

Concept of a Test Management System Based on Lessons Learned from ATV Test Campaigns

Workshop on Simulation for European Space Programmes (SESP)
24-26 March 2015

ESA-ESTEC, Noordwijk, The Netherlands

Johannes Ruetting⁽¹⁾, Michael Eimke⁽²⁾

⁽¹⁾Johannes Ruetting

Airbus DS GmbH Airbus-Allee 1, 28199 Bremen, Germany

Email: Johannes.Ruetting@astrium.eads.net

⁽²⁾Michael Eimke

Airbus DS GmbH Airbus-Allee 1, 28199 Bremen, Germany

Email: Michael.Eimke@astrium.eads.net

INTRODUCTION

This paper reports on experiences made with a prototype Test Management System used during the ATV test campaigns. The system had been developed by AIT users and provided useful and powerful means for planning, control and optimization of test activities. The lessons learned from this system are developed further and will be taken into account in the preparation and setup of the AIT campaign for the MPCV-ESM Project.

After a short introduction concerning the motivation for developing such a system to support the ATV integration, the paper will concentrate on the technical description of the system and provide some operational examples. In the conclusive part of this paper a retrospective on the lessons learnt from the usage of this prototype for future ground systems will be given. A re-use of such a Test Management System is foreseen in the MPCV-ESM ground facilities and for future EGS-CC based ground systems.

The Problem Statement: Need to access many different systems

AIT activities very often can be seen as a crucial point of every space project where all the pieces and parts of the project, all information, data and work in general that had been done during the previous years are coming together. Especially taking into account off-site or abroad activities, often under quite expansive conditions e.g. in rented test facilities, and tough schedules, resulting in a huge sensitivity of AIT phases regarding missing, not accessible or outdated information.

The ATV Integration and Test Team had been faced with the fact that the data and documents needed to perform the assembly and integration activities had been scattered over many different systems often coming with their own specific user interfaces, strengths and limitations; Figure 1 illustrates this situation:

- Documentation needs to be retrieved from external documentation systems without AIT/AIV context search and not always obvious where to find the latest valid specification or interface definition
- System Engineering data comes as data tables or database extracts without context links to the AIT/AIV environment
- System Requirements and related verification data (Verification Database) are hosted in separate systems without links to test procedure execution, functional unit under test or raised issues
- Issue Management and as-built configuration data is often held in external systems with limited search capabilities and no context link to actual AIT/AIV tasks
- The data generated by the ground systems (telemetry archives, front end recorded data, message and log files) are contained in dedicated files in different data stores and different formats without the possibility to correlate them easy to test procedure execution or functional units

In conclusion of this situation, many manual steps were required to put the data together for planning and execution of AIT/AIV tasks. This is inefficient, error-prone and offers very limited possibilities only for automation.

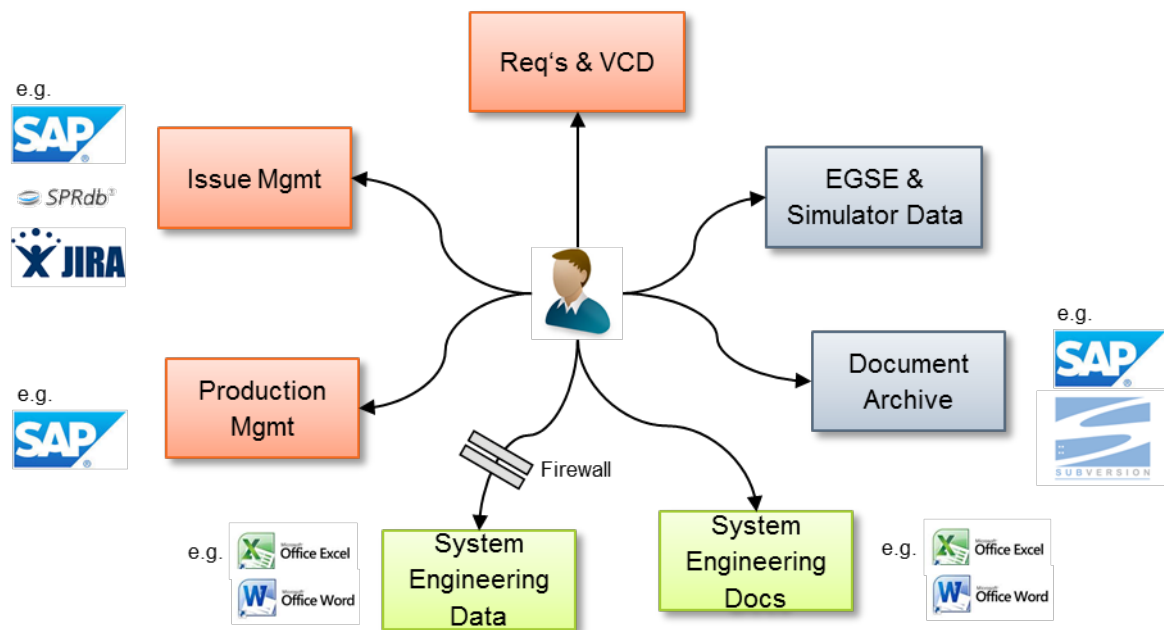


Figure 1: AIT/AIV Engineers with many dedicated and isolated user interfaces

The wanted solution

In order to improve the situation for the AIT and AIV teams a system supporting the functions listed below would be needed:

- Common Test Collaboration platform for the distributed AIT/AIV teams performing and supporting tests at various places in Europe, e.g. Bremen, Les Mureaux, Toulouse, Noordwijk and Kourou
- Efficient access to the latest ATV engineering documentation and data: Specifications, Design Descriptions, Interface Definitions, Harness and as-built configuration data, Issue Tracking system, etc.
- Test Management Tools to support test planning, preparation, execution, evaluation and reporting
- Common access to test execution systems and test results from different physical locations

The ATV Semantic Wiki Solution

In order to make a first step to improve the situation, a prototype of a semantic wiki had been developed and iteratively complemented with engineering and test data. This so-called "ATV Test Wiki" is shown in Figure 2 and contains the following components:

- The system is based on the "MediaWiki" open source software, known from Wikipedia, and extended by the "Semantic MediaWiki" extension in order to provide additional, semantic capabilities
- The user interface provides dedicated views derived from queries based on data, relations and roles
- File-based AIT data like test results are collected in a version controlled archive system; each data item automatically generates corresponding wiki pages/objects enriched by metadata (attributes and relations)
- External data like System documentation, Issues and Test Procedures are imported from other source systems, mapped to Wiki objects and also enriched with metadata (attributes and relations). For each object imported from external systems the location of the original information is kept as an additional attribute allowing the user to easily jump to the source system if required.

This system provides a unique access platform for the AIT and AIV users through views, forms and dashboards to all data. It allows search operations in the context of the integration tasks in all objects, without the need to individually log into documentation or issue tracking systems. Examples for test management functions supported by this system are provided on the next pages.

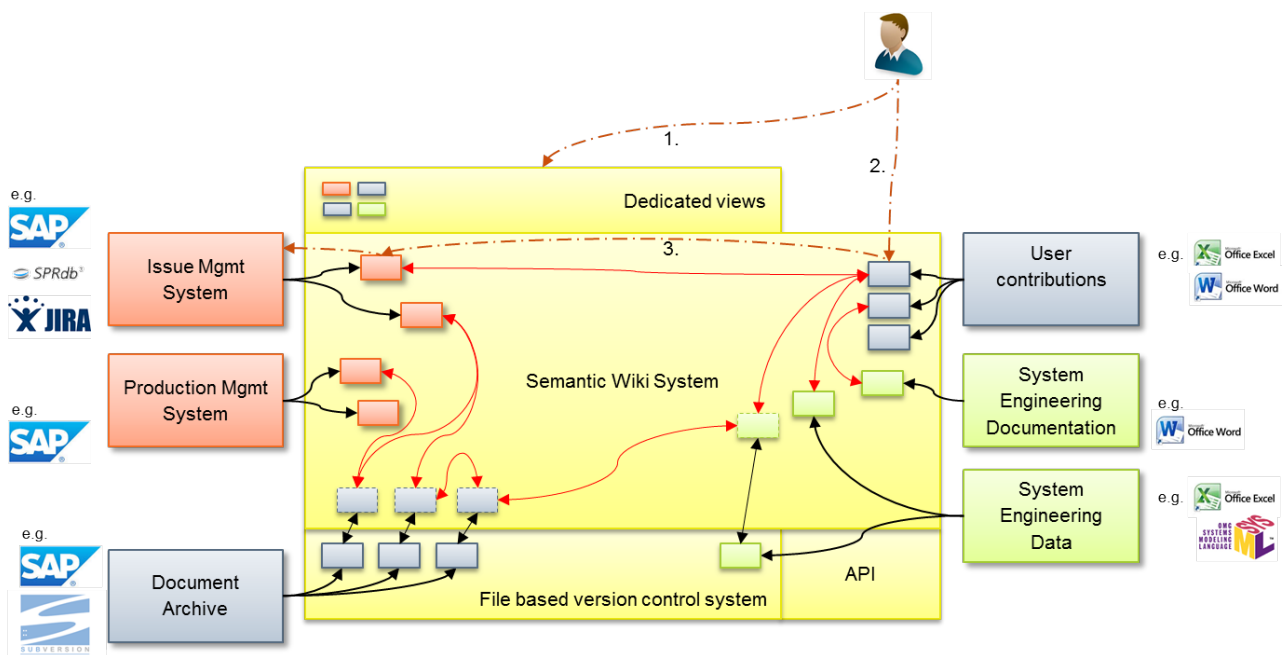


Figure 2: ATV Test Wiki concept

ATV TEST WIKI – SAMPLE FUNCTIONS

In the following some examples of major functions of the ATV Test Wiki are briefly described to give an idea of the provided capabilities.

Personal Dashboards

Each user has a unique dashboard generated based on the assigned roles of the particular user, e.g. „Functional Unit Responsible“. In the example the user is responsible for the GMS (GNC Measurement System) of Flight Model 4 (FM4) and the dashboard lists for this subsystem:

- The unfinished procedure executions for which he is responsible
- The latest 10 NCRs related to this subsystem (including status, procedure reference, etc.)
- The latest 10 procedure executions under his responsibility including execution run identifier, Flight Model, Operators, date, link to procedure, duration, status, and links to NCRs if applicable

Possible roles could be Functional Unit Manager, Test Responsible, Test Expert, Facility Responsible, STE Responsible, Test Operator, etc.

Flight Model focussed Dashboard

This Wiki page contains a summary view on all test procedure executions related to one Flight Model (selectable for ATV2, ATV3 and ATV4) and allows digging into any detail of the executed or planned test activities:

- Currently active procedures
- Procedure executions still in post processing
- Engineering post processing
- Last 15 procedures performed
- Upcoming test activities

Single Procedure Execution

Each Test Procedure execution is described by a dedicated wiki page containing information and links to:

- Participating team members
- date/time, duration
- raised problems
- Current status
- Attachments like scanned As-Run documentation, generated Test Results, pictures
- Automatically generated links to other relevant content, e.g. EGSE Test Session, Shift Report, NCRs, etc.

Shift Report

Each shift of test operators creates and maintains a dedicated wiki page containing additional remarks and annotations. It provides a very useful tool for shift hand-over, not only for the next shift responsible, but also for external contributors or observers. It also contains automatically generated links to other relevant content, e.g. raised problems, performed procedures, test sessions, etc.

Single Non Conformance report (NCR, QM)

Each NCR (QM) is mapped to a dedicated Wiki page to allow for search and link functions. Features are:

- Semi-automated data import from SAP
- Contains textual information and attached images / PDF files and supports thereby full-text search inside the NCR descriptive text and performed actions
- Automated linking, e.g. to Procedure Execution that caused that problem, Test Session, Shift Report, etc.

Online Test Session

This Wiki page contains a condensed view of Test Operator actions at the test console for each test session. A test session may contain more than one execution of test procedures. This Wiki page contains:

- Operator actions and system responses, operator annotations, screenshots, operator attachments, data tables generated by APs, etc.
- Automatically generated/updated every 10min from the different EGSE benches (from all locations that had been used during ATV-E (BRE, ESTEC, TLS, KRU)); this turned out to be a very efficient tool for remote test observers or test support by remote experts (e.g. EGSE support from Bremen online to the test execution personnel in Kourou)
- Automated linking to Procedure Executions, Shift Reports, etc.

Linked access to ATV System Engineering information

Wiki pages have been established containing the ATV functional breakdown (Product Tree, Engineering drawings, functional descriptions, Interface definitions, drawings and harness data up to connector and pin allocation). The content of these pages have been linked to

- Test Procedures
- Procedure Executions
- Test Record Sheets, etc.

This provides a powerful tool for the test teams (access to as-built engineering data in particular in case of trouble shooting activities) and for the engineering team during test analysis (access to test execution details and as-run details).

Especially the quick availability of detailed harness information, providing a full traceability of e.g. electrical signals through pins, connectors and cables, proved to be a very powerful tool for the investigation of occurred discrepancies directly on-floor during test execution.

STE Management

In addition to the System Engineering information, describing the to be tested product, also relevant information regarding the AIT environment and required tools had been introduced and managed, e.g. various types of Break-out boxes, Scopes and Connector Savers. Relating such items to e.g. Operational Procedures as “required items” or to As-Runs as “used items”, allowed very efficient test preparation and detailed traceability of problems related to STE tools.

Campaign Management

Test campaigns are defined within the system and may contain a set of Operational Procedures that have to be performed successfully in that frame. A dedicated dashboard for each campaign then provides detailed progress monitoring showing the status of all required test executions incl. occurred NCR's etc. All information generated during execution of Test Procedures in the frame of a test campaign may be exported into a data package, containing static versions of all relevant wiki pages as well as copies of all attached file resources, e.g. test result files acquired from the EGSE.

THE MEDIAWIKI TECHNOLOGY USED IN THE ATV TEST WIKI

As with every common wiki, every information inside ATV Test Wiki is organized as a number of wiki pages. Each page is identifiable by its unique name and may contain textual content as well as formatted rich-text and additional elements like tables, lists and images. The underlying MediaWiki wiki-engine supports both flat- and hierarchical organization of pages and subpages.

The “Semantic MediaWiki” extension implements a full set of semantic abilities on top of MediaWiki allowing the definition of semantic properties for each page. Each property defines a relationship between the embedding page and determinable values (Strings, Numbers, Dates, etc.) or other pages/objects. Semantic properties and values can be queried, processed and displayed using several pre-defined formats (HTML-Tables, XML, Excel, CSV, etc.) or custom templates. The integration of semantic queries into the common wiki content is achieved through several user-functions usable within every wiki page.

Using semantic properties, a data model can be established within the wiki system. Each “object”, e.g. an item from the product tree, is represented by a wiki page containing

- visual content (common wiki content, e.g. a description of the item with text, images and tables) and
- semantic properties (e.g. to express values like serial numbers, manufacturing dates, etc.)

Through the connection of objects using semantic relationships, complex networks can be built and queried in order to retrieve the desired information.

Some example applications for such queries in the domain of Functional Tests:

- Retrieve all NCRs related to a given subsystem
- Retrieve recursively all NCRs related to a given item (equipment, component, etc.) or a part of it (connector, pin, etc.)
- Identification of (flight-) connector pins verified by given test procedure executions

Semantic properties may be defined for each object separately or through “semantic templates”. The object page then only calls and fills the template while all the mapping of the values to semantic properties and the generation of visual content is done within the shared template. This approach also allows a very flexible management of the data model as a lot of modifications (e.g. renaming, segmenting or merging of properties) can already be achieved by editing the template and not necessarily each object. Based on semantic templates, forms may be developed simplifying the manual creation and modification of objects.

Integration of external information

One of the basic ideas behind the ATV Test Wiki had been to create “knowledge networks” in order to integrate information from external systems into a simplified data model forming a common “information space”. To avoid data redundancy as much as possible, not the complete information is retrieved and imported from the external source (e.g. a NCR from SAP), but only selected significant parts:

- Unique identifier of the information (e.g. NCR ID)
- Information needed for linking the information to other content (e.g. Flight Model that raised the NCR)
- Information that has to be accessed very quickly by the user and that shall be available through the wikis full-text search engine (e.g. the NCR descriptive text)
- The source location of the original information, preferably in a way that allows the user to directly browse to

The imported information then is stored inside the wiki represented by a dedicated wiki page containing visual information and semantic data. This page, describing the imported object, may then be used in order to establish relations to other objects present in the wiki, possibly imported from a different external data source.

All contained information may be queried either

- By the user through a dedicated query form,
- Through custom inline queries contained within other pages in order to e.g. display result tables etc.,
- Using a custom Web-API
- Using a standardized SPARQL-interface for semantic queries

Especially the SPARQL interface is expected to become extremely useful in order to correlate information from multiple instances of the system, e.g. for comparing information from different projects with similar, but not totally equal data models. In opposite to other approaches, this potentially improves the ability to query and harmonize information from different projects without necessarily enforcing them to use identical data models.

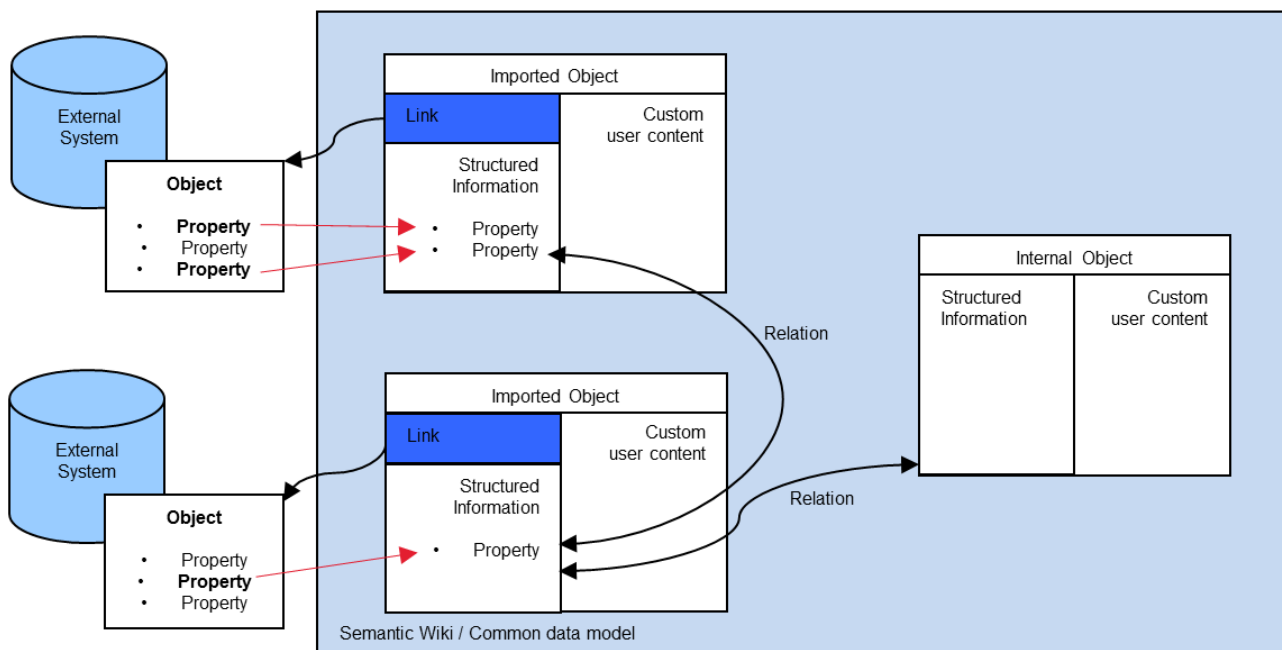


Figure 3: Wiki objects representing external information

LESSONS LEARNED

Due to the wide acceptance and well understood benefit of this system, it is planned to re-use such a system for MPCV-ESM ground facilities. Improvements and additional functionality is planned in the following areas:

- A procedure development tool based on a Domain Specific Language (DSL) for the development of manual and automated test procedures
- Integration with the Verification & Validation Management Tool (VVMT) to close the link between Test Specification and Execution
- Manage the Integration and V&V campaign with support of the Wiki as the collaboration platform for optimized access to engineering data, documentation, and test planning, execution and report generation.
- A unified and configuration controlled test data archive
- Interfaces with Verification Database and NCR System

CONCLUSION

Following the initial idea of the ATV Test Wiki and taking into account the extensions described above, a kind of collaboration platform for System Integration, Verification and Validation is being established. This platform could be used project independent following the high level architecture as shown in Figure 4.

The kernel of the project independent collaboration platform comprises:

- A semantic wiki system to manage the AIT/AIV data and tasks
- A DSL-based Procedure Editor to develop manual and automated test procedures taking full advantage of software development features
- A unified archive system to store and use data generated during test executions

Project specific data is imported from three models:

- Verification & Validation (V&V) Model providing requirements & test campaign data (VCD)
- System Model providing Spacecraft data as TM/TC definitions, product tree, ICD, harness
- AIT Model providing data of STE, breakout-boxes, test harness

The data processed in the collaboration platform is synchronized with external systems like:

- Production Management System for retrieval of production orders and ABCL data (e.g. from SAP systems)
- Document Archive to access baseline documentation as specifications, interface definitions and user manuals when needed in the AIT/AIV process

- Issue Management for retrieval of references and status information on NCRs or tickets during test preparation, execution and evaluation

Interfaces to different test execution platforms can be established to use data and products generated in this collaboration platform project independently. This offers high flexibility and significant cost savings:

- AIT and AIV products (e.g. test procedures) can be used on different target ground systems like CGS, OpenCenter or AITS, but also new EGS-CC-based facilities like ACCEPTO
- Re-use of AIT/AIV products, both vertical (within the same project) and horizontal (re-use of test artefacts between different projects)
- Savings in training and familiarization effort for AIT and AIV personnel

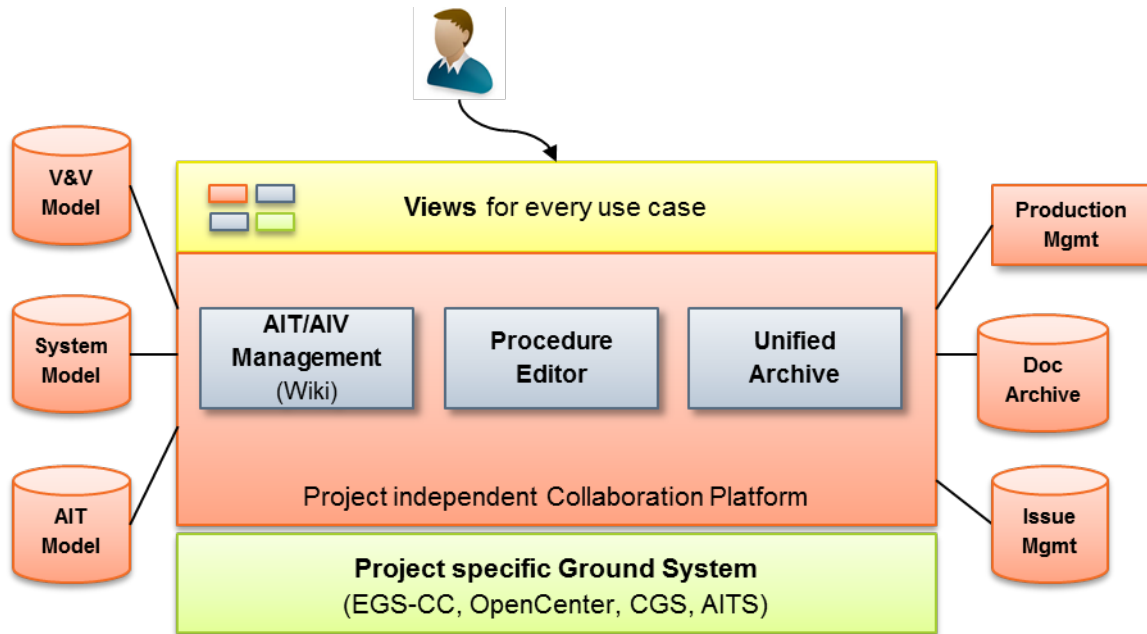


Figure 4: The ground facility independent AIT/AIV collaboration platform,