Dan Wang

REQUIREMENT SPECIFICATIONS
FOR RPA SOFTWARE - UiPath

Information Technology
2019
FOREWORD

This Bachelor’s thesis is done for Wärtsilä Finland Oy.

I would like to express special appreciation to my line manager, Mr. Asko Vakkila, and supervisor, Mr. Mikael Ehrs from Wärtsilä for offering this topic and remarkable suggestion, support, and encouragement all along the process. I’m also so grateful to my supervisor, Mr. Chavez Santiago, from Vaasa University of Applied Sciences who spent the time on wonderful periodical discussions and provided plenty of appreciative advice for my thesis.

Furthermore, I would like to thank all the mentors during the RPA training course in Wärtsilä who gave remarkable hands-on training and offered strong support during and after the training course.

I want to give great thanks to my boyfriend, Anh Duy Nguyen, for his selfless help and my cat, Melon Wang, for her companionship. In addition, I really appreciate the full understanding of my decision from my family which has been always providing me love and energy in my life.

Dan Wang

Vaasa, Finland

18 Apr 2019
In order to provide systematic instructions and replace trivial manual data entry workflow, this thesis work is mainly dedicated to building the methodology of data entry as a more advanced, efficient and accurate automation solution by Robotic Process Automation (RPA) software for Wärtsilä’s Product Information team.

In a word, this thesis work is designed for reducing consumption of time and improving data entry accuracy.

RPA is worldwide applied as an emerging form of business process automation technology. To achieve the requirements, UiPath is one of the most representative software in RPA technology which is adopted to build automation solutions.

The main work of this thesis was carried out in two parts: editing data entry update instructions and building up RPA solutions. The update instructions were meant to offer a manifest view of the methodology of data entry to web-based technical database maintenance application which stores search data of Wärtsilä engine properties. And the structure of RPA solutions is concerned with both completed update instructions and particularity with web-based technical database maintenance application to implement fully automatic processes from updating formatted Excel data source to the database.

Keywords Wärtsilä R&D and Engineering, RPA, Data entry
CONTENTS

FOREWORD

ABSTRACT

1 INTRODUCTION ....................................................................................... 7
  1.1 Purpose ............................................................................................... 7
  1.2 Background ........................................................................................ 7
    1.2.1 Wärtsilä Oy ................................................................................. 7
    1.2.2 Product Information Team ........................................................ 8
    1.2.1 Testing Laboratory ...................................................................... 9

2 LITERATURE REVIEW ............................................................................ 10
  2.1 Robotic Process Automation ............................................................ 10
  2.2 UiPath ............................................................................................... 11

3 METHODOLOGY ...................................................................................... 12
  3.1 Update instruction ............................................................................ 12
  3.2 Data Collection ................................................................................. 12
  3.3 Data Process ....................................................................................... 13
  3.4 UiPath Activities ............................................................................... 16

4 RESULT ..................................................................................................... 19
  4.1 Main table data normal update (normal/draft) ..................................... 19
    4.1.1 Requirement Specification ..................................................... 19
    4.1.2 Solution method ....................................................................... 19
  4.2 HB&EG&CA system data 1cylinder batch update (normal/draft) ......... 26
    4.2.1 Requirement Specification ..................................................... 26
    4.2.2 Solution method ....................................................................... 26
  4.3 Error handler and feedback message ............................................... 34

5 DISCUSSION AND FUTURE DEVELOPMENT ....................................... 35

REFERENCES .................................................................................................. 37
LIST OF ABBREVIATIONS

RPA  Robotic Process Automation
PI   Product Information
AI   Artificial Intelligence
APIs Application Programming Interfaces
GUI  Graphical User Interface
CN   Change Notice
URL  Uniform Resource Locator

LIST OF FIGURES

Figure 1. Structure of Product Information team  p.8
Figure 2. Structure of System Integration & Performance.  p.9
Figure 3. RPA benefits  p.10
Figure 4. Over 700 enterprise level customers using UiPath automation tool.  P.11
Figure 5. Pre-processed update information format  p.14
Figure 6. Pre-processed data table format  p.15
Figure 7. Pre-processed batch data table update format  p.16
Figure 8. Working flowchart for Main table normal update (normal/draft)  p.20
Figure 9. General UiPath process of Main table normal update solution  p.21
Figure 10. UiPath process of reading and storing information from Excel file  p.21
Figure 11. UiPath process of for each engine id loop  p.22
Figure 12. UiPath process of filtering out engine id and detail page  p.22
Figure 13. UiPath process of choosing update mode  p.23
Figure 14. UiPath process of getting textbox attribute with field name p.24
Figure 15. UiPath process of “Get Attribute” activity selector p.24
Figure 16. UiPath process of inputting data into matched textbox p.24
Figure 17. UiPath process of column and row counter p.25
Figure 18. UiPath process of checking if current data is empty or null p.25
Figure 19. UiPath process of saving the updates p.26
Figure 20. Working flowchart for 1cylinder table batch update (normal/draft) p.27
Figure 21. General UiPath process of 1cylinder batch update solution p.28
Figure 22. UiPath process of getting information ranges from Excel file p.28
Figure 23. UiPath process of for each update system name loop p.29
Figure 24. UiPath process of for each group of engine ids loop p.30
Figure 25. UiPath process of setting a new navigation p.30
Figure 26. UiPath process of choosing update mode p.31
Figure 27. UiPath process of dynamic html page selector p.31
Figure 28. UiPath process of getting textbox attribute with field name p.32
Figure 29. UiPath process of “Get Attribute” activity selector p.32
Figure 30. UiPath process of inputting data into matched textbox p.32
Figure 31. UiPath process of column and row counter p.33
Figure 32. UiPath process of checking if current data is empty or null p.33
Figure 33. UiPath process of saving the updates p.33
Table 1. List of UiPath activities appeared in solution p.17
1 INTRODUCTION

1.1 Purpose

Advanced technological development, rapid information circulation, intensive human communication and convenient life circumstances present our high-speed developed society with the dramatic growth of dataflow, which increases the complexity of data processing technology in the meanwhile. Hence, the digitalizing enterprises’ attention on intelligent working methodologies appears to experience an inspiring upward trend between working demands and practical intention day by day. Therefore, achieving automatic workflow becomes a topical theme which promotes the technical development of Robotic Process Automation (RPA).

As a leading industry of engine & marine solution, Wärtsilä confidently holds critical patent technologies with remarkable performance and functionality. Thus, the digital methodology of processing the complex data from plenty of engine types contributes a nonnegligible challenge. In order to implement Wärtsilä a functional digitalized industry, RPA technology development will be one of the most essential steps to the intelligent future company upgrade.

1.2 Background

1.2.1 Wärtsilä Oy

Wärtsilä is a global leader in smart technologies and complete lifecycle solutions for the marine and energy markets. By emphasizing sustainable innovation, total efficiency and data analytics, Wärtsilä maximizes the environmental and economic performance of the vessels and power plants of its customers. /1/

Wärtsilä aims at increasing efficiency while enabling a zero-emission society: /2/

- Unique market position
- Deep customer understanding
- Predictive analytics and asset optimisation
- Global service network
- Extensive product range
- Engineering and technology expertise
- Significant investments in future technology

1.2.2 Product Information Team

Product Information (PI) is responsible for processing, organizing and structuring the "raw" data after design and validation which is received from different internal sources, e.g. test laboratory. So as to secure consistent value delivery with integrated useful information to Wärtsilä’s internal and external customers, presenting the processed data in a given context is the subsequent requirement.

PI team is dedicated to creating possibilities that contribute to

- Accurate offering, selling and manufacturing of the products (via varying markets demands)
- The total lifecycle support that benefits both the customers and environment.

PI team’s globally responsible for maintenance and development of the following internal/external information products/tools as shown in Figure 1:

![Figure 1. Structure of Product Information team](image-url)
1.2.1 Testing Laboratory

Engine performance expertise is an activity within R&D and Engineering, System Integration and Performance (Figure 2).

The performance team is defining the engine concepts to meet the performance requirements of the product. In the initial phase of the project selling unit requirements are converted to technical requirements of engine technologies, components and functionalities, often in co-operation with suppliers.

Engine simulations are typically used for predicting the performance and operating parameters of the engine at this stage. Later, in the testing phase of the engine, performance experts plan, follow, analyse and report performance tests. Performance testing of a new product can take a year in the first round and the work continues further in form of continuous improvement and product updates. Engine testing is typically taking place in engine laboratory or in production, but also on the field.

Practically performance experts deal with thermodynamics, turbocharging, fuel injection, valve train and automation parameters as well as emissions. They take into account technical constraints of the engine ensuring best overall performance and maintaining engine reliability during concept development. Performance related documentation is released as a final outcome of the team activities being the source for guarantee values in binding contracts. Performance expertise is also involved in product upgrades for service and in field support. [3/]

Figure 2. Structure of System Integration & Performance
2 LITERATURE REVIEW

2.1 Robotic Process Automation

Robotic process automation (RPA) is an emerging form of business process automation technology based on the notion of software robots or artificial intelligence (AI) workers. In traditional workflow automation tools, a software developer produces a list of actions to automate a task and interface to the back-end system using internal application programming interfaces (APIs) or dedicated scripting language. In contrast, RPA systems develop the action list by watching the user perform that task in the application’s graphical user interface (GUI), and then perform the automation by repeating those tasks directly in the GUI. This can lower the barrier to use of automation in products that might not otherwise feature APIs for this purpose.

Robotic Process Automation allows anyone today to configure computer software, or a “robot” to emulate and integrate the actions of a human interacting within digital systems to execute a business process with no never sleeps, makes zero mistakes and costs a lot less than an employee (Figure 3).

![Process automation benefits](image)

**Figure 3. RPA benefits**

In contrast to other, traditional IT solutions, RPA allows organizations to automate at a fraction of the cost and time previously encountered. RPA is also non-intrusive in nature and leverages the existing infrastructure without causing disruption to underlying systems,
which would be difficult and costly to replace. With RPA, cost efficiency and compliance are no longer an operating cost but a byproduct of the automation. /8/

2.2 UiPath

UiPath is a global software company that develops a platform for Robotic Process Automation (RPA or RPAAI). /9/ The company is working towards a future where employees at every organization are empowered to automate tedious and time-consuming work, enabling them to focus on creative, challenging problems. /10/

UiPath is, in essence, a group of engineers driven by an enduring ambition to build the best technology they possibly could. They went in wholeheartedly and made UiPath the most widely used RPA platform in the world today, drawing together elite enterprises (Figure 4), global partners committed to excellence in implementation and product innovation, and the largest RPA developer community ready to make an impact on the world. /11/

Figure 4. Over 700 enterprise level customers using UiPath automation tool. /12/
3 METHODOLOGY

3.1 Update instruction

As concerning the complexity of data entry on the web-based technical database maintenance application, manifest update instructions are required for non-knowledge employees to obtain a better preview of basic data entry workflow.

There are mainly 11 update instructions which reflect ten types of data entry and one result list preview method involved in edition:

- Main table normal update (normal)
- Sub table normal update (normal)
- Main table normal update (draft)
- Sub table normal update (draft)
- Main table batch update (normal)
- Sub table batch update (normal)
- Main table batch update (draft)
- Sub table batch update (draft)
- HB&EG&CA 1cylinder batch update (normal)
- HB&EG&CA 1cylinder batch update (draft)
- Viewing Data Result List

The update instructions include process step description, annotations, and screenshot to support better understanding. At the same time, update instructions will be provided to the UiPath solution developer as fundamental structure references and basic workflow knowledge.

According to company security, this part will not have further information.

3.2 Data Collection

The updating data source is received from Engine Laboratory located in Vaasa city. The original data, or the so-called approved test data, will appear as data tables in specific datasheets in Excel files. The Excel files are created by different engine teams with a different formatted style which is difficult to be read by UiPath robot in every certain
solution, therefore, this situation will lead to the Data Process step before running UiPath solutions.

3.3 Data Process

With the variety of approved test data source Excel files as well as the convenience of building UiPath solution, reduces the possibility of data corruption during transmission and process, increases the content integrity and strengthens the solutions’ functionality. Also, the pre-processing of data is essential towards the later solution.

Generally, Data Process is to format the data table from the approved test data source Excel file to be able to be recognized and operated by UiPath robot.

In the later development, I integrated all the Excel formatted files for different solutions into one which can be applied to all kinds of updates.

The common format of a data table is stored in the data type datasheet for data entry is divided into two general types:

- Single engine id per time update (Figure 6)
- Multiple engine ids per time update (Figure 7)

In both types of the data table include three common parts:

- Id(s) and Messages
- Engine information
- Data table

For all kinds of updates, the update comment and Change Notice (CN) message are stored in the data type datasheet as independent optional information.

As for the sub system table update and sub system 1 cylinder table update is the sub system names. Also, all the ranges of the starting cell or cell’s name which are provided to UiPath to read from data type datasheet is stored in information type datasheet – to make the flexibility of operation in Excel files which means when there is any position change in the data table, you only need to relocate the starting cell or cell by rewriting the starting
cell or cell’s name in information type datasheet which doesn’t require any change on UiPath solution. However, all the position of the existing cell in the information type datasheet are not allowed to be changed, the new cell can be added at the end of row or column in each kind of table (Figure 5).

**Figure 5.** Pre-processed update information format
Figure 6. Pre-processed single data table update format
3.4 UiPath Activities

To make more adaptable solutions and take into account the real situation with various requirements and update methods, studying and using UiPath activities properly is considered as a prerequisite process. Table 1 is a list of all the UiPath activities appeared in these UiPath solutions.

(Note: The column with header “Input”, “Output” and “Remark” are not written as standard UiPath activities guide but considered with the practical situation in solutions.)
<table>
<thead>
<tr>
<th>Activity name</th>
<th>Function description</th>
<th>Output</th>
<th>Input</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Scope Activity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excel Application Scope</td>
<td>Opens an Excel workbook and provides a scope for Excel Activities</td>
<td>None</td>
<td>Workbook-Path</td>
<td>If the specified file does not exist, a new Excel file is created</td>
</tr>
<tr>
<td>Open Browser</td>
<td>Opens a browser at a specified URL and execute multiple activities within it</td>
<td>None</td>
<td>Url</td>
<td>The navigation of activities inside is only according to input URL</td>
</tr>
<tr>
<td>Attach Browser</td>
<td>Attaches to an already opened browser and perform multiple actions within it</td>
<td>None</td>
<td>Selector (Indicate on Screen)</td>
<td>Can be used to change covered activities’ navigation in <strong>Open Browser</strong></td>
</tr>
<tr>
<td>Close Application</td>
<td>Closes the application corresponding to a specified UI element</td>
<td>None</td>
<td>Selector (Indicate on Screen)</td>
<td>Corresponding to <strong>Open Browser</strong></td>
</tr>
<tr>
<td><strong>Functional Action Activity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Range/Column/Row/Cell</td>
<td>Reads the value of an Excel range/column/row/cell and stores it in UiPath variable</td>
<td>Data table/Array/String</td>
<td>Range/StartingCell/Cell, SheetName</td>
<td>Uncheck option AddHeaders to get variable without header</td>
</tr>
<tr>
<td>Click</td>
<td>Clicks a specified UI element</td>
<td>None</td>
<td>Selector (Indicate on Screen)</td>
<td>Manually move mouse may interrupt processing</td>
</tr>
<tr>
<td>Select Item</td>
<td>Selects an item from a combo box or list box</td>
<td>None</td>
<td>Item, Selector (Indicate on Screen)</td>
<td>After indicating on screen, the list with all the available options is displayed in the activity</td>
</tr>
<tr>
<td>Check</td>
<td>Selects or clears radio buttons and check boxes</td>
<td>None</td>
<td>Selector (Indicate on Screen)</td>
<td>Selects Check or Uncheck in Options, the default action is Check</td>
</tr>
<tr>
<td><strong>Type Into</strong></td>
<td>Sends keystrokes to a UI element</td>
<td>None</td>
<td>Text, Selector (Indicate on Screen)</td>
<td>Special keys are supported from the dropdown list, check EmptyField in Options is needed</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------</td>
<td>------</td>
<td>-----------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Assign</strong></td>
<td>Allocates any value to a variable or argument.</td>
<td>None</td>
<td>Variable/argument, Value</td>
<td>Are used for dynamic selector and counter</td>
</tr>
<tr>
<td><strong>Send Hotkey</strong></td>
<td>Sends keyboard shortcuts to a UI element.</td>
<td>Key, Selector (Indicate on Screen)</td>
<td>Special keys are supported and can be selected from the dropdown list</td>
<td></td>
</tr>
<tr>
<td><strong>Find Element</strong></td>
<td>Waits for the specified UI element to appear on the screen</td>
<td>None</td>
<td>Selector (Dynamic selector)</td>
<td>Dynamic selector will rewrite selector (Indicate on Screen)</td>
</tr>
<tr>
<td><strong>Get Attribute</strong></td>
<td>Retrieves the value of a specified attribute of a UI element</td>
<td>Value of attribute</td>
<td>Attribute, Selector (Dynamic selector)</td>
<td>Supports attribute list with all the available options is displayed in the activity</td>
</tr>
</tbody>
</table>

**Logical Decision Activity**

<table>
<thead>
<tr>
<th><strong>For each</strong></th>
<th>Performs an activity or a series of activities on each element of a collection.</th>
<th>None</th>
<th>Array</th>
<th>Loops activities until the end of array</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Switch</strong></td>
<td>Executes a single course of action out of multiple options, based on a preset condition</td>
<td>None</td>
<td>Expression, TypeArgument</td>
<td>Sets TypeArgument supports as String</td>
</tr>
<tr>
<td><strong>If</strong></td>
<td>Takes one of two different courses of action, depending on whether a specified condition is met.</td>
<td>None</td>
<td>Condition</td>
<td>Contains three sections: Condition, Then, and Else.</td>
</tr>
<tr>
<td><strong>Anchor Base</strong></td>
<td>Searches for a UI element by using other UI elements as anchors.</td>
<td>None</td>
<td>AnchorPosition</td>
<td>AnchorPosition is to locate later activities by the position of anchor activities, GUI will make difference</td>
</tr>
</tbody>
</table>

Table 2. List of UiPath activities appeared in solution
4 RESULT

4.1 Main table data normal update (normal/draft)

This part introduces the methodology of main table data normal update in normal mode or draft mode.

4.1.1 Requirement Specification

This task is to update a group of newly generated technical data which usually is separated to multiple field names into the main engine configuration data table with a single engine id per time on web-based technical database maintenance application.

4.1.2 Solution method

According to the update instruction and requirement, this solution will cover five sections (Figure 8):

- Reading data from formatted Excel file
- Storing data in UiPath
- Inserting update message with certain update mode with specified username for UiPath robot
- Updating data to the web-based technical database maintenance application
- Handling appearance of the error.
Figure 8. Working flowchart for Main table normal update (normal/draft)

The whole UiPath solution is divided into two ordered processes inside the flowchart: Excel process and web-based database browser process (Figure 9).
When the UiPath robot starts to run, the first step is to go through the Excel file with the file path, read and store the update starting cell or cell’s name and other information in the information type datasheet, and after that, with the stored starting cell or cell’s name as activity range, read and store the update data table in data type datasheet (Figure 10).

Secondly, the UiPath robot will execute the browser process which is looped by “For Each” activity with engine ids reading from Excel file (Figure 11).
Inside each “For Each” activity loop, the UiPath robot will go to the main configuration page, find the filtering section and filter out one engine with one engine id, then go to the detailed information page of the required engine (Figure 12).

At the beginning of building the solutions, due to the small number of update modes, UiPath “if” activity would be a simple and clearly understandable method. However, the method was upgraded to “Switch” activity which will present more
manifest and organized expressions. With “Switch” activity and information read from the Excel file, the UiPath robot will automatically choose the update mode by clicking on a different button (Figure 13).

Figure 13. UiPath process of choosing update mode

After choosing the update mode, the page will turn to the edition mode. Another “For Each” will be used to find and type the update data into all the required textboxes which is an important methodology through the requirement – how to find the textbox with a field name which may have a different value with the “name” attribute of the same textbox and how to input plenty of different data to the different matched textboxes.

To solve the first matter for each field name loop, “Anchor Base” activity is a suitable method to be applied. Using “Find Element” as the anchor and “Get Attribute” as the target, considering the practical situation, the anchor position should be set as “top”. Generally, it means the UiPath robot will execute the anchor activity first and run the target activity on the bottom position of the anchor on the same page. In this case, the UiPath robot will run the “Find Element” activity to find the field name and run the “Get Attribute” activity on the bottom position of the field name to get and store the “name” attribute of the textbox (Figure 14).
Figure 14. UiPath process of get textbox attribute with field name

To lock the “Get Attribute” activity and get the attribute only from the textbox type, set the “input” tag as the value of activity selector (Figure 15).

Figure 15. UiPath process of “Get Attribute” activity selector

As for the second matter, I adopted a dynamic selector to locate and relocate on the matched textbox by searching and overwriting the “name” attributes stored in a variable for each field name loop and type the data into the found textbox (Figure 16).

Figure 16. UiPath process of inputting data into the matched textbox
To find the correct data position and get the matched data, a counter is used to count the loop, which also is regarded as the data position in the data table. The row counter is counted for each engine id loop, the column counter is counted for each field name loop (Figure 17).

![Image of column and row counter](image1.png)

**Figure 17.** UiPath process of column and row counter

With the UiPath “For each” activity loop, the UiPath robot can input one data in one routine until it automatically recognizes the end of the array or data table. Therefore, it requires accurate formatted Excel files.

At the same time, the UiPath robot can recognize the empty update data and jump over the routine by “If” activity with “string.isNullorEmpty” function in the condition (Figure 18).

![Image of checking if current data is empty or null](image2.png)

**Figure 18.** UiPath process of checking if current data is empty or null

After running all the field name loops on the edition page, the UiPath robot will click on the “Save” button and run into the next engine id loop (Figure 19).
Figure 19. UiPath process of saving the updates

At the end of engine id loop, if there’s no error occurs, the UiPath robot will be shut down automatically and jump to UiPath application interface.

4.2 HB&EG&CA system data 1cylinder batch update (normal/draft)

This part introduces the methodology of HB&EG&CA system data 1cylinder batch update in normal mode or draft mode.

4.2.1 Requirement Specification

This task is to update a group of newly generated technical data which is separated to multiple datasheets into the 1cylinder data table with a group of engine ids per time on “1cylinder Batch Update” web-based technical database maintenance application.

4.2.2 Solution method

According to the update instruction and requirement, this solution will also cover five sections (Figure 20):

- Reading data from formatted Excel file
- Storing data in UiPath
- Inserting update message with certain update mode with specified username for UiPath robot
- Updating data to the web-based technical database maintenance application
- Handling appearance of the error.
First of all, the general view of this solution is divided into two ordered processes: Excel process and web-based database browser process (Figure 21).

**Figure 20.** Working flowchart for 1cylinder table batch update (normal/draft)
Figure 21. General UiPath process of 1cylinder batch update solution

The first step after running the UiPath is to read and store the update the starting cell or cell’s name and other information in the information type datasheet, then with the stored starting cell or cell’s name as activity range, read and store the update system name and browser Uniform Resource Locator (URL) into UiPath variables (Figure 22).

Figure 22. UiPath process of getting information ranges from Excel file

After that, the UiPath robot will go to a “For Each” activity loop with update system names read from the Excel file. One update system per loop should be
updated. Inside each update system name loop, the UiPath robot will find the data type datasheet with the update system name passed as the datasheet name, read and store the update data table in UiPath (Figure 23).

![UiPath for each sub system](image)

**Figure 23.** UiPath process of for each update system name loop

Then run the browser process. For each required group of engine ids, the UiPath robot will go to the main configuration page, find the filtering section and filter out the engines with engine ids and engine description, then go to batch update page (turn to a new tab) and have further configuration (Figure 24).

(One group of engine ids is stored in the Excel file as a string, when the UiPath robot needs to find the maximum and minimum number of engine ids which will be needed in the filtering section, I use the “split” function to split a string into an array of strings, and convert the items in an array from the string type to the integer type so that the UiPath robot can do the plus and minus calculations before typing ids into the filter)
Figure 24. UiPath process of for each group of engine ids loop

Before the UiPath robot operate activities on the batch update page, setting a new navigation is necessary (Figure 25) (As the navigation of old tab is from “Open Browser” activity, if there’s no new navigation towards the new tab, then all the activities will still automatically try to find the element with selector on the old tab which will have a wrong process direction or cause the error.)

Inside the batch update page, the UiPath robot will select the 1cylinder update option, update system name and update mode from Excel though the “Switch” and “Select item” activity (Figure 26).

Then confirm the information and go to 1cylinder batch update edition page which again has another different navigation, because the update system name is included in navigation and will be changed for each update system name loop, here I use a dynamic selector with update system name as a variable and pass to the “Attach Browser” activity selector to present navigation of 1cylinder batch update page in each loop (Figure 27).
Figure 26. UiPath process of choosing update mode

Figure 27. UiPath process of dynamic HTML page selector

Another “For Each” will be used to find and type the update data in the stored data table into all the required textboxes. I use the “Anchor Base” activity with the “Find Element” as the anchor and the “Get Attribute” as the target, considering the practical situation, anchor position should be set as “left”, the UiPath robot
will run the anchor activity “Find Element” to find the field name first and run the target activity “Get Attribute” on the right position of the anchor activity to get and store the “name” attribute of the textbox (Figure 28).

**Figure 28.** UiPath process of getting textbox attribute with the field name

To lock the “Get Attribute” activity and get the attribute only from the textbox type, set the “input” tag as the value of activity selector (Figure 29).

**Figure 29.** UiPath process of “Get Attribute” activity selector

A dynamic selector is used to type different data into the different matched textbox. To locate and relocate on the matched textbox by searching and overwriting the “name” attributes stored in a variable for each field name loop and type the data into the found textbox (Figure 30).

**Figure 30.** UiPath process of inputting data into the matched textbox
To find the correct data position and get the matched data, the counter is used to count the loop, and is also regarded as data position in the data table. The row counter is counted for each group of engine ids and the column counter is counted for each field name (Figure 31).

![Figure 31. UiPath process of column and row counter](image)

Also, the UiPath robot can recognize the empty update data and jump over the routine by “If” activity with “string.isNullorEmpty” function in the condition (Figure 32).

![Figure 32. UiPath process of checking if current data is empty or null](image)

After running all the field name loops on the edition page, the UiPath robot will click on “Save” button, close the current tab and run the next group of engine ids loop (Figure 33).

![Figure 33. UiPath process of saving the updates](image)
After running all the groups of engine ids loop in one update system name loop, UiPath robot will go to the next update system name loop until the end and close the browser. If there’s no error occurs, the UiPath robot will be shut down automatically and jump to UiPath application interface.

4.3 Error handler and feedback message

The error handler and the feedback message on UiPath are extremely important for solutions.

In some situations, when the UiPath robot faces a processing error but it’s not necessary to stop or shut down the current process, the error handler is required to be implemented.

In all five solutions, there will be similar error handlers for each solution:

- After filtering the engine id(s)
  The error handler will check if the certain engine id(s) is/are existing.

- After checking the current data empty or null
  The error handler will check if the UiPath robot can find the matched attribute with the field name.

- Before typing data into the textbox
  The error handler will check if the current data can covert the value to a specified type.

- After saving the updates
  The error handler will check if the UiPath robot can find the successful update message.

When any of the above situations returns false messages, the UiPath robot will execute the error handler which will continue to execute the next same level loop.

At the same time, to achieve to build a better version of functional solutions, tracing the running process and giving an understandable and analyzable feedback message is also a positive habit for a developer.

I set the “Log Message” activities before every different level of loop occurs with dynamic value and necessary explanation, as well as inside and outside of every error handler reporting an error message or a successful update message.
5 DISCUSSION AND FUTURE DEVELOPMENT

In this thesis work, data entry update instructions and UiPath solutions are the two main generated products which achieved all the required demands under the goal of great efficiency, low appearance of system error and scalable solution structure.

Through the process of building UiPath solutions, I faced quite many logic issues and I realized that I was lacking in the knowledge of function libraries in programming languages which are supported by UiPath. Also, a great amount of ideas came to my mind when building the UiPath solutions and learning about UiPath activities and its back-end code.

Substantially, all the products generated from this thesis work (5 UiPath solutions, solution user instruction, 11 update instructions and Excel model) will continue further development in the future and deploy from the testing environment to the production environment.

5 UiPath solutions will be integrated and adjusted to be able to upload on company’s Orchestrator by the RPA development team in Wärtsilä, which means later on the UiPath solutions will no longer run on a personal computer but the formatted Excel files will be sent to RPA development team by email and run on the Orchestrator. Several meetings will be held with PRA experts to discuss the future possibilities on UiPath solutions.

Meanwhile, matched update instructions will be provided together with solutions to the RPA experts as solution structure references. Also, update instructions are also created for instructing update methodology to non-knowledge employees.

In the later development, there still exists plenty of methodologies which have potential possibilities to be improved and performed:

- Separating similar sections in UiPath solution from one-piece solution to several common and specified modules
- Integrating more adaptable and common format of Excel file with less formatting worktime for both testing laboratory and our team
- Giving more meaningful feedback during processing
• Keeping running the UiPath process instead of breaking the loop when the current edition is locked by the current user
• Setting the source data and method to be able to remove unnecessary data by NULL
• Creating separately the info and error log message from Output section on UiPath to a txt file named by local date
• Analyzing the source Excel file in a general view to discuss and reach a usable agreement for common format of source Excel file and reporting the changes of newly updated parameter from performance experts
• Create a process for validating the updated data on web-based technical database application with approved test data from source Excel file automatically by UiPath robot

After researching the whole thesis work, I drew a conclusion from validating the full functional UiPath solutions that RPA technology leads data entry workflow in our team into a more advanced and digitalized way of working. I believe it has a great potential to become a more and more common trend in apply RPA technology into the daily work at Wärtsilä.
REFERENCES

/1/ This is Wärtsilä. Accessed 20th Dec 2018
https://www.wartsila.com/about

/2/ Wärtsilä corporation. Accessed 20th Dec 2018

/3/ Arto Järvi. Accessed 16th Apr 2018
A short message from an authoritative manager.

https://www.newscientist.com/article/mg22630151.700-ai-interns-softwarealready-taking-jobs-from-humans/#.VY2CxPIViko


/7/ RPA, cutting through the noise. Accessed 12th Mar 2019

/8/ How is RPA different from other enterprise automation tools? Accessed 11th Feb 2019


/10/ With India Focus, Global Robotics Company UiPath Raises $225 Mn. Accessed 8th Feb 2019
https://inc42.com/buzz/with-india-focus-global-robotics-company-UiPath-raises-225-mn/

/11/ We are UiPath. Accessed 8th Feb 2019

/12/ Why RPA UiPath is unique RPA software? Accessed 10th Feb

https://www.zarantech.com/blog/why-rpa-uipath-is-unique-rpa-software/