Paper SD09

Regulatory Report Automation with R

Shimeng Huang, Roche Canada, Mississauga, Canada
Suwen Li, Roche Canada, Mississauga, Canada

ABSTRACT

In order to speed up the process of filing, reducing redundant manual works and avoiding human errors are essential. To this end, automation of regulatory report writing is a promising and effective step. Taking advantage of the powerful R packages “rmarkdown” and “shiny”, we developed a web application to achieve data-dependent report automation for Clinical Study Report (CSR). This app is able to generate reproducible reports using quality checked data and standardized report components. The interactive user interface allows flexible specifications to be incorporated and the report can be dynamically updated accordingly. The draft report is displayed in the app, and is also downloadable with user-friendly styling in MS Word format.

INTRODUCTION

Today, dossier content is written in a highly manual way across the product life-cycle. Medical writers (MWs) create content for study and data-dependent portions of the Clinical Study Report (CSR) based on existing documents, siloed systems and repositories. This means that the content creation is inefficient and often contributes to high stress of the filing team. AutoCSR is created for the purpose of alleviating some of the manual work. Specifically, AutoCSR is an application written in the R language with the codebase consisting of two R packages that we developed which leverage the powerful “rmarkdown” and “shiny” R packages. The goal is to achieve auto-generation of a first draft of standardized, data-dependent categories in CSR, and produce standardized content that are consistent and easy to be reused across dossier documents

USER INTERFACE

AutoCSR is a web application with high levels of flexibility in the configuration for end users. The solution is being designed closely with medical writers to ensure usability and features are most beneficial to stakeholders. Overall, the interface has a landing “Home” page for configuring study information, a download section for extracting finished work, and a content section with category-specific features for fine tuning and generating the data-dependent content to be extracted.

On the home page (Figure 2), after a study number is selected, the arm names are auto-populated for the MWs to select as either the investigational arm or the comparison arm. The number of patients in the study is also auto-populated and displayed so that MWs can use their judgement to choose whether to use “large” or “small” study template in the content.

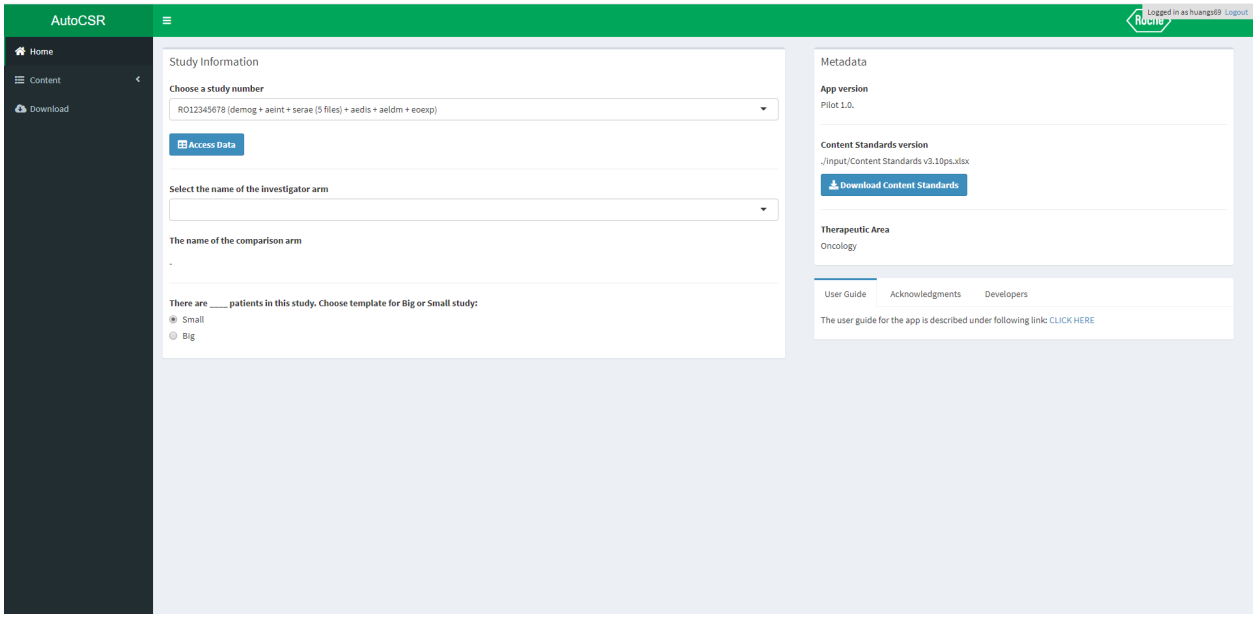

Figure 2. Home page of AutoCSR

In the “Content” Section, each data-dependent category in CSR has its own page containing two tabs, Content and Values, an example is given in Figure 3. The main area of the Content tab displays the template at the start, with all the placeholders highlighted in yellow, and instructional text in blue italic. On the right hand side, there is a control panel specifically for the category, allowing the users to flexibly choose threshold, optional paragraphs, or statistics to describe. The Values tab contains a table listing all the placeholders and its values. This allows for quick quality checks within the app. After selection in the control panel, the report is filled in and displayed in the same area after clicking a button (Figure 4). The user can go to the Download section once finished all categories they would like to produce, and the filled draft CSR can be downloaded in MS Word format (Figure 5-6).

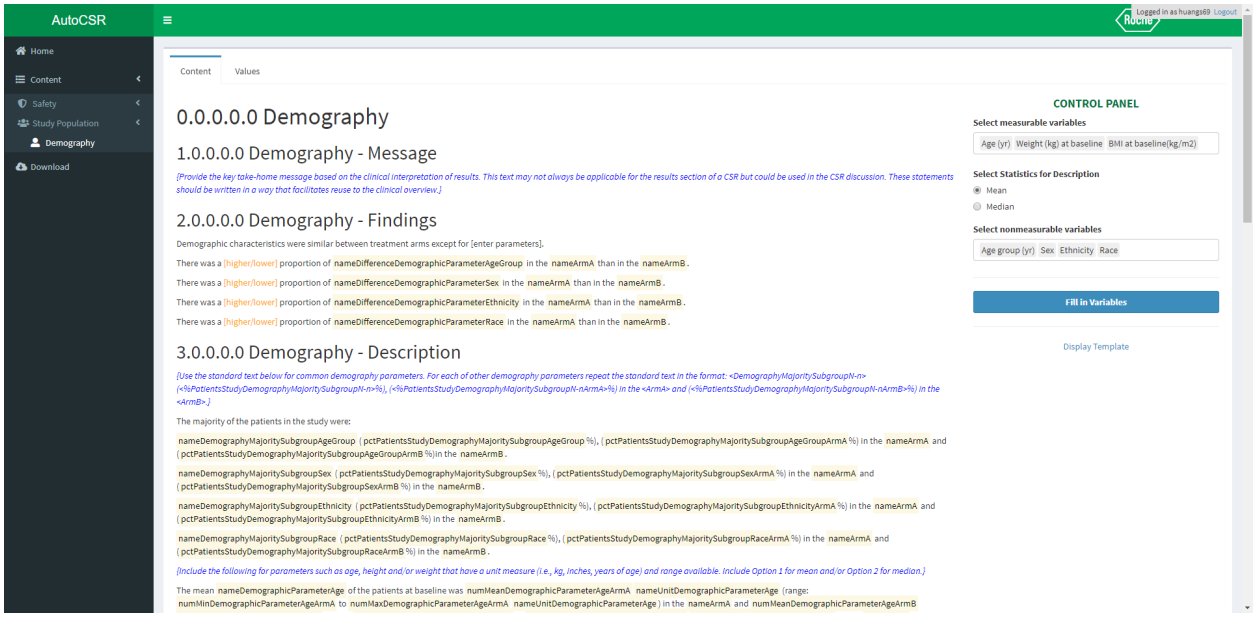

Figure 3. Display template at start

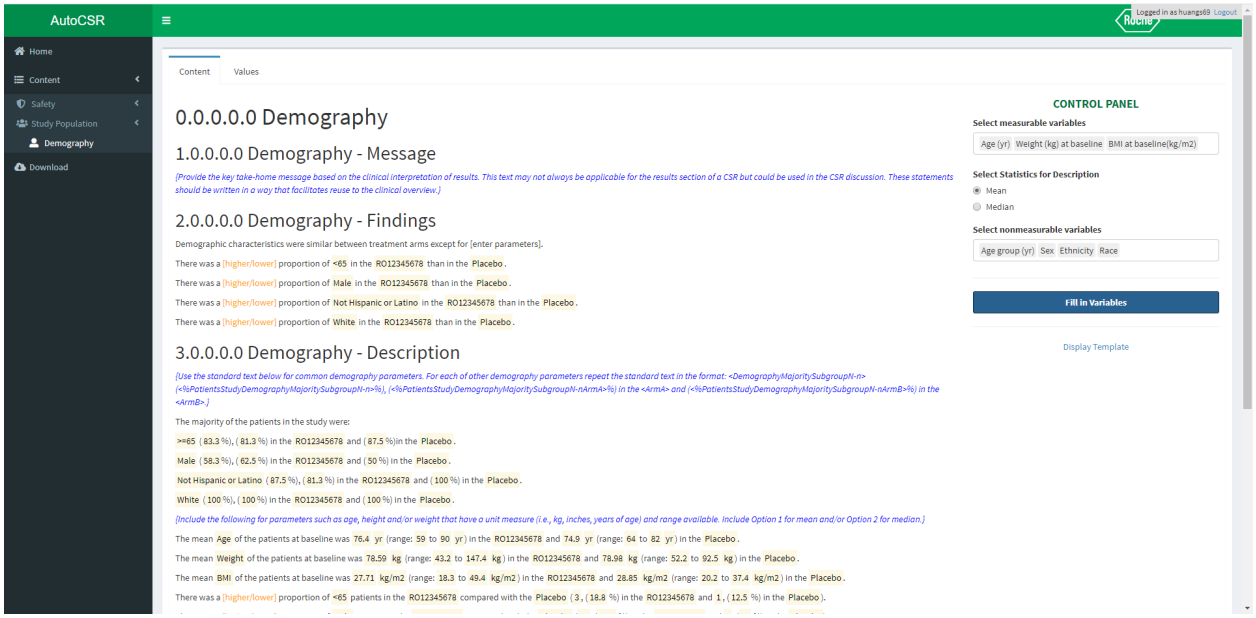

Figure 4. After filling in variables

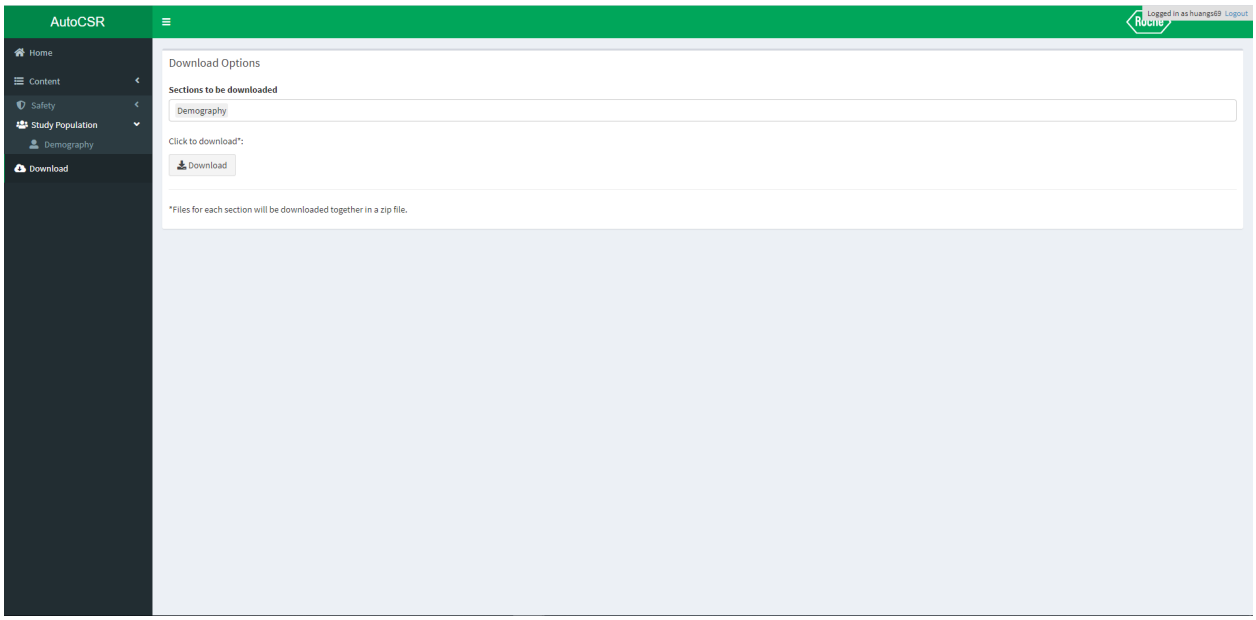

Figure 5. Download page

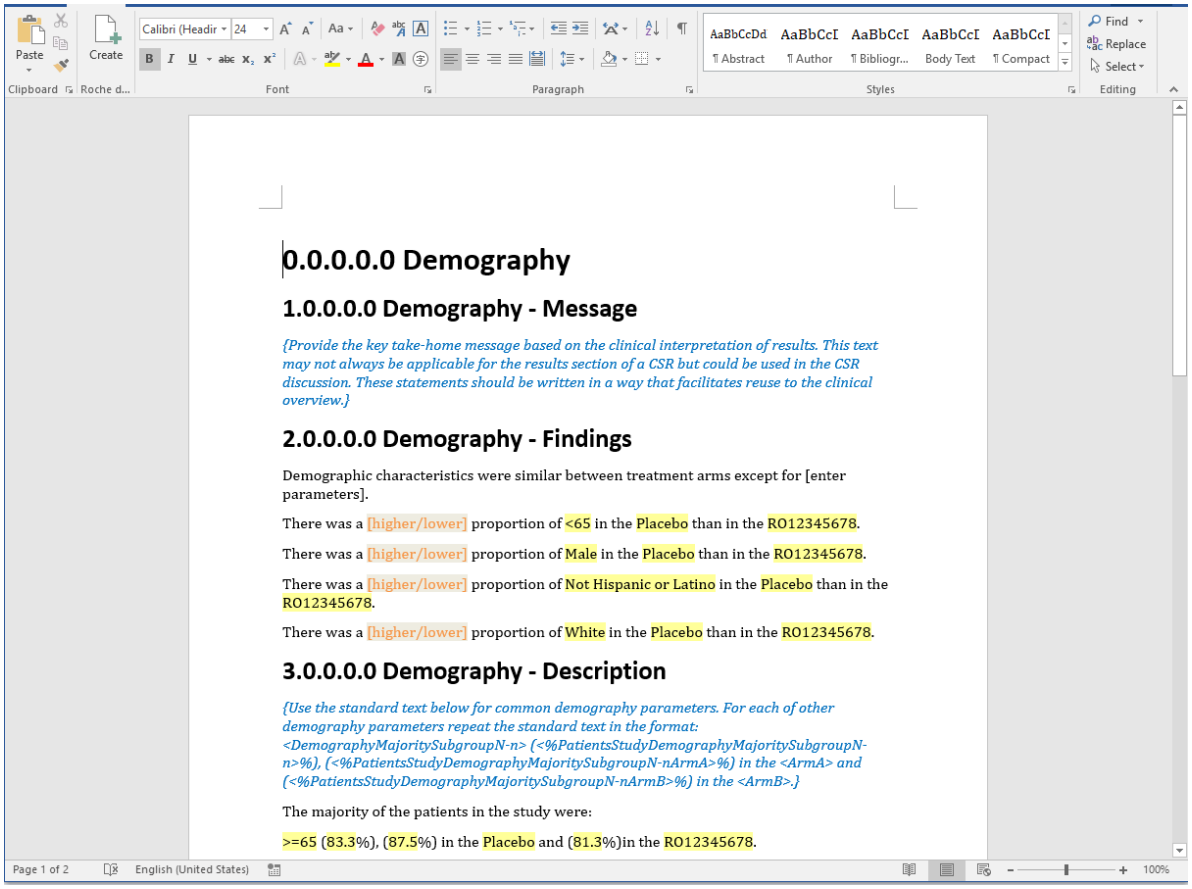

Figure 6. Downloaded draft CSR category

SYSTEM DESIGN

AutoCSR is a homegrown application with the user interface being a website created using a combination of the R package “shiny” and JavaScript. The process of data extraction, manipulation, and output generation are handled by R programs, where report generation leverages the R package “rmarkdown”.

The R code is developed in two R packages which we named as autoCSR and autoCSRweb. autoCSR is the main package that contains the functions for processing steps and generation of the draft report, whereas autoCSRweb consists of “shiny” modules, each corresponding to a page linked in the interface menu and calls functions in the autoCSR package.

The reason for this design is to allow flexibility and extensibility of the application. The package autoCSR can be used on its own by calling the functions in an R script, yet through autoCSRweb, users can interact with functions in autoCSR package through a web based interface. autoCSRweb package is designed in a modularized fashion, so that additional categories can be added easily.

The two main inputs of the application are the Content Standards and quality-checked analysis datasets from statistical programmers. The Content Standards provides the template of the CSR in an itemized way and defines the hierarchical structure of the report via item IDs, while the analysis datasets provide the numbers to be inserted in the output draft CSR.

A high-level structure of the system is demonstrated in Figure 1. Users (MWs) interact with the AutoCSR application by selecting input (via study number) on the home page and other widgets in the control panel on each page in the UI, which triggers data access, manipulation, and draft CSR output generation. Users can then download the output in Microsoft Word (.docx) format.

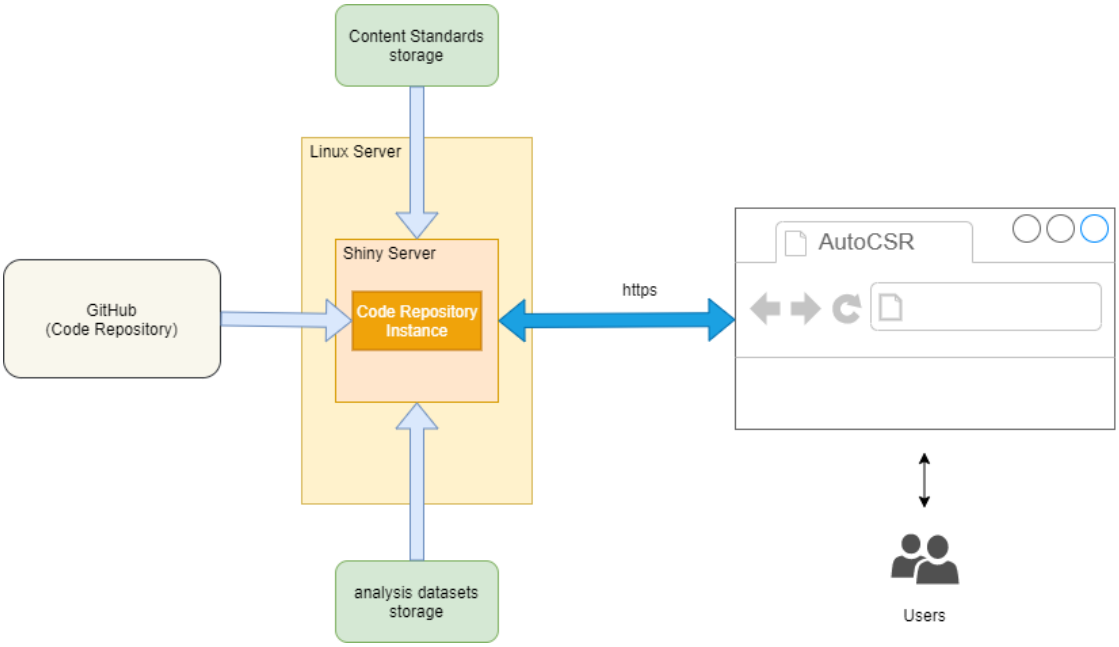

Figure 1. System Structure

CONCLUSION

AutoCSR is designed closely with the end users so that it is aligned with users' expectations and workflow. The code structure is flexible and easily extended, and the template can be updated with little or no coding efforts. The output of AutoCSR is in MS Word format with user friendly styling, so that the medical writers can easily see what has been filled in, and can further collaborate based on it. This tool is still under active development to incorporate new data-dependent categories in CSR as well as new features to assist the users.

ACKNOWLEDGMENTS

We would like to thank our colleagues for providing test datasets for this application, as well as their helpful comments, suggestions. We would also like to thank our managers for their continuous support and encouragement.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Shimeng Huang shimeng.huang@roche.com
Suwen Li suwen.li@roche.com
Roche Canada
7070 Mississauga Road
Mississauga, ON, Canada, L5N 5M8