User Requirement Specification (URS)

Project: Dashboard for visualization of pilot data
Location: Pilot site, Teknologisk Institut, Høje-Taastrup
Project responsible: RS, APV
Document responsible: APV

Background & current situation:

As of April 2024, Algiecel has been running a pilot unit for close to two years, at the facilities of the Teknologisk Institut in Høje-Taastrup. Several batches have been completed in this period of time, and process data has been gathered from different sources. Two main types of data can be distinguished here: inline data and offline data.

Inline data is collected from the sensors that are integrated into the pilot unit, these cover variables such as temperature, flowrate, dissolved oxygen, pH and pressure, among others. The readings from the sensors are stored in .csv format in the HMI's computer, located inside the container. Data is saved every 10 s, regardless of the operational state of the system (i.e. the HMI is never shut off, hence data is saved even when the reactor is not in operation and there is no batch running), and the files are organized by date (i.e. a new .csv file is created every single day, and named accordingly to indicate so). Once a day, at noon (12:00 midday), the computer at the pilot transfers the daily file to a computer in the office, via wire connection, given that the office computer is connected to the network. If not, the file is stored at the HMI's computer, and as soon as the office computer is available, all missing files are transferred. Further, a transfer of the .csv file that is currently being written with new daily data can be forced at any time by running a script in the computer at the office.

In the .csv files, data is saved in the following format, with different fields separated by semicolon:

"Variable name";"Timepoint";Variable value;Validity;Time since epoch

This information is also available in all .csv files, in a heading. The first two fields are of type *string*, the data is enclosed in quotation marks, and they contain the name of the sensor, and the date (DD-MM-YYYY) and time (hh:mm:ss), respectively. Variable values are *Floats* if the data is analog, and *Integers* otherwise; the validity is a *Boolean* value, and time since epoch is of type *Float*.

For a given point in time, data from different sensors is separated with a line break (i.e. each line indicates data from a different sensor). The order in which sensors are read and saved in the file is consistent between timepoints. There is no particular differentiation between timepoints other than a line break and a change in the *Timepoint* field.

Offline data, on the other hand, is obtained from samples taken daily or less often, and analyzed in the laboratory. This data can be further subdivided into two categories: "quick" analysis, and analytical analysis.

The first category refers to analysis performed by Algiecel in situ at Teknologisk Institut, and it is currently stored in spreadsheets, categorized by batches. It contains numerical information, binary information and qualitative information (i.e. "no contamination", "little contamination", "some contamination", etc.). Currently, due to its ease of access, this is the type of data that is most normally used to analyze and evaluate the performance of the reactor.

The second category refers to outsourced analytical tests performed weekly by a partner university. The information is reported back in .pdf and .docx format at the end of each batch (2 – 3 months), once all the samples for that batch have been analyzed, and the data is structured in tables. Data type is numerical, with the exception of the label "<LOQ" used for variables below the limit of quantification.

Project objective & requirements:

The objective of the present project is to develop a dashboard, to facilitate the visualization of the different data that is currently available at the company. The main objective of the dashboard is to serve the company as a learning tool, i.e. to evaluate performance and compare different batches; it is not intended to serve as a tool for real time visualization and decision making, hence updates on the displayed information can be done daily rather than minute / second basis.

The main requirements of the dashboard will be:

- **Manual definition of batches**: Once the inline data has been imported, the dashboard must allow the user to define different type of batches (i.e. CIP run, cultivation run, test run, etc.) by setting a starting point and an endpoint. Further into the future, we aim to add this feature as a variable in the HMI, thus the definition of batches could be done automatically by reading the process data.

- **Automatic importation and parsing of data**: Once a batch has been defined, the system would have to be able to identify the set of .csv files containing the process data for that batch, access them, extract the numerical data, categorize it into different variables and join the information of each variable, from all .csv files, into singular data series that can be displayed. The system should then be able to save this data in a new file, so the data importation and parsing does not have to be done every time we want to visualize the data of the batch. A similar process would be required for the outsourced analytical data: in order to facilitate its importation into the dashboard, at Algiecel we would manually move the .pdf / .docx data into .xslx files (Excel spreadsheets) and then export it as .csv files. The format of the data would be different from the one at the process logs, but we would arrange it and distribute it so that the resulting text file is as easily readable for the dashboard as possible.*
  *What is considered as *readable* can be discussed during the development of the project.

- **Manual input of data**: For the offline data collected in situ, where few new datapoints are generated daily, the dashboard must allow an authorized user to create new variables, and input their data manually. Formats that have to be available include numerical, binary or restricted strings (user has to choose between a selection of predefined labels, and be allowed to order them for the visualization according to any particular sense of hierarchy). An option to add units to the variable must be available as well; if selected, a list of valid units for the variable would have to be described. When filling data, fields that should be available would be a drop-down menu of the created variables, the value of the variable, the units and a field for extra notes.

- **Adaptative visualization**: For numeric timeseries, the dashboard should display the data as a line graph (variable vs time), with the real measured values being displayed with markers, connected with smoothed lines. Error bars indicating the standard deviation should be added to the markers of the offline data, whenever the value displayed has been obtained as the mean of replicate measurements. For binary data, a line graph can also be used, but transitions from one state to the other must be displayed with sharp edges, not smooth curves. For string data, a column graph would be a preferable option, as the use of line graphs would suggest the existence of intermediate transitional states.

- **Variable time scale**: The system would have to be able to switch between an absolute timeframe (date and hour) and a relative one (time elapsed with respect to the start of the batch).

- **Simultaneous visualization of multiple batches**: The dashboard should allow to visualize one or more variables from one or more batches at the same time, using a relative timeframe (as described above), to facilitate comparison between batches. If several variables with different units are to be displayed, new vertical axis should have to be dynamically created and scaled, one per variable.

- **Data exportation**: The system should allow to export the visualized data, both in numeric / text format (the raw data used to generate the plot) and in graphical format (the generated plot).

Additional requirements for the dashboard would include:

- **Unit management**: For a given particular magnitude, the user should have the capability of switching between commonly used units (i.e. from m^3/h to l/s, or from bar to atm or Pa). The units used in the .csv should be set as the default for that magnitude.
- **Definition of custom function**: The system should have the capacity to accept the definition of simple functions that could be used to transform or generate new variables based on the available ones. Examples of this could include the calculation of daily productivity as the increment in biomass with respect to the prior day, or the calculation of flow velocity based on the area of the reactor (constant) and the flowrate (variable).
- **Adaptative display of data**: Of particular importance for inline data (collected with a 10 s frequency), the dashboard should be able to adaptatively adjust the number of displayed data, depending on the resolution of the time scale (i.e. if data is visualized across several weeks, the dashboard should select the data to display with a frequency of hours. If we zoom in for a particular hour, the dashboard should now display data with a frequency of seconds).