ASM Sc. J., 17, 2022

https://doi.org/10.32802/asmscj.2022.1030

The Development of a Remote Restart Services System in Production Line Using PowerShell Script

Hapini Awang[1], Beh Choon Huat[1*], Athirah Rosli[1,2] and Abdulrazak F. Shahatha Al-Mashhadani[3,4]

[1]*Institute for Advanced and Smart Digital Opportunities (IASDO), School of Computing, Universiti Utara Malaysia, Malaysia*

[2]*Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Malaysia*

[3]*Azman Hashim International Business School, Universiti Teknologi Malaysia, Malaysia*

[4]*Faculty of Business, Sohar University, Oman*

The production line is an important part of most manufacturing industries. However, the services in the devices may experience some issues that disrupt the production line, requiring the assistance of an IT technician to resolve the problem. Therefore, this study seeks to bridge the gap through the design and development of a PowerShell Script to allow the user to remote control the services in a device such as computers and servers. Using this program, the production operator can solve the issue immediately rather than wait for an IT technician, thus increasing productivity and performance. The program design and development followed the Rational Unified Process (RUP) methodology. First, the functional requirements were collected through interviews and content analysis. Later, a prototype called Remote Restart Services System (RRS) is developed using PowerShell programming language on the Windows PowerShell ISE platform. An evaluation was later conducted to assess its usability. The findings suggested that RRS is practical and easy-to-use. The respondents were also pleased with its functions. The field study helps developers to refine the framework by providing direct input from the target audience. It can be a guide to identify areas for improvement and eventually help realise the goals.

Keywords: Remote Services; PowerShell; Production Line; Rational Unified Process; Manufacturing

I. INTRODUCTION

In general, global manufacturing companies are involved in the provision of logistic services and supply chains. These services involve electronic sourcing, supply chain management, supply chain solutions, logistics, advanced planning, standardised global systems, business analytics and many more (D'Aveni, 2017). The production line is a vital part of most of the manufacturing industries. Usually, the operation is controlled by an application that executes automatic tasks, listens for data requests from other software, or reacts to hardware events. These services are often automatically loaded at start-up and operate in the background with no user interaction in an operating system.

However, under some undesirable circumstances, the device's services might experience issues from time to time, affecting the production line and requiring the assistance of IT technicians to resolve the problem. Because of this issue, there is an urgent need for a system that can effectively enable users to remotely control the services from afar using a device such as a computer and a server. Using this system, the production operator will resolve issues quicker and not wait for an IT technician, resulting in increased productivity and performance. This system should be designed so that non-

*Corresponding author's e-mail: bch970130@gmail.com trained IT staff will easily enable remote control (Shahrin *et al.,* 2019; 2020). Hence, a Remote Restart Services System (RSS) is developed and evaluated based on the user requirements. This study will help developers comprehend the system requirements and serve as a guide for further enhancement of the system in the future.

II. BACKGROUND OF THE REMOTE RESTART SERVICES SYSTEM

Remote Restart Service (RRS) is developed using Windows PowerShell, allowing users to remotely control the services through computers and servers. This system is embedded in admin accounts and will permit users to start, stop, and restart the services. To control any devices in the production line, users only need to search for the device they want to control, and the system will show all the services of that particular device. If the device is valid, then the admin accounts will be granted access permission. Furthermore, users can search for and select the services to start, stop, or restart. Following that, the system will generate a file and record it each time it is used.

A. Tools for RRS Development

To develop RRS, two tools have been used. The two tools are Windows PowerShell and PS2EXE. Windows PowerShell is a configuration management framework and task automation by Microsoft, comprising a command-line shell and the related scripting language. Initially, it was used solely as a Windows module, known as Windows PowerShell, but it was made open-source and cross-platform on 18th August 2016 with the launch of PowerShell core. The former is developed on the .net framework, while the latter is on the .net core (Cwalina and Abrams 2008).

On the other hand, PS2EXE is a free tool for converting PowerShell scripts into executable files. This tool can also help prevent other users from gaining access to the scripts' embedded admin account (Lancaster, 2017).

III. METHODOLOGY OF THE STUDY

The study was done using Rational Unified Process (RUP) methodology by Rational Software Corporation. RUP offers a systematic method to assign tasks and assignments within a development organisation (Kruchten, 2004). Its objective is to guarantee great-quality computer software that meets up its end-users requirements in a foreseeable timetable and cost. It divides the development process into four distinct phases: inception, elaboration, construction, and transition. Figure 1 shows the process of RUP, while Table 1 elaborates the method and expected outcome at each cycle.

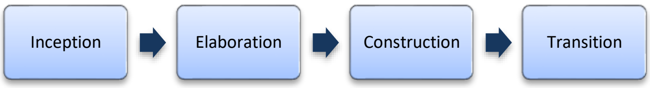


Figure 1. Phases of RUP

Table 1. Method and expected outcome for RUP

Rational Unified Process (RUP)

Process	Method	Expected Outcome
Inception	- Identify project title and problem - Find project objective - Find project scope - Create a project proposal	- Project title and problem identified - Project objective identified - Project proposal completed
Elaboration	- Identify techniques that will be used - Identify tools that will be used - Find functional requirement - Create a UML diagram based on functional requirement	- Techniques that will be used are identified - Tools that will be used are identified - Functional requirement identified - Completed UML diagram - Use case diagram - Sequence diagram - Activity diagram
Construction	- Create a graphical user interface - Build the system - Create a prototype - Test the system	- Graphical user interface designed and implemented - Complete system with source code - Prototype completed and ready for testing

		- Receive user feedback based on testing
Transition	- Identify errors in the system - Modify the system based on user feedback - Create a project report	- Errors are being identified and removed - Complete the system based on user feedback - Project report completed and sent to the evaluators

IV. DESIGN AND DEVELOPMENT OF RRS

At present, the RRS system is in the transition phase, where the system is under constant testing and modification. Hence, this article explains the process of designing and developing a system for remote restart service following the second and third phases of RUP. The second phases of the RUP are the elaboration phase, while the third phase is the construction phase.

A. Elaboration Phases

At the elaboration phase, the techniques and development tools used in the development phase are identified. The technique used is RUP, which is already discussed previously. For the development tools, the programming language used is Windows PowerShell and the tools used in Windows PowerShell ISE.

Next is searching for the requirements of the remote restart services. This procedure is carried out by interviewing the selected system operators and technicians in one of the manufacturing companies in Northern Malaysia. The interview was performed informally at their convenience. They were asked some open-ended questions about the potential system functionality. The examples of the questions are:

- *What are the features of a system for remote control services that you would like to have?*
- *Do you know how to remote restart a service manually?*

Their collective opinions on requirements were recorded. Later, they actively participated in the construction (development) phase, showing them the prototype interface to get their feedback and ideas.

Table 2. List of requirements of remote restart services system

ID	Requirement Description	Priority
1	**Service Device**	
1.1	Users can search for the devices that need remote restart services.	Mandatory
1.2	The system should be able to display the list of the services if the device is valid.	Mandatory
1.3	The system should be able to display an error message when the device is not found or not valid.	Optional
2	**Start Service**	
2.1	Users can start the selected service.	Mandatory
2.2	The system will prompt a message showing whether the service is started successfully or not.	Optional
3	**End Service**	
3.1	Users can end the selected service.	Mandatory
3.2	The system will prompt a message showing whether the service ends successfully or not.	Optional
4	**Restart Service**	
4.1	Users can restart the selected service.	Mandatory
4.2	The system will prompt a message showing whether the service is restarted successfully or not.	Optional
5	**Embedded with Admin Account**	
5.1	The system must be embedded with the admin account to allow only the selected user to start, end and restart services.	Mandatory
6	**Search Service**	
6.1	Users can search for the service.	Desirable
6.2	The system will display the services based on the keyword.	Desirable
6.3	The system should be able to display an error message when the services are not found or not valid.	Desirable

The requirements stated in Table 2 were interpreted into the computer system functionality. The next step was to use appropriate modelling methods and tools to visualise and model the system's needs. The Unified Modelling Language (UML) is employed to envision and model the requirements in this work. The models applied in this study are two behaviour diagrams, which are use case and activity diagrams representing the structural elements of the system. These diagrams were illustrated using Draw IO. Figure 2 shows the use case diagram and the connections between the use cases and RSS users. Six major use cases are: (i) search for device, (ii) start service, (iii) end service, (iv) restart service, (v) embedded with the admin account, and (vi) search service.

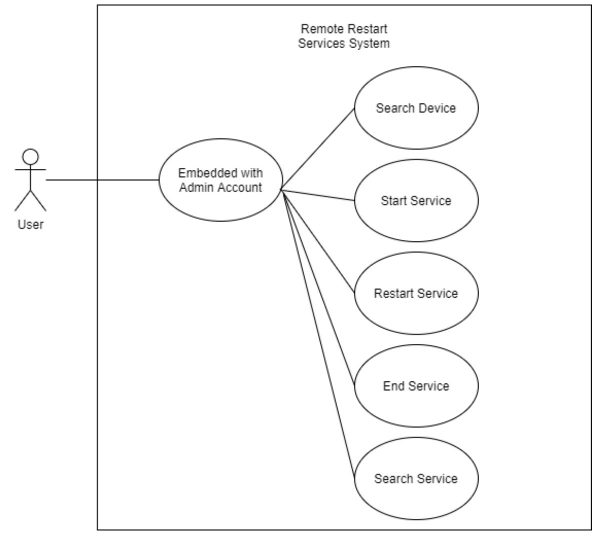


Figure 2. The use case diagram of RSS

The use case diagram is detailed to depict the system's dynamic behaviour where the operations concerned in applying the system for remote restart service are demonstrated in an activity diagram of Figure 3.

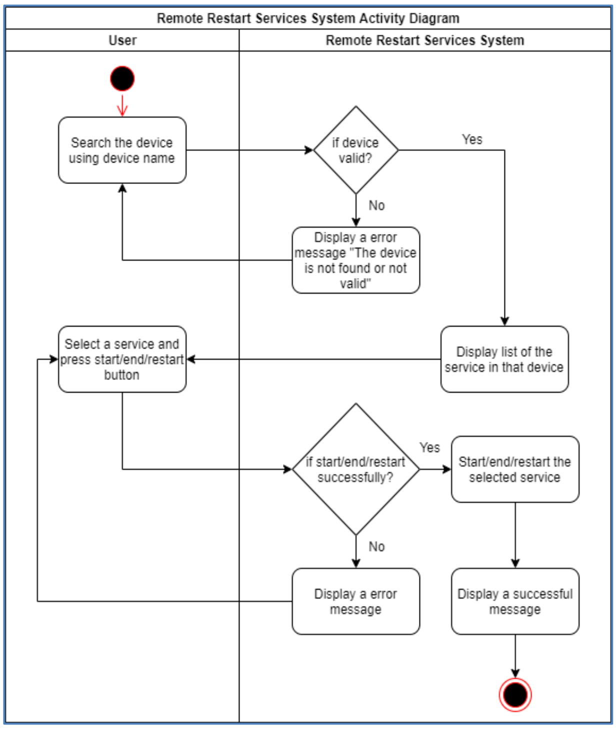

Remote Restart Services System Activity Diagram
User
Remote Restart Services System
Search the device using device name
if device valid?
Yes
No
Display a error message "The device is not found or not valid"
Display list of the service in that device
Select a service and press start/end/restart button
if start/end/restart successfully?
Yes
Start/end/restart the selected service
No
Display a error message
Display a successful message

Figure 3. The activity diagram of a system for remote restart service

V. THE RRS DEVELOPMENT

During the construction phase, the RRS prototype, a computer system for remote restart service, was developed according to the requirements explained in the previous section. Software prototyping is a standard method for demonstrating software requirements that allow the developer to obtain remarks and recommendations from users based on their experience interacting with the prototype. Windows PowerShell ISE is employed as the primary tool of the integrated development environment (IDE). In addition, the PS2EXE is also used to convert the .ps1 script to the .exe file. This should make it easier for users to use by starting, stopping, and restarting the service. Screenshots in Figures 4, 5, 6, 7, and 8 show the selected interfaces of RRS.

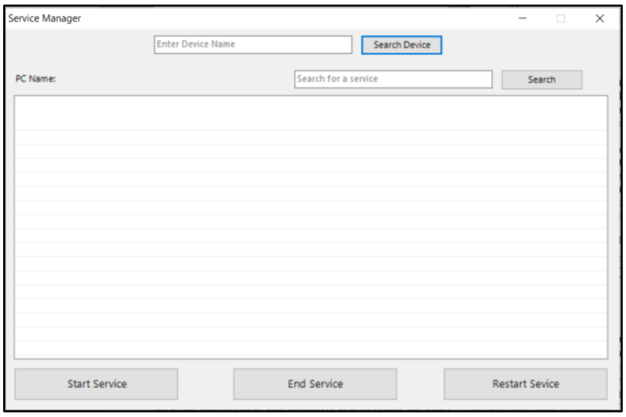
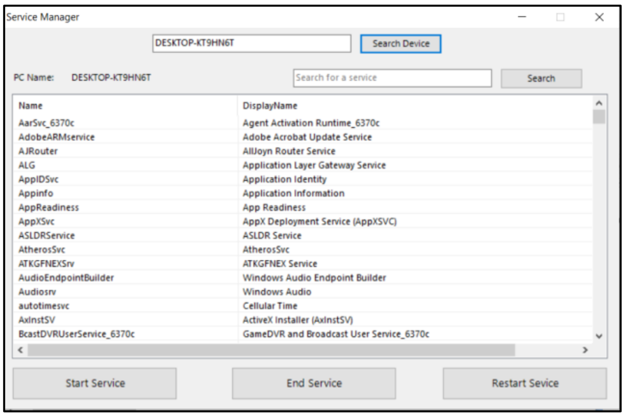


Figure 4. The main interface of RRS

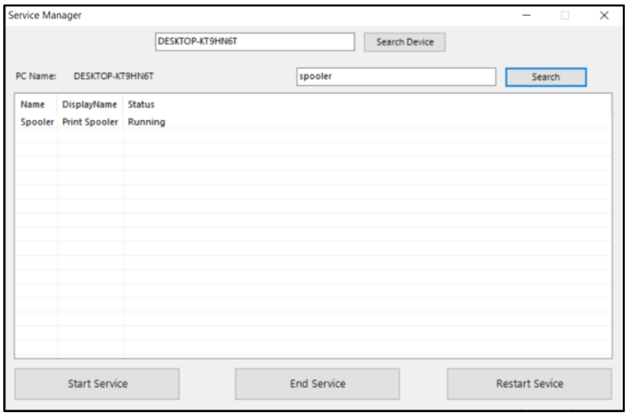


DESKTOP-KT9HN6T

PC Name: DESKTOP-KT9HN6T

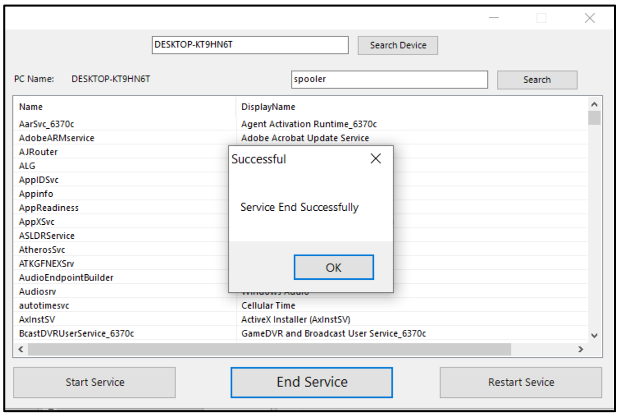
Name	DisplayName
AarSvc_6370c	Agent Activation Runtime_6370c
AdobeARMservice	Adobe Acrobat Update Service
AJRouter	AllJoyn Router Service
ALG	Application Layer Gateway Service
AppIDSvc	Application Identity
Appinfo	Application Information
AppReadiness	App Readiness
AppXSvc	AppX Deployment Service (AppXSVC)
ASLDRService	ASLDR Service
AtherosSvc	AtherosSvc
ATKGFNEXSrv	ATKGFNEX Service
AudioEndpointBuilder	Windows Audio Endpoint Builder
Audiosrv	Windows Audio
autotimesvc	Cellular Time
AxInstSV	ActiveX Installer (AxInstSV)
BcastDVRUserService_6370c	GameDVR and Broadcast User Service_6370c

Figure 5. Display the list of services while the search device is valid



Name	DisplayName	Status
Spooler	Print Spooler	Running

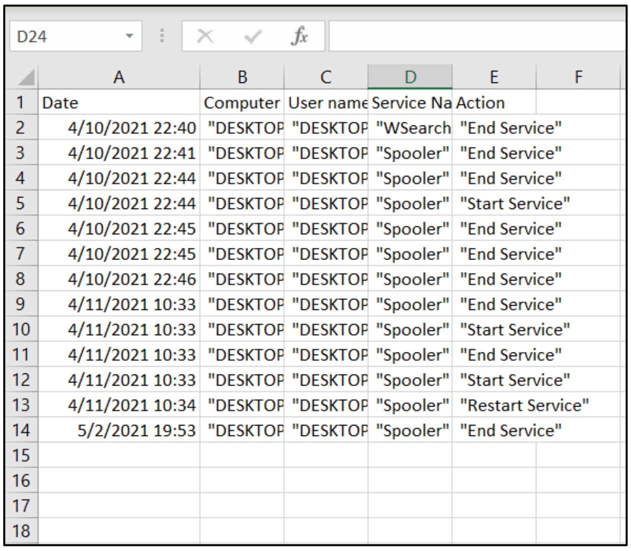
Figure 6. Search function



Service End Successfully

Name	DisplayName
AarSvc_6370c	Agent Activation Runtime_6370c
AdobeARMservice	Adobe Acrobat Update Service
AJRouter	
ALG	
AppIDSvc	
Appinfo	
AppReadiness	
AppXSvc	
ASLDRService	
AtherosSvc	
ATKGFNEXSrv	
AudioEndpointBuilder	
Audiosrv	
autotimesvc	Cellular Time
AxInstSV	ActiveX Installer (AxInstSV)
BcastDVRUserService_6370c	GameDVR and Broadcast User Service_6370c

Figure 7. Display successful message when end successful



	A	B	C	D	E	F
1	Date	Computer	User name	Service Na	Action	
2	4/10/2021 22:40	"DESKTOP	"DESKTOP	"WSearch	"End Service"	
3	4/10/2021 22:41	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
4	4/10/2021 22:44	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
5	4/10/2021 22:44	"DESKTOP	"DESKTOP	"Spooler"	"Start Service"	
6	4/10/2021 22:45	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
7	4/10/2021 22:45	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
8	4/10/2021 22:46	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
9	4/11/2021 10:33	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
10	4/11/2021 10:33	"DESKTOP	"DESKTOP	"Spooler"	"Start Service"	
11	4/11/2021 10:33	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
12	4/11/2021 10:33	"DESKTOP	"DESKTOP	"Spooler"	"Start Service"	
13	4/11/2021 10:34	"DESKTOP	"DESKTOP	"Spooler"	"Restart Service"	
14	5/2/2021 19:53	"DESKTOP	"DESKTOP	"Spooler"	"End Service"	
15						
16						
17						
18						

Figure 8. A record file will be created to save the past actions of the user

VI. THE EVALUATION OF RRS

A. *The Evaluation Setting*

A usability assessment was performed on 10 respondents, consist of technical team members of the same manufacturing company. Due to the work-from-home mode throughout the Movement Control Order (MCO) during the COVID-19 pandemic in Malaysia, the participants took part in this RSS evaluation using Microsoft Teams. They were provided with the RRS prototype, and the instrument used in the assessment is an online questionnaire survey. There are two parts to the questionnaire. Section A queried the respondents' demographic characteristics, while Section B is designed to gather their opinion about RRS. The agreement is measured in a five-point Likert scale, where one (1) denotes strongly disagree, and five (5) means strongly agree. The respondents follow the instruction inside the survey to evaluate the system: (1) Run the rrs.exe application, (2) interact with the RRS system, (3) follow the tasks of using the system, and finally (4) answer the questionnaire.

B. *The Respondents' Demographic Information*

Based on the respondents' demographic analysis, all 10 respondents (100%) are male. For the age groups, the group of 21-25 has 1 respondent (10%), followed by the group 26-30 with 5 respondents (50%), group 31-35 and 36-40 with 2 respondents (20%) while group over 40 does not has any respondent. For the question "*Do you know how to remote restart a service manually?*", 6 respondents (60%) know how to restart a service manually, while 4 other respondents (40%) do not know how to do it. The next question is "*Do you have the experience to write a script?*", 6 respondents (60%) have the experience, while 4 others (40%) do not.

C. *The Usability of RRS*

A descriptive analysis was performed on the respondents' answers in Section B of the survey instrument. The section measured the respondents' experience of RRS usefulness and ease-of-use. It also evaluated their satisfaction with RRS. Tables 3, 4, and 5 described the frequency of the responses. Most of the respondents rated the scales for the three usability facets as either neutral, agree, or strongly agree. On the other hand, none of the respondents rated strongly disagree or disagree.

Table 3. The responses on the RRS usefulness

The usefulness of Remote Restart Services Systems (RRS)	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
RRS is effective	0	0	0	8	2
RRS meets my needs	0	0	1	9	0
RRS is suitable for me	0	0	2	5	3
RRS is useful in overall	0	0	1	4	5

Table 4. The responses on the ease-of-use of RRS

Ease of Use of Remote Restart Services Systems (RRS)	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
RRS is easy to use	0	0	1	5	4
RRS is user friendly	0	0	2	6	2
RRS is easy to learn how to use it	0	0	0	8	2
I can easily remember how to use it	0	0	1	7	2

Table 5. The respondents' responses on the satisfaction of RRS

Satisfaction of Remote Restart Services Systems (RRS)	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I am very satisfied with the RRS	0	0	3	5	2
RRS works the way I want it to work	0	0	4	4	2
I feel I need to have the RRS	0	0	1	5	4
RRS is wonderful and pleasant to use	0	0	2	6	2
All functionality of RRS works well for me	0	0	1	7	2

The result of the evaluation shown that RRS is both useful and easy to use. Besides that, the respondents were satisfied with the feature offered by RRS for remote restart service. The majority of the respondents agreed that the RRS met their needs based on their feedback. In addition, most of the respondents agreed that RRS is easy to learn. They also believed that the RRS system is effective.

Most importantly, the respondents were satisfied with the RRS system. Nevertheless, this study fails to achieve a total agreement as some respondents demonstrated their uncertainties by rating the items neutral. This indicates that further improvements should be made from time to time to improve those aspects of RSS.

VII. CONCLUSION AND FUTURE WORKS

The design and development of a system to remotely restart a service was described in this article. Based on the usability analysis, it can be said that there are still several RRS aspects that can be improved. In the future, the strategy is to upload the record file to the cloud and make sure all related activities will be recorded once the system is used. This will be useful for monitoring, debugging and finding the unsought errors of RRS operation. This study also intends to add other languages such as Malay, Mandarin and Tamil, so that users who do not speak English will be able to use the system.

VIII. ACKNOWLEDGEMENT

This research was funded by Research and Innovation Management Centre (RIMC), Universiti Utara
Malaysia.

IX. REFERENCES

Cwalina, Krzysztof & Brad Abrams 2008, 'Framework design guidelines: conventions, idioms, and patterns for reusable', Net Libraries, Pearson Education.

D'Aveni, Richard 2017, 'Choosing Scope over Focus', MIT Sloan Management Review, vol. 58, no. 4, p. 22.

Kruchten, Philippe 2004, The Rational Unified Process: An Introduction. Addison-Wesley Professional.

Lancaster, Tom 2017, 'Muddying the Water: Targeted Attacks in the Middle East', Palo Alto Networks, Researchcenter, Paloaltonetworks, <Com/2017/11/Unit42-Muddying-the-Water-Targeted-Attacks-in-the-Middle-East>.

Shahrin, S, KAA, Rahman, KM, Kamarudin, R, Che Me, JR, Razali, I, Abidin, M, MM, Aminuddin, I, Mustaffa, MFA, Jalal & MFC, Hassan 2020, 'UDP-TRIZ Model Improvement with a Focus Group Discussion of TRIZ Experts', IOP Conference Series: Materials Science and Engineering, vol. 932, no. 1. doi: 10.1088/1757-899X/932/1/012100.

Shahrin, S, K Rahman, KM, Kamarudin & R Che Me 2019, 'Theoretical Framework of Universal Design Principle (UDP) Approach Considering Haptic Exploratory Procedure (HEP) of Visually Impaired Person in Using TRIZ Integration', in IOP Conference Series: Materials Science and Engineering, vol. 551. doi: 10.1088/1757-899X/551/1/012109.