Summer Student Report
Project Kryolize

Pawel Drozdowski

September 1, 2016

Abstract

The purpose of this document is to describe the work and results obtained by the author during his summer student internship at CERN. The author of this document was attached to the project Kryolize as a software developer, overtaking the job from a recently departed technical student.

1 Introduction

This project was carried out at CERN's Occupational Health & Safety and Environmental Protection Unit (HSE) under supervision of André Henriques.

> "Kryolize is a software tool for sizing the minimum discharge area of a safety protection device, against overpressure. Based on international (ISO), European (EN) and American (API) standards, Kryolize allows calculation and sizing of safety valves for cryogenic systems and is a novel tool that assists engineers with a uniform approach in the sizing of safety valves for cryogenics applications."

The purpose of the author's work was further development of the Kryolize tool, in particular development of new features and improvement of existing ones. This work was mainly carried out with the graphical user interface of the tool (figure 1). The overall goal was to increase the usability of the tool and preparing it for a first release to the customers.

1.1 Technologies

The technologies used in this project are listed below:

- Python 2
- wxWidgets
- Visual Studio
- Batch script
- Subversion (svn)
- JIRA

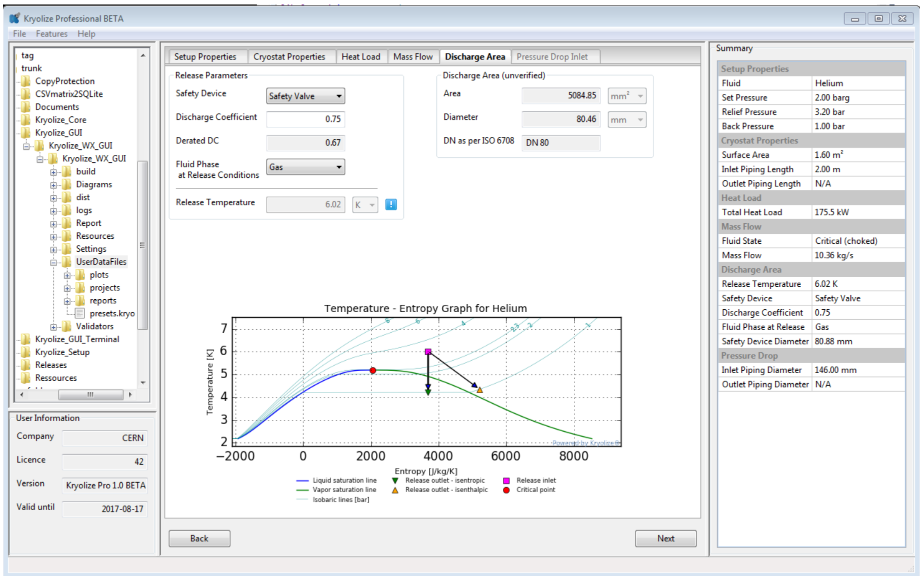

Figure 1: The Kryolize tool.

2 Contribution

During the 12 weeks of work, several contributions to the software itself as well as the development workflow have been made. This most substantial and important ones are briefly outlined below.

2.1 User Interface Features

The majority of work on the project pertained to the graphical user interface of the Kryolize tool.

2.1.1 Projects and Presets

A feature allowing the users to save their progress on a project and load it at later date. Additionally, the users are able to specify preset (default) values, which are automatically filled in all the projects. The projects are stored in simple JSON files with a custom extension: *.kryo*.

2.1.2 Thermodynamic Diagrams

This feature allows the users to create a variety of thermodynamic diagrams for large number of fluids in a semi-interactive manner. Figure 2 shows an example diagram produced with the tool.

2.1.3 Thermodynamic Library Query

A simple access layer to the CoolProp thermodynamic library has been created. This feature allows querying the library for a large number of fluid properties. Figure 3 shows this section of the Kryolize tool.

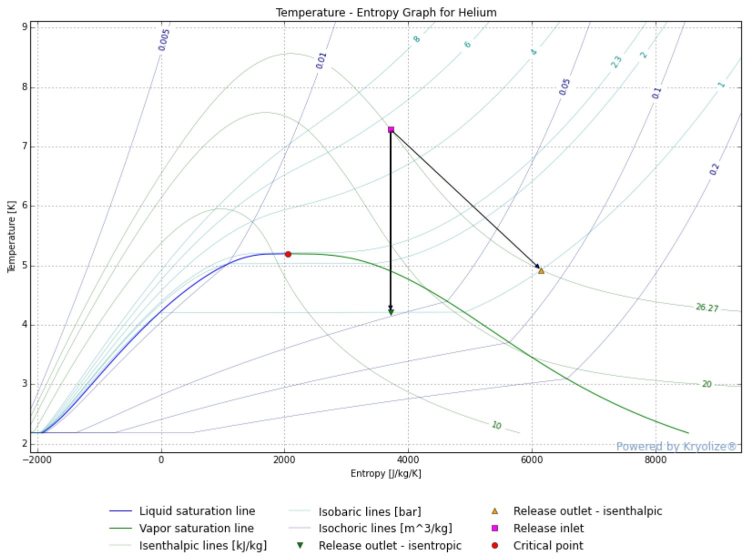

Figure 2: Example thermodynamic diagram produced by Kryolize.

2.1.4 Unit Support

By default, Kryolize uses a custom, SI-based system of units. In order to increase the usability of the tool and cater to a wider user audience, support for the imperial unit set has been implemented.

2.1.5 Internalisation

The storage static strings used in the graphical interface has been centralised. The benefits of this are twofold - the software maintainability increases and the software can be internationalised almost seamlessly. The static strings are stored in a single JSON file; one can translate this file to other languages, and thus allow the end-users to use the tool with a language of their choice. This implementation is flexible and even makes it possible for the users themselves to perform such translation.

2.2 JIRA Workflow

Until recently, the feature driven development of the Kryolize project relied mostly on an Excel spreadsheet, where the prospective features and bugs were listed and prioritised. This spreadsheet has been migrated to JIRA; future issues are now also registered using JIRA. The benefits include much better requirement and implementation traceability, tight integration with version control commits and allow to take advantage of the extensive JIRA feature set and UI.

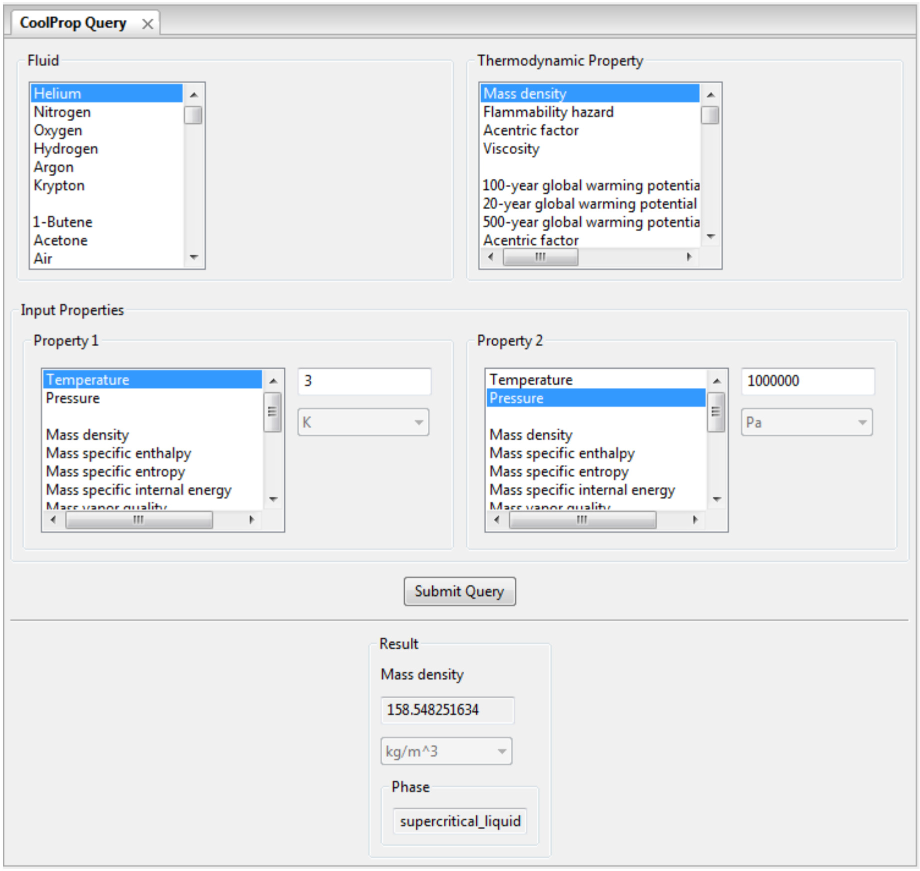

Figure 3: Querying a thermodynamic properties library.

2.3 Licence Check

Once Kryolize tool is distributed, licences will be issued to the customers. Until recently, the only possible licence was an on-line one - requiring the user to be connected to internet from a specific IP range. Naturally, for some users (e.g. consultants working in the field), this is not an optimal solution. For these cases, an off-line licence scheme based on asymmetric cryptography has been conjured and implemented. In addition, several vulnerabilities of the on-line licence protocol were discovered and brought to the attention of its developers.

2.4 Build Procedure

In order to meet the requirements for intellectual property protection set by CERN's knowledge transfer representatives, the Kryolize software has to be packaged in a certain, elaborate and non-standard way. Until recently, this process was largely manual and required some time and effort to be allotted for performing of repetitive tasks during each new build. As a remedy for this, several scripts have been written, thus making the clumsy build process nearly completely automatic.

2.5 Maintainability

In order to ensure maintainability of the software and ease the learning process for future developers, large amounts of code documentation and diagrams have been produced. The user interface codebase has been refactored (among other matters, as mentioned earlier, centralisation of static string storage) and proper logging was added to it. Finally, a suite of automated graphical user interface tests has been implemented in order to ensure the correctness of the tool under some common usage scenarios.

2.6 Bug Fixes

During the development of new features, several (sometimes critical) bugs have been discovered in the existing codebase. These have, naturally, been remedied.

2.7 Other

A substantial number of smaller features, adjustments and improvements have been implemented. They are too small to be discussed individually; generally speaking, their purpose is improvement of the end-user experience with the tool.

3 Conclusion

As a result of this project, the Kryolize tool has received a large number of new features, improvements and bug-fixes and is now ready for the first release to the customers. In addition to the actual software development, the project development workflow and the codebase maintainability have been improved.