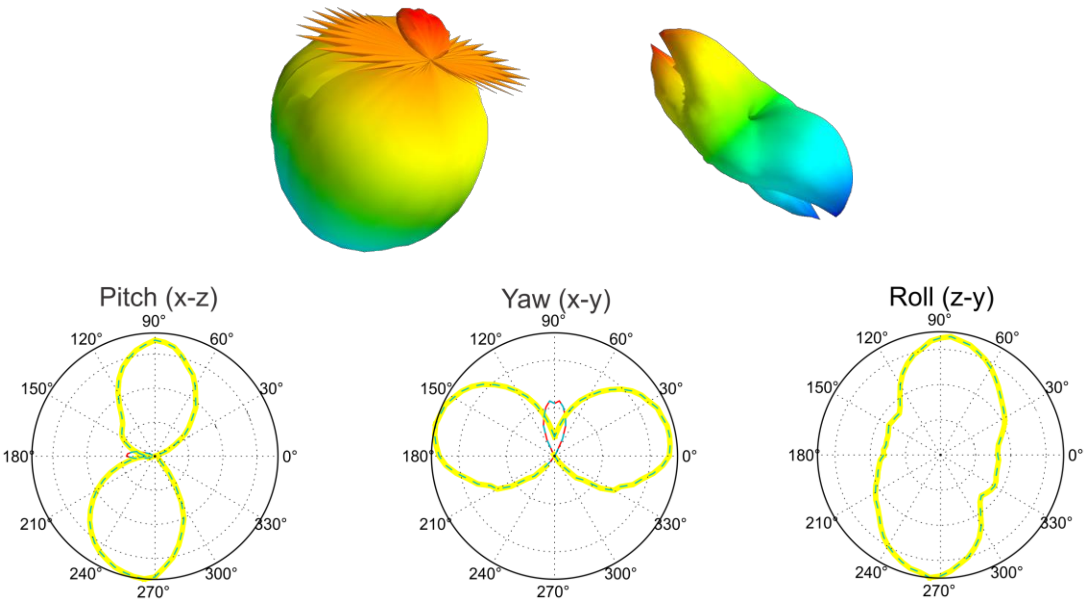


Figure 10: Spherical Plot of Intensity.

---

[1] Not to be confused with the spider/polar plots used to summarise the validation results.

5.7 Representation of Results

A convenient means to display the results is in the form of spider/polar plots, as shown in Figure 8. A polar plot has the useful property that the area ’within’ the curve is indicative of total strength or value, while linear plots carry the same information less effectively. If all the plots for a given model are made on the same graph, it is quite convenient to evaluate all the information at once. For example, Figure 8 shows a model with a good score across the ‘Geometric - Computer Model’ section, but a low score across the ‘Geometric - Reality’ section. This indicates poor information and documentation on, for example, field measurements or even poor field measurements itself. On the other hand, the computer model is well documented, and produces the expected output (as tested for example with regression tests). The analysis described here maps the model criteria along the first axis and the model instances along the second axis of a two-dimensional table. The polar plots can be grouped along any two axes, in the process highlighting different perspectives. When reviewing the results, the user should evaluate all the information in the polar plot in its entirety (i.e. the area inside the polar curve), in a valued judgement, considering risk and impact. It should never become a mechanistic numbers game.

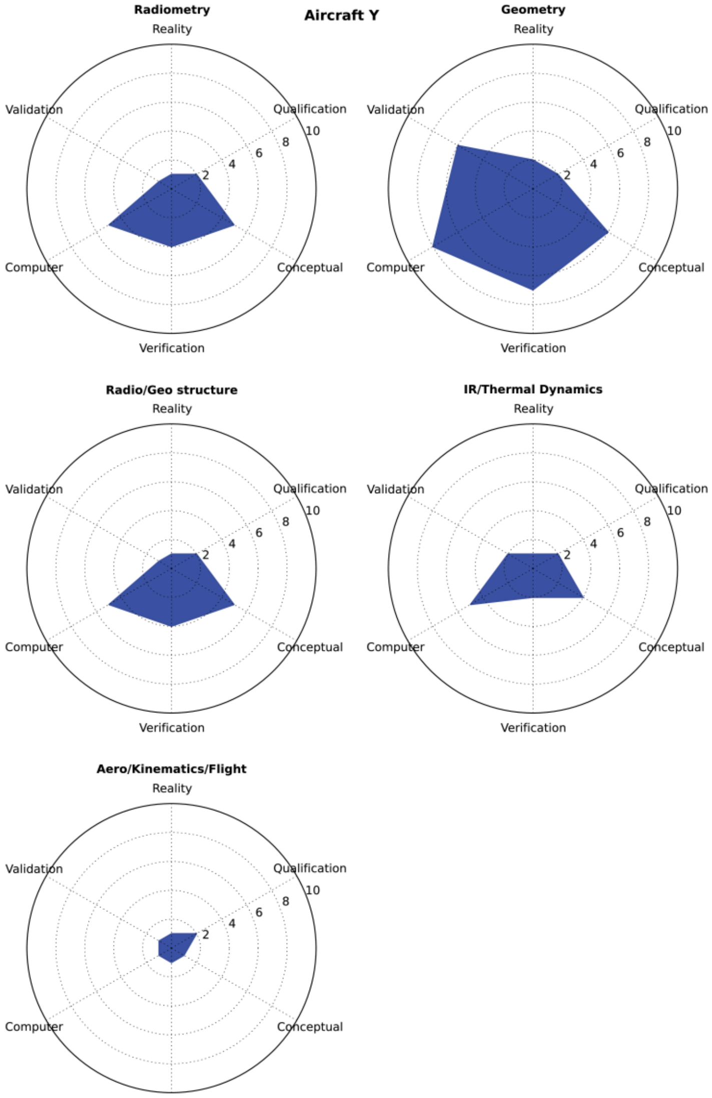


Figure 8: The Six Elements of the Validation Process Expressed with Spider Plots.

6 Lessons Learned

The validation procedure as reported in this article was applied to two models in the imaging infrared simulation system. The validation results of two models in the system were completely different with one model scoring very high points on most criteria and the other scoring very low. One of the most important requirements or/and deliverables from the validation and verification process are lessons learned from the validation criteria used to validate each system and subsystems. The lessons learned are used as input during the system updates and when new system is been developed. The section will present the lessons learned during the case study of two models, one developed using the data that is verified and documented properly and one will less information on how the data was captured, analysed and how the model was developed.

The following are some of the main lessons learned during the validation process of the two systems at two extremes:

- Strive to create the most accurate model, given the available information.
- Strive to model as close as possible (where relevant) to the underlying physics or other relevant real-world processes, instead of abstract behavioural models.
- Strive to repeat characterisation measurements (e.g. field trials), model updates and validation to improve the accuracy, scope and confidence in the model.
- Strive to over-document, rather than to under-document.
- A detail validation plan should be developed during the model development and implementation and the model parameters that needs to be validated should be identified.
- The model developer should provide and document extensive technical information of the model.
- Verification should be performed during the development of the simulation model. Note that model verification does not imply that the model is valid.
- Validation process should be planned and tailored such that the parameters used during validation process for each case is different even though the simulation system is the same.

7 Conclusions

In this article we build on the case made in previous articles that simulation is the only way to support the evaluation of hardware solutions cost-effectively. In order for the simulation to fulfil this role, one must have confidence in the system. Careful design and extrapolation of results leads to cost-effective and objective validation and verification which contributes to a high confidence in the simulation system. In this article we discuss the practicalities of a proposed validation process and elaborate on validation and verification techniques that can be used in the regular course of work. We also report on lessons learned.

One very important link is still lacking in this course of research: Although (Willers & Roodt 2011) made a strong case that validation can be done cost-effectively by carefully choosing critical points where validation must be performed, it is still chosen based on heuristic criteria. These criteria include (1) scenarios of most probable use, (2) scenarios of expected divergent or unstable behaviour, (3) scenarios with safety or cost driver implications, (4) scenarios of risk as identified by exhaustive Monte Carlo simulation and (5) scenarios identified by stakeholders. One possible direction of future research is to test these criteria more thoroughly to really understand its implications in choosing critical validation points.

Another very important condition for extrapolation of validation results to contribute to a high confidence in the simulation is the following as stated in (Willers & Roodt 2011): “Extrapolation is most accurate when conceptual and computer models reflect the underlying physics of the real world process.” This statement leaves quite a gap for research in the field of confidence and validation/verification for simulation systems focusing on emergent behaviour (agent based) and complex non-linear behaviour.

8 References

Forsberg, Kevin and Mooz, Harold, “The relationship of system engineering to the project cycle,” 1991.

INCOSE, “Systems engineering vision 2020,” 2007.

Irobi, Ijeoma Sandra, Irobi, Ra, Andersson, Johan and Wall, Anders, “Correctness criteria for models' validation - a philosophical perspective,” Tech. Rep., Departmentof Computer science & Computer Engineering (IDT), Maalardalen University, P.O. Box 883 721 23, Vasteras, Sweden., 2001.

Martis, M.S., “Validation of simulation based models: A theoretical outlook,” *The Electronic Journal of Business Research Methods*, vol. 4, no. 1, pp. 39–46, 2006.

Oreskes, N., “Verification, validation, and confirmation of numerical models in the earth sciences,” *Science*, vol. 263, no. 5147, pp. 641 – 646, February 1994.

Sargent, Robert G., “Validation and verification of simulation models,” in *Proceedings of the 1999 Winter Simulation Conference*, 1999, pp. 39–48.

Schlesinger, S, “SCS technical committee on model credibility: Terminology for model credibility,” *Simulation*, vol. 32, pp. 103-104, 1979.

Willers, C.J and Roodt, J. H. S., “Confidence estimation in the application of simulation in the development of aircraft self-protection measures,” in *Saudi International Electronics, Communications and Photonics Conference (SIECPC)*. IEEE Xplore, 2011.

Willers, C.J and Wheeler, M. S., “The validation of models in an imaging infrared simulation,” in *IEEE International Microwave and Optoelectronics Conference (IMOC)*. 2007, IEEE.

Willers, C.J, Willers, M.S, Santos, R.A.T, Van der Merwe, P.J, Calitz, J.J, de Waal, A. and Mudau, A.E., “Pyradi: an open-source toolkit for infrared calculation and data processing,” in *SPIE Proceedings Vol 8543, Security+Defence 2012, Technologies for Optical Countermeasures*, Edinburgh, 24-27 September 2012.

9 Biography

Alta de Waal is a senior researcher and developer in the Defence, Peace, Safety and Security (DPSS) Business Unit at the CSIR, South Africa. She is part of a team responsible for the design and development of an optronics images simulation software (OSSIM). She is responsible for the development and implementation of mathematical models for the software. Furthermore she manages customised versions of the software for clients. This involves design, development, implementation and maintenance of the software as well as training.

Alta holds a PhD in Engineering Science (North West University) and a Masters degree in Mathematical Statistics (University of the Free State).

Cornelius J. (Nelis) Willers completed a B.Eng (Honns) Electronics Engineering degree at the University of Pretoria in 1976 and an MS (Optical Engineering) degree at the University of Arizona in 1983. He is registered as a professional engineer. His 36 years of work experience includes electro-optical system development, system architecture and systems engineering, software development, and infrared scene simulation. His most notable achievements include being the chief architect and technical lead in establishing an imaging missile seeker technology base, and in the process, spearheading advanced physics-based infrared image simulation. The simulation system is currently used for a number of different applications in laboratories across the globe. His current interests include infrared signature measurement and data analysis, infrared system modeling and simulation, and the development of aircraft self-protection systems. He is leading the open-source, Python-based *pyradi* radiometry toolkit project. He has published a large number of technical and research reports. He also published a 500-

page textbook "*Electro-Optical System Analysis and Design: A Radiometry Perspective*", SPIE Press, Washington, USA, April 2013. His conference paper topics include infrared system modeling and simulation, and the modeling of military conflict using agent-based techniques. He teaches radiometry and infrared system design in short courses and at a masters-degree level at the University of Pretoria.

Dr Henk Roodt is based in Dunedin, New Zealand. His company delivers technology program establishment and management services. This is built on his experience of system design engineering and analysis of complex problem spaces. As a scientist and engineer he has extensive experience in dealing with international clients and developing big business proposals. He has managed large research teams and is a recognized modeling and simulation expert. He is currently employed by the Waikato Institute of Technology to establish an industry focused transdisciplinary research facility in industrial design, design engineering and process simulation in the Waikato region of New Zealand. Henk holds a PhD in Engineering Science and a Masters degree in Physics. He is a member of Simulation Australia, the Institute of IT Professionals of New Zealand, the New Zealand Institute of Healthcare Engineering, and the International Council on Systems Engineering.

Azwitamisi (Tami) Mudau is a researcher at the CSIR and he is busy with his PhD studies at the University of Pretoria. He received his B.Sc (Honours) and M.Sc degree in Physics from the Nelson Mandela Metropolitan University and B.Sc in Physics and mathematics from the Universirty of Venda. After his M.Sc degree he worked at Telkom SA as Product Developer before joining the Optronic Sensor Systems Competency Area of the Defence, Peace, Safety and Security research unit of the CSIR. His research interests include optical fibre, model validation, satellite calibration, air pollution, meteorology, infrared measurements and spectroscopy.