The Process to Design an Automation System

Le LIU
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100086, China
Email: liule200@sina.com

Abstract. Nowadays, automation has already entered human’s lives. And here comes the problem, how could we design a valuable, usable and reliable automatic system? We need to determine which function owns the high-rights and which owns the low-rights. As for the type of functions, this paper proposes that automation can be applied to four classes of functions: 1) information acquisition; 2) information analysis; 3) decision and action selection; 4) action implementation [1]. This paper also introduces a new model of automatical system, called automatical window system, using the previous methods and information. The complexity of automation design requires high logical process, related techniques and real execution. Every single part matters and should be clarified before starting.

1. Introduction
Nowadays, human believe the automation more and more. Because of the technical development of hardware and software, automation has already replaced numerous aspects of human-machine system. And here comes the problem, how could we design a valuable, usable and reliable automatic system?

The purpose of automatic is to make a software replace the human works to make people feel more convenient. Therefore, the most obvious thing is the automatic software need to face all the requirements from human. We make a software and make the pc like a real man who can help us for our events and can communicate with us. And here comes the challenge, replying the diversity of conditions and requirements.

In the other hand, we should think about the inference that the system make automatically. We may design several recursion projects or several judgment project to classify the situations and the requirements. Making the result of the software direct to the need of the users.

Moreover, testing is also a huge part of designing a system. And it may cost a lot of time and resource. Therefore, can we also make the testing automatical?

To summarize the problems, we should focus on two main questions:
1) How many functions can we automated? And what functions should be automated?
2) What levels should we define to every function?
To solve the problem, we should think about the process of automation.

2. Automation Level
Obviously, automation not only simply supplies our daily activity, but also makes change, or, to some degree, makes human life better. It takes over the tough job, replaces the human labor and help to save the resource.

Nevertheless, can we let the automatic system make decisions in any situation or at any time? The answer is no. Machines can work efficiently, but they don’t have emotion. Sometimes, it’s extremely
dangerous. For instance, if there’s something wrong with the system and people haven’t designed an exception handler. It may start to hurt people and can’t be stopped.

However, if we totally distrust the automation and do every simple thing by our hands, the speed of development will be slow down. Therefore, we can define the levels of automation. Of course, different parts are in different levels and granted different rights.

After make it clear in our mind, we should determine which function owns the high-right and which owns the low-right. The biggest difference between the functions is the difficulty, or, we can also call it ‘risk’. ’ High-risky’ means error is likely to happen. In this situation, it’s necessarily to inform human and invoke the exception handler. As for the low-risky functions, we can make it play the major role and inform the human when making decisions or facing errors. In a word, we may grant the high-risky functions low-level automation, while low-risky functions have high-level automation.

| Table 1[1] | Levels of Automation Definition |
|------------|--------------------------------|
| HIGH | The computer decides everything, acts autonomously, ignoring the human. |
| 1. The computer offers a complete set of decision/action alternatives, or |
| 2. The computer offers a complete set of decision/action alternatives, or |
| 3. narrows the selection down to a few, or |
| 4. suggests one alternative |
| 5. executes that suggestion if the human approves, or |
| 6. allows the human a restricted time to veto before automatic execution, or |
| 7. executes automatically, then necessarily informs the human, and |
| 8. informs the human only if asked, or |
| 9. informs the human only if it, the computer, decides to |
| LOW | The computer offers no assistance; human must take all decisions and actions. |

And table 1 [1] shows a 10-point scale, with higher levels representing increased autonomy of computer over human action, based on a previously proposed scale.

3. Automatic Function
Before trying to automate a system, we should divide the big project to several simple parts. We all know that a system consists of different types of functions. The requirements to a software system can be divided into two types: 1) functional requirements, and 2) non-functional requirements. Table II [2] gives the classification rules. Similarly, the functions also have their properties. Suppose that we can also define all the functions into two big sets: functional and non-functional.

3.1 ‘functional’ functions
Functional requirements are related to the system’s behavior, data, input, or reaction to input stimuli, regardless of the way how this is done.

We may propose that automation can be applied to four classes of ‘functional’ functions [1]: 1) information acquisition [1]

Comparing to the other classes, this might be the most low-risky one. Information acquisition applies to the sensing and registration of input. For instance, the sensors of vehicles are doing this job. The sensor collects the status and the condition. When there might be an accident, the sensor will send the danger signal to the controller to operate.
2) information acquisition [1]

This part is more difficult than the first part. There might be many situations. So, we have to consider about all cases and the solutions. Before giving the result to controller, the system should analyze the data, and classify it. For instance, if it is extremely serious? serious? or general?

3) decision and action selection [1]

This is the most important and the riskiest part. The system should decide which may exert a tremendous influence. The system should make the best choice. Therefore, when arriving at this step, the automation level of the system should be lowered.

Moreover, we can get the degree of severity from the last step. And we can reduce the level of automation according to the severity degree. For instance, if the situation is extremely serious, we should vary the automation level to level-3: narrows the selection to a few. It means human will take over the system and make the decision and the selection. However, if it’s not that serious, we can vary the automation level to level-6: allows the human a restricted time to veto before automatic execution.

In this way, we can maximize efficiency, at the same time, make sure the security and reliability.

action implementation [1]

The final stage of action implementation refers to the actual execution of the action choice. Automation of this stage involves different levels of machine execution of the choice of action, and typically replaces the hand or voice of the human.

What needs special attention is that the exception handler is highly necessary in the action implementation. When there’s error, human must have authority to pause the system, or even terminate it.

3.2 ‘non-functional’ functions

Definition. A non-functional requirement is an attribute of or a constraint on a system.

Non-functional requirements is 1) restrictions about timing, processing or reaction speed, data volume, or throughput, 2) a specific quality that the system or a component shall have, 3) any other restriction about what the system shall do, how it shall do it, or any prescribed solution or solution element [3].

Even if we cannot say a function is non-functional, there’re still some function which aren’t related to the data, input, or reaction to input stimuli.

So, there’s also two ‘non-functional’ functions:

1) automatic testing

Since we have two types of functions, we should test automation in two ways: Primary Evaluative Criteria and Secondary Evaluative Criteria [1].
The human performance consequences of specific types and levels of automation constitute the primary evaluative criteria for automation design using the model. Secondary evaluative criteria include automation reliability and the costs of action consequences.

In simple term, primary evaluative criteria is the evaluation for ‘functional’ function, while secondary evaluative criteria is the evaluation for the ‘non-functional’ function.

2) adaptive automation
This part is responsible for the change of automation level (we’ve discussed before).

4. Process of Automation

Fig. 1 shows the process of automation designing. It consists dividing, type classification, level classification and test.

In a more practical way, we can write a pseudo-code. The code can be found in Appendix A.

5. Example

5.1. Introduction
Nowadays, air pollution is getting more and more serious in a lot of area, especially some developing countries. For instance, China’s economic grew in the last decade and increased numbers of vehicles in the cities. More and more factories have been built so that the air condition has been getting worse and worse. Even if the government tried many kinds of methods to improve the current situations, there’re more and more people getting sick because of the air pollution every year.

In October 1948, the city of Donora, Pennsylvania(USA), 14,000 residents experienced severe respiratory or cardiovascular problems. The death toll during this incident rose to nearly 40 [4].

In September 2016, the World Health Organization (WHO) released a report whose key was ‘Air pollution is a major environmental risk to health’.

Figure 1 Process of Automation designing
In 2012 WHO released another report which concluded that “ambient (outdoor) air pollution is carcinogenic to humans, with the particulate matter (PM) component of air pollution most closely associated with increased cancer incidence, especially cancer of the lung”.

According to the research, the main harmful substance of air pollution are:
- Particulate matter (PM), PM10 and PM2.5,
- Ozone (O3),
- Nitrogen dioxide (NO2) and nitrogen oxides (NO, NOX),
- Sulfur dioxide (SO2),
- Carbon monoxide (CO) [4]

So, I’m eager to create a simple automation system that can reduce the harm, the automatical window system.

5.2. Function designing
As I’ve mentioned before, there’re four types of functions that we have to focus on: 1) information acquisition, 2) information analysis, 3) decision and action selection, 4) action implementation. Let’s think about them one by one.

1) Information acquisition
As for this part, we can design a sensor to detect the air condition both outdoor and indoor (to compare later). The sensor should record the content of PM10, PM2.5, O3, NO2, NO, NOX, SO2 and CO. The reports said air pollution is caused by vehicle and industry, then the outdoor air must be more dirty than the air indoor. In this case, why should we measure the content of air indoor?

I have seen a research years ago. Many children and even adults died of gas poisoning. If the sensor can detect the danger, many lifes can be saved.

Moreover, the sensor has to check the status of all the windows in the house.

After recording and checking, the sensor will deliver the data to the controller.

To be attentioned, if the sensor finds the content of SO2 or CO is extremely high, it can send a ‘DANGER’ signal directly to the decision-making function (skip the information analysis function) in order to save time.

Obviously, it’s not a high-risky function so that we can grant it a ‘level 8’ automation, which means the system will inform the human if asked.

2) Information analysis
After receiving the data of air condition and window status, controller starts to analyze the data.

First of all, there’ll be a database that contains all level of air condition (6-points). Table III shows an example.

According to the table, we can give the score to both outdoor air condition and indoor air condition. Then the calculator in the controller will compare two conditions, give the result to next function.

Similarly, it’s not a high-risky job either so that the automation will be at level 7, which means the system executes automatically and inform the human necessarily.
### Decision and action selection

For this part, we will consider two situations: extreme danger and normal control.

If the sensor has sent a ‘DANGER’ signal, the status of all the window will be open immediately to prevent gas poisoning.

Otherwise, we can only consider the normal situation. If the score of air indoor is higher than air outdoor, the system will decide to close several windows automatically. And who will decide how many windows to close? The answer is the difference value of two scores. For example, when the value of outdoor_score equals to 4 and indoor_score equals to 5, we’ll only close two windows. However, when the value of outdoor_score equals to 2 and indoor_score equals to 5, we’ll close six windows.

In addition, in the real operation, the system’ll close or open the windows by changing their status.

If it’s a extremely dangerous condition, we have to open the windows as soon as possible. Obviously, automation is much faster than human action. Therefore, the automation system will at level 7, which means the system executes automatically and inform the human necessarily.

Otherwise, the automation system will at level 5 which means executes the suggestion if human approves.

4) Action implementation

As for this stuff, we also grant it a ‘level 8’ automation.

### 5.3. Process designing

Fig.1 already shows the process flow chart of the automatical window system.

And to be more practical and technical, let’s write the pseudo-code. The code can be found in Appendix B.

### 6. Challenging and Improcement

### 6.1. Challenging

1) When we operate the system, the current data is not the only thing we have to consider. Actually, the history data is also important for analysis and maintaining.

2) As I just mentioned, maintainability is also a big challenge we should face. How to detect the error? How to determine the exception part? And how to fix it? They’re all serious questions we need to think about.

3) It is a little bit hard to make a test cases in this system. The only way to test the ‘functional’ and the ‘non-functional’ functions is implementation.

4) When granting the functions, it’s important to do tradeoff (which we can also improve). On the premise of to ensure the reliability, it’s significant to increase efficiency as much as possible.
5) Since the research about the air pollution is still being updated. Similarly, the automation and the database should be updated with the synchronization of current research.

6.2. Improvement

1) It will be better if there is a database to store all the history data, to analyze the recent air condition in this city and maintain the whole system.

2) To make sure or maintain the validity of the sensor, we can connect with the government’s air quality monitoring database, access the history data and compare with our own data automatically. If the difference is bigger than 20ppm, the sensor needs to be corrected.

7. Conclusion

When we try to design a automation system, it’s necessarily for us to consider the type of functions and its level of automation in order to balance the reliability and the efficiency. And every situation should be considered, otherwise, the system will be confused.

In addition, when designing the automation, we not only attempt to design the technical stuff, but also need to combine with the reality. We need to evaluate the ability of automation. Is it able to make this decision? Or is it better to keep the right for human being? Is there any danger handler to deal with the extreme situation? The goal of automation is to make benefits for human rather than hurt ourselves.

In a word, automation designing is a complex process that needs logic, technology and real execution. Every single part should be clarified before starting the work.

References

[1] Raja Parasuraman, Thomas B. Sheridan, Christopher D Wickens, “A model for types and levels of human interaction with automation”. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 30(3), 286-297.

[2] Pascal Raymond, Xavier Nicollin, Nicolas Halbwachs, “Automatic testing of reactive systems”.

[3] Martin Glinz, “On Non-Functional Requirements”.

[4] Athanasios – Valavanidis, Konstantinos Fiotakis, “Air Pollution as a Significant Cause of Diseases and Premature Death. Ambient Air Pollution in Urban Areas...”.

[5] Martin Janík, Michaela Ublová, Štěpánka Kučerová, Petr Hejna, “Carbon monoxide-related fatalities: A 60-year single institution experience”.
