Fostering network troubleshooting skills among CS undergraduates

Submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

by

Kavya Alse
(Roll Number: 144380001)

Supervisor(s):

Prof. Sridhar Iyer
Prof. Sasikumar M

Interdisciplinary Program in Educational Technology
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

2022

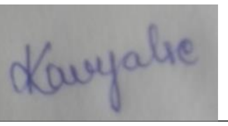
Dedication Sheet

Dedicated to

The goodness that makes this living worthwhile

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



(Signature)

Kavya Alse

(Name of the student)

144380001

(Roll No.)

Date: 20-05-2022

Approval Sheet

This thesis entitled "Fostering network troubleshooting skills among CS undergraduates" by Kavya A K Alse is approved for the degree of Doctor of Philosophy.

Examiners

Supervisor (s)

Chairperson

Date: 20-04-2022

Place:

Abstract

Computer Science engineers often have to troubleshoot system they are working with. However, teaching-learning of troubleshooting skill has not had much attention in the curriculum. It is typically limited to debugging in the context of programming projects or the introduction of simulators and packet analysers in the context of network troubleshooting. However, troubleshooting a network requires more than the knowledge of tools. It is a structured process of searching for the fault in the problem space and evaluating a set of possible causes carefully. It requires domain knowledge, skills to conduct a test, and ability to follow the troubleshooting procedure. It is an important skill for CS undergraduates and a complex one at that, so it needs to be taught explicitly to the students. In this research, we started with a broad goal of teaching network troubleshooting skills to CS undergraduates. We developed a TELE called PHyTeR (pronounced as fighter) to teach network troubleshooting skills. In this thesis, we describe the evolution of PHyTeR and its evaluation.

We adopted design-based research methodology (DBR) to create PHyTeR. DBR is suitable for creation of practical educational interventions for complex problems. DBR is iterative and addresses a dual goal of developing the intervention as well as discovering new knowledge that can provide insights to others working with similar problems. Each iteration consists of the phases: problem analysis & exploration, design & development, and evaluation & reflection. We implemented two cycles of DBR in this thesis. In the first iteration, we systematically surveyed the literature related to troubleshooting skills to characterize it from different domains and perspectives. We synthesized a process for network troubleshooting skill with four phases: Problem Space Understanding, Hypothesis Generation, Hypothesis Testing, and Result Interpretation. The learning goals and pedagogy in PHyTeR are based on these four phases.

We then conducted study1 (N = 5) to understand the student difficulties during network troubleshooting skill. It was an exploratory study where two researchers (including me) worked with each student. We generated themes related to the difficulties faced by students during troubleshooting along each phase of troubleshooting. Based on the phases of troubleshooting we designed a TELE, PHyTeR. We accounted for the identified difficulties by providing scaffolds that were known to support students in similar contexts. PHyTeR is based on the pedagogy of inquiry learning. PHyTeR pedagogy includes learning activities corresponding to each of these phases along with scaffolds to overcome the difficulties. It has reflection activities to guide students synthesize their actions in the framework of troubleshooting.

We evaluated the first version of PHyTeR (study2, N = 21) to identify if PHyTeR supports students in achieving the learning goals. At the time of the study, the students had completed a theory and a practical course related to computer networks. We analysed the perceptions of students about troubleshooting and PHyTeR. The students recognised some useful notions about troubleshooting like reproducing the error and understanding the problem space. However, some naive conceptions still existed. After analysing the results of this study, we identified the learning goals yet to be achieved by students after interacting with PHyTeR and the difficulties they face while interacting with PHyTeR. This led us to the second iteration of the DBR cycle.

In the second iteration, we modified PHyTeR from the perspective of learning, design and usability. These modifications were based on review of literature and consultation with a UI/UX designer. We evaluated this version of PHyTeR in study3 (N= 5) to identify how students use the features in PHyTeR to during troubleshooting. We followed this with study4 (N = 20) where we triangulated the results of study3 in a larger population. We found that, after interacting with PHyTeR to solve a troubleshooting scenario, the students' conceptions of troubleshooting included the notions of structured process having iterations of evidence based testing. They recognised the importance of comparing the behaviours of the given faulty network with that of an ideal version of the network. They identified the usefulness of hypotheses in the process of troubleshooting.

The main contributions of this thesis are the identified student difficulties, PHyTeR pedagogy. We have implemented this pedagogy in the namesake learning environment PHyTeR that can be used by CS undergraduates having pre-requisite knowledge.

Keywords: Network Troubleshooting Skills, TELE, Technology Enhanced Learning Environments, Design Based Research, PHyTeR

Table of Contents

List of Figures

List of Tables

Abbreviations and Nomenclature

TEL	Technology Enhanced Learning
TELE	Technology Enhanced Learning Environment
DBR	Design-Based Research
RQ	Research Question
DQ	Design Question
LQ	Literature Question
SUS	System Usability Survey
TELoTS	Technology-Enhanced Learning of Thinking Skills
LO	Learning Objective
IDD	Instructional Design Document
UI/UX	User Interface/ User Experience
PSU	Problem Space Understanding
HG	Hypothesis Generation
HT	Hypothesis Testing
RI	Result Interpretation
PHyTeR	Name of the TELE developed in this thesis. An acronym for **P**roblem Space Understanding, **Hy**pothesis Generation, Hypothesis **Te**sting, **R**esult Interpretation.

Chapter 1

Introduction

1.1 Background and Motivation

Consider the following scenario: A woman is trying to access a website from her laptop. However, she is getting a DNS server error. Figure 1 shows the browser displaying the error.

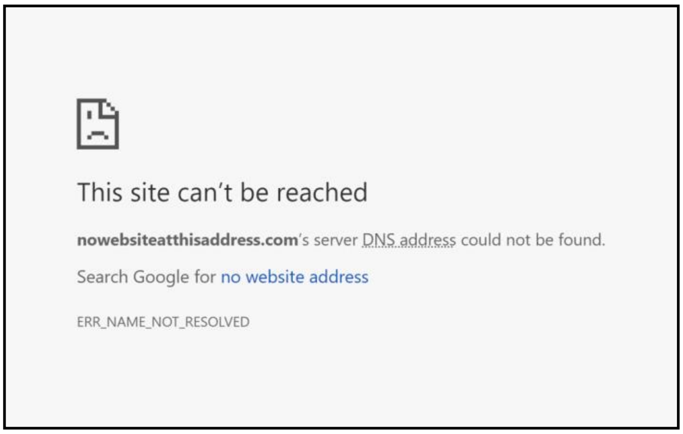


Figure 1: A screenshot of a browser showing a Domain Name Server error

Let us examine how such a problem can be resolved and what actions are required to solve such troubleshooting problems: In order to resolve this problem, one has to recreate the fault. That is, she has to identify the conditions under which the fault is observed. In addition, she has to know the correct output (the one without an error). Based on the correct output and the observed fault, she can come up with one or more plausible reasons for the error. Then she has to systematically investigate each of these reasons to identify the actual cause of the error. Once she has figured out the actual cause, she then has to correct it and then verify that correcting that cause has actually resolved the error and did not lead to unwanted repercussions.

A computer science undergraduate is expected to find out the reason for this type of error and to resolve them. This scenario is one example of troubleshooting problems which CS graduates are expected to solve. When we examine the existing CS curricula, we see that the emphasis given to such troubleshooting problems is passive at the best. This thesis started with a broad goal of solving some part of this problem - to develop troubleshooting skills among CS undergraduates.

1.2 Setting Context

Today's CS graduates work with a plethora of technologies. In any workplace, in addition to comprehending the code, they will have to assimilate the role each technology/framework plays in the software that they are building. Along with this, there will be a number of software development aids like debugger, logger, version controller, test automation etc. The skills needed to work in such environment are not generally taught in their classrooms. An essential skill needed to work in this environment is troubleshooting. It has been reported that troubleshooting expends a significant amount of time and effort of software developers (Cohane, 2017).

The curricula for undergraduate courses are mostly created from a design standpoint. That is, after completing the curricula, a student is expected to be able to design software. However, there are findings that indicate that learning concepts from a design perspective might not serve the purpose of troubleshooting (Jonassen, 2010). In order to troubleshoot, they need to understand the systems from a different perspective.

CS education researchers have tried to tackle this problem from different directions, and different perspectives. They have tried to explore how students comprehend a program, how students debug simple programs, how students get to a systemic understanding etc. In the next sub-section, we discuss the existing approaches related to troubleshooting in Computer Science.

1.2.1 Troubleshooting in Computer Science and Computer Networks

In the domain of computer science, students start with troubleshooting simple programs. This is generally called debugging. The students start with syntactic errors and move onto logical and semantic errors. In case of a program, one is involved in analysing the behaviour of a program with respect to its constituent functions, variables and other programming constructs. As students start writing complex applications, the troubleshooting involves multiple parts like APIs accessed by the application, libraries used in the application, multiple parts of the application like data storage, parts that manipulate this data and parts responsible for interactions with the user etc. One needs a comprehensive knowledge of how these different parts of an application interact with each other to produce the desired output. There are a number of tools available to help the programmers in these activities. A debugger helps one to identify errors in any part of the program. It enables one to observe how a particular function/variable is behaving for a given input. One can collect log files from multiple parts and analyse them.

In case of computer networks, students are introduced to tools and simulators like ns2, packet tracer, Wireshark etc. The lab exercises in a typical curriculum involve students creating different types of network and observing various parameters of a network in a simulator. There are generally no experiments focusing specifically on network troubleshooting skills. In addition, the simulators do not have features similar to watch variables and breakpoints that are present in a debugger. A network troubleshooting process includes multiple iterations of testing sub-networks and devices to check if they are working as expected. Keeping track of this process involves considerable mental effort. We illustrate this with an example as follows:

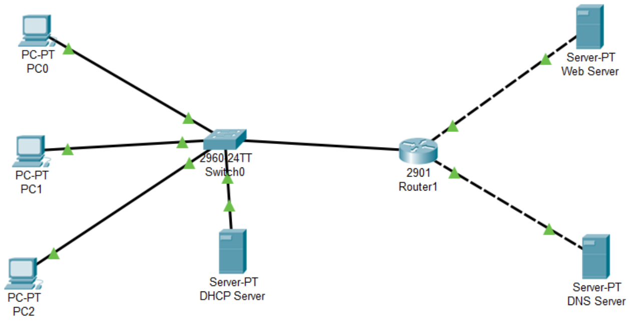


Figure 2: An example network

Consider a network as shown above. Assume an error where packets are not transferring between PC0 and web server. A troubleshooter would try sending packets within the sub-network of PC0. Then the connection between the switch and the router would be checked. Then the connection between the router and the web server would be checked. At each connection there might be multiple tests done with respect to individual devices. Conventional simulators do not have any scaffolds to keep track of this process. This makes doing and learning network troubleshooting harder. To summarize, we argue that network troubleshooting skill is required and that it is complex. The existing tools and curriculums are not capturing the complexity of the process of troubleshooting. In the next section, we discuss the troubleshooting instances in other domains.

1.2.2 Troubleshooting in other domains

Troubleshooting or debugging has been discussed in different contexts, domains. Troubleshooting problems are one of the six general problems considered in the problem solving literature (Jonassen, 2010; Pretz, Naples, & Sternberg, 2003). The pan-domain-ness of troubleshooting is evident from the fact that it is employed in a wide variety of contexts like the repair of mechanical or electronic systems to psychotherapists and public relation specialists (Ziegenfuss Jr, 1988). In domains like mechanical and electrical systems, troubleshooting has been focussed in the vocational training courses (Blackburn, 2013;

Lesgold, 1988; Ross & Orr, 2009). These courses are very specific to particular systems like aircraft maintenance, agricultural motor repair, marine frigate troubleshooting, chemical engineering etc. The students of these courses, most of the time, are also the prospective users of such systems.

A complex instance of troubleshooting can be seen in the medical domain where it is commonly referred to as diagnostic reasoning. Diagnosis is the root of what a doctor is supposed to do: when a patient comes in with a set of symptoms, the doctor is supposed to recognize her condition based on the symptoms, evidences related to those symptoms, anatomical understanding of a human body and also variations in case of the specific patient etc. Since there is a lot of variance in each of these aspects, medical diagnosis becomes the most difficult of all. Jonassen (Jonassen, 2010) places these diagnosis-solution problems at a higher level of complexity than machine troubleshooting problems. The argument is, it is hard to define cause and effect relationship in case of medical diagnosis and in some cases like a chronic ailment, it is not possible to return the system back to functioning but the troubleshooter has to search for workaround solutions.

Another set of complicated troubleshooting can be observed where the process involves human beings. Examples of these include teachers troubleshooting their learning designs, a manager troubleshooting the productivity of his team etc. In these cases, the complexity arises from the fact that these systems are made of components that cannot be cleanly modularized into mutually exclusive, causal components. Also unlike man-made systems, where the output for an input is fixed, it is not always the case where human beings are involved. Now we describe some concrete examples of troubleshooting in various domains:

- Medical diagnosis (Sharma, Sandeep, Hashmi, & Rawat, 2021): "A 33-year-old white female presents after admission to the general medical/surgical hospital ward with a chief complaint of shortness of breath on exertion. She reports that she was seen for similar symptoms previously at her primary care physician's office six months ago. At that time, she was diagnosed with acute bronchitis and treated with bronchodilators, empiric antibiotics, and a short course oral steroid taper. This management did not improve her symptoms, and she has gradually worsened over six months. She reports a 20-pound (9 kg) intentional weight loss over the past year. She denies camping, spelunking, or hunting activities. She denies any sick contacts. A brief review of systems is negative for fever, night sweats, palpitations, chest pain, nausea, vomiting, diarrhoea, constipation, abdominal pain, neural sensation changes, muscular changes, and increased bruising or bleeding. She admits a cough, shortness of breath, and shortness of breath on exertion." What is the diagnosis?

- Mechanical Troubleshooting (Donald R. Woods, 2006): The inlet and outlet valves of the startup heater were opened during the startup of reactors used to synthesize ammonia. The synthesis loop pressure was equalized. The start-up heater firing was increased. The high-pressure stage valves of the gas compressor were opened. Nonetheless, it was difficult to get the fuel-gas pressure greater than 80kPa. A rumbling noise was heard when the pressure was further increased. The process gas temperature was only 65º C. How to troubleshoot the scenario?

Although the scenarios described above belong to different contexts, we can abstract out the common features of these problems. In each of these, including the networking example discussed in section 1.2.1, there is some undesired behaviour one can observe in the system (the initial state) and a desired state that has to be attained by the system (the final state). There might be one or more ways of reaching the final state from the initial state. These behaviours are produced by the interaction between the components of the system in a particular way. A computer science graduate should be able to find ways to reach the final state from the initial state for simple network problems like the one described in 1.1. With this background, we describe the research objective of this thesis in the next section.

1.3 Research Objective

We want to develop network troubleshooting skills among CS undergraduates. In previous sections, we described that there is a lack of tools and pedagogies to develop network troubleshooting skills for this population. In addition, there is a lack of research on how CS undergraduates do network troubleshooting and about the difficulties they face during troubleshooting. We want to start out investigation by understanding the student difficulties and then build a learning environment to enhance their network troubleshooting skills. We then want to investigate the effect of the learning environment on the target population. We aim to understand how students learn troubleshooting and use this technology enhanced learning environment (TELE) and its features while learning troubleshooting.

1.4 Research Methodology

1.4.1 Design Based Research

Design Based Research (DBR) is a research methodology (Brown, 1992; Collins, Joseph, Bielaczyc, Collins, & Bielaczyc, 2004; McKenney & Reeves, 2013)appropriate for developing pragmatic educational interventions for complex problems. It includes multiple iterations of design, development, evaluation and reflection to reach the overarching goal. This methodology came into prominence while researchers were trying to develop interventions that could work in complex settings. In addition, an important goal of these research projects was also to find answers to questions like 'why did/did not an intervention work in a given setting?' This implies that the researchers trying to adopt this methodology should focus on the theoretical grounding of an intervention from the beginning so that it acts as an anchor while analysing and answering these 'why' type of questions. A DBR problem usually involves multiple stakeholders, for example, students, teachers, administrators, policy makers etc. DBR advocates a participatory design to include multiple perspectives from these stakeholders. In the next sub-section, we discuss how we used DBR in this research.

1.4.2 DBR in this thesis

The overarching goal of this thesis is to create a technology enhanced learning environment (TELE) to develop troubleshooting skills among CS undergraduates. In this regard, we started the first iteration by trying to understand what troubleshooting means, how troubleshooting is performed and taught in different contexts. A second direction we explored is to understand the difficulties of students while they troubleshoot or learn to troubleshoot. In addition, we conducted a study (study1) to identify the difficulties faced by students in our context. Building upon these two sources, we developed the first version of the learning environment called PHyTeR (pronounced as fighter). We conducted study2 to evaluate this learning environment. This was a single group post only study. We identified the students' perceptions related to troubleshooting skills and PHyTeR. In order to do this, we analysed the student responses during and after interacting with PHyTeR. At the end of this, we observed that the students were able to do some of the tasks in PHyTeR. Their understanding of troubleshooting skills included some desirable conceptions about troubleshooting skills like breaking down the problem, narrowing down problem space etc. We recognised that they were not able to completely follow the tasks in PHyTeR and recognise the overall process of troubleshooting. The first row in Figure 3 summarizes the activities performed in the first cycle of DBR.

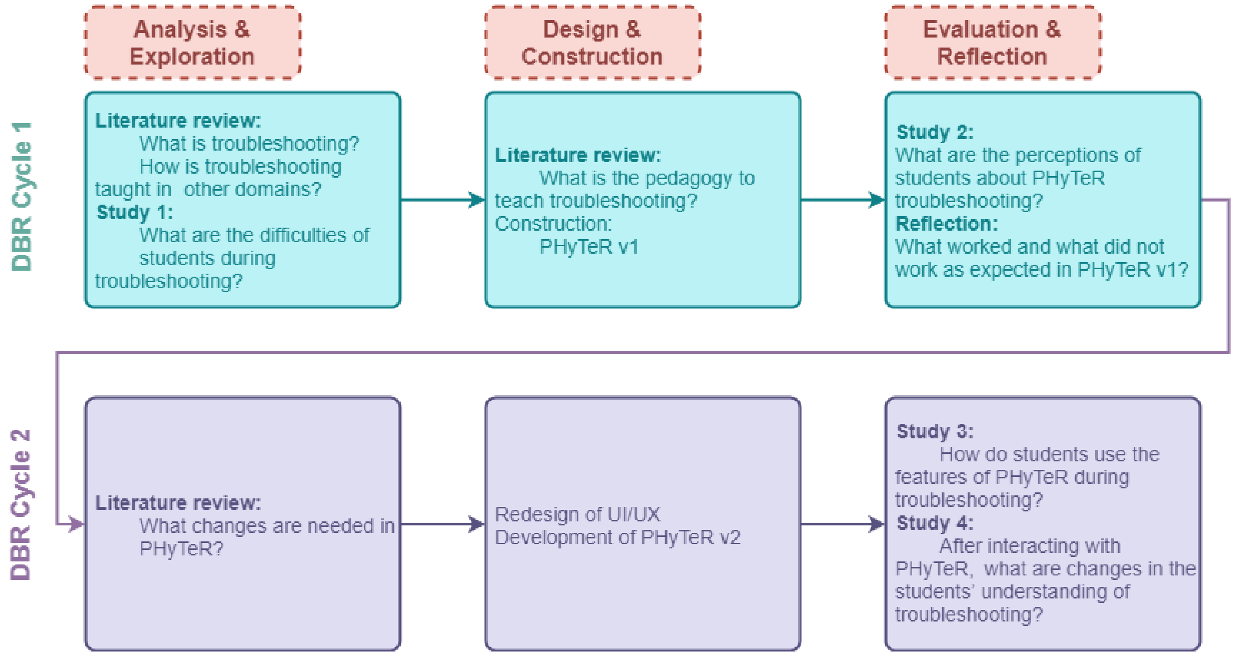


Figure 3: Overview of DBR cycles in this thesis

This led us to the second cycle of DBR. We started this cycle with a goal to improve PHyTeR based on the reflection from first cycle. In this cycle, we investigated the literature to identify the types of support that are required to overcome the difficulties, which were not resolved in PHyTeR v1. We incorporated process prompts and reflection activity to help students overcome these difficulties and developed PHyTeR v2. We evaluated this PHyTeR v2 in two studies (study3 & study4). Study3 is a qualitative study where the researcher closely observed five students while they were working with PHyTeR. Study4 followed mixedmethods to analyse the data. We use the data from study4 to triangulate the results from study3. After working with PHyTeR v2, we observed that the students identified that troubleshooting is a logical, iterative and evidence- based process. They recognised the usefulness of reproducing the fault before trying to solve it and the need to analyse the problem space. We describe a summary of this cycle in the second row of Figure 3.

1.5 Solution Overview

Our target population is Computer Science undergraduates studying in the third and fourth year of B.Tech (Bachelors of Engineering) programme. These students study a course (theory + lab) on computer networking in their second or third year, thus having the required domain knowledge on computer networking.

We have created a technology enhanced learning environment (TELE) called PHyTeR to develop troubleshooting skills among computer science undergraduates. PHyTeR (pronounced as 'fighter') is an acronym for **P**roblem Space Understanding, **Hy**pothesis Generation, Hypothesis **Te**sting and **R**esult Interpretation, the four phases of troubleshooting that we have employed. The basis of PHyTeR is making the process of troubleshooting explicit to students and it uses the hypothesis testing cycle adapted from inquiry learning (Lazonder, 2014). It contains multiple learning activities with scaffolding and feedback that provides multiple opportunities to practice a structured approach of troubleshooting. These tasks are grouped according to the phases of troubleshooting explained in section 2.4. In the next sub-section, we explain the learning environment in detail.

1.5.1 PHyTeR - The learning environment

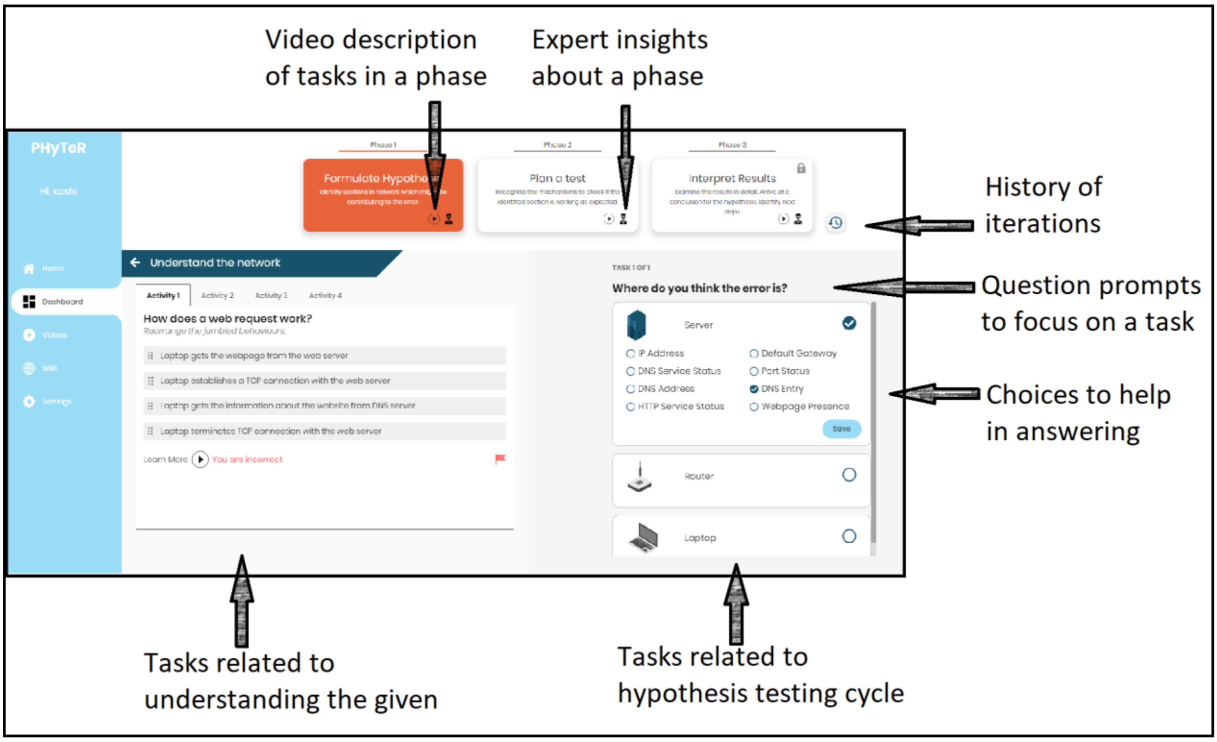


Figure 4: Annotated screenshot of PHyTeR

PHyTeR is a learning environment with two components - a simulator and a web based component. Figure 4 gives a summary of different features in PHyTeR. In the current version, we have used a simulator called Cisco Packet Tracer (version 7.2.1) in all our experiments with students. It is also possible to use other simulators with little change to the web component. The faulty network that the students have to troubleshoot exists in the simulator. The students start troubleshooting in the web component by familiarizing themselves to the concept of troubleshooting and the features of PHyTeR. Then, they start troubleshooting a scenario. Here, PHyTeR orchestrates the interactions between the simulator and the web component. Currently, PHyTeR has one ‘scenario’ that the students have to troubleshoot. There is provision to add more problems.

Now we describe various features of PHyTeR learning environment: The left most part of the screen has the navigation menu. Students can use this to manoeuvre between different pages in the web component like profile, dashboard, videos and wiki. Besides the navigation menu is the task area.

The student starts to troubleshoot a scenario by either reading the description of scenario and/or watching a video description. They are nudged to start with the problem space understanding activities. However, they can even start with hypothesis generation task. The top panel in the task area represents the buttons corresponding to the phases in the hypothesis testing cycle. One can click on the buttons to go to that phase (once unlocked). The left panel on the task area is where a student does activities to understand the problem space. After completing each of these activities, the student gets a prompt to generate hypotheses based on the understanding gained by doing that activity. For example, after completing 'how does a router work?' activity, there is a prompt that goes like 'Now, that you have understood how a router works, do you think the router in the faulty scenario is working accordingly? If you think there is some error in the router, do you want to formulate a hypothesis to check this?'

The right panel in the task area has tasks specific to a phase in the hypothesis testing cycle. The tasks are structured as question prompts. This is intended to help students focus on one task at a time. The students iterate over the tasks in hypothesis testing cycle until they find the solution. They can do a problem space understanding activity whenever they want. After a student has successfully troubleshot the scenario, she is navigated to a reflection activity that summarizes the process of troubleshooting. It contains activities to help students identify the tasks in PHyTeR and relate those tasks to the phases of troubleshooting.

1.5.2 Experiments and findings

In this section, we report a summary of the research questions (RQ) or design questions (DQ) investigated in this thesis.

- **To understand the difficulties faced by students during troubleshooting**
    - RQ 1 (Investigated in Study 1): What difficulties do students face during troubleshooting?

To answer this RQ, we conducted a study with five students who were studying in the 3rd year of computer science engineering programme. We asked them to solve a network-troubleshooting problem. Two researchers observed and guided them during the problem-solving session. The transcription of problem solving conversations were analysed to identify the difficulties. This was conducted as a part of the problem analysis phase in the first cycle of DBR.

- **To design a technology enhanced learning environment (PHyTeR) to teach troubleshooting skills**
    - DQ1: Which pedagogy is appropriate for PHyTeR?
    - DQ2: What features are required in PHyTeR?

We answer design question1 (DQ1) by reviewing relevant literature. We identified science inquiry as the broad pedagogy. Then we analysed additional literature to identify the features and design learning activities. Our focus was to develop a learning environment that would help in alleviating students' difficulties and guide them towards troubleshooting. We did this as a part of problem analysis and design phases in the first cycle of DBR.

- **To evaluate the effect of PHyTeR on students in terms of their understanding of troubleshooting skills and the process of doing troubleshooting**
    - RQ 2: How does students' understanding of troubleshooting skills change after using PHyTeR?
        - RQ 2.1 (Investigated in study2): After working with PHyTeR v1, what are the perceptions of the students about troubleshooting skills?
        - RQ 2.2 (Investigated in study3): After interacting with PHyTeR v2, what is the students' understanding of troubleshooting skills?
        - RQ 2.3 (Investigated in study4): After interacting with PHyTeR, what are changes in the students' understanding of troubleshooting skills?
        - RQ 2.4 (Investigated in study4): After interacting with PHyTeR, what are the changes in student perception of students related to troubleshooting actions, knowledge and confidence?
    - RQ 3: What is the role of PHyTeR features during troubleshooting?
        - RQ 3.1 (Investigated in study2): What are the perceptions of the student about PHyTeR v1?
        - RQ 3.2 (Investigated in study3): How do students use the features of PHyTeR v2 to troubleshoot?

We present the answer to these questions based on three studies. Study2 – where twenty-one students used PHyTeR v1, Study3 - where five students worked with PHyTeR v2 and Study4 where twenty students worked with PHyTeR v2. Study2 was conducted as a part of the evaluation phase in the first cycle of DBR. Study3 & 4 were conducted in the evaluation phase in the second cycle of DBR. In the context of these studies, we analyse pre, post form responses and interview transcripts to identify the students' perception of troubleshooting skills and how students use the features in PHyTeR during troubleshooting.

After interacting with PHyTeR, students perceived that troubleshooting is a systematic process. They recognized the usefulness of comparing between ideal and faulty network behaviour. They identified the role of hypothesis in troubleshooting. We discuss these and other related results in detail in section 8.3.

1.6 Scope of this thesis

This section describes the scope of the thesis along the following dimensions:

- Learner Characteristics: The goal of the thesis is to develop network troubleshooting skills for the specific population of undergraduates in the domain of Computer Science. All the students who participated in the study were in their 3rd year or 4th year of bachelors in CS engineering in India. These students had completed one course in computer networking. All of them had used one or more network simulators. They varied in their academic performance and were fluent in English. They grew up in urban, semi-urban cities of India in varied socio-economic backgrounds. They were familiar with other technologies used like working with a computer, accessing emails, Google forms, watching videos etc.

- Scope of the domain: PHyTeR consists of a troubleshooting scenario that is based on the concepts of network layers, HTTP, TCP/IP protocols, routing, and web servers. All these concepts were part of the curricula in the networking course that the students had completed. PHyTeR has a local wiki section that contains introductory information about these concepts.

- Scope of implementation methodology: PHyTeR consists of two parts: The simulation and the web-based interface. The web-based interface controls the orchestration of the whole learning environment so that it can be used with any network simulator. PHyTeR can be used in a lab session by a teacher or it can be used directly by the student for self-paced learning. Currently PHyTeR is designed to be used by individual students and it does not have activities/ support directed to support collaboration.

- Scope of UI design of the learning environment: An important consideration taken while designing PHyTeR is to reduce the load on working memory of the students along with making it easily accessible for them. It is assumed that the students in our population prefer watching videos - so most of the scaffolds, how-to guides in PHyTeR are created as videos. Though the web-based component can be accessed via a mobile phone, it is difficult to work on a network simulator in a mobile. So we have carried out all the experiments in desktop/laptop computers.

1.7 Contributions of this thesis

This thesis contributes to existing knowledge about teaching of network troubleshooting skills, the difficulties faced by students while troubleshooting. Specifically, it augments the research related to the development of technology enhanced learning environments for developing troubleshooting skills. The contributions are based on the research studies conducted as part of this thesis. We present the specific contributions below:

- We investigated the difficulties of students doing network troubleshooting. We categorise these difficulties along each phase of troubleshooting.

- We developed PHyTeR pedagogy to teach troubleshooting skills. This pedagogy is based on the theory of science inquiry learning. We synthesized the literature related to troubleshooting, problem solving to create this pedagogy and the learning activities in it. We developed scaffolds based on the results of student difficulty study.

- We developed a technology enhanced learning environment called PHyTeR that can be used by students and teachers in computer networking course to learn and teach troubleshooting skills.

1.8 Structure of this Thesis

This thesis is divided into 10 chapters. Chapter 1 elaborated the motivation of the problem and provided an overview of the research methodology, solution design and evaluation along with the contribution of the thesis. Chapter 2 describes in detail, the characterisation of the problem via a review of literature in the various domains of troubleshooting, teaching-learning of troubleshooting skills and related domains like problem-solving and science inquiry. Chapter 3 presents the overall research methodology of the thesis, that is, Design Based Research, along with a summary of two cycles of DBR.

Chapter 4 starts with analysis of what we know about troubleshooting and what is unknown in our context. Then we describe study1 that was conducted to identify the difficulties faced by students during troubleshooting. This constitutes the problem analysis phase of DBR cycle 1. Then in Chapter 5, we present the pedagogy and its implementation in PHyTeR v1 highlighting the critical features that we believe will help in learning troubleshooting skills. Chapter 6 describes the evaluation phase of DBR cycle 1. Here we present the details of study2, which is conducted to evaluate PHyTeR v1. We end with the reflections from DBR cycle 1 and questions arising at the end of this cycle.

Chapter 7 & 8 describe the 2nd cycle of DBR. In chapter 7, we start with the questions evolving from DBR cycle 1. We present the literature review done to answer those questions. The changes made to PHyTeR because of this analysis are explained in detail. Chapter 8 illustrates study3 & study4 that were done in order to evaluate PHyTeR v2. We discuss the results obtained from the experiments in the context of the broad goal of the thesis in chapter 9. Chapter 10 presents the conclusion along with contributions of the thesis and future work.

Chapter 2

Review of Literature

In this chapter, we present a systematic review of the literature related to troubleshooting and nurturing troubleshooting amongst students.

2.1 Process of literature review

We used 'troubleshooting' as the key search word while searching for literature. Related words like 'troubleshooting skills', 'diagnosis', 'diagnostic reasoning' and 'debugging' were also used when they seemed appropriate. We used Google Scholar as the main search engine. We noticed that the word 'troubleshooting' is used in literature related to problem solving and in literature related to repairing electric, mechanical systems. The word 'diagnosis' and 'diagnostic reasoning' has been used in the context of medical diagnosis and in literature related to automated troubleshooting systems. The word 'debugging' has been used mostly in the context of programming. We will use the term 'troubleshooting'/ 'troubleshooting skills' in this thesis and use other terms only when the distinction is needed.

Research areas that have investigated troubleshooting extensively are medical diagnosis, training and maintenance of electrical, chemical, and mechanical systems, human factors in engineering, training to provide technical support, ergonomics, problem solving and automatic diagnostic systems. 'Troubleshooting' has also been used in teaching and training literature in the context of troubleshooting a classroom/organisation, that is, finding out things that are not working as expected in the classroom/organisation. In this chapter, we focus on man-made systems and not on classroom/organisation or medical diagnosis where the diagnosis is of the human body.

The questions that guided the literature review are:

- What is troubleshooting?
- How is troubleshooting done in various domains?
- How to nurture troubleshooting skills among students?
- What can we learn from related disciplines of troubleshooting like problem-solving and science inquiry?

The following sections provide a summary of the investigation carried out based on these questions.

2.2 Characterizing troubleshooting

The word ‘troubleshooting’ is used to represent many situations: repairing a system which was previously behaving as expected, changing the design of a product to satisfy a requirement, analysing a prototype to suggest changes to be done in the final product etc. In other words, literature distinguishes between troubleshooting that happens during design from the one during maintenance. The salient difference between the two is - while designing, it is possible to change the structure and behaviour of the system to achieve the overall function. On the other hand, during maintenance, the correct behaviour is known. One cannot change design specifications usually. In addition, while maintaining a system, there is an actual system giving an actual fault. However, during design, one might have a prototype and not an actual system. In this thesis, we consider troubleshooting of existing systems with defined expected behaviour. In this section, we explore in detail, what such troubleshooting encapsulates.

Troubleshooting is defined as a process which ranges from the identification of problem symptoms to determining and implementing the action required to fix that problem (Schaafstal, Schraagen, & Berlo, 2000). It is mostly categorised as a cognitive task (Dounas-Frazer, Van De Bogart, Stetzer, & Lewandowski, 2016; Schaafstal et al., 2000; Steinberg & Gitomer, 1992) in literature related to problem solving. The problem solving literature classifies types of problems in a spectrum from well-structured to ill-structured. According to Jonassen (Jonassen, 2004), well-structured problems have a well-defined initial state, a known procedure for solving and a known goal state. Ill-structured problems lie at the other end of the spectrum. They often have unknown aspects about the initial and goal states. There might be multiple solution paths and one or more solutions or often no solution at all (Kitchener, 1983). These ill-structured problems are also called wicked problems and they are more often encountered in everyday and professional practice (Jonassen & Hung, 2006). In this spectrum, troubleshooting problems are classified as moderately ill-structured problems. According to Jonassen (Shin, Jonassen, & McGee, 2003), troubleshooting problems have ill-defined initial states, one or more solutions paths and require a reliable conceptual model of the system (one has to figure out what information is needed during different phases). The solution to a troubleshooting problem, on the other hand, is clearly interpreted by a well-defined success criteria. He notes that learning troubleshooting tends to be difficult for students because efficient troubleshooting depends on experience-based rules. The troubleshooting process requires learners to make assessments about the nature of the problem and this changes considerably in terms of system dynamics and complexity.

The researchers use a specific nomenclature to describe various aspects related to the troubleshooting process. We try to familiarize the readers of this thesis to the frequently encountered words and their meaning in the following section:

- Structure: It denotes the basic components in a system that serve a function individually or by working with other structural components. These can be hardware or software components. Example: The IP address of a router, the input and output ports.
- Function: It is the purpose of a component, a system or sub-system. Example: The function of a router is to forward packets.
- Behaviour: The observable attributes of the system that is produced because of the interaction between various structural components based on conditions at a given time. For example, the behaviour of a router is to check the routing table to check the mapping on arrival of a packet.

  Each of structure, function and behaviour can be defined at different levels of abstraction for a given network.
- Fault/ Error/ Bug: ―A state in the system which is the source for the unexpected behaviour.
- Symptom/ Discrepancy: ―The unexpected behaviour in the system caused by the fault.
- Fault/error isolation - Segregating the root cause of the symptoms.
- Fault/error correction - Changing the system to eliminate the unexpected behaviour from the system.

We describe the above-discussed terminology with an example of a simple network where a laptop is connected to the internet via a router. Let us consider the scenario where a user is trying to access a website from a browser in the laptop. The behaviour of the system in this scenario is described below: When the user types in the website URL in the address bar and hits enter, the browser generates a HTTP request to the website. Then the browser communicates with the DNS Server to get the IP address of the required website. After that, it sets up a TCP/IP connection with the website and sends the HTTP request. Here the laptop, the router, the physical cables connecting them, IP address allotted to the laptop and the 'internet' are structural components. However, the laptop is at a different level of abstraction as compared to the 'internet' and the IP address allotted. The laptop can be further divided into more structural components like the browser, operating system, network interface card etc. The functions of the laptop are to create a HTTP request, set up a TCP/UDP connection with the website and so on. The laptop has to send these requests to the router. The function of the router is to forward these packets. Let us assume that the router is not forwarding a specific type of packets. This is an unexpected behaviour. A firewall setting that caused this unexpected behaviour is the fault/bug. Fault isolation is the process of finding out that the firewall setting is the cause. The process of changing the firewall setting to allow the packets to be forwarded constitutes fault correction.

In literature (Jonassen, 2004; Jutten, Schaafstal, & Pel, 1999; Ross & Orr, 2009), troubleshooting has been primarily regarded as an activity that starts with the discovery of an unexpected behaviour and ends with restoring the expected behaviour. However, in some cases (Johnson, 1987) troubleshooting constitutes discovery of unexpected behaviours and finding out the root cause of the unexpected behaviour. The last part of restoring the system to produce expected behaviour is omitted in such descriptions of troubleshooting. In our context, we consider restoring the network to desired state as troubleshooting. The next three sub-sections try to capture the different aspects of characterization of troubleshooting.

2.2.1 Definitions of troubleshooting

In this section, we consider different definitions of troubleshooting as explained by researchers in different domains.

According to Perez (Perez, 2012), troubleshooting is locating the problem or malfunction in a system that is not working properly and then to repair or replace the faulty part. He argues that troubleshooting is closely associated with problem-solving skills that are relevant to a specific domain like computer programming, biology, medicine or psychology.

MacPherson (Macpherson, 1998) described the essence of troubleshooting as "…the question of how to progress from a given starting situation to a desired end situation". Here all problems contain an initial or "what is" state, a goal or "what is desired" state and a solution path.

Schaafstal and colleagues (Schaafstal et al., 2000) interpret troubleshooting as a cognitive task which includes searching through a problem space of possible causes for likely causes of the fault. After detecting the fault, troubleshooting includes repair or replacement of the faulty device. This usually takes place by searching for a component that is not behaving as expected and then searching for actions that would fix it back to desired behaviour.

Woods (Donald R. Woods, 2006), who works in the domain of chemical plants, defines troubleshooting as a problem where something unexpected happens to such an extent that it is perceived that a corrective action is required.

Ross (Ross & Orr, 2009) defined troubleshooting as a process which ranges from identification of a problem, uncovering the symptoms, and implementing the action required to fix the problem.

To summarize, the definitions describe the actions to be performed by a person to troubleshoot a system, the space in which troubleshooting will be performed as a concrete system and as an abstract search space. These definitions of troubleshooting consider it as a predominantly cognitive task. However, they do not specify the order (if any) of these actions to be performed. That leads us to look for descriptions of the process of troubleshooting.

2.2.2 Process of troubleshooting

In this section, we present various ways in which the process of troubleshooting has been illustrated by researchers.

Flesher (Flesher, 1993), following the footsteps of Johnson (Johnson, 1987), captures the process of troubleshooting in two phases: Hypothesis generation and hypothesis evaluation. Figure 5 describes the detailed flowchart given by Johnson to troubleshoot.

Initially, information is foraged to create a representation of the problem that is sufficient to create one or more hypotheses. This is followed by evaluating the hypotheses by acquiring more information and interpreting it in the context of the troubleshooting problem.

Later, Johnson (Johnson, 1995) revised his description of the troubleshooting process to having three phases: problem representation, fault isolation and solution verification. In the first phase, problem representation, the focus is on creating an initial frame of reference by the troubleshooter based on her assumptions and understanding of the problem. Information foraging happens to append this representation with more detail as required. Fault isolation is done through cycles of hypothesis creation and testing. This is done to reduce the size of the search space when done by expert troubleshooters. Solution verification is the phase that happens after fault isolation. This is done to verify that correcting the fault will actually resolve the undesired behaviour.

Axton (Axton, Doverspike, Park, & Barrett, 1997) analyses troubleshooting from the lens of cognitive ability and information processing requirements in the context of mechanical troubleshooting. According to this study, troubleshooting includes three phases: inspection, troubleshooting & search for actions that fix the discrepancy. Inspection involves checking what components are working as expected. Troubleshooting phase consists of narrowing down the subsystems till the component that is producing undesired behaviour is found. Then actions to fix that undesired behaviour are performed.

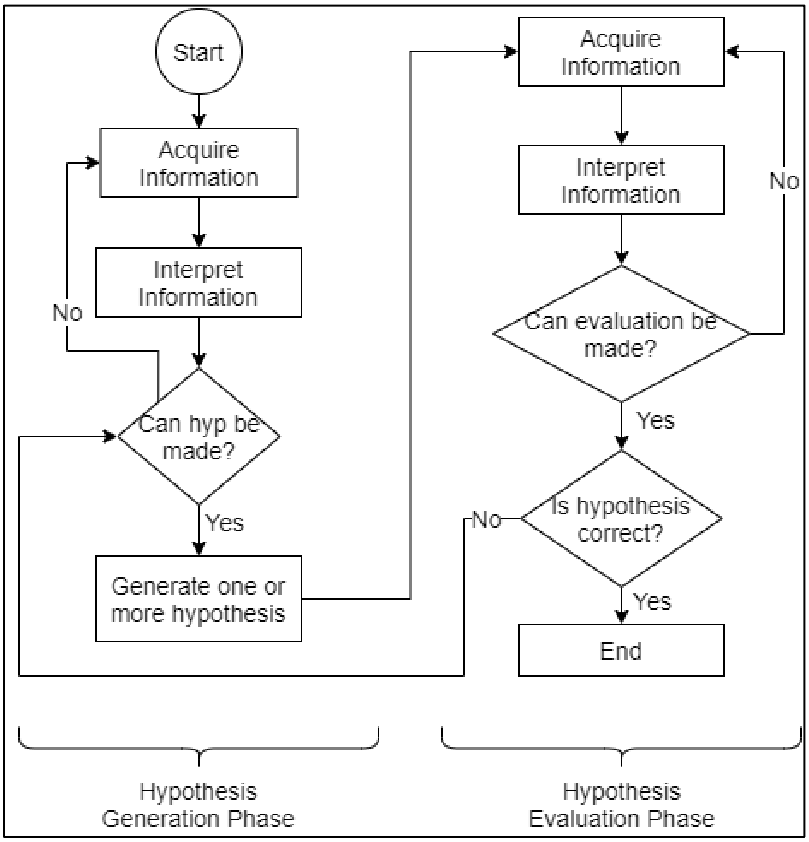


Figure 5: Technical Troubleshooting Model adapted from Johnson (Johnson, 1987)

Schaafstal (Jutten et al., 1999; Schaafstal et al., 2000; Schraagen & Schaafstal, 1996) and colleagues divide troubleshooting into four sub-tasks: formulate problem description, generate causes, test, and repair and evaluate. Problem description includes the troubleshooter's assessment of correctly working and incorrectly working components of the system. Generating causes for a previously encountered problem is done by the process of recognition. In case of unfamiliar problems (a problem that is encountered for the first time), a troubleshooter has to use reasoning and functional thinking to generate probable causes. Schaafstal and colleagues emphasize the use of right testing methods and means to evaluate a cause. They also discuss the benefit of having an expectation about the test result before the actual testing. After identifying the correct cause, narrowing down to the lowest replaceable unit and replacing it constitutes the repair part. The evaluation part consists of checking whether the replacement resolved the fault.

Like Schaafstal & colleagues, Woods (D. R Woods, 2006) differentiates the process of troubleshooting of previously encountered problems with new problems. According to him, a problem solver builds an internal representation of the problem. This includes matching the ‘cues’ and recognizing ‘patterns’ to classify it as a familiar or unfamiliar problem. The familiar problems are solved like ‘exercise’ problems whereas unfamiliar ones utilize domain knowledge and problem solving skills.

Ross and Orr (Ross & Orr, 2009) use a model called DECSAR to define and teach the process of troubleshooting. This involves 6 steps: Define the problem, Examine the environment, consider the Causes, consider the Solutions, Act and test, and Review the troubleshooting. The first step of this cyclic process involves the troubleshooter considering different options to maximize her chances of finding an effective solution. While examining the situation, she has to consider the functioning and malfunctioning components by making systematic observations. She then has to propose a ranked list of probable causes in the third step. As a part of considering the solution she has to have multiple courses of action if the first solution is ineffective. The highest ranked solution is tested in the next step and the solution is verified for efficacy. The last step is where the troubleshooter reviews the process undertaken with a focus on developing better understanding of the system and improving her troubleshooting skills.

We can see that the conceptualization of troubleshooting started as a type of problem solving. This can be seen in earlier process descriptions where the process steps are similar to that of a general problem solving process or they highlight only the parts where troubleshooting differs from general problem solving.

Table 1: Synthesis of process of troubleshooting according to various researchers

Researcher	**Problem Re-presentation**	**Analysing information**	**Planning Test**	**Test**	**Repair**	**Evaluate**
Flesher	Hypothesis generation		Hypothesis evaluation			
Johnson	Problem representation		Fault isolation		Solution verification	
Axton		Inspection		Trouble-shooting	Fix the discrepancy	
Schaafstal	Formulate problem description	Generate causes		Test	Repair	Evaluate

Woods	Engage with the problem	Explore & Define the stated problem	Plan to solving the problem	Carry out the plan	Check the accuracy and pertinence	
Ross	Define the problem	Examine the environment	Consider causes and solutions	Act and test		Review trouble-shooting

With growing research specific to the context of troubleshooting, we can see the nuances of troubleshooting appearing in the process descriptions like the need for evaluation and reflection after troubleshooting a fault. Form these research publications, we borrow and synthesize the process steps of troubleshooting (discussed in section 2.4). In this section, we summarised what needs to be done to troubleshoot, according various researchers. Table 1 provides a summary of this. In the next sub-section, we discuss about the research on mental capacities required for doing these process steps and their influence on troubleshooting skills.

2.2.3 Cognitive processes involved in troubleshooting

Troubleshooting is primarily interpreted using a cognitive lens throughout the literature. It is analysed mostly in the contexts where an individual is troubleshooting a system as opposed to a group of individuals. In these studies, the prior knowledge of the troubleshooter, her mental model and the actions performed by her are considered. Another lens that has been used is to interpret troubleshooting as a situated task (Alby & Zucchermaglio, 2009; Bogart, Dounas-frazer, Lewandowski, & Stetzer, 2017). Here the context includes the environment in which troubleshooting is being carried out. This involves the resources referred, artefacts generated, other people contacted etc. In this thesis, we consider a student working with a faulty network. We primarily take a cognitive lens to describe and analyse troubleshooting.

Teaching-learning of troubleshooting is crucial in some areas like naval and aircraft maintenance. Researchers in those areas employed cognitive task analysis of troubleshooting to gain more understanding about the types of knowledge, strategies and cognitive subtasks employed in competent troubleshooting (Johnson, 1995; Jonassen & Hung, 2006; Schaafstal et al., 2000). In this section, we describe the insights obtained by cognitive task analysis of troubleshooting.

2.2.3.1 Types of knowledge that facilitate troubleshooting

The section below summarizes the findings from various domains and different researchers (Gog, 2006; Jonassen & Hung, 2006).

a. Domain Knowledge: It is the theory and principles of the domain upon which the system is designed. In the context of networking, this constitutes theoretical concepts like routing, switching, protocols and algorithms.
b. System knowledge: It is the understanding of behaviour, function and structure of a system and the components within it. In the context of networking, this means being able to identify the structural components that make up a network, to describe the interconnections between them and to assert how the components combine to produce the desired behaviour.
c. Procedural knowledge is needed during testing to implement the test procedures and handle equipment.
d. Strategic knowledge is the knowledge about systematic approaches and heuristic techniques used during troubleshooting. An example is to divide the network into two parts and checking if each of them is working correctly.
e. Experiential knowledge is the knowledge acquired by experienced troubleshooters. It is used to generate hypothesis and narrow down the problem space.

As we can see from the above list, skilled troubleshooting involves a complex gamut of knowledge and strategies. This makes it harder for students to learn and practice in real time without scaffolds.

2.2.3.2 Mental capacities that support troubleshooting

While the types of knowledge are those that can be acquired/ can be looked up by a troubleshooter, the researchers distinguish them from mental capacities. They hint at mental capacities having some inherent component to it. An example of a mental capacity is working memory. Since troubleshooting is a complex task, one can hypothesize that working memory becomes an important mental facility to manage. In this section, we describe working memory and other mental capacities that support troubleshooting.

- Working memory: During troubleshooting, troubleshooters have to administer the cycles of hypothesis generation and testing along with domain knowledge and the troubleshooting strategies. Working memory can be easily overloaded while doing so. As the system complexity increases, this process becomes more demanding. Therefore, the performance of troubleshooting directly depends on the availability of working memory. Researchers agree that reduction of load on working memory is one of the main goals of troubleshooting teaching environments (Jonassen, 2010; Ross & Orr, 2009). Providing a visual summary of the troubleshooting process to reduce the load on working memory is suggested in multiple interventions to teach troubleshooting(Ross & Orr, 2009; Schaafstal et al., 2000). PHyTeR pedagogy has a structured task environment designed to reduce the load on working memory.

- Causal Reasoning (ability): Causal reasoning is required for understanding the cause-effect relationship between different components of the system. Causal reasoning enables the troubleshooter to make predictions about the faults and test hypotheses. It was shown that students tend to give rudimentary causal explanations especially when the domain is not very familiar to them (Perkins & Grotzer, 2000).

- Analytical Reasoning (ability): Analytical reasoning is the domain independent reasoning – the ability to reason without the components in a situation affecting the reasoning process. This reasoning helps in filtering task relevant information. In a study it was found that domain independence is the strongest predictor of diagnosis (Moran, 1986).

Along with these knowledge and mental capacities, metacognition is argued to influence for troubleshooting. We summarize the relevant literature on metacognition in the next section.

2.2.3.3 Metacognition in case of problem solving like troubleshooting

Metacognition is described as an explicitly performed active process directed towards one's own cognitive activity. It includes a closed, dynamic loop of reflecting, monitoring, evaluating and regulating one’s on-going tasks (Berardi-Coletta, Buyer, Dominowski, & Rellinger, 1995; Son & Schwartz, 2002). In other words, meta-cognition involves knowledge about cognition and regulation of cognition (Hong, 1998). Metacognition is shown to be an essential component for ill-structured problem-solving, science and mathematical problem solving (Ge, Law, & Huang, 2016; J. Y. Kim & Kyu, 2019). In the context of troubleshooting electronic circuits, Van De Bogart and colleagues (Bogart, Dounas-frazer, Lewandowski, & Stetzer, 2015) observed that, the regulatory mechanism of metacognition enables students to crystallize their domain understanding. They also reported that it helps students in taking strategic decisions during hypothesis testing. The TELE PHyTeR includes several activities to evaluate and reflect student decisions. These are based on the theory of metacognition.

In this section, we summarized the process of troubleshooting, cognitive processes and mental capacities involved in troubleshooting. We utilize this in creating the learning activities of PHyTeR. In the next section, we describe our learning from expert troubleshooting.

2.2.4 Insights from Expert Troubleshooting

Understanding how experts troubleshoot a network might be helpful to interpret different aspects of troubleshooting. Previous research has focused on differences in problem solving abilities of experts and novices. Some authors define an expert as having high competence in problem solving and novices as being familiar with problem solving, but exhibiting poor performance (Schunk, 2012). Others, however, have described experts and novices as differing in domain-specific knowledge (Simon, 1973). Few other researchers suggest that experts do not possess greater knowledge of problem solving strategies than novices (Zimmerman & Magda, 2003) but that experts’ knowledge was organized in such a way that access to relevant information was almost instantaneous (Larkin, McDermott, Simon, & Simon, 1980).

Farnham-Diggory (Farnham-Diggory, 1994) describes three paradigms of instruction based on the interpretation of i) the difference between experts and novices ii) the approach a novice follows to become an expert. The first paradigm, called behaviour, assumes that novices and experts can be measured using the same scale, only the quantity of a measure varies between them. A novice incrementally becomes an expert by accumulating skills and knowledge that lead to higher outcomes on that scale. The second paradigm is called development. Here, the experts are considered to have qualitative models that are different from the novices' model in multiple complex ways. The model includes one's personal beliefs, assumptions and explanations. Instruction happens when the novice models are questioned, challenged and contradicted. Thus by giving them opportunities to change their model, a novice becomes an expert with a new way of thinking and a comprehensive qualitative shift. The third paradigm is apprenticeship. A novice and an expert, in this paradigm, are described as 'from different worlds'. A novice becomes an expert by the mechanism of 'acculturation'.

To summarise, all these different paradigms think of instruction as some sort of bridge between experts and novices and investigating how experts think/act/behave in a situation is useful to get a sense of the learning goals of a novice. The researchers have explored the problem-solving process of these experts to understand strategies used to solve, the resources needed by them, actions performed and the types of knowledge utilized during problem solving etc. Though it might not be possible for novices to do everything that an expert does, understanding the process of an expert will help in creating learning objectives, designing scaffolds and question prompts in the design of a learning environment.

Experts organize domain knowledge in different levels of abstraction (Besnard & Bastien-Toniazzo, 1999; Johnson, 1987; Larkin et al., 1980; Schaafstal et al., 2000). The organization of domain knowledge in an experts' schema is based on experience in addition to domain knowledge. This organization helps the expert to move forward systematically, sort out the irrelevant information quickly and reduce working memory load (Ross & Orr, 2009; Teague & Allen, 1997). An expert tries to troubleshoot a fault by comparing it with previously solved similar tasks.

In a study, the theory instructor was found to have difficulties when asked to troubleshoot a system that he was teaching (Schraagen & Schaafstal, 1996). Also in the same study when two experts in one radar system were asked to troubleshoot another radar system, they failed in troubleshooting the system. This implies that the knowledge of the domain is not of much help unless it is combined with troubleshooting strategies. In addition, the contextual information is equally important. There are studies showing that when an expert is faced with a new problem, in a new domain, his/her recall is similar to those of novices (Besnard & Bastien-Toniazzo, 1999).

Based on the above-mentioned literature, we summarize that:

- Experts have a rich mental model. This helps them almost like a simulation. They can mentally do operations like ‘run the simulation’ and predict the results.
- One of the main differences between experts and novices is in the organization of knowledge.
- Experts are skilful in following top-down and breadth-first approaches to reduce the problem space.

Though we cannot expect novices to have a rich mental model or to construct one before troubleshooting, we can provide them the affordance of simulation externally in a learning environment. In addition, scaffolds to help them organise the learning activities can be provided to the students by appropriately structuring the learning environment.

After exploring the literature about troubleshooting, we surveyed literature on related domains like ill structure problem solving and science inquiry. We summarise the learnings in the next section.

2.3 Related Domains

In this section, we elaborate on two domains that are related to troubleshooting: hypothetico-deductive reasoning and ill structured problem solving. These are well-established domains with rich and detailed literature. We give a very brief summary of the relevant results fom these domains.

2.3.1 Hypothetico-deductive reasoning or Science inquiry

Scientific inquiry is the process followed by scientists while studying the natural world. It is the process based on which scientists propose a theory or description related to a natural phenomenon. Inquiry learning originated from the practice of scientific inquiry. It emphasizes gathering, analysing data, asking questions and constructing evidence-based arguments (Krajcik & Blumenfeld, 2006; Kuhn et al., 2000). A related field, problem based learning (PBL) originated from medical education and this emphasizes the hypothetico-deductive reasoning process (Hmelo-Silver, 2012).

The similarity between these and troubleshooting is in the cognitive tasks involved. Many of the environments that employ inquiry learning provide explicit practice to students in generating a hypothesis, gathering evidence to prove/disprove the hypothesis and constructing arguments based on these hypotheses. However, the nature of hypothesis is a little different in both contexts: In inquiry learning context, a hypothesis is devised to describe a natural phenomenon. Therefore, it will be more precise than a hypothesis in the context of troubleshooting. For example, a troubleshooting hypothesis would be 'there is some issue with the router'. After this, one can start comparing the desired behaviour of the router with the faulty observed behaviour. Then one narrows down to one or more variables causing this undesired behaviour. In case of scientific inquiry, the direction is reversed. While generating a hypothesis, the 'value' for one or more variables is known. It is used to construct a description of how a system behaves. The comparison here is between the expected behaviour according to the constructed description and the actual behaviour of the system (Siponen & Klaavuniemi, 2020).

Since the cognitive tasks involved in troubleshooting are very similar to those in inquiry learning, we looked at that literature related to inquiry learning and problem based learning to understand the difficulties faced by students and design of learning environments. The target audience in inquiry learning is mostly middle school and high school students and the domains explored are often 'scientific disciplines' (Hmelo-Silver, 2012). However, problem-based learning is employed in tertiary education especially in the medical domain. The benefits of PBL in developing skills like troubleshooting has also been reported (Donald R. Woods, 2012).

2.3.2 Ill Structured Problem solving

The domain of problem solving includes a continuum of problems from well-structured (logical, arithmetic problems etc.) to ill-structured (estimation, design etc.) problems. Along with the classification, research in this domain has identified the strategies to solve these types of problems (Jonassen, 1997; Pretz et al., 2003), identified mechanisms that experts use to solve these problems and teaching strategies employed to teach them.

Experts in problem solving generally have well-organized domain knowledge (Dunbar, 1998; Maloney, 2011). Novices tend to 'store' knowledge as isolated facts and procedures. On the other hand, experts' knowledge organisation reflects contexts of applicability. These researchers have recognized that the experts are also malleable in using knowledge with little effort to focus their attention.

Some important results that are relevant to us from these domains are:

- Novices have difficulty in generating hypothesis, in simulating a system and constructing arguments (van Joolingen & De Jong, 1991)
- Scaffolds are not just helpful but also necessary in learning environments developed to teach complex problems (Hmelo-Silver, Duncan, & Chinn, 2007; Jong & van Joolingen, 1998; Lazonder, Hagemans, & Jong, 2010)
- Types of scaffolds used in in these contexts for different learning goals and difficulties. Ways to structure the complex learning based on how experts organise knowledge, set goals and sub-goals, do information search (Arnold, Kremer, & Mayer, 2014; Eslinger, White, Frederiksen, & Brobst, 2008; Jonassen, 2010; Quintana et al., 2004; Reiser, 2004; Sharma & Hannafin, 2007; Xun & Land, 2004)

The broad task structure of PHyTeR pedagogy focuses on the hypothesis-testing cycle adapted from science inquiry literature. In addition, we have borrowed some scaffolds that were developed in the context of science inquiry and ill-structured problem solving. For example, PHyTeR gives a list of hypothesis to choose from instead of asking the student to formulate the hypothesis from scratch. We discuss the pedagogy and scaffolds in detail in section 5.1. Based on the review of literature described in previous sections, we now describe the process of troubleshooting as applicable to our context.

2.4 Our definition of troubleshooting skills

Based on reviewed literature, we characterize troubleshooting in the following section in the context of a computer network. We construe that the process of troubleshooting consists of 4 phases:

- Problem Space Understanding
- Hypothesis Generation
- Hypothesis Testing
- Result Interpretation

We describe these phases in the following subsections.

2.4.1 Problem Space Understanding

Having a good understanding of the problem space includes having thorough understanding of the concepts in the domain and being able to mentally simulate the working of the system in different contexts (Johnson, 1987). Problem space understanding, performed as an initial analysis of the system, consists of the troubleshooter identifying the undesired behaviour being produced by the system and recognising the relevant structural and functional components in producing the desired behaviour. This information will be at different levels of abstraction like system level, sub-system level, component level etc. The information gathered and representation evoked during this phase are intended to understand the problem space in terms of its goal, constraint, criteria etc. of the given system ultimately leading to generating hypotheses (Johnson, 1995). Jonassen (Jonassen & Hung, 2006) observes that the performance of troubleshooting depends on the functional understanding of the system. Hence, he points out that a troubleshooter needs to integrate different types of knowledge related to the system into a coherent mental model.

For a person familiar with the system, it is easier to identify this relevant information. Because of their experience, they have a robust conceptual model including their assumptions based on past experiences. This enables them to identify the required inputs/ outputs easily. A novice troubleshooter, because of the lack of a rich mental model, will have to explicitly spend effort to gather the required information. Schaafstal and colleagues (Schaafstal et al., 2000) suggest the novices to identify the parts of the system that are working as desired and also the parts that are non-functioning. A similar recommendation is made by Ross & Orr (Ross & Orr, 2009) to will produce a coherent representation based on which further decisions can be made. They recommend noting down the observations, which include the parts that function as expected and the non-functioning parts. This model might get updated in between when a troubleshooter acquires new information by testing a hypothesis or observing new behaviour in the network.

In the context of a computer network, it is the phase where one develops a model/representation of the network. This model contains network devices, their configuration, interconnections between the devices etc.

2.4.2 Hypothesis Generation

Once the troubleshooter is familiar with the problem space, she comes up with plausible reasons for the undesired behaviour of the system. Each such possible reason is called a hypothesis. It is a tentative speculation about the fault. It might be as broad as a part of the network being faulty or as specific as a configuration setting of a device. The hypotheses can be at different levels of abstraction. For example:

- System – Ex: The internet isn’t working
- Sub-system – Ex: The problem is at my end and not the ISP
- Device – Ex: The router isn’t working
- Component – Ex: Firewall is blocked

Like in the problem space understanding phase, the way in which a troubleshooter generates hypotheses depends on her experience. When an experienced troubleshooter encounters a faulty situation, hypotheses are fired up based on previous encounters of similar faults. However when a novice troubleshooter encounters the fault, she is required to employ functional understanding of the system and analytical reasoning about it (Schaafstal et al., 2000). This difference in generating hypotheses has also been a marker between experts and novices in troubleshooting.

For someone who is familiar with the system, frequently occurring faults and their causes and fixes are known too well – so they might skip the procedure of comparing and generating hypotheses. Hypotheses are then generated from memory by a process of recognition. Rasmussen (Rasmussen & Jensen, 1974) calls this as symptomatic search because the symptoms act as a trigger. Klein (Klein, 1989) calls it recognition primed decision-making.

In order to construct hypotheses in an unfamiliar system, a troubleshooter reconstructs the way a functional block works without any fault. Then they compare it with the faulty working system at hand. Any inconsistencies between the two might lead to the generation of one or more hypotheses (Klein, 1989).

Johnson (Johnson, 1995) observed that even in cases when the high and low performers generated a similar number of hypotheses, they vary in their ability to evaluate the information and interpret its meaning. Expertise and nature of work (whether they are designers or technicians by profession) plays a role in how the initial frame of reference and set of hypotheses are generated. It is observed (Macpherson, 1998) that designers have a different initial frame of reference than that of mechanics even when their troubleshooting processes are similar. Mechanics were found to ignore design flaws and focus more on aspects like wear and tear, corrosion etc.

A troubleshooter might come up with one or more hypotheses at a time. In such situations, prioritization helps to conduct investigation in a structured manner. Johnson (Johnson, 1995) identified a number of strategies that are utilized to prioritize one hypothesis among many generated: Trial and error, Topographic, Half/split, Exhaustive, Functional strategies.

Jonassen (Jonassen & Hung, 2006) critiqued these strategies mentioning that out of these mentioned strategies, topographic, half/split and functional are desirable. Schaafstal and colleagues (Schaafstal et al., 2000) report that troubleshooters do hypothesis prioritisation based on 2 factors:

- The likelihood of an hypothesis being the cause for an error
- The interdependence of the component considered and the symptoms seen

In the field of medical diagnosis, Bowen (Bowen, 2006) suggests that asking the troubleshooters to give a justification along with the prioritisation helps them to create links between what is already known about the problem and relevant hypotheses.

Johnson (Johnson, 1995) postulated that an important goal of hypothesis generation is to reduce the problem space. A troubleshooter can systematically go about discarding the parts of the system that are working correctly from further consideration. A hypothesis acts as an anchor to focus during the testing and result interpretation phases. After the prioritisation of hypotheses, Ross and Orr (Ross & Orr, 2009) suggest that a novice troubleshooter should come up with multiple courses of actions (testing methods and means) for the hypothesis selected. This will help her in keeping the troubleshooting process on track.

In the case of novices, we do not expect them to create multiple hypotheses at one point of time. We believe that they have to pass the hurdles of generating a hypothesis before learning the need to do prioritisation. Therefore, we emphasize on the generation of a hypothesis in our intervention.

2.4.3 Hypothesis Testing

After a troubleshooter prioritizes that a particular hypothesis might be the cause for the error, she has to confirm if it is the actual cause or not. Hypothesis testing is done to confirm this. This includes choosing testing instruments like what commands to execute and check, what log files to look into etc. She then performs the actual test.

Once they have a hypothesis, experts have a repertoire of testing means and methods from which they can select an optimum test (in terms of time and cost efficiency). Novices lack the knowledge related to planning and conducting tests (Kester, Kirschner, & van Merriënboer, 2004).

Schaafstal and colleagues (Schaafstal et al., 2000) suggest that choosing the right testing methods and testing means (tools and equipment) to test the hypotheses generated are important. In addition, in order to interpret the results of the test, he says, having to have an expectation of the outcome of the test helps. Testing will continue until the correct cause for the fault is found and then the troubleshooter has to eliminate the cause.

2.4.4 Result Interpretation

In this phase, the first step is to analyse the result of the test and decide if the hypothesis is accepted or rejected. The result of the test is used to accept or reject the hypothesis. Further one has to consider if the model representing problem space has to be changed or a new hypothesis has to be generated and tested. This iterative process continues until the correct cause for the fault has been found.

Along with these phases, we think reflecting on one's process of troubleshooting is necessary to transfer their learning to a similar new problem. Reflection includes abstracting what actions and decisions were helpful to solve the problem and taking note of what actions and decisions did not help. Doing reflection might help them to abstract how each sub-processes of troubleshooting will contribute towards the problem (Bogart et al., 2015; Eslinger et al., 2008).

Until now, we summarised the literature on troubleshooting skills, what it constitutes of, and related domains like science inquiry. Now we shift our focus on literature related to teaching troubleshooting. The next section describes various tools and approaches developed by researchers to teach troubleshooting.

2.5 Nurturing troubleshooting skills

2.5.1 The need for teaching troubleshooting skills

Troubleshooting is an important part of an engineer's work. Be it developing a new system or maintaining an existing system, engineers have to fix things. In the case of programmers and software developers, a troubleshooting scenario might be finding an error in a simple program to find the cause for unexpected behaviours in complex systems like a network, a data server etc.

We have seen in previous sections that troubleshooting is an ill-structured problem requiring human troubleshooters to involve in complex activities like analysing the behaviour of a system, generating multiple hypotheses that are plausible causes for the problem, keeping track of the troubleshooting process etc. The aspect of complex ever-changing technology adds to the ill-structured nature of troubleshooting in case of IT professionals. This implies the professionals need troubleshooting training to execute their jobs smoothly. An ill-structured problem like troubleshooting needs to be taught using carefully designed interventions keeping in mind the learner needs and learning objectives.

2.5.2 Role of technology in learning troubleshooting skills

The IT professionals will be interacting with different technologies for their routine work, the medium of interaction primarily being computers. That being said, there are a lot of checklist/ flowchart based decision making tools printed on paper to use during troubleshooting episodes. So one might question the necessity of developing technology based environments to develop troubleshooting skills. Technology has been argued (Guzdial, 1994; Reiser, 2004) to be more helpful in developing learning environments because of its ability to provide different types of scaffolds and adapting ability. In this context, researchers argue that technology allows one to simulate the fault and shift easily between different levels of abstractions. They also argue that integrating different aspects of problem solving like information search, planning and reflection can be done in a holistic manner.

2.5.3 Choosing a pedagogy for troubleshooting

Troubleshooting is a complex task. It requires one to have a functional understanding of the domain and logically reason about the fault. It requires one to shift often between two viewpoints: a particular hypothesis and the broad context of the problem. It is similar to what Dunbar and Klahr, in the context of scientific reasoning, term as searching in dual spaces of hypothesis and experiment (Klahr & Dunbar, 1988). Significant efforts are required to plan and execute a test and search for required information. It includes assimilating required knowledge, skills and attitudes. According to constructivist paradigm (Fosnot & Perry, 1996), learning has been conceptualised as an active, self-regulated, cumulative and situated process of knowledge building. Instruction is viewed as an enabler of this learning process through a set of activities. This includes helping students, guiding them to actively synthesize information, monitoring their performance and giving feedback according to the learning activities (Elen 1995, Merrill 2013).

There are various educational approaches prescribed for such complex learning like problem based learning (De Graaff & Kolmos, 2003), guided discovery learning (De Jong & Lazonder, 2005; Hmelo-Silver et al., 2007), case based learning etc. In addition, there are models like cognitive apprenticeship (Collins, 2006; Collins, Brown, & Newman, 1987), constructivist learning environments (Jonassen & Rohrer-Murphy, 1999), four component instructional design model (Van Merrienboer, Clark, & de Croock, 2002). At one end of the spectrum, there is a very structured approach – of providing students with checklists, worked examples etc. In other words, here, the rules are given and the students have to practice applying these rules in different contexts. At the other end is the completely open-ended approach, called as pure discovery learning. Here, the troubleshooting problem is given to the students. The students have to 'discover' both the rules and how to apply those rules in a given context. Researchers have argued against pure discovery approaches because of load it creates on the working memory of the student (Kirschner, Sweller, & Clark, 2006). Another argument is that pure discovery neglects useful teaching approaches that are cognitively active and behaviourally inactive (Mayer, 2004). However, the researchers arguing from different perspectives recommend following a guided discovery based approach with carefully designed scaffolds (Hmelo-Silver et al., 2007; van Merriënboer & Kirschner, 2017). We follow one such approach – inquiry learning – as the basis of our pedagogy. We describe the details of how this is implemented in our context in section 5.1. Next, we list the existing approaches to teach troubleshooting skills.

2.6 Existing approaches/tools to teach troubleshooting skills

We reviewed various available network simulators to see if they support network troubleshooting skills. We describe the synthesis of this review in next sub-section. In addition, we reviewed the literature related to teaching troubleshooting. We describe six tools and approaches used in domain other than computer networking in the subsequent sub-section.

2.6.1 Tools used to teach Computer Networking

In this section, we describe various tools that are used in teaching computer networking. The tools provide specific features to create, simulate and analyse simple to complex networks. Some of these also emulate the network devices by running the actual codes used in those devices. However, none of the tools has troubleshooting process specific features that support students to do hypothesis testing. Though it is possible for an expert to retrieve most of the required information in NS-2 or Wireshark, these tools are overwhelming for novices. They do not have any explicit scaffolds designed to teach the process of troubleshooting.

We looked for tools that can support building a troubleshooting learning environment on top of it. However, we found that the tools that supported our requirements were mostly proprietary and did not allow addition of more features. On the other hand, tools which had open-source code and allowed addition of features did not support the types of visualizations we needed and/or was buggy. Some tools (esp. emulators) were not feasible to use because of their large memory requirement. Table 2 summarizes our analyses.

Table 2: Tools used to teach computer networking

Tools	Purpose/feature	Limitation

ns-2	Performance Analysis and Evaluation of computer networks	No protocol level analysis
OPNET	Modeling and Simulation of Computer Networks & Network Configuration Provides Networks diagrams	Proprietary
cnet	Network Simulator enabling experimentation with protocols in any layer of LAN or WAN	No guidance on the process of troubleshooting
JASPER	Protocol simulator that can be used as an aid to enhance teaching and learning communication protocols	Pre-defined set of protocols. New protocols have to be scripted using JavaScript and run
WebLan - Designer	WebLan-Designer is a Web-based tool for interactive teaching and learning both wired and wireless LAN design and class demonstration	Only Network level protocols- component level protocols are missing, No guidance on the process of troubleshooting
DlpSim	Data link protocol simulator used to enhance teaching protocols through simulation	Only Data Link Protocols, no more supported
WebTrafMon	The WebTrafMon is a Web-based system for network analysis and traffic monitoring	Focuses exclusively on network analysis and traffic monitoring
Cisco Packet Tracer	A simulator which supports building complex networks with various types of devices.	Proprietary, Some commands are specific to cisco devices
NetMod	The NetMod is a network modeling tool which uses some simple analytical models, providing designers of large, interconnected local area networks with an in-depth analysis of the potential performance of such systems	Only useful as a demonstration tool in classroom, No longer in circulation

iNetwork	Network is an interactive software tool for teaching and learning data communication networks	No guidance on the process of troubleshooting
Wireshark	A comprehensive tool for analysing packets. Supports live traffic analysis.	No guidance on the process of troubleshooting

In the next sub-sections, we list and describe various pedagogies designed and developed to teach troubleshooting skills. In the end we summarize what we learnt from the above systems - in terms of pedagogy, scaffolds, what works, what doesn't etc.

2.6.2 Troubleshooting Learning Environment (TLE)

This is an architecture of a learning environment proposed by Jonassen (Jonassen & Hung, 2006) for building computer based troubleshooting learning environments. There are three main system components in this model. These are a multi-layered system model, a case library and a simulator. This model also includes two instructional components: practice and worked examples.

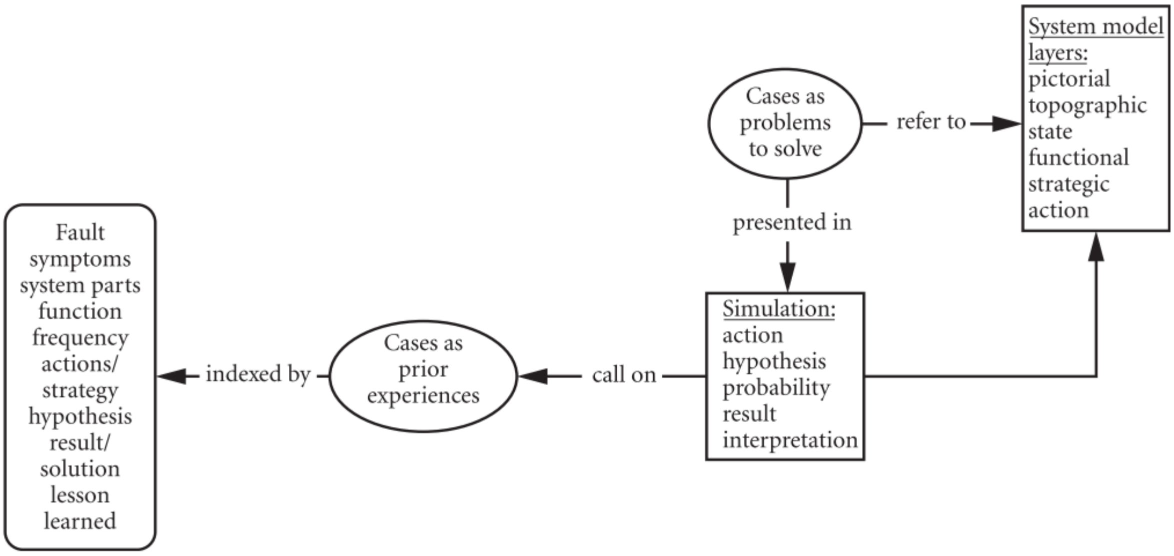


Figure 6: Architecture of troubleshooting learning environment proposed by Jonassen

The underlying assumption of this architecture is that solving troubleshooting problems is the most effective way to learn troubleshooting. In other words, it tries to 'build experience' among students to make them better troubleshooters. The learners are presented with problems as cases with symptoms. The learner can refer to the multi-level model of the system being troubleshot and use a simulator to test the system. This multilevel system model supports exploration of configuration of the components and controls in a device. Many layers with each layer depicting one aspect of the system are overlaid in the model such that the learners can interact with them using zoom in and zoom out functions. The interactions are supposed to facilitate the learner in inferring the procedural aspects of the system. The various layers include pictorial, topographic, state, functional, strategic, and action layers.

The simulator is the core of the TLE. The actions that a learner does in a simulator is based on the PARI system (Hall, Gott, & Pokorny, 1995) of analysis (Precursor, action, result, interpretation). In the precursor stage, the learner studies the problem description with symptoms. Based on that, she has to select an action and a hypothesis corresponding to that action. If the hypothesis is not an appropriate 'justification' for the action being performed, the TLE provides immediate feedback. Then the learner has to attach the hypothesis with a probability that the hypothesis is the actual cause. This is intended as a metacognitive prompt to help the learner assert her certainty in hypothesis selection. When the learner sees the results of her 'actions', she has to select an interpretation of the result value. A feedback is given if the interpretation is inconsistent with the result. Then, she has to select another interpretation. Each case that is solved by the learner is added to her personal case library. The case library is a collection of system faults storing as many troubleshooting ‘experiences’ as possible. That is, it consists of stories about how experienced troubleshooters have solved similar problems.

Take away: This model gives a useful way to integrate different types of information while trying to balance the load on the learner's working memory. This happens via the multi-level system model and integrated tasks based on the PARI model. This influenced the activities in PHyTeR pedagogy focusing on understanding the network at different levels of abstraction.

2.6.3 DECSAR

The six-step troubleshooting strategy of DECSAR (Ross & Orr, 2009) was built to model effective troubleshooting. The steps are: **D**efine the problem; **E**xamine the situation; consider the **C**auses; consider the **S**olution; **A**ct and test; **R**eview the troubleshooting.

This method is intended to support a student’s transition into the practice of structured troubleshooting by providing them troubleshooting templates. They argue that these templates reduce the working memory load and provide a conceptual framework of troubleshooting. Below is a brief overview of what happens in each step of DECSAR:

- Defining the problem: In this step, students are advised to carefully consider the cause for the malfunction. This would maximize their chance of finding a solution.
- Examine the situation: Here students make systematic observations about the system. They are supposed to identify both functioning and mal-functioning components. They are asked to note down these observations for future reference.
- Consider the cause: In this step, students have to come up with possible reasons for the malfunction. Then they have to prioritize these reasons according to the probability that it is the actual reason for the malfunction.
- Consider the solution: Here the students have to ensure that they have multiple sets of actions and rank them. This is to help them have a plan in case the first course of action fails. Then they select the highest-ranked solution for the next step.

- Act and test: Here students compare the current malfunctioning system with notes made during the second step.
- Review troubleshooting: This happens once the system is repaired. They have to review their troubleshooting to get a complete understanding of the system. They have to look for errors/shortcuts in their troubleshooting process.

Takeaway: A significant contribution of this model has been the addition of reviewing step. The reflection activity in PHyTeR, based on this step, reinforces the process of troubleshooting. We think that the final step in the model - to review one's own troubleshooting process is useful for reinforcing the conceptual framework of troubleshooting among learners.

2.6.4 Sherlock

Sherlock (Lesgold, 1988) is a computer-based environment developed for sophisticated troubleshooting tasks in the electronic components of F-13 aeroplanes belonging to the air force of the USA. The electronic components of these aeroplanes are routinely maintained and troubleshot by attaching them on a test station. The technicians make use of prescribed test routines to isolate the fault in these electronic components. However, when the large test station ($40ft^3$) itself has a malfunction, the technician is left on her own to resolve the faults.

Sherlock was developed to train the technicians who will have to troubleshoot the test station in the field, far away from help. Sherlock was designed to provide efficient practice with feedback and support which is not possible in its real application context. It made the practice effective by maximizing the time spent on cognitive activity. That is, it simulates time consuming physical actions like waiting for parts, dismantling components etc. It tries to reduce cognitive overload by keeping track of what the technical trainee has already done and reminding them about it. It gives advice on overcoming the knowledge gaps. Since the test station is very complex, it provides context-specific abstracted problem space for each training problem.

Sherlock's curriculum focuses on a) curriculum strategy b)mental models of test configurations that can be created with the test station c) using test instruments to make measurements. It keeps track of students using a competence model and a performance model.

Takeaway: Reducing time-taking routine tasks can be done to reduce cognitive load of the learner. PHyTeR provides hints for some of routine domain tasks (like conducting a particular test), so that the students can focus on troubleshooting part.

2.6.5 The Troubleshooter

Schaafstal and colleagues (Jutten et al., 1999) have developed a tool called the Troubleshooter to improve troubleshooting performance of technicians in the domain of naval frigates. The pedagogical basis of the Troubleshooter is the structured troubleshooting process summarized in section 2.2.2. This process consists of four phases: Formulate problem description, Generate causes, Test, Repair and Evaluate. This tool combines an intelligent tutoring system and a virtual job environment. It guides students based on various parts of the problem solving process by giving task related guidance and domain related guidance. It acts as a formal coach or a facilitator based on the progress made by students in their learning tasks.

Troubleshooter has a multimedia virtual environment where the students move around in a simulated building. They can have conversations with other simulated people in this environment. The students get the opportunity to act as an authentic troubleshooter doing a realistic job with appropriate documentation, tools and telecommunication facilities etc. This is intended to train them in a strict, technical sense and also to give practice on communicating with end-users (Ex: asking right questions).

Troubleshooter has over fifty problems with varying difficulties. It has a form called fault isolation form that has to be filled in during troubleshooting. It helps in tracking the progress of the student and elucidates the knowledge, skill, misconceptions and bug information of the student. It evaluates students based on an overlay model. This overlay model compares an ideal expert solution with that of the student's. Students have the option to choose between a free method and a heavily guided method for solving problems. Students are closely monitored in their use of fault isolation form and feedback is given at the end of every stage of the problem-solving process. An overall feedback is given at the end to evaluate their performance.

Takeaway: Providing different types of domain knowledge as and when required would reduce the load on working memory. In PHyTeR, an explicit task structure will guide the students to be on track and make it easier for them to come back if they lose track because of working memory overload.

2.6.6 Thinking aloud pair problem solving (TAPPS)

This is a pedagogical approach to developed in the context of problem – solving. There are some studies of its usage in the context of troubleshooting. The basic block of this pedagogy (Johnson & Chung, 1999) is metacognition (Hartman, 1998): the argument that active monitoring of cognitive processes in relation to a concrete goal and subsequent regulation of these processes and actions are helpful for learning. Verbalizing their (students') thought process is a way that the researchers are using to make their thinking process explicit for themselves. It requires two students. They assume the role of problem solver and listener and have strict role protocols to be observed while cooperatively solving the problem. The task of the problem solver is to try to solve the problem. While doing so, she has to try to fully verbalize her thought process. The listener develops a deep understanding of every strategy, step and assumptions of the problem solver. The listener also has to make sure that the problem solver is verbalizing her thought process by using prompts.

Research studies that used TAPPS have reported increased success levels in problem solving (Berardi-Coletta et al., 1995; Hogan, 1999) and metacognition (Chi, De Leeuw, Chiu, & Lavancher, 1994; Johnson & Chung, 1999). Berardi-Coletta and colleagues argue that the reason for this increase is due to the prompts given by the listener. They hypothesize that when the listener asks for explanation from the problem-solver, then the problem solver has to shift her focus and reflect on her actions and reasoning. This shift and explicit thinking about the problem-solving process might lead to increased performance by helping the students to be on track.

TAPPS approach improved the performance of students in fault detection and evaluation of hypotheses(Johnson & Chung, 1999). However, in an ABAB study design, even after being introduced to thinking aloud strategy in the first round, the control group students did not employ this in the second round. The authors argue that the students need practice and support to include this strategy in their problem-solving approaches.

Takeaway: This approach is developed to help students collaboratively solve problems. Though students do not collaborate in our context, we recognise that making students reflect on their actions and verbalizing them is useful. We incorporate this in some of the question prompts in result interpretation and reflection activities in PHyTeR.

2.6.7 Worked Examples

This is another pedagogical approach applicable to the whole of problem – solving that has been employed in the context of troubleshooting. Worked examples have been consistently shown to help in the initial stages of a cognitive skill acquisition when compared to solving problems directly (Atkinson, Derry, Renkl, & Wortham, 2000; Sweller, Merrienboer, & Paas, 1998). One explanation for this result is based on the cognitive load theory (Sweller & Chandler, 1991): that the worked examples reduce extraneous cognitive load for a beginner allowing schema construction and automation (Gog, 2006). There are research studies that investigated the effect of various types of worked examples on troubleshooting performance and transfer.

A product-oriented worked example includes the solution for a problem. A process-oriented worked example explains the rationale for the solution along with the solution. In this context, researchers (Gog, 2006; Kalyuga, Ayres, Chandler, & Sweller, 2003) discuss an expertise-reversal effect happening. They note that when students had low domain knowledge and no proper schemata of the domain, the process-oriented worked examples benefitted them in learning. In case of students with a developed schemata or high prior knowledge, the steps in the worked examples were redundant and might have even hampered their learning.

In another study, it was found that studying process-oriented examples in the beginning would lead to higher efficiency as compared to studying product-oriented worked examples. However, once the students had learned the process information, it became redundant and started to hamper learning. In addition, the students who started with product-oriented examples seemed to close the gap with some practice.

Takeaway: Fading of process-oriented scaffolds seems necessary as the students learn the process knowledge. PHyTeR has hints based on worked examples to help them perform the tasks corresponding to a phase of troubleshooting. These hints might have to be faded in subsequent scenarios.

In this section, we described various existing approaches to teach troubleshooting. We note that none of these are developed or tested for network troubleshooting in the Indian context. Some of the previously discussed approaches did not have technological implementations. We want to address this gap by developing a TELE that implements some of these recommendations and by evaluating them.

2.7 Recommendations to develop troubleshooting

In this section, we list out the recommendations given by various researchers to teach troubleshooting. We observe that much of the recommendations from literature are about the ‘content’ to be taught and how to teach that content but less on appropriate technological features for the instruction.

Recommendations from S.D Johnson (Johnson & Chung, 1999)

- Ensure that students acquire a clear conceptual understanding of system structure, function, and operation of the systems they will be working on. Conceptual illustrations such as functional flow diagrams seem to be useful for developing conceptual system understanding.
- Support students develop capability in a wide variety of troubleshooting strategies.
- Explicitly teach students to follow the "generate and test" process of troubleshooting. Besides knowing how to follow the troubleshooting process, they should be aware of what they are doing and why it is important.
- Facilitate extensive shop, laboratory, and field experiences in order to develop the consistent patterns of behaviour that are associated with expert troubleshooters.

- Emphasize the importance of using the senses to obtain initial problem information and have enough practice to become proficient at acquiring information in that manner.
- Facilitate extensive practice to develop good system and circuit tracing skills.

Recommendations from Schaafstal and Colleagues (Schaafstal et al., 2000)

Troubleshooting training should have the following elements:

- A system-independent strategy for troubleshooting that prevents information overload and ensures a consistent approach across systems
- Functional models of particular systems (system specific)
- Underlying domain knowledge of various types (system specific)

They also recommend an early integration of troubleshooting skills with domain knowledge and also giving different context in which troubleshooting can be used. The rationale behind this is that if the troubleshooting strategies are learnt with simple concepts then applying them to a difficult concept would be easier.

Recommendations from Ross and Orr

According to Ross and Orr (Ross & Orr, 2009), one of the main goals of troubleshooting training is to reduce the load on working memory. For this, they suggest giving a visual summary of the troubleshooting process. Another element of troubleshooting training that is emphasized by them is the reflection on part of both instructor and student. This reflection gives more understanding about why some solutions failed and how they can be improved.

We have used some of these recommendations while designing PHyTeR. We have structured the learning environment to make the troubleshooting process explicit. We provide a visual summary of the troubleshooting process at different places in the learning environment. We prescribe learning activities that focus on different types of knowledge and focusing on functional model of the network. We describe these decisions in detail section 5.1.

2.8 Assessment of troubleshooting skills

2.8.1 Existing metrics to assess troubleshooting skills

Different parameters are used to evaluate troubleshooting. At a broader granularity, to evaluate the 'output' of troubleshooting, metrics like time required to troubleshoot, number of tests performed before finding the actual cause are used.

While comparing experts and novices, the following parameters have been used:

- The location of hypothesis: Experts tend to generate more hypotheses near the location of fault whereas the hypotheses generated by novices were spread all over the system. While teaching novices, this might be used as an indication of novices moving towards the right direction, that is, their trial-and-error approach is reduced to give way to a more structured approach.
- Functional reasoning about the system

Schaafstal & colleagues (Schaafstal et al., 2000) evaluated troubleshooting along 3 dimensions:

- Quality of solution: Was the problem located at sufficient granularity? Ex: at component level than at a subsystem level
- Systematicity of reasoning: Did the troubleshooter follow the steps as taught?
- Functional understanding of the system: Did the troubleshooters exclude (during problem space reduction) complete functional blocks, and did they divide functional blocks into sub-blocks?

Woods (Donald R. Woods, 2006) provides a self - reflection questionnaire to assess the troubleshooting skills based on five key elements. The participants have to rate a number between 1-10 based on their perception of their ability to perform the skill. A summary of that is given in below:

- Element 1: Problem-solving skill as applied to troubleshooting: This includes monitoring the process, data handling, collecting, evaluating and drawing conclusions. In addition, it measures creating and working with hypothesis, and decision making.

- Element 2: Experience with process equipment: This includes equipment specific questions related to the domain of chemical engineering.

- Element 3: Knowledge about safety and properties of material on the processes with which one works: This includes knowledge about flammable risks, health risks, explosive risk, and mechanical risk. In addition, this evaluates the knowledge of unique physical and thermal properties of materials used.

- Element 4: "Systems" thinking: This includes understanding of faulty operation of and carryover from/to upstream/downstream equipment, impact of environmental conditions, pressure profile and process control etc.

- Element 5: People skills: This includes communication skills, listening skills, fundamentals of relationships, developing and building trust and building on another's personal preferences.

2.8.2 A note on troubleshooting success in our context

Traditional metrics like time taken to troubleshoot, number of tests performed to find the actual cause are not sufficient. Neither do they help in identifying the difficulties of a student nor do they help in measuring their improvement at a minute level.

When students are learning such a complex ability like troubleshooting, it is necessary that their microscopic progress be precisely identified and given feedback on. We base the assessment of troubleshooting in our context on the phases of troubleshooting. The details of the assessment metrics are given in Appendix I.

2.9 Summary and emerging research questions

Based on the review of the literature described in this chapter we can summarise the following:

- Troubleshooting is an expected skill for CS graduates. (Section 1.2.1)
- Troubleshooting is complex with multiple dimensions like knowledge, reasoning and skills. Because of this complexity, it is difficult for students to pick up efficient troubleshooting by themselves. Therefore, it is necessary to provide suitably designed learning opportunities.
- A technology enhanced learning environment is one way of providing the learning opportunity.
- Troubleshooting is taught in different contexts using different approaches. However, there is no TELE to teach computer network troubleshooting skills.
- The pedagogy for troubleshooting skills should include structured practice, scaffolds to help students perform and reflect on the process of troubleshooting.

From the broad research goal described in chapter 1 and the literature reviewed in the current chapter, we derive the following research questions:

- How do computer science engineers understand and characterize troubleshooting?
- How to include troubleshooting in the CS curriculum?
- How to include troubleshooting in the Computer Networking curriculum?
- How to develop network troubleshooting skills using a TELE?
- What difficulties do students face during troubleshooting?
- How do the students use the scaffolds during troubleshooting?
- How do students learn troubleshooting skills?

Among the above research questions, we chose to start with understanding the difficulties of the students. Then based on this understanding and from literature in other domains, we want to create a TELE to develop troubleshooting skills. Using this TELE, we tried to investigate how students use the scaffolds during troubleshooting. The next chapter outlines the research methodology applied in this thesis with specific research questions answered.

Chapter 3

Research Methodology: DBR

3.1 Research Objective

Our broad research goal is to develop a TELE (PHyTeR) to teach network troubleshooting skills to CS undergraduates. Developing an intervention to teach a complex problem like troubleshooting includes intricate interactions between the learner, the troubleshooting problem (in a network simulator) and PHyTeR. We want to understand how students approach solving a network troubleshooting problem and the difficulties they face. We then want to develop a technology enhanced learning environment that gives them appropriate scaffolds. For these research goals, we need to answer questions like the following:

1. What difficulties do students face while doing network troubleshooting?
2. What should be the pedagogy of the PHyTeR?
3. What features should PHyTeR contain?

4. Do students learn troubleshooting skills after interacting with PHyTeR?
5. How do students use PHyTeR to understand and learn troubleshooting skills?
6. Is PHyTeR useful to students?
7. Is PHyTeR usable by students?

This implies that we might have to consider both qualitative and quantitative methods to answer our research questions. In addition, because we are trying to understand how students use PHyTeR while learning troubleshooting, this requires some theory building. That is, we wanted to develop an intervention for a complex problem and understand how the intervention helps the target audience. Considering all these factors, we chose DBR as the research methodology.

3.2 Design Based Research

Design based research (DBR) or Educational Design Research (EDeR) is a methodology employed in investigating complex, practical problems by developing solutions iteratively. These solutions can be educational policies, products, processes or programs. Bereiter (Bereiter, 2002) indicates that design research is not defined by its methods but by the goals of those who pursue it. Richter et al (Richter & Allert, 2017) point out that in design-based research neither the problem nor the possible solutions are given but are actually created in the process of design.

The Design-based Research collective, as described by Randolph (Randolph, 2009), summarizes the critical questions generally answered by design-based research as follows:

- Research questions that investigate and produce theories or 'proto-theories' of learning
- Research questions that explore the interplay between an authentic setting and an intervention
- Research questions that examine how an intervention leads to the desired goals
- Development questions that investigate how to improve an intervention

DBR (McKenney & Reeves, 2013) focuses on two things – i) solving problems with the participation of all stakeholders and ii) generating new knowledge/process about solving that particular problem which can be reused by others solving similar problems. A solution is developed iteratively during multiple design cycles. DBR has an overarching goal and goals for each design cycle. DBR starts with analysis of the problem and the problem-solving context, then one starts with the design of the solution, develops a prototype and evaluates it. The output of this evaluation is not just whether the solution worked or not but also learning (theory) about the solution development process. The researchers move back and forth between the phases and iterate the cycles with increasing implementation and spread. Bannan-Ritland (Bannan-Ritland, 2003) notes that the category of the research question (as listed above) directs the specific actions/output in the phases of DBR.

We are following the DBR methodology as described in McKenney & Reeves (McKenney & Reeves, 2014). The following diagram describes phases:

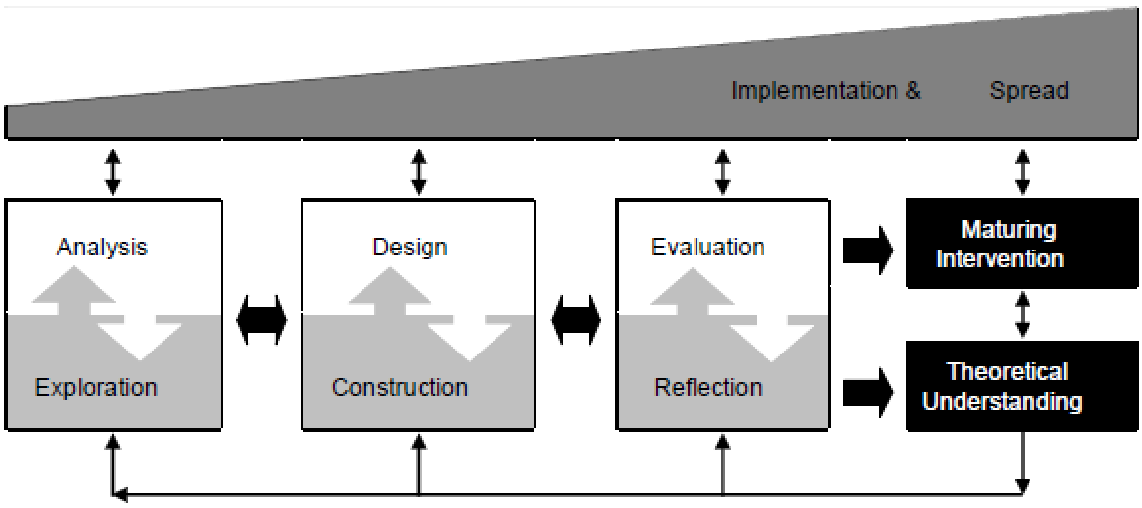


Figure 7: Overview of EDR process (McKenney & Reeves, 2012)

In this thesis, the overarching goal is developing a TELE to teach network troubleshooting skills to Computer Science undergraduates. We have implemented two iterations of DBR cycles. The work done corresponding to phases in each iteration is described in the next section.

3.3 Two DBR cycles of this thesis

The goal for the first cycle was to characterize network troubleshooting skills and to identify and implement key pieces of the intended TELE. We reviewed literature on troubleshooting in different domains, the nature of expertise in troubleshooting, difference between experts and novices in troubleshooting, teaching-learning of Computer Networks, designing learning environments etc. We expanded our search to problems that were similar to troubleshooting like ill-structured problem solving and science inquiry domains. This enabled us not only to get an overview of how these are taught to different populations but also the pedagogical basis of the interventions. Based on these investigations, we gathered some features of TELEs that were shown to help in student learning in those contexts. The particular questions that guided this process were:

1. How is troubleshooting defined/described?
2. How is it related to problem-solving?
3. How do experts troubleshoot/ solve similar problems?
4. How are troubleshooting skills taught in other contexts?
5. How is network troubleshooting taught?
6. What pedagogies are used to teach troubleshooting skills/ problem – solving?

In study 1, we then identified the difficulties faced by students while trying to solve a network troubleshooting problem. We gave the students a faulty network in a network simulator. We observed and supported them while they were trying to troubleshoot the network. We then analysed the observations and audio transcription of the researcher- student interaction to deduce themes of difficulties. The literature review and study 1 constitute problem analysis phase of DBR cycle 1.

Based on the literature and results of study 1, we designed learning goals for the students in our context. We identified the pedagogy and specific features of PHyTeR. We implemented some of these features in the first version of PHyTeR. This forms the design and development phase of DBR cycle 1. With respect to our overall goal, we wanted to see if the students are able to learn troubleshooting skills in PHyTeR v1. We evaluated PHyTeR v1 in study 2 where 21 students participated. We collected data to answer the following research questions:

- After working with PHyTeR v1, what are the perceptions of the students about troubleshooting skills?
- What are the perceptions of the student about PHyTeR v1?

Based on the results of this study, we identified

- Scaffolds that were helpful in resolving student difficulties
- Features that were leading to desired learning outcomes
- Features that would need some modifications either because they were not leading to desired learning outcomes or because they were difficult to use.

In study 2, we found that the students were unable to comprehend how one learning task in PHyTeR connects to another task. Few of them had difficulties in applying given procedures. The set of learning outcomes that were not achieved through PHyTeR v1 along with the features that were difficult to use in PHyTeR v1 lead us to the goal of the second cycle of DBR. A summary of DBR cycle 1 is described in Table 3.

In the second iteration, we added new features to achieve desired learning outcomes and modified some features to enhance usability of PHyTeR. We reviewed literature to identify existing strategies/features to overcome the difficulties. We took help of a UI/UX designer to enhance the usability of PHyTeR and to present a coherent learning environment. In the second version of PHyTeR, we implemented the new features and changes suggested by the designer. We conducted two studies to answer the following research questions in the context of PHyTeR v2:

- After interacting with PHyTeR, what is the students' understanding of troubleshooting skills?
- How do students use the features of PHyTeR during troubleshooting?

We conducted a detailed qualitative study (Study3) of student interaction with PHyTeR v2. Five students participated in this study. We explored how students use the affordances of PHyTeR during troubleshooting. In study4, twenty students used PHyTeR to solve a network troubleshooting problems. We gave them questionnaires before and after the intervention. This questionnaire was designed to elicit their approach towards troubleshooting. We analyse the questionnaire responses to triangulate our findings from study3. Finally, we report the reflections after conducting these iterations with respect to our broad research goal. The second column in Table 3 summarizes the activities performed in second cycle of DBR.

Table 3: Summary of DBR cycles

DBR Cycles/Phases	Cycle 1	Cycle 2
Problem Analysis	• We characterized troubleshooting including process and actions involved via literature review • Study 1 (N = 5) - We identified the difficulties faced by students during troubleshooting • We constructed the learning objectives for the students	• We analysed the results of study 2 with respect to the overall learning goal. • We recognised the places where students are facing difficulty to achieve the learning objectives. • We identified scaffolds to help students achieve the learning objectives
Design & Development	Based on the problem analysis, we identified the pedagogy, created learning activities and developed first version of PHyTeR	Based on the newly identified scaffolds and design issues identified in Study 2, we developed the second version of PHyTeR
Solution	PHyTeR v1	PHyTeR v2
Evaluation	Study 2 (N = 21): It was conducted to evaluate the effect of PHyTeR v1 on students' understanding of troubleshooting skills. Important result: Students were able to recognize the importance of problem space understanding but had difficulties in connecting it to hypothesis testing cycle.	• Study 3 (N = 5): We conducted this to evaluate the effect of PHyTeR v2 on students' understanding of troubleshooting skills. We identified how the students use the features in PHyTeR v2 during troubleshooting. • Study 4 (N = 20): We conducted this study to triangulate some of

		the findings of study 3.
Reflection	We identified the concerns along the dimensions of doing the tasks, learning and usability of PHyTeR. Summary: Integrate Problem Space Understanding activities with the troubleshooting tasks, more support to do some phase specific tasks, to identify the phases of troubleshooting and reflection activities are needed	We construe local learning theories on how students use the features in PHyTeR during troubleshooting.

3.4 Summary

In this chapter, we discussed the research methodology of design-based research and why it is appropriate for our research goal. We also gave a summary of the two iterations of DBR performed as a part of this thesis. In the next chapter, we start with the detailed description of problem analysis phase of DBR cycle 1.

Chapter 4

DBR Cycle 1: Problem Analysis

In this chapter, we describe problem analysis phase of DBR cycle 1. In problem analysis phase, we surveyed the literature based on the research goal. We have summarized this in detail in Chapter 2. We conducted a study to identify the difficulties faced by students during network troubleshooting skills. We describe the study in detail in this chapter. We start with summarizing what we already know about the context and then describe study 1.

4.1 What we already know about the context

In previous chapter, we analysed the literature related to troubleshooting in other domains. From this analysis, we synthesized the troubleshooting process as applicable in the context of computer networks. Based on this, we developed the learning goals for the students trying to learn troubleshooting skills. The following table describes those learning goals:

Table 4: List of learning goals corresponding to each phase of troubleshooting

Phase	**Learning Goal** **After doing the learning activities corresponding to this phase, the students will be able to,**
Problem Space Understanding	Identify the relevant behavioural, function & structural elements at different levels and inter-relationships between them Distinguish faulty behaviour from the ideal behaviour
Hypothesis Generation	Identify a device/part of network to be behaving erroneously
Hypothesis Testing	Construct a test that is appropriate for the hypothesis selected Predict the result at different levels Conduct the test in a simulator and observe the result
Result Interpretation	Concluding the hypothesis testing Drawing inferences related to the broader problem
Overall process	Discern and follow a structured process of troubleshooting Apply troubleshooting strategies to narrow down the problem space

In addition to understanding of the process of troubleshooting, we have analysed literature related to teaching of troubleshooting skills. Based on this, we recognise the following about our context:

- The reported research studies were with technicians or technicians in training. In other words, the job emphasizes on troubleshooting. This is in contrast with our context where troubleshooting is not emphasized as a critical component.

- Many difficulties that the novices encounter are because of cognitive overload. Cognitive overload results in them losing track of the process, difficulty to interpret information in a comprehensive manner. These difficulties are reported in contexts like electronic and mechanical systems. We recognised that the difficulties identified in other context are insufficient to build a TELE in our context. This is because PHyTeR would need to understand user actions at a fine level of detail and give feedback on that. We want to identify the specific impediments of students during network troubleshooting. We want to identify the type of guidance that will lead students towards the solution.

This analysis leads us to design and conduct a study to understand the student difficulties in detail. We describe the study in detail in the next sections of this chapter.

4.2 Study 1 - Understanding student difficulties

The goal of this study was to understand the difficulties faced by the students while they try to troubleshoot a faulty network. We assume that the students are familiar with the domain concepts required and a familiarity with the network simulator. We recognise that the difficulties might be due to insufficient domain knowledge, lack of experience in using the network simulator and lack of troubleshooting experience. Our priority is to simplify the later - the process of troubleshooting. Therefore, we do not focus much on the difficulties that were inferred as due to lack of domain knowledge and tool usage.

4.2.1 Research Method

4.2.1.1 Research Design

The design was an exploratory problem solving session. We conducted this study in our research lab. We used a network simulator called Cisco Packet Tracer to host the faulty network. The students had used this simulator in two of their lab sessions and had basic familiarity of the simulator. The simulator had the ability to support all the interactions required to troubleshoot the network. We considered the following research question to guide the research design:

RQ 1: What difficulties do students have while solving a troubleshooting problem in data communication networks?

4.2.1.2 Population and Sampling

The students who participated in the study were third year undergraduate (20-21 years old) students of Computer Science and Engineering department in a tier-3 engineering college in Mumbai, India. The course "Computer Networks" or "Data Communication Networks" is generally taught in the second or third year of CS engineering curriculum. This includes both theory and practical sessions. The students work with some simulator during the practical sessions.

The teacher who taught them computer networks announced in their class about the study. The interested students filled registration form to participate in the study. Five students (4 male and 1 female) volunteered to participate in the study. They had completed "data communication networks" course in their previous semester (both theory and practical). The study was structured as a workshop to learn network troubleshooting skills. After they completed the study, they received a certificate of participation in the workshop from the research lab. Apart from this, the students participating in the study received no other incentives.

4.2.1.3 Study Procedure

Every student worked individually, with two researchers, my colleague and me. Figure 8 shows the sequence of activities done by students during the study. Each session started with researchers introducing themselves, the work we do in the research lab and an overview of the study cum workshop. We gave consent form to the students with a brief explanation of what it constitutes. This took around 10 minutes. We allowed the students to leave the study at any point in time.

We asked them to watch a video about the features of the network simulator. This was meant to refresh their experience of using the simulator. The video was around 5 minutes length. The students were allowed to pause, rewind and replay the video. This activity took about 15 minutes.

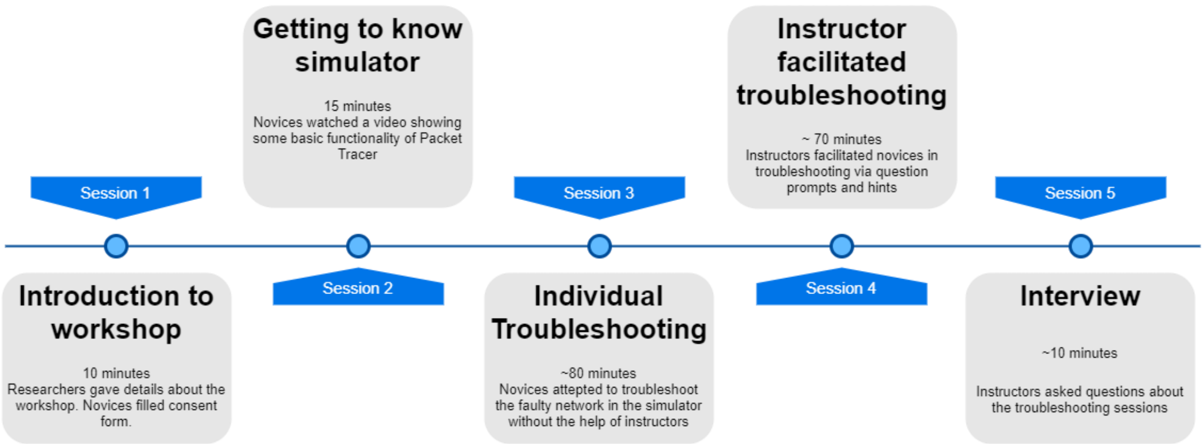


Figure 8: Sequence of activities during the study 1

After this, I introduced the troubleshooting problem by showing a demo of the faulty behaviour of the network in the simulator. The faulty behaviour was that a browser in a computer in the simulator gave ‘host name unresolved’ error. I asked them to resolve the faulty behaviour. They were allowed to use the internet while doing so. I instructed them not to delete any device or connection from the faulty network since such operations are costly/last resort even in real networks. One of the researchers observed the student while he/she was solving the problem and noted down those observations. The students were allowed to talk to the researchers if they encountered any doubt or difficulty. If the question was about logistical issues (Ex: one session had issues with internet connectivity, a student required headphones to watch videos, a student wanted to know where the play button was located in simulator etc.) they were resolved immediately. However, if the questions were conceptual or related to troubleshooting, they were encouraged to look for solutions themselves during individual problem solving activity. The minimal interference of researchers during this activity was intended to observe their natural problem-solving process and to allow them to immerse into the problem.

The researcher-aided problem solving started after allowing students to individually tackle the problem for about 80 minutes. The researchers intervened after observing that the student is stuck or after the student himself/herself asked for help. In the first case they started by asking 'Do you need any help?' In the latter case, they asked 'Can you explain what you were trying to do?' The researchers tried to guide their problem solving process by question prompts and explanations that often used some analogies. Below is a representational example of the conversation between researchers and students.

Researcher: Do you want any help?
Student: Yes. I think the link here (showing a link) is up but ping is not working.
R: Okay. What is needed for a ping to work?
S: Link
R: And?
S: And… (silence)
R: What do you mean by a link? What constitutes it?
S: Connection between both (devices)
R: How is a connection between two devices setup? What is required here for example?
S: For example, see this (points at laptop1) is connected to switch with IP address 20.0.0.3 (points at the switch). (//Misconception: Thinks that a level 2 switch has an IP address)
R: Does switch have an IP address of 20.0.0.3?
S: Yes. When I pinged from laptop1 to switch it worked. But now when I am trying to ping from laptop1 to router is not working.
R: Can you show me the IP address of the switch? Where is it assigned?
S: (searches for a while) I'm not able to find.
R: What is a switch? How does it work?
S: It broadcasts messages within a network.

R: Right. Within a network. Why can't it broadcast outside the network?

S: (Thinks for a while) I don't know.

R: Okay, This switch here is a level 2 device. It is connected to devices only within this network using MAC address. It does not have an IP address.

S: Okay.

R: With this understanding, can you explain to me what happened when you used this ping option?

As we can note from the above example, if the student was stuck with a misconception or incorrect domain knowledge, researchers resolved it and directed them towards the problem solving process. If the student faced difficulties related to the process of problem solving, we prompted them with questions like, 'why are you searching for this information?', 'Now that you have observed this result, what do you want to do next?' Table 5 lists the prompts given by researchers during the study. These question prompts were based on the conceptual framework proposed by Xun and Land (Xun & Land, 2004) and reported student difficulties in literature.

Table 5: Examples of prompts used by researchers during the study

Prompt to	**Example prompts**
Help them mentally simulate	Can you explain to me the working of a router?
Help them think about the network without error (ideal network)	Can you explain to me how a website request works in an ideal case?
Recognize a device or component as faulty (generate hypothesis)	Now that you have understood how ping works, can you tell me where the error is?
Conduct a test	How will you ensure that the link is working?
Interpret a result	What do you think 'host name unresolved' means here?
Get students back on track	Why were you searching for this information? What were you trying to do? Based on this, what do you want to do next?

After helping the student to solve the problem in this manner, researchers had a concluding session with each student. In this session, they gave an overview of troubleshooting and four phases involved with examples from their problem solving session. Each student took an average of 2 hours before solving the problem (in case of four students) or quitting (in case of one student).

4.2.1.4 Network Simulator and the troubleshooting problem

We used Cisco Packet Tracer v7.2.1 during this study to host the faulty network. In this simulator, it is possible to execute most of the commands used in real network devices, configure devices by changing settings, set-up, and access services like HTTP, DNS, DHCP, etc. It has the ability to add/remove various types of devices and links. It has a real-time mode and a simulator mode. In real-time mode, the devices behave like in a live network. In the later mode, one can see the working of a network in a step-by-step manner by using play and pause controls. Here one can see every packet that gets transmitted between the devices. One can click on those packets to open them and see structural information like the source, destination IP addresses and other header information. The packet also displays behavioural information on how the packet gets processed at each layer of the network in a device. Along with the simulator, we allowed students to use the internet and search for anything while solving. We allowed them to install any software if needed. In their registration form, all students had mentioned familiarity with Cisco Packet Tracer. However, during the study they reported that it is overwhelming/daunting to use. They informed the researchers that previously, they had not used Cisco Packet Tracer to solve these kinds of problems.

The topology of the faulty network is shown in Figure 9. We gave the students the following problem-statement printed on a paper - ‘Meenu is trying to connect to www.et.co.in from PC0. However, she is not able to access it. Can you resolve her problem?’ They had to identify that the problem was due to the server port being turned off in the webserver.

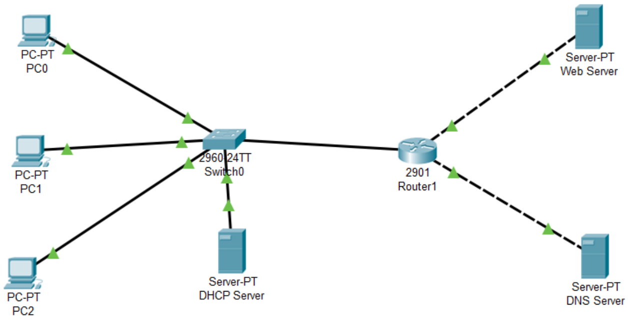


Figure 9: The faulty network topology that the students had to troubleshoot in study 1

4.2.1.5 Data Collection

Figure 10: The study setup showing a student working with the simulator

Each student worked with a laptop that had the faulty network residing in the network simulator. The laptop had internet connectivity. We did the screen recording for the complete duration of troubleshooting. We also audio recorded the conversation between the students and the researchers. We asked the students to use a paper and pen if they want to note down anything during troubleshooting. We also recorded the video of the problem solving session as shown in Figure 10. We used it mainly to synchronize between audio and screen recording and to identify when the student was writing something on the paper.

4.2.2 Data Analysis

4.2.2.1 Data Analysis Framework

In this section, we describe the data analysis procedure in detail. The basis for our analysis is the process of troubleshooting as described in section 2.4. We intend to categorize the difficulties along each phase of troubleshooting. We have learning goals corresponding to each of these phases as described in section 4.1. We want to identify the difficulties faced to achieve these learning goals.

Based on those goals we identified the representative example actions that would help in achieving the goals. We asked an expert in network troubleshooting to evaluate the mapping between goals and phases based on the following questions: 1) Do you agree with the goals of each phase of troubleshooting for the given context of Computer Networking? If no, please explain why. 2) Are there any salient goals missing for each phase? 3) Are the actions appropriate for the goals to be achieved by students? These evaluated goals and corresponding example actions are listed in Table 6.

Table 6: Mapping between phases and goals used in analysis

Phases	**Goals**	**Example actions that leads to achieving the goal**

Problem Space Understanding	Identifying the relevant behavioural, functional & structural elements at different levels and inter-relationships between them	• Runs the simulator to understand how a device/part of the network works • Collects information about a device/part of the network by using search engines, manuals, etc. • Draws a diagram to understand the working of whole/part of network • Understanding the behaviour of a protocol ex. TCP/UDP
	Distinguishing faulty behaviour from the ideal behaviour	• Compares two diagrams / notes / video / simulation to find differences in structure/function/behaviour of the corresponding networks • Changes a configuration and observes changes to identify the exact faulty behaviour/structure/function
Hypothesis Generation	Identifying a device/part of network to be behaving erroneously	• Compares the simulation with expected behaviour in any form (video, diagram, description, etc.) – Notes down the exact erroneous behaviour • Identifies a device/part of the network to start testing
	Relates the hypothesis to the broader problem	• Notes down other possible hypotheses • Notes down other paths to take if the current hypothesis doesn't reveal the error • Notes down justifications for selection of a hypothesis

Hypothesis Testing	Constructing a test that is appropriate for the hypothesis selected	• Collects information on how to tweak a device/part of the network via video/manual etc. • Records where to look for outputs • Identifies dependencies related to the changes being made (Ex: If I change the IP address in the router, IP addresses of connected devices have to be changed) • Identifies ways to check only the hypothesis and nothing else
	Predicting at different levels	• Notes down what output to expect (ex: I will get a response for the ping in … form) • Records the expected function/behaviour (Connectivity between router and server be restored after this change)
	Conducting the test	• Executes the planned changes in the simulator (Checks/ changes a configuration, check connectivity etc) • Observes output
Result Interpretation	Concluding the hypothesis testing	• Looks for and notes down relevant output and finds the meaning of relevant error codes & or description • Interprets the output observed in terms of hypothesis – whether the hypothesis is true or false • Examines if any changes made to the network needs to reverted
	Drawing inferences related to the broader problem	• Identifies gaps in understanding of working of network • Verifies assumptions made • Identifies the next step/path to take based on the conclusion & justification

4.2.2.2 Data Preparation

We transcribed the audio conversations between researcher and the student. We annotated the screen recordings using the software ELAN (Wittenburg, Brugman, Russel, Klassmann, & Sloetjes, 2006). We used the audio transcription as a primary source to recreate ‘episodes’ of problem solving by a student. Coding for screen recording was done and used only where more clarification was needed for the observation and audio transcript data.

4.2.2.3 Data Analysis Process

We used screen recording + transcript of audio conversation + researcher observations to generate a workflow of the problem solving process by the student. We performed deductive thematic analysis (Braun & Clarke, 2012) of the actions performed by students to solve the problem. We divided the transcription into ‘episodes’ of troubleshooting by combining a group of statements. We made detailed notes about an episode and generated initial codes from these by trying to understand the actions of students in context of our analysis framework. In order to make comparisons between codes easier, device specific details like server, port status, IP address were abstracted to device independent things like ‘a device’, ‘configuration settings’ etc. when necessary. Table 7 and the following paragraph describe an example of the same.

Table 7: An example of analysis

Start time (m:s.ms)	End time (m:s.ms)	Actions
02:05.7	02:34.7	Reads the slide with the description 'Click on the interface through which router0 is connected with router 1. The slide also has a screenshot of the interface configurations in the simulator.
02:36.9	02:43.9	Reads the next slide which says 'Insert the IP and check port status'. The screenshot now has the port status turned on.
02:44.5	02:45.7	Goes back to the previous slide.
02:46.7	03:05.2	Opens the simulator window. Clicks on router-interface connected to server and sees if the port status is on or not.
03:07.5	03:18.4	Increases the size of the router window. Closes it

03:19.3	03:39.4	Clicks on the switch. Clicks on settings option. Clicks on the interface option. Checks if port status is on or not.
03:40.2	03:43.4	Toggles the bandwidth option in the switch

Note: We considered previous and subsequent contexts while making detailed notes and initial codes from the episodes. The following paragraphs summarise the creation of themes.

We did not expect students to exactly follow the sub-processes involved in troubleshooting as we had listed. While we did not expect them to generate a hypothesis in a required form, we assume that a student would have his/her own versions of what might be a ‘proto-hypothesis’. We classified something as a difficulty when a student found it hard to utilize that proto-hypothesis in the process of troubleshooting. For example, a student had doubts that laptop1 might be faulty. However, while testing, instead of testing laptop1 separately, she was testing the whole network. It did not allow her to conclude if laptop1 is faulty or not. In that case, we classified her actions as ‘Difficulty in grounding tests on a hypothesis’. We describe the themes that emerged for each phase in the next section.

Detailed Notes for the above set of actions: Here the student was trying to understand the working of the router. On finding the first configuration (IP and port status) to be set, opens the simulator and starts checking those configurations in every device. The student does not continue this comparison for every setting however. While looking for port status of switch in simulator, he finds another configuration (Bandwidth). He selects different options for bandwidth.

Initial Codes for the above: i) Compares at a settings level, does not relate that setting to the behaviour of the device ii) Does not follow a strategy to its logical end iii) Tries to make multiple independent changes in many devices before testing.

Final Themes: i) Difficulty in identifying structural components required to perform a function. ii) Difficulty in keeping track of a troubleshooting strategy iii) Difficulty in basing tests on a hypothesis

One researcher generated the themes for one transcript and I generated theme for another transcript. We discussed and combined the themes. I used this set of combined themes to code other three transcripts. We grouped the themes according to the learning goals for each phase of troubleshooting. We present the complete list of difficulties for each learning goal in the next section.

4.2.3 Results

There were around 60 instances of troubleshooting in between 5 students. We analysed these instances to synthesize the difficulties. In this section, we summarize the themes corresponding to the learning goals in each phase of troubleshooting with an example. In the end, we list the difficulties corresponding to overall process of troubleshooting.

Problem Space Understanding

While trying to understand the problem space, none of the students followed standard practices like noting down the initial configurations and trying to reproduce the erroneous behaviour. They directly jumped into ‘troubleshooting’ mode that is, changing configurations and checking if the error is resolved. A common approach was to set up a new network from scratch. This is also indicated by their initial search phrases in Google (How to setup a router, how to connect laptop to router in Cisco Packet Tracer etc.) and then wanting to delete a few devices and add new ones. During instructor facilitated troubleshooting, we got an opportunity to understand students' understanding of the problem space. Students answered the researcher prompted questions like ‘How does a router work?’ in a ‘textbook-ish’ manner, that is, they were able to tell the working of a router as a standard description. When we asked questions to connect that description to the router to the given faulty network, they were not able to apply their understanding in the given context.

Table 8 has the themes related to difficulties in problem space understanding. We observed that students did not spend much time trying to understand the problem during individual troubleshooting. All themes related to problem space understanding have been generated from analysis of instructor facilitated troubleshooting.

Table 8: Themes on difficulties related to problem space understanding

Phase: Problem Space Understanding		
Goals	**Difficulty in pursuing goals**	**Examples**
Identifying the exact faulty behaviour of the system / Distinguishing faulty behaviour from the ideal behaviour	Difficulty in identifying structural components required to perform a function	Difficulty in identifying that correct IP addresses, default gateway and link/port status are needed for packet transfer between a laptop and a router
	Difficulty in identifying how functions of 2 devices contribute to the behaviour of the network	In a DHCP setting, difficulty to identify how a laptop gets its configuration settings and how it leads the laptop to get connected to the network
	Difficulty in translating known conceptual knowledge to practical knowledge	Knows that a router is used to connect 2 networks. But is not aware that a router uses 2 IP addresses (belonging to 2 different networks) in 2 different ports to connect 2 networks
	Difficult to attribute a function/behaviour to one device	Confusion on how a device gets its IP address - from the default gateway or DHCP server.
	Difficulty to distinguish between function/behaviour of 2 similar devices	Difficulty to distinguish between how a router and switch differ while both of them are used to connect various devices

Hypothesis Generation

The initial hypothesis generated by students was usually based on what they ‘thought’ was wrong and not on any observation of the given context. For example, one student said ‘*the router may be faulty because it has so many configurations. Something might not be configured properly’*.

The proto-hypotheses, if we can call them so, that were generated by students were primitive in the sense that they did not think of them as a testable cause for the error but they had an idea that there might be something wrong with a device. All the hypotheses generated were at device level or about the connection between two devices. They were mostly vague (not testable) and rarely were they generated based on observations made by the student. We observed that students often dismissed these proto-hypotheses without proper testing. Because of this, they still 'thought' that there is something wrong with a device but were not able to explain what exactly is wrong. They were not able to explain what about a device was working as expected and what was not working as expected. This 'thought' persisted and influenced their further actions – they found it hard to think of another device than the initial one they had suspected. In other words, none of them treated hypothesis as tentative and generated another hypothesis. Most of them were stuck with one hypothesis that they generated at the beginning. Table 9 lists the themes of difficulties faced by students related to hypothesis generation.

Table 9: Themes on difficulties related to hypothesis generation

Phase: Hypothesis Generation		
Goals	**Difficulty in pursuing goals**	**Examples**
Identifying a device/part of network to be behaving erroneously / Relates the hypothesis to the broader problem	Difficulty in generating a testable hypothesis	The hypothesis was 'there is some fault in laptop1' but had no idea what exactly the fault is or how to test if there is a fault with laptop1
	Difficulty to acknowledge that a hypothesis is tentative	Difficulty in discarding a hypothesis 'There is some fault in the IP address of router', even after verifying the hypothesis to be false
	Difficulty to generate justifications for selecting a hypothesis	Thinks that the fault might be in laptop1. However, does not have any support to make that statement. Is not following the strategy of "I'll test every device one by one"

	Difficulty to distinguish between actions to understand the problem (What if I make this change to a device) and hypothesis	The student wanted to see what happens when a setting is changed (not to see if the problem gets solved but to understand the working of a device). But after changing the setting, checks to see if the problem gets solved.
	Difficulty to generate hypothesis based on his/her understanding of the problem state	The student is able to explain the behaviour of the device (That some device is not working as expected). However, is not able to convert that understanding in a hypothesis form.

Hypothesis Testing

Themes related to difficulties faced by students while hypothesis testing are described in Table 10. While testing, we noted that after changing a configuration of a device they mostly tested for the broad problem instead of the hypothesis selected. For example, one student thought router was not connected to the laptop. Then after assigning the router's IP address as the laptop's IP address (which is what he thought of connecting two devices), instead of checking if the laptop is connected to the router he checked if he can access the website from the laptop (which was the original problem). We observed them doing the same test multiple times in a hope to observe different results! They found it hard to keep track of the testing process too – they failed to adjust the dependencies based on the configuration they changed. For example, one student changed the IP address of a router but forgot to change the corresponding default gateway address in the devices connected to that router. He also forgot to undo the changes made when he thought that the test was not useful in revealing new things.

The episodes where they lost track of troubleshooting process were more during the testing phase because we observed that i) sometimes when they saw a configuration setting (unrelated to current testing), they went ahead and changed that. This affected the current test and made it more difficult for them to interpret results. ii) sometimes they changed a setting at device level and tested if the original problem has been solved. This also resulted in having done a test without proper observations and conclusions. Thus, it became more difficult for them to interpret results based on a test conducted.

Table 10: Themes on difficulties related to hypothesis testing

Phase: Hypothesis Testing		
Goals	**Difficulty in pursuing goals**	**Examples**
Constructing a test appropriate for the hypothesis selected	Difficulty in identifying dependencies of a configuration change	The student wants to change the IP address of the router. She does not realize that correspondingly default gateway address has to be changed in the laptops connected to router
	Difficulty in recognizing that only hypothesis has to be tested and not the broader problem	Changes the IP address of laptop1. Ideally has to check if laptop1 is connected to router (that was the hypothesis) but checks if the website is accessible (the broader error).
	Difficulty in searching for information needed to design a test	When they want to know about a configuration, they search for general information related to the device.
	Difficulty in grounding tests on a hypothesis	The student wanted to change a configuration and see output. But she did not have any justification for changing that configuration.

Predicting at different levels	Difficulty to expect the outputs based on the changes made	The student changed DNS configuration and was observing packets in simulation mode. One type of packet was moving between devices and another type of packets was not moving. Student did not check the type of packets but incorrectly concluded that the packets are moving between devices.
Conducting the test	Difficulty in finding the required testing tools in a simulator	The student wanted to test the DNS server address in a laptop. But she spent a lot of time in searching for the option and getting confirmed that she has selected the correct option.

Result Interpretation

Table 11 has the themes related to difficulties in result interpretation. Some students, after interpreting that a particular device was working properly based on a test, were unable to conclude if they can abandon that device from further investigation or not. For example, a student tested the connectivity of the router and arrived at a conclusion that the connectivity is intact. However, she failed to consider that in the given context, connectivity of the router was the only relevant part about it. She continued 'thinking' there might be something else wrong with it. This is related to the justification for the hypothesis she gave: 'there are so many configurations here, something might be wrong'. Most of the time, students interpreted the result only in the context of the test and did not consider the implications of the result on hypothesis and the problem space unless prompted by the researcher. For example, a student identified that the packets were being dropped between the router and the server. However, she failed to attribute that behaviour to either router or server and thus identify that either router or the server as faulty. Sometimes, students missed to read error messages and came to different conclusions of the performed test. For example, when the error message was 'unable to start a service', a student failed to read it and interpreted it as 'unable to connect'.

Table 11: Themes on difficulties related to result interpretation

Phase: Result Interpretation		
Goals	**Difficulty in pursuing goals**	**Examples**
Concluding the hypothesis testing	Difficulty in interpreting error messages	A student did not read the error message completely but assumed the meaning of the error code.
	Difficulty in deciding whether a device/part of network is fault-free	A student checked that the connectivity in router is intact. However, he could not arrive at the conclusion that the desired function of the router is connectivity and the router need not be tested further. He continued to think that there might be something else faulty with the router.
	Difficulty in deciding next steps based on observation in context of the hypothesis	A student observed that the connectivity between laptop and router was not restored after making changes. She could not decide if she has to do another test or if the observed result was expected/ conclusive.
Drawing inferences related to the broader problem	Difficulty to switch between broader problem level and granular hypothesis testing level	A student concluded that there is no error in router. However, she could not decide her next steps based on that conclusion.

Overall troubleshooting process

Apart from the difficulties related to each of these phases of troubleshooting, we derived themes related to the overall process of troubleshooting. These themes are listed in Table 12.

Table 12: Themes on difficulties related to overall troubleshooting process

Overall process of troubleshooting		
Goals	**Difficulties**	**Example**
Discern and follow a structured process of troubleshooting	Difficulty to keep track of the troubleshooting process	Student started with the aim to find the network connectivity between each pair of device. She was trying to find how to test the connection status in a switch. When she found an unfamiliar configuration setting (that was not related to connectivity), she started exploring about that configuration and forgot about checking connection status in a switch and did not complete testing connectivity between each pair.
Apply troubleshooting strategies to narrow problem space	Difficult to follow a strategy consistently to its logical end	A student started with the strategy of analysing every configuration in every device (exhaustive search). However, after checking one configuration in laptop, he started checking configurations in router.

4.2.4 Discussion and Conclusion

In this study, we identified a set of difficulties that students face while network troubleshooting. We did not aim to be comprehensive in understanding the difficulties but to gain insights into the critical difficulties faced by students. We hypothesize that PHyTeR should have activities corresponding to each phase of troubleshooting. The themes evolved would help us in designing scaffolds for these activities. For example, we identified that students have difficulty in distinguishing function/behaviour of two similar devices. We have explanatory videos, comparison activities with specific feedback in PHyTeR to address this difficulty.

We recognised that the troubleshooting problem that was given to the students in study1 was found to be very difficult for them. In addition to the complexity of the simulator, the complexity of the network also added to their difficulty. The given network used the following concepts: Routing, Switching, DHCP service, DNS service, HTTP service. We decided to have a simpler network constituting few topics for the further studies. This is because we do not want the lack of domain understanding to impede their troubleshooting performance. We decided to discard all but the following concepts: Routing, DNS service and HTTP service.

In this chapter, we described study1 that was conducted to identify student difficulties. We designed scaffolds in PHyTeR to overcome these difficulties. We describe the design and development of PHyTeR including the learning activities and scaffolds in the next chapter.

Chapter 5

DBR Cycle 1: Development of Solution

In previous chapters, we explored the definition of troubleshooting, actions that helps a troubleshooter in resolving the fault, literature on teaching troubleshooting skills and the difficulties faced by students to do those actions. In this chapter, we describe the design and development of PHyTeR based on this understanding.

In section 2.5.3, we described the rationale for choosing inquiry learning as the pedagogy. Inquiry learning is grounded on the doctrine that students reach an understanding of a phenomenon by self-directed investigations. It is shaped by research in the fields of scientific reasoning, scaffolding and development of software based learning environments (Lazonder, 2014). It possesses a two-fold nature of “inquiry as ends” and “inquiry as means” which depend on each other (van Joolingen & Zacharia, 2009). De Jong (De Jong, 2006) classifies the processes involved in inquiry as follows: orientation, hypothesis generation, experimentation, and drawing conclusion. We want to emphasize that these processes are similar to the ones described earlier (section 2.4) in the context of troubleshooting. In the next sections of this chapter, we describe the adaptation of this approach in the context of troubleshooting.

5.1 PHyTeR Pedagogy

Our approach is to first introduce the four phases of troubleshooting. Then, provide them opportunities to practice the activities corresponding to these phases. Once they get used to ‘doing’ these activities, we introduce reflection activities allowing them to contemplate on the need for the activities. To design these learning activities, we considered the learning goals, the recommendations from literature about similar learning goals, and difficulties faced by the students while accomplishing these learning goals. We describe these learning activities in this section. Figure 11 depicts the key pieces in PHyTeR pedagogy.

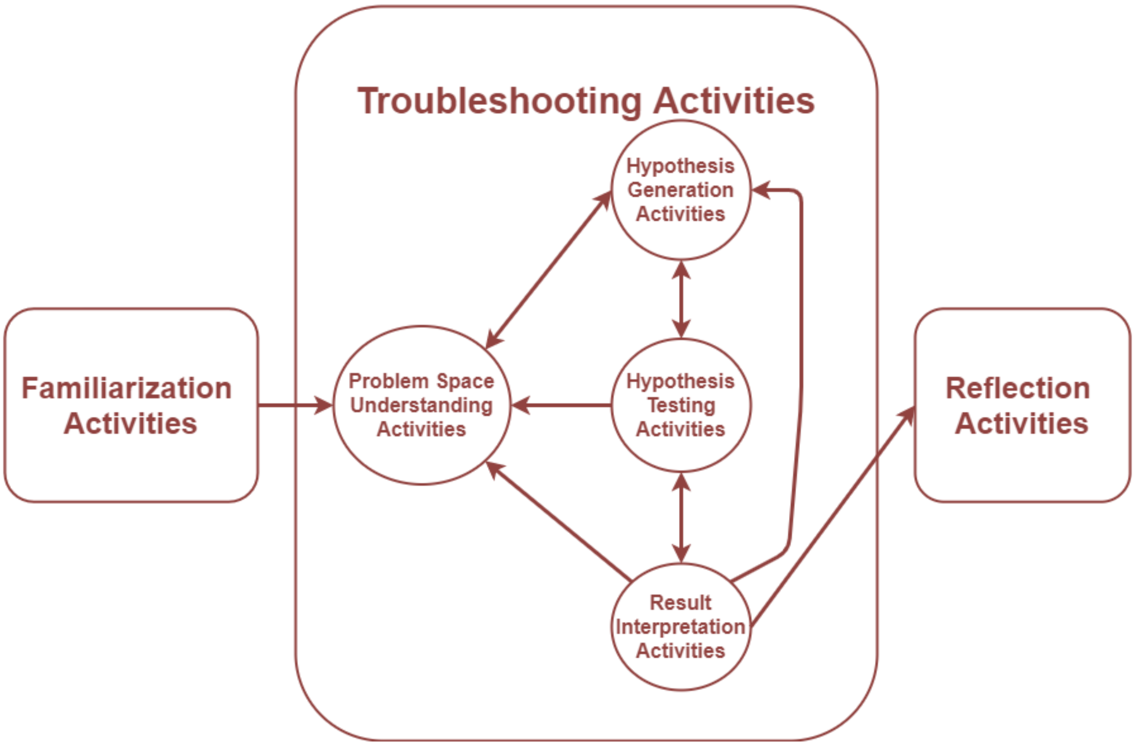


Figure 11: Activities in PHyTeR Pedagogy

Students start their interaction with PHyTeR with the familiarization activities. These activities introduce the students to the context of network troubleshooting. They describe the process of troubleshooting and give an overview of activities corresponding to each phase. In addition, they emphasize the importance of troubleshooting knowledge for a CS undergraduate. This helps the students to prepare for the subsequent tasks. In current implementation, the familiarization activity is manifested as three videos including animations and screen recordings. This activity is not tied to any particular troubleshooting scenario. PHyTeR presents this activity before the students start troubleshooting a scenario. Next, we describe the scenario specific features in PHyTeR.

Scenario description

PHyTeR recommends a clear detailed description of the faulty scenario. This can be a textual description or a video of the faulty network in the simulator or a combination of both. It should enable students to clearly recognise the fault. It should be accessible for the students throughout the troubleshooting process.

Problem Space Understanding activities

The goal of problem space understanding phase is to make students identify the structure, function and behaviour of the network and its devices at various levels. In addition, they should be able to compare and contrast the ideal behaviour of the network with the faulty one. In study 1, we found that students have difficulty to understand the problem space because of insufficient domain knowledge. The students have difficulty in identifying the structural components required to perform a function, identifying how functions of two devices contribute to the behaviour of the system. They find it hard to translate known theoretical concepts to practical knowledge. They find it confusing to distinguish between function and behaviour of two similar/ related devices. Researchers have emphasized on getting students to have structural, functional and operational understanding of the system to support troubleshooting in this phase (Johnson & Chung, 1999; Schaafstal et al., 2000).

To overcome such difficulties, Reid and colleagues (Reid, Zhang, & Chen, 2003) suggest to make students explicitly identify information at different levels and providing scaffolds for interpreting the working of a system. Quintana and colleagues (Quintana et al., 2004) recommend to focus student attention on important tasks. Sharma and Hannafin (Sharma & Hannafin, 2007) suggest providing opportunities to clarify and externalize the students' misconceptions. Based on these recommendations and the learning goals for PSU, PHyTeR pedagogy has the following:

- A structured space to do Problem Space Understanding (PSU) activities
- Activity to elicit the relationship between devices
- Activity to recognise relevant structural component of a device
- Activity to link the structure to function and function to behaviour
- Activity to recognise and compare ideal and faulty behaviours
- Activity to elicit and clarify student misconceptions

Hypothesis Generation

The goal of hypothesis generation phase is to make students generate testable hypothesis by making them identify a device or part of the network behaving erroneously. We noticed that the students have difficulty in forming a testable hypothesis and recognizing that it is tentative. They do not base their hypothesis on the observations of the faulty behaviour. For these difficulties, literature suggests to have external representations that a student can interact with (Kirsh, 2004, 2010), give useful structure to restrict the complex task (Quintana et al., 2004), to provide process prompts to support task achievement (Sharma & Hannafin, 2007), and to embed expert guidance (Quintana et al., 2004). Based on the learning goals and the recommendations, PHyTeR pedagogy provides the following activities in hypothesis generation phase:

- A structured space to do hypothesis generation activities
- Focussed task to select/ generate hypothesis that are testable
- Scaffolds to help students connect the hypothesis to problem space understanding, testing and result interpretation tasks
- Expert insights to convey how to generate hypothesis and example hypotheses

Hypothesis Testing

The goal of hypothesis testing phase is to design and conduct tests that are aligned to the hypothesis selected. We identified that students have difficulty in aligning the test with the selected hypothesis. They do not keep track of the changes made to the network while testing, making it difficult for them to revert these changes. They have difficulty in searching for and obtaining the exact information needed to conduct a specific test. We identified that it is hard for them to predict the output of a test. It is difficult for them to find the suitable testing tools in a simulator.

We referred literature to find recommendations for similar learning goals and difficulties. Researchers suggest supporting the student actions by setting useful boundaries (Quintana et al., 2004), providing process prompts for task achievement (Sharma & Hannafin, 2007), providing external representations to depend on (Kirsh, 2010) and embedding expert guidance and insights (Quintana et al., 2004). Considering the learning goals and the recommendations, the pedagogy of PHyTeR has the following:

- Separate tasks for designing test, predicting result and conducting test
- Scaffolds to elicit and represent test and prediction
- Explicit prompt to evaluate the alignment of the test with hypothesis
- Expert insights and useful heuristics related to testing
- Scaffolds to help them with conducting the test in a simulator

Result Interpretation

The goal of result interpretation is twofold. The first goal is to conclude the hypothesis using the test result. The second one is to interpret the hypothesis acceptance/rejection in the context of the bigger problem. In result interpretation phase, students find it difficult to conclusively decide if a device is faulty/fault free, based on the result of testing. They find it hard to decide the next step to be taken based on the hypothesis testing. Literature suggests the learning environments to have divide the complex task (Quintana et al., 2004) and provide structural support (Sharma & Hannafin, 2007). Reid and colleagues (Reid et al., 2003) suggest to have activities making students explicitly identify different levels of information. Considering the above-mentioned learning goals and recommendations, the PHyTeR pedagogy has the following:

- Separate tasks for concluding the hypothesis and connecting it to the broader problem
- Prompts to connect the result with testing and hypothesis generation phases
- Prompt to consider the broad picture while deciding the next step
- Scaffolding to help students focus on the task

Overall troubleshooting process

The goal here is to support students in following the process of troubleshooting process. We found that the students have difficulty in following one strategy to its logical end. They have difficulty to keep track of the troubleshooting process while performing the tasks corresponding to phases of troubleshooting. For similar difficulties, literature suggests providing visualization of process steps and progress (Quintana et al., 2004), familiarizing them to the overall process, authentic problems which demand application of process steps and expert insights (M. C. Kim & Hannafin, 2011). Researchers have recommended providing explicit practice to follow the troubleshooting process and providing visual summary of the process to reduce the cognitive load (Johnson & Chung, 1999; Ross & Orr, 2009).

Considering these recommendations, PHyTeR pedagogy has the following:

- Dividing the learning environment to consistent spaces for each type of activity

- Visual affordances representing the process steps and progress
- Summary of the overall process
- Prompts to connect different activities with respect to the overall process
- Reflection activity to relate the phase activities with the overall process

In the next section, we describe the manifestation of this pedagogy as the learning environment PHyTeR.

5.2 PHyTeR v1 – The learning environment

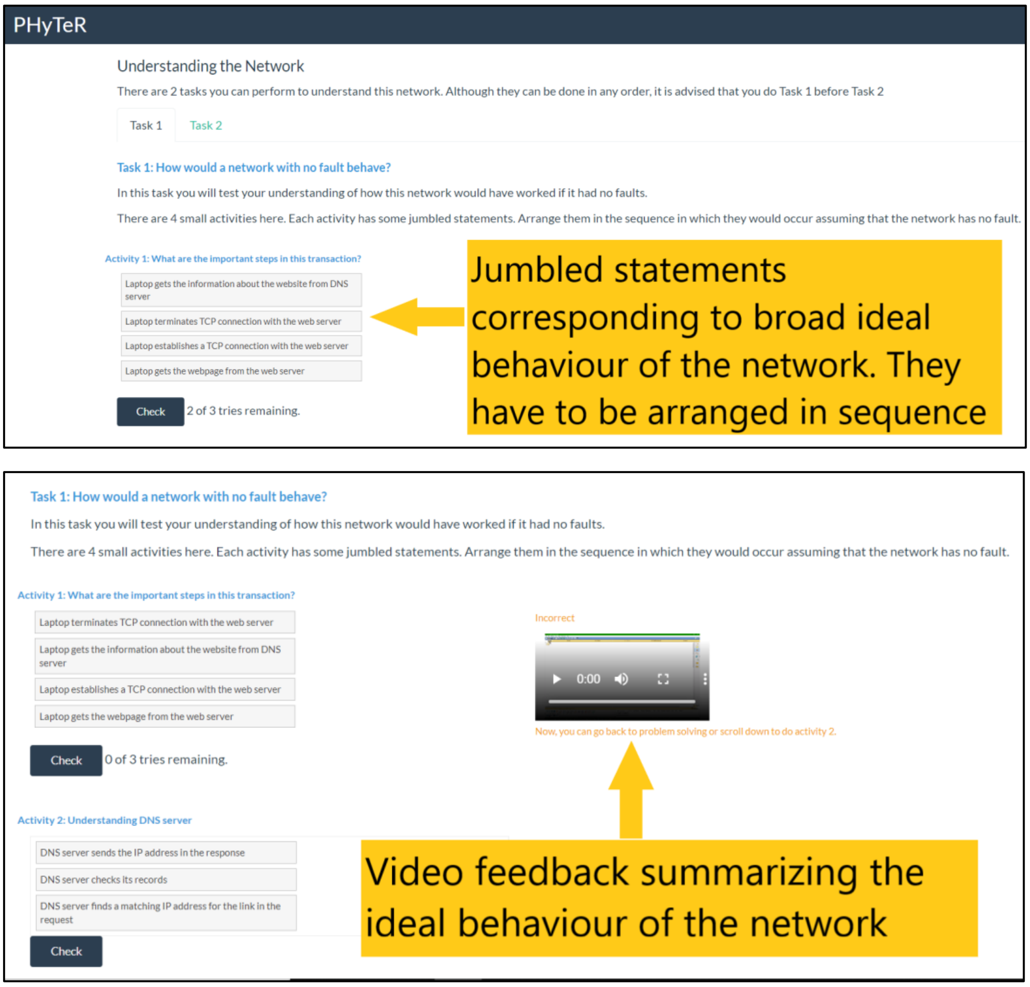
The word PHyTeR is derived from the phases of troubleshooting: Problem space understanding, Hypothesis generation, hypothesis Testing and Result interpretation. It is a web-based learning environment built using Node.js, React, Express framework, MongoDB. More details about PHyTeR can be accessed here. In this section, we describe how the PHyTeR pedagogy is implemented in PHyTeR learning environment.

5.2.1 Problem Space Understanding

The aim of these activities is to make students understand the network in terms of its behaviour, structure and function. There are two sets - one set to understand the network as a whole and a set each for understanding every device in the network. The activities are optional and students can do it whenever they want.

Understanding the network

We start with an activity that checks their understanding of the broad ideal behaviour of the network. We do this by asking them to arrange a set of behaviours in the correct order. They can drag and arrange the set of behaviours in the correct order and check their answers. A textual/visual feedback is given according to their answers, which guides them towards the solution. If they fail to provide the correct answer within three attempts, the correct answer is shown with a video of the correct behaviour. Once they complete the activity corresponding to broad level behaviour, more similar activities corresponding to detailed behaviour are given. Figure 12 shows the screenshots of task 1 in PHyTeR.



a)
b)

Figure 12: Screenshots of task 1 in understanding network

The second task in understanding network focuses on making students identify the differences between ideal and faulty networks. A list of statements is presented to the students. There are two bins corresponding to ideal and faulty network. The student has to drag each statement and drop it in the correct bin. There is a check button along with each bin. The student can click on it to verify if the statements dropped in a bin actually correspond to that bin or not. While doing this activity, we hypothesize that students will try to examine the working of faulty network in detail. Figure 13 depicts the screenshots corresponding to this activity.

a)

b)

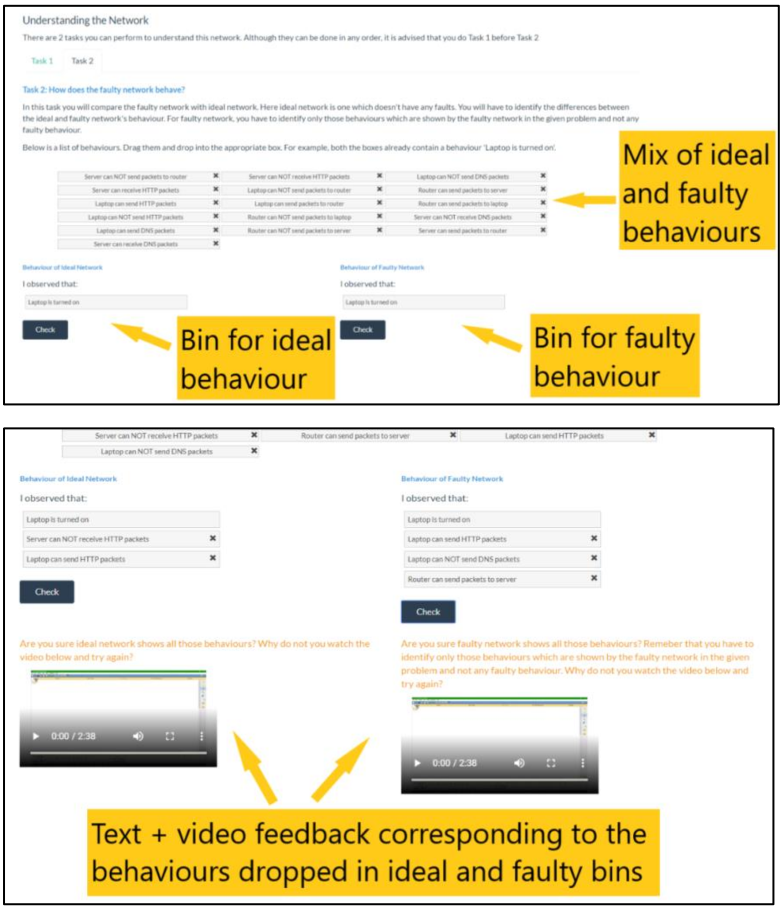


Figure 13: Screenshots of task 2 in understanding network activity

Understanding a device:

This set has two activities. The first activity is to understand the broad level behaviour of the device in the context of the troubleshooting scenario. In this activity, a set of jumbled statements are given and the students have to arrange these in the correct order of occurrence. The second activity is a set of multiple choice questions or true/false questions designed to bring out some misconceptions of the students. PHyTeR provides visual and textual feedback to reinforce the ideal behaviour of the device. Figure 14 shows a screenshot of this activity.

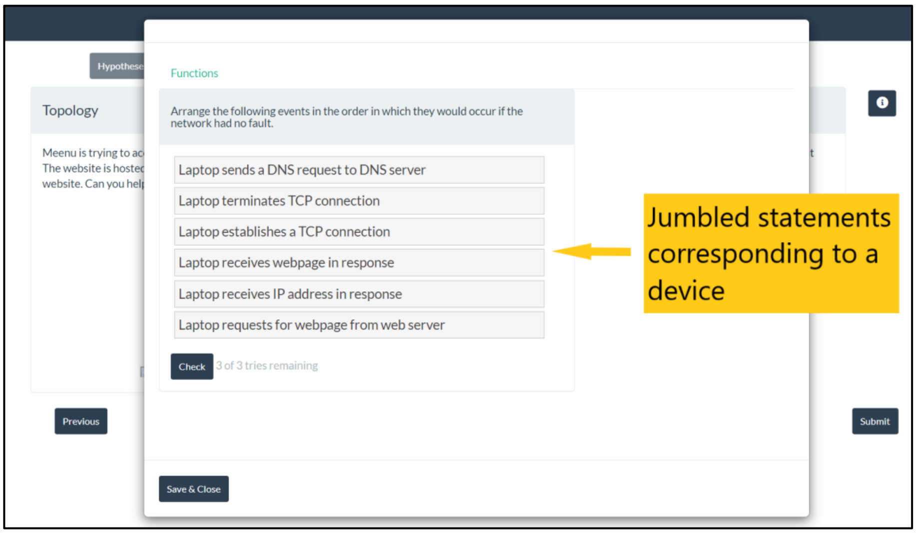


Figure 14: Screenshot of understanding a device activity

5.2.2 Hypothesis Generation Activities

In this activity, a list of possible hypotheses is presented to the students. These hypotheses are at network device level. The students have to select the device and a component within the device that they think is faulty. PHyTeR displays this selection in all further tasks that they do. After they select the hypothesis, they have to justify their selection. They can select these justifications from a given list. An initial version of PHyTeR allowed students to create new hypothesis but students found it confusing. Therefore, we decided to provide a structure to hypothesis creation task in this version. Figure 15 shows the screenshots of these activities.

a)

b)

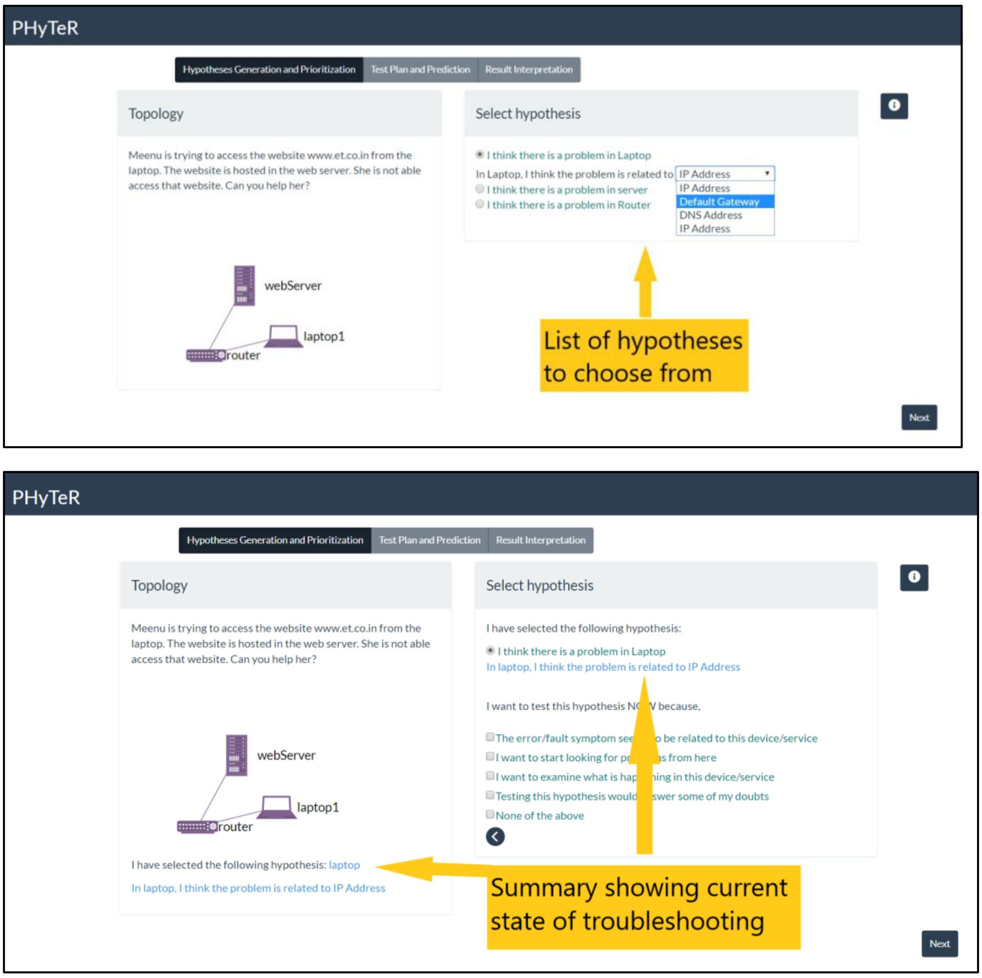
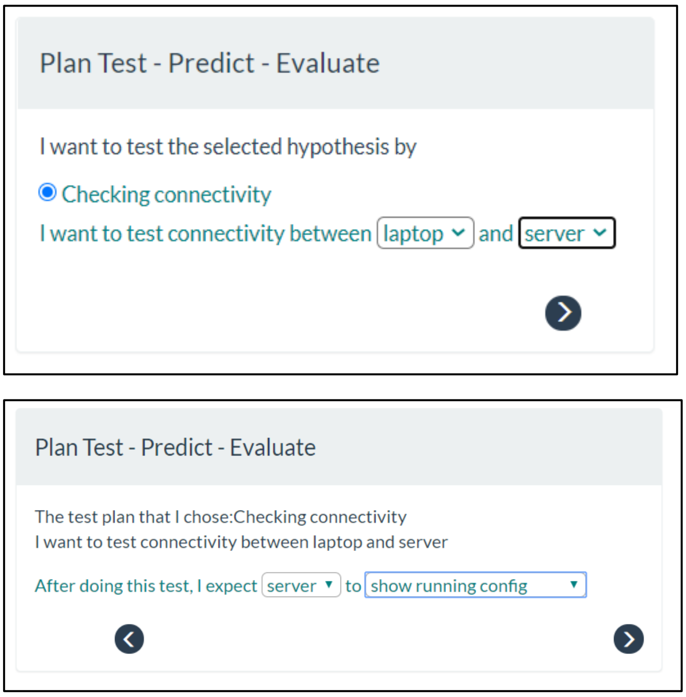


Figure 15: Screenshots showing a) hypothesis generation b) justification selection activities

5.2.3 Hypothesis Testing Activities

After they have selected a hypothesis, PHyTeR asks them to select a test corresponding to that hypothesis. Again, we start from a broad level and then go into the details of a chosen test. It displays the selected hypothesis and presents three test options to select from: checking the connectivity between two devices, checking or changing a configuration setting in a device, and executing a command in a device. Once they select one of these options, they can select details corresponding to that option. For example, if they select the test option of checking the connectivity between two devices, the details include selecting the two devices.



a)

b)

Figure 16: Screenshot of hypothesis testing activities: a) selecting the test b) Predicting the result

After they have designed the test including the required amount of detail, they are asked to predict the result of the test performed in the faulty network. Then they are asked to explicitly compare the selected test with the hypothesis chosen. This is a prompt for them to understand that a test needs to be specific for the hypothesis chosen. Then PHyTeR displays the complete test plan created by the students. It instructs them to carry out the test in the network simulator, observe the results and come back to PHyTeR.

5.2.4 Result Interpretation Activities

Once they have conducted the test in the network simulator, they switch to PHyTeR interface and report their observed results. PHyTeR prompts reflection question: Have you expected this result? This is intended to make them compare their predicted results with the observed ones. Then the student has to decide if a device can be classified as faulty/not faulty based on the hypothesis testing conducted. This activity is included to introduce the notion of 'reducing the search space' with each iteration of hypothesis testing. In the end, there is a prompt asking them to select the faulty device and component. If they identify the actual fault, that ends the troubleshooting scenario in PHyTeR. If they cannot identify the actual fault yet, they choose the next step. PHyTeR provides prompts to select the next step based on the conclusion from result interpretation activity. Figure 17 shows the screenshot of this result interpretation activity.

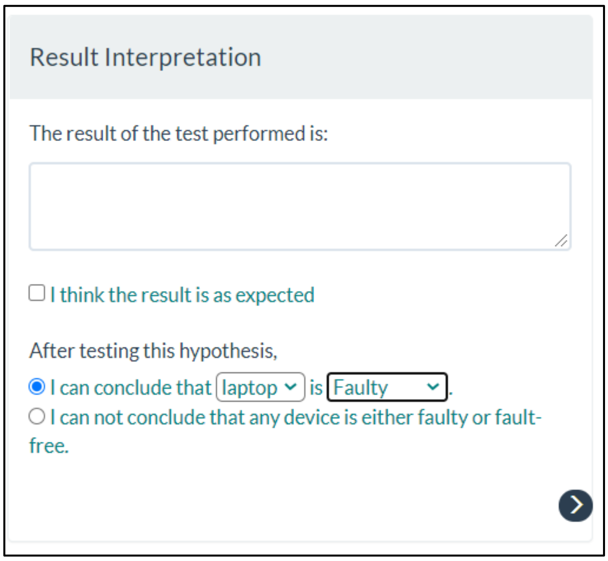


Figure 17: Screenshot of a part of result interpretation activity

In addition to activities corresponding to each phase, PHyTeR v1 has two separate pages to provide hints and to provide required information. We describe these two features in the next sub-sections.

5.2.5 Hints

There are hints to support hypothesis generation and hypothesis testing activities. The hints given during hypothesis generation were question prompts. For example: Think of the ideal behaviour. Where do you think the network is deviating from the ideal behaviour? After multiple hypothesis testing iterations, if they failed to select the correct device, these hints nudged them to look at the device with error.

The hints given during the hypothesis-testing phase were more direct. These hints showed short video clips of doing a particular test in packet tracer. Figure 18 shows the PHyTeR interface for watching video hints corresponding to hypothesis testing.

Figure 18: Screenshot of an example video hint based on the device and component selected

5.2.6 Local Wiki - Information Centre

PHyTeR has a local wiki available to students. This contains information about various networking concepts like networking devices, layers in a computer network, concepts of routing, switching etc. It has another section related to troubleshooting concepts. This explains the phases of troubleshooting in detail and contains expert insights and heuristics related to the phases. Figure 19 shows the screenshot of the local wiki.

Figure 19: Screenshot of the local wiki present in PHyTeR v1

5.3 Summary

In this chapter, we described the evolution of the solution. We described the broad pedagogy and its theoretical basis. We also explained one instance of implementation of the pedagogy in term of the TELE PHyTeR. We want to evaluate if the pedagogy and the manifestation of the pedagogy enables students in troubleshooting a faulty network scenario. We conducted a study to evaluate this version of PHyTeR. We describe this study in the next chapter.

Chapter 6

DBR Cycle 1: Evaluation

In this chapter, we elaborate study2 that was conducted to evaluate the design and implementation of PHyTeR v1. We end this chapter with the reflection phase of DBR cycle 1.

6.1 Research Questions

We designed this study to investigate the following research questions:

RQ 2.1: After working with PHyTeR v1, what are the perceptions of the students about troubleshooting skills?
RQ 3.1: What are the perceptions of the student about PHyTeR v1?

6.2 Methods and materials

6.2.1 Population and Sampling

There were 19 (15 male and 4 female) students who participated in this study. It was conducted in the laboratory of their college. All of them were studying in third of Computer Science Engineering. They had studied Computer Networking course in their previous semester and had familiarity with packet tracer. Their class teacher had sent a registration form for this activity and they volunteered to participate. They received a certificate for participating in this study.

6.2.2 Study Procedure

We conducted this study in a workshop mode in a laboratory in the students' college. After introducing the goals of the workshop, we (I and a colleague of mine) asked the students to watch a small video on how to use the simulator. Then we gave them a faulty network to troubleshoot in the network simulator. We fixed a time limit of 30 minutes to solve the problem. This was to familiarize them with the simulator and the context of troubleshooting. After 30 minutes, I called six of them, individually, and conducted interviews to understand their perceptions of troubleshooting skills.

Then we introduced them to PHyTeR using a video. We gave another faulty network in the simulator and asked them to use PHyTeR along with the simulator to solve the troubleshooting scenario. After 30 minutes into the second problem, I interviewed the same six students on their approach to troubleshooting and features of PHyTeR that were either useful or difficult. The students took around 2 hours to solve the problem. After the problem solving session, we interviewed the same six students to understand their perceptions of troubleshooting skills and PHyTeR. This was a focus group interview.

6.2.3 Data Collection

The students were working on lab computers. We recorded their screen while they were interacting with PHyTeR and the simulator. The interviews were one-on-one with the researcher during the study and as a focus group after the study. We recorded the audio during these interviews. The interview questions asked before intervention are:

1. What do you understand by troubleshooting?
2. What are you doing to solve this problem?

The interview questions asked during the intervention are:

1. Can you explain what you were trying to do?
2. How are you trying to solve the problem?

The questions asked in the post intervention focus group interview are:

1. What is troubleshooting according to you?
2. How will you solve a similar network-troubleshooting problem?
3. What features of PHyTeR were useful or difficult for you?
4. What features of PHyTeR were difficult or confusing for you?

Two researchers made observation notes during the study. They observed three students each. In addition, the students filled out a survey after the study regarding the usability of PHyTeR.

6.3 Data Analysis

We transcribed the audio recordings. We derived the themes (Braun & Clarke, 2012) using inductive thematic analysis. We considered the transcripts of three students to generate inductive themes. These initial codes were used to code the other three transcripts. We added the additional codes that emerged during this to the original list. We generated the final list of themes by comparing the final codes again with the transcripts.

Now we describe the process of thematic analysis for these questions with an example. Here is excerpts from one transcript:

Researcher: Since you mentioned hypothesis I'm going to ask another question. What is a hypothesis?
Student: Hypothesis is… it is basically an idea, an idea to get to the solution.
R: So how do you go about that idea? What would happen if you have that idea?
S: If I have that idea then I can work further on that idea to resolve the system
R: Okay, How?
S: If I have an idea about something, then I'll go in that direction only.
R: Okay, Can you give concrete example?
S: Here I selected router, then I'll go and search only router

This excerpt was annotated with the following initial codes:

- Hypothesis is an idea
- Hypothesis gives a sense of direction

After annotating 3 transcripts at this detail, I read through all the annotation and clubbed the similar ones. Here is an example of one such grouping related to hypothesis:

- Hypothesis is an idea
- Hypothesis gives a sense of direction
- Hypothesis helps to break down the problem
- Hypothesis gives a checklist
- Hypothesis gives options to choose from

I compared this list with the annotated transcripts again. I clubbed related transcripts and removed redundant ones. The resulting list for hypothesis is given below:

- Hypothesis gives a sense of direction
- Hypothesis helps to break down the problem
- Hypothesis gives a checklist to choose from

I used the list of themes generated at the end of this process to code remaining transcripts. We describe the themes generated in the next section. Before describing the results, we describe a representative workflow of a student in study 2.

6.4 Workflow

In this section, we present the workflow of student S1 as he progressed through the session. The session started with me explaining them the concept of troubleshooting and the phases involved in troubleshooting. After this, S1 logged into PHyTeR and started solving the scenario.

He read the scenario question, and entered the PHyTeR main screen. He selected a hypothesis, provided a justification for the selection. After this, he selected a test from the list provided by PHyTeR. After this, it prompted him to predict the result of the test by selecting one possible result from given drop down lists. Subsequently, PHyTeR provided a link to the hints page. The student clicked on this to open the hints page. He read the textual hints related to hypothesis generation activity. He did not do any action based on this. After this, he selected the device and component for the test hint activity. The corresponding test hint video was displayed. He watched the video. He selected a different combination of devices and components and watched the corresponding test videos. This selection of device & components and watching of videos continued until he found a video whose actions he wanted to emulate in packet tracer.

He opened the faulty network in packet tracer and tried to repeat the actions shown in the video. The video showed how to check and change HTTP settings in the server. Changing these settings in the simulator did not cause the error to be resolved. S1 switched between hint videos and packet tracer without understanding why that test was needed or how to interpret the result of the test. I prompted S1 to go to PHyTeR main screen and connect the result of the test to hypothesis by answering the questions in the result interpretation step. When he opened the PHyTeR main screen to do the result interpretation tasks, he was confused. Because he had done multiple rounds of testing, it was difficult for him to interpret the results of those tests in the context of the selected hypothesis. I had to give multiple prompts to support him connect the selected hypothesis to the performed tests to the result interpretation task. He ended with identifying that there should be some problem with laptop. This is not the correct diagnosis of the problem.

6.5 Results

6.5.1 After working with PHyTeR v1, what are the perceptions of the students about troubleshooting skills?

To answer this research question, we consider the student responses of interviews conducted during and after intervention. We elaborate on the themes generated in the following section. A summary of these themes is shown in Figure 20. For most of the students, this was their first time trying to solve an authentic troubleshooting problem. Though students had familiarity with building network topologies in packet tracer, they had not performed any troubleshooting. Two of them said that they had to do troubleshooting during building network topology, but it was not “*this type of troubleshooting*”.

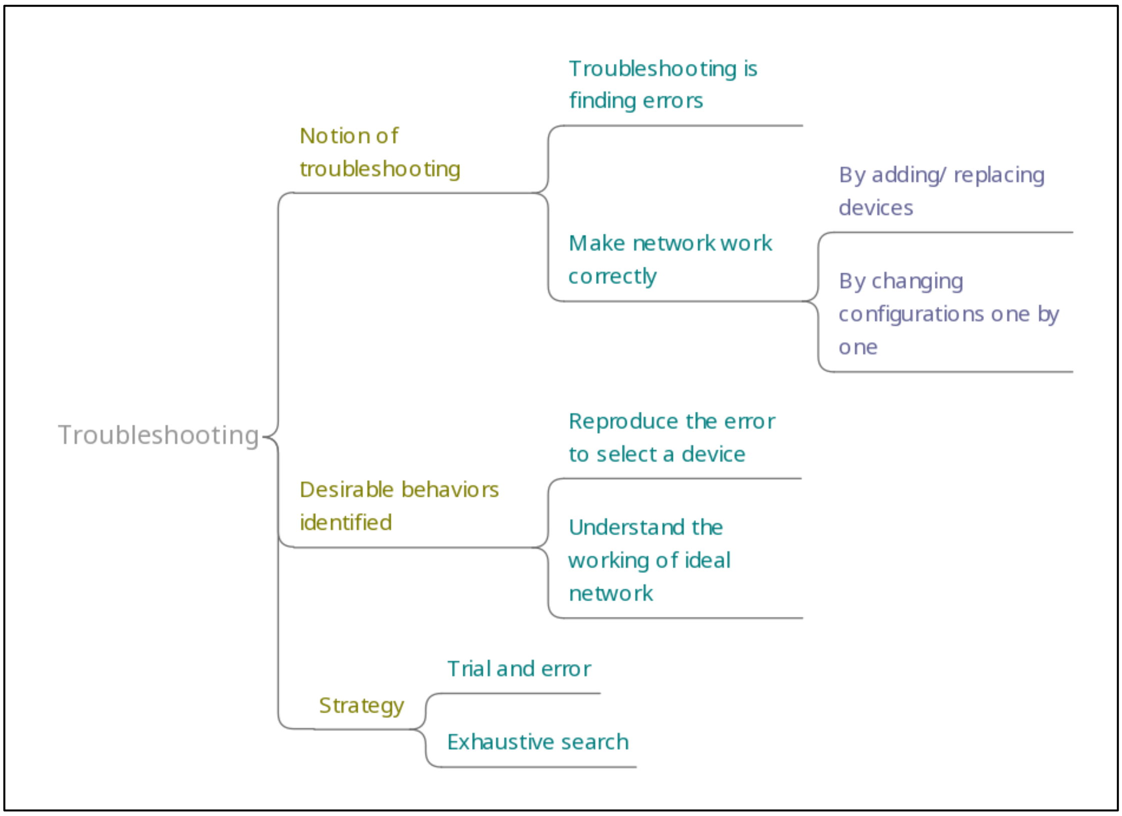


Figure 20: Summary of themes related to student conception of troubleshooting in study 1

Their conception of troubleshooting was largely positioned around 'repairing'. Everyone recognised that troubleshooting is done when there is a fault and the goal is to identify the fault and rectify it. S3 put it as "*... so that we can identify what the problem is, where it is...*" However, few of them were able to describe a structured process of how one goes about doing troubleshooting. Their approach was based on trial and error methods like changing configurations in every device. They described the process of troubleshooting as "modifying the network", "changing configurations": *"I'll modify.. like these devices till I get the error resolved", "According to me it's like - sometimes in troubleshooting we don't configure like IPs or destinations. So I have tried that",* "*so…in case when the network is down, we have to get it running again so network troubleshooting is looking at the errors and getting a solution for the problem*"

We found that the students were able to identify some desired behaviours like understanding the working of ideal network: "*I would first see this video* [the video showing ideal behaviour] *and then see what to do* ", "*I'll first do this activity* [task 1 in understanding the network] *it will give me the steps*". Three students said they would check where the error is before starting to troubleshoot: "*I'll send packets and see where it fails*". However, the students still had some naive conceptions like once a device is identified, changing and testing every configuration detail is needed: "*After that I'll see* [at] *which device is packet failing, then I'll see and change the configurations*".

Some of their actions were based on shallow understandings of theoretical concepts. When asked about which device you will check first – Three of the students said router but they were not able to provide any suitable justification for selecting router: "*I think router has an issue*", "*There are like different types of router, so maybe I'll add another type of router and see if this work*", "*Router has so many things, we can modify so many things, I'll change them and see*".

Similarly, the students identified the usefulness of test hint videos. However, they ended up performing some tests just because they knew how to perform that test and not because that test was required at that time: "*This video showed me... to check IP… then I will solve it by checking if, you know, the IPs are correct, or this DNS layer values are correct and you know, if we've connected them properly, if there is a mistake in connection. That's how... how I would go about it*", *"Because I thought that the destination was not set or it is sending with IPv6 and port for IPv6 is not there or something like that*"

To summarise, they identified some of the desired actions like the need to understand ideal behaviour, they identified that test hint videos are useful. However, they were still interpreting the information at a shallow level – I see the test hint videos, I want to repeat those in the simulator and see what happens. They were not able to tie it to the troubleshooting process. They were not able to conceive troubleshooting as a structured process.

6.5.2 What are the perceptions of the student about PHyTeR v1?

We answer this question based on the student responses in interviews conducted during and after intervention. We elaborate on the themes generated in the following section. Figure 21 shows a summary of these themes.

Most of the students perceived that the Problem Space Understanding (PSU) activity to understand the ideal network (Task 1) was helpful to understand the problem ("*I watched that and got to know what really should happen*", "*it told me exactly how it should work*") and then to get an idea of the solution ("*Then I should make the network act like this*", "*then I could go and check for these things…*", " *again this also is like a checklist, I can go and check each of these… whether it is happening or not*"). They used this activity to decide their further actions like comparing this step by step behaviour in detail with the faulty network or considering the behaviours as step-by-step prompts for testing. Few of the students found task1 and task2 very similar thus confusing: "*I didn't know what else to do here*", "*…like this is almost like the previous one I did*". Some of them mentioned difficulty in parsing the device specific statements in the activities because they are very similar. Some students felt that the task 2 in PSU was confusing. They were not sure what to do: "*The sentences were almost same… I was confused*", "*I read it again… [the words] ideal and faulty come again and again.. so..*", "*I was not sure what to do in this*".

Many of the students recognised the importance of hypothesis generation activity. They ascribed that a hypothesis helps them to focus on one thing at a time and thus reduces the problem space for that iteration: "*If I have an idea (hypothesis) about something, then I'll go in that direction only*", "*Yeah.. its like now I have only that problem, I have to find if that (device) is (working) correct or not*". Some students thought of hypothesis as a checklist that can be tested one by one: "*So now I can check these one by one*", "*Its like a list that I can check one by one*".

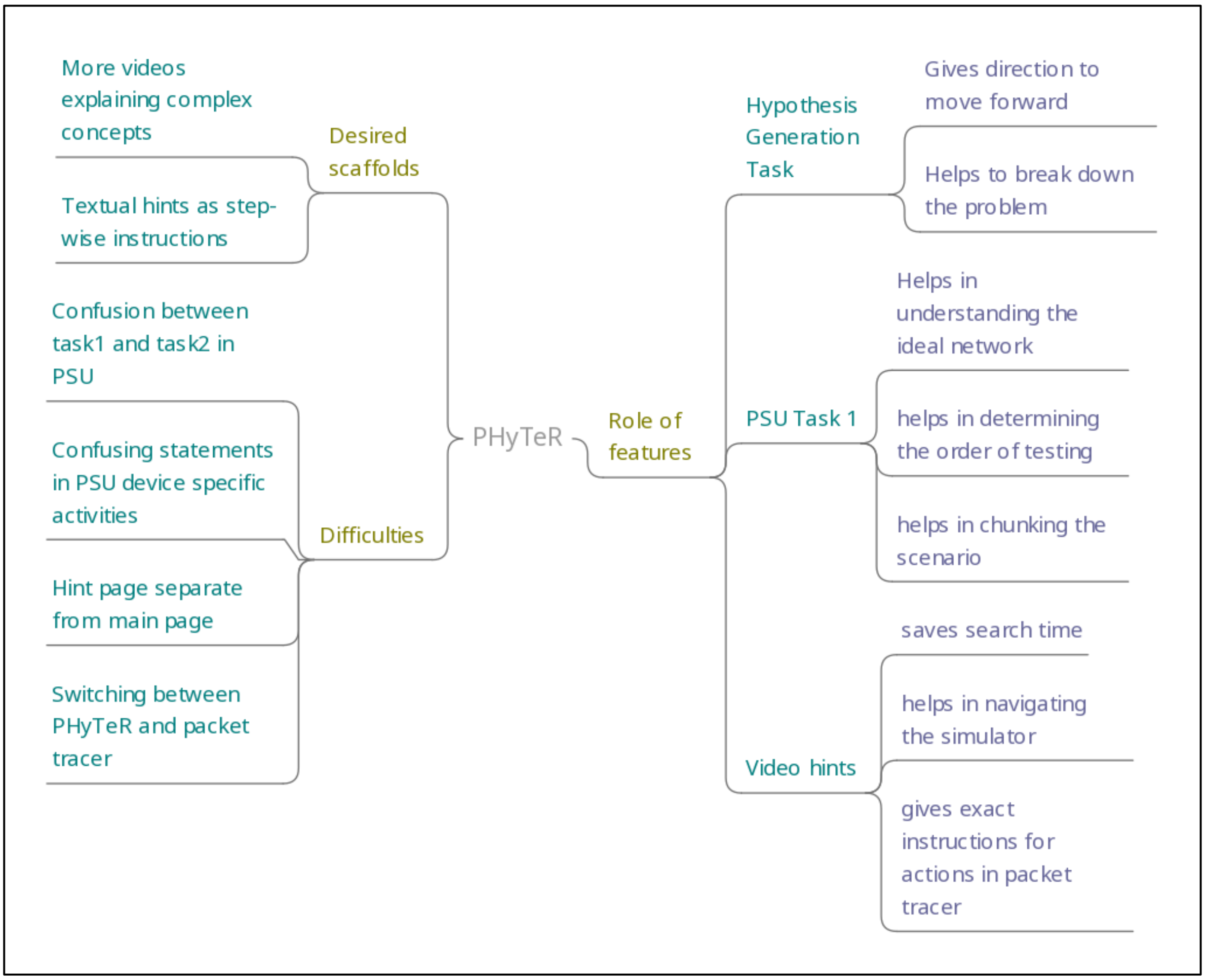


Figure 21: Summary of themes related to student perceptions related to PHyTeR in study 1

Another feature of PHyTeR that the students perceived as most useful was the test hint videos. Most of the students watched all test hint videos and tried to replicate them on the faulty network in the simulator: "*its like exactly what I wanted*", "*If had to search for these videos then… I don't know… it will be 2-3 hours*", "*so I saw all these together… got to know what is possible in the simulator*". Because of these videos, they explored various options available in packet tracer; however, we observed that they were not able to connect it to the troubleshooting process. In addition, they felt the need for more videos that describe complex network concepts: "*It would have been good if we had such videos that combine two or more concepts*", "*…a video explaining how this request actually works…*".

A source of difficulty for many students during the session was that they had to switch between PHyTeR main page, the hints page and packet tracer. They reflected that just in time hints would have been more useful than a separate hints page: "*It would be good if I… like get the exact hints for this test here*", "*… like you can add these* [test hint videos] *in the test page* *itself*'. We also observed that once they enter the hints page, they inclined to watch many videos one after the other, without actually implementing them in packet tracer. Few of the students felt that the textual hints in the hints page were confusing and desired for step-wise instructions: "*I did not understand what it was saying*", "*I read it all and didn't know what to do*", "*It would've been better if it came one by one*".

In this section, we articulated the student perceptions of troubleshooting and PHyTeR after working with PHyTeR v1. We reflect on the implications of this on the pedagogy and design of PHyTeR in the next section.

6.6 Discussion

In this section, we describe the reflection phase of first DBR cycle. We started this cycle with an aim to develop a TELE to teach troubleshooting skills. We analysed relevant literature to understand what is troubleshooting and how it is taught in other contexts. Then, we understood the critical difficulties faced by students in our context. We developed a TELE, PHyTeR and asked students to use it during a troubleshooting session. We identified and synthesized from student responses the features that were useful for them to solve the troubleshooting problem, the scaffolds they desire for and the difficulties they face while using PHyTeR. We now deliberate on what is working as desired in PHyTeR and what needs to be modified/ improved. We do this along each phase of troubleshooting.

Problem Space Understanding

The goal of problem space understanding during troubleshooting is to identify the faulty behaviour (as opposed to the desired behaviour). Specifically one has to identify the devices/component that might be causing the faulty behaviour and systematically find out the real cause. After analysing the difficulties faced by students to do this, we designed problem space understanding activities in PHyTeR.

Now, we summarize our reflection on these activities:

- Activities designed to help students understand the behaviour of a device/network: These activities are one of the two activities that the students perceived as most helpful. They used this as a template/check list to perform tests. That is, they considered the ideal behaviours one by one and crosschecked it with what is happening in the faulty network.

  Reflection: The students used the list of statements given in these activities in a manner that is similar to how an expert would use mental simulation. However, they did not formulate any hypothesis based on it. They tried to compare the ideal behaviour given in the statements with the faulty behaviour of the network in the simulator. When they did not match, they were not sure how to proceed. We infer that they need prompts to connect PSU activities and hypothesis generation activities.

- Activities designed to help students understand the structure & function of a device: The students reported that these activities helped them to clarify some misunderstandings they had about the working of a network. We wanted them to generate hypothesis after getting the misconceptions clarified. However, we observed that the students who tried these activities tend to complete all questions in one go.

  Reflection: We note that they help the students in overcoming alternate conceptions. However, we recognise that these activities do not succeed in helping the students to move towards the solution. That is, the students answer all the questions in one go and are not sure what to do next. Similar to understanding the behaviour, in this case too, the students need prompts to connect these activities to hypothesis generation activities.

Hypothesis Generation

The role of hypothesis in troubleshooting is to provide a structure to the testing cycles. That is, when a troubleshooter starts testing a hypothesis, it provides an anchor to test only that hypothesis and interpret the result in the context of that hypothesis. Thus, it helps a troubleshooter to not lose track of troubleshooting and to narrow down the problem space with every iteration. The summary of our reflection on hypothesis generation activities are reported below:

- To select a hypothesis in PHyTeR, students had to select a device and its component from a drop down menu. Then they had to select a justification for selecting that hypothesis at that point in time. The students used the list of device and components again as a checklist. Many students reported "*This drop down gives a list of all that could go wrong in a router. So I can check each one of them*". In their interviews, they reported that they did not really understand what to do with the justification options.

  Reflection: We identify that the hypothesis is not formulated as a result of observed faulty behaviour but 'selected' from the given list. Ideally, we wanted them to generate a hypothesis after a PSU activity. We recognise that it is hard for them to make the connection between PSU and hypothesis generation activities. We also observed that asking them to justify right after selecting a hypothesis is only adding to their cognitive overload. Therefore, we decide to remove the justification step in the next version.

Hypothesis Testing

An aptly designed test is important for a troubleshooter to accept/reject the hypothesis. It will help her to narrow down the problem space. This test should be aligned to the hypothesis. In PHyTeR, there are multiple tasks corresponding to hypothesis testing. We describe the corresponding reflections below:

- The task to select the test corresponding to the selected hypothesis: In this task, we wanted the students to select a test and evaluate if the test (the results of it) will actually help in concluding that the hypothesis has to be accepted or rejected. We found that the students were not doing the evaluation question like intended. They thought every test that they designed was appropriate for the selected hypothesis.

  Reflection: The students are not connecting the test with the hypothesis. They look at it as another list of options to check. This might also be causing them to think that all tests are appropriate for the selected hypothesis. We decide to reframe the question in the next version of PHyTeR.

- The task to select prediction: This was a drop down menu where students had to select predicted output of the test. There were insufficient prediction options. Students were not able to select an appropriate one from the given options.

  Reflection: Add more options corresponding to all possible tests.

- Hint to conduct the test: In study 1, we found that students often have difficulty to gather the information required to conduct a test. Therefore, we created short video hints showing them how to conduct a particular test. All such videos were available in hints page of PHyTeR v1. After a student had completed prediction task, PHyTeR instructed her to go to hints page in case she did not know how to do the test in the simulator. We observed that once a student clicks on the hints page, she would watch many videos in a stretch instead of watching one video and switching to simulator to conduct the test.

  Reflection: The students are not using this as a hint to conduct the test as expected. Instead of this helping them to focus on doing the task, they are losing track of troubleshooting by watching many videos at a stretch. We decide to integrate test videos in the PHyTeR main screen so that only videos corresponding to the test are accessible.

- Conducting test in the simulator: This was reported as a difficulty in study 1. Therefore, we provided hints in PHyTeR. After watching the short hint videos, the students were able to perform the tests.

  Reflection: No change needed to the video. However, we need prompts to direct them towards next troubleshooting phase.

Result Interpretation

This phase constitutes tasks to compare the predicted and observed result, concluding the hypothesis and deciding the next step. The observations from student interaction with these tasks and corresponding reflection are summarized below:

- Noting down observation and concluding the hypothesis: By the time students saw this task, they had done multiple tests in the simulator and had forgotten the initial selected hypothesis. The students reported that the task made little sense to them. In addition, with respect to the conclusion task, the students reported that they did not understand what they were supposed to do because the options were confusing.

  Reflection: We recognise that the instructions given along with result interpretation task were insufficient. They did not understand what has to be done and how it is connected to the process of troubleshooting. We divide the activities into multiple tasks with clear instruction and prompts in the next version of PHyTeR.

Overall process of troubleshooting

The researchers had verbally described the overall process of troubleshooting before the students started to solve the scenario in PHyTeR. In addition, there is a section in the local wiki that described each phase of troubleshooting in detail. PHyTeR main screen has the phases of troubleshooting as main navigation buttons. We expected them to connect the activities to the phases of troubleshooting.

Reflection: Most of the instructions supported students in executing the activities and creating an artefact (ex: hypothesis, a test plan). However, they did not succeed in making the students understand the need of those activities and artefacts in the process of troubleshooting. The prompts and instructions were not sufficient for them to decide the next step in the process. Often they were not able to grasp the connection between one activity and another. We decide to add elements and prompts that will help the students to make these connections. We recognised that the instructions in result interpretation activity are not supporting them in even doing the activity. The students are selecting options passively in that activity.

In addition to these reflections specific to the activities, we recognised that the students were unable to think of the process as a whole. When asked about the process of troubleshooting, most of them were unable to remember and explain the phases of troubleshooting. We infer that we need to have features that explicitly reiterate the connection between activities, phases and overall troubleshooting.

Based on these observations and reflections, we start the second cycle of DBR. We describe the actions taken in the second cycle in the next chapters.

Chapter 7

DBR Cycle 2: Refinement of Solution

In DBR cycle 1, we started with the goal to implement a TELE to teach troubleshooting skills. After reviewing literature and conducting a study to understand student difficulties, we developed first version of the PHyTeR. We conducted another study to evaluate how students use this during troubleshooting. Based on this study we identified some affordances of PHyTeR that helped students to accomplish learning goals. We identified the affordances that have to be modified and new features that have to be added. In the following section, we summarize the reflection of DBR cycle 1.

7.1 Summary of reflections from DBR Cycle 1

In their perception interview during study2, the students were able to recall and describe hypothesis generation and problem space understanding phases. They identified the importance of these phases in the context of troubleshooting. However, they did not recall/ describe the testing and result interpretation phases. They found it difficult to connect how some activities are connected to other activities. Another broad level theme that cropped up is the students not comprehending the question prompts and getting confused by some statements in PHyTeR. We summarize these in the following:

- The students were not able to recall/recognise all phases of troubleshooting or the need for these phases
- The students were not able to identify the relationship between the phases of troubleshooting
- The students had difficulty in performing some tasks in PHyTeR

In the next section, we describe how we went about addressing these issues. We list the changes made in PHyTeR at the end.

7.2 Changes in Pedagogy/Tool

In this section, we describe the how we addressed the questions from DBR cycle 1. We reviewed literature to identify scaffolds that can address our concerns. We explain these scaffolds in the following paragraphs corresponding to each concern:

- What changes are made to make students aware of all phases of troubleshooting and its need in troubleshooting?

  We added multiple levels of explanatory videos to explain the process of troubleshooting. We added three videos before they started solving the problem. The first video explains the concept of troubleshooting in detail with an example. It explains what the phases are in the context of the example. The second video explains doing troubleshooting in PHyTeR. Here again the phases and activities corresponding to the phases are illustrated in the context of overall troubleshooting process. The third video demonstrates an iteration of activities in PHyTeR with explanations. In addition to these, there are videos attached specific to each phase of troubleshooting that explains how to do activities in that phase and the how those activities contribute towards troubleshooting.

  In addition, we added a reflection activity at the end. The objective of this activity is to summarise the phases of troubleshooting and the activities corresponding to each phase. It asks the students to identify the need of those activities during troubleshooting a network.

- What changes are made to make students connect tasks in one phase to tasks in another phase of troubleshooting?

  We added 'process prompts' to support students in making these connections. These are the prompts that appear at the end of an activity. They summarise the actions performed in that activity and guide the students towards possible next steps. For example at the end of a video feedback in a PSU activity, the narrator says: "Now that you have understood how a network works, do you want to check if this is happening in the faulty network? Do you want to generate a hypothesis based on this?" Another example at the end of each test hint video: "Remember to perform only this test in the simulator and come back to PHyTeR after doing these tests" We added this prompt to prevent the students from changing more configurations than required thus losing track of the troubleshooting process.

- What changes are made to resolve students' confusion while following some instructions?

Emphasize the focus on the task

We changed the structure of activities in the HG, HT and RI phases. In PHyTeR v1 the structure was that the student had to complete a statement. Figure 22 shows an example of prediction activity in PHyTeR v1. The students have to complete the statement: After doing this this test, I… We see that this statement appears after the summary of student actions in previous phases. That is, the actual activity is the third thing that a user sees. In addition, we recognise that the heading (Plan Test – Predict - Evaluate) is not precise for the activity. This heading included all activities in the HT phase and not just prediction.

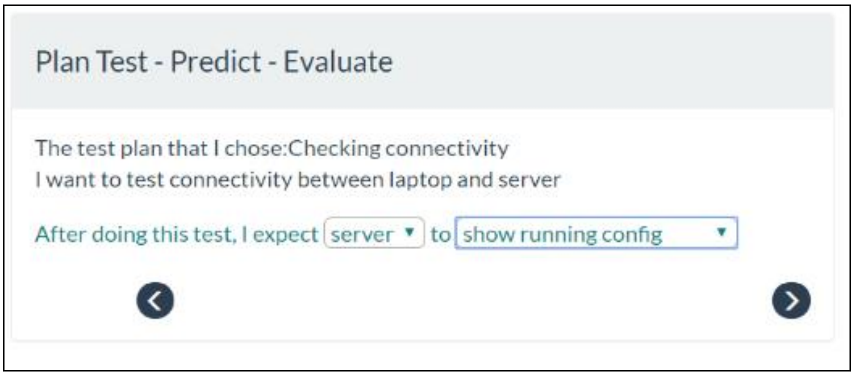


Figure 22: Prediction activity in PHyTeR v1

We changed the structure of this activity as shown in Figure 23. Here, the activity to be done is clearly mentioned as a question prompt. This is the main thing that they see in the screen. The summaries of previous phase decisions are shown as icons in the sidebar. Similar to this example, we changed the structure of all HG, HT and RI activities to consist focussed questions prompts.

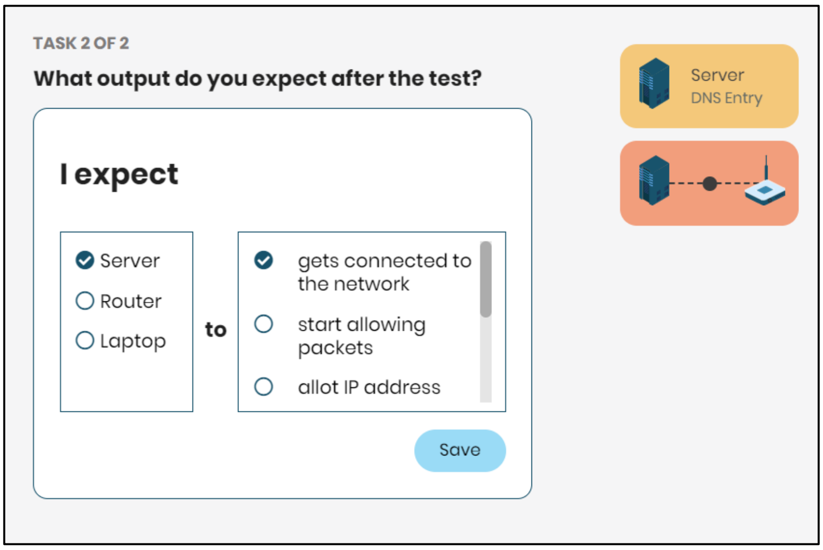


Figure 23: Prediction activity in PHyTeR v2

Divide a complex task into multiple simple tasks

Another change that we made in the second version is to help the students in comprehending the activities by simplifying them. Consider the result interpretation screen from PHyTeR v1 shown in Figure 24. This screen includes multiple tasks like – noting down the observation, comparing it with prediction and concluding the hypothesis. The instructions for concluding the hypothesis were confusing. In addition, this one panel consisted three activities. We simplified this by making three separate panels for each activity. Every panel consists of questions prompts to guide student actions. This can be seen in Figure 25.

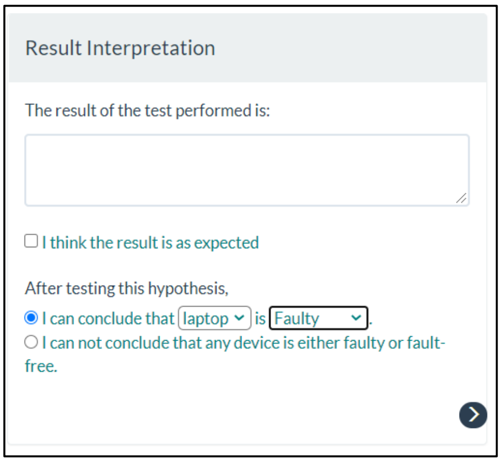
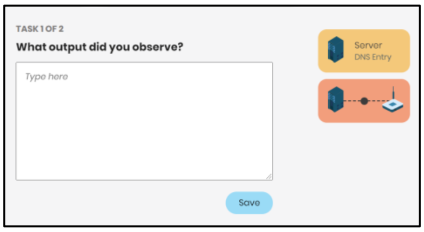
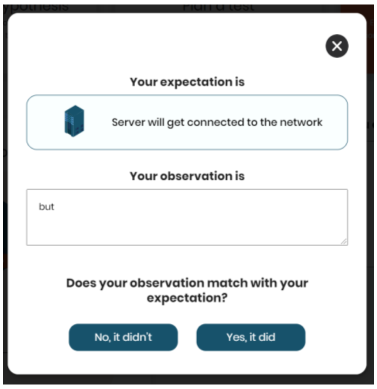


Figure 24: Result interpretation in PHyTeR v1

a) 

b) 

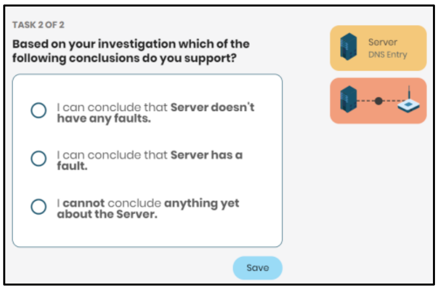
c) 

Figure 25: Result Interpretation activities in PHyTeR v2

In addition, we consulted a UI/UX designer to address the usability issues identified. The UI/UX designer was also present during study2 and noted down her observations regarding UI/UX difficulties of students. Then the researcher worked closely with the designer to overhaul the user interaction of PHyTeR accommodating the newly incorporated scaffolds. In the next section, we explain the second version of PHyTeR in detail.

7.3 PHyTeR v2 – the refined learning environment

In this section, we describe the second version of PHyTeR. After a student logs into PHyTeR, she sees the videos page as shown in Figure 26. In this screen, there are three videos:

- What is troubleshooting: This video explains the systematic process of troubleshooting that is employed by PHyTeR. It describes the 4 phases of troubleshooting in detail with an example in the networking context.
- Introduction to PHyTeR: This video explains various features of PHyTeR and how to use them.
- Iteration in PHyTeR: This video explains a complete iteration of 4 phases in PHyTeR. It illustrates all the tasks that have to be completed in order to complete an iteration from a student's point of view.

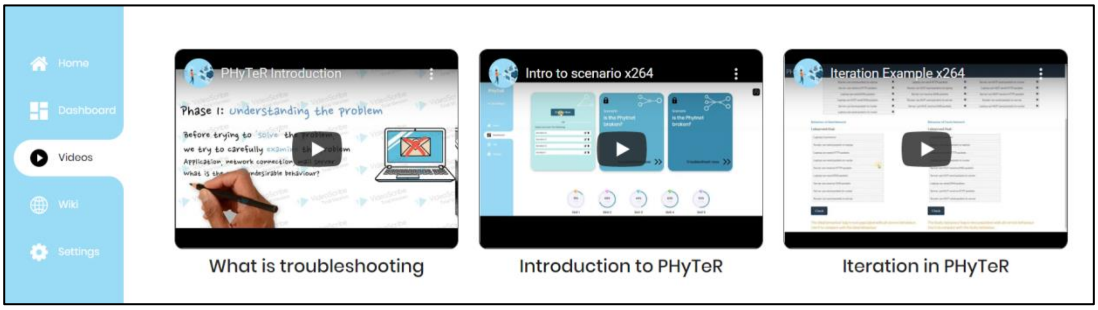


Figure 26: The introduction videos in PHyTeR

These videos are not mandatory and the student can skip watching these videos. After that, she goes to the dashboard to start troubleshooting a network scenario. When she clicks to start troubleshooting the scenario, the first iteration begins. She can come back to this dashboard and access all previous iterations anytime during troubleshooting. The dashboard with iteration history is shown in Figure 27. She can create a new iteration (mandatory for first time user) or she can select a previous iteration.

When they enter the troubleshooting screen, the scenario that has to be solved is displayed as shown in Figure 28. There is a video icon beside the description. On clicking this icon, a video of the troubleshooting scenario as seen in packet tracer is played. They can click on their name and access these problem descriptions anytime during troubleshooting.

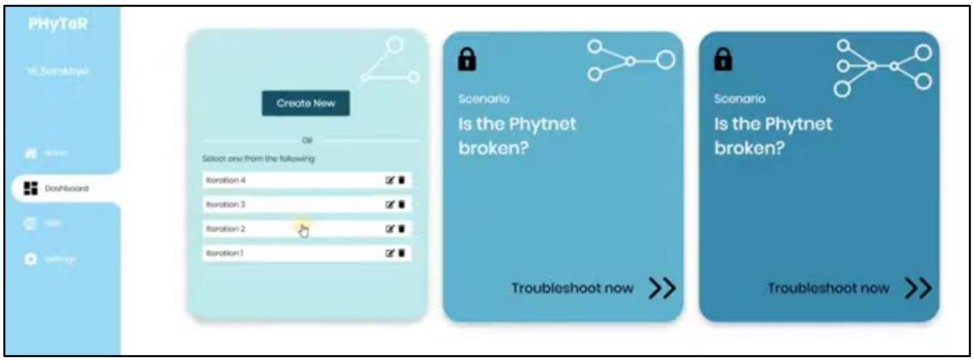


Figure 27: Iterations in PHyTeR v2

The scenario that has to be solved is: You are trying to access the website www.et.co.in from laptop0. The website is hosted on the web server. You are unable to access the website. What will you do?

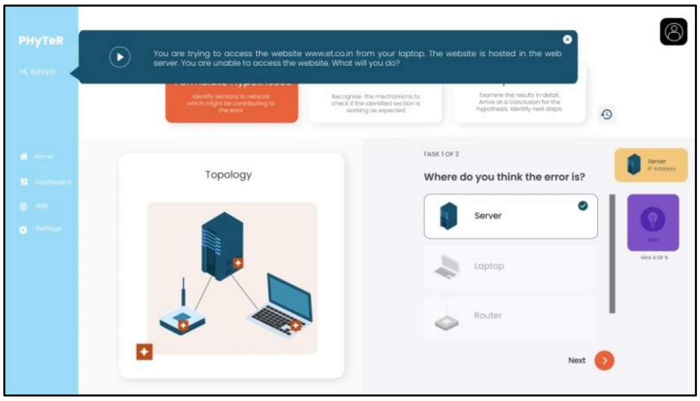


Figure 28: Scenario description in PHyTeR v2

Each phase has a tab and that is activated when the students starts doing the activity corresponding to that phase. It also contains a video hint that explains the need for that phase and how to do the activities corresponding to that phase. It has another hint that provides expert insights corresponding to the phase. The video icon and the expert icon in Figure 29 represent this.

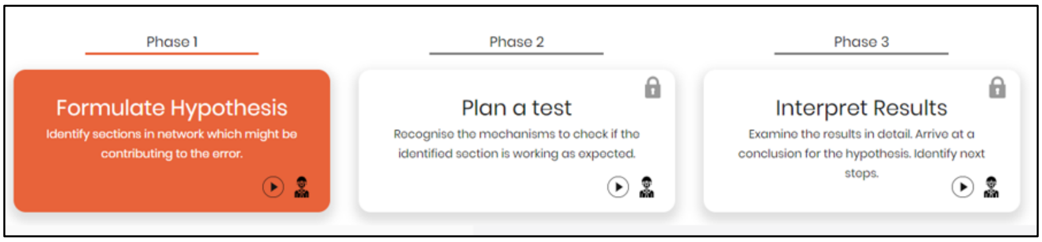


Figure 29: Phases with the scaffolds

The problem space understanding (PSU) activities are present on the right side of the screen. There are four activities corresponding to the understanding of the entire network. The first three activities are intended to evaluate their understanding of behaviour of the network. In these activities, the student is asked a question (Ex: How does a web request work?). In addition, PHyTeR gives the statements corresponding to the behaviour of the network in a jumbled manner. She has to rearrange the statements in the correct order within three tries. When she finds the correct answer or is out of tries, she can watch a video. This video shows the correct behaviour corresponding to the question in packet tracer. At the end of each such video is a prompt that asks the student to generate a hypothesis based on the knowledge acquired from the video. This activity and video feedback are shown in Figure 30.

a) b)

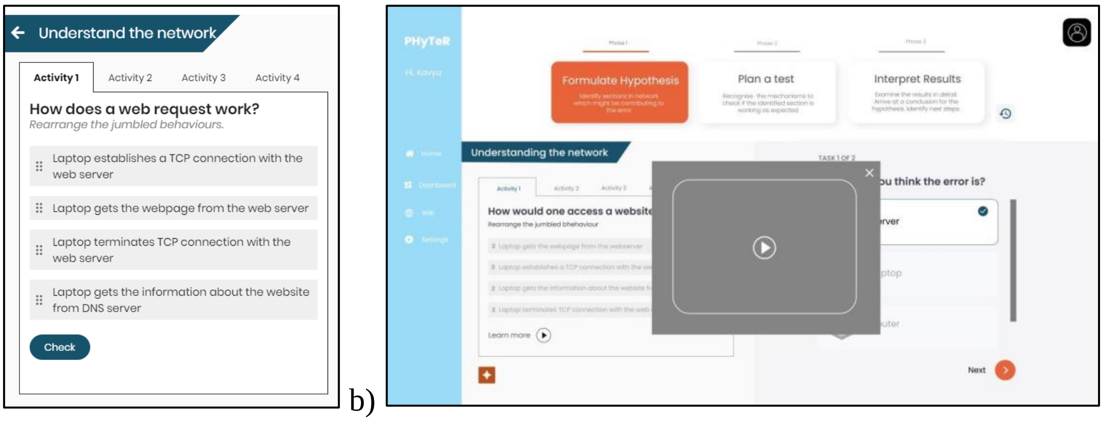


Figure 30: PSU activity with video feedback

The fourth activity corresponding to understanding the entire network is designed to help the student understand the erroneous network. The previous three activities will help her in understanding the ideal behaviour of the network. Here in the fourth activity she can depend on that understanding of the ideal network and compare the step-by-step behaviour of the ideal and faulty network. Again at the end of this activity, there is video reinforcing the understanding of ideal and faulty networks. At the end of each of these videos, there are prompts to use the differences between ideal and faulty networks to generate a hypothesis.

The right side panel of the screen corresponds to the activities corresponding to hypothesis testing cycle. The activities are similar to those in PHyTeR v1, which are explained in section 5.2. We have changed the structure of the activities to include question prompts and divided the complex activities into simple ones, as explained in section 7.2. The student iterates through the hypothesis testing cycle referring the PSU activities as and when required until the fault has been recognised. At the end of each iteration, PHyTeR prompts to find out if the actual fault has been recognised as shown in Figure 31. If the student has found out the actual cause, they have to select the device and the component where the fault was found. If it is correct, PHyTeR infers that troubleshooting is complete and shows the reflection activity to the student. If the selected option is incorrect, then PHyTeR prompts the student to perform another iteration.

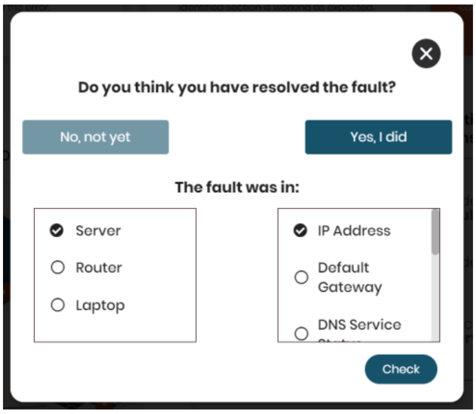


Figure 31: Prompt to recognise if the student has found the actual fault

The screenshot of reflection activity is shown in Figure 32. In the first reflection activity, the students watch an interactive video that summarizes the activities performed during the course of troubleshooting. This video also has reflection questions in between to make sure that students have understood the concepts. After this, the students have to do a fill in the blanks activity. In this, the student sees a list of actions and she has to drag and drop the actions in the phase in which the actions are done.

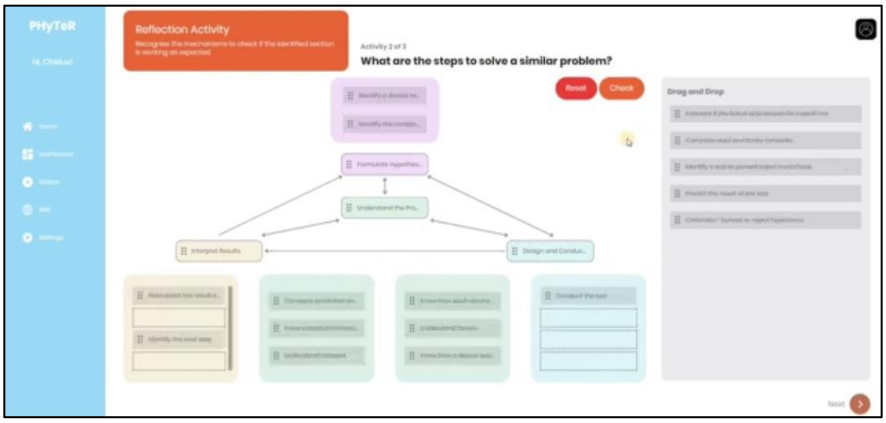


Figure 32: Reflection activity in PHyTeR v2

7.4 Summary

In this chapter, we described the problem analysis phase of DBR cycle 2. We elaborated the modifications done in PHyTeR based on the analysis. We discuss the evaluation of PHyTeR version 2 in the next chapter.

Chapter 8

DBR Cycle 2: Evaluation

In previous chapter, we described the changes done in PHyTeR after the first round of evaluation. The rationale for those changes is to help students accomplish the learning goals that were difficult to achieve with PHyTeR v1. In this chapter, we elaborate two studies done to evaluate the second version of PHyTeR. The first one, study3, is a qualitative study carried out to methodically understand student interactions with PHyTeR. We conducted study4 to triangulate the results from study 3 using other measures.

8.1 Study 3 - Evaluating PHyTeR v2

In this section, we elaborate the research questions, methods, data analysis techniques and results of study 3. We discuss the results after describing study4 in next section.

8.1.1 Research Questions

The research questions that guided this study are:

RQ 2.2: After interacting with PHyTeR, what is the students' understanding of troubleshooting skills?
RQ 3.2: How do students use the features of PHyTeR v2 to troubleshoot?

8.1.2 Methods and materials

8.1.2.1 Population and sampling

Five female students volunteered to participate in this study. Two of them were studying in their 4th year of Bachelors in Engineering degree and three of them in 3rd year. We conducted this study in the workshop mode at the researcher's house (because of lockdown restrictions). The workshop was conducted over a period of four days. Figure 33 describes the sessions conducted over four days. Each colour represents sessions that happened in one day. In the following section, we describe the details of these sessions.

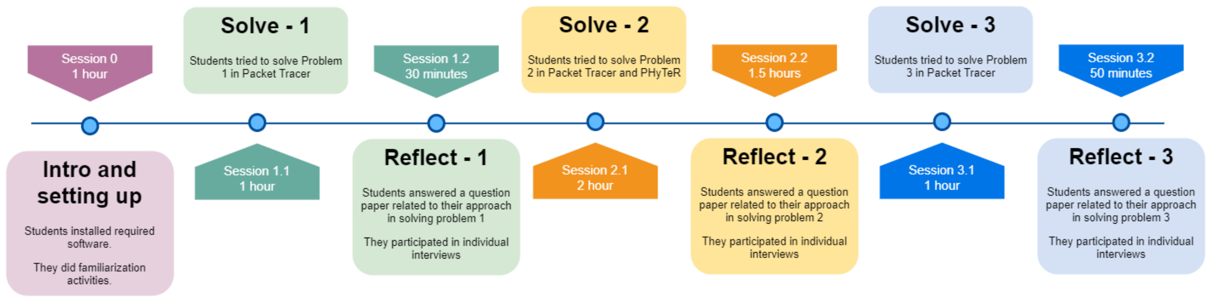


Figure 33: Timeline of Study3

8.1.2.2 Procedure

Day1:

The first day was just an introduction to the workshop. I asked the students to install the required software for network simulator and screen recording. The network simulator used was Cisco Packet Tracer. I asked them to install a software called auto screen capture. I gave them instruction to set up the screen capture software so that it captures a screen grab once in every 3 second. I asked them to do some dummy activities while screen recording was on. This was to ensure that everything was working properly.

After that, I asked them to watch four videos. Three of these videos were about configuring one device in Packet Tracer. One video was about checking connectivity between devices and using ‘simulation’ mode in Packet Tracer. These videos briefly described all the configuration settings that were needed to solve all three problems during the workshop. Each of these videos was 1 to 2 minute long. These videos were accessible to them throughout the workshop.

Day 2:

In the first session of the second day of the workshop, I asked the students to solve a network-troubleshooting problem (Problem 1) using Packet Tracer. They were allowed to use the internet and search engines if they needed any information. They were given around 1 hour to solve this problem.

Problem 1 description: You have to send packets (ping) from laptop 0 to laptop 1. However, that is not happening. Troubleshoot this issue. Figure 34 shows the topology of problem 1. The solution to this problem was turning the ports on in the router.

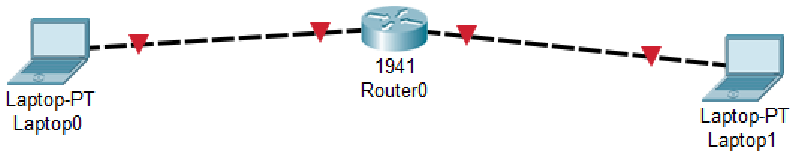


Figure 34: The faulty topology used in the pre-test (Problem 1)

After they spent around 1 hour trying to solve the problem, they were asked to stop. The second session of the second day began. I gave them the question paper reproduced in Appendix IV. This question paper has four questions. Each of them corresponds to 4 phases of troubleshooting. I gave this question paper after each problem in the workshop. After collecting the student's response sheet, I conducted an interview with each of them. That marked the end of the second day.

Day 3:

On the third day, I introduced them to PHyTeR. The students registered into PHyTeR and logged in. After that, they started with the familiarization videos. Then, they started solving problem 2 (Figure 35) using PHyTeR and Packet Tracer. The faulty topology was in Packet Tracer. They were instructed to start doing activities in PHyTeR.

Problem 2 description: You are trying to access the website www.et.co.in from your laptop. The website is hosted on the web server. You are unable to access the website. What will you do? The faulty topology is shown below.

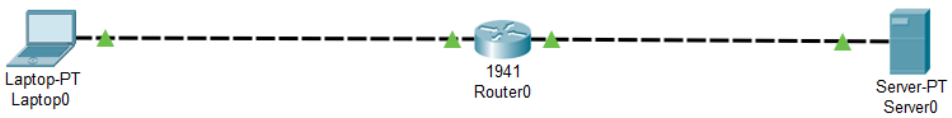


Figure 35: The faulty topology used in intervention (Problem 2)

The solution to this problem is: Every website has a corresponding DNS Entry which maps the IP address to the website's URL. This DNS Entry is stored in the DNS Server, which is the same as the web server in this case. There was an error in the DNS Entry for this website. They had to change it to the correct address.

They used both PHyTeR and Packet Tracer for around 2 hours while trying to solve problem2. After that, I asked them to answer the question paper. Then I interviewed them individually. It was a stimulated recall interview where the student described their troubleshooting process by using the screen-capture images as a memory aid. That was the end of the third day.

Day 4:

On the first session of the fourth day, I gave them another network troubleshooting problem (Problem 3) as shown in Figure 36. They had to solve this without PHyTeR using only Packet Tracer.

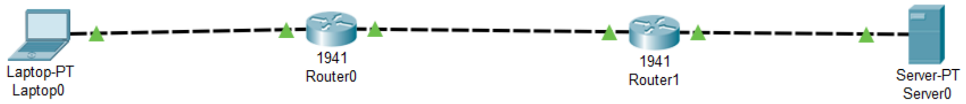


Figure 36: Faulty topology used in post-test (Problem 3)

Description for problem 3: You are sending packets (ping) from laptop0 to server0. However, the packets are not reaching server0. Troubleshoot this issue. The solution to this problem was that there was no route set up between the two routers. So there were no packets moving between them. In order to troubleshoot, they had to set up a route between the two routers.

They spent around 1 hour doing this. After that, they I gave them the same question paper and a feedback form. Individual interviews about their conceptions of troubleshooting followed.

8.1.3 Data Analysis

We had interview recordings and screen capture images. We generated around 120 ‘episodes’ of troubleshooting actions by the students. These episodes were analysed to answer the research questions.

In order to answer RQ 2.2, we considered the student responses to the following interview questions: What do you understand by troubleshooting?, How would you troubleshoot a similar problem?

I listened to the complete interview of two students and read the transcripts relevant to these two questions. I used the data of two students to generate initial codes set of codes. The unit of analysis was a sentence. Using these initial set of codes, I coded the responses of other three students using these codes. I verified the codes and descriptions by comparing it with the transcriptions to ensure consistency.

In order to answer RQ 3.2, I considered the students responses to the question: How did you solve this problem? While conducting the interview, this was the seed questions and then the interviewer followed up with many questions to understand the students' approach of troubleshooting in PHyTeR.

I followed a similar data analysis procedure like the one followed to answer RQ1. I read the transcript of the answers and created a broad workflow for each participant. This workflow consisted of two levels: actions performed, their explanation of the action performed obtained from the interview response. For the actions in the workflow, I annotated the PHyTeR features that were used to perform the action and meta-level notes. Then I clubbed the parts related to one feature to derive themes.

8.1.4 Workflow

In this section, we describe the workflow of student S2 as a representative of how students interacted with PHyTeR during the study.

S2 opened PHyTeR in a web browser. She registered an account in PHyTeR and logged into it. She watched the three videos that are part of the familiarization activity. She clicked on the scenario to start troubleshooting. She read the problem description and watched the problem description video. She watched the hypothesis generation video and then the problem space understanding video. These videos explain the goal of the phase and describe how to perform the tasks in the corresponding phase using examples. S2 paused the PSU video at the example screen and studied the example carefully. She watched the example parts 2-3 times.

Subsequently, she did three activities in the “understanding the network” part of the PSU activities. These activities are – “How does a web request work?”, “How does a DNS request work?” and “How does a web server work?” She saw the feedback video when she could not get the correct answer for the first activity. She got the correct answers for second and third of these activities and chose not to see the feedback videos.

After doing these activities she started the formulate hypothesis task. She selected the server and its HTTP service status as the possible cause of the error. In order to test this, she selected the option ‘to check the connection between laptop and the server’. She watched the test hint video and opened packet tracer to perform that test. When she performed the test, she could not decide anything about the hypothesis because the test was not aligned to the hypothesis that she had formulated. She changed the IP addresses of the server and the laptop and then tried to send packets again. This time the packets sent from the Laptop successfully reached the server. She tried accessing the website from the Laptop and found out that it was still inaccessible.

S2 opened PHyTeR and started doing result interpretation tasks. Because she had done multiple tests, she could not do the result interpretation task correctly. She clicked on “Plan a test again”, and completed the task by entering the details of most recent test done. This ends the first iteration of hypothesis testing by S2.

After this, she started doing the remaining activities in “Understanding the network”. She did the activity of comparing ideal and faulty networks. She watched the feedback videos twice. She formulated another hypothesis. She selected router as the probable cause of error. She selected to test the IP address settings of the router. She opened packet tracer to inspect the IP address in router. After checking that, she sent a packet from the laptop to the server. The packet failed to reach the server.

She went back to PHyTeR and watched the videos describing ideal and faulty networks. She opened packet tracer, tried to access the website and compared the behaviour of the network in packet tracer with that of the ideal network (as shown in the video). She did this by switching multiple times between packet tracer and video in PHyTeR. She observed that a packet reached router and did not travel further. She also observed the type of packets that gets transferred (TCP packets) and the type of packets that get failed (HTTP packets). She returned to result interpretation task in PHyTeR and concluded the iteration by marking the router as faulty.

She then did two activities in “understanding the laptop”. Based on this she went and changed IP addresses in laptop configuration settings. She tried accessing the website from laptop again. This time she was able to access the website. The troubleshooting session ended. The researcher asker her a few questions about ‘What was the error?’ and ‘How did you resolve the error?’. She was able to explain the actual cause and important aspects of the solution process.

8.1.5 Results

All five students who participated in the study identified the correct device at the end of the troubleshooting session. Three of them identified the device component correctly, one required additional prompts to identify and another could not identify the device component before she gave up. We analysed around 120 episodes of troubleshooting actions performed by these 5 students.

In the next sub-section, we elaborate the conceptions of students about troubleshooting and the role of some features in PHyTeR.

RQ 2.2: After interacting with PHyTeR, what is the students’ understanding of troubleshooting skills?

We answer this question based on the student interviews after the intervention and the interviews after the post-test.

We asked them the following questions in the interview:

1. How did you solve this problem?

2. What do you understand by the term ‘troubleshooting’?
3. How would you troubleshoot a similar problem?

Students identify that troubleshooting is a systematic process: “*Troubleshooting is a step-by-step procedure…*”, “*It is a process where we have to first analyse the problem, think of possible solutions and then implement the solution*”. Their understanding is not limited to cursory mentions of the steps of the process but this structured-ness is involved in their narration of their own troubleshooting episode: “*First I sent a packet from the laptop. It was not reaching this router… So I thought the error might be here in this router… Then I selected this* [checking connectivity between router and server] *option to test this…*”, “*First I’ll understand the problem, then select a device… I’ll see if there is error there… if not I’ll select another device*”, “*I’ll send packets and see where it fails… that is where I’ve to test… If error is still there, I’ll go and select another device*”

Though they do not use the words PSU, HG, HT and RI in their description, they include the crux of these phases in their description. They allude to problem space understanding as ‘*understanding the network*’, ‘*understand what is happening*’, and ‘*understand the existing thing*’. They identify the need for problem space understanding and possible actions to do that: “*First I need to know what is happening here…*”, “*I’ll first try to understand where the packet is failing...*”, “*I will try to search for what should happen ideally*”. Some of the students referred to hypothesis while others used ‘selecting a device’ to denote generating a hypothesis. However, all of them recognised the usefulness of having a hypothesis to “*check whether it is correct or not*”.

Another theme emerging from the analysis is that students talk about troubleshooting as evidence based activity. In other words, they acknowledge that they need to collect data from the existing network before performing any actions: “*I’ll send a packet from laptop1… I’ll see where is fails, then I’ll check that device*”, “*I’ll ping and see till where the packet goes.. if it breaks here, then that means there is some problem here*”, “*I need to first see where the packet is dropped”, “I’ll search for what is wrong*”. We observed this approach in relation to hypothesis generation and result interpretation too: “*If the packet gets dropped here, then I’ll select this device*”, “*I’ll first try to see what is wrong, based on that I’ll select a device*”, “*after you do the test, see if the packet goes from router till here* [server]… *then decide if the first* *selected device is correct or not*", "*the IP address was correct there* [in the router]*, so I though it is not the problem*"

The third theme about the understanding of troubleshooting is that students think of it as an iterative process. This is in contrast with what students perceived before the intervention: "*if this did not work then I'll select another hypothesis*", *"...I'll see if the problem is solved, if not I'll go back to first step*", "*by selecting one device and checking if has the error, … if not then select another device*"

The fourth theme about student understanding of troubleshooting is that they perceive that troubleshooting can be broken down to smaller problems. This breaking down is in terms of both the problem space and the process of troubleshooting: "… *I'll now go with this device* [to generate a hypothesis]", "… *so this device has to be tested*", "*so I can focus only on this device now*". Figure 37 depicts a summary of these themes.

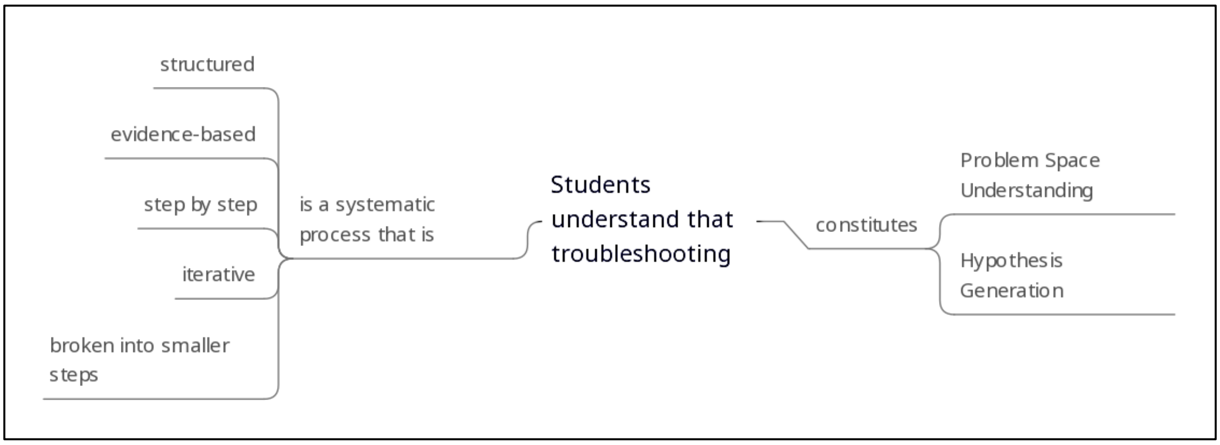


Figure 37: Student understanding of troubleshooting – Study 3 Results

RQ 3.2: How do students use the features of PHyTeR v2 to troubleshoot?

We analysed the interview responses of the students to identify the parts where they had talked about PHyTeR and its features. During the post-intervention interview, we started with a broad question – 'How did you solve this problem?' This was a stimulated-recall interview (Lyle, 2003). We used a screen capturing software to save an image once every 3 second. We used this approach instead of capturing a video to prevent the screen capturing software from crashing. We used these images as a memory aid for students during the interview. We used these images during the analysis only to verify if there was any confusion in the audio recording.

Role of PSU activities:

More than one student mentioned that the activities helped them in thinking about the network in a systematic manner: *"Yes... It was like stepwise thing, I could go to simulator and check each of these... were there","... It gave me the exact steps to do things…*" Few students perceived that the activities help students to think of the system as made up of multiple parts working together: *"it… gave me an idea of how different parts of this network behave… I can only see this (server) in detail now"*. This ability to break down also helped to narrow down the problem space. They also recognised that the activities help them to think of the network in ideal terms: "*… it explains the role of each device, what it should actually do*", *"I knew what a router does… I mean it forwards packets… But I really understand now how it happens", "I thought the router will have only one IP address, but when this question showed me wrong, I checked it again".* They found that the activities cleared their understanding of the network.

Role of structured task environment

The structured task environment consists of different tabs corresponding to each phase of troubleshooting. Each tab has a video explaining the rationale and tasks corresponding to that phase, expert insights specific to that phase and question prompts to direct them towards tasks corresponding to that phase.

The students mention that the just-in-time instructions given in the form of videos helped them to set sub-goals during troubleshooting: "*after that I watched this video… then I understood what has to be done*", "*In the first video it said all the things to be done, but this short video made it clear again…*". The expert insights provided in the form a list of points were too abstract for the students to appreciate, thus they found it either confusing or futile: "*I did not understand what it was saying…*", "*I opened it… read a little bit and closed it…*".

Students found it easier to work with prompts compared to the question paper that was given in pre-test. S1 told: "*Yeah, the question paper had one big question, I didn't know what* *exactly to do. Here, this tells me what to do*". The question prompts guided the students throughout the troubleshooting process: "*Then it [PHyTeR] asked me to make a test for the hypothesis*", "*It asked me to predict the result*", "*It told me to select one among the three*". Figure 38 shows a summary of these themes.

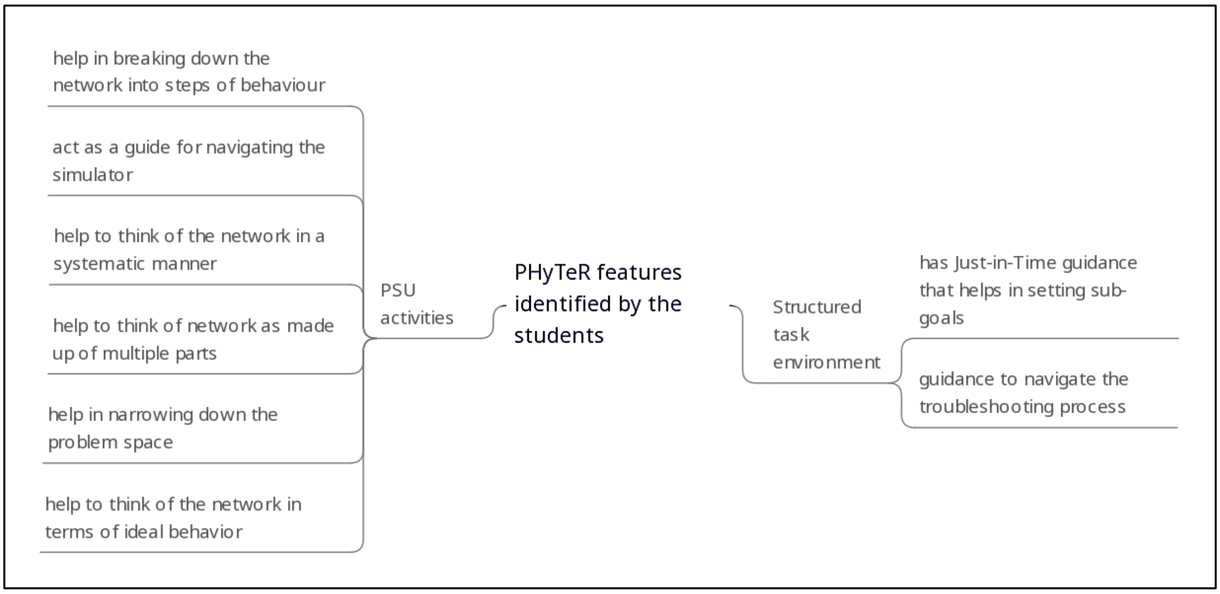


Figure 38: Students perceptions about role of PHyTeR features

In this section, we described the student perceptions about troubleshooting and the role of some features in PHyTeR. In the next sections, we elaborate study4. We conducted study 4 to triangulate the results of study3 doing quantitative confirmation of the qualitative results. We write the discussions related to study3 and study4 at the end.

8.2 Study 4 - Evaluating PHyTeR v2

8.2.1 Research Questions

In study 4, we evaluated the following research questions:

RQ 2.3: After interacting with PHyTeR, what are changes in the students' understanding of troubleshooting skills?

RQ 2.4: After interacting with PHyTeR, what are the changes in student perception of students related to troubleshooting actions, knowledge and confidence?

8.2.2 Method and materials

This study was conducted as an online network troubleshooting workshop in a college in Mumbai. The teacher teaching computer networking announced about this workshop to the students with the registration link. Totally 53 students registered for the workshop. The workshop was conducted for two days, one session per day. We sent two familiarization videos to the students who registered. We asked them to watch the videos before the workshop. Figure 39 shows the timeline followed in this study. Registration happened before the workshop. Questionnaire1, PHyTeR activities happened on day1 and Reflection and questionnaire2 happened on day 2.

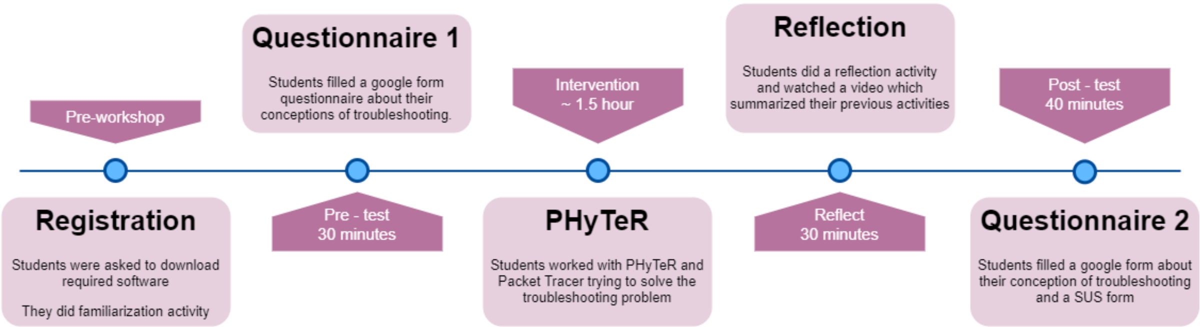


Figure 39: Timeline followed in study 4

Day 1:

Out of the 53 students who registered, 42 students attended the first day of the workshop. On the first day, we gave an introduction about the sessions to the students. Then we gave a questionnaire in Google forms. We designed this questionnaire to understand their approach towards troubleshooting. The questionnaire can be found in appendix V. After completing the questionnaire, the students solved a troubleshooting scenario in PHyTeR and packet tracer. They did all activities in PHyTeR except the reflection activity. That ended the first day.

Day 2:

Only 20 of the 42 students attended the second day of the workshop. I gave them a summary of the activities that they performed on Day 1. Then they did the reflection activity in PHyTeR. After completing this, I gave them another questionnaire. The questionnaire can be found in appendix V. We designed this questionnaire to understand their approach towards troubleshooting after interacting with PHyTeR and to identify the changes in their approach.

8.2.3 Data Analysis

We collected data using detailed questionnaire given to students before and after their interaction with PHyTeR. We did content analysis of the form responses to generate themes to answer some questions. For others, we depict the difference between pre and post in graphs. There were nine male and 11 female students. The workshop was conducted at the end of semester in which they studied computer networking course.

8.2.4 Results

8.2.4.1 Understanding of troubleshooting

RQ 2.3a After interacting with PHyTeR, what are changes in the students' understanding of troubleshooting skills?

In order to answer this question we compare the student responses to some of the questions in the pre and post intervention questionnaire. In the pre- intervention questionnaire, we consider the responses to the following questions:

1. What do you understand by the term 'troubleshooting'?
2. How will you approach troubleshooting this scenario?

In the post-intervention questionnaire, we consider the responses to the following questions:

1. What do you now understand by the term 'troubleshooting'?
2. How will you approach troubleshooting this scenario?
3. Your understanding of the term 'troubleshooting' - Has it changed from previous understanding? If yes, why has it changed and how did it change?
4. Your approach towards troubleshooting a given scenario - Has it changed from previous? If yes, why has it changed and how did it change?

We break the answer along the following aspects: their understanding of the term troubleshooting, their approach towards troubleshooting, phases of troubleshooting.

Understanding of the term and overall process of 'troubleshooting'

Before the intervention, students connected troubleshooting with repair - so the description of troubleshooting was mostly in the form of *'find the error and resolve it'*. After undergoing the intervention, students have recognised that troubleshooting is a '*logical*', '*systematic*', '*stepwise*' and '*iterative*' process/ procedure. Even when they had thought of it as logical before the intervention, they perceive that the "*details of this logical process are clear"* after the intervention. In their post-intervention questionnaire, some of the students talk about hypothesis as an integral part of troubleshooting: "*the process of rectifying an error after coming up with various hypotheses*", "*... identifies the fault, formulates hypothesis for the problem, checks all the possible cases ...*", "*find out the error in a particular system and try to list down the possible issues or faults*"

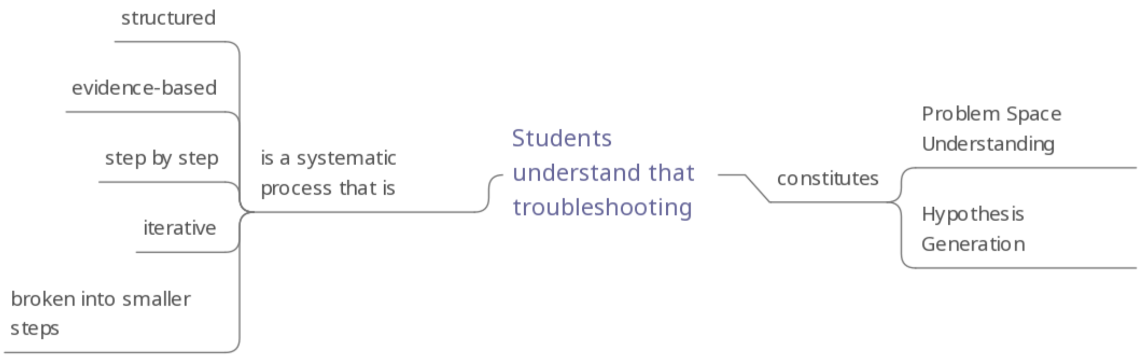


Figure 40:Student understanding of troubleshooting

Another change in some students' understanding is with respect to the domain of computer networks. When asked about understanding of troubleshooting before the intervention they described troubleshooting in generic terms as recognising the fault, resolving the error. After the intervention, the description involves computer network specific terminology like - 'ensuring smooth transfer of data packets', 'ensuring that the network devices have been configured properly'

Another aspect of change that we observed in their description of troubleshooting is related to the strategy used to narrow down the problem space. Before the intervention, most students perceived that strategies like trial and error and exhaustive search are feasible: *"I'll start with this device and go on checking all the devices one by one"*, "*checks for all issues possible and provides a solution to it*". After interacting with PHyTeR, many of them appreciated the importance of reducing the problem space using optimal strategies: *"I'll send the packet and see till where it goes. I'll then check the device where the packet fails"*, "*…make decisions to move to the right track to complete the faulty part*" In addition to this, we observe that, this narrowing down approach has been recognised at the device level. That is, they identify a device as faulty and then still follow trial and error or exhaustive search within that device: *"After selecting the device, I'll check every configuration in that device"*.

RQ 2.3b. After interacting with PHyTeR, what is the student understanding of phases of troubleshooting?

In the post-intervention questionnaire, we asked students about their understanding of each phase via the following questions:

1. What do you understand by the term 'X'?
2. What actions/searches would you do to 'X'?

Where 'X' stands for each phase of troubleshooting. In order to answer this question, we analysed the student responses for eight questions, two each related to one phase of troubleshooting.

Problem Space Understanding

After interacting with PHyTeR, some students recognise PSU as the starting point in a troubleshooting process: "*first I need to understand the theory*", "*In order to formulate a hypothesis, we need to understand the scenario that's going on*".

They recognise that understanding a network includes identifying the connections between the devices and required functions of devices: "*basically the understanding of the communication and interaction of the network with the devices involving data transmission*", "*proper functioning of devices connected in a network*", "*devices should be connected to each other in* *a proper way in order to facilitate messages between each other properly*", "*…how the devices are connected to each other on a network*". A few students have identified the means of achieving this understanding: "*I'll go and read wiki*", "*I'll search in Google for this info*", "*I'll run the simulator and see what is happening*". Some students have identified the need to understand the problem space and the role of comparison with the ideal network: "*when I understand the correct working, then I can go to packet tracer and see what is actually happening there*", "*Once we understand what is going on, then formulating the hypothesis on basis of assumptions is really simple*", "*Then I will try to find out how the connection would have ideally taken place without the error*"

Hypothesis Generation:

Many students have identified the link between PSU and HG, and that HG flows organically from PSU: "*I'll understand this* [how the network behaves] *and then generate*", "*First we have to understand the network and the device connections and then we have to lead by selecting a route to find the error*", "*In order to formulate a hypothesis, we need to understand the scenario that's going on*"

Some students think of hypothesis as informing further actions during troubleshooting: "*then we have to lead by selecting a route to find the error*", "*Formulating a Hypothesis is essential in any test as it helps in telling us whether our assumption at the beginning was correct or incorrect*". We observe that most of them have started thinking in terms of multiple hypotheses: "*define different cases and points which could have caused an error*", "*coming up with possible faults*", "*If our hypothesis turns out to be incorrect, we test another hypothesis*". This is in contrast to what we observed in the student difficulty study where they fixated on only one cause.

However, we find that the students still think of exhaustive search as a feasible option at times: "*After I identify the device, I'll go through every setting in detail there*", "*In a device I'll list all possible options*", "*I will list down the possible scenarios and cases and check for the faults and errors one by one*", "*I would note down all the possible faults in the system firstly*"

Hypothesis Testing:

Almost everyone identified the alignment between the generated hypothesis and the test: "*Based on hypothesis, find solution based on device which …*", *"… plan for a test and actually implement it on our network to check if our formulated hypothesis is true*", "*After formulation of a hypothesis, we need to take action to test our hypothesis*".

Few students have recognised the relevant actions to perform a test: "*Check connection between devices or check configuration*", "*I would go with checking the configuration of the device since it provides a full view of the device details*", "*I would check connection, change configuration, execute some command…*" and the importance of meticulous planning: "*taking all factors into consideration and lastly thinking about things that could go wrong*", "*decide the test that has to be run and for that we need an accurate plan to do so*"

We identified that a few of the students think of testing as 'solution' to the troubleshooting problem. This might be because when they go and check the configuration and see that it is incorrect, they change it and see the network working correctly as a result of the change made: "*plan the solution for the hypothesis, check the solution by testing the hypothesis*", "*identifying the causes of error and solving them one by one*"

Result Interpretation:

We note that when talking about result interpretation, most of the students associate it with concluding the hypothesis: "*error which you had assumed is correct or no*", "*We have to conclude the result with the proceedings according to hypothesis and plan a test*", "*whether or not our test and thus hypothesis has been successful or not*". Few of them have identified that comparing the predicted and observed result is how one arrives at the conclusion: "*check the difference in expected and observed results*", "*First we have to write the observation from our point of view and then we have to compare it with the observation obtained*".

Fewer of the students have identified that RI includes choosing next step based on current iteration: "*arrive at a conclusion and come up with the next step*", "*And if hypothesis is not successful, we start afresh with another hypothesis*"

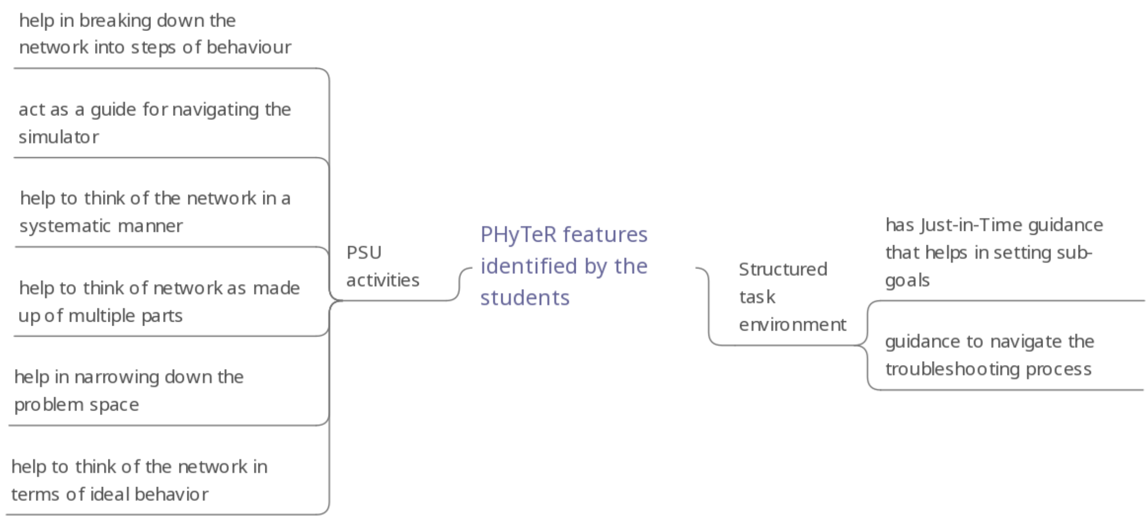


Figure 41: Student understanding of phases of troubleshooting

In study3, we observed the students during troubleshooting. We conducted detailed stimulated recall interviews and analysed them to come up with the themes about student perceptions and understanding of troubleshooting. In study4, we were collecting quantitative data about student perception and understanding of troubleshooting. Along with the open ended questions mentioned in the previous section, we asked them questions on what would they do (actions), think (knowledge) and feel (confidence) in a troubleshooting situation to collect rich information about their behaviour. We answer the below RQ in the following sections.

RQ 2.4: After interacting with PHyTeR, what are the changes in student perception of students related to troubleshooting actions, knowledge and confidence?

8.2.4.2 Confidence in doing troubleshooting

In the pre and post questionnaire, we asked the students about their confidence in doing various troubleshooting tasks. Table 13 gives the list of questions asked in pre and post questionnaires.

Table 13: Statements used in self-reporting the confidence of doing troubleshooting

Troubleshooting Phase	Question Number	Statements given in Pre questionnaire	Statements given in Post questionnaire

Problem Space Understanding	1	Describe the ideal working of the network (I can explain how the given network works when there is no error)	Identify the ideal behavior of a given network
	2	Identify the ideal behavior (behavior in case of no error) of a network device in the given network	Identify the ideal behaviour of a network device
	3	Identify the difference between working of ideal and given faulty network	Identify the difference between ideal and faulty network
Hypothesis Generation	4	List one or more probable causes for the error	Come up with hypotheses
Hypothesis Testing	5	Systematically examine each cause listed to check if it is the actual cause	Design appropriate test for a given hypothesis
Result Interpretation	6	Decide which cause to examine next based on my examination	Interpret the result of a test

Figure 42 shows the change in their confidence levels along various tasks related to each phase of troubleshooting. Though we do not report statistically significant increase in their perceived confidence levels, we observe that the confidence levels have increased for all tasks in the post-questionnaire. We also observe that this increase in confidence is greater for problem space understanding.

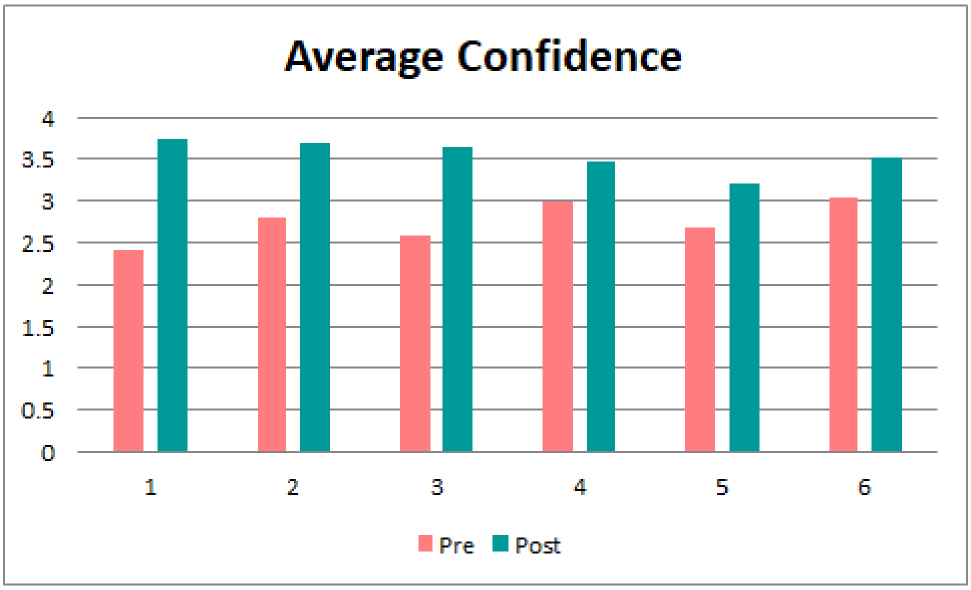


Figure 42: Pre-Post comparison of confidence of doing troubleshooting actions

8.2.4.3 Troubleshooting action preferences

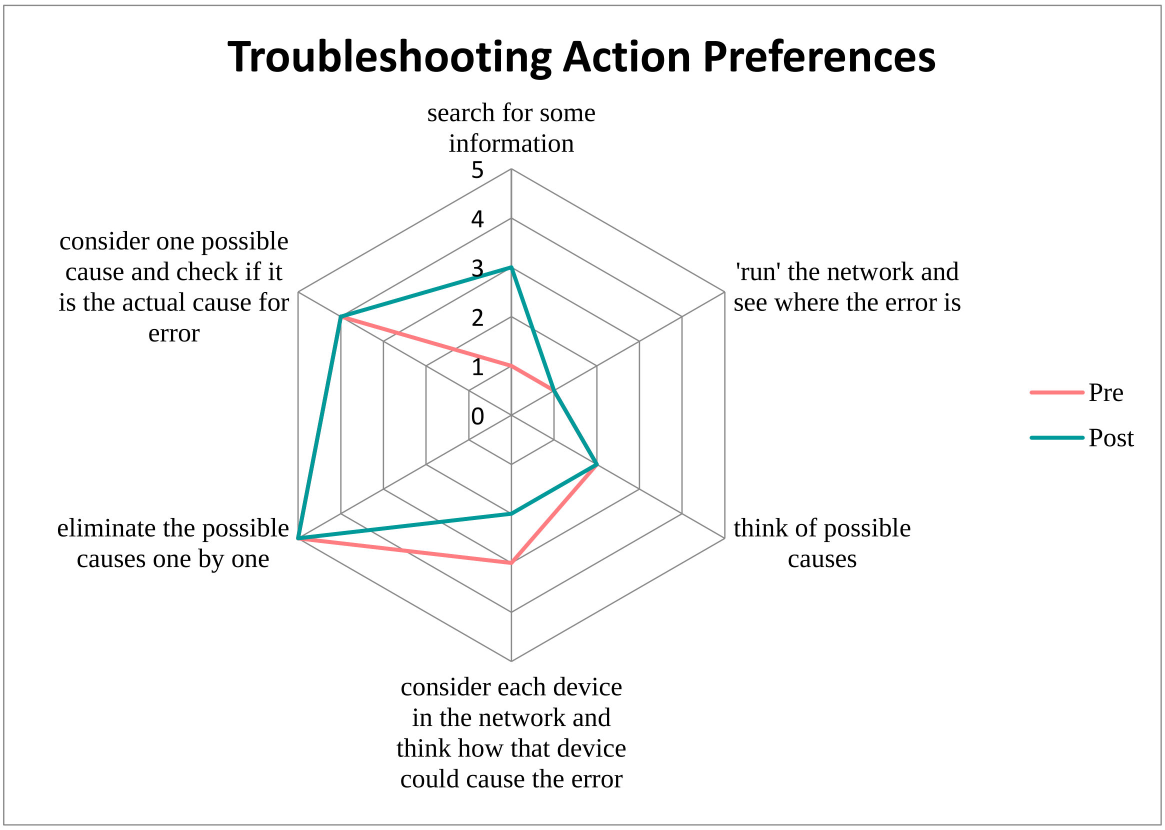


Figure 43: Pre-post comparison of action preferences during troubleshooting

The priority in pre-intervention condition:

Priority	Actions
1	search for some information
1	'run' the network and see where the error is
2	think of possible causes
3	consider each device in the network and think how that device could cause the error
4	consider one possible cause and check if it is the actual cause for error
5	eliminate the possible causes one by one

The priority in post-intervention condition:

Priority	Actions
1	'run' the network and see where the error is
2	consider each device in the network and think how that device could cause the error
2	think of possible causes
3	search for some information
4	consider one possible cause and check if it is the actual cause for error
5	eliminate the possible causes one by one

In the radar graph shown in Figure 43, we depict the changes in the troubleshooting action preferences of the students from pre to post questionnaires. This is based on the student response to the question shown in Figure 44. We see that before interacting with PHyTeR, students would start with searching for information or ‘running’ the network. After the intervention, searching for information moved to third priority. Most of them want to ‘run’ the network first, think of possible causes and then search for information. We also observe, in the case of post-intervention, many of them assign same priority to PSU activities (consider each device in the network and think how that device could cause the error) and HG activities (think of possible causes).

While troubleshooting the above scenario, which of the below mentioned actions would you do and in which order?

The rows correspond to actions that you might do while troubleshooting. Note that all actions are not mandatory. Select only those which you are more likely to do. The columns represent the order in which you would do them.

	1	2	3	4	5	6
I'll think of possible causes for the error						
I'll eliminate the possible causes one by one						
I'll consider each device in the network and think how that device could cause the error						
I'll 'run' the network and see where the error is						
I'll consider one possible cause and check if it is the actual cause for error						
I'll search for some information						

Figure 44: Action preference question asked in the questionnaires

8.2.4.4 Perception of knowledge required for troubleshooting

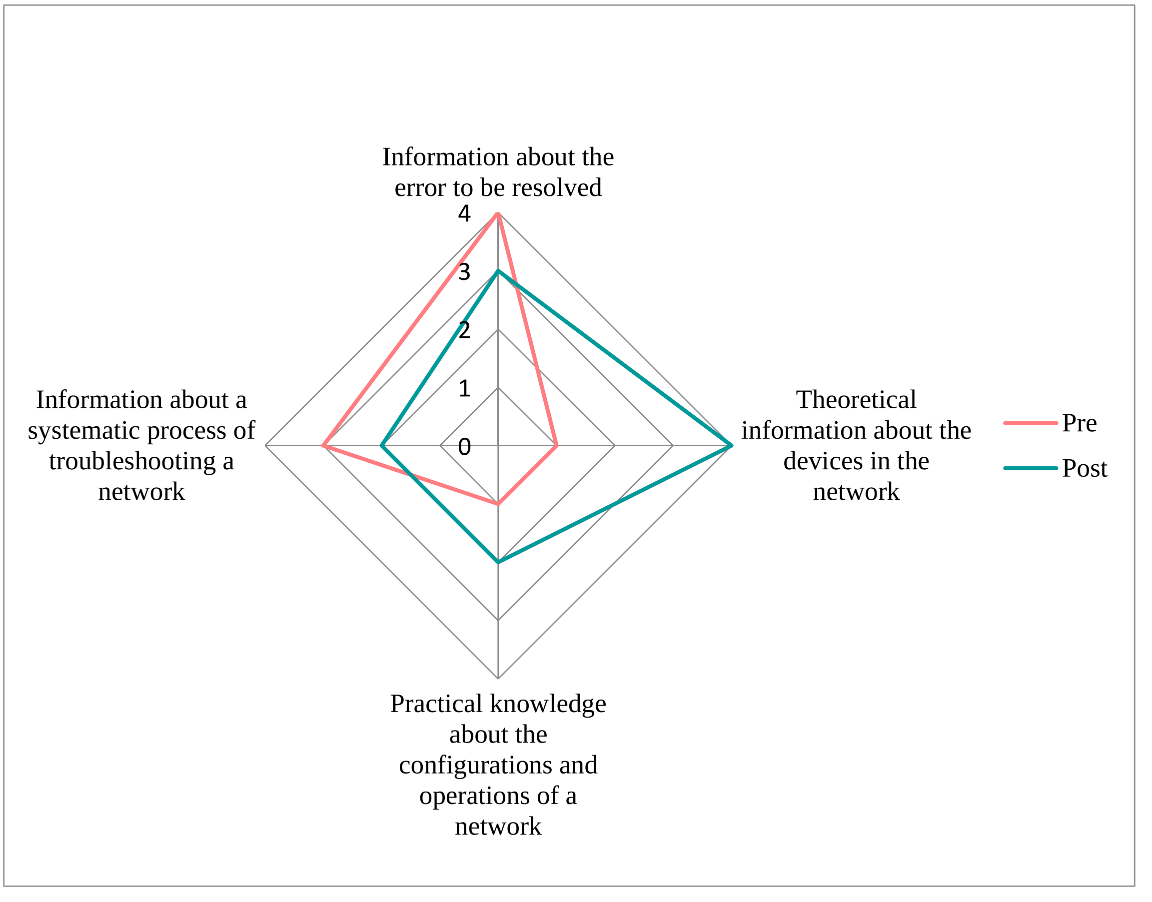


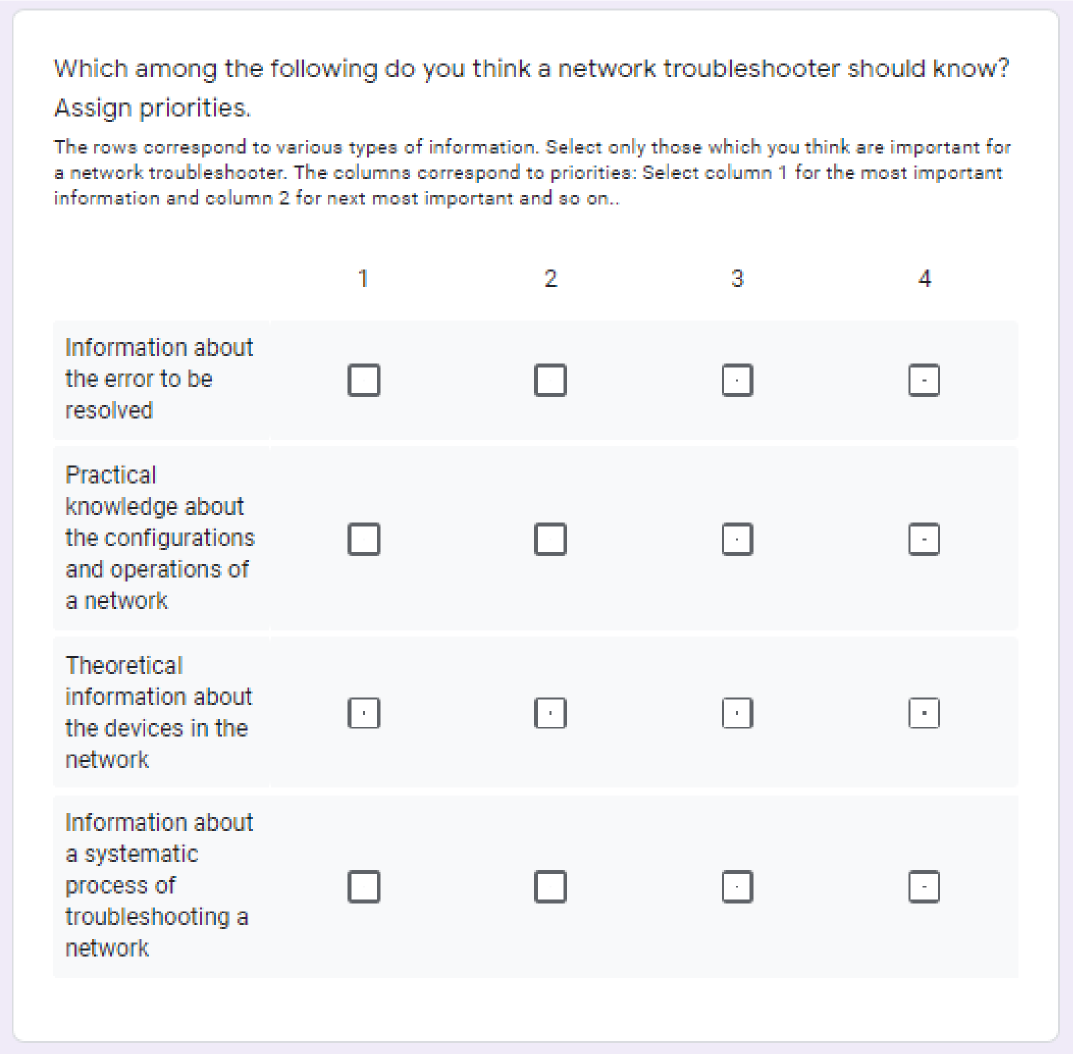
Figure 45: Pre-post comparison of perception of knowledge required for troubleshooting

Table 14: Knowledge priorities before intervention

Priority	Knowledge
1	Theoretical information about the devices in the network
1	Practical knowledge about the configurations and operations of a network
3	Information about a systematic process of troubleshooting a network
4	Information about the error to be resolved

Table 15: Knowledge priorities after intervention

Priority	Knowledge
2	Practical knowledge about the configurations and operations of a network
2	Information about a systematic process of troubleshooting a network
3	Information about the error to be resolved
4	Theoretical information about the devices in the network



Which among the following do you think a network troubleshooter should know? Assign priorities.

The rows correspond to various types of information. Select only those which you think are important for a network troubleshooter. The columns correspond to priorities: Select column 1 for the most important information and column 2 for next most important and so on..

	1	2	3	4
Information about the error to be resolved	☐	☐	☐	☐
Practical knowledge about the configurations and operations of a network	☐	☐	☐	☐
Theoretical information about the devices in the network	☐	☐	☐	☐
Information about a systematic process of troubleshooting a network	☐	☐	☐	☐

Figure 46: Questions used to elicit the knowledge perception of students

Figure 45 shows the change in perception of knowledge required for troubleshooting from pre to post. We note that, before intervention, students thought theoretical information about devices as most important along with the practical knowledge. After interacting with PHyTeR, they give last priority to the theoretical information related to the devices. This might be because, i) after working with PHyTeR, they realise that they can narrow down the problem and then search for the required information ii) while interacting with PHyTeR, they have learnt sufficient theoretical information and do not feel the need to look up again. The first possibility here is supported by their action preferences (8.2.4.3) too. Another observation is that they think knowledge of systematic troubleshooting process is more important than the information about the actual error. They probably think of narrowing down the problem using systematic methods and then search for information about the errors. This is based on the responses to the question shown in Figure 46.

8.3 Discussion with respect to Study 3 & Study 4

The objective of study3 and study4 was to discern the student understanding of troubleshooting after interacting with PHyTeR. In study3, 4 out of 5 students successfully identified the correct device. In study4, 8 out of 20 students identified the correct device at the end of intervention. In study3, we used the stimulated recall interviews where the participants narrated their troubleshooting episode with screenshot images as memory aids. In study4, we gathered data using detailed questionnaires that were given to the students before and after they interacted with PHyTeR. In study3, the students started by answering the question “How did you solve this problem?”, whereas in study 4 we asked them about the phases, actions and knowledge directly.

Both the studies indicate a shift in student perception of troubleshooting towards troubleshooting as a systematic, iterative and evidence based activity. In study3, we derive this from their implicit understanding of the phases and actions required at each phase. In study4, we asked them explicit questions about each phase to elicit their understanding of troubleshooting and their perceptions about actions and knowledge required. We hypothesize that PHyTeR prompting them to follow 3 phases sequentially, and them getting practice while doing multiple iterations is a probable reason for the changes in their understanding of troubleshooting. We want to contrast this with the reflections from study 2 where students identified some desirable behaviour but were having difficulty in connecting them to the overall troubleshooting process.

We observe that the students have identified the usefulness of PSU activities in breaking down the problem space, understanding the problem in detail. They think of PSU as the first step in troubleshooting and as a necessary predecessor to hypothesis generation. We hypothesize that the process prompts that connect every PSU activity to hypothesis generation has helped the students to make this connection between PSU and hypothesis generation. We make similar arguments for the evaluation prompts in hypothesis testing (that ask students to check the alignment between the hypothesis and the planned test) and result interpretation (that ask students to infer the result in the context of the hypothesis) have enabled students to make connection between hypothesis – testing – result interpretation. Most of the students performed three iterations in PHyTeR, so they got repeated practice in this structured environment.

Another notable change that we observed in both studies was the way in which they generated their hypothesis: Most of the students said that they would ‘send a packet’ and see where it fails before selecting a device to dig deeper. However, we did not always observe this logical approach when they were selecting a component within the device. This might be because most of the process prompts and hints talk at a device level and not very specific to components. PHyTeR does not give any guidance/feedback of selecting the configuration within a device. This lack of contextual feedback at component level might be the reason for students still thinking that trial and error approach is feasible at a component level.

Though many students have recognised the importance of ‘logical’, ‘stepwise’ and ‘systematic’ approach, very few have included all 4 phases in their understanding/description of troubleshooting. We hypothesize that it would need some more practice before they can appreciate how each phase is important and why/when to do a ‘phase’. In the next chapter, we discuss these results in the context of the broad research goals of this thesis.

Chapter 9

Discussion

9.1 Summary of the research

In this thesis, we started with the broad objective of developing troubleshooting among CS undergraduates. We followed the research methodology of design-based research (DBR) to investigate the research objective. We carried out two iterations of DBR cycles. In the first cycle, we started with analysing the nature of troubleshooting. In this problem analysis phase, we explored the meaning and process of troubleshooting pertinent to domains like electronics, aviation, chemical engineering, and mechanical systems. We investigated the process of expert troubleshooting in various domains. We conducted a study (study1) with five students from the target population to identify the difficulties faced by them during network troubleshooting. At the end of this phase, we defined the process of troubleshooting in the context of network troubleshooting. We split the process into 4 phases - problem space understanding (PSU), hypothesis generation (HG), hypothesis testing (HT), and result interpretation (RI). We identified the learning objectives for each phase and assessment rubrics for the same.

In the design and development phase of the first cycle, we identified the broad pedagogy of the technology enhanced learning environment (TELE) based on the learning objectives. We also designed the learning activities and scaffolds required to perform those learning activities. These learning activities along with the scaffolds were implemented in the first version of the TELE called PHyTeR. This version of PHyTeR was used by 21 students while trying to solve a network troubleshooting problem (study2). In this evaluation, we identified the features of PHyTeR that helped students to move along the desired direction. We carefully investigated the features which did not lead to intended interaction with the students. These results and reflections from the evaluation study served as the starting point for the second cycle of DBR.

We identified that though PHyTeR v1 helped students in doing individual tasks, some of the learning goals were yet to be achieved. In addition, students had difficulty in doing few tasks in PHyTeR v1. We started the second cycle of DBR with an intention to resolve these concerns. We looked the literature to answer these concerns. We introduced process prompts, reflection activities to PHyTeR. We modified the UI to integrate different tasks in a coherent way. We changed the task structure to have question prompts to help students focus on the task. This PHyTeR v2 is created in the design and development phase of DBR cycle 2.

We evaluated PHyTeR v2 in two studies. Study3 is a qualitative study. Five students participated in a workshop including pre-test, intervention and post-test. The researcher observed the students during the intervention. She interviewed the participants after pre-test, intervention and post-test. The interview after the intervention was a stimulated recall interview. We carefully examined how student use the features of PHyTeR during troubleshooting. We conducted study4 with 20 students. The goal was to identify how student understanding of troubleshooting changes after interacting with PHyTeR. We found that the students identified the importance of a structural approach to troubleshooting. They acquired desired strategies like observing the faulty behaviour and narrowing down to one device before starting troubleshooting. They are aware of the benefits of comparing the faulty network with the ideal one.

In the next section, we elaborate the answers to the research goals in the broader context of teaching troubleshooting skills.

9.2 Answering Research Questions

We started with the broad goal of teaching troubleshooting skills to computer science undergraduates and developing a TELE in this regard. We derived specific research questions in relation to the broad goal along the course of this thesis. We discussed the answers to the research question in the context of a particular study in previous chapters. In this section, we discuss the answers to these research questions in the context of broader research goal.

9.2.1 What difficulties do students face during troubleshooting?

We conducted a study (study1) to determine the difficulties faced by students during network troubleshooting. This study was an open-ended study where we observed five students while they tried to troubleshoot a faulty network in a simulator. Each student worked independently for initial 80 minutes. When they could not troubleshoot in that time, the student worked with two instructors in an instructor-facilitated session to troubleshoot the network.

We identified their difficulties in terms of skill, knowledge and actions. We observed that they did not spend much time in understanding the problem. Their domain knowledge was insufficient to equip them to generate hypothesis, plan and conduct tests and interpret the results of those tests. We identified difficulties specific to each phase of troubleshooting - that students have difficulty in thinking about the functions of a device and the structural components required to attain that function. We noticed that they work with 'proto-hypothesis' that neither helps them to design an appropriate test nor allows them to arrive at useful conclusions. We noticed that they tend to lose track of troubleshooting more often in the testing phase, and that they are unable to think of dependencies while changing a configuration. They find it difficult to follow the thread of hypothesis until its logical conclusion. We identified that the students are unable to systematically move from interpreting the result of one hypothesis to the conclusions about the broader troubleshooting problem.

These difficulties prevented them from taking useful steps while troubleshooting the network. The results of this study formed one of the bases to design the intervention PHyTeR and to evaluate the effectiveness of PHyTeR in resolving these difficulties.

A possible explanation for difficulties related to problem space understanding can be students' lack of a rich mental model and thus difficulty in simulating the behaviour of the network (both ideal and faulty ones). This has been observed in other domains like electrical troubleshooting [20], car mechanics [23], aviation [18] etc.

Johnson [20] reported that students performed with little or low difference to expert level performance in terms of acquiring and interpreting most types of information, generating hypotheses or carrying out procedural tests. Though we did not have an expert level to compare with, we can say that students faced many struggles in our context that an expert might not have faced. Unlike other studies reporting student difficulties in technical troubleshooting where students were mostly vocational students (getting trained to become troubleshooters in their respective domain), students in our context were undergraduate students with no special training on troubleshooting. As reported by them, this was their first attempt at solving such troubleshooting problems in the simulator. This might explain many of the difficulties that we found during hypothesis generation and hypothesis testing phases.

Some of these difficulties have been identified in other domains and reported by other researchers mainly difficulties related to functional understanding of students and hypothesis testing cycle. Our study provides the manifestation of these difficulties in the context of computer networks - that the students have difficulty in identifying how functions of 2 devices contribute to the behaviour of the network, they get confused between function/behaviour of 2 similar devices, they have difficulty in testing one hypothesis and evaluating the broader problem, difficulty in identifying the dependencies of a configuration change etc.

9.2.2 After interacting with PHyTeR, how does the student understanding of troubleshooting change?

We answer this question based on study 2, 3 and 4. We compare the students' understanding of troubleshooting before and after interaction with PHyTeR. In study 2 and 3, we had the students solve a troubleshooting problem before introducing PHyTeR. After this, we interviewed the students about their approach and understanding of troubleshooting. In study 4, students filled a questionnaire about their troubleshooting approach. In study 2, since it is the cycle1 of DBR, we have collected perception data as a proxy for understanding.

We recognised that the students are aware and follow some structured approach of troubleshooting after interacting with PHyTeR. For some it is basing their hypothesis on observed faulty behaviour. For others it is narrowing the problem space to a specific device by doing tests.

We did not expect the students to recognise the specific phases of troubleshooting and start using them after solving one scenario in PHyTeR. We know that, even with enough troubleshooting expertise in one domain, people have difficulty in troubleshooting a problem in another domain. Literature also provides evidence where expert designers are not always expert troubleshooters. This implies that procedural knowledge of troubleshooting in a related domain is not sufficient to troubleshoot even with enough conceptual knowledge. The expertise in troubleshooting is based on the richness and organisation of knowledge based on previous experiences. We tried to provide a structure to this organisation and we see evidences of it helping the students during troubleshooting.

9.2.3 How do students use the features of PHyTeR during troubleshooting?

When students enter the PHyTeR learning environment, they are presented with three videos about troubleshooting and PHyTeR. They watch these videos to get themselves familiarized with their tasks. They started solving the scenario by reading the description and sometimes watching the description video.

Most of the students started with PSU activities. They watched the phase video and did one or more activities in PSU. If they felt a need to understand the problem more, they continued doing the activities. They created hypothesis when they found a behaviour that they wanted to verify in the faulty network. Few students watched the feedback videos only if their answer was incorrect while few of them exhausted three tries to watch the feedback videos. We argue that the students use these as an alternative for mentally simulating the network - to chunk the network and come up with step-by-step behaviours of the devices.

We observed few students switching to packet tracer from PSU without formulating a hypothesis mostly during first iteration. They struggled with hypothesis testing and result interpretation activities in such cases. However, after completing one iteration they followed the phases in sequence. They performed 1-5 iterations to find the faulty device and component.

9.3 Claims

In this section, we elaborate the claims that we make based on our analysis of experiment results and reflections in the context of the design based research.

- We assert that students have difficulty in executing a structured troubleshooting process. This includes:
    - difficulty in analysing the network in terms of its constituent devices, connections between them and a detailed knowledge of how those devices behave in the given situation
    - difficulty to manage the process of troubleshooting methodically from generation till the conclusion of a hypothesis,
    - difficulty to switch between the context of broad troubleshooting problem and carrying out the nitty-gritties of troubleshooting.

  We make this claim based on study1 and study2. In study1, we collected data to understand these difficulties in detail. In study2, we analysed the student interactions with PHyTeR v1. Even with several scaffolds to aid their troubleshooting, students still faced difficulties in executing troubleshooting tasks. This leads us to the second claim.

- We assert that PHyTeR Pedagogy helps students to
    - Identify the systematic nature of troubleshooting
    - Recognise the importance of knowing the ideal behaviour of a network
    - Apply hypothesis testing cycle tasks to network troubleshooting problems

  We make this claim based on study2, study3 and study4. In study2 that used PHyTeR v1, many of the feedbacks were not interwoven with the corresponding actions. We observed that students had difficulty in navigating through the learning environment, there was no sense of 'iteration' and students were not able to get an overall idea of the process. In study 3 & 4 where PHyTeR v2 was employed, we observed that there were fewer instances of students getting lost during the process and mostly they were able to come back and complete the iteration when they got lost. They had a more structured notion of troubleshooting and the actions associated to troubleshoot a network. This included thinking of troubleshooting as an iterative and evidence based process. They recognised that knowing how a network behaves helps them to come up with hypothesis and breakdown the problem space. Repeated practice of doing hypothesis testing cycle tasks gave them a sense of a complete iteration and the tasks involved in it.

- Our third claim is related to problem space understanding activities present in the PHyTeR learning environment. The PSU activities where students had to sort jumbled statements and categorise ideal and faulty behaviours seem to have made students think about the detailed working of the network. These activities presented the information at different levels of abstraction and different points of view (overall network view, device specific views, ideal and faulty views). After doing these activities, most of the students were able to describe the detailed working of the network in a logical manner, were able to identify the roles of devices.

We base this on the results of study2, study3 and study4. We observed that few students used these activities as a revision of domain knowledge while others learnt important networking concepts in detail. After doing these activities, most of the students were able to describe the detailed working of the network in a logical manner, and were able to identify the roles of devices.

9.4 Limitations

In this section, we elaborate various limitations of this research:

- **Networking concepts and nature of errors:** PHyTeR has a troubleshooting scenario based on the concepts of IP addresses, routing, DNS server. Apart from this, the students in our study worked with two more problems in pre and post-tests. These problems were derived from the same concepts as well. All these problems had single errors. Although the process of troubleshooting includes the same four phases for problems with multiple errors, these problems would be more complex. The pedagogy would need explicit prompts in hypothesis generation and result interpretation phase. We could not implement these in the context of the thesis. We wanted the students to learn to troubleshoot deterministic problems first. Randomly occurring problems are out of scope of this thesis. The objective was to give the student a robust structure of troubleshooting process – so some domain complexities are intentionally excluded in the first problem. Once the student is comfortable with generating hypothesis and separating the malfunctioning part, such complexities can be introduced.

- **Learner Attributes**: In this thesis, we investigated and analysed student troubleshooting from a cognitive perspective. We believe that other dimensions like affect, motivation are equally important. However, more studies have to be conducted before claiming anything related to those perspectives. All students who participated in our studies volunteered to take part. This implies certain level of motivation on part of these students. This thesis did not investigate the effect of PHyTeR on students who have low motivation. Though we have added familiarization activities in the beginning to set the context and give a broad idea of troubleshooting by giving real life examples, we have to investigate if these prompts are sufficient for students with low motivation. All the students who participated in studies were from tier-3 engineering colleges in India. The results might not be completely applicable to students from other distinguishable cohorts like students from tier-1 colleges.

- **Evaluation studies**: Including PHyTeR v1 and v2, around 70 students have interacted with PHyTeR. However, we were able to use the data of only 46 students. All the studies that we conducted had single group pre-post design. We have evaluated PHyTeR v2 with 25 students. Although these students are from the target population, we discern that it is insufficient to make statistically significant claims. Nonetheless, we want to emphasize that the claims we make in this thesis are important for the domain of teaching troubleshooting and can be used by different cohorts for various purposes.

- **Learning Environment**: The current version of PHyTeR has one troubleshooting scenario. Many scaffolds that are provided in this scenario will have to carefully reduced while adding more scenarios. For example, we provide readymade hypotheses that can be constructed from a list. This was a conscious decision to reduce the complexity. In new scenarios, students can be provided opportunity to create their own hypothesis from scratch. The scenarios also need to be made progressively complex to imitate real-world troubleshooting settings.

Apart from the above listed limitations, we note that the duration of intervention is probably low for making very strong/longitudinal claims. The students interacted with PHyTeR for 1-4 hours in these experiments. These experiments are insufficient to make claims about their long-term changes in the understanding and performance of troubleshooting. We believe that more practice problems with single and multiple errors will aid the students.

9.5 Generalizability

We started this thesis with a broad goal to improve network troubleshooting skills among computer science undergraduates. We developed PHyTeR pedagogy by synthesizing literature from various domains. We conducted studies to identify student difficulties and to evaluate two version of PHyTeR. In this section, we elaborate the generalizability of these studies along various dimensions.

- **Other network troubleshooting problems:** The student difficulties that we have investigated and the troubleshooting scenario in PHyTeR are all based on problems having single error. Device connectivity and configuration faults are typically encountered. Hence, this thesis has focussed on such faults. Faults such as persistent packet drops indicate sub optimal sizing and design of the network. These are complex for novices to acquire and hence out of scope. However, the ideas of comparison between ideal and faulty behaviour would still be applicable in the case of protocols. The hypothesis might not be of the form: “There is some error in <device> and <configuration>”. Also, separating the faulty part here becomes not the physical area but the part of protocol that is not as expected

  We construe that the process of troubleshooting described in this thesis will be useful in contexts including other networking concepts and multiple errors. We argue that this pedagogy is applicable for other concepts in networking like firewalls and services provided by a server. In the context of problems where multiple errors occur in different parts of the network, some scaffolding might be required to help students split the network into multiple functional and non-functional parts in addition to the existing PSU activities. This would lead to generation of one or more hypothesis followed by designing a test that that is aligned with the hypothesis. More prompts might be needed to interpret the results in the context of a hypothesis, a sub-network and the broad troubleshooting problem.

- **Troubleshooting problems in other domains:** The domain of computer networking deals with the structure, function and behaviour of various devices. Domains like embedded systems, operating systems, microcontrollers etc. can be thought of as having devices with defined structure, function and behaviour. We speculate that PHyTeR pedagogy will be helpful in troubleshooting contexts in areas similar to computer network. The PSU activities that focus on comparison between ideal and faulty behaviour holds water in these contexts too. We have analysed the troubleshooting literature in the domains of mechanical and electrical systems. We know that understanding of these systems in terms of its structure, function and behaviour is important for troubleshooting in those domains. Hence, we deduce that the PHyTeR pedagogy can be utilized in these domains.

9.6 Guidelines to researchers and teachers

During the period of this research, the researcher had opportunities to discuss different aspects of the research problem with experts in the field. We did detailed literature review, conducted experiments and analysed the results and reflected on the same. We came up with different pilot versions of the intervention and tried to evaluate it. In this section, we synthesize the few learning from such attempts that can be useful to teachers and other researchers in supporting network troubleshooting.

- Familiarise the students with the domain and the process of troubleshooting: The domains of both networking and troubleshooting are complex, consisting of tangible structures and abstract processes. The students require some practice to get accustomed with these processes and terminologies used. We suggest the intervention developers to have explicit activities that familiarize the students with these processes and terminologies that clearly set the expectations from students. In the context of PHyTeR, students reported that both initial familiarization activity and the phase videos were helpful in understanding 'what they were supposed to do'. We observed them going back and forth between the phase videos and the phase specific tasks. Along with the familiarisation activities, we found that providing structure to the tasks helped them in identifying the troubleshooting process. A teacher can create a worksheet for this purpose. A TELE designer can provide this structure in various forms recommended by various researchers (Hmelo-Silver et al., 2007; M. C. Kim & Hannafin, 2011; Sharma & Hannafin, 2007).

- Provide opportunities to analyse ideal and faulty behaviour: The students have difficulty in imagining/reproducing the ideal dynamic behaviour of a network. It is important to support the students to reproduce the system behaviours. Teachers can use question prompts like ‘How will the router work?’, ‘What will the server do in this situation?’ even when there is no TELE to support students to focus on the behaviour of a network. A TELE can provide a richer set of options for this purpose including videos, interactive simulations, question prompts etc. In addition to this understanding, we saw that the students have difficulty in connecting the domain knowledge to the process of troubleshooting. Question prompts like 'Now that you have understood how a router would behave in this situation, do you think your router is behaving accordingly?', 'Now that you have identified that the router is behaving correctly, what do you want to investigate next?' etc. can be used to help students make this connection.

- Allow students to revisit their steps & decisions: A troubleshooting session usually consist of several iterations of hypothesis testing. Students find it difficult to keep track of the troubleshooting process. Teachers can provide structured worksheets that will enable students to go back and refer their troubleshooting history. A TELE can have a interactive version of this history, providing them different views and enabling them to zoom in and out of the iterations.

Apart from these, the researchers and teachers can directly use the list of difficulties and have scaffolds in their intervention to resolve them.

9.7 Summary

In this chapter, we gave an overview of the research conducted in this thesis. We discussed the research questions in the context of the broad goal. We elaborated on the limitations of various aspects of the thesis like learner attributes, duration of intervention to delineate the results of this thesis. We ended the chapter with speculations on the applicability of the results to related domain and contexts.

Chapter 10

Conclusion

In this chapter, we discuss the contributions of this thesis and possible future directions of research and development. We end this chapter with final reflections.

10.1 Contributions of this thesis

This thesis contributes to existing knowledge about the teaching - learning of troubleshooting skills in the domain of computer networks. Specifically, it augments the research related to the difficulties faced by students while troubleshooting and the development of technology enhanced learning environments for developing troubleshooting. The contributions are based on the analysis of the results of the research studies conducted as part of this thesis. We present the specific contributions below:

- **Characterisation of student difficulties:**

In study 1 conducted as a part of DBR cycle 1 of this thesis, we collected data to identify the difficulties of students while solving network troubleshooting problems. These difficulties are further categorized as themes corresponding to phases of troubleshooting and generic troubleshooting strategies.

  - **Who can benefit:**
    - Teachers who intend to develop network troubleshooting skills among their students can refer to these listed difficulties and create learning activities to alleviate these difficulties.
    - Developers who want to create learning environments to teach troubleshooting skills can be wary of these difficulties to design scaffolds and focussed activities in their learning environments.

- **PHyTeR Pedagogy and the learning activities**

Based on the review of literature and identified student difficulties, we proposed a pedagogy designed to help students in developing troubleshooting skills. This pedagogy is based on the theory of science inquiry learning, scaffolding and metacognition. It is structured along the phases of troubleshooting and a reflection activity. After coming up with the pedagogy of PHyTeR, in this thesis, we have materialized the components of pedagogy into technological features. For example, the pedagogy recommends a learning activity to compare the ideal and faulty states of the network. In PHyTeR learning environment, this is done by giving the students a set of statements. The students have to drag these statements from the pool and drop it into an 'ideal' or a 'faulty' bag.

- **Who can benefit:**
  - Teachers can use the pedagogy and create more troubleshooting scenarios for their students following the same structure.
  - Researchers who want to create TELEs in related domains can use some scaffolds of PHyTeR and modify them as appropriate to their context. Researchers working with scaffolding and its effect on various parameters can use the features in PHyTeR TELE in their studies, for example, to investigate the information foraging behaviour of students while using these scaffolds to solve a troubleshooting scenario.

- **PHyTeR TELE**

Finally, the learning environment PHyTeR can be used by students and teachers directly. More details about PHyTeR can be accessed here. It is recommended to use the Cisco Packet Tracer simulator along with this.

- **Who can benefit:**
  - Students with an introductory knowledge of computer networking concepts can create an account in PHyTeR and interact with the learning environment.
  - Teachers can ask their students to interact with this TELE as a part of their lab-work or assignment.

10.2 Future work

In this thesis, we started with a broad goal of developing TELE to teach troubleshooting skills in the domain of computer network. We followed the methodology of design-based research to develop the intervention and evaluated it. We have identified key features of PHyTeR that helped student to reach a desirable understanding about troubleshooting. Now we note the possible future directions this research can take:

- **Large scale evaluation of PHyTeR:** In this thesis, we investigated qualitative questions like what difficulties do students face, what are the changes in their understanding of troubleshooting, how they use features in PHyTeR. These questions helped us to understand if we are travelling in the desired direction. However, these questions are not sufficient to answer how much have we travelled in the desired direction. That is, we want to evaluate student learning of troubleshooting against the rubrics developed. In thesis, we identified the aspects of teaching troubleshooting skills that will help students. By doing large-scale studies, we want to investigate questions like which amongst the identified aspects are most helpful for students. Possible research questions for the large scale study:

    - RQ 1: What is the effect of PHyTeR on student network troubleshooting skills?
    - RQ 2: What is the perception of students on the usefulness of PHyTeR?

      We plan to answer RQ1 by doing a controlled experimental study. RQ2 can be answered by handing out a survey to the experimental group. Answering these questions would help in making stronger claims related to the pedagogy and implementation of PHyTeR

- **Developing instructor interface for PHyTeR:** At present PHyTeR has one troubleshooting scenario. Teachers in their classroom can use this for one session. However, it would help the teachers more if they can add their scenarios to PHyTeR. These scenarios can provide more practice to the students. The teachers can benefit from this by adding scenarios that are appropriate to their context. Adding more scenarios to PHyTeR required more than adding just the code. Some of the scaffolds and prompts in PHyTeR are customized to the current scenario. We need to provide interface to teachers to add these customized scaffolds for the problems.

- **Mining student actions in PHyTeR:** We have collected students' interaction logs with PHyTeR. We expect to collect more log data in the large-scale study. We intend to analyse this log data to answer the following research questions:

    - RQ 1: What are the learning paths followed by students while interacting with PHyTeR?

Answering this question will help us in identifying the desirable learning paths. These can be identified by considering learning paths of students who were able to solve the scenario, students solved it in less time, students who solved it using minimum number of hypothesis etc. We can later suggest these pathways to students in PHyTeR.

- RQ 2: How do students use the scaffolds provided by PHyTeR during troubleshooting?

  We have answered the questions "How do students use the features in PHyTeR during troubleshooting" using qualitative analysis in this thesis. Answering the above question using student log data will help us in identifying the places where students found the scaffolds useful and the places where it became a distraction.

- **Understanding the role of affective parameters on student learning of troubleshooting skills:** Troubleshooting in real-life can be time consuming and frustrating sometimes. We speculate that interest and motivation also play a role in the learning and performance of troubleshooting. Along with the knowledge of the structured troubleshooting process, one needs to manage these affective states. In study 1, we identified some affective parameters exhibited by students during troubleshooting. However, current pedagogy of PHyTeR does not accommodate the affective parameters in depth. We have familiarization activity to make the students comfortable with the context. We speculate that we need to conduct a study to understand affective parameters in detail and add affordances in PHyTeR to manage these.

- **Including scenarios that contain multiple errors:** The current version of PHyTeR has scenario with a single error. We have some conjectures about the PHyTeR pedagogy applied to scenarios to with multiple errors as explained in section 9.5. We want to incorporate additional features in PHyTeR and evaluate PHyTeR for its effectiveness in case of scenarios with multiple errors.

- **Alternative lenses to understand troubleshooting:** We considered a context where students worked with a self-paced system. We chose this context to focus on a structured process of troubleshooting. However, we recognise that troubleshooting rarely happens in such an isolated system in real life. Two important aspects of real life troubleshooting are:

- A troubleshooter has to gather information of various types from multiple sources
- A troubleshooter often works in a team to resolve a fault

We want to explore the effect of PHyTeR pedagogy in contexts including these real life complexities.

10.3 Final Reflections

I started this research with a broad objective of fostering troubleshooting skills among CS undergraduates. The main driving force during this journey has been two-fold: the prospect of developing an intervention that students can use and the intellectual stimulus I got from doing various research related activities. During the course of this research, I got opportunities to act and ruminate on various skills: to understand a complex phenomenon by synthesizing literature, to cogitate about learning of troubleshooting, to create a learning environment that aims to teach a complex phenomenon, to design and conduct experiments to evaluate the learning environment and to systematically communicate each of these. Each of this was possible because of my colleagues at my research lab, making me realize that the journey of scientific inquiry is impossible without supportive companions. In addition, it made me realize that I have just scratched the surface of research related to the teaching learning of troubleshooting and that there is a long way ahead. I intend to use the learning from this experience to tackle complex, interesting educational problems.

Appendix I
Assessment of troubleshooting

The existing literature on troubleshooting gave us some pointers related to assessing troubleshooting. We gather that time taken to troubleshoot is the most widely used metric to assess the success of troubleshooting, especially in real-time and critical systems. Sometimes it is also measured based on the non-repetition of the error after it was resolved. These measures are necessary but are very broad measures of troubleshooting. They are not sufficient to measure if a student is learning to troubleshoot in a structured manner. They are not formative measures – they don't give the details of 'improvement' in a student from one problem to another and they do not inform about the difficulties a student might have.

We came up with a rubric to assess troubleshooting that helps teachers to give formative feedback. The table below describes the rubric:

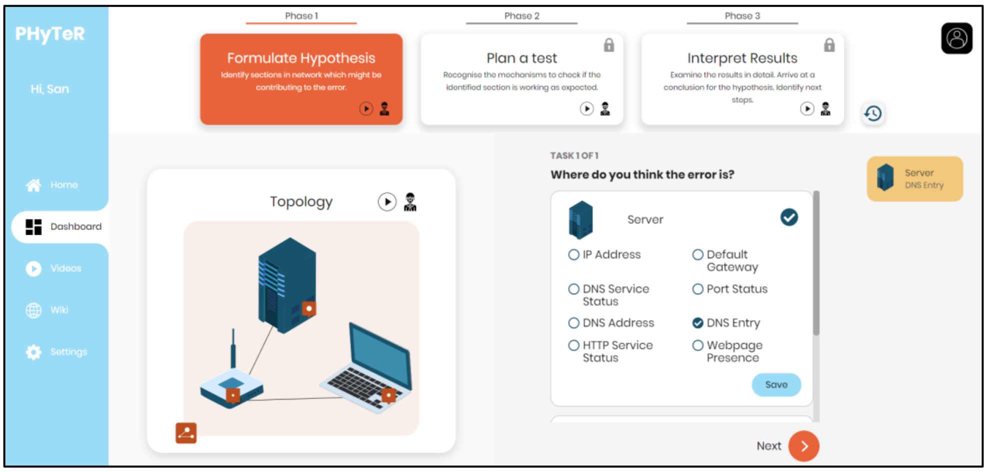
		Assessment		
Phases	Goals	High	Medium	Low
PSU	Identifying the relevant behavioural, function & structural elements at different levels and inter-relationships between them	Identifies all the structural and functional components relevant to the given faulty scenario	Identifies some of the relevant structural and functional components	Identifies none
	Distinguishing faulty behaviour from the ideal behaviour	Identifies most of the ideal behaviour	Identifies some of the ideal behaviour	Identifies hardly any ideal behaviour
		Identifies most of the faulty	Identifies some of the	Identifies hardly any

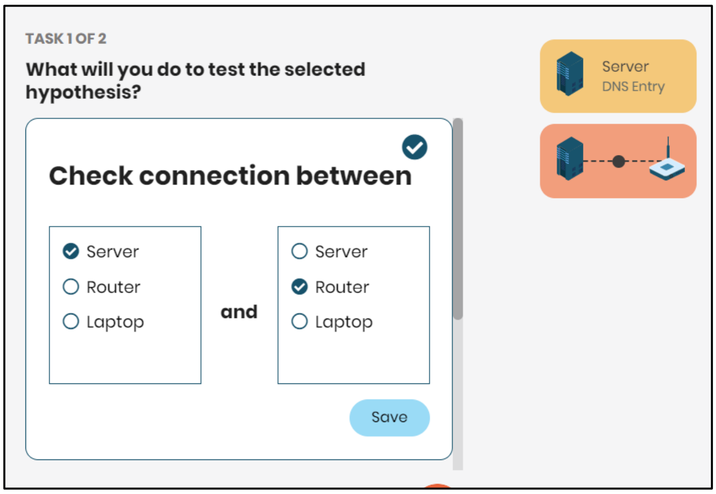
		behaviour correctly	faulty behaviour	faulty behaviour
		Identifies all the differences between ideal and faulty behaviour, in a actionable way	Identifies some, not actionable	Identifies none or not actionable
HG	Identifying a device/part of network to be behaving erroneously	Identifies the device and component	Identifies only the device/ generates 2 testable hypothesis	generates one hypothesis/ generates vague hypothesis that are not testable
	Relates the hypothesis to the broader problem	Provides apt justification for selection	Provides some logical justification	Provides no justification
HT	Constructing a test that is appropriate for the hypothesis selected	It is possible to accept/reject the hypo thesis based on the test	It is possible to decide if the device is faulty/ not faulty based on the test	It is not possible to conclude anything about the device based on the result of the test
	Predicting at different levels	Predicts the correct expected result	-	Predicts incorrect expected result
	Conducting the test	Considers all aspects of testing	-	Is able to follow

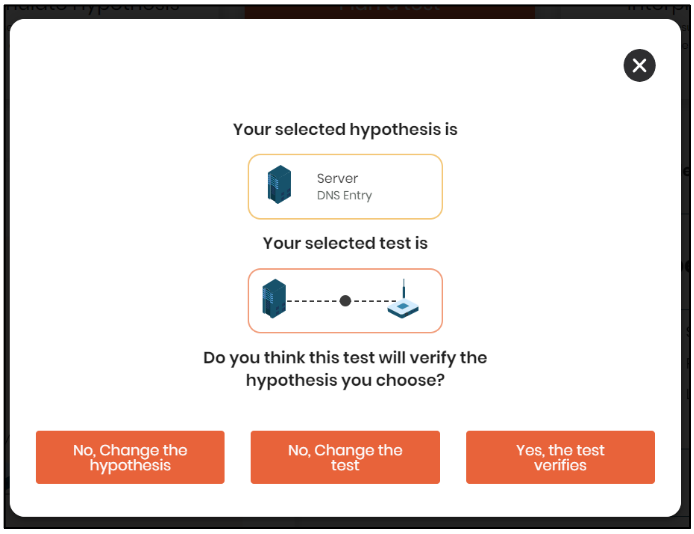
		- dependencies, changes to be made Identifies the correct instruments for testing and is able to handle those instruments		instructions to conduct a test
RI	Concluding the hypothesis testing	Notes down correct observation	Notes down incorrect observation	Doesn’t note down any observation
		Correctly accepts/rejects the hypothesis based on the test result	Incorrect conclusion based on the test result	No conclusion
	Drawing inferences related to the broader problem	The inference is based on results and lead to the solution	The inference is based on result but not leading to solution OR The inference leads towards solution but not based on result	The inference is not based on result and do not lead towards the solution

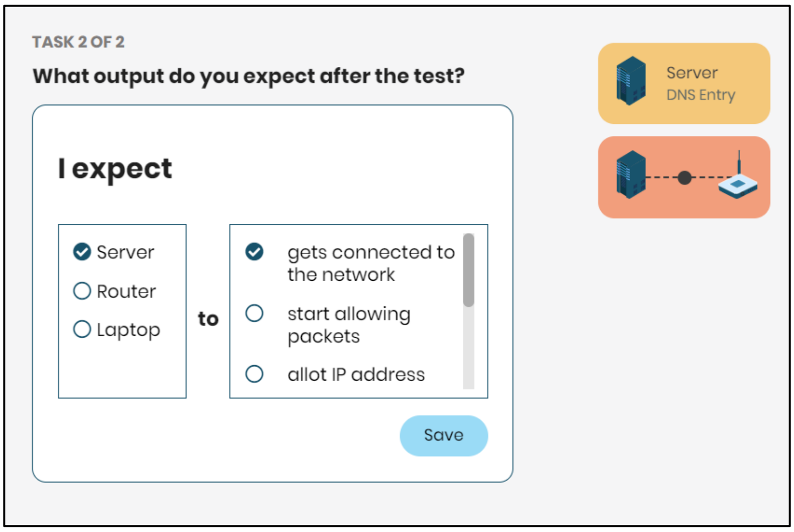
Appendix II

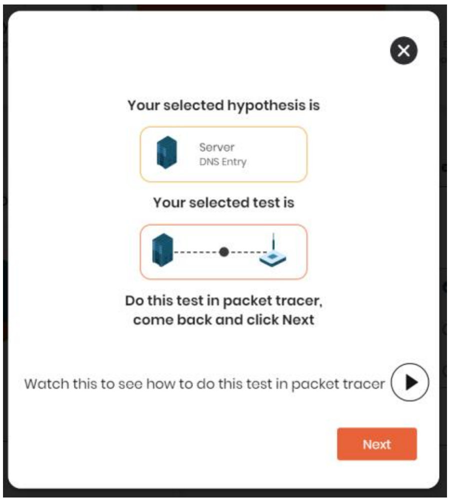
PHyTeR v2 Screenshots

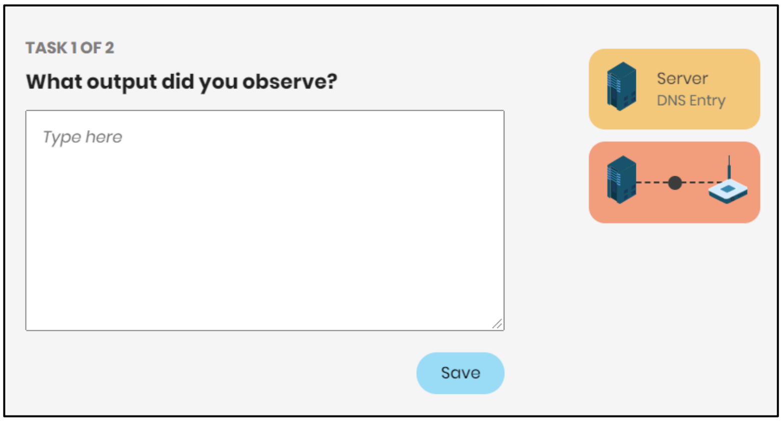


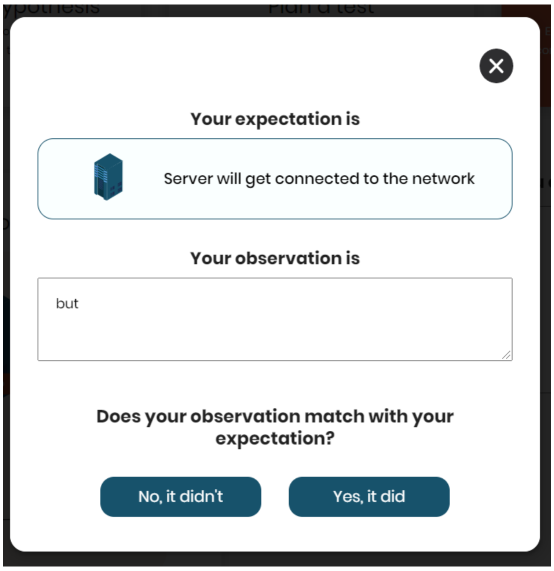


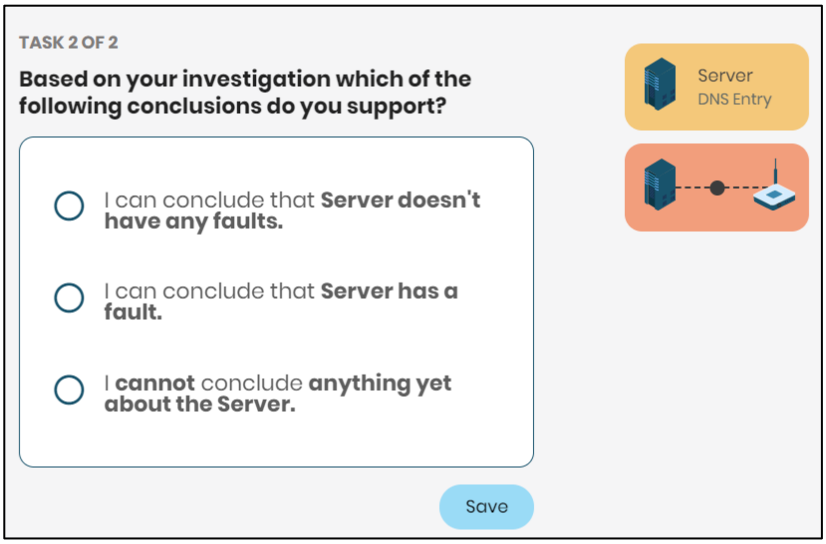


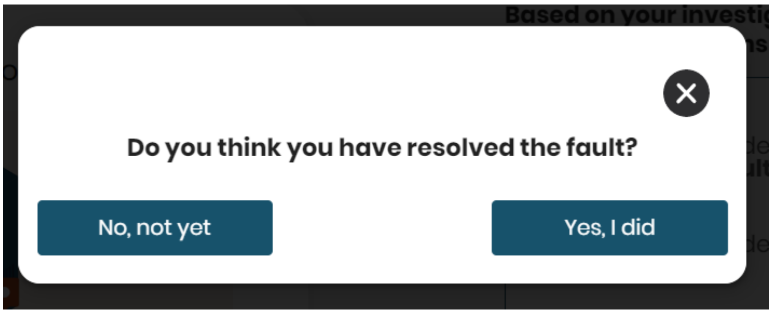

TASK 2 OF 2
What output do you expect after the test?
I expect
Server
Router
Laptop
to
gets connected to the network
start allowing packets
allot IP address
Save
Server
DNS Entry

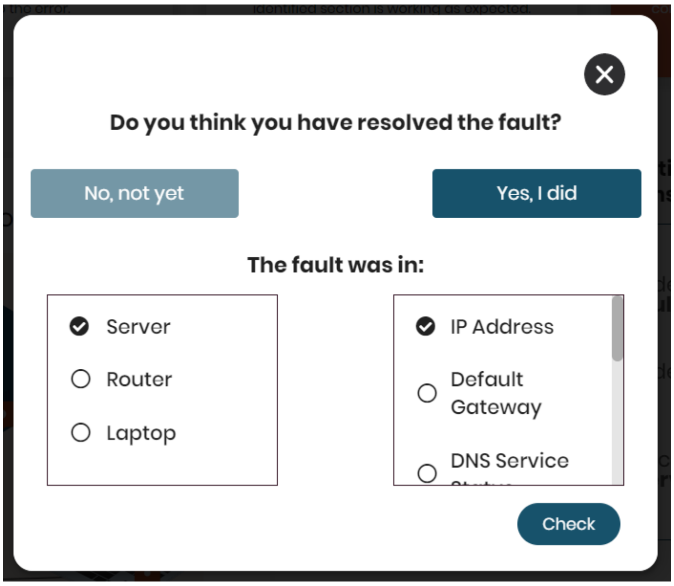

Your selected hypothesis is
Server
DNS Entry
Your selected test is
Do this test in packet tracer, come back and click Next
Watch this to see how to do this test in packet tracer
Next

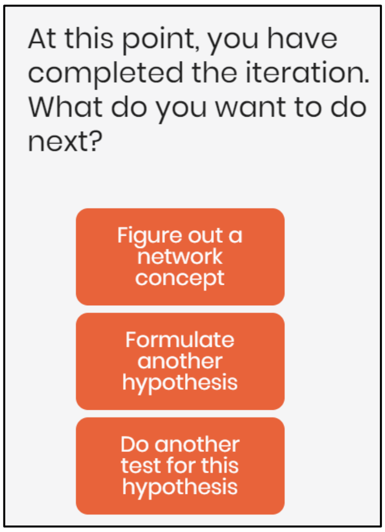

TASK 1 OF 2
What output did you observe?
Type here
Save
Server
DNS Entry

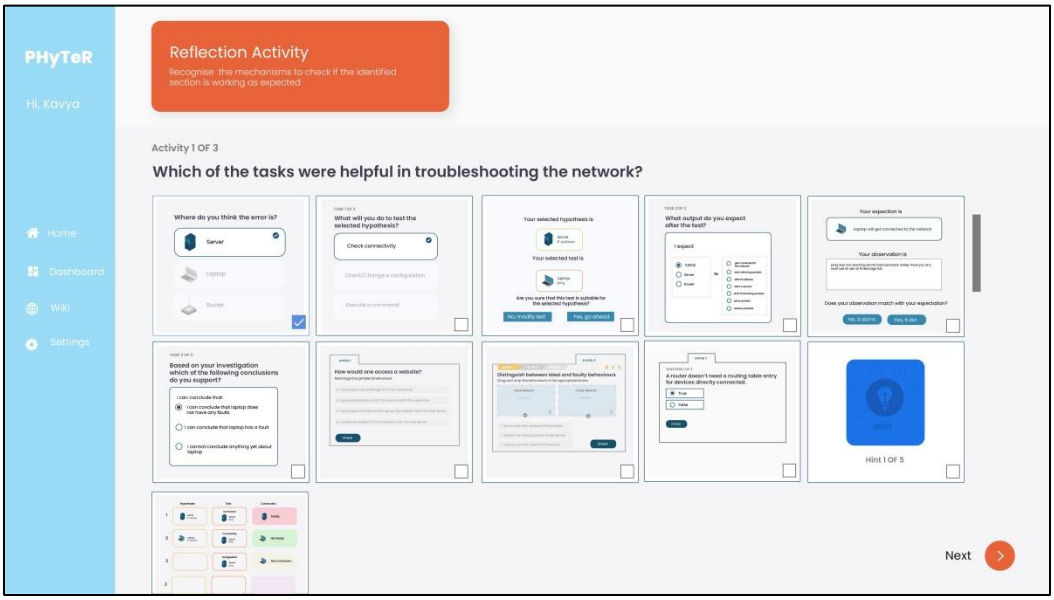

Your expectation is
Server will get connected to the network
Your observation is
but
Does your observation match with your expectation?
No, it didn't
Yes, it did

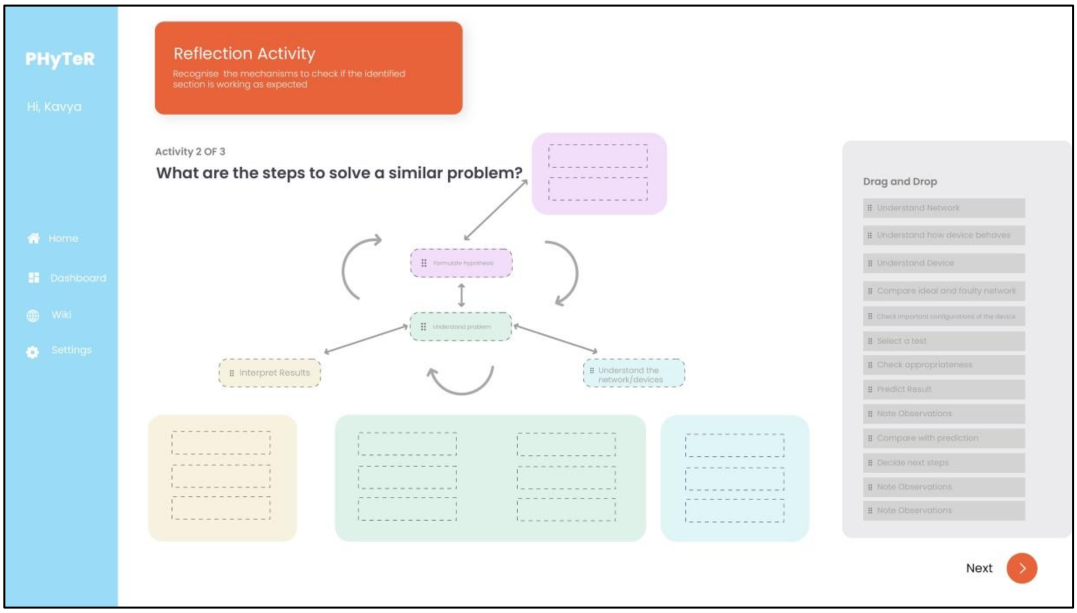

TASK 2 OF 2
Based on your investigation which of the following conclusions do you support?
I can conclude that Server doesn't have any faults.
I can conclude that Server has a fault.
I cannot conclude anything yet about the Server.
Save
Server
DNS Entry


Do you think you have resolved the fault?
No, not yet
Yes, I did


Do you think you have resolved the fault?
No, not yet
Yes, I did
The fault was in:
Server
Router
Laptop
IP Address
Default Gateway
DNS Service
Check


At this point, you have completed the iteration. What do you want to do next?
Figure out a network concept
Formulate another hypothesis
Do another test for this hypothesis


PHyTeR
Hi, Kavya
Reflection Activity
Recognise the mechanisms to check if the identified section is working as expected
Activity 1 OF 3
Which of the tasks were helpful in troubleshooting the network?
Home
Dashboard
Wiki
Settings
Next


PHyTeR
Hi, Kavya
Reflection Activity
Recognise the mechanisms to check if the identified section is working as expected
Activity 2 OF 3
What are the steps to solve a similar problem?
Home
Dashboard
Wiki
Settings
Interpret Results
Drag and Drop
Next

Appendix III

Participant Consent Form

STUDY TITLE: Study of student use of PHyTeR technology-enhanced learning environment

You have been asked to participate in a research study conducted by Kavya Alse from the Inter-Disciplinary Program in Educational Technology at the Indian Institute of Technology Bombay (IITB). The purpose of the study is to understand how students use PHyTeR to troubleshoot a network. The results of this study will be included in the Ph.D. thesis of Kavya Alse. You were selected as a possible participant in this study because you are a Computer Science undergraduate.

You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- In this study you will be asked to network troubleshooting problems using the PHyTeR technology-enhanced learning environment.

- Your solutions will be used for research purposes only by the investigators of this study.

- Participating in this research study is voluntary. You have the right not to answer any question, and to stop your participation in the study at any time. We expect that the study will take ~2.5 hours.

- You will not be compensated for the participation.

- We will not use your name in publications; however we may need to use your academic qualification details if you give us permission.

- We would like to record the audio of your interview so that we can use it for reference while proceeding with this study. If you grant permission for this interview to be recorded, you have the right to revoke recording permission and/or end your participation at any time. If we use your voice anywhere it will not be identified by name.

- We would like to capture your computer screen using CamStudio software as you solve the problems so that we can use it for reference while proceeding with this study. If you grant permission for this screen capture, you have the right to revoke recording permission and/or end your participation at any time. If we use this screen capture anywhere, we will not blank out your personal information.

I give permission for the following information to be included in publications resulting from this study (Please check all that apply):

□ My academic qualification details
□ Direct quotes from my audio recordings
□ Screenshots from my computer screen

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

Please contact Kavya Alse (kavyaalse@iitb.ac.in) or Prof. Sridhar Iyer, IDP in ET IITB (sri@iitb.ac.in) with any questions or concerns.
Name: ___________________
Date: ____________________

Appendix IV
Question Paper used in Study3

Answer the following questions in the context of the problem you solved. There is an example answer given with each question. These example answers are given in the context of the following example: You are trying to stream a video from an application. The video plays for 4 seconds and then stops playing. A loading circle appears on screen. You are trying to troubleshoot.

The example answers are meant to give you an idea of what details we are expecting from your answers. Be honest and write your answers in as much detail as possible. There is no time limit.

1. Explain the working of the faulty network. Mention what is working as expected and what is not working as expected.

Example answer: I opened the streaming service. I clicked on the video “abcd”. Player window opened. The video started playing. It played for 4 seconds. All these were expected behaviours. Then the loading circle appeared. It didn’t go away even after a long time. This was unexpected. The video should have played without any glitches.

2. List out all the possible reasons/causes you think resulted in the fault in the network

Example answer:

- Possible causes I thought of:
- There is network connectivity issue
- RAM is overloaded
- I have been logged out of the streaming app
- There is some issue with streaming service
- There is some issue with that particular video

3. How did you verify the reasons listed in question 2?

Example answer:

For cause #2, this is how I verified: RAM is overloaded. I used another app to flush out unwanted things from RAM. I also cleared local storage for the streaming app. Then tried to play the video. But the issue with playing persisted.

For cause #3: I have been logged out. I tried logging in…

4. How did you arrive at your final answer? Describe the process that you undertook to arrive at the final answer. Write about any strategy, plan that you followed.

Example answer:

I considered the possible causes one by one and verified if it was the actual cause for the error… I selected the one that was easy for me to verify…

Appendix V

Questionnaires used in Study4

Pre-questionnaire

Understanding troubleshooting

In this form, there are some questions that will help you to elicit your understanding about troubleshooting in general and network troubleshooting in specific.

There is no one 'correct' answer for these questions. But you need to give answers according to your approach. So DO NOT google for answers to these questions or DO NOT ask anyone else for answers to these questions.

These questions are meant to help you with the activities that follow. So BE HONEST in answering these questions.

Feel free to ask any doubts you have to the session coordinator.

*Required

Email *

Your email address

Name *

Your answer

What do you understand by the term 'troubleshooting'? *

Write in as much detail as possible. You can also explain it using an example.

Your answer

Have you done any network troubleshooting before? *

This includes troubleshooting a real network or a simulated network.

- ◯ Yes
- ◯ No
- ◯ I'm not sure what network troubleshooting constitutes

Understanding troubleshooting

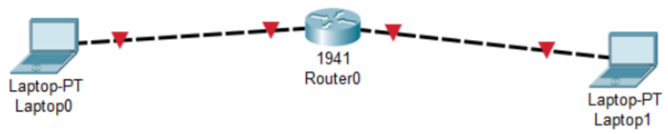
*Required

How will you approach?

In the following sections, you are asked some questions related to a network troubleshooting scenario. Answer those questions based on what you are most likely to do in that scenario.

Scenario: Below is a network with 2 laptops connected using a router. You are trying to ping (send packets) from Laptop0 to Laptop1. But the ping is failing.

Scenario topology with 2 laptops connected via a router



How will you approach troubleshooting this scenario? *

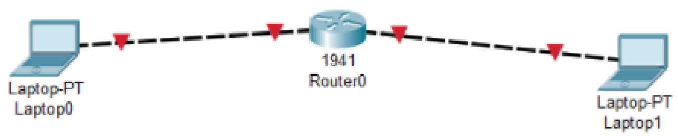
Note that you are asked about your approach and not what the error would be. There is no one correct 'approach' to troubleshoot. So write about how will you try solving this problem. Explain what you would think and do while trying to troubleshoot. It will help you in the further activities if you try to answer this on your own. So be honest and don't refer anything else to answer this. Also, write your answer in a step-by-step manner. For example: 1. I will google "abcdef". 2. I'll try to do "xyz" with router 3. I think there is an error in laptop0 because "klmn" etc.

Your answer

What will you do?

In this section there are some specific questions related to the same scenario. Answer these questions based on what you are most likely to do.

Scenario: Below is a network with 2 laptops connected using a router. You are trying to ping (send packets) from Laptop0 to Laptop1. But the ping is failing.



If you have to search for information about this scenario, what information will you search for and in which order?

The rows correspond to the information search you might do. Select only those searches you would do and leave blank for other rows. The columns correspond to the order in which you look for that information. If the first thing you search is "how to send packets from one laptop to another via a router", tick column 1 in the row corresponding to that.

	1	2	3	4	5	6
I will search for "how to setup a network with 2 laptops connected via a router"	☐	☐	☐	☐	☐	☐
I will search for "how does a router connect 2 laptops"	☐	☐	☐	☐	☐	☐
I will search for "how to send packets from one laptop to another via a router"	☐	☐	☐	☐	☐	☐
I will search for "most frequent errors in a network with 2 laptops and a router"	☐	☐	☐	☐	☐	☐
I will search for "ping failing from one laptop to another"	☐	☐	☐	☐	☐	☐
I will copy and paste the error message	☐	☐	☐	☐	☐	☐

If you want to search for more information, what that would be? *

Your answer

While troubleshooting the above scenario, which of the below mentioned actions would you do and in which order?

The rows correspond to actions that you might do while troubleshooting. Note that all actions are not mandatory. Select only those which you are more likely to do. The columns represent the order in which you would do them.

	1	2	3	4	5	6
I'll think of possible causes for the error	☐	☐	☐	☐	☐	☐
I'll eliminate the possible causes one by one	☐	☐	☐	☐	☐	☐
I'll consider each device in the network and think how that device could cause the error	☐	☐	☐	☐	☐	☐
I'll 'run' the network and see where the error is	☐	☐	☐	☐	☐	☐
I'll consider one possible cause and check if it is the actual cause for error	☐	☐	☐	☐	☐	☐
I'll search for some information	☐	☐	☐	☐	☐	☐

Are there any actions that would do while troubleshooting the above scenario which are missing in the list above? Or are there actions which you would do differently? Write those actions here. *

Your answer

Do you expect to repeat any of the above action multiple times while troubleshooting the above scenario? Write the actions that you would do again and again and why? *

Your answer

Understanding troubleshooting

In the above scenario, if you want to start investigating each device, which is the first device you will choose to investigate and why? *

Write the device name and reason in the following format: Laptop0, because "lorem ipsum ..."

Your answer

Write at least 3 probable causes for the error in the above scenario *

Your answer

Which among the following do you think a network troubleshooter should know? Assign priorities.

The rows correspond to various types of information. Select only those which you think are important for a network troubleshooter. The columns correspond to priorities: Select column 1 for the most important information and column 2 for next most important and so on..

	1	2	3	4
Information about the error to be resolved	☐	☐	☐	☐
Practical knowledge about the configurations and operations of a network	☐	☐	☐	☐
Theoretical information about the devices in the network	☐	☐	☐	☐
Information about a systematic process of troubleshooting a network	☐	☐	☐	☐

Rate the approach

1 indicates "Not confident at all", 5 indicates "Absolutely confident". If a specific term or task is totally unfamiliar to you, please mark 1

Given a troubleshooting scenario like the one mentioned above, rate the following statements depending on your ability to do the indicated things

I can describe the ideal working of the network (I can explain how the given network works when there is no error) *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

I can identify the difference between working of ideal and given faulty network *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

I can identify the ideal behavior (behavior in case of no error) of a network device in the given network *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

I can list one or more probable causes for the error *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

I can systematically examine each cause listed to check if it is the actual cause *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

I can decide which cause to examine next based on my examination *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Post – Questionnaire

Understanding troubleshooting

In this form, there are some questions that will help you to elicit your understanding about troubleshooting in general and network troubleshooting in specific.

There is no one 'correct' answer for these questions. But you need to give answers according to your approach. So DO NOT google for answers to these questions or DO NOT ask anyone else for answers to these questions.

These questions are meant to help you with the activities that follow. So BE HONEST in answering these questions.

Feel free to ask any doubts you have to the session coordinator.

*Required

Email *

Your email address

Name *

Your answer

What do you now understand by the term 'troubleshooting'? *

Write in as much detail as possible. You can also explain it using an example.

Your answer

Feedback about usability of PHyTeR

In the upcoming sections we will be asking your feedback about PHyTeR. Do provide us your honest feedback and do not hesitate to point out shortcomings.

I think that I would like to use PHyTeR frequently *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I found PHyTeR unnecessarily complex *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I thought PHyTeR was easy to use *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I think that I would need the support of a technical person to be able to use PHyTeR *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I found the various functions in PHyTeR were well integrated *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I thought there was too much inconsistency in PHyTeR *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I would imagine that most people would learn to use PHyTeR very quickly *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

I found PHyTeR very cumbersome to use *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I felt very confident using PHyTeR *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I needed to learn a lot of things before I could get going with PHyTeR *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate your confidence regarding your knowledge of the following

1 indicates "Not confident at all", 5 indicates "Absolutely confident". If a specific term or task is totally unfamiliar to you, please mark 1

Identify the ideal behavior of a given network *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Identify the faulty behavior of a given network *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Identify the difference between ideal and faulty network *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Identify the ideal behaviour of a network device *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Coming up with hypotheses *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Designing appropriate test for a given hypothesis *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Interpreting the result of a test *

	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

Reflecting on the process of troubleshooting *

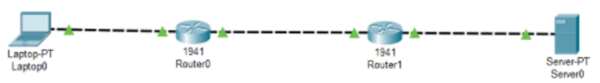
	1	2	3	4	5	
Not confident at all	○	○	○	○	○	Absolutely confident

How will you approach?

In the following sections, you are asked some questions related to a network troubleshooting scenario. Answer those questions based on what you are most likely to do in that scenario.

Scenario: Below is a network with a laptop, 2 routers and a server. You are trying to ping (send packets) from Laptop0 to Server0. You see that the ping is failing.

Topology of the scenario



How will you approach troubleshooting this scenario? *

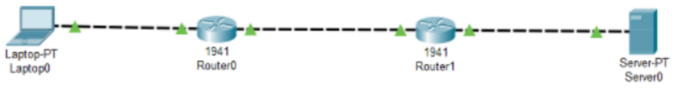
Note that you are asked about your approach and not what the error would be. There is no one correct 'approach' to troubleshoot. So write about how will you try solving this problem. Explain what you would think and do while trying to troubleshoot. It will help you in the further activities if you try to answer this on your own. So be honest and don't refer anything else to answer this. Also, write your answer in a step-by-step manner. For example: 1. I will google "abcdef". 2. I'll try to do "xyz" with router 3. I think there is an error in laptop0 because "klmn" etc.

Your answer

What will you do?

In this section there are some specific questions related to the same scenario. Answer these questions based on what you are most likely to do.

Below is a network with a laptop, 2 routers and a server. You are trying to ping (send packets) from Laptop0 to Server0. You see that the ping is failing.



If you have to search for information about this scenario, what information will you search for and in which order?

The rows correspond to the information search you might do. Select only those searches you would do and leave blank for other rows. The columns correspond to the order in which you look for that information. If the first thing you search is "how to send packets from one laptop to another via a router", tick column 1 in the row corresponding to that.

	1	2	3	4	5	6	7
I will search for "how to send packets from one laptop to server via 2 routers"	☐	☐	☐	☐	☐	☐	☐
I will search for "how to setup a network with a laptop connected to a server via a router"	☐	☐	☐	☐	☐	☐	☐
I will search for "most frequent errors in a network with laptop, 2 routers and a server"'	☐	☐	☐	☐	☐	☐	☐

I will search for "how does a router connects to a laptop and a server"	☐	☐	☐	☐	☐	☐	☐
I will copy and paste the error message	☐	☐	☐	☐	☐	☐	☐
I will search for "ping failing from laptop to server via 2 routers"	☐	☐	☐	☐	☐	☐	☐
I will search for "how does a router connect with another router"	☐	☐	☐	☐	☐	☐	☐

If you want to search for more information, what that would be? *

Your answer

While troubleshooting the above scenario, which of the below mentioned actions would you do and in which order?

The rows correspond to actions that you might do while troubleshooting. Note that all actions are not mandatory. Select only those which you are more likely to do. The columns represent the order in which you would do them.

	1	2	3	4	5	6
I'll 'run' the network and see where the error is	☐	☐	☐	☐	☐	☐
I'll eliminate the possible causes one by one	☐	☐	☐	☐	☐	☐
I'll think of possible causes for the error	☐	☐	☐	☐	☐	☐
I'll search for some information	☐	☐	☐	☐	☐	☐
I'll consider each device in the network and think how that device could cause the error	☐	☐	☐	☐	☐	☐
I'll consider one possible cause and check if it is the actual cause for error	☐	☐	☐	☐	☐	☐

Are there any actions that would do while troubleshooting the above scenario which are missing in the list above? Or are there actions which you would do differently? Write those actions here. *

Your answer

Do you expect to repeat any of the above action multiple times while troubleshooting the above scenario? Write the actions that you would do again and again and why? *

Your answer

Understanding Troubleshooting

In the above scenario, if you want to start investigating each device, which is the first device you will choose to investigate and why? *

Write the device name and reason in the following format: Laptop0, because "lorem ipsum ..."

.

Write at least 3 probable causes for the error in the above scenario *

.

Which among the following do you think a network troubleshooter should know? Assign priorities.

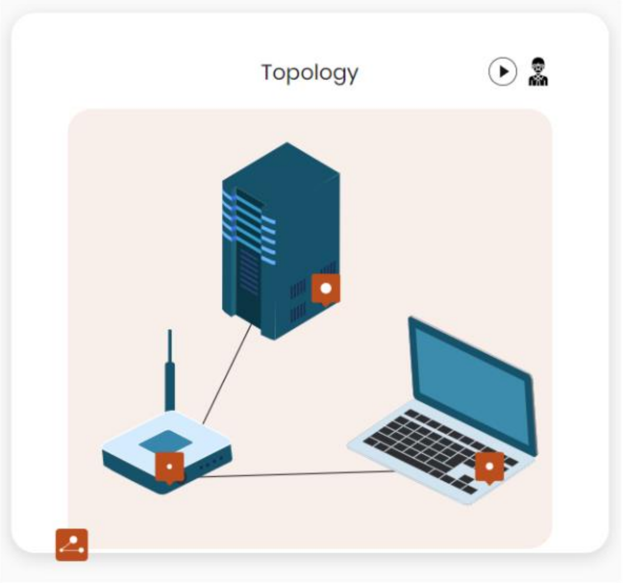
The rows correspond to various types of information. Select only those which you think are important for a network troubleshooter. The columns correspond to priorities: Select column 1 for the most important information and column 2 for next most important and so on..

	1	2	3	4
Information about a systematic process of troubleshooting a network	☐	☐	☐	☐
Practical knowledge about the configurations and operations of a network	☐	☐	☐	☐
Information about the error to be resolved	☐	☐	☐	☐
Theoretical information about the devices in the network	☐	☐	☐	☐

Phases of troubleshooting

In this section the questions are related to the phases of troubleshooting. Answer those based on your understanding of those phases.

Understanding the network and devices in PHyTeR



What do you understand by the term 'Understanding the network and devices'? *

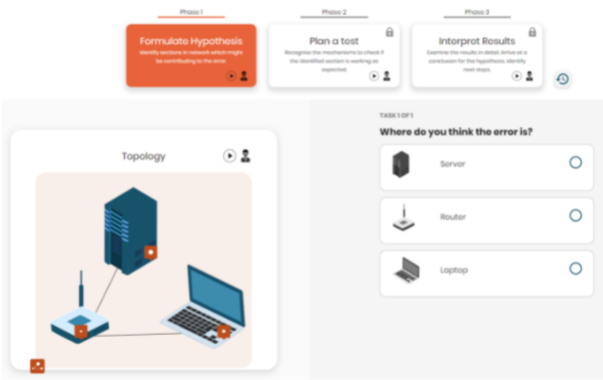
Your answer

What actions/searches would you do to "understand the network and device"? *

Your answer

Formulating Hypothesis

Formulating a hypothesis in PHyTeR



What do you understand by the term formulating hypothesis? *

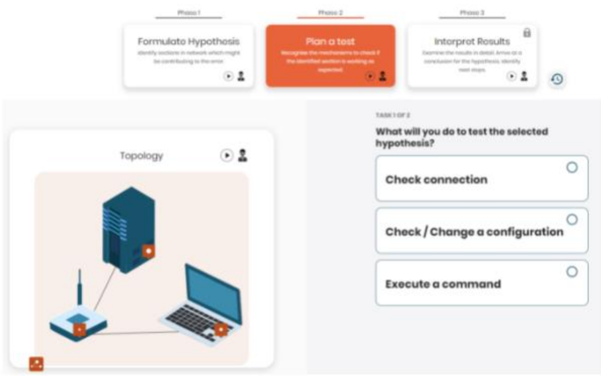
Your answer

What actions would you do to formulate a hypothesis? *

Your answer

Plan and run a test

Plan and run a test in PHyTeR



What do you understand by the term 'plan and run a test'? *

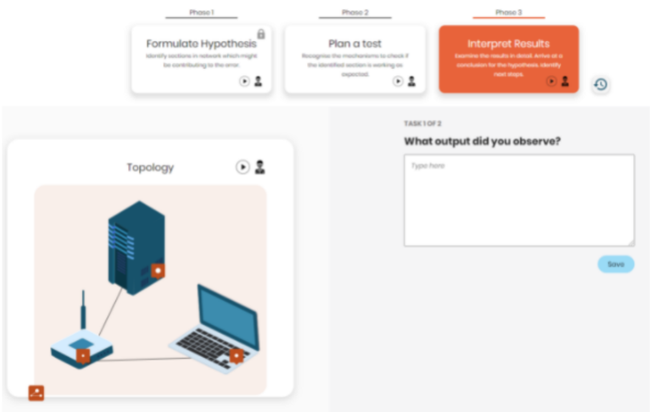
Your answer

What actions would you do to 'plan and run a test'? *

Your answer

Interpreting Results

Interpreting results in PHyTeR



What do you understand by the term interpret results'? *

Your answer

What actions would you do to 'interpret results'? *

Your answer

Reflections

Reflections in PHyTeR

What did you reflect on after troubleshooting the scenario? *

Your answer

Change in approach

Your understanding of the term 'troubleshooting' - Has it changed from previous understanding? If yes, why has it changed and how did it change? *

Your answer

Your approach towards troubleshooting a given scenario - Has the approach changed from previous? If yes, why has it changed and how did it change? *

Your answer

Detailed Feedback

One last section! This information will help us improve the workshop for future participants. Thank you so much for providing this feedback.

What are the main things you learned from the workshop? *

Your answer

What features of PHyTeR did you find most useful ? *

Your answer

What features of PHyTeR did you find challenging/ frustrating ? *

Your answer

Any suggestions to improve when we design the next online workshop for you. *

Your answer

References

Alby, F., & Zucchermaglio, C. (2009). Time, Narratives and Participation Frameworks in Software Troubleshooting. *Computer Supported Cooperative Work (CSCW)*, *18*(2–3), 129–146. https://doi.org/10.1007/s10606-008-9090-7

Arnold, J. C., Kremer, K., & Mayer, J. (2014). Understanding Students' Experiments-What kind of support do they need in inquiry tasks? *International Journal of Science Education*, *36*(16), 2719–2749. https://doi.org/10.1080/09500693.2014.930209

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples : Instructional Principles from the Worked Examples. *Review of Educational Research*, *70*(2), 181–214. Retrieved from https://www.jstor.org/stable/1170661

Axton, T., Doverspike, D., Park, S., & Barrett, G. (1997). A model of the information-processing and cognitive ability requirements for mechanical troubleshooting. *Int. J. Cogn. Ergon*. Retrieved from https://scholar.google.co.in/scholar?hl=en&q=A+model+of+the+information-processing+and+cognitive+ability+requirements+for+mechanical+troubleshooting&btnG=#0

Bannan-Ritland, B. (2003). The Role of Design in Research: The Integrative Learning Design Framework. *Educational Researcher*, *32*(1), 21–24. https://doi.org/10.3102/0013189X032001021

Berardi-Coletta, B., Buyer, L. S., Dominowski, R. L., & Rellinger, E. R. (1995). Metacognition and problem solving: A process-oriented approach. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(1), 205–223. Retrieved from https://doi.org/10.1037/0278-7393.21.1.205

Bereiter, C. (2002). Design research for sustained innovation. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, *9*(3), 321–327.

Besnard, D., & Bastien-Toniazzo, M. (1999). Expert error in trouble-shooting: an exploratory study in electronics. Retrieved August 20, 2015, from https://hal.archives-ouvertes.fr/hal-00691823/document

Blackburn, J. J. (2013). *Assessing the effects of cognitive style, hypothesis generation and problem complexity on the problem solving ability of school-based agricultural education students: An experimental study*.

Bogart, K. L. Van De, Dounas-frazer, D. R., Lewandowski, H. J., & Stetzer, M. R. (2015). The role of metacognition in troubleshooting : an example from electronics, 339–342. https://doi.org/10.1119/perc.2015.pr.080

Bogart, K. L. Van De, Dounas-frazer, D. R., Lewandowski, H. J., & Stetzer, M. R. (2017). Investigating the role of socially mediated metacognition during collaborative troubleshooting of electric circuits, *020116*, 1–19.

https://doi.org/10.1103/PhysRevPhysEducRes.13.020116

Bowen, J. L. (2006). Educational Strategies to Promote Clinical Diagnostic Reasoning, 2217–2225.

Braun, V., & Clarke, V. (2012). Thematic analysis. *APA Handbook of Research Methods in Psychology, Vol 2: Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological.*, *2*, 57–71. https://doi.org/10.1037/13620-004

Brown, A. L. (1992). Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. *Journal of the Learning Sciences*, *2*(2), 141–178. https://doi.org/10.1207/s15327809jls0202_2

Chi, M. T. H., De Leeuw, N., Chiu, M. H., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439–477. https://doi.org/10.1016/0364-0213(94)90016-7

Cohane, R. (2017). Financial Cost of Software Bugs. Retrieved from https://medium.com/@ryancohane/financial-cost-of-software-bugs-51b4d193f107

Collins, A. (2006). Cognitive Apprenticeship. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (Vol. 4, pp. 47–60). Cambridge University Press.

Collins, A., Brown, J. S., & Newman, S. E. (1987). *Cognitive Apprenticeship: Teaching the craft of reading, writing and mathematics*. *Centre for the study of Reading*.

Collins, A., Joseph, D., Bielaczyc, K., Collins, A., & Bielaczyc, K. (2004). Design Research : Theoretical and Methodological Issues Linked references are available on JSTOR for this article : Design Research : Theoretical and Methodological Issues. *The Journal of the Learning Sciences*, *13*(1), 15–42.

De Graaff, E., & Kolmos, A. (2003). Characteristics of Problem-Based Learning. *International Journal of Engineering Education*, *19*(5), 657–662.

De Jong, T. (2006). Scaffolds for Scientific Discovery Learning. In *Handling Complexity in Learning Environments: Theory and Research* (pp. 107–128).

De Jong, T., & Lazonder, A. W. (2005). The guided discovery principle in multimedia learning. *The Cambridge Handbook of Multimedia Learning*, 215–228.

Dounas-Frazer, D. R., Van De Bogart, K. L., Stetzer, M. R., & Lewandowski, H. J. (2016). Investigating the role of model-based reasoning while troubleshooting an electric circuit. *Physical Review Physics Education Research*, *12*(1), 1–20. https://doi.org/10.1103/PhysRevPhysEducRes.12.010137

Dunbar, K. (1998). Problem solving Problem solving. *A Companion to Cognitive Science*, (May), 289–298.

Eslinger, E., White, B., Frederiksen, J., & Brobst, J. (2008). Supporting inquiry processes with an interactive learning environment: Inquiry island. *Journal of Science Education and Technology*, *17*(6), 610–617. https://doi.org/10.1007/s10956-008-9130-6

Farnham-Diggory, S. (1994). Paradigms of Knowledge and Instruction. *Review of Educational Research*, *64*(3), 463–477. Retrieved from https://www.jstor.org/stable/1170679

Flesher, J. W. (1993). Context and Expertise: The Case of Electronic Troubleshooting. Retrieved from http://eric.ed.gov/?id=ED359264

Fosnot, C. T., & Perry, R. S. (1996). Constructivism: A Psychological Theory of Learning. *Constructivism: Theory, Perspectives, and Practices*, *2*(1), 8–33. Retrieved from http://rsperry.com/fosnotandperry.pdf

Ge, X., Law, V., & Huang, K. (2016). Detangling the interrelationships between self-regulation and ill-structured problem solving in problem-based learning. *Interdisciplinary Journal of Problem-Based Learning*, *10*(2), 11.

Gog, T. Van. (2006). *Uncovering the problem-solving process to design effective worked examples*.

Guzdial, M. (1994). Software-Realized Scaffolding to Facilitate Programming for Science Learning. *Interactive Learning Environments*. https://doi.org/10.1080/1049482940040101

Hall, E. P., Gott, S. P., & Pokorny, R. A. (1995). A PROCEDURAL GUIDE TO COGNITIVE, (January 1985).

Hartman, H. J. (1998). Metacognition in teaching and learning : An introduction, *26*(1/2), 1–3. Retrieved from https://www.jstor.org/stable/23371261

Hmelo-Silver, C. E. (2012). International Perspectives on Problem-based Learning: Contexts, Cultures, Challenges, and Adaptations. *Interdisciplinary Journal of Problem-Based Learning*, *6*(1). https://doi.org/10.7771/1541-5015.1310

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. a. (2007). Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, *42*(2), 99–107. https://doi.org/10.1080/00461520701263368

Hogan, K. (1999). Thinking aloud together: A test of an intervention to foster students' collaborative scientific reasoning. *Journal of Research in Science Teaching*, *36*(10), 1085–1109.

Hong, N. S. (1998). *The relationship between well-structured and ill-structured problem-solving in multimedia simulation*. The Pennsylvania State University. Retrieved from http://www.cet.edu/pdf/structure.pdf

Johnson, S. D. (1987). Knowledge and skill differences between expert and novice service technicians on technical troubleshooting tasks. Retrieved from http://eric.ed.gov/?id=ED290043

Johnson, S. D. (1995). Understanding Troubleshooting Styles To Improve Training Methods. Retrieved from http://eric.ed.gov/?id=ED389948

Johnson, S. D., & Chung, S.-P. (1999). The Effect of Thinking Aloud Pair Problem Solving

(TAPPS) on the Troubleshooting Ability of Aviation Technician Students. *Journal of Industrial Teacher Education*, *37*(1), 7–25. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=EJ596672&site=ehost-live

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology Research and Development*, *45*(1), 65–94. https://doi.org/10.1007/bf02299613

Jonassen, D. H. (2004). Learning to Solve Problems: An Instructional Design Guide, 253. Retrieved from http://21it.files.wordpress.com/2008/09/pfeiffer-learning-to-solve-problems-an-instructional-design-guide-2004-isbn0787964379.pdf%5Cnhttp://books.google.ca/books?id=g0ffeIYunUwC

Jonassen, D. H. (2010). *Learning to Solve Problems: A handbook for designing problem-solving learning environments*. *Routledge*. https://doi.org/10.1017/CBO9781107415324.004

Jonassen, D. H., & Hung, W. (2006). Learning to Troubleshoot: A New Theory-Based Design Architecture. *Educational Psychology Review*, *18*(1), 77–114. https://doi.org/10.1007/s10648-006-9001-8

Jonassen, D. H., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and Development*, *47*(1), 61–79. https://doi.org/10.1007/BF02299477

Jong, T. De, & van Joolingen, W. R. (1998). Scientific Discovery Learning with Computer Simulations of Conceptual Domains Author (s): Ton de Jong and Wouter R . van Joolingen Source : Review of Educational Research , Vol . 68 , No . 2 (Summer , 1998), pp . 179-201 Published by : American Educat. *Review of Educational Research*, *68*(2), 179–201.

Jutten, E., Schaafstal, A., & Pel, P. (1999). The Troubleshooter . The Acquisition of Troubleshooting Expertise in a Virtual Environment 2 . Training of Troubleshooting : A Structured Approach. In *World Conference on Educational Media and Technology* (pp. 765–768).

Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). The expertise reversal effect. *Educational Psychologist*, *38*(1), 23–31. https://doi.org/10.1207/S15326985EP3801_4

Kester, L., Kirschner, P. A., & van Merriënboer, J. J. G. (2004). Information presentation and troubleshooting in electrical circuits. *International Journal of Science Education*, *26*(2), 239–256. https://doi.org/10.1080/69032000072809

Kim, J. Y., & Kyu, Y. L. (2019). Promoting learning in online, ill-structured problem solving: The effects of scaffolding type and metacognition level. *Computers & Education*, (138), 116–129.

Kim, M. C., & Hannafin, M. J. (2011). Scaffolding problem solving in technology-enhanced learning environments (TELEs): Bridging research and theory with practice. *Computers and Education*, *56*(2), 403–417. https://doi.org/10.1016/j.compedu.2010.08.024

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why Minimal Guidance During Instruction Does Not Work. *Educational Psychologist*, *41*(March 2015), 87–98. https://doi.org/10.1207/s15326985ep4102

Kirsh, D. (2004). Metacognition , Distributed Cognition and Visual Design, 1–22.

Kirsh, D. (2010). Thinking with external representations, 441–454. https://doi.org/10.1007/s00146-010-0272-8

Kitchener, K. S. (1983). Cognition , Metacognition , and Epistemic Cognition : A Three-Level Model of Cognitive Processing. *Human Development*, *26*(4), 222–232.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, *12*(1), 1–48. https://doi.org/10.1016/0364-0213(88)90007-9

Klein, G. A. (1989). Recognition-primed decisions. In, WB Rouse (Ed.) Advances in man-machine systems research. Vol. 5. JAI Press: Greenwich CT.

Krajcik, J. S., & Blumenfeld, P. C. (2006). Project Based Learning (pp. 317–334).

Kuhn, D., Black, J., Keselman, A., Kaplan, D., Cognition, S., Kuhn, D., … Kaplan, D. (2000). The Development of Cognitive Skills to Support Inquiry Learning, *18*(4), 495–523. Retrieved from https://www.jstor.org/stable/3233891

Larkin, J., McDermott, J., Simon, D. P., & Simon, H. a. (1980). Expert and novice performance in solving physics problems. *Science (New York, N.Y.)*, *208*(4450), 1335–1342. https://doi.org/10.1126/science.208.4450.1335

Lazonder, A. W. (2014). Inquiry Learning. In *Handbook of Research on Educational Communications and Technology: Fourth Edition* (pp. 453–464). https://doi.org/10.1007/978-1-4614-3185-5_53

Lazonder, A. W., Hagemans, M. G., & Jong, T. De. (2010). Offering and discovering domain information in simulation-based inquiry learning. *Learning and Instruction*, *20*(6), 511–520. https://doi.org/10.1016/j.learninstruc.2009.08.001

Lesgold, A. (1988). SHERLOCK: A coached practice environment for an electronics troubleshooting job. https://doi.org/10.1017/CBO9781107415324.004

Lyle, J. (2003). Stimulated Recall: A report on its use in naturalistic research. *British Educational Research Journal*, *29*(6), 861–878. https://doi.org/10.1080/0141192032000137349

Macpherson, R. T. (1998). Factors Affecting Technological Trouble Shooting Skills. *Journal of Industrial Teacher Education*, *35*(4).

Maloney, D. P. (2011). *An overview of physics education research on problem solving*. (C. Henderson & K. Harper, Eds.).

Mayer, R. E. (2004). Should There Be a Three-Strikes Rule against Pure Discovery Learning? The Case for Guided Methods of Instruction. *American Psychologist*, *59*(1), 14–19. https://doi.org/10.1037/0003-066X.59.1.14

McKenney, S., & Reeves, T. . (2013). Educational Design Research. In J. M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop (Eds.), *Handbook of Research on Educational Communications and Technology* (Vol. Springer N, pp. 131–140). Springer. https://doi.org/10.1007/978-1-4614-3185-5

Moran, A. (1986). Field Independence and Proficiency in Electrical Fault Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, *16*(1), 162–165. https://doi.org/10.1109/TSMC.1986.289294

Perez, R. S. (2012). A view from troubleshooting. In *Toward a unified theory of problem solving* (pp. 127–166).

Perkins, D. N., & Grotzer, T. A. (2000). Models and Moves: Focusing on Dimensions of Causal Complexity To Achieve Deeper Scientific Understanding. Retrieved from http://eric.ed.gov/?id=ED441698

Pretz, J. E., Naples, A. J., & Sternberg, R. J. (2003). Recognizing, Defining, and Representing Problems. In *The Psychology of Problem Solving* (pp. 3–30). https://doi.org/10.1017/CBO9780511615771

Quintana, C., Reiser, B. J., Davis, E. a., Krajcik, J., Fretz, E., Duncan, R. G., … Soloway, E. (2004). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, *13*(May 2014), 37–41. https://doi.org/10.1207/s15327809jls1303

Randolph, J. J. (2009). *Multidisciplinary Methods in Educational Technology Research and Development*. *HAMK university of applied sciences*. HAMK University of Applied Sciences. https://doi.org/10.1007/978-3-211-89836-9_920

Rasmussen, J., & Jensen, A. (1974). Mental Procedures in Real-Life Tasks : A Case Study of Electronic Trouble Shooting. *Ergonomics*, *17*(3), 293–307. https://doi.org/10.1080/00140137408931355

Reid, D. J., Zhang, J., & Chen, Q. (2003). Supporting scientific discovery learning in a simulation environment. *Journal of Computer Assisted Learning*, *19*(1), 9–20. https://doi.org/10.1046/j.0266-4909.2003.00002.x

Reiser, B. J. (2004). Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work. *Journal of the Learning Sciences*, *13*(3), 273–304. https://doi.org/10.1207/s15327809jls1303_2

Richter, C., & Allert, H. (2017). Design as critical engagement in and for education. *EDeR. Educational Design Research*, *1*(1). https://doi.org/10.15460/eder.1.1.1023

Ross, C., & Orr, R. R. (2009). Teaching structured troubleshooting: integrating a standard methodology into an information technology program. *Educational Technology Research and Development*, *57*(2), 251–265. https://doi.org/10.1007/s11423-007-9047-4

Schaafstal, A., Schraagen, J. M., & Berlo, M. Van. (2000). Cognitive Task Analysis and Innovation of Training : The Case of Structured Troubleshooting. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *42*(1), 75–86.

Schraagen, J. M. C., & Schaafstal, A. M. (1996). Training of Systematic Diagnosis: A case study in Electronic Troubleshooting. *Le Travail Humain*, *59*(1), 5–21. Retrieved from http://www.jstor.org/stable/40659988

Schunk, D. H. (2012). *Learning theories : An educational perspective*. Pearson.

Sharma, Sandeep, Hashmi, M. F., & Rawat, D. (2021). Case Study: 33-Year-Old Female Presents with Chronic SOB and Cough. In *StatPearls*. Treasure Island (FL): StatPearls Publishing. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK500024/

Sharma, P., & Hannafin, M. (2007). Scaffolding in technology-enhanced learning environments. *Interactive Learning Environments*, *15*(1), 27–46. https://doi.org/10.1080/10494820600996972

Shin, N., Jonassen, D. H., & McGee, S. (2003). Predictors of well-structured and ill-structured problem solving in an astronomy simulation. *Journal of Research in Science Teaching*, *40*(1), 6–33. https://doi.org/10.1002/tea.10058

Simon, H. A. (1973). The structure of ill structured problems. *Artificial Intelligence*, *4*(3–4), 181–201. https://doi.org/10.1016/0004-3702(73)90011-8

Siponen, M., & Klaavuniemi, T. (2020). Why is the hypothetico-deductive (H-D) method in information systems not an H-D method? *Information and Organization*, *30*(1), 100287. https://doi.org/10.1016/j.infoandorg.2020.100287

Son, L. K., & Schwartz, B. L. (2002). The relation between metacognitive monitoring and control. In T. J. Perfect & B. L. Schwartz (Eds.), *Applied metacognition* (pp. 15–38). Cambridge University Press. https://doi.org/https://doi.org/10.1017/CBO9780511489976.003

Steinberg, L. S., & Gitomer, D. H. (1992). *Cognitive Task Analysis, interface Design and technical troubleshooting*.

Sweller, J., & Chandler, P. (1991). Evidence for Cognitive Load Theory. *Cognition and Instruction*, *8*(4), 351–362.

Sweller, J., Merrienboer, J. J. . G. . van, & Paas, F. G. . W. . C. . (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, *10*(3), 251–296. Retrieved from https://www.jstor.org/stable/23359412 REFERENCES

Teague, R. C., & Allen, J. A. (1997). The Reduction of Uncertainty and Troubleshooting Performance, *39*(2), 254–267.

van Joolingen, W. R., & De Jong, T. (1991). Supporting hypothesis generation by learners exploring an interactive computer simulation. *Instructional Science*, *20*(5–6), 389–404. https://doi.org/10.1007/BF00116355

van Joolingen, W. R., & Zacharia, Z. C. (2009). Developments in Inquiry Learning. In *Technology-Enhanced Learning: Principles and Products* (pp. 21–37).

Van Merrienboer, J. J. G., Clark, R. E., & de Croock, M. B. M. (2002). Blueprints for complex learning: The 4C/ID-model. *Educational Technology Research and*

Development, 50(2), 39–64. https://doi.org/10.1007/bf02504993

van Merriënboer, J. J. G., & Kirschner, P. A. (2017). *Ten Steps to Complex Learning*. *Ten Steps to Complex Learning*. https://doi.org/10.4324/9781315113210

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., & Sloetjes, H. (2006). ELAN: A professional framework for multimodality research. In *5th International Conference on Language Resources and Evaluation* (pp. 1556–1559).

Woods, D. R. (2006). *Successful trouble shooting for process engineers: a complete course in case studies*. Retrieved from https://books.google.co.in/books?hl=en&lr=&id=jBGWytm2aFwC&oi=fnd&pg=PR5&dq=Successful+trouble+shooting+for+process+engineers:+a+complete+course+in+case+studies&ots=xXyG4cmNJm&sig=g6l-XuSVO37JT7BkOVysR3gtBrQ

Woods, Donald R. (2006). *Successful trouble shooting for process engineers: a complete course in case studies*. John Wiley & Sons. Retrieved from https://books.google.co.in/books?hl=en&lr=&id=jBGWytm2aFwC&oi=fnd&pg=PR5&dq=Successful+trouble+shooting+for+process+engineers:+a+complete+course+in+case+studies&ots=xXyG4cmNJm&sig=g6l-XuSVO37JT7BkOVysR3gtBrQ

Woods, Donald R. (2012). PBL: An Evaluation of the Effectiveness of authentic Problem-based Learning (aPBL). *Chemical Engineering Education*, *46*(2), 135–144.

Xun, G., & Land, S. M. (2004). A conceptual framework for scaffolding Ill-structured problem-solving processes using question prompts and peer interactions. *Educational Technology Research and Development*, *52*(2), 5–22. https://doi.org/10.1007/BF02504836

Ziegenfuss Jr, J. T. (1988). *Organizational troubleshooters: Resolving problems with customers and employees*. Jossey-Bass/Wiley.

Zimmerman, B. J., & Magda, C. (2003). Motivating self-regulated problem solvers. In J. E. Davidson & R. J. Sternberg (Eds.), *The psychology of problem solving* (pp. 233–262).

Appendix VI

List of Publications

In Peer-reviewed conference proceedings:

1. Reddy, Deepti, Kavya Alse, Lakshmi TG, Prajish Prasad, and Sridhar Iyer. "Learning Environments for Fostering Disciplinary Practices in CS Undergraduates." In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 1287-1287. 2021.
2. Alse, Kavya, Lakshmi Ganesh, Prajish Prasad, Maiga Chang, and Sridhar Iyer. "Assessing Students' Conceptual Knowledge of Computer Networks in Open Wonderland." In 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT), pp. 513-517. IEEE, 2016.
3. Alse, Kavya, Mukund Lahoti, Meenakshi Verma, and Sridhar Iyer. "GATutor: A guided discovery based tutor for designing greedy algorithm." In 2015 IEEE Seventh International Conference on Technology for Education (T4E), pp. 61-68. IEEE, 2015.
4. Anand, Abhinav, Shitanshu Mishra, Anurag Deep, and Kavya Alse. "Generation of Educational Technology Research Problems Using Design Thinking Framework." In 2015 IEEE Seventh International Conference on Technology for Education (T4E), pp. 69-72. IEEE, 2015.
5. Kavya Alse, Anurag deep, Sridhar Iyer. "Teaching Hypothetico-Deductive Reasoning through murder mystery game." In Workshop on Technology Enhanced Thinking Skills, 23rd International Conference on Computers in Education (ICCE), Hangzhou, China, Dec 2015.
6. Majumdar, Rwitajit, Kavya Alse, and Sridhar Iyer. "Interactive Stratified Attribute Tracking diagram for learning analytics." In 2014 IEEE Sixth International Conference on Technology for Education, pp. 138-139. IEEE, 2014.
7. Anand, Abhinav, Aditi Kothiyal, Anita Diwakar, Anura Kenkre, Anurag Deep, Deepti Reddy, Jayakrishnan Warriem et al. "Designing Engineering Curricula Based on Phenomenographic Results: Relating Theory to Practice." In *2014 IEEE Sixth International Conference on Technology for Education*, pp. 80-87. IEEE, 2014.

Doctoral Consortiums:

1. Alse, Kavya. "Teaching: Learning of Troubleshooting Skills for Computer Science Undergraduates." In Proceedings of the 2016 ACM Conference on International Computing Education Research, pp. 281-282. 2016.
2. Kavya Alse, Teaching – learning of Troubleshooting Skills for CS undergraduates, In Internatoinal Conference on Computers in Education, 2016

3. Kavya Alse, How to teach troubleshooting skill to Computer Science undergraduates? In Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education, 2016

Acknowledgement

This journey of PhD has been an eventful and fruitful one in many ways. It has given me opportunities to learn new things not just academically but also idiosyncrasies of myself and those around me. I am happy that I got these opportunities to interact with various people and learn from them. I want to express my sincere gratitude to each one of them.

I express my heartfelt gratitude to my research advisors Prof. Sridhar Iyer and Prof. Sasikumar M. They believed in me and allowed me to try things and fail at them. Prof. Sridhar has patiently heard my (probably incoherent sometimes) rumblings and provided constructive feedback always. I am deeply indebted to him for pushing me and standing behind me always. Prof. Sasikumar provided useful critiques and new perspectives in every meeting that helped to raise the quality of my work.

It is fair to say that Prof. Sahana Murthy laid the foundations of this journey via the wisely designed courses that she taught – Introduction to Educational Technology and Research Methods in Education. The ‘so what’ questions that she would regularly ask in these classes have helped to crystallize my thoughts often all these years. As a woman in STEM research, she has been a role model in the department. I am grateful for her guidance throughout. The other research progress committee member, Prof. Chandan Dasgupta has suggested judicious and sensible critique on my work. I thank him for all the valuable feedback.

I am indebted to all the members of the Interdisciplinary Programme in Educational Technology, IIT Bombay, my home away from home during this course. I learned equally from the formal and informal interactions with my colleagues here. My friends and colleagues who travelled together in the journey of educational research have made this more meaningful.

I thank Kapil, Shitanshu, JK, Rwitajit, Aditi, Abhinav and Deepti for their friendship and guidance. I cherish my interactions with Anurag and thank him for pushing me to be pragmatic. I am grateful to Soumya, Prajish, Lakshmi and Ashutosh for their support and discussions. I thank Pankaj, Navneet, Rumana, Ulfa and Sunita for their friendship and support all these years. Radhika, Soumya, Vaishnavi and Pritam – you made the mess food bearable and the conversations always made me cheerful. Thank you!

I thank Kinnari who urged me to think about the TELE carefully with heaps of questions. I thank Prajwal Barapatre and Bhupender Singh for the development of PHyTeR. I am grateful that Ajit, Rumana, Daevesh, Priya and Anurag supported me while conducting studies. I am thankful to Pallavi, Seema, Vidya, Prakash and Asha from the ET and academic office for processing my documents on time.

I thank all the students who participated in the studies and the faculty who helped me in organising these studies. I thank Prof. Viraj Kumar and Prof. Srinivasan L at Dayananda Sagar University, Bengaluru for all the encouragement and support they provided while writing the thesis. Alaka, Akshata and Anand have been compassionate and backed me throughout this journey. I especially want to thank Alaka for being there and finding the right words to cheer me up always. I am grateful to Dr. Radhika Sharma at the IITB hospital and Dr. Malati V Rao for helping me to tackle a lot of obstacles.

Lastly, I acknowledge the encouragement my family provided. I am fortunate to have Amma, a teacher- researcher herself and inspiring me to be a teacher – researcher; Appa, who always encouraged me to read and ask all sorts of questions; and Atte for all the love and care she provided. I thank my in-laws, Sudarshan, Shashi and Sowmya for their support. This thesis would not have been possible without the support of my husband, Udaya Kiran – thank you!

I take the privilege to thank all the people who helped and supported me directly or indirectly throughout my academic career.

Date: 20th May 2022 Kavya Alse

IIT Bombay, India 144380001