

# **TALENT – An advanced test analysis and evaluation framework**

**Workshop on Simulation and EGSE for Space Programmes (SESP)  
26 - 28 March 2019**

**ESA-ESTEC, Noordwijk, The Netherlands**

Bjoern Kircher<sup>(1)</sup>, Tobias Hoppe<sup>(2)</sup>

*<sup>(1)</sup>Airbus Defence and Space GmbH  
Claude Dornier Straße  
88090 Immenstaad  
Germany  
Email: bjoern.kircher@airbus.com*

*<sup>(2)</sup>Airbus Defence and Space GmbH  
Claude Dornier Straße  
88090 Immenstaad  
Germany  
Email: tobias.hoppe@airbus.com*

## **INTRODUCTION**

Test evaluation and detailed data analysis is a complex and time-consuming task. Not only is there a significant amount of data to be analyzed, but it is also becoming more difficult to make a correct interpretation of the data itself. When considering a complex AIT test bench which includes several SCOEs and simulators, an average duration test session is currently producing gigabytes of data. Within this session, not only are there hundreds of files to be analyzed, but typically they are stored in different formats and potentially not even synchronized to a common time base. This makes it already very cumbersome to sift through the data and generate a meaningful statement for the test evaluation. In addition to this flood of data, a lot of associated documentation is needed to interpret the data in the correct way.

TALENT – Test Analysis & Evaluation Tool – is a framework developed during the last year with the goal of improving and supporting the test analysis and evaluation activities during FV and AIT campaigns. The basic concept is to store all the data generated during a test session in a central “big data” system. However, this involves not only storing the data in a single place (acting as a kind of archive), but additionally the imported data is transformed from a tool specific format to a generic format in the data lake. This data transformation firstly enables the use of generic data viewers that are independent from the tooling that generated the data - but more importantly, having a common format for the data from several tools allows to visualize them together and better yet brings them into a common context.

Another crucial point that is addressed by storing all data in a big data system is the availability of data over the complete test campaign life-cycle. In the past, test session related data was typically stored only in the context of the test itself. Trying to compare several sessions, calculating overall statistics or analyzing data over more than one session was a common problem and not supported by the used tooling.

With TALENT, a common data lake integrating the data continuously over the life-cycle and from all the various test benches allows an overall data access independent of individual test sessions. And additionally it enables the possibility to calculate supplementary metadata presenting the actual state of the test campaigns. Additional metadata or overall statistics are used to compute, for example, TM/TC coverage or management of life-limited items in terms of overall on-time and number of power-cycles.

## **ARCHITECTURE**

TALENT is embedded in the existing overall tool landscape for functional verification (FV) execution and evaluation by providing a rich set of analysis functions for data post-processing, including issue detection and inspection, as well as report generation for subsequent post-processing tools. TALENT is implemented as a web application providing a

lightweight frontend running in a browser, and a powerful backend which communicates with the big data system and prepares the data to be shown in the front-end.

### Used Technologies and Tools

An overview about the general architecture is given in Fig. 1. All data generated during a test session including SCOEs data, simulators data, and of course data from the Central Checkout System (CCS) is imported into a data lake. This data lake is used by the web application to enable data access in real-time. Since the data is stored in raw format the System Reference Database (SRDB) is used to calibrate data according to the selected calibration baseline. Data can be browsed directly in the front-end, or exported in the form of reports for documentation purposes. Alternatively, data can be exported in the form of tool exchange formats to enable specific analysis in related tools such as Functional Verification Manager (FVM), Matlab, or MS Excel. Thanks to the inclusion of a Semantic Media Wiki (SMW) [1], documentation of the various data elements is available, providing supplementary information according to its related context or version. This SMW integration enables a direct navigation possibility between test data under investigation and the describing documentation data contained in the SMW. The goal of this close SMW integration is the avoidance to operate additional tools or handbooks in parallel while analyzing test data.

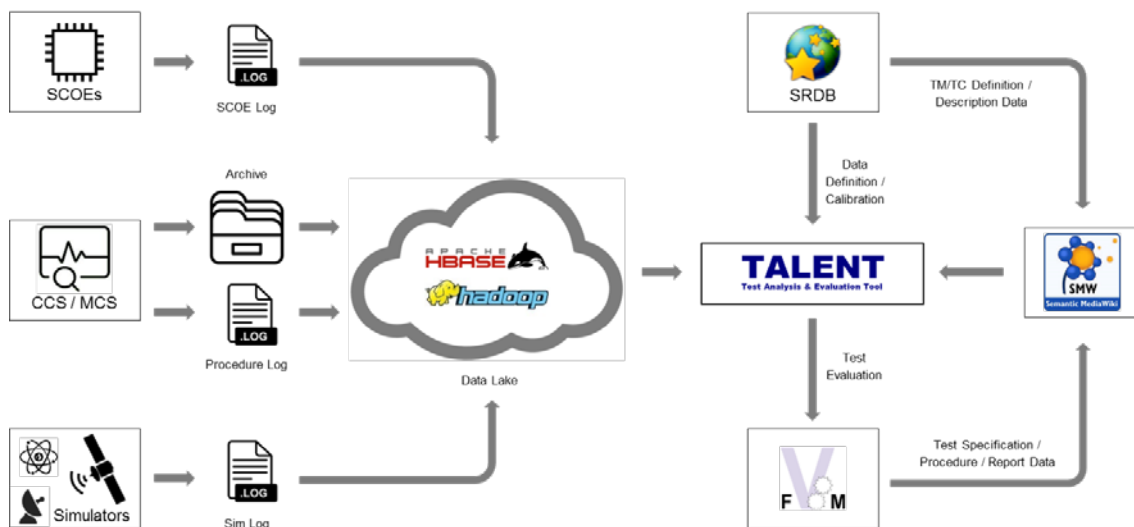


Fig. 1. Architecture

The data backend is realized as a data lake based on state-of-the-art big data technologies, enabling storage of massive amounts of data and providing fast access to queried data. Hadoop [2] has been selected as data storage system to enable more scalable data storage. The Hadoop internal file system, named the “Hadoop File System” (HDFS), stores files on multiple machines redundantly in a cluster to provide faster access by automated load balancing and data redundancy in case a data server is not available. In addition to HDFS, Hadoop also supports the use of data bases within the data lake. HBase [3] was chosen as the database technology in which to store most of the data, providing fast access and query options for the data. Furthermore, HBase utilizes the redundancy and workload balancing mechanisms of Hadoop to enable distributed and scalable data access in a multi-user environment.

HBase is structured like a classical relational database consisting of basic tables. Tables, and especially the so called column families, are represented as dedicated files in HDFS. Each column family consists of multiple columns which can be added at runtime. For any entry, represented by a row, the row key is the primary and mandatory value to uniquely identify this entry. All tables are sorted by these row keys. Instead of scanning through the complete table, if queried row keys are next to each other, data reading proceeds much faster. The format of this row key is therefore extremely important for fast data retrieval.

Further tools are linked to enable direct data access within TALENT. The source code for e.g. automated test procedures (ATPs) can be inspected directly in the web application, because this data is integrated from the corresponding project’s data repository containing the ATPs. If an executed data session has been selected, even the manually applied changes are included.

The TALENT web application also provides a Jira [4] interface to directly create observations based on the occurrence of suspicious data.

## **Data Structuring**

The used data structures have to provide fast data access possibilities to enable real-time data analysis in a big data environment. As mentioned in the previous subsection, HBase is used as backend database technology for TALENT. This requires a transformation of all input data into a generic data structure to benefit from the powerful data access performance of HBase.

The structure of the HBase tables was a subject of continual change during the TALENT development, as additional data analytics use cases required an updated structure of the tables to be performant. Moreover, the stored information evolved during the tool development to cover additional input data sources, e.g. to support different CCS frameworks or adding data from specific EGSE devices.

CCS related data such as TM/TC packets are imported into dedicated tables, e.g. dedicated PUS TC and PUS TM related table as well as CCSDS related table containing data from SCOEs. This separation allows faster access to the different types of packets which is especially essential for TC packet and TM packet analysis, because of the different filter and access needs of the end user. Furthermore, an ordering according to the time a packet was sent or received is mandatory to guarantee correct order of packet streams. For this reason, the time value becomes a significant part of the row key.

As different kinds of queries may need a specific structuring of data, the final data structure is always a compromise to support all possible needs. However, data duplication is to be discouraged due to the large amount of data that would be duplicated. For this reason, index tables are introduced to speed up queries by providing subsets of data items and HBase row keys. This enables the application to directly obtain the required full data set instead of scanning the table. An example use of an index table is the one used for TM/TC packet retrieval. This table contains one entry for each kind of TC and TM packet that has been sent in each session. This results in fewer rows than in the corresponding TM/TC tables – for example if a packet has been received 30,000 times, it has the same amount of entries in the TM packet table whereas in the index table it occurs only once.

The parameter values are stored only as raw values. The engineering values are calculated on the fly based on the selected SRDB version. This enables the use of different SRDB data versions to calibrate parameters and reduces the amount of saved data for parameters dramatically by roughly 50%.

## **ONLINE DATA ANALYSIS AND REPORT GENERATION CAPABILITIES**

The TALENT web-frontend provides an overview about all imported test sessions. As well as selecting a test session context, all the related data (e.g. FVM or used ATPs) can be browsed. This enables direct inspection of identified points of interest, such as occurred onboard events, together with the related data context. In addition, complete session analysis including all available data can be performed within the same frontend without caring about the different formats, related tools, or supplementary information.

### **Session-specific Data Analysis**

The selection of a CCS session in TALENT defines the time related context of data and the corresponding functions that can be executed. All data belonging to this context is linked to the selected session (see Fig. 2). A session overview page is generated per session providing a summary containing general information such as the session's name, the bench name, start and end time, the sent and received amount of packets and data as well as the used SRDB version. More importantly, an overview regarding occurred onboard events, TC acknowledges summary and the status of executed ATPs is shown to guide the user to the important data events of the session.

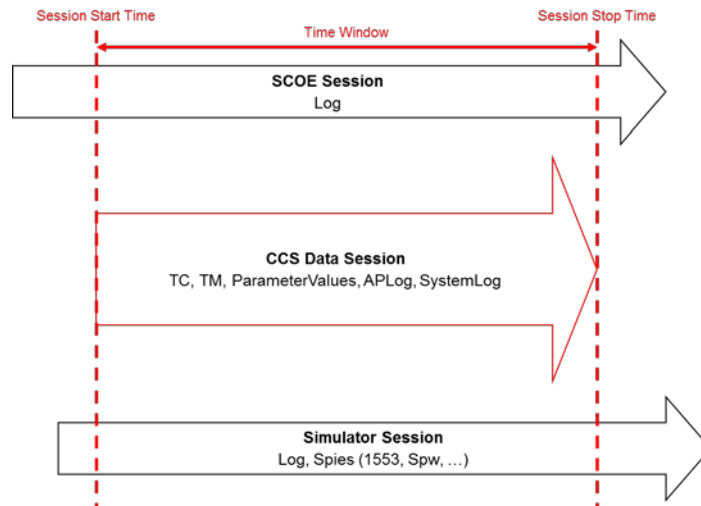


Fig. 2. Session Scope

From the session summary page (Fig. 3) the user can directly navigate to the dedicated views showing detailed information about and around the event. If a session includes for example a critical onboard event, the user can directly jump to the event view, providing all the details needed to analyze the event. In addition, for each event the details can be inspected by making the parameter values available in raw and engineering.

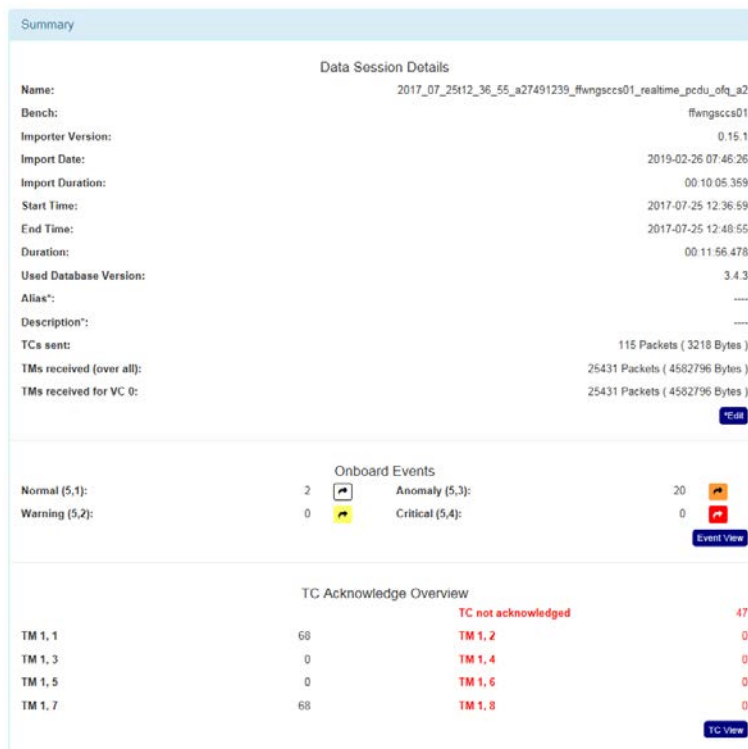


Fig. 3. Session Summary

To enable analysis of causal chains, observation of root causes or any other data of interest, the context view is provided. The user can select any data such as an onboard event in the event view in order to define the base context element for the context view. In this view all TM/TC, log entries, and simulator data are shown ordered by time. The timeframe around the base context element can then be adjusted by the user. Moreover, the shown data can be filtered according to different kinds, e.g. TCs, TM acknowledges, any other TM packets, high priority commands, simulator data, system log entries, etc. In the frame of the example shown below (Fig. 4) the user is interested in why the anomaly

onboard event occurred. Thus below TC packets and their corresponding acknowledge reports are enabled to determine if the event was a result of an incorrect command.

Timeframe	Local Time	Mission Time	Name	Description	SPID	APID	Type	Values
Start time: 2018-11-29 10:12:35.593	2018-11-29 10:12:48.920	2018-11-29 10:12:46.877	CSY_ACK20057	TCACK_ACC_SUCCESS	20057	161	1,1	08A1C0620011
	2018-11-29 10:12:48.920	2018-11-29 10:12:46.878	CSY_ACK20082	TcExeSucc	20082	161	1,7	08A1C0630011
End time: 2018-11-29 10:13:20.594	2018-11-29 10:12:50.415	2018-11-29 10:12:48.377	CSY_ACK20057	TCACK_ACC_SUCCESS	20057	161	1,1	08A1C0640011
	2018-11-29 10:12:50.574	2018-11-29 10:12:48.504	CSY_ACK20082	TcExeSucc	20082	161	1,7	08A1C0650011
Includes	2018-11-29 10:12:51.928		CSCEQ000	GetValOnBrdParam		172	140,2	18ACC03D000A
	2018-11-29 10:12:52.039	2018-11-29 10:12:50.002	CSY_ACK20057	TCACK_ACC_SUCCESS	20057	161	1,1	08A1C0660011
Events	2018-11-29 10:12:52.215	2018-11-29 10:12:50.130	CSY_ACK20082	TcExeSucc	20082	161	1,7	08A1C0670011
	2018-11-29 10:12:53.376		ASCFL022	Activate_SADE_function		188	161,6 1	18BCC00F0005
TCs	2018-11-29 10:12:53.574	2018-11-29 10:12:51.514	ASY_ACK30011	TcExeSucc	30011	177	1,1	08B1C01F0011
	2018-11-29 10:12:53.574	2018-11-29 10:12:51.510	ASY_ACK30014	TCACK_ACC_SUCCESS	30014	177	1,1	08B1C01E0011
Acknowledges	2018-11-29 10:13:00.594	2018-11-29 10:12:58.546	ASY_EVT30741	P_SADE1PosErr_AL	30741	183	5,3	08B7C0060085
TMs								
Include (3,X)								
Include (6,X)								
Simulation Log								

Fig. 4. Context view with anomaly onboard event, TCs, and acknowledges

In addition to this table-based context specific view, TALENT provides plotting functions to visualize the various numerical parameters. The user can directly navigate to it from e.g. the context view shown previously in Fig 4, and the values can be plotted over the duration of the selected session. Parts of it can be viewed in detail by using zoom and data inspection functions of the plotting view. The user can select the parameters that shall be shown together, for example to observe correlations. For TM parameters, not only raw values are shown, but also engineering values computed using the corresponding SRDB version. Occurred events can also be included in a plot as related aspects to enable correlation analysis of events and parameter value changes. Finally, a created plot configuration can be saved to apply it also on another data session. In Fig. 5 a correlation of different parameters is visualized in conjunction with an occurred event named CFY\_EVT25117 that has been chosen to be shown in the plot.

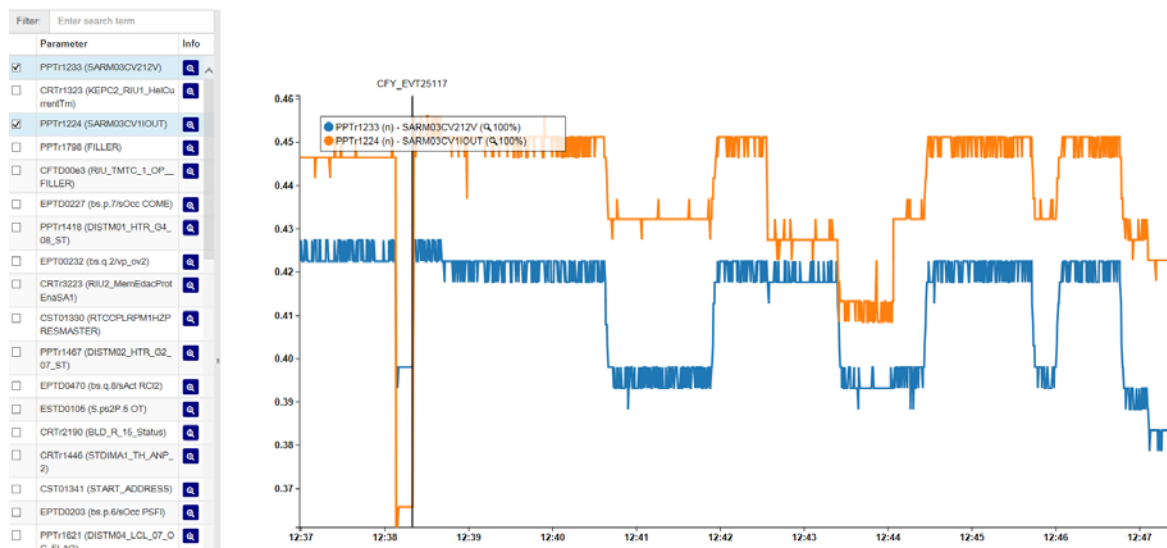


Fig. 5 Parameters visualization and onboard event correlation

Another start point for the data inspection is the ATP list view. It shows all the executed ATPs in hierarchical order. For each ATP a report with executed test steps in conjunction with the related data is generated. This report highlights successful and failed test steps and provides all information about sent TCs and received TM packets. In addition, the executed source code of each ATP is linked and can be browsed together with the generated report. Similar to all the other views, data from different sources is linked and it is possible to jump within the application e.g. from a step in the report to the ATP itself, from the code of the ATP to an executed TC or from the TC to the context view again.

Other views like a specific source sequence counter (SSC) view or one dedicated to checksum verification are added in addition to evaluate issues in detail.

## **Data Export**

In addition to the various data views contained in the web frontend TALENT also provides a set of export and report functions to extract and transform the data to specific formats. Data exports can be configured in terms of content (data to included), structure (columns and rows) and formats. Support export formats are MS Excel, Matlab, csv, JSON or plain text. This generic export functionality allows the user to configure their use case specific export format.

In addition to the generic export functionality, a large number of predefined exports are already available and can be executed directly. These pre-configured exports (e.g. parameter and packet) can be used instantaneously without the need of re-configuration. These predefined exports are mainly focused on numerical data; however a selection of summary reports is available as well. The available summaries include reports on onboard events, TC, or SCOE commands, as well as reports containing the before described context. Finally, summaries focusing on the high level test synopsis are also made available.

Exports and reports can be generated automatically based on events such as the import of a new test session or alternatively scheduled on predefined dates. All exports are available within the frontend or can be automatically pushed to a configured data share.

## **Statistics**

The goal of TALENT and its central data lake is to store data of the complete test campaign life-cycle. Thanks to this big archive of data it is obvious to use the data for life-cycle related statistical calculations.

### *TM/TC Coverage*

One example to use data across several test sessions is the calculation of TM and TC coverage. Since all sent TCs and resp. all received TMs are archived in the data lake together with the used version of the SRDB, it is possible to generate lists of used TCs and TMs per SRDB version. Comparing the list of used TM/TCs with the set of TCs and TMs specified in the SRDB, TALENT is able generate a basic TM and TC coverage report. Adding additional checks such as “are the TCs successfully accepted and successfully executed” and “were these TCs extracted from a formal test run” allows the generation of a formal report showing the TC coverage, with the direct link to the test session and the related procedures that were executed. But this statistic is not only used to generate the coverage report, it shows also the actual status of test coverage and the areas which are not processed in detail.

### *Life-limited Items*

Another very useful application of life-cycle related statistics is the management of life-limited items. Items or parts that are considered as life-limited items need to be monitored in detail, specifically checking that the foreseen usage of these items is not exceeded. Particularly power-cycles or on-times of these items are accumulated and need to be within the specified ranges.

Similar to the example above explaining the TC coverage, this monitoring function of life-limited items can be done thanks to the data in the data lake. The status TM or power consumptions of an item can be used to calculate the on-time whereas e.g. successful executed switch on commands can be used to calculate the number of power cycles. This calculation of data related to life-limited items depends of course on the knowledge which items are part of the test setup. This link is established by connecting the Hardware Software Configuration Manager (HSCM) of FVM with TALENT. The HSCM provides all details related to connected HW as well as SW version that are related to the test setup. The information about usage of life-limited items is used to generate a usage report covering the complete test campaign cycle as well as providing an always up-to-date overview on, for example, how many hours or how many power cycles can be used for testing.

### *Data Analytics*

Data analytics is mainly used to detect anomalies within the data that are not directly visible to the user or cannot be checked by classical verification stages or protocol checks. Data events that are different from “normal” behavior or occur very rarely should be identified and highlighted to the user. The data analytics part of TALENT is mainly based

on statistical methods to describe the data on a higher level. Basic statistical values like min, max, mean, standard deviation or variance as well as more complex functions to find e.g. mathematical outliers are computed for each parameter. These statistical values are used to provide an overview of the system but more important to compare with historical data and identify trends.

Another powerful statistic that is calculated is the correlation of data. The computed correlation matrix shows direct dependencies between the different data items and visualizes the expected but also not known relations between data. Disturbing systems can be identified by analyzing the data correlation. Monitoring the various correlation factors over the life-cycle is another key factor to identify trends.

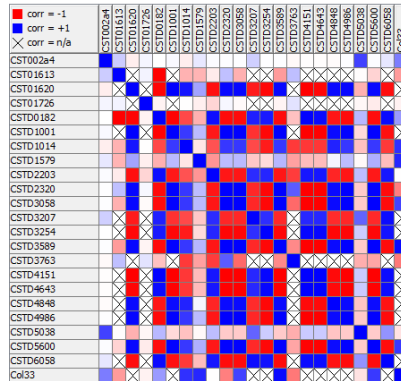


Fig. 6. Correlation Matrix

## CONCLUSION

The introduction of the Test Analysis & Evaluation Tool framework “TALENT” has shown the clear need of simplified and computer supported ways to analyze and evaluate data much easier. The possibility of browsing the data in a generic and tool independent way allows to customize the data view related to the problem to be investigated. Focus can be placed on relevant data, without being concerned where the data is from or in which format it was generated.

Additionally the introduction of the generic storage format in the data lake and the related interface in TALENT facilitates the usage of a harmonized tooling between the different disciplines or projects. Independent of particular project constraints in terms of tooling (e.g. used Central Checkout System, Simulators, SCOE, ...) the data analysis environment stays the same. Test reports, additional exports, external I/O interfaces and most important the user interfaces are identical between the different projects, and thus supporting the introduction of a standardized tooling infrastructure across different disciplines and projects.

TALENT’s user frontend is designed to provide simple and intuitive functions to analyze the data. Guiding the user through the data, providing summaries, and pointing out possible anomalies is one of TALENT’s key assets.

Already planned evolutions will focus more and more on automated evaluations using modern data analytics mechanisms, minimizing the need of detailed manual data analysis as much as possible. Additional test status and health information, provided by assessing historical data gathered during the development and test phases, will be used to identify repeating patterns, detected trends or highlight correlations.

Another possible next step would be to extend the context of data related to the development and verification phase and target the operational phases and the data related to these, too. Combining data of all phases allows early recognition of trends and thus enabling the mechanisms for automated defect detection.

In summary, the introduction of TALENT showed expansive advantages and process improvements in terms of data analysis and evaluation. The actual feature status today is just an intermediate step on the way to a fully digitalized and automated test evaluation process.

## REFERENCES

- [1] Semantic Media Wiki, Online: <http://www.semantic-mediawiki.org>
- [2] Apache Hadoop, Online: <https://hadoop.apache.org>
- [3] Apache HBase, Online: <https://hbase.apache.org>
- [4] Jira Software, Online: <https://de.atlassian.com/software/jira>