Paper AD05

Development of a Customised Statistical Computing Environment for Clinical Trial Data Analysis

Mario Lozina, Boehringer-Ingelheim GmbH & Co. KG, Biberach, Germany
Christoph Frank, HMS Analytical Software GmbH, Heidelberg, Germany
Darius Fischer, HMS Analytical Software GmbH, Heidelberg, Germany

ABSTRACT

This paper provides an overview of the Reporting Integrated Statistical Environment (RISE), a custom-developed, statistical computing environment within Boehringer Ingelheim. It explores the main features of RISE, its system architecture, and provides an outlook on future development.

INTRODUCTION

Statistical computing environments are an essential part of the clinical study submission process, and, as such, play a crucial role in the tool landscape of Boehringer Ingelheim. The Reporting Integrated Statistical Environment (RISE) is a fully custom-developed, cloud-based statistical computing environment. It provides a comprehensive set of tools to program analysis macros, validate them, organise the results into a report form, and generate the PDF reports for submission to regulatory agencies. In the following chapter, the main features of the RISE application are presented. Afterwards, the technical infrastructure and the release-process are described. Lastly, an outlook on future developments is given.

RISE WEB APP FEATURES

The following sections highlight the most important features of the RISE application:

PLANNING

The centrepiece of the RISE user interface (UI) is the Table of Contents (TOC), which is a tree-structure of chapters, sections, and figures/tables, organised in the same way that they will appear in the generated report (see Figure 1). Besides providing the basis for the report, the TOC also acts as a collaborative to-do list for a team of statistical programmers and statisticians, allowing users to fill in settings like the programmer, validator and validation method, before any program is assigned and executed.

Drag and drop functionality, as well as the ability to import from other TOCs, allows for convenient and efficient creation of a new TOC.

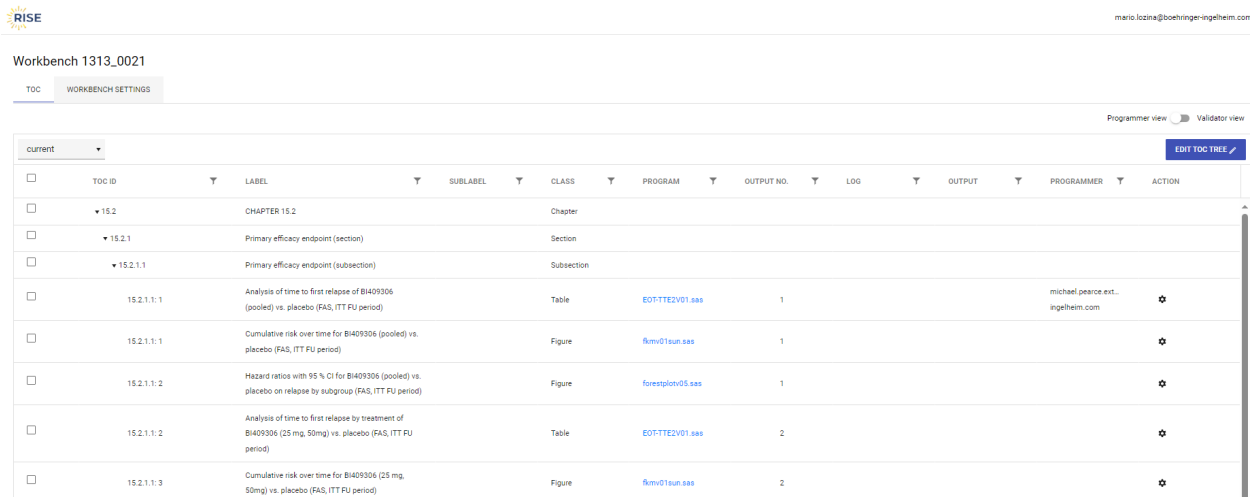

Figure 1: A TOC, expanded down to Tables and Figures.

ORGANIZING

RISE provides different views of the same TOC for programmers and validators, showing only the relevant information for each role. It also allows users to search and filter the TOC for their tasks.

DEVELOPING

Each TOC item can be associated with a main program, included programs (optionally) and a validation program. Clicking on the program name opens a session in the corresponding Integrated Development Environment (IDE) in a separate browser tab. SAS programs are already supported via SAS Studio (Figure 2) on SAS Viya, R programs will follow soon. Variables can be defined for a whole TOC or individual TOC item, both of which are made available in the IDE sessions.

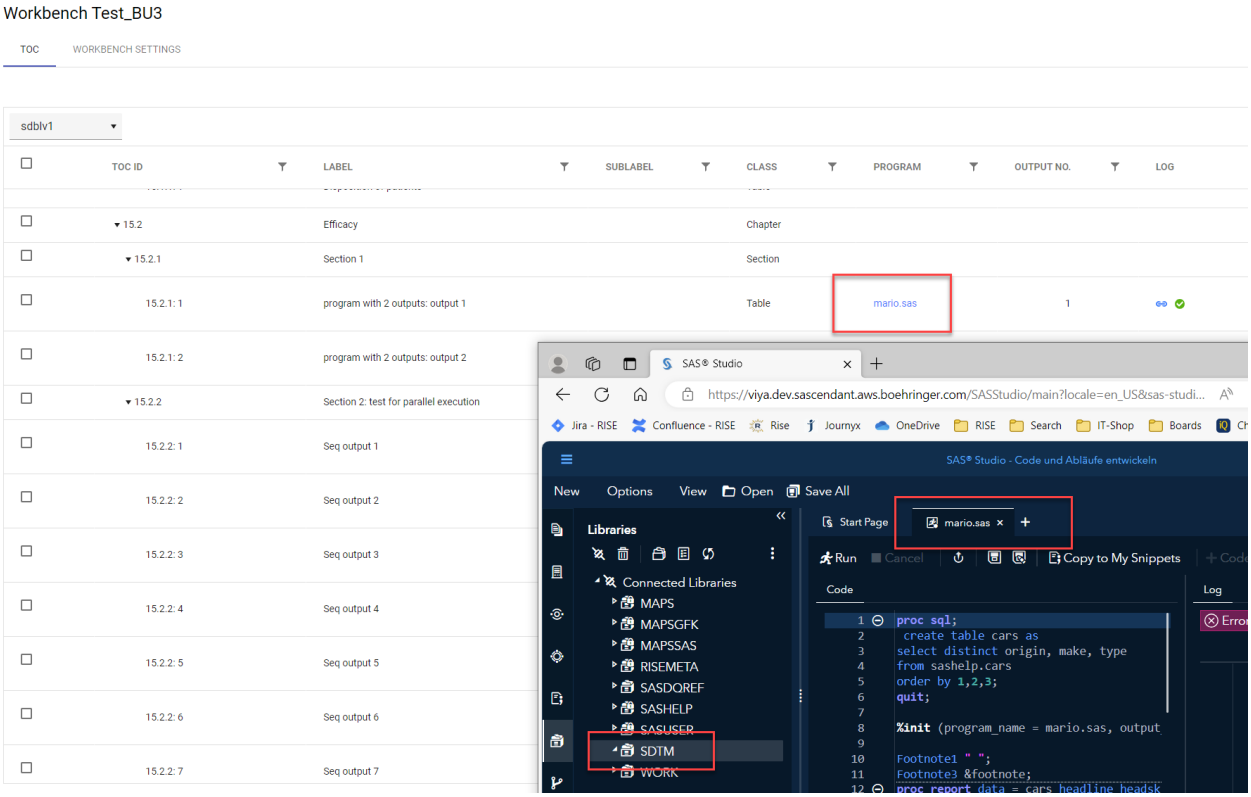

Figure 2: Opening a program in SAS Studio from RISE.

EXECUTING

Programs can be executed individually, or multiple programs selected and executed in batch mode. Users can choose between sequential and parallel execution. In the future, we will also allow scheduling runs.

The current status of execution, as well as the result, is visible to all team members (Figure 3).

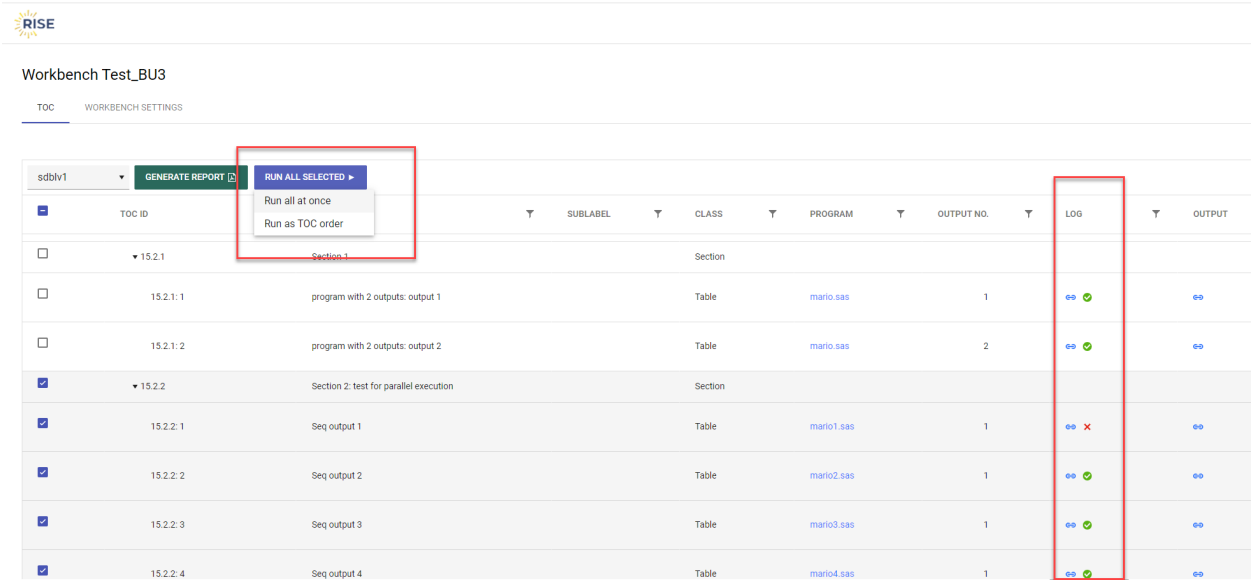

Figure 3: Program execution

OUTPUT QUALITY CONTROL

The validation process of programs is implemented as a series of stages (column *Status* in Figure 4), at the start of which a read-only copy of the program is created and a version number assigned. Previous versions can be retrieved from the *Version History*. When running programs, users can decide which program version to execute. When a program is validated, that version's metadata can no longer be changed. There are different validation methods available to choose from depending on the criticality and complexity of the program.

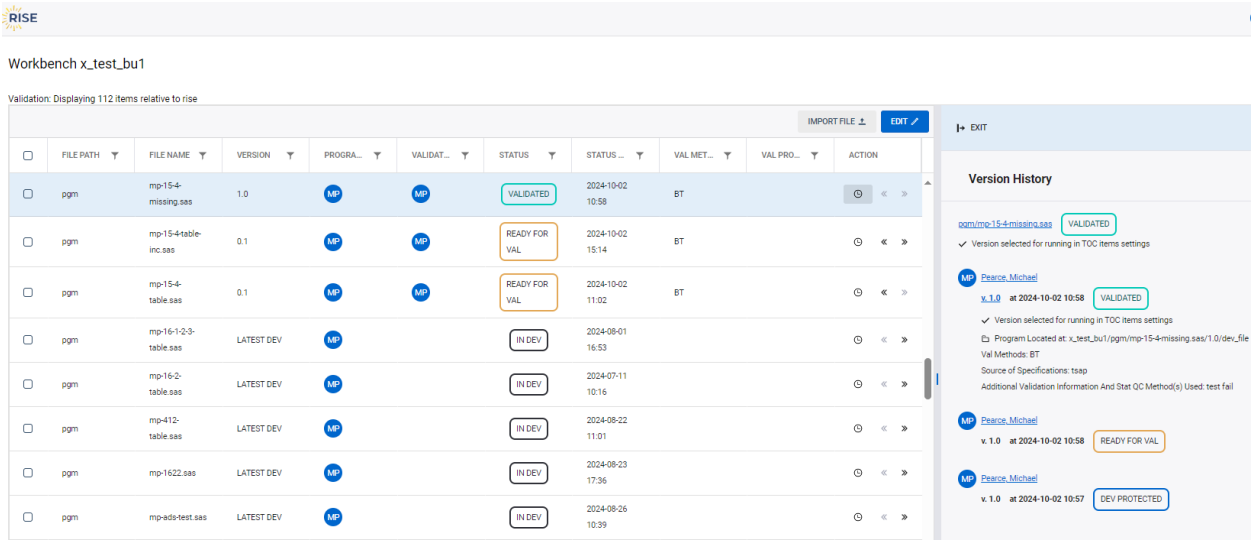

Figure 4: Validation process

PDF CREATION

Individual outputs, sections, chapters, or the whole TOC can be exported as a PDF report. Users have several options to customize content and layout, ensuring that the final document meets the conditions for publishing.

SYSTEM ARCHITECTURE & TECHNICAL INSIGHTS

The RISE UI is a single-page web application based on Angular. It is hosted on a Boehringer-managed Openshift Cluster (see next chapter). The business logic is implemented on Amazon Web Service “Lambdas”. The authentication is handled via an AWS API Gateway, which connects to Boehringer’s internal Active Directory.

Application metadata is stored in an AWS PostgreSQL database, while the clinical data and programs reside in existing, on-premise Network Attached Storages. Additionally, a read-only storage on AWS S3 is used to store the program versions.

SAS sessions run on a SAS Viya cluster, which runs on AWS EKS.

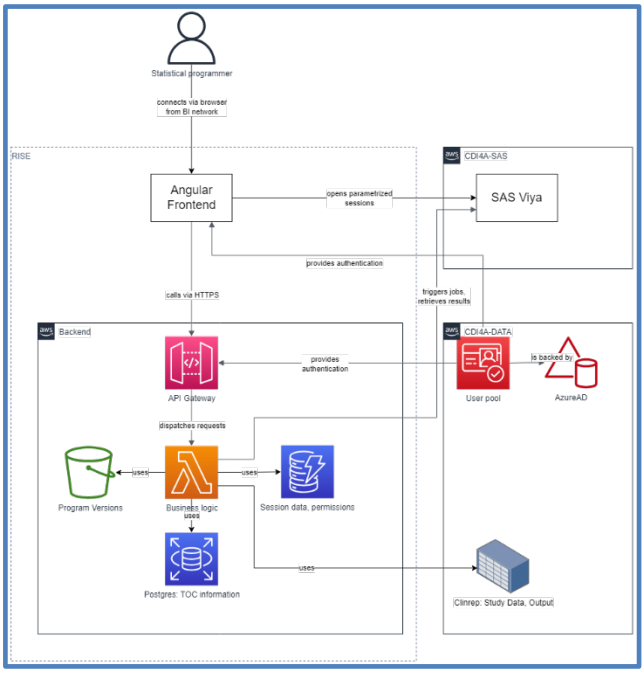

Figure 5: System Architecture

DEVELOPMENT AND VALIDATION

RISE is deployed using the OpenDevStack platform [1]. This platform uses tools from Atlassian [2] (Bitbucket, Jira, Confluence) for storing and documenting code, as well as project management activities. It uses Jenkins [3] for automatically building, testing, and deploying code changes made in the Bitbucket code repository to the Openshift platform [4].

Using this platform, the whole release process is automated and standardised within Boehringer, for the development, quality assurance, and production environment. Documents necessary for the validation (such as technical specifications) are partially generated from the documentation provided in JIRA, allowing for a much more efficient and reproducible validation process, while still ensuring all regulatory requirements.

OUTLOOK

Future versions of RISE aim to integrate R for batch running and interactive sessions, introduce batch job scheduling, an interactive dashboard, a notification centre, and upgrades based on user feedback.

CONCLUSION

In this paper, we presented Boehringer Ingelheim’s approach to a state-of-the-art, statistical computing environment termed RISE. While ensuring that the essential features of the previously used system are still supported, we took on a fresh design perspective on the user interface.

On the technical side, use of the powerful OpenDevStack platform within Boehringer and well-established Amazon Web Service services make for a reliable, cost-efficient and scalable platform.

REFERENCES

[1] OpenDevStack Documentation: https://www.opendevstack.org/ods-documentation/opendevstack/latest/getting-started/index.html

[2] https://www.atlassian.com/

[3] https://www.jenkins.io/

[4] https://www.redhat.com/en/technologies/cloud-computing/openshift

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at: Mario Lozina, mario.lozina@boehringer-ingelheim.com

Brand and product names are trademarks of their respective companies.