World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:16, No:11, 2022

# Intelligent Aid-Analysis Based on the Use of Digital Twin: Application to Electronic Warfare System

L. Chaussy, M. Nouvel

Open Science Index, Aerospace and Mechanical Engineering Vol:16, No:11, 2022 publications.waset.org/10012793.pdf

***Abstract*****—**Workload of the system engineers during Integration Validation Verification process of Electronic Warfare Systems (EWS) is growing with complexity of the systems and with the diversity of tested cases (diversity of operational scenario in front of EWS). Even if the use of Digital Twin makes easier conception and development phases in term of planning and test equipment availability, time to analyze tests results is still too long and too complex. The idea to reduce the system engineer's workload and improve test coverage is to introduce some intelligent and aid-analysis algorithms to improve this step.

***Keywords*****—**Analysis tools, automatic testing, digital twin, electronic warfare system.

## I. Introduction

MODERN combat aircraft and warships all include a EWS to get electromagnetic environment via detectors and to provide self-protection (jamming, decoy, etc.).

EWS in military aviation first appear during the Second World War. Since that time, they have improved a lot and they integrated more and more functionalities [1]. Over the years, software components of EWS have become increasingly important and play today a central place.

Today, EWS is composed of many sub-systems and defined by more than one thousand high-level requirements. Usually, an EWS is composed by multiple detectors and effectors managed by a software core. Software in EWS makes data fusion from sensors and generates effects performed by the effectors.

EWS are very complex systems on which not everything can be tested on real equipment (in flight, on sea or on the ground). Flight tests are complex to implement and very expensive [5]. Testing on test bench requires complex and expensive facilities to use and maintain.

Simulations is the best way to perform test and bench EWS software, it can improve robustness [3] and simulate scenario impossible to play in the real life. Simulations integrate many models (electro-magnetic propagation, antennas, digital processing, etc. [2]) allowing software under test to be stimulated with realistic inputs data.

Usually, simulation is realized in three parts:

- Generation or chose a scenario (action to perform during the test),
- Execution of the test,
- Analyze of the result.

## II. Digital Twin Classification

A digital twin is a virtual representation that serves as the real-time digital counterpart of a physical object or process. Usually, EWS digital twin is composed by:

- the real software (under test),
- a simulation of electromagnetic environment (treat, natural effect like cloud, …)
- a simulation of the platform hardware (plane or ship)

In our simulation testing approach, we classify digital twins into three categories:

1. "Real" digital twin including some of the real software component used by the system.
2. "Perfect" digital twin including perfect software component (which cheat using context information from the scenario). This type of digital twin is an ideal system achieving maximum theoretical performance.
3. "Performance" digital twin including software components, which respect or following the expected performance. This is a digital twin of initial specification. It differs from the "Perfect" digital twin because it is limited to the commercial performances and is by nature a subset of the maximal performances.

## III. Existing Digital Twin Usage

In order to test, validate, even qualify EWS software, system engineers use functional digital twin of the real sensors that integrates the real-time software components (subsystem-level testing described in [4]). Digital twins are used to massively test a large number of scenarios or use cases.

In Fig. 3, we show the global workflow of the IVV.

- A scenario (position of treat, platform trajectory…) is created to perform the test.
- This scenario is used during execution by the digital twin to perform the test.
- Integration Verification and Validation (IVV) engineer analyses with multiple tools output of the execution step.

In the last step, the system engineers compare software component result to input data in order to verify the quality of the global result regarding requirements. Moreover, these processes are more adapted to analyze global operational system performances and not necessarily the fine performances of algorithms.

The major difficulties to increase the number of tests made in simulation are the time-consuming aspect of the tests results

To simplify the message delivered by this document, illustrations show a very simple software component chain.

L. Chaussy and M. Nouvel are with Thales Defence Mission Systems, 78851 Elancourt France (e-mail: laurent.chaussy@fr.thalesgroup.com, myriam.nouvel@fr.thalesgroup.com).

analysis.

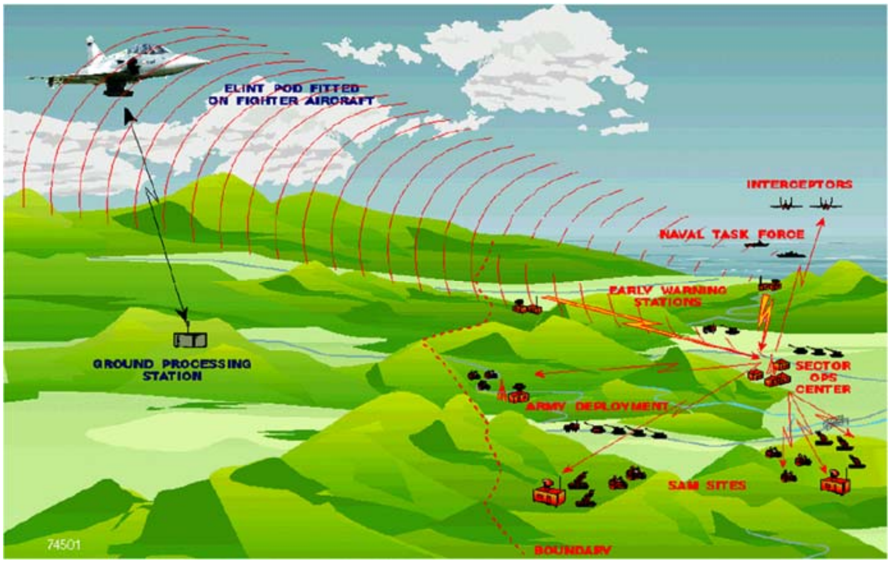


Fig. 1 Electronic Warfare on battlefield

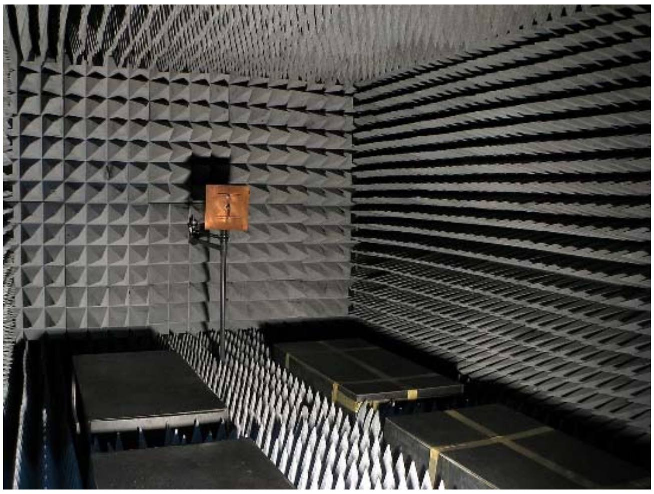

Fig. 2 Electromagnetic anechoic chamber

When identifying a problem, system engineers usually perform the following actions:

1. Identification and research of scenario data that led to the problem
2. Establishment of expected behavior
3. Study of the overall context to judge whether the observed problem is a defect in the software, or an intrinsic limitation in the system.

One of the limiting factors is the human time required to do all these analyses (which can reach many days for a complex case). Example of such a complex analysis: At a given moment in the scenario, the output quality of data fusion algorithm is not that expected by its specifications but at this precise moment, the airplane carrying the EWS is performing a rapid maneuver, which disturbs the measurements of the sensors. The degradation of performance should therefore not be considered as a defect at this precise moment of the scenario. To overcome this limitation, the idea is to introduce a better and more important aid thanks to the simulation to accelerate the analysis. The first help is to reduce research of scenario's input data as much as possible. The second help is to offer a dynamical establishment of the expected behavior by a second simulation. The third help is to introduce an intelligent log comparison system.

Finally, to increase coverage of non-regression tests we will try to automate the whole testing process.

## IV. Improvements Done on the Digital Twin

From this existing functional digital twin (a native simulation) of EWS, the process implemented is (cf. Fig. 4):

1. Introduction of tags in the simulation and in the software component in order to track the origin of the data: Those tags are introduced by the scenario generator and are disseminated in simulation software components, algorithms, software under test's component and outputs. N.B.: This additional information may go through processing related to the nature of the algorithms passed through (for example, it goes through fusion processing when it passes through a fusion algorithm).
2. Introduction of a theoretical (and perfect) software component chain composed of software components that are "cheating" by using the tags: The perfect software component chain is running at the same time as the "real" software component chain.
3. Introduction of intelligent comparison logs using the contextual information's of tags.

At the beginning of the test process, introduction of the perfect software component chain allows to simulate a perfect system and to obtain the reference data at each point. Those intermediate data are in the same format than those of the "real" software component chain.

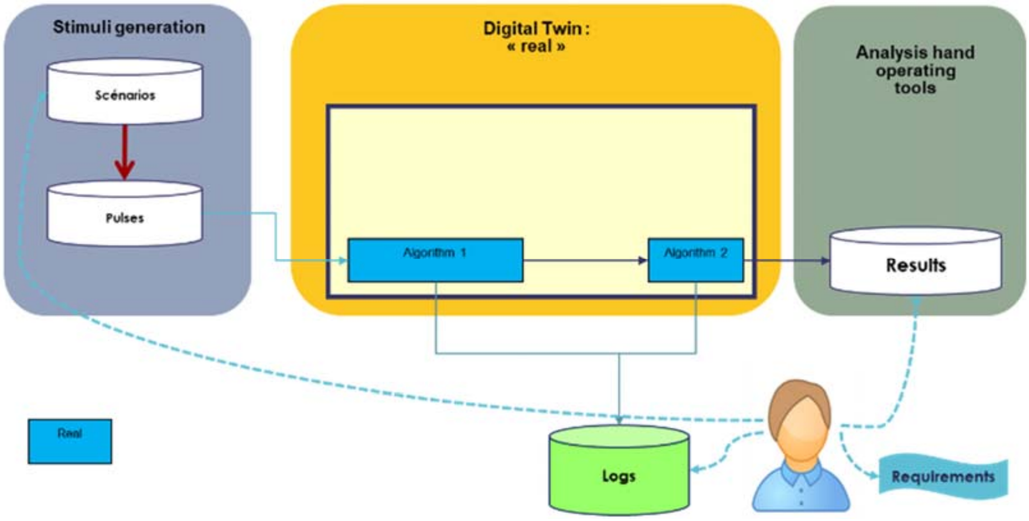


Fig. 3 Classic Digital Twin Usage (very long analysis time)

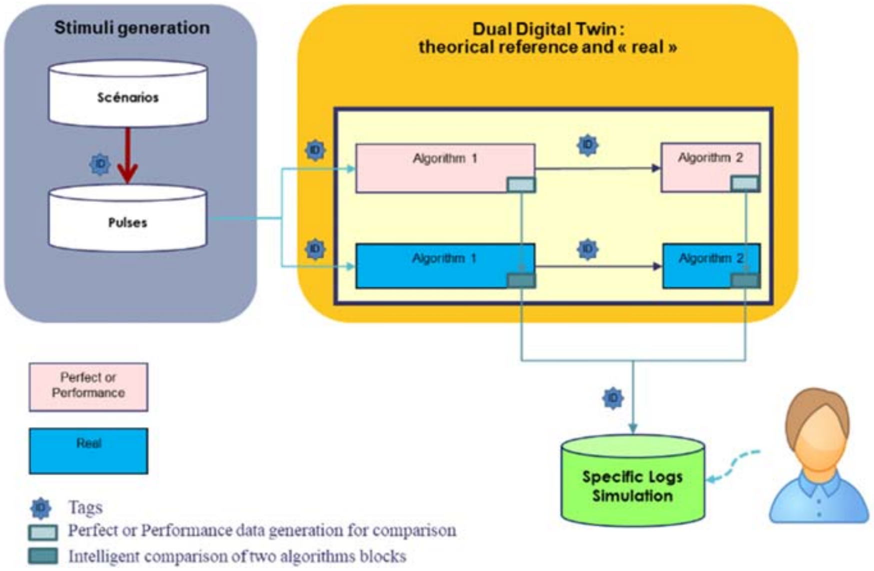


Fig. 4 Dual Digital Twin Real / Perfect (Shorter analysis time)

The simulation is built with two functional digital twins, one integrating the chain of real software components and the other integrating the chain of theoretical software components. This is a kind of dual simulation. From this dual simulation, intelligent logs, using the data from the two software component chains, are introduced in order to track and indicate the differences with the associated context.

By contrast with classical logs, that only represent internal data already existing of a software component, those intelligent logs are composed with comparison and complex analyze mechanism. It enables an automatic and pertinent report of defaults. They are part of the analytical work that engineers do.

The example of complex analysis carried out by the engineers on data fusion algorithm is now carried out by intelligent logs. It is a huge time saver in the whole analysis process.

## V. Usage of the Dual Digital Twin Approach in Automated Testing

The classic approach (Fig. 5) when performing automated tests (regardless of the nature of the test) is to stimulate the component under test and to compare the result produced with an expected result. The expected result depends on the stimulus and the specifications of the component under test. The implementation of this expected result is a human job that can be quite complicated and time-consuming.

Because of the time required, keeping this approach is a major obstacle to increasing the number of tests.

We took the approach described in Fig. 4 by replacing the perfect digital twin with a performance digital twin (Fig. 6). The dual digital twin approach adopted here replaces all the expected results (one per test) with a single performance digital twin and removes the main obstacle to the multiplication of tests. With this approach, the performance digital twin calculates the expected results dynamically at each test. With this approach, adding a new test is limited to creating a new input scenario. We save the time to create the expected results. It is a very important time saving in the whole process.

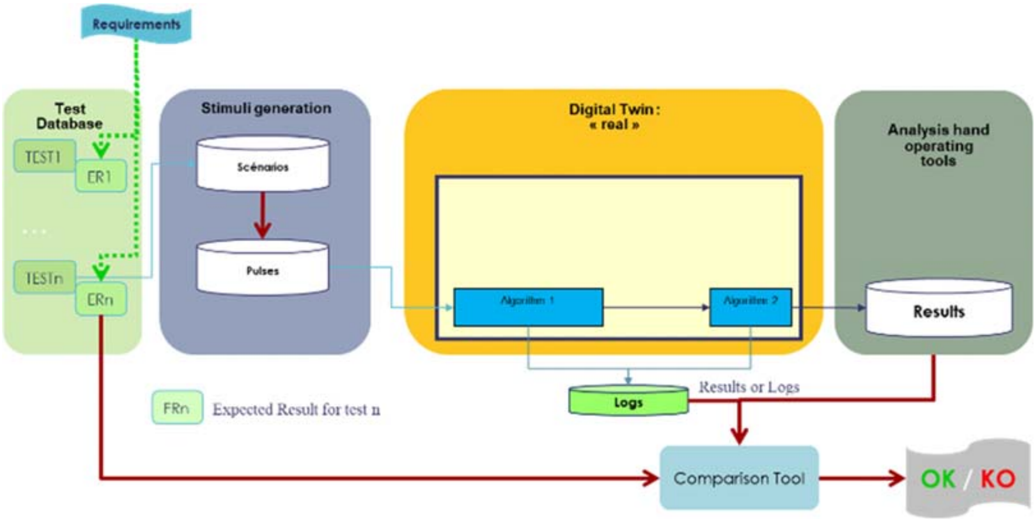


Fig. 5 Classic Digital Twin usage in automated testing

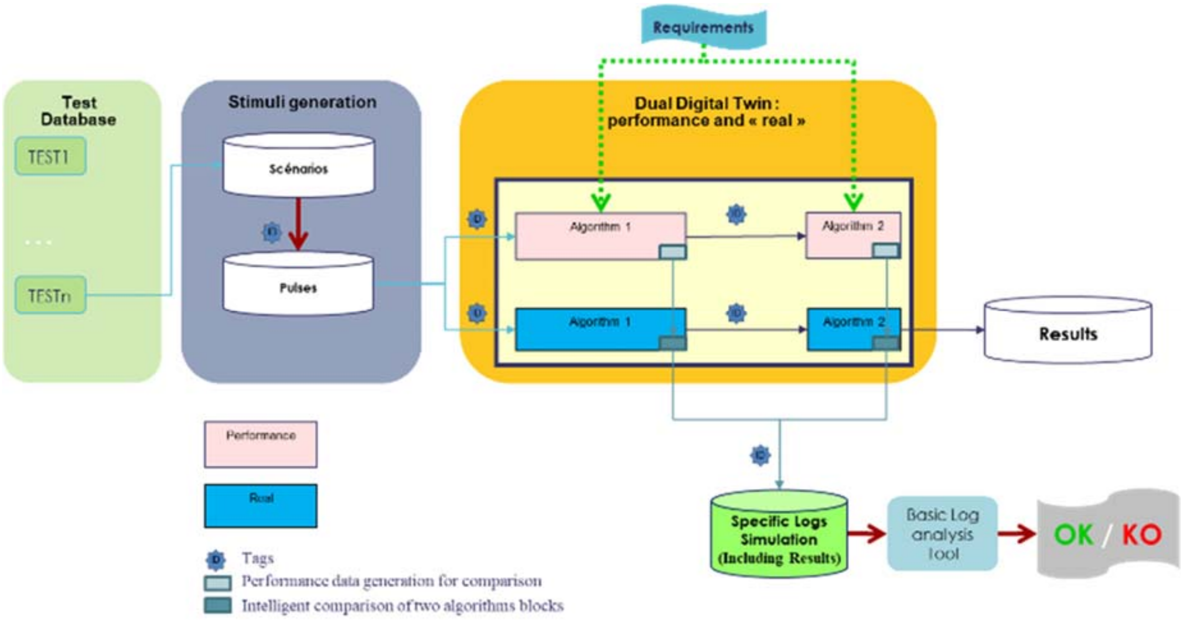


Fig. 6 Dual Digital Twin Real / Performance in automated testing

## VI. Pragmatic Approach

Depending on the nature of the system, the algorithms, the level of detail of the specifications… it can be tricky to achieve a performance digital twin.

A pragmatic approach (Fig. 7) is to make a perfect digital twin whose realization is much more obvious and simpler and then to constitute a comparison function that takes into account the difference between the perfect software component and the specified one.

The easiest way to implement dual digital twin solutions is to use a simulation framework that allows having a lot of flexibility to constitute a simulation integrating the two digital twins to be compared. To validate EWS, we use a proprietary simulation engine that allows us to easily synchronize all the models and software present in the dual digital twin.

## VII. Conclusion

We introduced three general concepts in our EWS digital twin:

1. Tagging of the input data of the system and propagation throughout the system,
2. Confrontation of several digital twins of the same system,
3. Replacement of the human defined expected result for each test by a performance digital twin in automated testing.

The results obtained during this preliminary study are very encouraging and promising to enhance testability of complex embedded equipment like EWS.

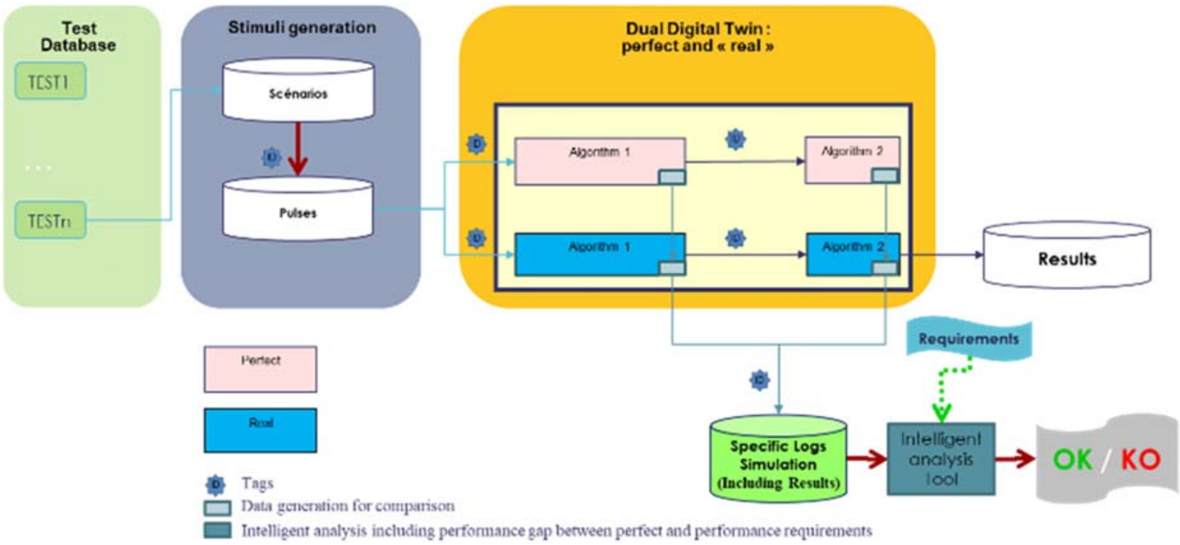


Fig. 7 Dual Digital Twin Real/Perfect in automated testing

On the system tested, this approach allows to significantly reduce analysis time (searching for contextual data is almost reduced to zero) and to quickly and easily add a large number of tests (digital twin calculates the expected results dynamically at each test). The time to create a new test is almost reduced to the time of creating and adding a new scenario in the test database.

In addition to the advantages mentioned above, it is often observed that the time devoted to test results analysis is often the cause of non-repeat tests or even their abandonment. Replacing the expected result by a digital twin is quite disruptive in the way of writing tests and requires a little time of appropriation. This technique is particularly suitable for systems having a great diversity in the variability of the inputs.

## References

[1] Mr. Martin Welch and Mr. Mike Pywell, *Electronic Warfare Test and Evaluation*, RTO of NATO AGARDograph 300 Flight Test Technique Series – Volume 28 December 2012.

[2] Electronic Warfare and Radar Systems Engineering Handbook (A Comprehensive Handbook for Electronic Warfare and Radar Systems Engineers) Naval Air Warfare Center Weapons Division Point Mugu, CA 93042 October 2013.

[3] Syed Muhammad Ali Shah, Birgitta Lindström, Daniel Sundmark, Sten F. Andler, *Robustness Testing of Embedded Software Systems: An Industrial Interview Study*, DOI: 10.1109/ACCESS.2016.2544951 January 2016.

[4] Bickford J, Van Bossuyt DL, Beery P, Pollman A. *Operationalizing digital twins through model-based systems engineering methods*. Systems Engineering. 2020; 1–27. https://doi.org/10.1002/sys.21559

[5] Timothy D. Westa and Mark Blackburnb, *Is Digital Thread/Digital Twin Affordable? A Systemic Assessment of the Cost of DoD's Latest Manhattan Project*. Presented at the Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.