Usability Evaluation Approaches in the Context of Robot Programming

Niladri Mondal[1] [0000−0002−3573−0703]

Universität Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
niladri.ofcl@gmail.com

Abstract. Usability has multiple aspects and perspectives to it when evaluated under different contexts or scenarios. The limited understanding in the concept or method of usability, developer's resistance towards adoption of usability refinement and cost of performing a usability evaluation for robot programming can affect the satisfaction and efficiency of the final product being developed. This can lead to robots being used and programmed for missions only by the core experts in the field. With the increasing demand of automation in industries to make systems more reliable and accurate to expectations of developers and end users, robot programming should be an achievable target by a larger group of contributors. This can be accomplished by usability evaluation and approaches that can address the required abstraction needed to deliver robot programming interfaces and environments which should deem fit for usage by non-experts. This paper compares and characterises such usability evaluation approaches for robot programming based on efficiency, effectiveness and satisfaction.

Keywords: Usability · Usability Evaluation Approaches · Interfaces · Programming by Demonstration · Robot Programming · Robot Operating System · Robotic Development Environments · Evaluation Approaches.

1 Introduction

Robots are machines that execute a set of tasks which should be programmable with a computer. In the recent days of research and academia, although there has been a huge contribution in this field aiding to which availability of robots in the commercial space has been possible but, in a limited spectrum within industries. The crux in this limitation is that end users have limited knowledge concerning programming the robots being produced or manufactured. Currently, in a big picture, only the field experts are involved in programming and customisation, and this results in production of robots with predetermined sets of actions. Even in the occasion of such platforms being made available to customise and program the robots, these platforms are not enough in the aspects of usability to be credited 'fit to use' for novice end users. We have to make peace with the fact that all users of robots may not be sheer experts in the fields of mechanical, electrical or computer engineering. But, at the same time, automation in various industries is in demand. It is not a surprise given the amount of iterations, via controlled loops, and precision; with which a robot can be programmed to perform the tasks.

To bridge this gap and provide for the demand created, we need to look into the usability perspective of robot programming. A target achievement should be to produce robots which can be programmable even by the end users, to fit their needs within a specified set of allowed tasks. This directs towards production of ubiquitous robots which should be programmable for general tasks. For this, we need to have appropriate usability evaluation approaches, which considers both the advantages and disadvantages of a system being developed in a justified manner.

This paper has discussed robot programming, aspects of usability and eventually the usability evaluation approaches applied on development environments, robot programming approach, robot programming interface and for profiling robot systems. Some experiments conducted in this regard are studied and the outcomes are analysed to evaluate the aspects of usability. Some issues that might arise due to application of usability evaluation approach is also discussed at the end.

2 Robot Programming

Robots are complex machines used in various walks of industries and research. It might be indulged in simple repetitive tasks such as in manufacturing assembly lines or in handling comparatively more complex and intelligent tasks alike robots deployed for space missions. In case of movable robots, understanding the six degrees of freedom (6DoF) and the concept of odometry involved in a fully operational robot plays a vital role. This is needed to operate in three dimensions of the real world. In some cases, robots may not exist in the real world at all and can just be present as a software in algorithmic form. One such example can be forex trading bots. Such software systems can also be referred to as 'Knowledge Robots' or 'bots' [9].

To program such robots and accomplish the tasks intended, robot programmers have to learn and understand a wide range of technologies. Starting with the popular languages used in robot programming like C, C++, Python and Java, programmers should have a grasp on mathematical tools and data analysis. This means that a robot programmer should have a proper usable platform to process data and produce reliable results. These results can be produced from but not limited to hardware and software sensors. Another important platform is *MATLAB* which can be very helpful in analysing and assisting in successful development of a robot system. A good interface developed to program a robot system can significantly reduce the above challenges by handling much of the crucial work in the abstraction of end users.

It is understandable that for managing multiple codes, libraries and message passing tasks among various software and hardware components, we need a platform which can handle the background tasks for us. This can be achieved by using a meta-operating system platform. Such frameworks are available and some of them are namely *ROS*, *CARMEN*, *Microsoft Robotics Studio*, *MOOS*, *Orca*, etc. This paper discusses in brief about *ROS*.

2.1 Robotic Operating System (ROS)

Robot Operating System (*ROS*) is an open source and acts as a meta-operating system for robots to be programmed. Since a robot is a combination and synchronisation of electrical, mechanical and/or computer engineering, it needs some layers of low-level abstractions enabling convenient writing of codes to achieve a mission or task. *ROS* provides a platform with such abstractions (including the low-level controls), message passing among processes and package management.

Since robot programming requires synchronisation and continuum across several systems which might exist in distributed fashion. For this we require a special design of communication systems such that the functions written by robot programmers are actually executed by utilising a remote address space on a shared network. This is known as *Remote Procedure Call* (*RPC*). This actually allows the programmers to write the procedure as if it will be executed locally which means the procedures can be coded without specifying the details for remote interaction. *ROS* provides a synchronous *RPC* functionality with asynchronous streaming of data over topics. This technique is mainly used as a request-response paradigm whereas for handling messages involving many-to-many mapping of components, *ROS* provides a technique with publish and subscribe method.

ROS is paving the way towards development of future robots and that beyond academia. *ROS* being an open source, can be found to be weakly documented on some aspects but otherwise in terms of usability, it forms a firm support over multitude of systems. This makes it a 'fit to use' framework in robot programming. In the realms of robot programming, *ROS* serves to have a general cumulative characteristics which can be reused across multiple projects. For example, if a programmer learns how different components communicate among themselves in an *ROS* framework, this idea can be reused in another robotic project. There is a possibility of co-existing which means that a python coded node can easily work and communicate with a C++ coded node. *ROS* has multiple libraries to support such tasks.

For assistance in developing a project with *ROS* including the language support of C++ and Python, which are used commonly with a *ROS* framework, IDEs like *Eclipse* and *CLion* can be used. Building and running projects using *ROS* requires the *ROS* environment to be set up. *CLion* has an advantage that it includes plugins which help in automatically setting up the environment. Further, other IDEs usually have configuration for this activity. Another easy way to do this can be to run it from an *ROS*-sourced shell. A sample graphical user interface of *ROS* environment incorporated in *CLion* is presented in figure 1.

It has been often studied in practical experience that usability is given lower importance compared to any other factors while programming a robot. It is usually in the hands of experts that can get the robot concerned to complete a

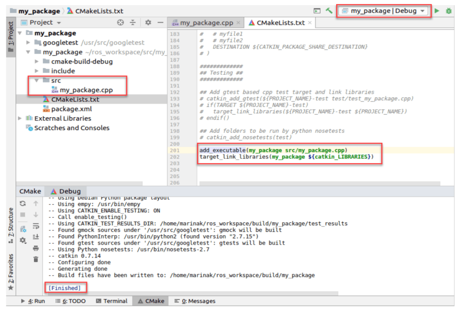

Fig. 1. ROS using CLion [16]

mission or task [2]. This hampers the usability aspects in the sense that other non-experts may not get a hands on to the programming of robots so as to achieve target tasks. Another way of assessing usability in robot programming is to what extent is a robot programmer able to accomplish the tasks that the programmer intends to achieve. If we go at a more detailed level, we can assess if the programmer needed to import any third party external library within the framework to accomplish the tasks. Based on such factors, we might be able to arrive at a conclusion against the usability evaluation of the robot programming.

Since the early days, there has been evolution of multiple robot programming languages and there has been entry of some traditional domain software programming languages in this specialised field [3]. Noticeably, they have all been introduced in increasing fashion of usability in some aspect or the other. However, we cannot conclude firmly that all the aspects of usability are addressed in linearly or exponentially incremental trends as one travels down the timeline of evolution of robot programming languages.

3 Usability

Usability of a system is the extent or degree to which a system can be used [8]. After carrying out sufficient assessments and tests it can be ensured that the system behaves as expected according to the requirements specified [1]. Then to evaluate the usability of the system is the next logical step to be taken. Robots are machines with a blend of mechanical, electrical, and computer engineering which can bring complexities in the aspects of usability. System developers are often occupied intensively into the very process of development to produce a working program. In this process, addressing the user needs and comforts are often overlooked. This builds up a barrier and is referred to as ”developer mindset” [4]. This mindset influences the developers not to prioritise and accept the usability items that should otherwise be significant to the end users. Hence, measuring this mindset is important and can be achieved from the work of Law [20] and Sawyer et al.[21]. The concept of ‘downstream utility’ can be used to obtain *Committed Impact Ratio (CIR)* and *Competed-to-Date Impact Ratio (CDIR)*.

Reproducing from [22], CIR and CDIR measure if developers solve the usability issues before and after implementation respectively. It can be expressed as given below.

$$CIR = \frac{\text{No. of problems committed to fix}}{\text{Total no. of problems found}} \quad (1)$$

$$CDIR = \frac{\text{No. of problems fixed}}{\text{Total no. of problems found}} \quad (2)$$

Going by the definition of ISO, we have three primary aspects to usability:[6]

(a) Effectiveness. Effectiveness is the accuracy and completeness with which users achieve certain goals. In terms of robot programming, we can think of the goal as a robot mission. The standard of a solution and success rate of a system under consideration can be the metrics to discuss this aspect.

(b) Efficiency. Efficiency is understood as accomplishing a certain robot mission by maintaining a trade-off between the accuracy of the solution and its completeness. The hardware, software or physical resources used to achieve the mission is also considered. The time taken to accomplish such a mission can be counted as the metric of efficiency.

(c) Satisfaction. Satisfaction is the user's comfort and positive attitude in operating the system. This is a subjective matter of discussion, however utilisation of attitude rating scales such as *SUMI* can be beneficial in providing the necessary measurements. In this aspect, preference of a system as a whole can be of immense importance for judging satisfaction.

Apart from aspects discussed, safety is another aspect that should be considered for the users. A robot system should be robust and effective enough to be deemed safe for usage. Automatic driverless cars can be considered an example and while benefits are all known, there are multiple areas which should be improved upon. Glitches in released codes, potential to be hacked, failure of mechanical parts, cognitivity with human signals are crucial among other items that should be in the scope of improvement [23].

3.1 Relation among Efficiency, Effectiveness and Satisfaction

All the above three aspects are known to have a straightforward relation among them but that has been done or proven with some high and risky assumptions [6]. While it is easy to draw general relations between efficiency, effectiveness and satisfaction for a given interface or application, at the same time it will depend on multiple issues like context in which the application is being used and the complexity of the task. It also holds a relevant dependency on the experience of the user and the domain for which the application is developed. There can be routine or repetitive robotic tasks for which we focus on efficiency, so that stable and high quality results can be obtained. This may lead to task completion time to decide the usability of the entire system. Another set of tasks can be the complex ones for which completion time plays a secondary role. In such scenarios, finding a feasible or valid solution has to be the target, and hence results of high-quality may not be produced. This shows that the aspect of efficiency might not be the utmost priority in this case.

According to the study conducted by Nielsen and Levy [17], no firm correlation could be established between efficiency, effectiveness and satisfaction. User preference was used as a parameter for satisfaction. After analysing 113 cases extracted from 57 human computer interaction studies, it concluded that the majority of the users preferred a system which they were efficient in using. However interestingly, around 25% of the users did not prefer such a system which they could operate efficiently. Furthermore, even after scrutinising in detail, they could not establish any correlation between effectiveness and satisfaction. Many such studies were conducted but a constructive correlation among the three aspects of usability could not be developed.

The CHI-study by Walker et al [15]: This is the only CHI-study by *Walker et al.* [15], among eight studies concerning measures of three usability aspects, with an analysis of correlations between the three aspects of usability. A comparison of the number of works done till [6] was published and the corresponding aspects of usability with which it deals, has been presented in figure 2 as reproduced from the paper [6]. In this work, spoken language design interfaces to email were studied. It included two designs, one with a mixed-initiative dialogue, where users can flexibly have control over the dialogues and other one with system-initiative dialogue, where the dialogue is controlled by the system. The three aspects of effectiveness, efficiency and satisfaction were considered via the following parameters.

(a) Effectiveness with automatic speech recognition rejects,
(b) Efficiency with number of dialogue turns and task completion time, and
(c) Satisfaction of users using a multiple-choice survey.

Fig. 2. The usability aspects studies: Eight of the studies include measures of all three usability aspects, seven studies measure two aspects, and four studies only one aspect [6].

The outcome of the study indicated that even though mixed-initiative dialogue stands out to be more efficient, which is measured in terms of task completion time and number of dialogue turns, users preferred the system-initiative dialogue. Now if satisfaction is to be used as the primary aspect of analysis, user satisfaction or preferences are not purely determined by efficiency. It was observed that user satisfaction was mostly driven by the effectiveness of the system and thus users preferred a system which was easy to learn, and the one which holds up to the user expectations.

As the system gets more complex, which is quite expected in a scenario where we are concerned with programming of robots, the relation between the aspects of effectiveness, efficiency, and satisfaction can tend to be more disjoint. This is mainly due to the fact that as a system gets more complex, there appears to be significant perspectives to it. Following such facts, it is safe to consider, study and build a robot system keeping one of the three aspects in mind while evaluating the same in terms of usability. From the studies, the three aspects of usability are only weakly correlated and such is true in different domains as well. For robotic systems executing complex tasks, we have to consider the three aspects of usability and the evaluation measures concerning the same only after clearly understanding the application domain and context in which it will be used. The important fact being not to base firm conclusions regarding one of the aspects of usability by just evaluating and measuring the other two aspects.

The usability test by *Walker et al.* [15] is helpful in analysing the usability aspects based on software systems. A robot system often includes operating a movable mechanical part or parts controlled via the software components. Therefore, synchronisation and precision plays an important role in evaluating a robot system otherwise it might threaten physical safety of users. In the development stage, usability of software packages and libraries to operate robotic arms should also be analysed which corresponds to examining cross-disciplinary aspects.

4 Robotic Development Environments (RDEs)

Robotic Development Environments have a significant role to play in robotics but however, there have been limited works suggesting the usability evaluation on such environments. It is necessary to evaluate such RDEs so that the strengths and weaknesses are firmly established for developers to decide which RDE to be used to fit their needs and chalk out trends for future development in RDEs [11]. Robots are systems which deal with real-time and real world constraints such that the sensors and effectors, having particular physical characteristics, need to be controlled. Many RDEs have been developed with the notion to manage the complexities involved, including the message communication among the components of a robot system.

A complete systematic analysis and accounting is hard to establish for all RDEs, because new systems are being released frequently and the number of RDEs available are huge in number. Also, robotics as a field has a vast scope such that we cannot consider every perspective in evaluating the platforms or frameworks. To deal with such problems, usually a specific set of principle constraints are considered. This is done because after considering those constraints, we can make a subset of representatives of RDEs expressing the constraints.

Adopting from [11] a usability evaluation of three such RDEs is done by feature comparison.

4.1 Advanced robotics interface for applications (ARIA)

ARIA is a freely available software package that comes along with MobileRobots robots. MobileRobots robots have to be purchased and since *ARIA* is developed by MobileRobots Inc., it can only be used on MobileRobots platform. This fact limits it's usability because all users will be able to make use of it practically. *ARIA* is a collection of downloadable classes written in C++ and provides the software support for system architecture and has the capability to describe the sensors and effectors involved in the robot system. Also, it provides an interaction implementation between low level hardware components and software. *ARIA* is capable of distributed computing by means of packages acting as wrapper around socket communications. It provides the Simplified Wrapper and Interface Generator tool which supports Java and Python programming languages.

There are some supporting software packages that are available, but are limited in the sense that they are licensed or required to be purchased. Some limited features can be used full-fledged once purchased. There are a lot of open source contributions towards its enhancement which provide functionalities like speech recognition, creating and editing maps, object tracking, etc.

4.2 Python Robotics (Pyro)

Pyro is developed for educational purposes and thus this is a different development environment in terms of use-context. It is a robot programming or developing environment that has been crafted in a top-down approach for designing controllers. It applies abstraction of low-level details, but at the same time it can allow access to low-level details on need basis. Interestingly, it includes wrapping ARIA and thus the components developed or to be developed for *ARIA* can even be re-used in *Pyro*. With reusability in focus, reusable control codes are generated using *Pyro*. An example of this concept can be the use of 'robot units' that replace traditional measurements such as meters and this can be applied across different robots.

Pyro has been developed using the Python programming language. Python being an interpreted language can result in slow execution but it has its own advantages as well. It was chosen because beginner students can learn it faster and with ease of usability. But, this factor does not bound the advanced developers to work on writing advanced levels of programs. An interpreted language was chosen due to its ease of use for beginning students, while permitting more advanced minds to write more complex codes.

In terms of usability evaluation, this can be seen through the aspect of efficiency as it clearly involves re-using code, and the aspect of effectiveness that it successfully abstracts the lower levels.

4.3 MissionLab

MissionLab is a robot development environment built with the goal of executing military task executions. It might involve an individual robot or a team of robots working towards achieving a specific set of tasks. It was developed under the project named DARPA Mobile Autonomous Robot Software (MARS) project. *MissionLab* aims to operate robots in a highly dynamic and unpredictable environment. *MissionLab* can operate robots in groups and such groups are termed as configuration. Collaboration and systematic coordination becomes a very important notion to achieve in a hostile environment. These configurations operate by replicating and aggregating the behaviour of a single robot unit. The Configuration Description Language (CDL) is used to define the configurations. A graphical tool named CfgEdit comes as handy. CDL in turn is compiled as Configuration Network Language (CNL). CNL is then compiled as C++ language and ultimately compiled as a machine language. The basic communication concept consists of client-server model. Developers can make use of the control interface on the server (HServer) side to communicate with robot hardware.

Focusing on usability, *MissionLab* provides a very effective and efficient graphical interface. The interface is good enough even for a non-expert to write control codes without any programming. Log files include information such as position of robot, its velocity, current state of robot in relation to time, etc. Debugging feature allows to display program output to a console so that developers can easily understand the flow of control and if the code logic is at fault. This serves for an easy rectification and debugging of the program code. The inclusion of 'Motivational Variables' make *MissionLab* a more effective way to develop a robot system as it attaches a personality to it.

4.4 RDE Feature Criteria Evaluations

A very useful and meaningful study of different parameters has been presented in the paper by *J. Kramer et al* [11]. The table 1 lists the three platforms discussed and how they fit to the different parameters against which they have been compared. Based on the scores assigned to the parameters we can put forward usability evaluation approaches to the platforms on the aspect of efficiency, effectiveness and satisfaction. The different types of assignments used are as follows. Binary assignment signified by YES or NO. Ternary assignment signified by NS (not supported), PS (partially supported), and WS (well supported). Finally, listings has been used as text description.

5 One-Shot Robot Programming

Programming by Demonstration (PbD) approach can be used to introduce flexibility in robot missions even by non-experts and this is tested against the usability evaluation [2]. In kinesthetic *PbD*, necessary programming information like trajectory and gripper position is given by guiding the robot through a new

Table 1. Feature criteria evaluation by RDE[11].

Criteria	**ARIA**	**Pyro**	**MissionLab**
Programming Language	C++	Python	C++
Graphical Interface	NS	WS	PS
Debugging facilities	PS	FS	FS
Logging facilities	FS	PS	PS
Scalability	PS	NS	FS
Security	PS	NS	NS
Fault-tolerance	NS	NS	PS
Software Integration	NS	FS	NS
Speech Recognition	YES	YES	NO
Vision processing	YES	YES	YES
Learning	NO	YES	YES

task directly. It is during this phase that the robot is being automatically programmed via the movements and hence it can be said that the robot system is learning. User's motion need not be tracked for this method and provision for immediate feedback is possible by the robot system.

In this method, one can make an assessment of intuitiveness and robustness of the robot system and hence usability evaluation on efficiency, effectiveness and satisfaction can be obtained. For this study, questionnaires were constructed and the response of the users were collected. The feedback received were classified in groups representing efficiency, effectiveness and satisfaction. Mental effort required while programming the robot system and overall attitude towards the robot were also mapped to the responses from the users. The responses from users, clearly indicated that *PbD* is intuitive in nature.

In this experiment, the non-experts have been deployed for working on training the robot systems with new tasks. But, it is realised that there were some problems that arose due to the novice users. Since the approach was kinesthetically *PbD*, problems with handling of many degrees of freedom of robot components, very small distances to obstacles and complicated gripper positions. These problems led to the lower quality of task execution which meant a low user satisfaction. This study has been considered to address this problem via robustness. The robustness of this robot system suggests that no matter even if programmed by a non-expert, the robot system must be able to accomplish the task as good as possible when programmed by an expert. The primary goal of the experiment was to study the evaluation of *PbD* approach in the view of usability for non-expert users. In order to chalk out the details properly and define the points of intuitiveness and robustness, five research questions were designed. The questions have been listed reproducing from the paper [2] as below.

(a) How satisfied, efficient and effective feel non-experts after programming the robot?
(b) How much information is necessary to be able to operate the system?

(c) Which expectations have non-experts at programming a robot by themselves?
(d) How satisfied are non-experts with the reproduction of a self-programmed task demonstration?
(e) Which requirements have non-experts for the fulfillment of a task?

Three scenarios of different complexities were considered knowing that the probability of failure increases with increasing levels of difficulty and this leads to a negative user satisfaction rating. The scenarios considered were: **(a)** Repositioning scenario, **(b)** Stacking scenario, and **(c)** Sorting scenario.

The questionnaire was carefully designed so as to establish the intuitive usability of the system and thereby indicating the three aspects of usability. The questionnaire consisted of three parts namely Q1, Q2, and Q3 which were aimed at addressing exactly the three aspects of usability. Reproducing from the paper [2], the questions are listed as below.

(Q1) The first part notes the expectancy value of the user. User estimation of complicity of operating system is recorded. Emotional impressions are also noted because physical contact with the robot system is required in order to program it.
(Q2) The second questionnaire scans the mental and physical effort that a user has to perform in order to accomplish a task. This questionnaire also collects assessments of satisfaction in terms of human computer interaction.
(Q3) The third part focuses on the fact that the task has been accomplished and thus the focus has to be on operating concept.

The study has depicted intuitiveness of PbD due to high efficiency and effectiveness of the approach, while the effort that is required while training the robot is low. Therefore, this programming approach can instruct the novice users to successfully achieve the mission that they intend to. A brief representation of the aspects of usability reproduced from the study in [2] has been presented in figure 3, 4, and 5. This experiment had also established the demands of an intuitive robot programming concept for non-expert users. This study showed that successful users were satisfied with the programming system and approach. After analysing the outcomes of this study and comparing it with a multi-shot programming approach, it hypothetically indicated that robot systems which are highly intuitive in its learning approach, may not perform the same in terms of robustness.

6 Usability Evaluation of VR Interface for Mobile Robot Teleoperation

Virtual Reality is one of the advanced technologies and application areas developed in computer science and engineering. Initially it used to be deployed bounded in the fields of space science and military operation. But, as the cost of technology became cheaper, use of VR interfaces became common especially

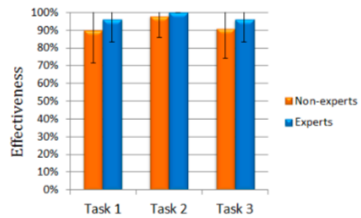

Fig. 3. The rated effectiveness of different tasks. High values represent high effectiveness[2].

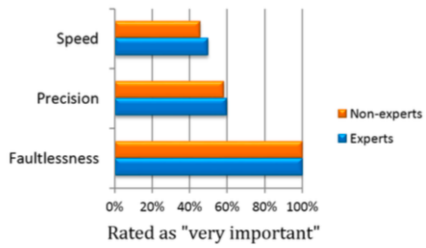

Fig. 4. Participants rating of different requirements regarding the task execution by a robot[2].

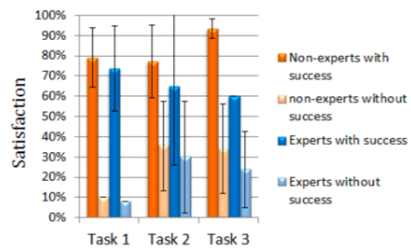

Fig. 5. Participants average rating of satisfaction after completing a task[2].

for robots that can be implemented in environments which can be termed as dangerous for human health or human life or hazardous locations and places which are difficult to explore. Studies have shown that virtual reality techniques applied in the interfaces for robots prove to be increasing the productivity of users. Even the distance evaluation can be done to a good level of accuracy. VR proves to be a factor in decreasing the robot task execution time. This is both effective and efficient in the aspects of usability. Hence, it also reduced the time which is needed to adapt the user to the control interface. Finally, it also increases the level of intuitive control.

The main task of robots which can be deployed using VR are meant for inspection of a given environment, and hence monitoring and recording data becomes a primary task of the robots. An example scenario can be deployment of robots in an unknown topography for searching through rubble after an earthquake [10]. Another important scenario where robots deployed with VR technology has been in the field of medical science such as teleoperation. It is also used in the mining industry with fruitful outcomes. Now when more accurate and advanced systems are developed, the matter of comfort of use and intuitive control takes the utmost concern. The advantages of using VR technology coupled with a movable robot has been studied and presented in the paper [18]. It used a stereo camera which was planted tightly on a robot arm and a mobile platform. The images obtained as a consequence were displayed. In the study by Tran et al. [19], a research study on data gloves was made. Data glove was used as a remote control of an inspection mobile robot by means of the gestures by the user or operator.

6.1 A Study of an Inspection Robot

With the purpose of testing the interface, a low-cost inspection mobile robot was developed [10]. The robot built consisted of three main components. They were namely a mobile platform, a manipulator, and a camera set. The chassis of robot contained caterpillar treads and consists of two motors with gears, a battery, motor drivers, a computer, an electronic board with microcontroller, and two wireless cards used for wireless connection. The manipulator arm, excluding the end-effector, is a 5 Degree of Freedom arm fitted to a robot platform and driven by servo motors. A 2 degree of freedom camera set driven by a servo motor allows rotation of ninety degrees left/right and sixty degree up or down. To study the effect of VR, two comparative interfaces were developed namely 'STEREO' and 'MONO'. The VR interface made use of VR technique equipment, data glove and a joystick. The STEREO interface consisted of stereoscopic instruments and equipment with joysticks. The MONO interface deployed monoscopic instruments and equipment with a joystick. The data glove used in the VR interface system, provided information about the degree of tightening the hand. This information is used to control movement of robotic arms and camera sets. The robot movement is controlled by a joystick. The figure 6 as reproduced from the paper [10], provides a glimpse of the technique used to control the robot.

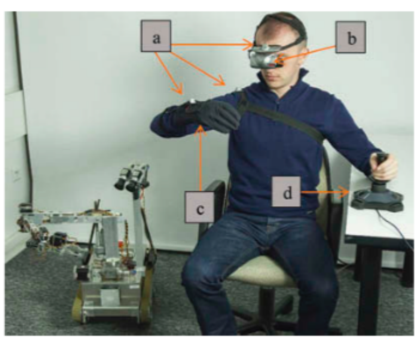

Fig. 6. The VR interface utilising the VR techniques: (a) motion tracking, (b) head-mounted display, (c) data glove, (d) joystick [10]

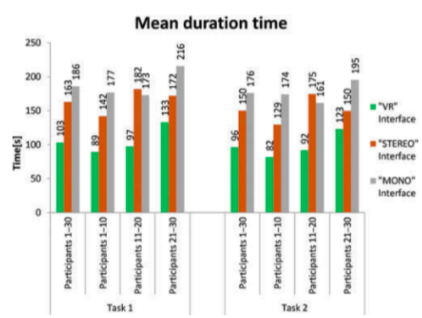

Fig. 7. Mean duration time of completing a task. Note, VR = interface that uses VR techniques equipment and a joystick; STEREO = interface that uses stereoscopic liquid crystal display monitors, stereoscopic glasses, and a joystick; MONO = interface that uses a monoscopic liquid crystal display monitor and a joystick [10].

User Testing method from the work on [13] was adopted for evaluating the interface. The evaluation consisted of multiple cycles with each cycle having the same set of tasks. Objective indicators measuring the user's task completion time and the number of errors that occurred towards accomplishing the tasks were used. To measure the satisfaction and collect user's reviews, subjective sets of questionnaires were prepared. Since the users were non-experts in the area of VR and teleoperation, these data were very crucial as this interface was meant for novice users to accomplish robot missions. The questionnaires were grouped as given below and have been reproduced from paper [10].

(a) Spatial presence questionnaire *(Vorderer et al., 2004)* [14]
(b) Group of questions about the assessment of usability, intuitiveness, and ease of use of individual interfaces.
(c) Survey on the best single choice of interfaces in terms of usability, intuitiveness, and ease of use.

With the spatial presence questionnaire after each cycle, participants filled in a questionnaire for assessing comfort, intuitiveness, and ease of use of particular control interfaces. Spatial presence or tele-presence is an important factor when it comes to VR. As the term itself means the state of consciousness that gives the impression of being physically present in the mediated world [19].

A five point Likert scale was used for all the questionnaire items such that score of 1 means 'strongly disagree' and 5 means 'strongly agree'. During the operation with the inspection robot, it was found that performance of users proved to be a lot better using VR interface when compared with other two interfaces. The mean duration time recorded in case of VR proved to be better when compared to that in case of MONO and STEREO for each set of participants and for each category of tasks which can be distinctly visible in figure 7 as reproduced from the paper [10]. This means that VR should be efficient and satisfactory from the user's perspective. Also, spatial presence of a user is affected enormously by the interface that is being used to operate the robot. This is understandable since the general awareness of movement, interaction and participation is representational of spatial presence, depends and is controlled by the interface. By summarising the survey, it clearly indicated that for all the interfaces studied for robotic arm and camera set, VR interface provides the most intuitive and comfortable one.

7 SUMI

As mentioned in the above sections, there have been a number of applications, packages and development environments released concerning robot programming. It becomes essential to evaluate them based on usability from the perspective of end users. *Software Usability Measurement Inventory (SUMI)* comes as a powerful usability evaluation approach to do so [7]. It provides a valid and reliable mechanism to make comparison of products which are competitive to each other and also to compare various versions of the same product. This is a very helpful study to establish affirming among the developers and the end-users in differing perspectives. One example is that this method provides information based on diagnostics of the products which can be a guiding light for future developments.

This method deploys questionnaire items, which are specially designed to assess the product concerned, and should be ideally answered by a sample of users who have some experience using the product. The concept of usability as assessed by *SUMI*, draws on the definition in IS0 9241, and relates to the European Directive on Health and Safety Standards for Workers with *Visual Display Unit (VDU)* Equipment. To use *SUMI* effectively, it is recommended that a sample of ten users should be present for the experiment. If *SUMI* is to be used for diagnostic purposes then even a lesser sample of users will be fit for the purpose. To evaluate products using *SUMI*, a working version of the product is required. This means that the usability aspects cannot be tested for the products right from the early stages of the product. Hence, usability is only evaluated from the end-users perspectives.

SUMI consists of a standardised database which contains usability profiles of different kinds of commonly used applications. *SUMI* is claimed to be a method for usability evaluation which can be put to use for any kind of application which has a method of collecting input from users and displaying some output. The questionnaire to be made should be standardised and this requires a lot of data collection activity. If a product or series of products is to be evaluated using *SUMI*, either a product-against-product comparison is to be performed as an experiment, or comparison of each product against the standardised database. As a result of the experiment, it can be checked how the product is performing when compared against an average state-of-the-art market profile. A sample usability evaluation by using *SUMI* methodology has been presented in figure 8 as reproduced from the paper [7].

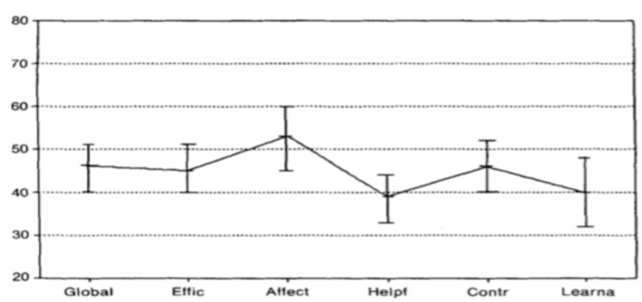

Fig. 8. A sample profile showing SUMI scales [7].

For the purpose of proper analysis, *SUMI* evaluation provides output in three hierarchical layers. The first layer provides a summary which is not very useful for metric based comparison. The second layer represents an evaluation based on five subscales which can be listed as affect, efficiency, learnability, helpfulness, and control. These subscale measurements help in understanding and comparing the user perspective of different systems. This approach can be extended to determine safety of the system for users and it should be measurable from the subscale parameters. The third layer of output is referred to as Item Consensual Analysis. It compares the response pattern on the questionnaires which are received from the users and the responses predicted from a standardised database. This is very helpful in identifying which aspects of the system can be improved upon and what are the strengths of the system.

8 Evaluating the Usability Evaluation Approaches

As discussed above, there are and can be multiple usability evaluation approaches depending on the platform or the type of system under concern. Most of the evaluation approaches should somehow surround a User Centered Development (*UCD*) and or a Participatory Design (*PD*). But, it becomes important to compare and study the usability evaluation methods themselves so that researchers and developers can pick up a potentially competitive and effective method for usability evaluation. The usability evaluation methods available to assess and improve systems often lack to fully understand the capabilities and limitations of each. However, usability evaluation methods cannot be evaluated and compared reliably because of the lack of standard criteria for comparison.

Most of the user interface problems are tackled or solved using iterative process [5]. This involves primarily design, evaluation and re-design. The stages of initial, prototype and final design form the core of the iterative process. Summative evaluation is then used to determine whether final design meets the required criteria to be declared as usable or fit to use. Usability evaluations help in a constructive prototype designing and by means of iteration, it tries to eliminate usability problems. An effective and collective effort from users and experts, to inspect the system under development can facilitate a formative evaluation approach. A notable work in this regard was presented by *Kies et al.* [12] and is picturised in figure 9 reproduced from [5].

It has been well established among both the developers and end-users, that usability needs to be improved and forms an essential quality of systems. The below points as reproduced and adapted from [5] are critical for deciding on usability evaluation approaches.

(a) Usability is core of interaction design.
(b) An iterative, evaluation centered process is essential for developing high usability in interaction designs.
(c) A class of usability techniques called usability evaluation methods have emerged to support that development process by evaluating the usability of interaction designs and identifying usability problems to be corrected.

Practically, it is quite difficult for researchers to reliably compare such evaluation methods due to lack of the following information as reproduced from [5].

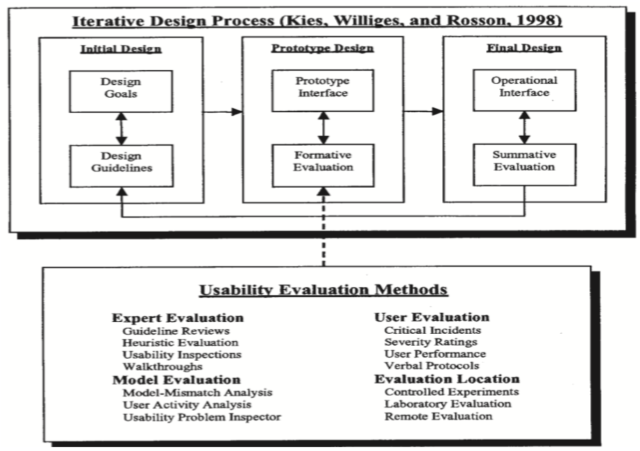

Fig. 9. Usability evaluation approaches used in formative usability evaluation [5].

(a) Standard criteria for comparison.
(b) Standard definitions, measures, and metrics on which to base the criteria.
(c) Stable, standard processes for usability evaluation approaches' evaluation and comparison.

Also, usability of usability evaluation approach plays an important role in selecting the approach so that it can be effectively and efficiently used for identifying usability problems in a robot system or application. As noted above, since it is difficult to identify ultimate criteria and to measure them directly, researchers actually select some of the many possible actual criteria to bring an approximation to ultimate criteria. This criteria is then fed into comparing the potential of an evaluation approach. In this study, a target design of the system is considered and the usability evaluation approaches to be compared are applied. This results in a usability problem list as a consequence which is then compared to the standard list of usability problems. The competitive usability evaluation approaches are now compared based on performance metrics which are computed from problem lists. The figure 10 reproduced from [5] displays the intention of this study and process.

8.1 Issues with Usability Evaluation

Agreeing on the fact that usability plays a significant role in programming and crafting out a project, or an independent system, usability evaluation approaches can have some negative effects as well if studied carefully. Following the line of usability evaluation in robot programming, if it's aspects are implemented by purely following hard and first rules, it can prove to be ineffective [1]. If a room for innovation is considered, then while programming the robot interface, the robot programmer might try to implement a feature in the interface which might not

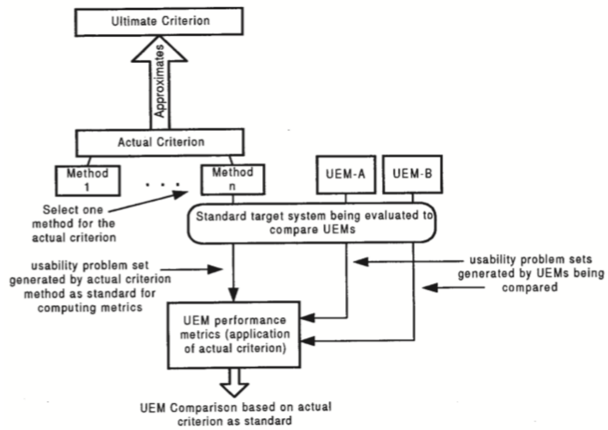

Fig. 10. Roadmap of concepts [5].

conform to the strict rules of usability. In this scenario, programmers have to stick by the book of rule rather than to try new features. Such a feature could have been as well a future inspiring vision. This calls for a meaningful critique of the design under discussion. Thus, the evaluation approach must be decided upon carefully and should be pointing to the core topic in concern.

Such perspectives can be discussed along the line of innovation, risk taking factor, being stagnant, etc. If a robot programmer 'prefers' a particular platform or language for some reason, this could mean that risks being taken are less, and hence this leads to lesser degree of innovation being made resulting in a system being developed to be stagnant. Usability evaluation approach taken and the result obtained as a consequence cannot guarantee a user centered design.

Usability evaluation approaches are pushed heavily into producing any system. In human computer interaction education, usability evaluation approaches hold a central component in teaching students. Interface specialists in industries, prefer usability evaluation as a very important and major aspect of work practice. In short, the fields of education, academy and industry hold usability evaluation as a critical and necessary component. While usability evaluation provides multiple positive outcomes, the real problem lies on the fact that it is applied often blindly to scenarios. When a design meets a wrong usability evaluation approach, it often results in a negative outcome to the extent that might lead to defending the purpose of the system being developed. This in turn provides a different direction and guideline for future designs and development which might not be very fruitful.

An important step is to produce replicating experimentation with the chosen usability evaluation approach on the robot system under concern. This has many advantages which are verification and ensuring consistency of results produced, among others. Replication helps in understanding and discovering the boundaries of the evaluation approach as well as its strength.

9 Conclusion

After studying the concepts on usability, and understanding some usability evaluation approaches taken to examine robot development environments, robot programming approaches and robot systems as a whole, it is realised that one should not adopt a standardised approach on usability evaluation. However, similar approaches can be used for groups of robot systems with similar use-contexts, nature of tasks and interfaces. Hence, choosing a valid approach can be a strenuous task in itself and the usability faults identified thereafter by such an approach should be prioritised and worked upon. Therefore, this must ensure that the robot system can be programmed by end users with much ease and comfort of use.

10 Future Work

Usability evaluation approaches require continuous evaluations, while being aware of the latest technologies in use. It must strive to bring in new perspectives like safety to carefully analyse development interfaces and environments concerned. The usability evaluation approach on VR interface for inspection robot, on *PbD* programming method and *SUMI* profiling should be stretched to explicitly determine safety as a measurable aspect. The usability evaluation approach designed must not harm the essence of innovation. Also, combinations of various evaluation approaches can also be an interesting area of research, for its potential as a valid approach to bring out usability in robot programming.

References

1. Greenberg S., Buxton B.. Usability Evaluation Considered Harmful (Some of the Time). In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 111–120. ACM, NY,USA. (2008)
2. Orendt E.M., Fichtner M., Henrich D. Robot Programming by Non-Experts: Intuitiveness and Robustness of One-Shot Robot Programming. In: 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) , pp. 192–199. (2016)
3. MacKenzie D.C., Arkin R.C. Evaluating the Usability of Robot Programming Toolsets. The International Journal of Robotics Research,(17), 381–401 (1998)
4. Bruun A., Stage J. Training Software Development Practitioners in Usability Testing: An Assessment Acceptance and Prioritization. In: Proceedings of the 24th Australian Computer-Human Interaction Conference, pp. 52–60. (2012)
5. Hartson H.R., Andre T.S., Williges R.C. Criteria For Evaluating Usability Evaluation Methods. International Journal of Human-Computer Interaction,(13), 373–410 (2001)

6. Frøkjær E., Hornbæk K., Hertzum M. Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated?. In: CHI '00: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 373–380. ACM, NY, USA (2000)
7. Kirukowski L., Corbett M. SUMI: the Software Usability Measurement Inventory. British Journal of Educational Technology,(24), 210–212 (1993)
8. Bruun A. New approaches to usability evaluation in software development: Barefoot and crowdsourcing. Journal of Systems and Software,(105), 40–53 (2015)
9. Scholtz J., Antonishek B., Young J. Evaluation of a Human-Robot Interface: Development of a Situational Awareness Methodology. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences . Big Island, HI, USA (2004)
10. Jankowski J., Grabowski A. Usability Evaluation of VR Interface for Mobile Robot Teleoperation. International Journal of Human-Computer Interaction, 882–889 (2015)
11. Kramer J., Scheutz M. Development environments for autonomous mobile robots: A survey. Auton Robot 22, 101—132 (2007)
12. Kies, J. K., Williges, R. C., Rosson, M. B. Coordinating computer-supported cooperative work: A review of research issues and strategies. Journal of the American Society for Information Science, 49, pp. 776—779. (1998).
13. Bach, C., Scapin, D. Comparing inspections and user testing for the evaluation of virtual environments. International Journal of Human- Computer Interaction, 26. pp. 786—824. (2010).
14. Vorderer P., Wirth W., Gouveia F. R., Biocca F., Saari T., Jänck, F., Jäncke P MEC Spatial Presence Questionnaire(MEC-SPQ): Short documentation and instructions for application (Report to the European Community). (2004).
15. Walker M.A., Fromer J., Di Fabbrizio G., Mestel C., Hindle D. What can I say?: Evaluating A Spoken Language Interface to Email. In: Proceedings of CHI 98 . pp. 582–589. (1998)
16. ROS using CLion : https://www.jetbrains.com/help/clion/ros-setup-tutorial.html
17. Jakob Nielsen J., Levy J. 1994. Measuring usability: preference vs. performance. Commun. ACM 37, 4 pp. 66-75. (1994)
18. Chen, J. Y., Oden, R. N., Kenny, C., Merritt, J. O. Stereoscopic displays for robot teleoperation and simulated driving. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 54, pp. 1488–1492. (2010).
19. Tran N. X., Phan H., Dinh V. V., Ellen J., Ber, B., Lum J., . . . Duffy L. Wireless data glove for gesture-based robotic control. In J. A. Jacko (Ed.), Human–computer interaction, Novel interaction methods and techniques. Lecture notes in computer science (Vol. 5611, pp. 271–280). Berlin, Germany: Springer Verlag. (2009).
20. Law, E. "Evaluating the downstream utility of user tests and examining the developer effect: A case study", International Journal of Human-Computer Interaction, (21:2), pp. 147-172, Taylor Francis, London. (2006).
21. Sawyer, P., Flanders, A. and Wixon, D. "Making A Difference – The Impact of Inspections", in Proceedings of CHI, ACM Press, New York. (1996).
22. Bruun, A., Stage J. Training software development practitioners in usability testing: An assessment acceptance and prioritization. Proceedings of the 24th Australian Computer-Human Interaction Conference, OzCHI 2012. pp. 52-60. (2012).
23. https://www.keithmichaels.co.uk/news/driverless-cars-pros-and-cons/