Sudan University of Science and Technology

College of Graduated Studies

Design and Evaluation of three IT Artifacts (Dr. SUST) for Increasing Students' Engagement in Java Programming Course

تصميم و تقييم ثلاث إسهامات تقنية معلوماتيه (د.سست)

لزيادة تجاوب الطلاب في مقررالبرمجة بلغة جافا

Dissertation
Submitted in partial fulfilment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY
in
Information Technology

By

Nejood Mohammed

Supervisor

Dr. Laurie Butgereit

ABSTRACT

Programming is a discipline that requires ample amount of time to be dedicated to different learning activities to be mastered. Educators try to find ways to engage students and urge them to spend more time in learning and practicing programming. At Sudan University (SUST), programming is a fundamental subject to computing students which is usually taught during their first year of study, and that will affect their entire subsequent studies. The staff teaching programming subject are concerned because some students seem to be disengaged. Disengagement leads students not to dedicate all the possible time they get in performing programming learning related tasks. On the other hand, when students are engaged related research states that it is positively correlated with desirable learning outcomes, including: general abilities and critical thinking, cognitive development, student satisfaction, improved grades and persistence.Timely interference to help and support students could affect students' engagement positively. Hence, personalized learning could help in providing detailed help and guidance to different students based on students' details or students' models. Personalized learning or adaptive systems that rely on user model/profile could be designed to meet individual students' needs. This research aims at increasing and evaluating students' engagement in programming learning using technology enhanced learning methods. Three artifacts were produced in this work to meet the research objectives: Firstly, the attributes contributing to students' engagement were perceived from three sources: the literature, students' quantitative and qualitative surveys, and from evaluating the usage of the designed solution. Secondly, overall technical details of the suggested solution and the design decisions were presented as an engaging adaptive model. And the final artifact was the design of the adaptive system DrSUST. There were three iterations in developing the technical solution. The last objective was to evaluate students' engagement in the designed solution. To monitor students' engagement while using the adaptive system, analyzing web system logs was performed. And hence, the following measurements were used to measure users' engagement: Click-through rates, time spent on site or dwell time, frequency of return visits (during single or multiple sessions), number of tasks, and reading amount. Adaptive systems differ in their implementation based on the aspects of the design that need to be emphasized and improved in the system. In this work the emphasis was on increasing students' engagement while learning programming. This was reflected in the design by involving the students early on in the implementation of the solution and studying their current situation and the aspects that they needed to be provided to achieve better engagement. After running the system for three iterations the attributes that were used for modeling the students are: navigation pointer, language, quizzes level and exercises level. The activities that were attractive to students were programs solutions sharing, questions and comments, simplified summary and quizzes. It was realized that students follow different paths when studying. Thus students could be categorized based on their learning styles according to the path they follow during learning. Some students prefer to continue reading without performing related quizzes and exercises and delaying these tasks to the end. While other students finish each topic along with the related work before moving to advance topics.

المستخلص

البرمجةهي مجال يتطلب كمية وافرة من الوقت لتكون مكرسة لأنشطة التعلم المختلفة إلى أن يتقن. المعلمين يحاولون ايجادسبل لإشراك الطلاب وحثهم على قضاء المزيد من الوقت في تعلم وممارسة البرمجة. في جامعة السودان (SUST)،البرمجةهي مادة أساسية لطلاب الحاسوب والتي عادة مايتم تدريسها خلال السنة الأولى من الدراسة، والتي من شأنها أن تؤثر على دراستهم كلها. اساتذه مادة البرمجة قلقون لأن بعض الطلاب يبدون غير متجاوبين مع المادة. عدم تجاوب الطلاب يؤدي إلى ان الطلاب لايكرسون كل الوقت الممكن في اداء النشاطات المتعلقة بمادة البرمجة. من ناحية أخرى اشارت البحوث إلى ارتباط تجاوب الطلاب عند التعلم بصورة إيجابية مع نواتج التعلم المرغوبة، بمافي ذلك: القدرات العامة والتفكير النقدي، النمو المعرفي، رضاالطلاب، تحسين الدرجات والمثابرة. التدخل في الوقت المناسب لمساعدة ودعم الطلبة قد يؤثر على تجاوب اللطلاب بشكل إيجابي. وبالتالي، يمكن لوسائل التعلم الشخصية أن تساهم في تقديم المساعدة المفصلة والتوجيهات المختلفة للطلبة استنادا إلى وصف الطالب أو نموذج الطالب. وسائل التعلم الشخصية أو الأنظمة التكيفية التي تعتمد على نموذج المستخدم يمكن تصميمها لتلبية احتياجات الطلاب الفردية.يهدف هذاالبحث إلى زيادة وتقييم التجاوب في تعلم مادة البرمجة الأولية عبر إشراك الطلاب في مقرر برمجة الجافا من خلال استخدام نظام الدروس المكيفة حسب معلومات الطالب. حيث يتم تزويد الطلاب بمجموعة من الأنشطة ذات الصلة بتعلم البرمجة والتي تكون جاذبة ومفيدة. علاوة على ذلك سيتم تتبع استخدام الأنشطة المختلفة للطلاب اثناء استخدام النظام لتحديد الجانب الأكثر جاذبية لتعلم البرمجة تجريبيا.يحتوي وقد تم إنتاج ثلاثة مساهمات في هذا العمل لتحقيق أهداف البحث: أولا، تم النظر إلى السمات التي تسهم في مشاركة الطلاب من ثلاثة مصادر: الدراسات ذات الصله و الاستبيانات الكمية والنوعية للطلاب، ومن تقييم استخداميةالنظام. ثانيا، تم عرض التفاصيل الفنية الشاملة للحل المقترح وقرارات التصميم كنموذج تكيفي جاذب. وكان المنتج الأخير هو تصميم نظام قابل للتكيف (DrSUST).كانت هناك ثلاثة تكرارات في تطوير الحل التقني. وكان الهدف الأخير هو تقييم تجاوب الطلاب في الحل المصمم. لمراقبة مشاركة الطلاب أثناء استخدام نظام التكيف، تم إجراء تحليل لسجلات النظام. وبالتالي، تم استخدام القياسات التالية لقياس تجاوب المستخدمين: معدلات النقر

إلى الظهور، والوقت الذي يقضيه في الموقع أو وقت الإقامة، وتكرار الزيارات العائدة (خلال جلسات فردية أو متعددة)، وعدد المهام، ومقدار القراءة. في هذا العمل كان التركيز على زيادة تجاوب الطلاب عند تعلم البرمجة. وقد انعكس ذلك في التصميم من خلال إشراك الطلاب في وقت مبكر في تنفيذ الحل ودراسة وضعهم الحالي والجوانب التي كانوا بحاجة إلى توفيرها لتحقيق مشاركة أفضل. كانت هناك ثلاثة تكرارات في تطوير هذا الحل التقني. تمت إضافة السمات التي تمثل الطلاب بشكل متزايد على كل التكرارات للنظام. بعد تطوير و استخدام النظام لثلاث دورات كانت السمات التي تستخدم لتمثيل نماذج للطلاب هي: مؤشر الملاحة, اللغة, مستوى الاختبارات, و مستوى التمارين. وكانت الأنشطة الأكثر جاذبية للطلاب هي تقاسم حلول البرامج، الأسئلة والتعليقات، الملخصات المبسطه للمواضيع بالاضافة للاختبارت.وقد تبين أن الطلاب يتبعون مسارات مختلفة عند الدراسة. وبالتالي يمكن تصنيف الطلاب على أساس أنماط التعلم الخاصة بهم وفقا للمسار الذي يتبعونه أثناء التعلم. بعض الطلاب يفضلون مواصلة القراءة دون أداء الاختبارات ذات الصلة والتمارين وتأخير هذه المهام حتى النهاية. في حين أن الطلاب الآخرين الانتهاء من كل موضوع طويل مع العمل ذات الصلة قبل الانتقال إلى المواضيع المتقدمة.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 Introduction

Higher education studies require individuals to have abilities in researching and qualifying oneself with knowledge. This requires interest, self-efficacy and discipline from students. Students differ in the way they learn and respond to learning methodologies and materials presentation styles. Several positive students' attitudes can aid in the process of learning. Some of these attributes fall under the umbrella of engagement. Learning different subjects requires a variety of learning methods and activities to be performed. Programming has unique characteristics as it is a subject that requires skill and higher order thinking in addition to memorizing syntax. Students may lack interest in any of the activities needed to learn programming and hence face difficulties in learning.

Programming is a discipline that requires ample amount of time to be dedicated to learning activities in order to be mastered. Educators try to find ways to engage students and urge them to spend more time in learning and practicing programming. At Sudan University (SUST), staff teaching programming subject are concerned because some students seem to be disengaged. Disengagement leads students not to dedicate all the possible time they get in performing programming learning related tasks. On the other hand when students are engaged related research stated that it is positively correlate with desirable learning outcomes.

In this chapter issues related to programming teaching and learning are introduced and students' characteristic that can have positive effect on learning is presented, namely students' engagement. Some of the issues affecting engagement in programming learning that could be handled by personalized tutoring are discussed. Also, an alternative technology enhanced learning solutions will be presented. The motivations for this work are listed along with the intended research objectives.

1.2 Programming Teaching and learning

In a study by Hawi (2010), some of the factors that lead to success or failure were identified. The ten causal attributes identified in his work were: learning strategy, lack of study, lack of practice, subject difficulty, lack of effort, appropriate teaching method, exam anxiety, cheating, lack of time, and unfair treatment. Learning strategy was on the top of the list of Hawi.

In a work by Robins, an explanation to the phenomenon of introductory programming grade distribution was proposed. It has been observed that the grades usually have higher than usual low and high grades creating bimodal grade distribution. The proposed explanation was called the learning edge momentum (LEM) effect (Robins, 2010). The concept states that since there is a tight relation between programming structures/concepts, understanding one structure or concept will help in progressing while failing to understand a structure or concept will lead to difficulty in understanding the following structures or concepts. Understanding a topic will increase the comfort level and confidence of the students which affect students' engagement. When the structures of a course are tightly related, this will lead to bimodal grade distribution. While independent course structures will result in a normal grade distribution. This can also help in understanding the same phenomenon occurring in other subjects. From their study, students pointed out that in programming the material quickly "builds on itself" and that falling behind is a problem as it is difficult to catch up (Robins, 2010).

It is difficult for many students to use programming languages to write programs for solving problems. One of the reasons that causes learning difficulties is the lack of problem solving abilities that many students show. Solving problems is not easy to learn and novices usually don't know how to create algorithms. Training is required in order to help students obtain that skill. In the work by Gomes and Mendes, they proposed building a tool that helps students practice developing and testing algorithms; their tool is named SICAS (Interactive System for Construction of Algorithms and its Simulation)(Gomes and Mendes, 2007).

Ozmen et al. have attempted to determine the reasons of failure of undergraduate students in programming courses and the difficulties they encountered. In this regard, it has been observed that students' difficulties were mainly related to programming knowledge, programming skills, understanding semantics of the program and debugging. Difficulties related to programming knowledge can be listed in the following order; syntax, knowing the concepts or principles related to the programming language, remembering the functions and its parameters, defining variable and choosing the decision structures and loops that will be used in program. In their study, almost all of the students experienced problems in programming knowledge, and as a result of this, they had difficulty in syntax. In addition, it was found that most of the students had trouble in understanding semantics, debugging and programming skills. Programming skills theme refers to students' ability to design solutions to problems in programming and to determine strategy to be followed while reviewing his/her programming knowledge. In this study, it was observed that students who had considerably higher level of programming success also had a higher level of self-efficacy as well. These students mentioned that programming is actually an easy process as long as necessary repetitions are made and they started with developing algorithm before writing the program. Therefore, successful students believe that they can write program codes successfully if they are given enough time (Özmen and Altun , 2014).

Programming is a craft that often demands that learners engage in a significantly high level of individual practice and experimentation in order to acquire basic skills. However, practice behaviours can be undermined during the early stages of instruction. This practice when left unchecked; create cognitive-affective barriers that interact with learners' self-beliefs which will potentially reduce practice. Scott and Ghinea seek to ascertain how to design a learning environment that can address this issue. They proposed that analytical and adaptable approaches, which might include soft scaffolding, on-going detailed informative feedback and a focus on self-enhancement alongside skill development, can help overcome such barriers (Scott and Ghinea, 2013).

In addition, Rogerson and Scott examined how students' experiences of learning to program are affected by feelings of fear, using a phenomenological approach to elicit rich descriptions of personal experiences. During the analysis of the data, six main themes emerged regarding the students' experience of learning programming. For their study, the word "fear" is regarded as a descriptor for denoting a lack of interest in programming as a discipline, lack of confidence or hesitation regarding their ability to code or program. This fear affects other aspects related to their studies, such as self-confidence, time management, and problem solving skills. For those students, there is a critical need for intervention, and some suggestions have been made. For example, formal, one-on-one consultations with the lecturer at strategic intervals may help students to overcome their fears sooner and, as a result, increase their comfort and enjoyment levels. The aim is to allow the students to reach their full potential without fear (Rogerson and Scott, 2010).

1.3 Students' Engagement

Student engagement can be interpreted in different ways and there was effort made to set a definition by several authors. In a review by (Trowler, 2010) the author tried to define what is meant by the term engagement in educational survey and he listed a group of students types and their way of engaging. There is a robust correlation between student involvement in a subset of 'educationally purposive activities', and positive outcomes of student success and development, including satisfaction, persistence, academic achievement and social engagement. The following definition of engagement was proposed by Trowler from his understanding of the reviewed research:

"Student engagement is concerned with the interaction between the time, effort and other relevant resources invested by both students and their institutions intended to optimize the student experience and enhance the learning outcomes and development of students and the performance, and reputation of the institution."

To improve the process of teaching or education, the way students learn need to be inspected and understood. One of the biggest challenges in education is how to keep students interested and motivated for the amount of time necessary for learning. Educators are researching ways to engage their students.

1.3.1 Why Engagement?

For learners, engagement correlates with improvements in specific desirable outcomes (Trowler, 2010) including:

- general abilities and critical thinking
- cognitive development
- student satisfaction

- improved grades
- persistence

Struggling in programming courses can result from low engagement. Better achievement can result in the programming courses if engagement level increased. Giving choices in the learning materials and providing continuous support can result in greater engagement.

1.4 Adaptive systems to engage students

Protecting students from the negative feelings that could lead to disengagement is possible if there was a good tutor. Tutors should be sensitive to students' feedback and detect the feelings like boredom and frustration from the start and put a plan to overcome them. Being sensitive to students' feedback is possible in small classes but unlikely to work in large classes.

ICT offers countless solutions that can be utilized in education. This range of solutions is growing rapidly as more computational power and networking bandwidth are made available every day. Even the mobile devices are becoming very powerful and programmers can design applications without being limited by the different factors such as size, speed, and networking capabilities. It is challenging to survive in the development market without a well-designed application that meets the demand of users by frequent updates. It is no longer enough for users to face the challenges hardcoded in the application, intelligent solutions, and or solutions involving other capable users are more likely to last longer and attract more users for example applications with sharing, collaborating, and challenging interaction.

Technology has affected the way people perform activities and provide services. There are many possibilities and opportunities that can result when employing technology in education. It is about time that students benefit from technological advances in learning. A range of technologies support education (e.g. multimedia), personalized learning, and more interestingly there are some technologies that aim at reducing the load on educators (i.e. personalized learning using technology). The use of personalized learning has several advantages for students such as having support outside the usual working hours and visiting the materials as many times as needed. Extending the tutoring hours beyond the class hours will enable the students to understand different topics better and also it will consider students with different learning speed.

Adaptive could refer to personalization or sensitivity to users' information. Systems that implement adaptivity in their work generally rely on users' attributes (model or user profile) to arrange or display the contents (Brusilovsky and Millán, 2007)

The research in personalized learning using technology (adaptive systems and Intelligent Tutoring System) agrees generally that there are three main components involved, namely (Sottilare et al., 2013):

- Domain model - information that represents the knowledge in the system.
- Pedagogical model - handles the actual "tutoring" aspects of the system, and do adaptive selection of the materials to be presented to students.
- Student or learner model - used to determine the student's progress.

Sometimes the communication interface is also considered as a fourth component.

1.5 Motivation

In this research, the author has taught programming in higher education for many years. Frustration was observed among some of the students who feel left behind which will affect their entire study. In addition to that, trials to increase students' engagement were studied and it was observed that the existing solutions were not meeting all the of SUST students' needs. Mainly there are two motivations for this work:

1. Struggling in programming courses can result from low engagement.
2. Engagement level can be influenced by manipulating students' engagement attributes.

1.6 Objectives

This research is trying to find out:

- What are the attributes of students' disengagement in learning programming at SUST?
- How can adaptive tutoring systems increase students' engagement?
- How can students' engagement be measured?

1.7 Conclusion

Issues related to programming teaching and learning were investigated. From the literature, it is found that this is an active area of research as programming represents a skill that has challenges for novice learners and that requires critical thinking. Since programming skill need to be developed through intensive practice, students need to allocate time for performing the related learning tasks. Students can allocate time for learning if they engage with the subject. This from both literature and experience seems to be missing by many of programming learners. Students' engagement is affected by many factors, including timely feedback and scaffolding.

It is difficult to meet the needs of the increasing number of students by their tutors.

The aim of this work is to investigate the current technologies for personalized learning and to implement a solution for helping novice programming learners. Also the research aims to find the attributes for the selected technology that could be adjusted to increase students' engagement in programming learning.

1.8 Thesis Organization

The thesis is organized into seven chapters:

CHAPTER TWO: LITERARTURE REVIEW: Discusses the literature review and it is divided into 3 major sections

Section 1: Presents a review on students' engagement

Section 2: Presents Literature on Adaptive and intelligent systems

Section 3: Is dedicated to reviewing methods on measuring user engagement from systems logs.

CHAPTER THREE: METHODLOGY: Discusses the research methodology used in this work and the overall system framework.

CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT: Presents the results of a survey that tried to find the level of students' engagement and the requirements for designing a more engaging course obtained from interviewing the students of introductory programming course.

CHAPTER FIVE: The Model.

CHAPTER SIX: Presents the Adaptive System (DrSUST).

CHAPTER SEVEN: RESULTS and ANALYSIS, shows the system log data results and analysis.

CHAPTER EIGHT: Presents the conclusion and future work.

CHAPTER TWO: LITERARTURE REVIEW

2.1 Introduction

In the previous chapter (CHAPTER ONE: INTRODUCTION) the research problems and questions were discussed. The research aims to address a real problem and hence the nature of this work is multifaceted. Knowledge about both students' engagement in programming learning and the problems they face and also knowledge about the up to date technological approaches that could be implemented as a solution need to be covered.

In this chapter the literature in multidisciplinary fields that is required and connected to this research is demonstrated. Theliterature needs to cover several aspects. The first aspect is about the theoretical background of students' engagement that is required for understanding ways of addressing it later in the implementation. The second aspectis to investigate the technologies that can be implemented to help in individualized learning.And at last aspect is to find information aboutevaluating the students' interest or engagement on the developed solution. Therefore the literature is divided into the following three major literature components:

Section 2.2 presents the literature of students' engagement and also shows the current challenges faced by educators and specially when teaching programming.

Section 2.3 presents adaptive systems as a mean of providing personalized learning. The section shows various techniques including: recommender navigation systems, intelligent tutoring system and adaptive systems.

The last section 2.4 shows ways of measuring users' engagement in systems generally and users' engagement when trying to analyze it from the web. The online systems have a greater challenge which is the possibility of having huge amount of data due to the open access which will limit the choices of the data to be analyzed.

By studying these different aspects, this would enable decisions to be made regarding the research methodology. Attributes and recommendations that are acquired from first sections would aid the design of the technological solution. In addition the technological review on the second section will emphasize the technology to be adapted in the solution design. And lastly the last review should aid the evaluation of the implemented solution.

2.2 Literature Review on Students' Engagement

2.2.1 Introduction

Students' engagement affects students' performance in programming learning. Only engaged students can allocate time for doing programming related activities. Frustration can hinder the process of learning and it can occur when teaching methods don't appeal to students or when students don't receive timely feedback on their programming errors. Several IT solutions exist these days that could help in providing a quality learning experience and improve engagement such as education management systems, adaptive systems and intelligent tutoring systems (Lowyck, 2014).

2.2.2 Factors Affecting Students' Engagement

Several factors and attributes are believed to affect engagement and give a good indicator of engagement in learning. The following are some factors identified in the work of Guenther and Miller (Guentherand Miller, 2011) and the Australasian Survey of Student Engagement (AUSSE) (designed to measure student engagement in higher education) *(*AUSSE, 2009) are:

2.2.2.1 Individual Factors that Contribute to Student Engagement:

1. Academic Challenge
2. Perceived Control and Autonomy
3. Perceptions of a Supportive Environment
4. Achievement Motivation and Goals

2.2.2.2 Educational Practices that Contribute to Student Engagement

1. Active Learning
2. Enriching Educational Experiences

3. Diversity Experiences
4. Shared-Learning Opportunities
5. Student-Faculty Interaction
6. High Expectations
7. Work Integrated learning

2.2.2.3 Additional attributes related to students' engagement

1. Collaboration
2. Self-efficacy/ effort
3. Time availability/extra-curricular activities/ responsibilities
4. Interest
5. Support / Scaffolding

Many of the above factors apply for all types of subjects learning while additional issues exist when discussing programming learning such as the availability of devices and resources (the ability to practice outside class).

The levels in which students enjoy programming differ throughout the programming course. Initially, almost all students enjoy the concepts when they are first introduced to programming. During the course as new concepts are introduced and students' mastering levels differ significantly, some students feel left behind as noted by Rogerson and Scott (2010). Students who are unable to follow the course speed might feel frustrated and eventually become disengaged. Interventions can be designed to prevent students' frustration and hence disengagement such as giving several opportunities for students to make up for poor and missing assignments.

An additional factor that is suggested by this research is the purpose of learning the programming language that is to say: the feeling of pressure/no pressure. Many of the postgraduates who learn programming for their research don't complain from this requirement. In addition, many people learn programming as a matter of interest. Why does it seem easier for those individuals, the reason can probably be: firstly; nonexistence of exams, secondly; they are mostly achievers (hard workers), thirdly; the usage of visualization tools, or because they don't write code from scratch and they just modify existing code in addition to reusing chunks of ready code.

In a review of students' engagement (Trowler , 2010), the following factors were identified as the elements that can be influenced to increase students' engagement. The attributes are:

1. Students
2. Staff
3. Local context - inclusive environment (equality)
4. Institution's duty
5. Educational ideology - learning is influenced by how an individual participates in educationally purposeful activities
6. National policy
7. Linking the levels

In programming teaching investigation, the research will only consider the three top attributes. To improve the process of teaching or education, studying and understanding the way students learn is essential. One of the biggest challenges in education is how to keep students interested and engaged for the amount of time necessary for learning. Educators are encouraged to research ways to engage their students.

ICT offers countless solutions that can be utilized in education. This range of solutions is growing rapidly as more powers and communication bandwidth are made available every day.

One solution is blended learning. Blended learning is a classroom format that consists of a mixture of activities, balancing between online and face-to-face interaction as well as thoughtful sequencing of academic activities. Two factors make blended learning a necessity and a need, namely: The need for time saving activities due to students' overloaded schedule. And the availability/accessibility of all types of cutting edge technologies that can be utilized in education. A difficult step in adopting this approach is finding a method to incorporate these technologies in learning environment; especially when grades are involved (Villanueva, S , 2011). When two pedagogical methods - using online tutorials in combination with lectures and hands-on exercises- were used in class, positive results were encountered. This practice, particularly worked for motivating and activating the growing number of students whose will to learn is low - “minimalist students”- (Thomsen, 2008).

2.2.3 Students’ Engagement and Programming Pedagogy

For learners, engagement correlates with improvements in specific desirable outcomes (Trowler , 2010), some of these outcomes are:

- General abilities and critical thinking.
- Cognitive development.
- Student satisfaction.
- Improved grades.
- Persistence.

Some of the factors that can result in disengagement among students as described by (Bennedsenet al, 2008) are:

- The current style of education doesn’t appeal to everyone.
- Some educators are not sensitive to students’ responses to educational methods in class
- Students don’t do what they must do, but what they want to do.
- For some students, there will be no satisfaction as their goals and aspirations never occur.
- Students overwhelmed by all they have to do - ‘quality of effort’

The computer programming pedagogy research addresses the problem of finding effective ways to teach programming. A search for the appropriate pedagogical approach to teaching programming has been studied extensively.

Gomes and Mendes (2007) stated that the following factors need to be considered when teaching programming: firstly; programming demands a high abstraction level-generalized way of thinking; secondly, programming needs a good level of both knowledge and practical problem solving techniques; thirdly, programming requires a very practical and intensive study; fourthly; use individualized way of supporting students (i.e. know their errors pattern); in addition to using innovative way of teaching programming due to its dynamic nature.

Several factors were identified by Jenkins (2002) that needs to be adopted to successfully teach first programming course. Some of these factors that should be changed in order to reduce the perceived difficulty to learn programming are: Firstly; the language used should be chosen for pedagogic suitability and no other reasons. Secondly; programming should be taught by those who can teach programming. Thirdly; the pressure on students should be reduced by eliminating continuous assessment. And finally; departments should support students in their programming learning adequately since the subject is perceived difficulty.

Adispute exists on how first programming course should be taught and what are the syllabus of the course. Generally there are no fixed guidelines and in addition, lecturers are aware that they do not do well on teaching testing and debugging. One aspect that makes teaching programming a challenge is that programming is a difficult mixture of art and science (difficult to do and more difficult to teach). More research, experimentation, assessment, discussion, and debate need to be done to discover new methods of teaching programming (Robins et al. , 2003). As with all aspects of teaching reflection of new methods should be shared and teaching models need to be communicated with others on the field. This can help in optimizing the work of instructors and students to increase learning gains and lecturers' satisfaction. Disengaged class is hard to handle by both instructors and students.

Novices differ from expert on the way they think and hence the way they solve problems. For example, they approach programming "line by line" rather than on the level of meaningful structures. Novices have problems in allocating enough time for code planning and code testing. When novices learn programming, they usually have a range of background knowledge and attitude towards programming. One of the attitudes that affect their learning is their behaviour towards mistakes/errors. Students who get frustrated by errors are likely to quit coding when they are faced by them i.e. become stoppers (Robins et al. , 2003). A good pedagogical theory for teaching programming should focuses on students' learning, and effective communication between teacher and student. This could be achieved by clearly stating goals and keep the students motivated. Keeping students motivated/engaged can be achieved by: firstly, stimulating interest and involvement with the course; secondly, actively engaging the student with the materials; and finally, Use appropriate assessment and feedback.

Programming learning requires an intensive level of practice. This nature of programming makes students complain from the effort they allocate in the subject which can affect their other activities.

In the work in Alsaggaf (2013), it was suggested that students play active role on the learning process by doing practice while attending the theoretical lecture as a proactive practice-based learning. This will engage the students and help the learner to construct his or her own knowledge of the concepts provided by the instructors. Positive feedback was received from students and lecturers about applying a mobile-based constructivist learning and teaching approach, although there are some practical issues for successfully incorporating mobile devices in class.

Learning to program requires intensive amount of practice. A study by Konecki and Marko suggested giving students a large body of code examples. This rich resource will give students a good idea about what constitutes a good code and will help them in learning by exploring. The reinforcement of theory through practice is achieved by interspersing lecture and discussions, presentations with hands-on implementation and code exploration exercises (KoneckiandPetrlić, 2014).

Another research (Nikula et al, 2011) tried to answer three questions related to difficulty of programming courses. The first question was about finding out the reason for the high failure rate in programming and whether it is due to the subject difficulty or the reason is the teaching methods. The answer to this question is still undergoing research. This could be attributed partly to the gap mentioned by Don Norman (Norman and Draper, 1986) about algorithmic thinking and problem-solving skills. This could be solved by making the machine think like humans or make humans understand machine coding which is the norm. The second question tried to identify and analyze attitudes and habits of programming novices that could affect learning. The complexity and novice nature of the subject has been found to affect learning. The third question tried to summarize the efforts made to improve success rates. A number of researches have been presented in this regards including:

increasing motivation and remove demotivation and handling first year students' issues by postponing the course to senior level and using supportive tools like visualization.

2.2.4 Students' personalities

Students' beliefs about their academic level and achievement expectations have found to play a major rule in academic performance. Students who believe that they are competent, will never give up and will try to understand the materials in a number of ways. This character has also been observed in games playing, in which players are classified into different categories, namely: killers (killer is someone who enjoys the ability to defeat others), achievers (they want to achieve within the context of the game, e.g. be leaders), socializers (they enjoy socializing, engage in conversations and help other players) and explorers (they aim to see how far they can go, explore environment). In many games, the name represents exactly what this player does (Kapp, 2012).

The following table shows the conclusions of two researches that tried to classify the students according to the effort they put while programming.

Table 2-1: Personal problem solving styles

Students' first classification **Hosseini et al.(Hosseiniet al , 2014)**	**Students' second classification** **Perkins et al. (Perkins et al. , 1986)**
Builders: behave exactly as expected; adding concepts that increase correctness	Movers: steadily work towards a solution
Massagers: long streaks when they are trying to get to the next level of correctness by doing small code changes without adding or removing concepts	Tinkerers: writing a small code and making small changes to make it work
Reducers : behavior opposite to the building when students remove concepts while maintaining or reducing the correctness level (optimization)	Massagers and Reducers could be probably considered as a mixture of Perkins' movers and tinkerers.
	Stoppers : freeze when facing problems

Students' first classification **Hosseini et al.(Hosseiniet al , 2014)**	**Students' second classification** **Perkins et al. (Perkins et al. , 1986)**
Strugglers: These students spend considerable amounts of time to pass the first correctness test. They do all kinds of code changes, but probably have too little knowledge to get the code working.	

2.3 Literature Review of Adaptive Systems

2.3.1 Introduction

Several adaptive systems that rely on user model or user profile to arrange or display contents exist (Brusilovsky and Millán,2007) such as:

- Information retrieval and filtering systems (Recommender systems): uses user's *interests* in terms of keywords or concepts to find relevant documents to users.
- Intelligent tutoring systems (ITS): aim at providing personalized educational activities and individual feedback to users
- Adaptive Hypermedia (AH) and Adaptive Educational Systems (AES): employ the user knowledge models to present materials suitable to the student's various attributes.

2.3.2 Information retrieval and filtering systems

Vast materials are available online that students can use to study. Sometimes the amount of materials exists online might be overwhelming and students can waste time going from topic to another **(**Labaj and Bieliková, 2014**)**. Systems exist that try to arrange the retrieval of the contents and try to recommend a learning path to students thus:

- Helping both tutors and students to find learning materials

- Reduce the looping or materials revisiting for students and make them select the optimal materials that is suitable for the individual learners. (Lalmaset al. ,2014)

2.3.3 Recommender systems

Recommender systems are used to improve the students' learning experience of the vast materials available online about a particular topic. The materials found in education could range from texts (explanations) to interactive content such as exercises and questions. As the levels of students differ a personalized recommendation is desired to select materials that are optimal for the students. The purpose of the recommender system is to select items with optimal difficulty that is to say difficult enough to keep them occupied to solve it, and not too difficult to make them quit. In the work by Labaj and Bieliková (2014**)**, they described a learning object recommendation method based on:

1. Students' explicit difficulty ratings: ask the students during and after exercise/question solving.
2. Implicit rating: or to predict ratings from users' actions this second method will reduce the disparity in explicit ratings since students' ratings could be affected by other factors besides students' knowledge. **(**Labaj and Bieliková, 2014**).**

2.3.4 Intelligent Tutoring System

2.3.4.1 ITS Technologies and Performance Attributes (VanLehn, 2011):

Several technologies exist for building such systems and each technology can be used in a particular context. The Artificial Intelligent approaches used in encoding conceptual models are: fuzzy logic, neural networks, genetic programming and any of these methods combinations (hybrid approaches).

It has been reported that tutoring systems in general (human based or automated) increase learning outcomes, and the best tutoring is one-to-one tutoring. In addition a good one-to-one human tutor outperforms ITS and the work in (VanLehn, 2011) explored the reasons of the performance differences. This work examined several hypotheses explaining why human tutoring outperforms computer intelligent tutoring. The importance of their survey is that it identifies the attributes for successful tutors. Researchers can benefit from it by focusing on fixing the weakness of the (ITS). Improving the effective aspects about students learning will certainly improve the students learning capabilities when using ITS. The expected reasons for the human tutoring system's superiority are summarized in table1 below.

Table 2-2: Hypothesis explaining the reasons for human tutoring outperformance (VanLehn, 2011)

Claim	Claim description	Claim Validity
Detailed Diagnostic Assessments	Tutors use information about the levels of the individual students. And accordingly tutors will prepare lessons to adapt to the needs of the individual students.	Practically, it doesn’t seem that tutors assess their students. And even if assessment information is presented to them, they don’t benefit from it.
Individualized Task Selection	Tutors can select tasks to suit the learning needs of the individual students and to cover their weaknesses. However, it has been suggested that human tutors select tasks using a curriculum script.	Curriculum scripts are also used by some computer tutors and others use even more advanced methods. Thus, this hypothesis does not explain human tutor superiority.
Sophisticated Tutorial Strategies	Studies have shown in several areas with tutors having different degrees of service expertise that it is rare that tutors use sophisticated strategies	Using sophisticated tutorial strategies does not explain performance differences between human and computer tutors.
Learner Control of	Human tutors can change topics based on discussion with	Analyses showed that human dialogues are not frequent and

Claim	Claim description	Claim Validity
Dialogues	students unlike ITS. In addition, students can ask any question even outside the current topic.	thus this doesn't explain performance difference.
Broader Domain Knowledge	Human tutors have broader knowledge compared to computer tutors as they only know information related to the tutorials.	Human tutors don't tend to show their knowledge during tutorials. And if they show, it doesn't seem to cause larger learning gains.
Motivation	Human tutors praise may urge students to engage more and thus increases learning. In contrast computer's praise doesn't have effects on learning.	It is not clear exactly what the effect of human praise on learning.
Feedback	Tutors respond immediately to students' progress. They help students if they are stuck and encourage them to explain and correct their reasoning.	This hypothesis seems like a valid explanation of why human tutors outperform computerized tutors.
Scaffolding	Helping students by guiding them along the same way of thinking. Tutors make the students find connections and solve problems by extending their reasoning.	Scaffolding is an effective instructional method that can also explain human

The above table lists different attributes that need to be adopted by human tutors or embedded in computer tutors to ensure an efficient way of student learning. It has been stated that out of the eight attributes two of them need to be of interest and need improvement in computerized tutoring systems namely timely feedback and scaffolding.

2.3.4.2 Techniques for ITS

There are two fundamentally different types of instructional (or selection) models for intelligent learning systems that is: rule-based and algorithm-based. Some of the algorithms used in adaptive learning are:

- Variations of Bayesian data analysis, which involves the calculation of conditional probabilities.
- Bayesian inference networks (known as Bayes Nets) or slightly simplified classification systems called Naïve Bayesian analysis.
- Knowledge Tracing techniques and Markov Chain analyses.
- And for adaptive assessment systems uses a method referred to as Item Response Theory (Oxman al., 2014).

Research involving tutoring systems is an old field, and the work has covered several aspects of the intelligent interface which can manipulate natural language for easy development of tutors (i.e. Authoring tools). Now even individual schools and individual teachers can create adaptive learning using Authoring tools. Some of the tools that can be used for authoring tutors are:

- Cognitive Tutor Authoring Tools (CTAT) from Carnegie Mellon. CTAT make it easier to create Cognitive Tutors using two interface flash or Java. If much adaptation is needed experts can use Java for building the tutors, which is difficult for non-programmers. (ctat.pact.cs.cmu.edu)
- Generalized Intelligent Framework for Tutoring (GIFT) (gifttutoring.org).

2.3.4.3 Generalized Intelligent Framework for Tutoring (GIFT)

GIFT can work as a framework for authoring new ITS components, methods, strategies, and whole tutoring systems in addition to other functionalities namely: instructional management, and analysis (Goldberg and Hoffman, 2015).

To adaptively select learning content for users GIFT supports the Engine for Macro and Micro-Adaptive Pedagogy (eM^{2}AP) framework as shown in Figure1. State of the user is combined with the current quadrant (i.e., rules, examples, recall, and practice) to identify the attributes leading to the next quadrant, these attributes are then used to find the best match selected, resulting in the presentation of the associated content to the learner. GIFT will maintain a hierarchical representation of the course concepts, in which each level in the concept tree is represented by one or more units of contents for each node. When each level is covered a quiz must be taken by users to check on learning (Sottilareet al., 2013).

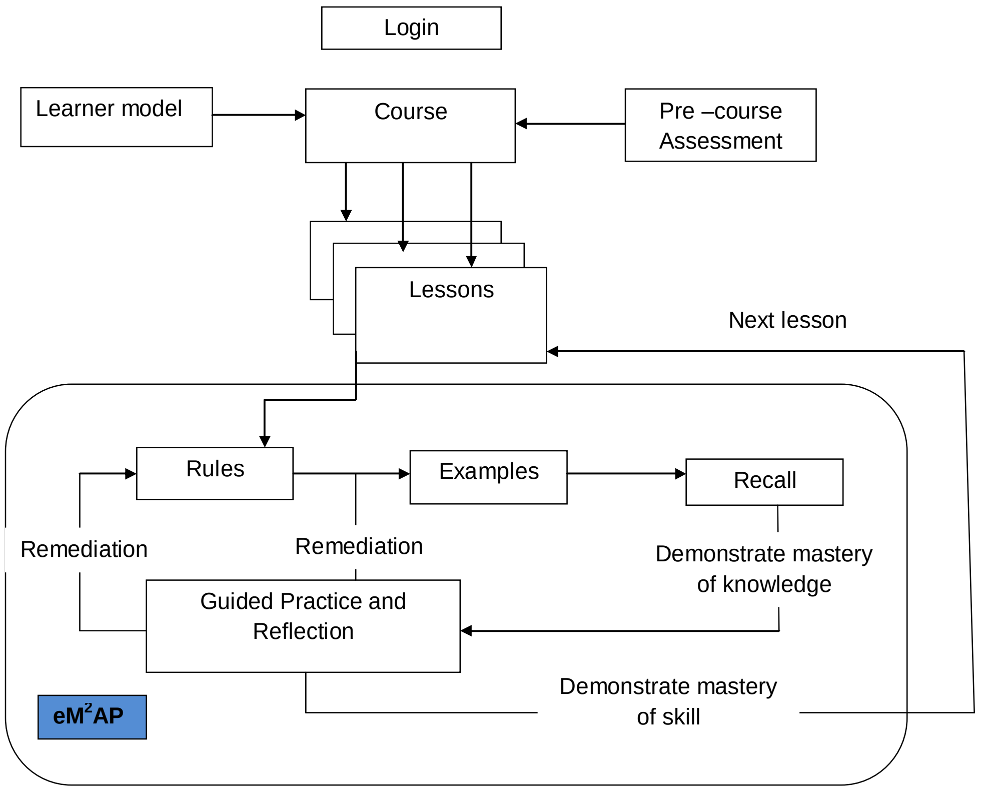


Figure 2-1: Macro-Adaptive Strategies in GIFT (Sottilare et al., 2013)

2.3.4.4 Operational Settings

Several technologies exist and for selecting a technology the developer has to consider the domain the setting and the data available (materials, info about the students, and usage data). It is even possible for non-programmers to create tutors with the help of tutoring authoring tools. When building a tutoring system, the technology selected for developing the system must be selected based on the data available at hand. Some methods for building tutoring system require huge data of students' interaction with the system (e.g. Using data mining techniques). If big data is not available on hand some assumptions regarding the students and educational theories must be employed.

Some of the related area of research tried to identify students’ behavior while interacting with the system. Determining the students’ engagement level can help in adapting the tasks. For example, if a student is bored by easy tasks presents more challenging tasks, if a student felt frustrated by difficult task present easier ones. Observing students' engagement can be obtained from analyzing students' features (visual and audio) or their keystroke patterns. Since there is a range of solutions for the different technological aspects, the challenges is of structural and operational nature (Oxman et al., 2014).

Researchers are encouraged to propose new ways of integrating technology in education and to overcome some of the operational challenges. For example, some of the obstacles when using ITS are: how can it be used with regular courses. Can certificate be given based on course completion? And since the speed of the students differ in these systems what is the implication of finishing the syllabus fast and successfully? Will the learner be exempted from taking the formal course? (Oxman et al., 2014).

2.3.4.5 Evaluation Metrics for ITS

To validate the results of the systems used in education, measurable criteria must be obtained. Those are the evaluation metrics for the systems. In the following a list containing some of the evaluation metrics used to evaluate ITS (Kristen ,2012):

- Active engagement
- Adapting the system to users’ needs
- Immediate feedback
- Detailed assessment
- Mastering concepts and – Levelling up
- Grades
- Time spent while using the system
- The statistics about users’ activities.

The learner is the center of attention and the system should try to engage the user and fulfil his needs. One factor which might be missing in ITS evaluation is the user enjoyment.

2.3.5 Adaptive Hypermedia Systems

In an adaptive system, minimal number of components should be displayed to the user to avoid overloading the user (cognitive overload) (Baig, 2014). Developing Adaptive Systems focuses on two major aspects; namely:

a) The structure of the system
b) The content to be displayed

2.3.5.1 Adaptivity and Adaptability

Both adaptation and adaptivity are based on the users' interests and preferences. Adaptable interface is about following the choices of the user in regard to the appearance of the interface or locating site content based on users' **preferences (explicit)**. To implement an adaptable interface, user choices are remembered and it is registered as part of the user model. While adaptivity is the *ability of the system to recommend to a user educational mediatic object believed to be of her/his* ***interests (implicit)***. Adaptive system records and make choices based on the interaction with the interface and monitor selective actions performed by the user and hence use it to make recommendations in the future. Adaptivity depends on user's interests or level while adaptability depends on user's preferences about the location of educational objects. The work of (Rodríguez and Ayala, 2012) is proposing the ADA+ALOI architecture (Architecture for the Design of Adaptivity + Adaptability of the Learning Object Interface) for the design of adaptive and adaptable learning objects interfaces.

Another architecture is (Educational Adaptive Hypermedia Applications (EAHA)) that displays personalized content to individual learners and adaptive sequencing (navigation) over the learning content based on user model requirements and the instructional strategies (Retalis and Papasalouros, 2005).

2.3.5.2 Users Modelling Attributes

A common feature of various adaptive Web systems is the application of user models (also known as profiles) to adapt the systems' behavior of individual users. As mentioned in 1.4, in order to have adaptive system knowledge about a set of information about the user is needed which will help in deciding the appropriate set of actions. According to (Koch, 1998), the following user's characteristics were identified as a leading attributes in modelling users:

- Knowledge
- Preferences
- Navigation abilities

The paper (Brusilovsky and Millán,2007) reviewed user models and user modelling approaches applied in adaptive Web systems. The following key questions were investigated: what is being modelled, how it is modelled, and how the models are maintained.
Five main features were found to describe users, specifically:

1. The user's knowledge: This is a feature that changes with the time as it can increases with learning and decreases by forgetting. This is the most important feature for educational systems and it can be the only modelled feature in some systems,
2. Interests,
3. Goals,
4. Background (stable features) extracted through interviews or tests.
5. And individual character represented by user features that define a user as an individual. These features are stable and they are extracted through specially-designed psychological tests. The features that could describe personality are: **character** (active/passive), **cognitive style** (holistic/ serialist), **cognitive factors** and **learning style**. Mainly, researchers use cognitive styles and learning styles *to* **model** an individual character on adaptive systems.
6. Context of user's work

Web-based adaptive educational systems (AES) work mostly by utilizing user knowledge and learning goals capitalizing on the modelling and representation techniques established in the field of Intelligent Tutoring Systems (ITS). Adaptive information systems and Web recommenders focus on modelling the user's interests and extend modelling approaches originally developed for adaptive information retrieval systems.
Meanwhile, adaptive hypermedia systems attempt to represent and employ an even wider range of user features. In addition to user knowledge and interests, these systems frequently model user goals, individual traits, and the context of user's

work.The following figure represents user features or attributes that are typically modelled by different classes of adaptive web systems.

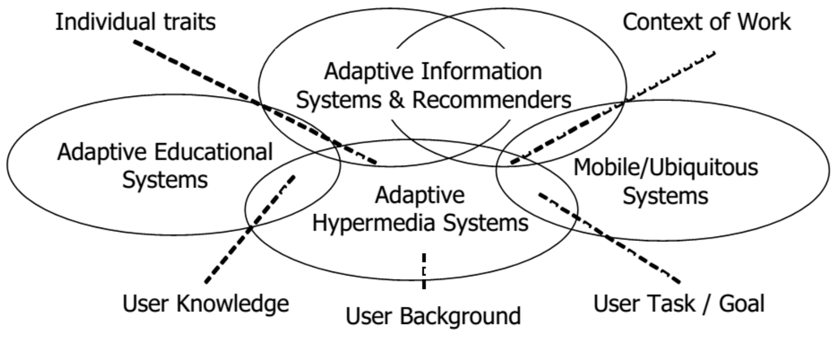

Individual traits
Context of Work
Adaptive Information Systems & Recommenders
Adaptive Educational Systems
Adaptive Hypermedia Systems
Mobile/Ubiquitous Systems
User Knowledge
User Background
User Task / Goal

Figure 2-2: User features typically modeled by different classes of adaptive web systems. Adapted from (Brusilovsky and Millán,2007)

2.3.5.3 Additional aspects used for user modelling (Emotional changes)

The work of (ZatarainCabada et al., 2015) presents Java Sensei, an Intelligent Learning Environment (ILE) for learning Java programming. The ILE is formed by an affective tutoring system working in a Web environment. The tutoring system was implemented under different learning methodologies like problem-solving for the pedagogical model, knowledge space for the expert module, and overlays for the student module. The main contribution of this work was the inclusion of emotion in the systems to be able to reflect empathy to the student. The following actions of the tutor are similar to the emotional response of a human tutor and represent the output of the fuzzy logic system:

- Feedback: Evaluating the students' academic progress and give a suitable description.
- Empathetic and emotional responses: trying to converse with the students after recognizing their emotional state. For example, encouraging them.
- Facial expression: imitating the human tutor by performing similar human expressions.
- Intervention: The Pedagogical Agent need to decide the need of an intervention or not to the student.

2.3.5.4 *User Knowledge Models for Adaptive Hypermedia and Adaptive Educational Systems*

Four types of user knowledge models identified by (Brusilovsky and Millán, 2007) are:

1- **Scalar is modelling /Stereotype knowledge**: Aims to divide users into scalar groups, e.g. beginners, intermediate, and expert. It has a limitation since the knowledge of the user can vary among different concepts, e.g. beginner in a topic and expert in another. Sometimes these systems can do averaging to calculate user knowledge.

2- **Overlay modelling**: to overcome limitations of scalar modelling, structural models is used in which the body of domain knowledge can be divided into certain independent fragments. One of the well-known types of structural modelling is overlay modelling. Since the overlay model represents the user's knowledge as a (weighted) subset of expert knowledge, the nature of the user knowledge reflected in the overlay model depends on the nature of the expert knowledge represented in a specific system. The following graph shows a representation of knowledge in as a related concept or topics covering some domain.

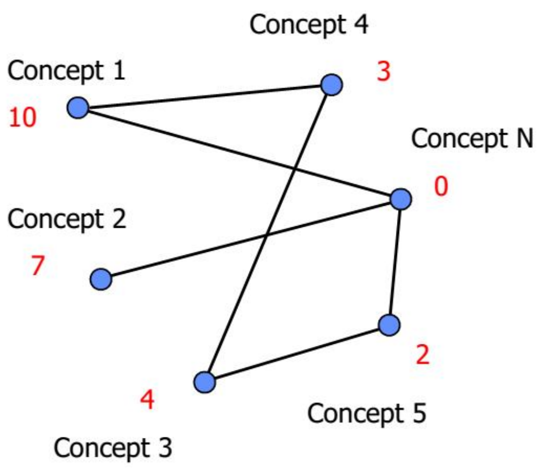


Figure 2-3: A network domain model with a simple numeric overlay user model. Adapted from (Brusilovsky and Millán,2007)

"All kinds of links between concepts are used to improve the precision of user modelling. When the user demonstrates a lack of knowledge, links can help to locate the most likely concepts that will remedy the situation."

Links between concepts are manipulated as students' knowledge changes. In problems that cover related concepts, if a student shows lack of knowledge the source of the problem will be investigated from all the related concepts and a concept with fewer connections to the well-known can be identified as a source of the problem. A recommendation would be given to strengthen the concept with fewer links. On the other hand, if the student shows presence of knowledge all the concept and all the related will be updated to reflect the current knowledge (knowledge propagation).

3- **Bug models**: allow the systems to recognize misconceptions in the users' problem-solving knowledge, distinguish it from random slips, such as typos and calculation errors, and provide a useful personalized explanation. The goal of a system with a bug model is not just to declare that a specific element of domain knowledge is incomplete or missing, but to identify, if possible, specific buggy knowledge that can be used to provide a higher quality adaptation.

4- **Genetic model**: An even richer model that makes it possible to reflect the development (genesis) of user knowledge from the simple to the complex and from the specific to the general is known as a genetic model.

2.3.5.5 A statechart-based model for hypermedia applications (De Oliveira et al, 2001)

The use of statecharts has been proposed in (De Oliveira et al, 2001) for modelling of hypertext and web based applications.

This paper presents a formal definition for HMBS (Hypermedia Model Based on Statecharts). HMBS utilizes the structure and execution semantics of statecharts to determine both the basic association and the browsing of hypermedia applications. Statecharts are an extension of finite-state machines. HMBS can model hierarchy and synchronization of information; provision of mechanisms for specifying access structures, navigational contexts, access control, multiple tailored versions, and hierarchical views. By analyzing a statechart machine page reachability can be detected, valid paths, and other properties. It is therefore helpful in the development of structured applications. The following figure shows a sample statechart diagram for creating an adaptive hypermedia system.

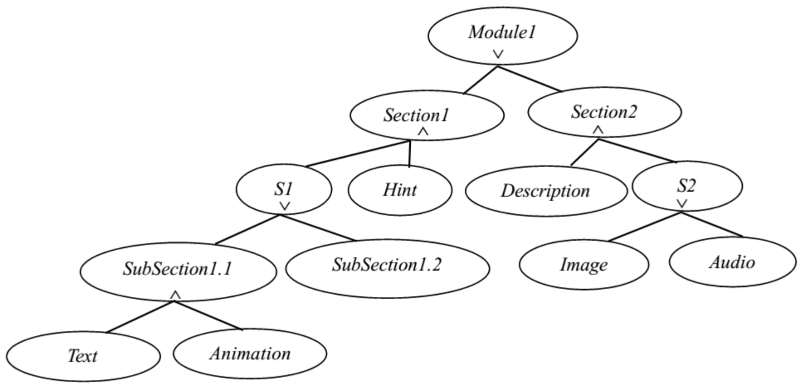


Figure 2-4: AND/OR statechart tree. Adapted from (De Oliveira et al, 2001)

By transitioning in a statechart diagram the state of the nodes will change (disable/enable) according to your current location in the diagram. As users move from one node to another the state will toggle at some nodes as an effect of the transition. And thus a new current state configuration is obtained.

2.3.5.6 Adaptive System Framework and methodologies

To support the development of AH applications an engineering approach is required that considers user modelling and adaptive aspects. This paper presents a Methodology for Adaptive Hypermedia Systems Development (AHDM) covering the whole life cycle of AH applications. It includes phases for the development of the user model, the adaptive interface and the dialogue component responsible for the modelling of the user’s behavior. For each analysis and design phase an appropriate notation is proposed.

Their methodology: Adaptive Hypermedia Systems Development (AHDM) is composed of the normal software design steps with extra details on the analysis and design steps.

1. Analysis:

   a) User analysis: The aim of this step is to obtain the information needed to build an appropriate user model for the AHS. Therefore, it has to be determined how the user's behavior is captured by the system and how the user model is to be adapted dynamically to this behavior
   b) Requirement analysis
   c) Strategy planning
2. Design : In the design steps all the following components need to be decided:
   a) User model design
   b) Conceptual modelling
   c) Navigational design
   d) Abstract interface design
   e) Dialogue modelling

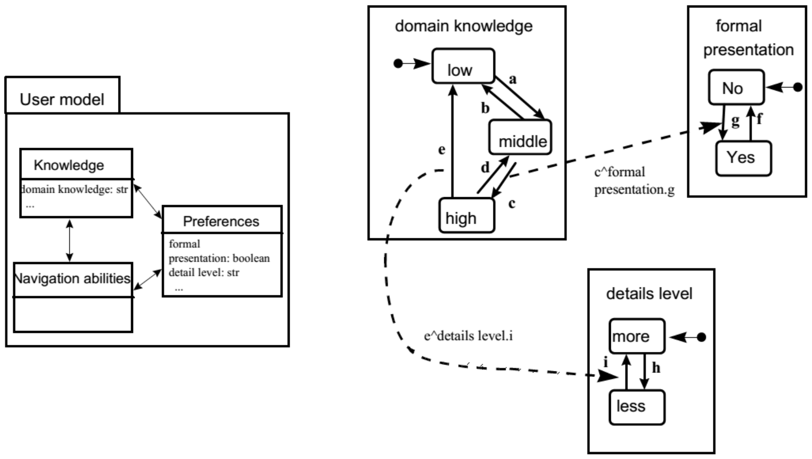


Figure 2-5: Partial representation of user's model (Koch, 1998)

These steps of AHDM describe the life cycle of AHS and are performed in a mix of incremental, iterative and prototype-based development style.

The work in (Retalis and Papasalouros, 2005) presents an EAHA that is built using a model-driven design process. The adaptive aspect or conditional changes of the presentation of the resources in the system are modelled using the attributes: the user interaction with the system and the learner type. A framework for authors of hypertext applications was used to design of hypermedia applications. The design of the application was performed in three stages: conceptual, navigational and presentational which are related to adaptation, navigation and presentation.

A reusable framework could make developing adaptive hypermedia systems easier. Currently there is no standard framework that is acceptable in developing adaptive systems and the work by in (Martin and Ivan, 2013 ;Knutovet al. , 2011) tried to suggest a standard.

The steps used in (Martin and Ivan, 2013) aimed to set a standard and to formalize the adaptive system architecture. A reusable adaptive web user interface components were developed. In the user model part the attributes were divided into user profile (page setting preference) and user model (usage data). The information in users' models is: e.g. users' knowledge of the topic, users' preferences, or their past experience.

Another framework suggested by (Knutovet al., 2011) named Generic Adaptation Framework (GAF) which was developed to advice in the process of creating adaptive systems. In following figure the evolution of the Hypertext reference models, from Hypertext to the new Generic Adaptation Framework (GAF); that contains recent advances of adaptive system is shown.

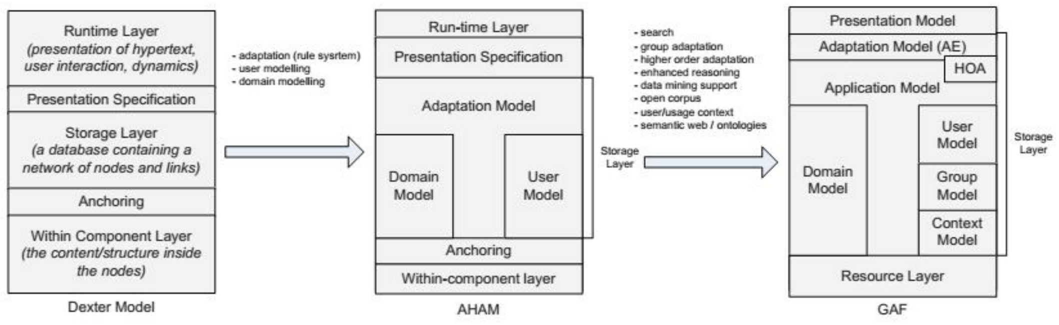


Figure 2-6: From Hypertext (Dexter), through AHAM, to GAF. Adapted from (Knutovet al. , 2011)

Additional attributes/adaptation rules were used in the GAF framework that represents the user, including: behavior, knowledge and interest.

In the work of (Mezhoudiet al. , 2015) a theoretical framework integrating agile practices for UIs runtime context-awareness is presented. The framework outlines a flexible life cycle of agile adaptation maintaining six dimensions to identify adaptation features; namely: To what? When? How? Why? And where?.

One of the features of agile development is that it allows iteration in building systems, which is a desirable feature in adaptive systems. Adaptivity is built based on user feedbacks from usage of systems that allow evaluation, and update of rules.

When to adapt: a sensor can obtain information about the user that influences the choices of the responses. For example a sensor about the location can make the system to guide or give a recommendation to the user to the service he needs (e.g. restaurants, car rental agencies, hotels, stations).

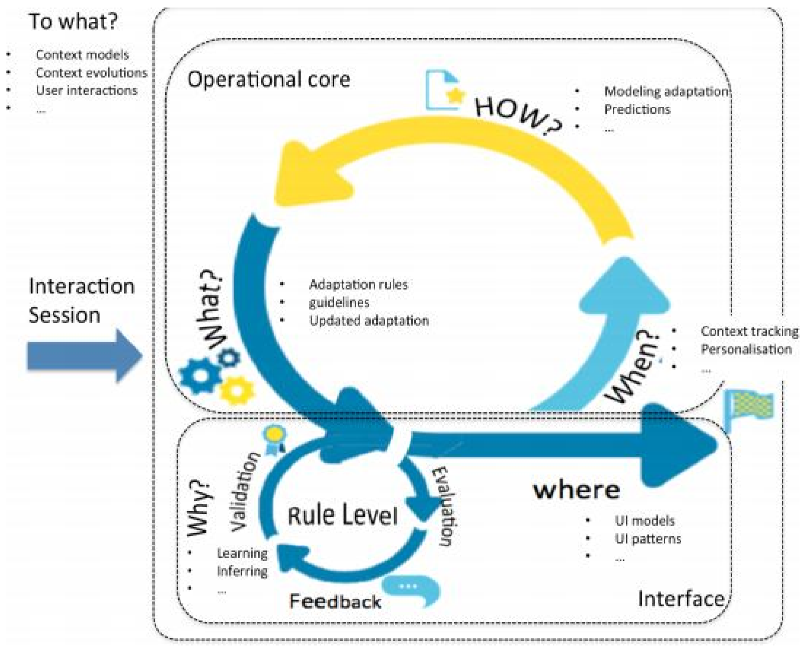


Figure 2-7: Agile runtime adaptation lifecycle. Adapted from (Mezhoudiet al. , 2015)

2.4 Literature Review on Measuring Users' Engagement

2.4.1 Introduction

User engagement in using a particular system is only applicable to situations in which using the software is optional. Many of the criteria mentioned in this section will not reflect user desire to use the software when the user is forced to use the software in any way.

Users' engagement on using systems is one aspect that defines software usability and users feeling when using a particular solution. The measurement of user engagement can be of many forms (Brown and Howard, 2014)

1. Sensory measures (invasive and non-invasive): This type of measures has the risk of users changing their behavior because of their awareness that they are being observed. In addition, when this data is analyzed automatically it can be computationally heavy.
2. Usage analytics: These types of measures are easy to process such as time spent on task, navigation, keystrokes, and mouse movements.
3. Quantitative and qualitative surveys: users' feedback about their experience of using the system.

2.4.2 Measuring User engagement in web based systems

Students' activities from log data could be collected and analyzed to obtain useful information about their behavior. Deciding which data to be collected depends on the purpose for studying the data. In a study in (Dobashi, K., 2015), in order to obtain useful data for class improvement, the researchers collected data from course online system regarding the use of the online materials in the weekly class such as date ,time, student ID number, viewing time, and page view history.

Another study (Mahajan, R. and Mahajan, V, 2014), analyzed and presented attributes of how learners interact with an e-learning site. The attributes aimed to measure personal acceptance of e-Learning and the engagement. Namely the attributes were: frequency of access, and attributes influencing learners' satisfaction (it increases by interaction with instructors, promote feedback from instructors, and ease in seeking clarification).

There are several methods for measuring user engagement, including: self-reporting methods, physiological approaches and web analytics. Web analytic is concerned with measuring attributes about visiting a web site such as (Lalmasetal. ,2014):

- Click-through rates
- Number of page views
- Time spent on site or dwell time
- Frequency of return visits (during single or multiple sessions)

Additional web analytics attributes are (Dupret and Lalmas , 2013) :

- Time between visits or absence time
- Number of tasks
- Reading amount

In an attempt to improve online distant students' opportunities, the research in (Goldingay and Land , 2014) tried to motivate them to practice the required skills before the practicum by modifying the online course contents. Methods that were used to increase students' engagement and to improve the learning process were a combination of video-based content delivery and on-going formative peer- and self-assessment. The resulting effect was measured using web analytics and student evaluation survey. Results from web analytics showed that there was a lack of change in access to the modified course content. Initially there was greater access showing initial interest in this content followed by a similar decline in access of course materials.

After first creating a scale to measure online student engagement, and then surveying 186 students from six campuses in the Midwest, the results indicate that there is no particular activity that will automatically help students to be more engaged in online classes. However, the results also suggest that multiple communication channels may be related to higher engagement and that student-student and instructor-student communication are clearly strongly correlated with higher student engagement with the course, in general. Thus, advice for online instructors is not only to use active learning but to be sure to incorporate meaningful and multiple ways of interacting with students and encouraging/requiring students to interact with each other (Dixson , 2012).

2.5 Conclusion

As presented in section 2.2.4 (Educational Practices that Contribute to Student Engagement), in order to improve programming learning several factors should be considered. In addition, the attitude of both students and staff could be modified to result in a higher engagement level. Engagement in learning leads to desirable learning outcomes.

From the literature the following: learning strategy, collaboration, self-efficacy, time availability, interest, support, scaffolding, availability of devices and resources and comfort level were identified as the leading attributes that affect engagement in programming. A good Human tutor helps in engagement since he can give timely feedback that will reduce frustration and help student to understand the topics better. Also a good tutor will force students to dedicate time and keep moving, forces students to collaborate and find ways that will engage and excite students. Unfortunately, one to one tutoring is not feasible in most of the current higher education settings. This is fairly applicable for a small number of students. And usually in larger class lecturers might not be able to identify students' knowledge gaps since many students are passive and there is no way to converse with each of them.

Research is still needed to find out whether it is the college experience or the student's ability that shapes grades, although there are many studies linking student engagement to academic programming requires intensive effort and time to be allocated by students in order to understand the content and to be able to solve novice problems. This can only be achieved when students are engaged. Maximizing student engagement is critical to achieving many of the educational outcomes.

Several IT solutions exist that could help in proving a quality learning experience and improve engagement. Such as: (1) recommender system, (2) adaptive systems and (3) intelligent tutoring systems that can help in increasing engagement in programming learning. The following list is a desirable outcome of these systems:

- Virtual one on one tutoring.
- Make students active.

- Converse with each of them.
- Give timely feedback and help.
- To force students to dedicate time and keep moving.
- Help each student to work on their own speed; weak students can revisit materials and stronger students get more challenging exercises.
- Provide a way of practice problems solving on the algorithmic level and allow students to test their proposed solution.
- The system should also help students in recalling statements/language syntax.

From the range of available technologies for tutoring adaptive tutors are easy to implement and it overlay with the research objectives. Personalized contents based on students' attributes will be further investigated in this research.

As mentioned in chapter one, the objectives of the research is to influence (enhance) and monitor students' engagement for students while learning programming. The influencing part will be performed by introducing and affecting students' attributes on different iteration of system implementation.

To monitor students' engagement while using the adaptive system, analyzing web system logs will be performed. From the literature the following measurements are used to measure users' engagement: Click-through rates, Number of page views, Time spent on site or dwell time, Frequency of return visits (during single or multiple sessions), Time between visits or absence time, Number of tasks, Reading amount.

The following chapter (CHAPTER THREE: METHODLOGY) will set the layout of the work and provide in details the steps used in finding and refining the techniques and the attributes related to students' engagement in programming learning. The research methodology followed in this work is Design based Science Research which iteratively finds solution to real world problems.

CHAPTER THREE: METHODLOGY

3.1 Introduction

The research methodology that was followed in this work is design science research (DSR), which is discussed in the following sections. DSR is a research framework for addressing research that tries to deal with real problems. The problem which is addressed in this work is students' disengagement in programming learning. One of the objectives is to find the attributes that affect this problem the most. The steps listed in this work are followed from the research methodology; design based science research. The technical solution that would be suggested should be up to date scientifically and address the problem effectively.

In this work design science research (DSR) methodology will be followed to achieve the objectives. DSR involves working with people to find rigorous IS/ IT solution to real problems. In order to follow DSR framework, researchers could follow the listed guidelines as presented in section 3.2. In addition, the DSR process has an iterative nature in general and the process can be summarized according to the work in Vaishnavi and Kuechler (2008) into 5 iterative steps as shown in section 3.3. Also DSR steps could be arranged in 3 cycles as seen in section 3.4.

3.2 Design Science Research (DSR) Guidelines

In the design-science paradigm, knowledge and reasons of a problem knowledge domain and its solution are perceived through construction and practical application of the designed artifact. The work in (Hevner, et al. , 2004) presented a framework and clear guidelines for understanding, executing, and evaluating the design-science research. The seven guidelines for following design science research are as follows:

3.2.1 Design as an artifact

"Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation." (Hevner, et al. ,2004)

At Sudan University (SUST), programming is a fundamental subject to computing students, which is usually taught during their first year of study, and that will affect their entire subsequent studies. Staffs teaching programming subject are concerned because some students seem to be disengaged. Disengagement leads students not to dedicate all the possible time they get in performing programming learning related tasks. Therefore more research needs to be done to gain insight about ways to increase students' engagement in programming courses.

In this work there are 3 main artifacts:

a- **The construct language (Engagement attributes):** This was obtained and summarized from (CHAPTER TWO: LITERARTURE REVIEW - Factors Affecting Students’ Engagement) and from the results of a survey in (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT).

b- **The model:** shows general guidelines for implementing an engaging adaptive system for learning and the model is presented in (CHAPTER FIVE: THE MODEL).

c- **The instantiation (DrSust):** implementation of the model was performed in several iterations as demonstrated in (CHAPTER SIX: DR SUST).

3.2.2 Problem Relevance

“The objective of design-science research is to develop technology-based solutions to important and relevant business problems.” (Hevner, et al. ,2004)

As shown in CHAPTER ONE: INTRODUCTION the problem of disengagement in programming learning was addressed and it was shown that struggle with this course will causes computer related studies students to face difficulties in the rest of their studies. The problem of disengagement is addressed in learning since it is a key attributes that affect the learning experience. There is a lack of studies regarding engagement in technological solutions. What to integrate to make a technology enhanced learning solution better and how to measure students’ engagement while using the solution.

Section 2.2 presents the engagement concept and the attributes that could influence it. It also addresses the current challenges faced when teaching and learning programming. For that this section represents the first stage of DSR (Literature research—part I).

Prior to finding a proper technical solution to increasing engagement in programming learning, the current educational solutions for providing personalized learning that is sensitive to students' attributes is investigated. In section 2.3 adaptive techniques for personalized learning and the different stages in implementing them were reviewed (Literature research—part II). The design of the adaptive system that would adhere to the finding of CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT and incorporated the technical aspects from current practices to creating personalized learning in the Literature Review of Adaptive Systems and the development detail was shown in CHAPTER SIX: DR SUST . The results extracted from usage data of the system was listed in CHAPTER SEVEN: RESULTS and ANALYSIS.

3.2.3 Design evaluation

"The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods." (Hevner, et al. ,2004)

After developing and running the solution the system is evaluated in (CHAPTER EIGHT: CONCLUSION)CHAPTER EIGHT: . Evaluation was achieved by showing applicability in practice. The log data of the system usage were further analyzed to find the activities that were more engaging for the students. As mentioned in 2.4.2 the measurement that can indicate users engagement with the system are: Click-through rates, Number of page views, Time spent on site or dwell time, Frequency of return visits (during single or multiple sessions), Time between visits or absence time, Number of tasks, Reading amount. In addition, for students as users additional attributes are involved:

- Syllabus completion
- Questions made
- Comments made
- Participation and discussions

3.2.4 Research contributions

"Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies." (Hevner, et al. ,2004)

The overall purpose of this research is to increase students' engagement in programming learning. The contribution of this research is a deeper understanding of the phenomena of engagement in programming learning in the form of the three artifacts mentioned in section 3.2.1. One of the artifacts was the development of an adaptive system that meets the current students' needs as a mean of increasing students' engagement in programming courses.

3.2.5 Research rigor

"Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact." (Hevner, et al. ,2004)

From the literature the following: learning strategy, collaboration, self-efficacy, time availability, interest, support, scaffolding, availability of devices and resources and comfort level were identified as the leading attributes that affect engagement in programming. A good Human tutor helps in engagement since he can give timely feedback that will reduce frustration and help student to understand the topics better. Unfortunately, one to one tutoring is not feasible in most of the current higher education settings. This is fairly applicable for a small number of students. Research is still needed to explore the best ways of engaging students in different subjects and the effect of the several solutions on engagement. This work focuses on increasing engagement in programming learning; since programming requires intensive effort and time to be allocated by students in order to understand the content and to be able to solve novice problems. Several IT solutions exist that could help in providing a quality learning experience and improving engagement. Such as: (1) recommender systems, (2) adaptive systems and (3) intelligent tutoring systems that can help in increasing engagement in programming learning.

Presenting a model and an instantiation for an adaptive system with the focus of increasing students' engagement is presented in this work. The results of analyzing usage data from different iterations of the instantiation (DrSUST) will be compared to articulate improvement.

3.2.6 Design as a search process

"The search for an effective artifact requires utilizing available means to reach the desired ends while satisfying laws in the problem environment." (Hevner, et al. ,2004)

From the literature, it has been noticed that technologies that provides tailored (one to one) tutoring can be utilized to meet the needs of students and hence increase engagement. Students' requirements that need to be met and changes in the course was perceived from the students' survey that was presented in (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT).

3.2.7 Communication of research

"Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences." (Hevner, et al. ,2004)

Discussions with colleagues about the usefulness of an engaging application were performed. In addition, the research ideas and results were presented at 2 conferences (Technical (ICCNEEE), 2015 International Conference on (pp. 101-106)) and educational (SUST first education conference)). The research papers produced that describe this work and that were meant to communicate this research can be found in appendix B.

3.2.8 Conclusion

Section 3.3.1 to section 3.3.7 presented the guidelines for implementing DSR framework in the current research and showed how this research adhered to each of the seven guidelines ensuring that this research follows an accepted methodology.

3.3 DSR Process

A model adapted from the design process model developed by Takeda, et al. (1990) is presented in this section. Design process and design science research share the different phases, but the activities performed within the phases are considerably different. The key difference between the standard design process and the design science research process is the need to focus on knowledge contribution. The following figure shows the research process model. (Vaishnavi and Kuechler , 2004)

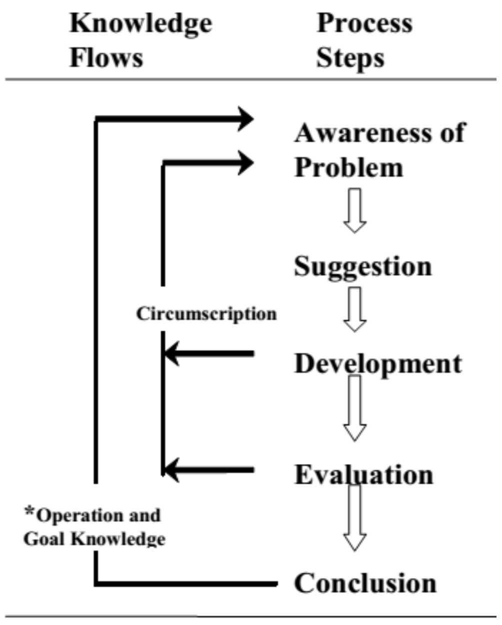


Figure 3-1: Design Research Iterations(Vaishnavi and Kuechler , 2004)

The following table demonstrates the process steps that were implemented in this work.

Table 3-1: Design Research Process

Process steps	**Output**
Awareness of the problem	Proposal: In (CHAPTER ONE: INTRODUCTION), the special nature of programming learning (1.2Programming Teaching and learning) and the importance of students' engagement in learning (1.3Students' Engagement and 2.2Literature Review on Students' Engagement) were introduced. This required the researchers to investigate the current trends in using technologies to enhance students' engagement in learning and the details of the review were presented in (2.3Literature Review of Adaptive Systems).
Suggestion / Prototypes	Quantitative and Qualitative Survey was performed and data were gathered from 100 students who studied the

	introductory programming course. The outcome of the survey demonstrated needs and additional attributes that need to be considered in the design. Details of the survey are presented in CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT. Also, preliminary prototypes were presented to the students to investigate the applicability of the proposed research. In chapter 7(section 7.3 DrSUST 0.0: Chatting Sessions as a Way of Observing Usefulness of Group Study)
Development	Artifact As mentioned in section 3.2.1 (Design as an artifact) 3 artifacts constitute the contribution of this research, namely: a- The construct language (Engagement attributes) b- The model c- The instantiation (DrSust)
Evaluation	Evaluation was split into two parts: 1. Experiments with the students in the developed adaptive system and performance measures were used to compare the improvements throughout the different iterations in the developed system. 2. And presenting and discussing quantitative and qualitative analysis with some of the users and experts.
Conclusion	Requirements were perceived from students' needs and personalized systems survey and several factors that can increase engagement were identified. In building the system students' engagement attributes were divided to generic attributes which were used as guidelines when implementing the system, and attributes that are part of the students' model.

3.4 DSR Cycles

Design science research could be represented using three cycles of activities as shown in figure 3-2.

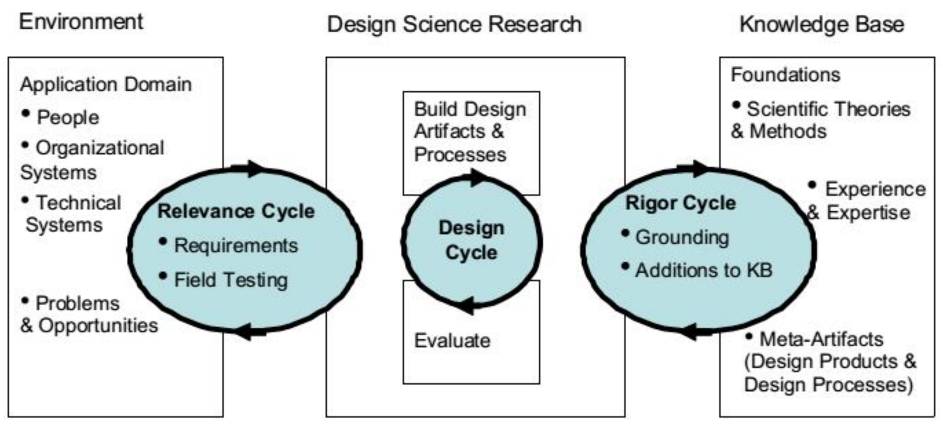


Figure 3-2: Design Science Research Cycles (Hevner, A.R., 2007)

Details of the above cycles and their implementation in this research are presented below:

a) The relevance cycle:

Research in design science involves working with people to solve real problems and it also work to finding areas and problems in current applications. The relevance cycle is the start in DSR as work initiates in an application context to obtain the requirements of the research. Also an evaluation method for accepting the research results in the desired environment should be identified in this step. (Hevner, A.R., 2007)

In this research, the researcher has been working in teaching programming in higher education for many years. Frustration was observed among some of the students who feel left behind which will affect their entire study. Details of the disengagement problem are presented in CHAPTER ONE: INTRODUCTION. In addition to that, experiences to increase students' engagement were studied via students' survey were performed and the finding showed that the existing IT solutions were not meeting all the of SUST students' needs.

b) The rigor cycle:

The research should adhere to the rigorous research tradition in terms of its being innovative and to be linked to the past knowledge to the research project. Linking the research to existing research thoroughly and referencing the knowledge base would guarantee that the designs produced are research contributions and not routine designs based upon the application of well-known processes. (Hevner, A.R., 2007)

There are many iterations involved in developing technical solutions when following DSR. And hence the research work will develop and refine the proposed technical solution in iterations. Additionally, iterations are also involved in addressing literature covering several parts of the problem. A literature to identify the problem (Literature Review on Students' Engagement) and literature part2 to review the current methods of developing technical solutions and the gap that exist in developing the technical solution (Literature Review of Adaptive Systems) and ways of measuring engagement (Literature Review on Measuring Users' Engagement).

c) **The central Design Cycle:**

This cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research. (Hevner, A.R., 2007)

CHAPTER FIVE: THE MODEL lays down the broad activities for implementing an engaging tutoring solution that, meets students' needs. And the details of this research and the different cycles design cycles are presented in CHAPTER SIX: DR SUST .

3.5 Research Ethics

For all the iterations of the system, using the system was an additional resource for the course and it was not a mean to communicate with the specific course tutor. So using the system was completely optional and no marks would be awarded as a result of any use or submissions in the system. That is to say, the usage of the system and the supporting chatting sessions were made optional and no grades were given during the communication.

In addition, students sometimes feel hesitant to declare that they have technological access to online materials and that they can be available online. Because that might mean forcing them to allocate more time to the subject. And hence students were allowed to use the system anonymously.

3.6 Research Instruments

In order to engage students, the researcher had to interview them to identify their needs and what is their view about engaging elements in programming teaching and learning. A questionnaire was designed to obtain data from senior computing students who studied introduction to programming using Java at SUST. 100 senior students completed the questionnaire covering a range of possible attributes, details of this questionnaire and the results obtained is provided in the coming chapter (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT). Attributes related to students' engagement in programming course that were identified by the survey will be utilized in developing instantiation of the engaging model.

3.7 Research objectives

Implementation of the research steps to meet research objectives were demonstrated in the following table. Increasing students' engagement was performed via developing 2 artifacts; a model and an instantiation.

Table 3-2: Research objectives implementation

Research Objectives	**Research Output**
What are the attributes of students' disengagement in learning programming at SUST?	Attributes contributing to students' engagement were obtained from three sources: literature (Literature Review on Students' Engagement), students' quantitative and qualitative surveys (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT), and from

	evaluating the usage of the designed solution.
How can adaptive tutoring systems increase students' engagement?	Chapter five (CHAPTER FIVE: THE MODEL) provides technical details of the suggested technical solution. In chapter six (CHAPTER SIX: DR SUST) the design of the adaptive system DrSUST is presented. There were three cycles in developing this technical solution and in this chapter listed the distinctive features of every system's iteration.
How can students' engagement be measured?	Chapter seven (CHAPTER SEVEN: RESULTS and ANALYSIS) presented the evaluation of the students' engagement obtained from each of the systems' iteration and analysis of the findings.

3.8 Conclusion

In this work, personalized learning in the form of adaptive system will be implemented following the design science research process steps shown in Design Science Research (DSR). As mentioned in chapter one, the objective of the research is to influence (enhance) and evaluate students' engagement for students while learning programming. To monitor students' engagement while using the adaptive system, analyzing web system logs was performed.

Since the research involves a real problem, current situation needs to be studied in addition to covering the literature. The following chapter covers the aspect of working with people to identify their view of the current situation. Chapter 4 presents finding of students quantitative and qualitative surveys (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT) from which the findings will be incorporated into the designed solution. Also interventions that seems important from their point of view needed to be discovered and implemented.

CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT

4.1 Introduction

From the related work documented in chapter 2 (Literature Review on Students' Engagement), it is clear that students needed to be highly engaged and motivated in order to achieve richer programming learning experience. As stated by Trowler (2010), several stakeholders affect students' engagement in learning, including teachers, resources and most importantly students.

Chapter one stated the objectives of this research which aims to influence (enhance) and monitor students' engagement for students while learning programming. The influencing part will be performed by introducing and affecting students' attributes on different iterations of system implementation. Chapter two presented the literature in multidisciplinary fields that is required and connected to this research. The literature needs to cover several aspects. The first aspect is about the theoretical background of students' engagement that is required for understanding ways of addressing it later in the implementation. The second aspect is to investigate the technologies that can be implemented to help in individualized learning. And the last aspect is to find information about evaluating the students' interest or engagement on the developed solution. Chapter three (CHAPTER THREE: METHODLOGY) sets the layout for the work and provided in details the steps used in finding and refining the techniques and the attributes related to students' engagement in programming learning. The research methodology followed in this work is Design based Science Research (DSR) which iteratively finds solution to real world problems. In DSR the research should produce any or a combination of the following workable artifacts (Hevner, et al. ,2004)):

1- A construct
2- A model
3- A method
4- An instantiation

The construct language (Engagement attributes): This was obtained and summarized from (CHAPTER TWO: LITERARTURE REVIEW - Factors Affecting Students' Engagement) and from the results of a survey in this chapter (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT).

This is one of the artifacts that were mentioned in section 3.2.1. The results of this survey influenced the model - instantiation

In this chapter the outcome of the IT senior students' survey on learning programming is presented. From this survey the objective was to find what the students' needs are and how to increase students' engagement in learning to program. The summary of this survey can be used to provide students with a suitable learning environment that increases engagement and motivate students to invest more time in the programming learning. In the questionnaire senior students were interviewed to understand the problems and obstacles they faced when they learned programming and to get their suggestion regarding students' engagement in learning. The expected outcome should help educators to provide a supportive learning environment for their students.

4.2 Methodology and Analysis

Questionnaires were collected from 100 third year students who studied introduction to programming in their first year and they also studied additional programming courses in their second year. It is worth noticing that the nature of the programming subject enforces a greater level of collaboration between learners. Some of the questions in the survey were adopted from the Australian Survey of students' engagement in higher education (AUSSE, 2009) while other questions were specifically designed for this particular survey regarding programming learning. The questionnaire is attached in Appendix A (Appendix A: Students Needs in programming learning Survey).

To guarantee a higher response rate of the questionnaires the students were rewarded in a form of printed HCI subject lecture notes for the students responding to the survey. The questionnaire was also related to the HCI subject as it gave students an idea of designing the questionnaires for data collection and evaluation in addition a discussion about how to increase the response rates when performing surveys.

The following sections are arranged into three parts: firstly, the quantitative survey; Secondly, relation between some of the related questions and; Thirdly, The qualitative survey.

4.2.1 Quantitative Questionnaires

The quantitative questionnaire aimed to find answers about students and their characters (active/passive), their relations with staff, their studying habits, their collaboration, availability of devices, subject time quality, and their achieved level. Responses to questions about the mentioned programming learning attributes were collected and the analysis is presented below:

4.2.1.1 Class Participation

Students were asked about their participation in class "Do you ask questions or contribute to discussions in class or online?". When students were asked about their class participation, 61% responded sometimes. This shows readiness to participate. It is a good teaching practice if implemented, but it needs practice and management of class. This attribute could reflect students' engagement, although it is hard to allow fair participation in normal classes as noted by Kinzie (2010). It could be replaced by quizzes since fast feedback can be given to students.

4.2.1.2 Teacher Advice

In responses to a question about seeking advice from academic staff, the responses showed a high tendency to seeking advice. If students are seeking advice it means that in general they approve the teaching method and that they respect the lecturer. Only 16% never approached the lecturers for advice while the rest approached the lecturers at different levels.

In contrast, when students were asked about whether they discussed their grades or assignments with teaching staff, the responses were very low. This might show that, lecturers should be open to students about giving marks and discussing them so that students can work to improve it or maintain it. 55% of the students mentioned that they never discussed their marks with the academic staff while 33% reported a low discussion rate. There is a need from staff to be transparent in evaluation and marking. Students need to know how to improve their grades and what parts of the programming learning activities they should invest time in.

4.2.1.3 Relation with Staff

Another question was to investigate the students' thinking about their staff in respect to their readiness to help and interact. The question was "Which of these boxes best represent the quality of your relationships with people at your institution?". The student needed to give a range from 1 which indicates that the staffs were "Unhelpful, inconsiderate, rigid" to 7 which mean that the staffs were "Helpful, considerate, flexible". The responses were as follows:

Table 4-1: Relation with the Staff

	"Unhelpful, inconsiderate, rigid"	**==========➔**					**"Helpful, considerate, flexible"**
Strength	1	2	3	4	5	6	7
Frequency	11	26	10	13	5	6	31

The results were contradictory since the students were being taught by the same staff. The distribution is bimodal showing a high tendency to both clusters that classify staff as helpful and unhelpful. A possible cause for this is that for students to get help from staff, they need to have done their part of the work first. Teaching staffs are willing to help in filling the gap in knowledge given that students are not pushing them to do the work for them.

4.2.1.4 Timely Completion of Assignments

In response to the question about students' readiness when they come to class "Came to class having completed readings or solving assignments", the responses were as follows:

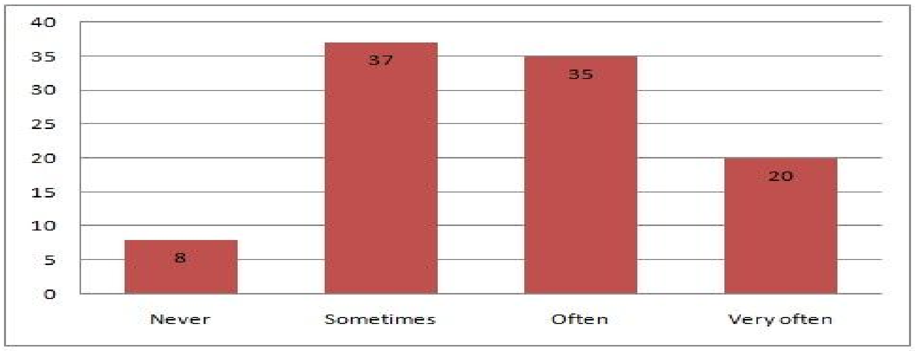


Figure 4-1: Timely Doing Assignments Response

Although Programming subject topics are tightly related and the materials build up quickly, still students showed that they are not doing their assignments timely. What could be the reason that 45% of the students are mostly not prepared for their class? Is the reason for the delay being one of the following: 1. The students inability to manage their time, 2- Students need pushing and reminders in order to meet deadlines, 3. Assignments difficulty, or 4. No devices available for students outside university.

4.2.1.5 Close Follow up

In a study by Rogerson and Scott (2010), data collected from students described that the programming concepts build up quickly and that can cause fear for students. When the students were asked whether they kept up to date with your studies, their responses were:

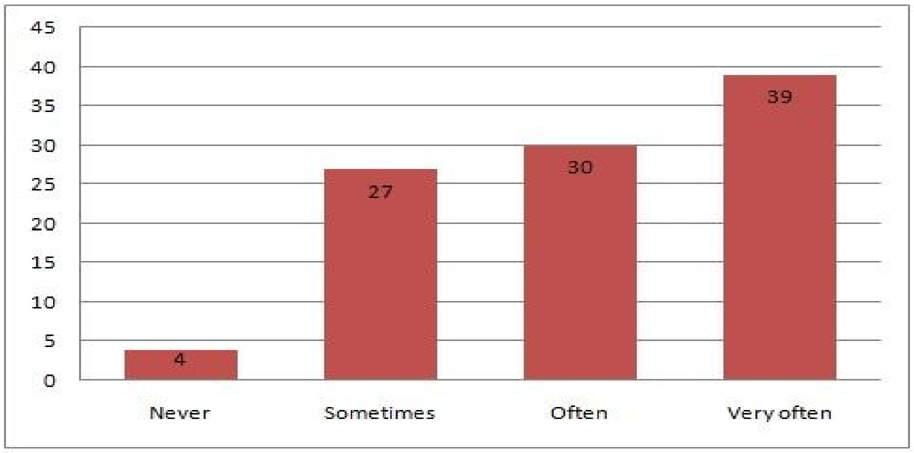


Figure 4-2: Subject Follow Up

Discussion with the students showed that, at the beginning of programming course, all students will be engaged and slowly some students feel frustrated because they cannot follow the base or they don't know how to develop the newly required skills like critical thinking and novice problem solving.

4.2.1.6 Collaboration

The following question aimed to find if students did some level of collaboration during their studies.

"In your experience at your institution during the current academic year, about how often have you worked with other students on projects during class? "

The responses to this question were depicted in the following figure:

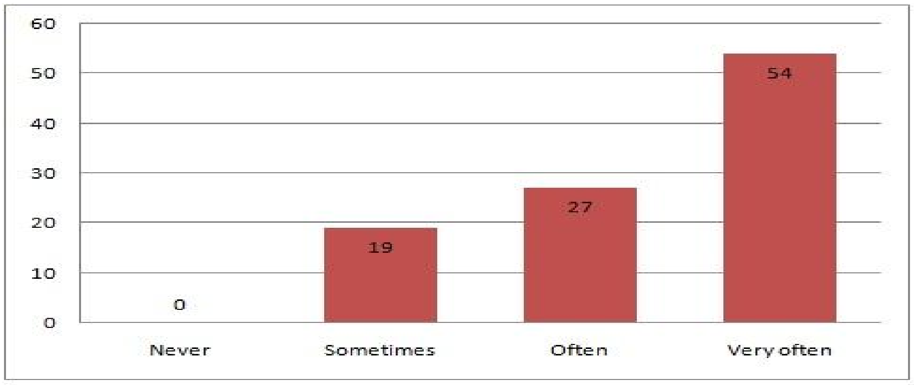


Figure 4-3: Frequency of Collaboration

It is realized that none of the students claimed to work alone on this subject and thus collaboration is needed to study programming. Frequent collaboration was the choice made by the majority of students.

4.2.1.7 Presenting Tutorials and Helping Colleagues

While there is a high level of collaboration in learning programming subjects, there is another level of interaction, in which students take the rule of the lecturer at times. Some students can explain solutions to some problems to their colleagues. It will be good to let students take an active role in teaching, and although it is related to increasing engagement practice, still there are some concerns about applying it.

For students to explain some sections for their colleagues, this is only possible in tutorial sessions. The risk is that the lecturer can lose control of the class, if not performed carefully it will be dominated by some students. Baccarani (2014) gave a suggestion for the cause of students being passive and that is due to the fear of being judged or not having something interesting to say. And this is consistent with the responses to this question here as only 9% of the students showed readiness to explain the programming concepts to their colleagues.

In courses at the university level when recorded materials are not provided to students, students have to make up for missed sessions by working alone or asking someone to help with the difficult parts. And since there is no way of revisiting lectures some students act as mentors for those who missed classes and students who needed revisiting of the explanation. Students also share their own summary of the topics with their colleagues.

4.2.1.8 Availability of Devices Outside University

Three questions were asked to identify the importance of having computer devices or practicing outside university. The questions were as follows:

a) Do you think it is necessary to practice programming out of university labs?

1-Yes　　　　2- No (lab time is enough)

Only 6% of the students thought that the time in university labs is enough for students to master the subject. There is a majority agreement that there should be devices available for students outside the university and some attributed their problems in learning programming to lack of practice.

b) In the survey students were asked if they could practice Java programming outside university.

5- Yes, I have a personal computer	**4- We have shared computer at home**
3- I can use my friends' computers	**2- I used to study with friends having computers**
1- No	

The responses were not as expected, as the number of students who didn't have access to computers outside the university was low. The responses were as follows:

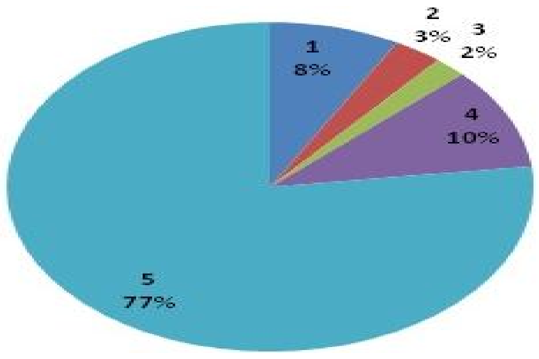


Figure 4-4: Owing Access to a Computer Device

Only 8% of the students stated that they don't have access to practicing Java programming outside university.

c) The third question was: "Will it be useful if you could write and test Java programs on mobiles?"

1- No 2- Sort of 3- Useful 4- Necessary

Surprisingly, some students majoring in computer studies didn't think that it is necessary to have a compiler on smart phones for practicing coding. So the responses to the question about the importance of mobile compiler were affected by the negative image of smart mobile usage. The results were as follows:

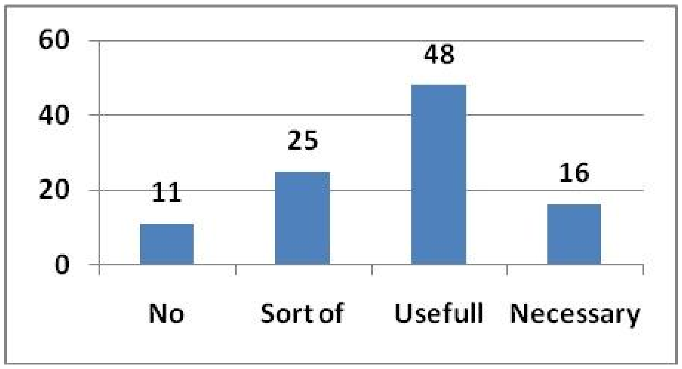


Figure 4-5: Need for a Mobile Java Compiler

The responses showed that students in general don't find having a mobile Java compiler necessary only (16%) thought it is necessary. A cause for might be due to

a) Majority of students have access to computers in general and b) It is more convenient to use computers for programming and c) Another reason could be that students don't spend a long time away from computers that could be utilized by using mobile compilers.

4.2.1.9 Programming Learning Frustration with Learning to Program

Students can only dedicate quality time in learning programming if they are engaged and interested in the activities associated with the subject. The following question was meant to capture the students' image about learning the subject.

How did you find studying the subject?

1- Boring and easy (7%)

2- Boring and hard (24%)

3- Enjoyable and not hard (20%)

4- Enjoyable and challenging (49%)

Many students seemed to approve of the way the programming subject is delivered to them and are not suggesting any changes on the course setup. This was also reflected here as 69% of the students find studying the subject enjoyable. So in the current context, many practices seem to have attracted students while there is still some of them in need of help (24%). On the other hand, there is a group of students (7%) who think that the subject is easy and the time dedicated to it by the university need to be reduced. The last group could be handled by adding challenging bonus questions to be attempted by the group who find the given exercises to be easy and boring.

4.2.1.10 Achieved Level

The phenomenon of courses that has both high and low levels of success and failure rate has been described by Robins (2010). And the cause of this is that the course materials are highly related. Students can achieve high levels in the subject if they can invest time in productive aspects of the subject learning. The following figure displays the marks distribution for the programming subject to the students. A student fails the subject if his score was below 45%.

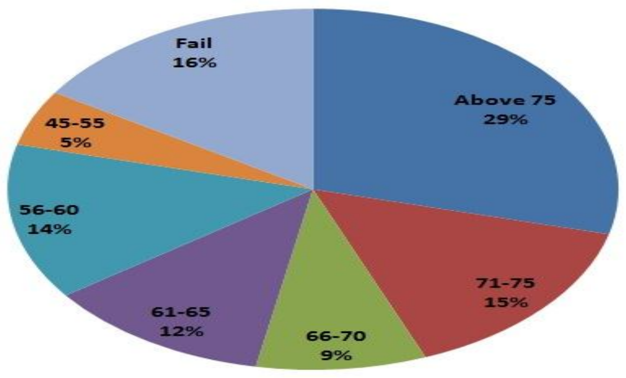


Figure 4-6: Achieved Level in Programming Subject

As depicted in the figure, the subject doesn't seem to be unique or too bad in the mark distribution, then why the fact of programming learning and teaching seem to have worried the teaching staff? As a matter of fact, there is a high number of excellent grades (29%) of the students achieved a mark over 75. This could be justified by the fact that programming is a skill that needs to be learned and mastered to a good extent for it to be useful.

4.2.2 Inspecting Relations between some of the Attributes

Some of the attributes seem to have an effect on other attributes. In the following section some of the interrelated attributes were jointly plotted.

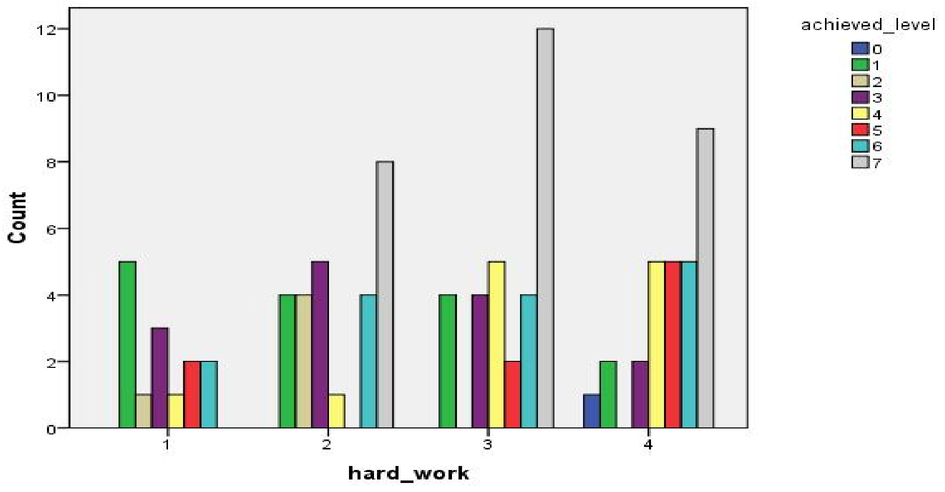


Figure 4-7: Relation between Hard-work and Achieved-level

From the figure, the distribution of higher level of hard work correlates with higher achieved levels.

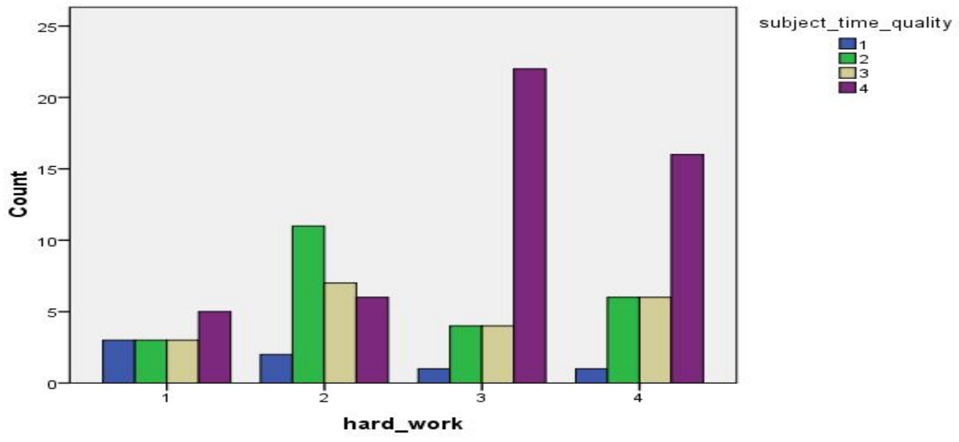


Figure 4-8: Relation between Hard-work level and Subject-time-quality

Students who enjoy the subject seem to be able to work hard in programming subject compared to students with low level of enjoyment.

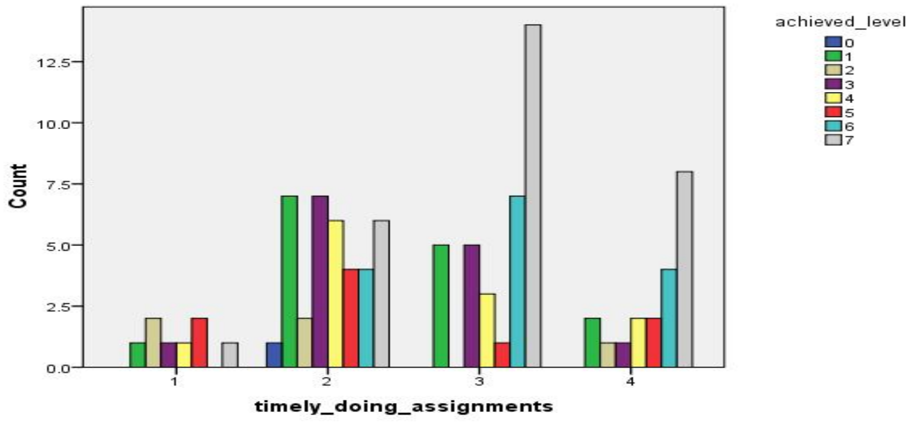


Figure 4-9: Relation between Timely-doing-Assignments and Achieved-Level

From the figure, it could be realized that the students who achieved a high level of students' good marks matches the group of students who reported turning in assignments in a timely manner.

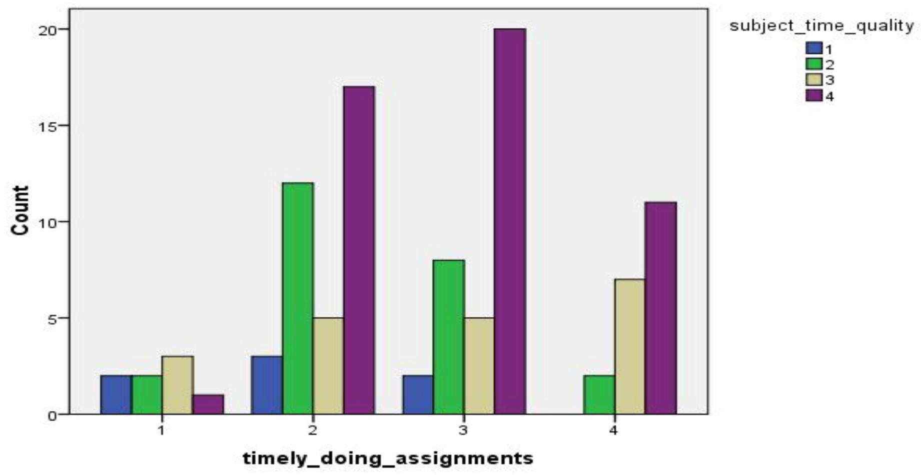


Figure 4-10: Relation between Timely-doing-assignments and Subject-time-quality

From the figure, there are no students who think the subject is boring in the group that always submit assignments on time. While careless behavior seems to be distributed among all enjoyment levels.

4.2.3 Qualitative Questionnaire

In addition the quantitative questionnaire, free form questions were given to the students to a) describe their engagement level and the reasons that make some students disengaged and the suggested modifications on the subject teaching to make it more engaging, b) The students' study sources and their sources for getting help, and c) The advices that they have for future programming learners based on their experience.

4.2.3.1 Suggests Modification on Subject Teaching

Students were asked about how to make the subject more engaging. Their recommendations vary based on their abilities and level. In some points, conflicting opinions were received from students who attended the same course about necessary changes in the course contents.

- It is very essential to have plenty of practice of coding
- It would be good if there is a mean of running programs/examples during the lecture time.

- Students propose arranging competitions in coding and developing algorithms within their own university's students and other universities.
- Students believe that it is good to be able to view different ways of solving exercises. There should be sharing of solutions that can be seen by the students to take advantage of the presence of more than one way to resolve the correct software. Also, there should be a vote or discussions about the best way of solving programs i.e. a winning program.
- Some students indicated that they were attracted at the beginning of the course, because they could understand the concepts and they could also meet deadlines in submitting the practical assignments. As the concepts accumulate, students feel that they need more time to solve assignments and they start to miss deadlines and lose interest to continue learning.
- Some of the students did not give an opinion on making the material more attractive. This could be due to one of two reasons:
    - The course is good the way it is (the students approve the current teaching method)
    - Nothing Can be done to make the learning programming engaging at all

- Add more practical, meaningful assignments or projects.
- Don’t give complicated assignments so that students don’t feel unable and give up and hence copy solutions.
- Easy assignments so that students' confidence in their level builds up.
- Break projects into modules with proper help when needed from tutors.
- Students mentioned in the quantitative questionnaires that they would like to see how a problem can be solved in different ways share and rank solutions or programming assignments.

The following table showed the categories of the responses about the suggested modification on the subject:

Table 4-2: Categories of students’ responses about the suggested modification

The Suggestions	Count
The simplified explanation and the presence of professors who are competent and sympathetic and available	15

The Suggestions	**Count**
Allocate more time for teaching programming	11
Focus on the practical aspect	10
Give several study sources	6
The subject is perfect the way it is no need for changes	6
Do programming contests, games and competitions	5
Support students to have their own devices	5
Assignments should have practical usefulness related to student environment	5
Reduce the time required for the material and the number of assignments	4
Execute examples in the lectures	4
Suitable syllabus	4
Increase the fun and challenge and avoid boredom	3
Study the subject in the lab	3
Give timely feedback for assignments and put extra effort on them	3
Do more tests	2
Tutors should allocate office hours in the lab with the students	2
The use of attractive presentation medium (make the lectures interesting)	2
Use flow charts and algorithms for problems discussions	2
Use simple language to explain preferably Arabic language	2
Provide recorded lectures	2
Reduce theory	1
Update syllabus and cope with market needs	1
Provide enough time in the computer laboratory	1
Give assignments with suitable difficulty, so that students don't feel frustrated and copy	1
Allocate some time to communicate with students online	1
Give extra optional assignments for the student to make up for missing assignments – no frustration	1
Reduce the load of other subjects in the semester so that students can focus on programming	1
Concentration of effort with weak students in the subject and give optional training for those for those who are willing to	1

4.2.3.2 The sources for study and help

In responses to the question about students' studying resources and the sources of tutoring that helped students beside the teaching staff. Their responses showed that they benefited to a great extent from colleagues and seniors. Students look for several resources about the same topics hoping to find the source that gives a simple explanation of the topics.

Several activities can improve and accelerate programming learning, e.g. practical assignments, tutorials, recorded audio and video revisiting materials can be like a second chance for students to understand the materials and make up for the wasted time and missed sessions. Interestingly, students might get bored doing the same activity for a particular concept several times. So providing a range of resources can help students focus and revisit the concept.

As an alternative method of visualizing textual data, are plotted using word clouds. The key words can be seen in this cloud of words in which text size and color darkness correspond to word frequencies. The tool used for plotting the figure is http://tagcrowd.com.

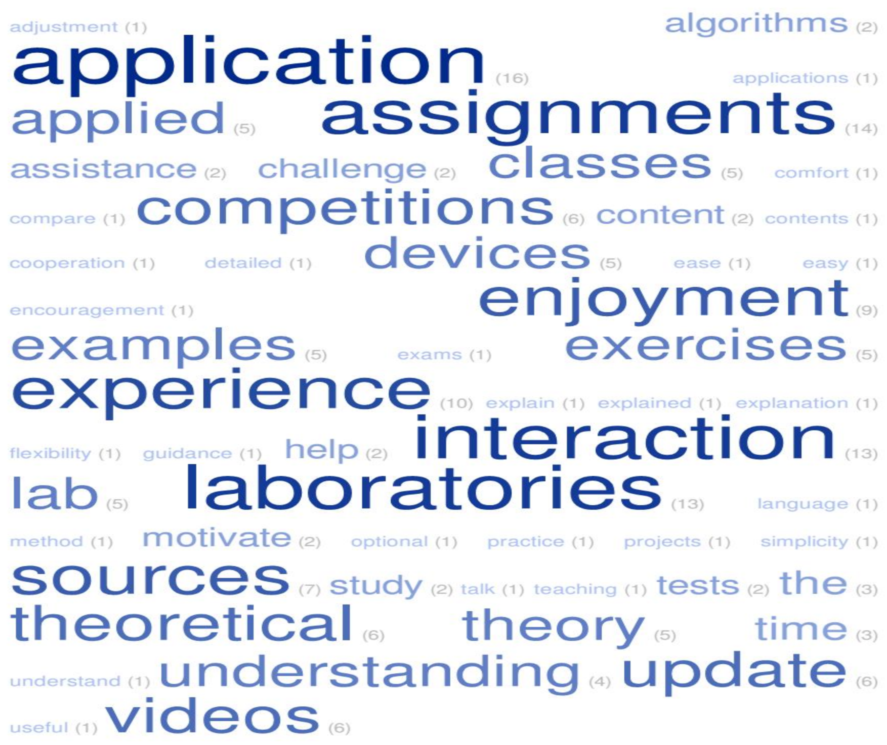


Figure 4-11: Suggested Modification on Subject Teaching

In the previous figure just showed the keywords without the desired action that is required by the students which is decrease and increase. In the following figure the same data is shown again with the desired action + or -.

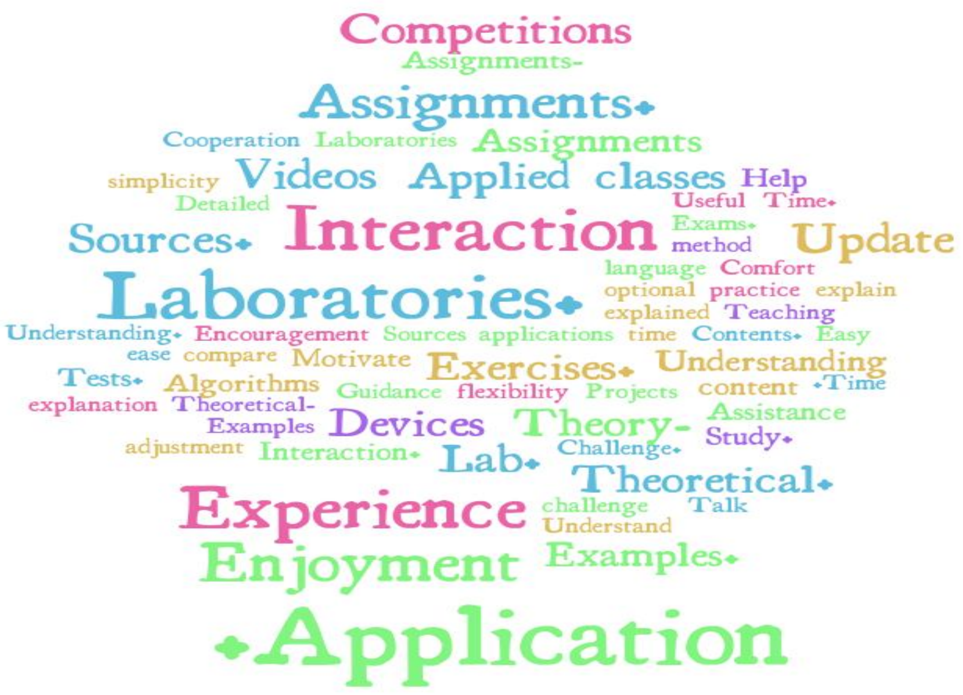


Figure 4-12: Suggested modification on subject teaching with desired action attenutated (+ or -)

It is realized that some of the desired actions were divided into smaller categories, for example assignments is divided into (assignments+, assignments-) and theory was also divided into (theory+, theory-). This indicates that there is no agreement among the students about these actions.

The tool used for plotting the previous figure is http://worditout.com/. In this figure only the size correspond to the frequencies while the color is assigned randomly for distinguishing words.

4.2.3.3 Students' advices for future students

Their advices for a future programming course, students were categorized and listed in the following table:

Table 44-3: Categories of students' Advice for novices

Advices from senior to novice programming students	Count
Do plenty of coding and study lots of examples	28
Timely study and solving exercises	13
Attend lectures and labs and pay better attention on them	13
If this subject is mastered then the following years of study will be easier	5

Do plenty of search and trying to tackle difficulties	5
Depend on self-study	4
Study references	4
Access to diverse sources	4
Don’t fear from the subject difficulty	3
Use the lectures available online	2
Collaborate with colleagues	2
Application of useful projects	2
Have positive qualities such as desire, challenge, ambition, research and diligence	2
Obtain your own device	1
Improve the English language level	1

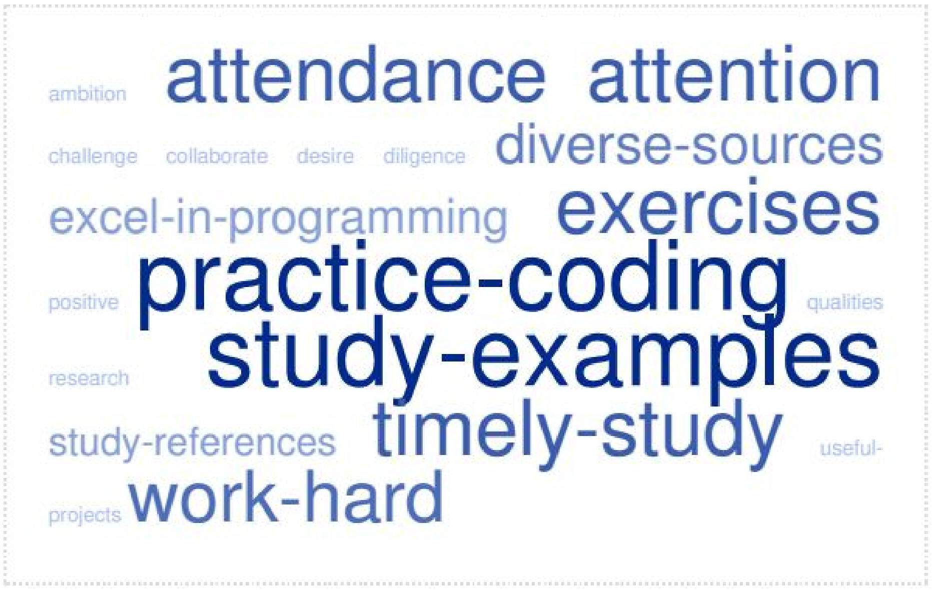


Figure 4-13: Figure of top 50 words using http://tagcrowd.com

One of the questions for the students was about their utilization of the time they had, the majority of the students admitted that they wasted time that they could allocate to studying. While few mentioned that they were saturated and that the effort they allocate to studying programming was satisfactory for them and they couldn’t have done more about learning programming. This reflects that there is a room for enhancing programming learning, but the choices of activities available for students were not engaging.

4.3 Conclusion

The current survey shows many defects or possible areas of improvements in teaching and learning programming, from the questionnaires it is found that:

- Students need flexibility when taking a programming course. Flexibility means to allow multiple learning path options and providing various range of materials and finding an assessment scheme that allow students to make up for poor or missing assignments. It will be good practice if students could get multiple chances to improve their scores which will give them a reason to continue working hard. Frustration can occur when students don’t get timely feedback or help when faced with difficulties and students can’t follow the materials taught timely and they feel left behind. In addition, some students feel frustrated when they are faced with complicated assignments and projects that can’t be divided into a smaller problems.
- The findings emphasize the rule of collaboration among students. Most of the students appreciated the timely feedback, help and hints they are receiving from their colleagues and seniors.
- Students pointed out that it is useful to share coding solutions and the different ways of solving a problem.
- Students mentioned that it is good to have a feeling of competition in solving problems. It is very clear from the responses that this is a subject that needs close follow up of students during learning. Disengagement can also happen to good students who feel that the subject is easy and boring. If timely doing assignments and close follow up is performed by the students this will have a positive effect on the achieved level and subject time quality.

This Chapter described the first artifact created in terms of Design Science Research – the Construct. The following chapter (CHAPTER FIVE: THE MODEL)describes the second artifact which is the model and provides technical details of the suggested solution and the design decisions obtained from usage data that were used in each of the systems iterations.

CHAPTER FIVE: THE MODEL FOR AN ENGAGING ADPATIVE SYSTEM

5.1 Introduction

As mentioned in (CHAPTER ONE: INTRODUCTION), one of the objectives of this research is to influence (increase) and evaluate students' engagement while learning programming. The influencing part will be performed by introducing and manipulating students' attributes on different iterations of the proposed solution's implementation. In chapter two - section one (Literature Review on Students' Engagement) suggested that personalized solution will help in solving most of the disengagement elements in programming learning. And since one to one tutoring is not feasible, section two (Literature Review of Adaptive Systems) showed that from the range of available technologies for tutoring; adaptive systems are easy to implement and it overlay with the research objectives. Personalized contents based on students' attributes will be further investigated in this research. Chapter four (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT) covers the aspect of working with students to identify their views about the current way of learning programming. Also interventions that seems important from their point of view needed to be considered or investigated.

In this work, personalized learning in the form of adaptive system will be planned and presented as a model as in the design science research process steps shown in (CHAPTER THREE: METHODLOGY - DSR Process). The research method used to achieve the objectives is design based science research in which the various steps of designing the system are implemented and reiterated as needed after performing the necessary steps (Offermannetalf 2009). DSR involves working with people to find rigorous IS/ IT solution to real problems.

In order to increase engagement, engagement attributes in learning need to be manipulated; one to one tutoring was the optimal situation. As this is not affordable in today's teaching and learning adaptive systems can also be utilized to influence engagement attributes. After implementing the adaptive solution the affects need to be measured.

Tutoring systems as a method of teaching are classified as behavioristic methods since the system's actions will be decided based on the students' behaviors and responses while using the system (Lowyck, 2014).

This chapter describes the second artifact that was produced in this work, which is (the model) and provides technical details of the suggested solution for implementing an adaptive system that aims to increase students' engagement.

5.2 Adaptive Hypermedia Systems

Adaptivity is the ability of the system to recommend to users educational objects believed to be of their **interests (implicit)**. In an adaptive system; minimal number of components should be displayed to individual users to avoid overloading them (cognitive overload) (Baig, 2014). Developing Adaptive Systems focuses on two major aspects; namely:

a) The structure of the system
b) The **content to be displayed**

In this work the focus is on the content to be displayed to students.

5.2.1 Users Modeling Attributes

A common feature of various adaptive Web systems is the application of **user models** (also known as profiles) to adapt the systems' behavior of individual users. According to (Koch, 1998), the following user's characteristics were identified as a leading attributes in modelling students:

- Knowledge
- Preferences
- Navigation abilities

As a start the proposed model that represents an engaging adaptive learning system will focus on controlling **the contents** to be displayed based on **student's model (user's model)** that is having several attributes with **knowledge** attribute being the core attribute. This model is a variation of the regular adaptive systems because it focuses on enhancing the aspect of students' engagement while using the adaptive system for programming learning. It focuses on students and most of the requirements stem from students' needs, requirements and usage patterns.

5.2.2 User Knowledge Models for Adaptive Hypermedia and Adaptive Educational Systems

The knowledge model in the system or the contents to be displayed could be arranged using a variety of setups. The concept could be represented as nodes and the relationship between the components could be in the form of edges connecting the nodes. Some of the types of knowledge models identified by (Brusilovsky and Millán, 2007) are:

- **Scalar modelling /Stereotype knowledge**: Aims to divide users into scalar groups, e.g. beginners, intermediate, and expert.
- **Overlay modelling**: to overcome limitations of the scalar modelling, structural model is used in which the body of domain knowledge can be divided into certain independent fragments.

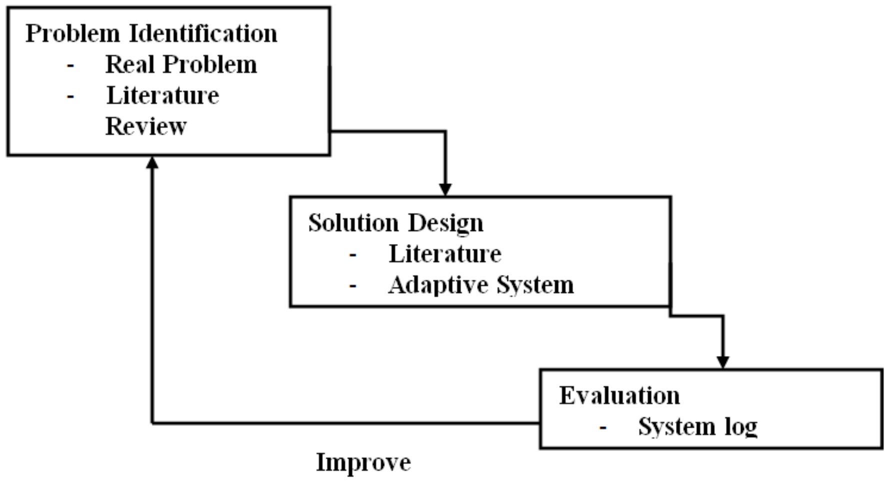
The use of statecharts has been proposed in (*De Oliveira et al, 2001*) for modelling of hypertext and web based applications. By transitioning in a statechart diagram the state of the nodes will change (disable/enable) according to your current location in the diagram. As users move from one node to another the state will toggle at some nodes as an effect of the transition. And thus a new current state **configuration** is obtained.

The steps used in (Martin and Ivan, 2013) aimed to set a standard and to formalize the adaptive system architecture. A reusable adaptive web user interface components were developed. In the user model part the attributes were divided into user profile (page setting preference) and user model (**usage data**). The information in a user model contains: e.g. users' knowledge of the topic, users' preferences, or their past experience.

5.3 The Model

This section explains the process of model development and it elaborates the details of the stages involved in the model.

As mentioned in CHAPTER THREE: METHODLOGY several steps were discussed in creating the suggested solution. The overall design of the system will take the following life cycle.



5-1: System Model

Another way of listing the various steps for developing an engaging adaptive system is shown in the following figure:

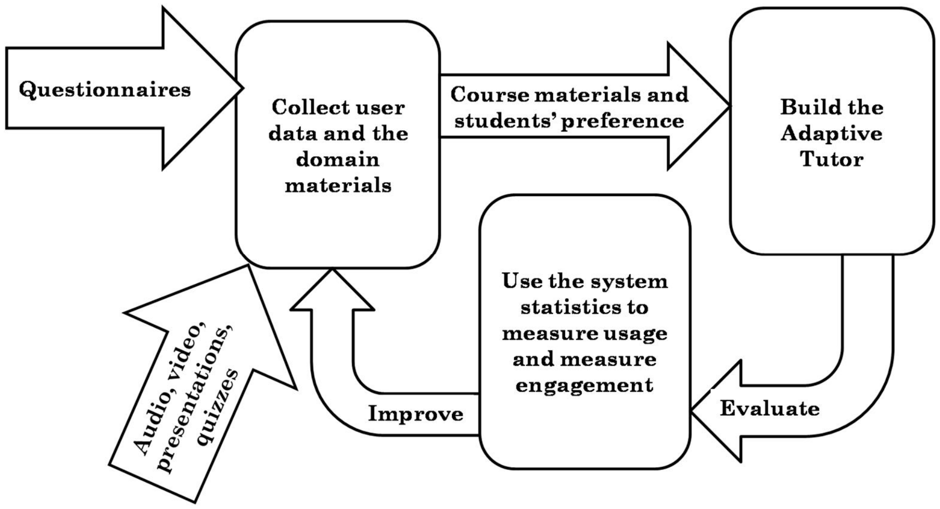

Questionnaires
Collect user data and the domain materials
Course materials and students' preference
Build the Adaptive Tutor
Audio, video, presentations, quizzes
Use the system statistics to measure usage and measure engagement
Improve
Evaluate

Figure 5-2: The Proposed Model

One of the features of the model development is that it allows iteration in building systems, which is a desirable feature in adaptive systems. Adaptivity is built based on user feedbacks from the usage of the systems that will allow evaluation and thusthis will be used for enhancing the design of the following iteration.

5.3.1.1 User data and domain materials (Student's Model)

The first step was to collect the related materials in addition to students' preference. As presented in CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT) a survey was performed to identify students' needs and suggestions regarding programming learning and teaching. These suggestions have been incorporated in designing the Adaptive system.

In this work the following initial attributes were used to describe and direct behavior

1- Students' level
2- Preferred learning language

And the following attributes comprise the desired learning practices or features to be incorporated in the adaptive tutor; that were learned from (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT) are:

1. Time management and reminders.
2. Contents suitable to level/ Adaptive.
3. Timely feedback.
4. Collaboration.
5. Availability of resources.
6. Algorithmic Thinking.
7. Scaffolding.
8. Interest / Motivation.
9. No Permanent effect of Failure.
10. Simplified explanation of materials.
11. Projects that can be divided to simple tasks.
12. Sharing solutions to get experience in solving the same problem in several methods.
13. Students wanted to have a feeling of competitions and ranking.

5.3.1.2 Building the Adaptive Tutor

This section explains the process of model development, including procedures used for literature search, instrument development, data collection, and analysis.

Representing students or users in the system will enable the adaptation and personalization of the system. To build an adaptive system a student model has to be built that is a set of knowledge components (KC) which will be encoded in the adaptive tutor to model students and hence adapt the system accordingly. The set of KCs includes the component skills, concepts, or precepts that a student must acquire to be successful in targeted tasks (Li et al, 2011).

Navigation through the KC will be allowed based on the information encoded in the adaptive tutor in the form of stored attributes (student model/profile). These attributes could be static/fixed such as student ID or dynamic that could be updated while using the system such as level and completed tasks.

Knowledge Components/ Domain

To build an adaptive tutor one of the following two approaches can be used

1- Manually building of student models.
2- Using machine learning techniques.

There are three stages of building intelligent tutoring system:

1- Collect the course materials from students: There will be two types of users in the system (tutors, students). Users can rate, identify mistakes, or miss conceptions. (In this step a hypothesis about students' models will be built-identifying the students' model).
2- Use the materials prepared in step 1 to build the system with emphasize on the features that distinguish a good human tutor and makes it superior to a tutoring system that were presented in section 2.3.4 (Intelligent Tutoring System). The system should be adapted to student level and interaction and hence giving timely feedback. Two components should be decided when implementing an adaptive system that is, the user models (attributes distinguishing users) and the knowledge representation (preferably: state transition diagram).

The knowledge representation in the system or the contents to be displayed could be arranged using a variety of setups. As mentioned in section 5.2.2, one of the ways to represent model is the overlay model in which the knowledge components are represented in a graph (in this work transition diagram). The nodes of this graph represent the available studying materials. The transition diagram should limit the student transition in the system according to the design of the system. In a particular moment the navigation in the system for the student is limited by the state of the student (or level) in that particular time. The following figure 5-2 shows a possible scenario of transition in which a particular student can only access quizzes and exercises that are related to his current visited topic.

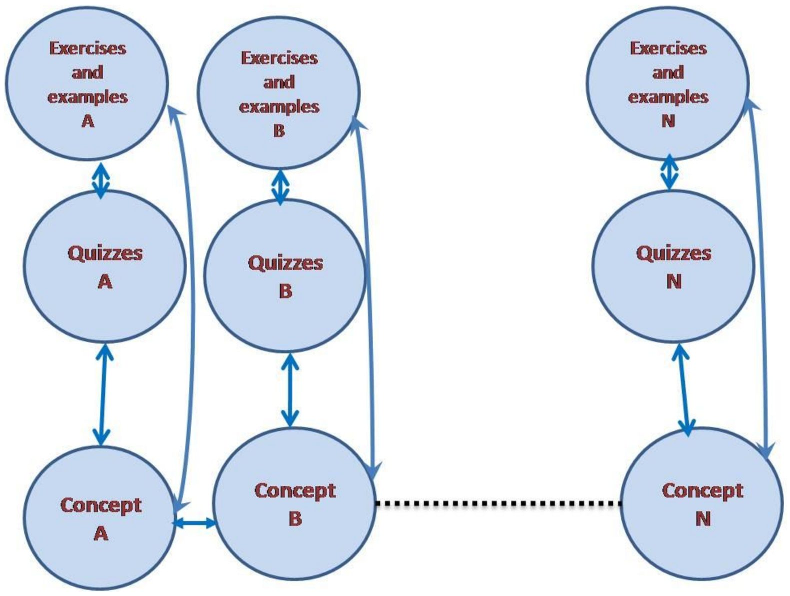


Figure 5-3: Allowed transition in the system

A variation of this graph could be adopted based on the attributes used in representing student's models.

3- Collect usage statistics to evaluate and enhance the system. Many data can be collected from the students' interaction with the system including the most downloaded materials, the time spent to read text, and the topics in which the students get the worst results.

Pedagogical model

For all the iterations of the systems, the students were not forced to use the system by their tutors. Adaptive systems are classified as behavioristic learning methods since the systems' actions will be decided on the students' behaviors and responses to the systems. (Lowyck J , 2014)

Important Features of the Proposed Model

The system will not involve working with sensory data and it basically handled students' activities such as pages visited, duration, and clicks activity. Sensory data could be considered in future extensions of the model.

In addition, the following features should be provided by systems instantiations:

Build an adaptive system that constructs students' models having both types of attributes (static and dynamic). The attributes should stem from both the survey and the literature.

Following are the objectives of the proposed model's instantiation:

- The system should provide diversity of activities to be practiced by the students that are related to programming learning.
- To provide information with portable access (access via different devices).
- To enable most of the traditional educational content management systems' features, such as contact with human tutors and colleagues.
- To provide utilities based on students' specific needs (obtained from the survey).
- To save aspects of students' activities that could be useful in adapting the system and aid in providing a better understanding of engagement in programming learning.

The system that will be implemented by instantiating this model should provide the functionality given in the following diagram:

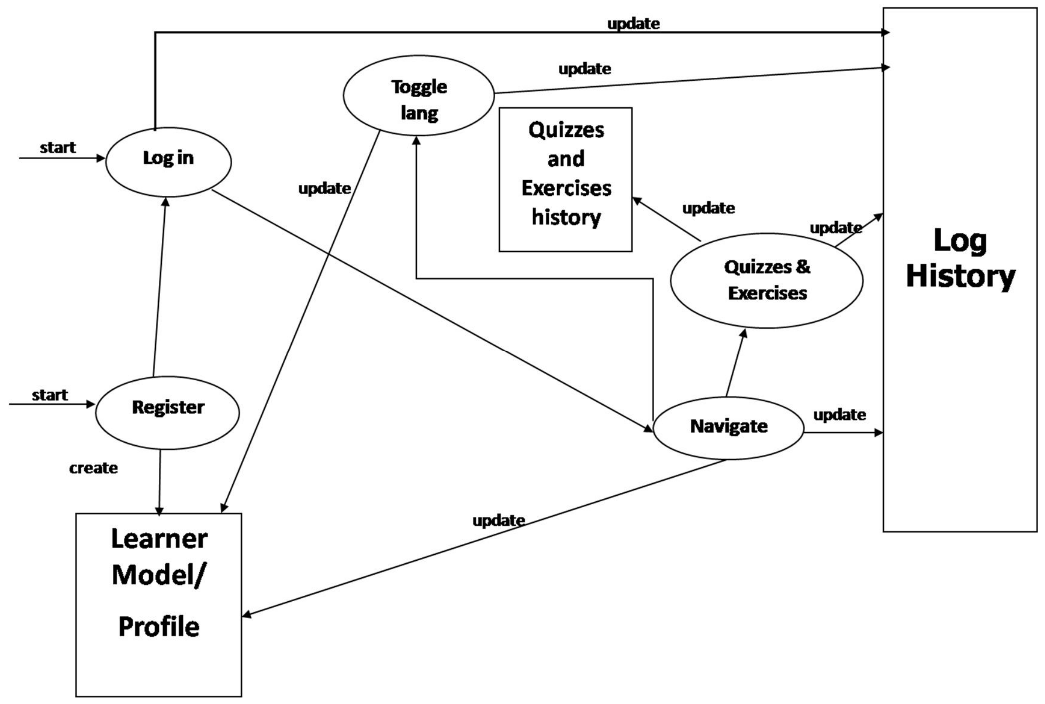


Figure 5-4: The adaptive engaging system functionalities

5.3.1.3 Evaluating engagement

The assumptions made are validated and tested via students' responses to the system from the web logs. To monitor students' engagement while using the adaptive system, analyzing web system logs will be performed. From the literature the following measurements are used to measure users' engagement: Click-through rates, Number of page views, Time spent on site or dwell time, Frequency of return visits (during single or multiple sessions), Time between visits or absence time, Number of tasks, Reading amount.Details of the evaluating the instantiation (DrSUST) will be provided in chapter seven.

Improve the engaging adaptive system

In the first iteration a hypothesis about students' models will be built; identifying the students' models. Every iteration of the system's development will be preceded by an evaluation step based on analyzing the log data of the previous iteration trial. New attributes could be added to every new iteration's implementation in addition to modification of the overall system design.

5.4 Conclusion

Adaptive systems differ in their implementation based on the aspects of the design that need to be emphasized and improved in the system. In this work the emphasis was on increasing students' engagement while learning programming. This was reflected in the design by involving the students early on in the implementation of the solution and studying their current situation and the aspects that they needed to be provided to achieve better engagement.

The model which is presented in this chapter for developing an engaging adaptive system was further implemented in the next chapter (CHAPTER SIX: DR SUST INSTANTIATION) and was tested with real students and the details of students' usage data was provided in the chapter seven (CHAPTER SEVEN: RESULTS and ANALYSIS).

CHAPTER SIX: DR SUST INSTANTIATION

6.1 Introduction

It is difficult for many students to use programming languages to write programs to solve problems. One of the reasons that causes learning problem is the lack of problem solving abilities that many students show. Solving problems is not easy to learn and novices usually don't know how to create algorithms. Training is required in order to help students obtain that skill. In the work by Gomes and Mendes (2007), they proposed building a tool that helps students practice developing and testing algorithms; their tool is named SICAS (Interactive System for Construction of Algorithms and its Simulation).

Programming often demands learners to engage in a significantly high level of individual practice and experimentation in order to master basic skills. Less practice creates cognitive-affective barriers that interact with learners' self-beliefs which will potentially reduce learning. Scott and Ghinea (2013) seek to ascertain how to design a learning environment that can address this issue. To overcome such barriers analytical and adaptable approaches is proposed that include using: soft scaffolding, ongoing, detailed informative feedback and a focus on self-enhancement together with skill development.

Struggling in programming courses can result from low engagement. Better achievement in programming courses can result if the engagement level increased. Giving choices in learning materials and providing continuous support can result in greater engagement.

This chapter describes the fourth artifact of the DSR types of artifact and it is the third that is produced in this work, which is (the **instantiation**), the third artifact as mentioned in section 3.2.1 (Design as an artifact) is t**he instantiation (DrSust).** Implementation of the second artifact (the model) that was described in (CHAPTER FIVE: THE MODEL) in several iterations is demonstrated in this chapter.

In section 2.2, several attributes were summarized from the literature that found to affect engagement. The following figure summarizes some of the attributes that increase engagement in programming learning.

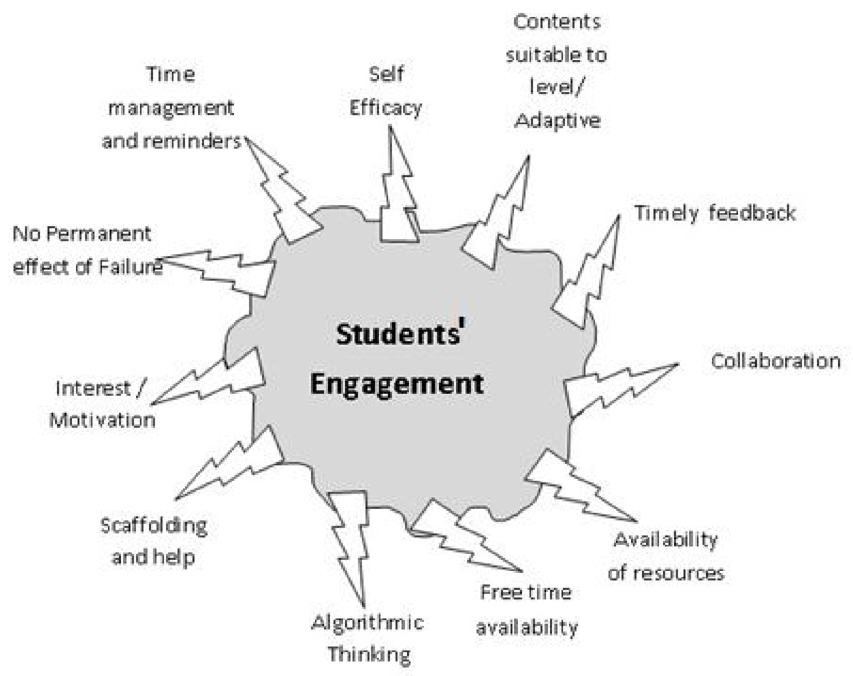


Figure 6-1: Attributes that increases students' engagement in programming learning

6.2 The Supporting Adaptive Website (Dr SUST)

The introductory programming course at SUST uses Java as a language for teaching programming concepts. The approach used for teaching is programming first approach, which means focusing on problem solving and not on teaching object oriented programming. The main reference used is: Introduction to Java Programming by Y. Daniel Liang (http://www.cs.armstrong.edu/liang/). The book has a web site (http://www.cs.armstrong.edu/liang/intro9e/) with prearranged materials and slides and quizzes for the different concepts. The quizzes and part of the exercises in the system were obtained from the book mentioned above (Liang, 2009).

Three units should be decided before using any adaptive system, namely:

- Student or learner model - used to determine the student’s progress.
- Domain model - information that represents the knowledge in an Adaptive system.
- Pedagogical model - handles the actual “tutoring” aspects of the tutoring system, adaptive selection of the materials to be presented to students.

6.2.1 Users' model/ Students' Model

This section will provide attributes that constitute the user model or profile on the different iterations of the DrSUST. Modelling students can have different state attributes. The attributes used can cover the aspects related to students, namely:

- Learner state.
- Performance state.
- Cognitive state.
- Affective state.

The table below lists the derived attributes related to students' engagement and the way they were implemented in the system.

Table 6-1: Users' Attributes

Criterion	Individual Attribute	Generic Design
Contents language	√	
Simplified explanation		√
Quizzes with immediate feedback		√
Anonymous Assistant		√
Controlled navigation vs. free navigation	√	
Unlocking exercises' solutions	√	
Devices used for accessing the system		√
Level, Q_level, Ex_level	√	
Chatting help and support from lecturers and colleagues		√
Learning Styles (holistic/serialist)	√	

From the table, attributes that could describe the users were divided into two categories. Firstly, attributes stored and used as part of the students' model which will eventually be used for adapting the system. And secondly, attributes that were implemented as a general solution in the design.

The user model is the critical component in this research. Initial modelling of the user had one attribute as a start and additional attributes were added in every iteration and development of a new version of the adaptive system.

1. **Navigation Pointer**: the **navigation pointer** which lets the students continue from the last place he visited in the previous visit/session.
2. **Reading Level**: initially both reading level and navigation pointer were considered to be identical attributes. But the fact that students revisit previously read concept created a difference. Reading Level does only increment while the navigation pointer can increment or decrement.
3. **Preferred language**: Initially the domain knowledge materials were in Arabic language only as the interviewed students mentioned some difficulties in studying English. And after interviewing more students it has been realized that not all students preferred to study in Arabic. So the **preferred language** attribute was added as a student attribute that can be altered at any time.
4. **Exercises level**: Students who master a concept and perform the necessary exercises will level up in Ex-level and get more difficult exercises.
5. **Quizzes level**: Students who master a concept and perform the necessary quizzes will level up in Q-level and get more difficult quizzes.
6. **Learning Style**: holist/serialist

Future attributes:

State	**Concept Ex Access**
Students can view 3 solutions of every concept	L_i_Access=3
Additional solution access could be obtained with every trial of solving exercise	L_i_Access++

- **Quizzes-Level**: This attribute is used to generate hints and feedback.

State	**Action**
The Student knows the concept	Skip similar questions
Student forgot	Give student a hint
Guessing	Advice to study the concept

An automated help generation could be considered in future versions. Currently the question is posted to one of the available tutors and to answer it, the tutor could access a summary about the student record to understand the student better and hence respond accordingly.

How and when to produce hints or help students? Usually help is given to students if they seemed to be stuck or ask for help. But there is a risk that students might not try hard enough at times asking for more and more hints i.e. manipulating the system.A student could be told to retry if he didn't try hard enough or help him if he has been struggling for a while. At some point, the system might not be able to help the student so this student will be directed to meet their human tutors.

6.2.2 The Domain:

Three iterations of the system were performed in which the following sections present the details of the development and the changes that were made. The tutoring system DrSUST was built as follows:

6.2.2.1 DrSUST 1:

The contents of the course are divided into 6 units/ concepts. All students start at level 0. The materials for Unit1 can be accessed by any student in addition tounit1 quizzes regardless of their level. A student cannot proceed and study any advanced unit without passing the tests of the previous units or complete reading the previous unit.

Table 6-2: The course contents

Level	open materials based on the Level	No access
L1	Unit1 – Introduction and program structure+ examples+ Quiz1	L3,L4,L5,L6 All other levels' Ex & Q
L2	Unit2 - Primitive data types and operations + examples+ Quiz2	L4,L5,L6 All other levels' Ex & Q
L3	Unit 3 - Selection statements+ examples+ Quiz3	L1,L5,L6 All other levels' Ex & Q
L4	Unit4 – Loops + examples+ Quiz4	L1,L2,L6 All other levels' Ex & Q
L5	Unit5 – Methods + examples+ Quiz5	L1,L2,L3 All other levels' Ex & Q
L6	Unit6 – Arrays + examples+ Quiz6	L1,L2,L3,L4 All other levels' Ex & Q

Students have to go through all the previous units to move to the next unit. The displayed exercises and quizzes accessible to the students are the ones related to the current level. At each particular time a student can advance by one level only or go back to one level only. For example a student in level 2 can also go forward to level 3 or backward to level 1.

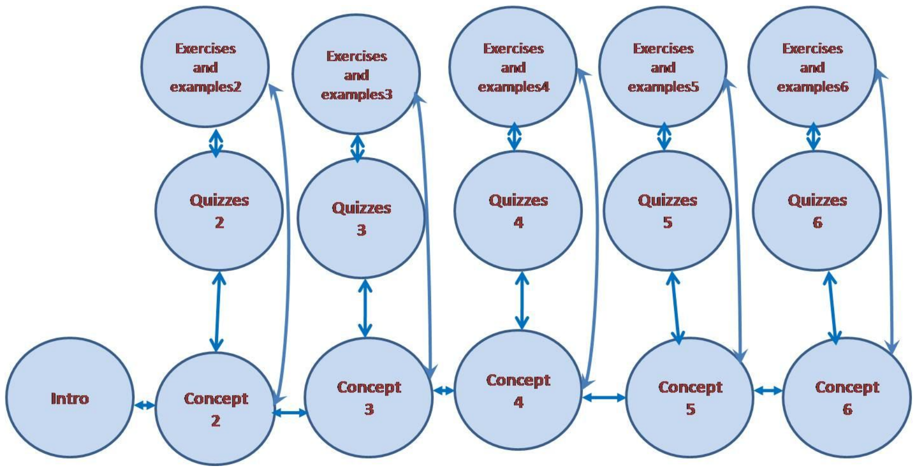


Figure 6-2: Transition Diagram of DrSUST 1.0 based on navigation

System's Requirements

1. Access to adaptive system is done vial links validation and manipulation.
2. The student cannot access the different concepts' sections by simply typing the link, his level should allow that access.

3. Access to exercises and quizzes is granted if a student is having access to related topic or concept.
4. Applying problem solving was the most important recommendation that was highlighted by most of the students. For that this was applied by allowing students to practice programs and submit solutions and get feedback.
5. In addition, students also mentioned that they would benefit from seeing several solutions for the same problem. And thus the system will share correct solutions for the exercises to other students.
6. Shared exercises solutions can only be viewed by a student if he unlocked the particular exercise solutions by submitting a serious attempt to solve that particular exercise.

Intelligent and adaptive systems are classified as behavioristic learning methods since the systems' actions will be decided on the students' behaviors and responses to the systems. (JoostLowyck, 2014)

6.2.2.2 DrSUST2.0 :

The most engaging parts of the system were found to be lecture notes and quizzes. It is normal to stay long in reading a particular note, but it was not logical for students to open quizzes with no attempt to solving them. For the purpose of urging the students to try to solve quizzes the quiz was divided into individual questions that have immediate feedback and also a skip choice.

6.2.2.3 DrSUST3.0 :

Students do revisit the materials they have read after they progress further. According to the domain model in the first version of the system the student could not access the links immediately after passing them with more than one level. For example, if a student is at level 3 and he needed access to level one he will not be able to do that immediately, he has to return to level 2 and from there to level 1. In the third iteration of the system students were free to visit previous material that they need to revisit as shown in the following table 6-2 and figure 6-3. In this version of the system, there was a difference between reading level and the navigation pointer. Reading level can only be incremented while the navigation pointer can be incremented and decremented. Both students' information will be remembered (reading level and the navigation pointer) and used for displaying the appropriate contents.

Table 6-3: The course contents transition for Drsust 3.0

Level	Current materials that corresponds to the Levels	No access
L1	Unit1 – Introduction and program structure+ examples+ Quiz1	L3,L4,L5,L6 All other levels Ex & Q
L2	Unit2 - Primitive data types and operations + examples+ Quiz2	L4,L5,L6 All other levels Ex & Q
L3	Unit 3 - Selection statements+ examples+ Quiz3	L5,L6 All other levels Ex & Q
L4	Unit4 – Loops + examples+ Quiz4	L6 All other levels Ex & Q
L5	Unit5 – Methods + examples+ Quiz5	- All other levels Ex & Q
L6	Unit6 – Arrays + examples+ Quiz6	- All other levels Ex & Q

Students have to go through all the previous units to move to the next unit. The displayed exercises and quizzes accessible to the students are the ones related to the current level.

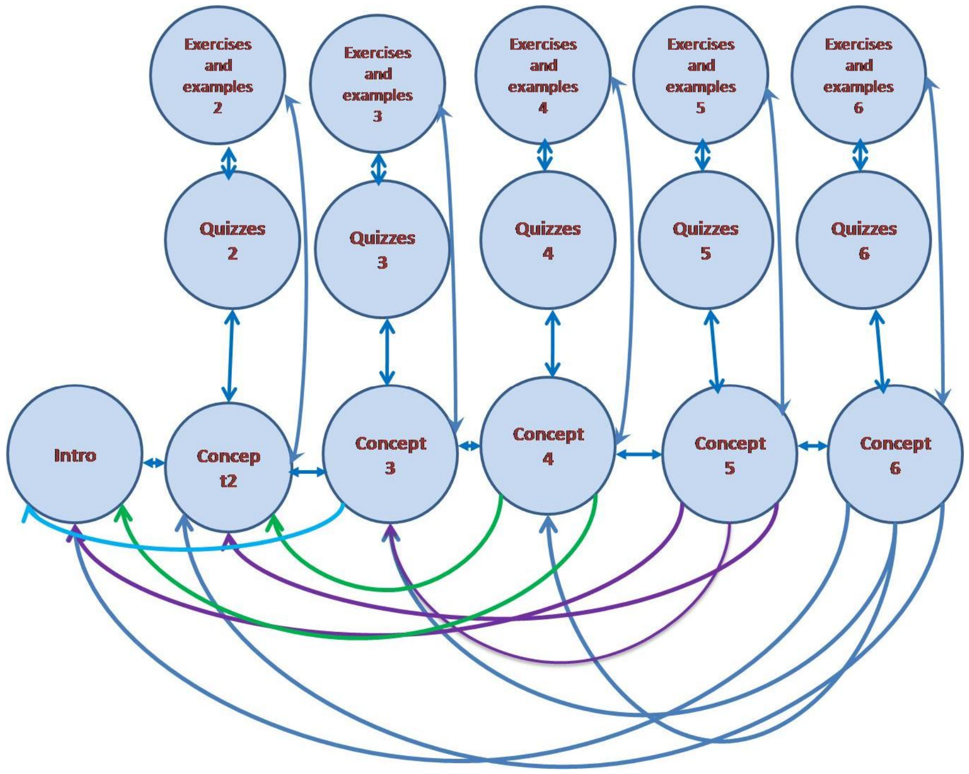


Figure 6-3: Transition Diagram of DrSUST 3.0 based on the student's level

New Requirements

1. Access to adaptive system is done vial links validation and manipulation. The modification in the new version of the system was on revisiting materials. Students can freely navigate backwards if they needed to revisit previously studied materials.
2. Shared exercises solutions can only be viewed by a student if he unlocked the particular exercise solutions by submitting a serious attempt to solve it.

6.3 Conclusion

In this chapter the design of the adaptive system DrSUST is presented. There were three cycles in developing this technical solution and in this chapter listed the distinctive features of every system's iteration.

For all the iterations of DrSUST, the contents of the course are divided into 6 units. All students start at level 0.

In the first and second iterations, the materials for Unit1 could be accessed by any student in addition to all the quizzes regardless of their level. A student cannot proceed and study any advanced unit without passing the tests of the previous units or complete reading the previous unit.

An improvement on the design of the second systems' iterations was to divide the quizzes into a question by question scheme. As an attempt to urge students to try to solve quizzes as individual questions that have immediate feedback with also a skip option.

For the third iteration of the system, students were allowed to navigate easier, especially when revisiting materials they have read after they progress further.

The following chapter will present the evaluation of the students' engagement obtained from each of the systems' iteration and analysis of the findings.

CHAPTER SEVEN: RESULTS AND ANALYSIS

7.1 Introduction

In this chapter the results from users logs on the three system’s iterations of developing and testing DrSUST is presented and analyzed. In addition, results of a preliminary study were also presented and it is called DrSUST 0.0.

User engagement in using a particular system is only applicable to situations in which using the software is optional. Many of the criteria mentioned about users’ engagement will not reflect user interest in using the software if the user has to use the software in any way.

User engagement is considered one of the factors that interest HCI practitioners when evaluating systems and trying to measure the quality of user experience. User engagement can be measured from observing users log data in a system and this particularly can be obtained from the following measures:

1. How many of the students logs into the system?
2. How far did each student go?
    a. (Mastering concepts and – Levelling up).
    b. Number of page views.
3. What is the total duration spent on the site by each student?
4. What is the completion rate among the students?
5. What is the duration between the visits?
6. Most frequently accessed pages.
7. Frequency of return visits (during single or multiple sessions).

7.2 Evaluation

Evaluation was split in three parts:

3. A laboratory experiment,
4. Experiments with the students in the developed adaptive system
5. And presentation of the artifacts with some of the users and experts.

7.3 DrSUST 0.0: Chatting Sessions as a Way of Observing Usefulness of Group Study

Before launching DrSUST, there was enough time to investigate chatting support that would be part of the system. The members of the chatting group were statistics students (67 students) studying an introductory programming course. A scheduled chatting sessions were started as an optional tutorial for the subject.

Risks:

- Some students claim that the pace was fast for them.
- The usage of the group was not specific to the subject, on the first day messages kept coming all the time, even after the agreed session were finished and it kept coming until they were asked to stop. With a gentle reminder it has been agreed to limit the communication with the group to limited times and if desired any other chats should be on the private accounts.

The group started 1stAugust 2015, the first lecture after starting the group was on 6th August, students felt that they can be heard and started to make suggestions. Only 4 complained that they were not in the group, one of them has a smart phone and said he will join the group while the rest didn’t share the reason (most probably they don’t have smart device). It is stated again that the group is optional and asked the ones that don’t have access to use the office hours so that they can ask questions and get support.

First session 01/08/2015 ،9:12 PM - 11:39 PM (24 students)

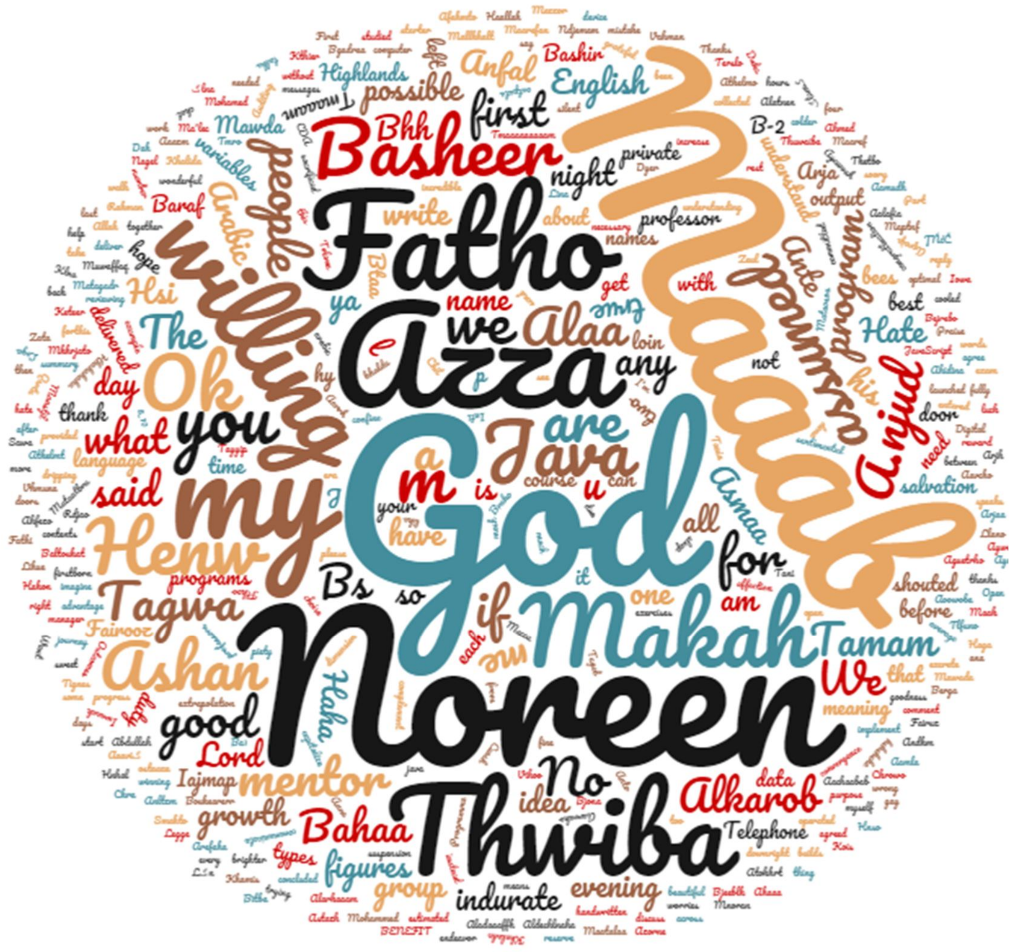


Figure 7-1: Word cloud of the first session's chat

Table 7-1: : Students and their participation frequency on the first session

Student	Comments	Student	Comments	Student	Comments
Noreen	19	Alaa	5	Khalida	1
Maaab	19	Asmaa	3	Mohamed	1
Azza	13	Anfal	3	Fairuz	1
Fatho	12	Fairooz	2	Mecca	1
Thwiba	11	Bashir	2	Ahmed	1
Makah	9	Mawda	2	Arjaa	1
Basheer	7	Thuwaiba	1	Fthoo	1
Tagwa	5	Abdullah	1	Duha	1

Second meeting 02/08/2015 ،7:38 PM - 02/08/2015 ،9:49 PM(27 students)

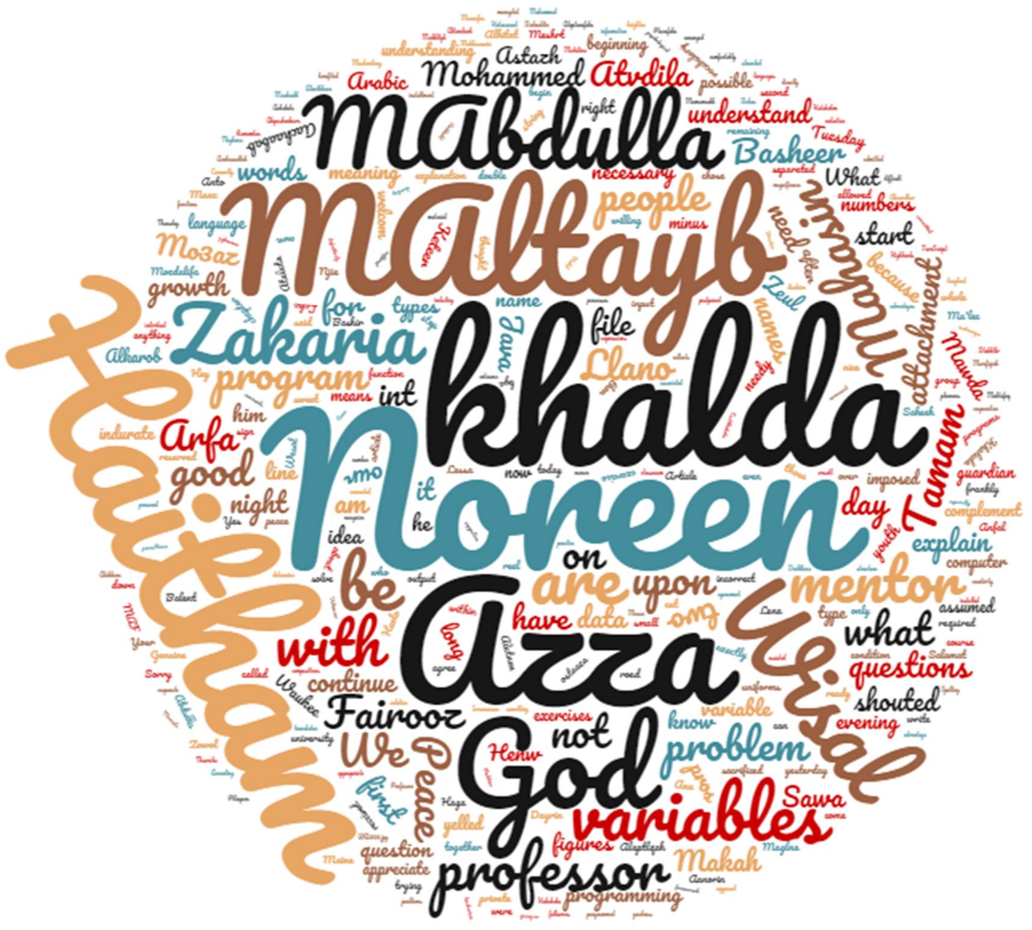


Figure 7-2:Word cloud of the second session’s chat

Table 7-2: Students and their participation frequency on the second session

Student	Comments	Student	Comments	Student	Comments
Noreen	36	Fairooz	8	Khaleda	2
khalda	33	Arfa	7	Bashir	2
Haitham	32	Mohammed	6	Wesal	2
Azza	29	Basheer	6	Anfal	2
MAltayb	25	Mo3az	5	Badruddin	1
MAbdulla	17	Makah	5	Thuwaiba	1
Wisal	17	Mawda	4	Mohammad	1
Zakaria	14	Mozdalifa	2	noreen	1
Mahasin	10	Abdulla	2	Mawada	1

Third meeting 04/08/2015 ،8:12 PM - 04/08/2015 ،11:05 PM (14 students)

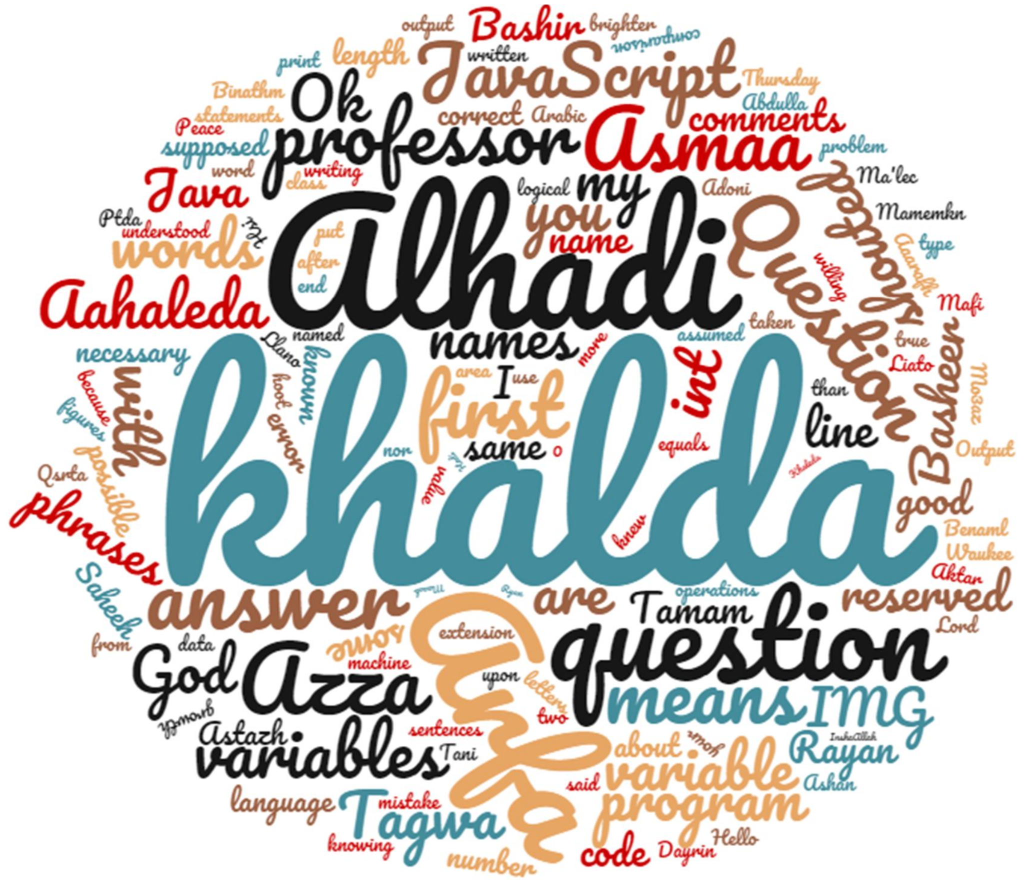


Figure 7-3:Word cloud of the third session’s chat

Table 7-3: Students and their participation frequency on the third session

Student	Comments	Student	Comments
khalda	28	Bashir	4
Alhadi	17	Rayan	4
Arfa	14	Abdulla	2
Azza	9	Mo3az	2
Asmaa	8	Khaleda	1
Tagwa	6	Maaab	1
Basheer	5	Ryan	1

Fourth chatting session 07/08/2015 ،7:57 PM -07/08/2015 ،11:26 PM (27 students)

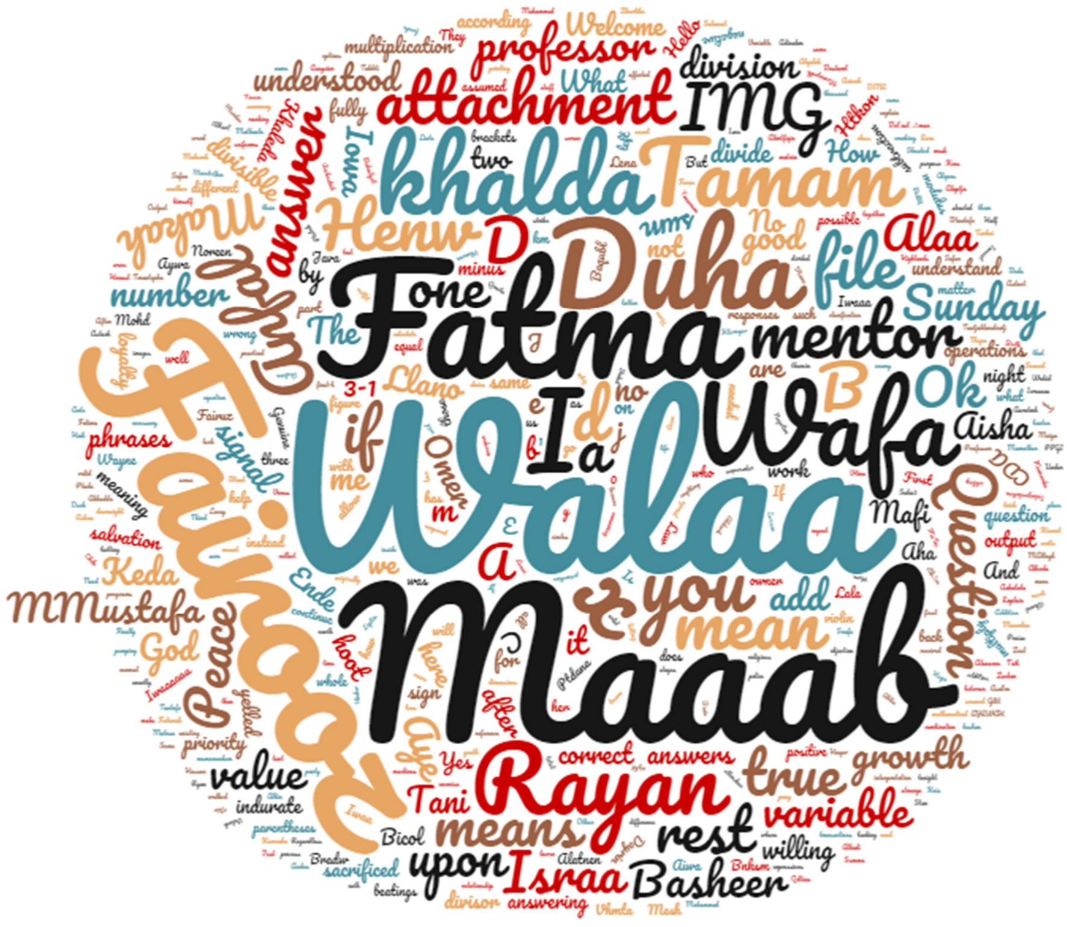


Figure 7-4: Word cloud of the fourth session's chat

Table 7-4: Students and their participation frequency on the fourth visit

Student	Comments	Student	Comments	Student	Comments
Walaa	32	Israa	10	MAltayb	1
Maaab	30	Basheer	7	Alhadi	1
Fairooz	27	Alaa	7	Fatima	2
Fatma	23	MMustafa	6	Feroz	1
Duha	17	Aisha	5	Safaa	1
Wafa	16	Omer	5	Ahmed	1
khalda	15	Fairuz	2	Ryan	1
Rayan	14	Noreen	2	Noon	1
Anfal	10	Mohd	2	Mohammad	1

Fifth session 07/08/2015 ،7:57 PM -07/08/2015 ،11:26 PM (10 students)

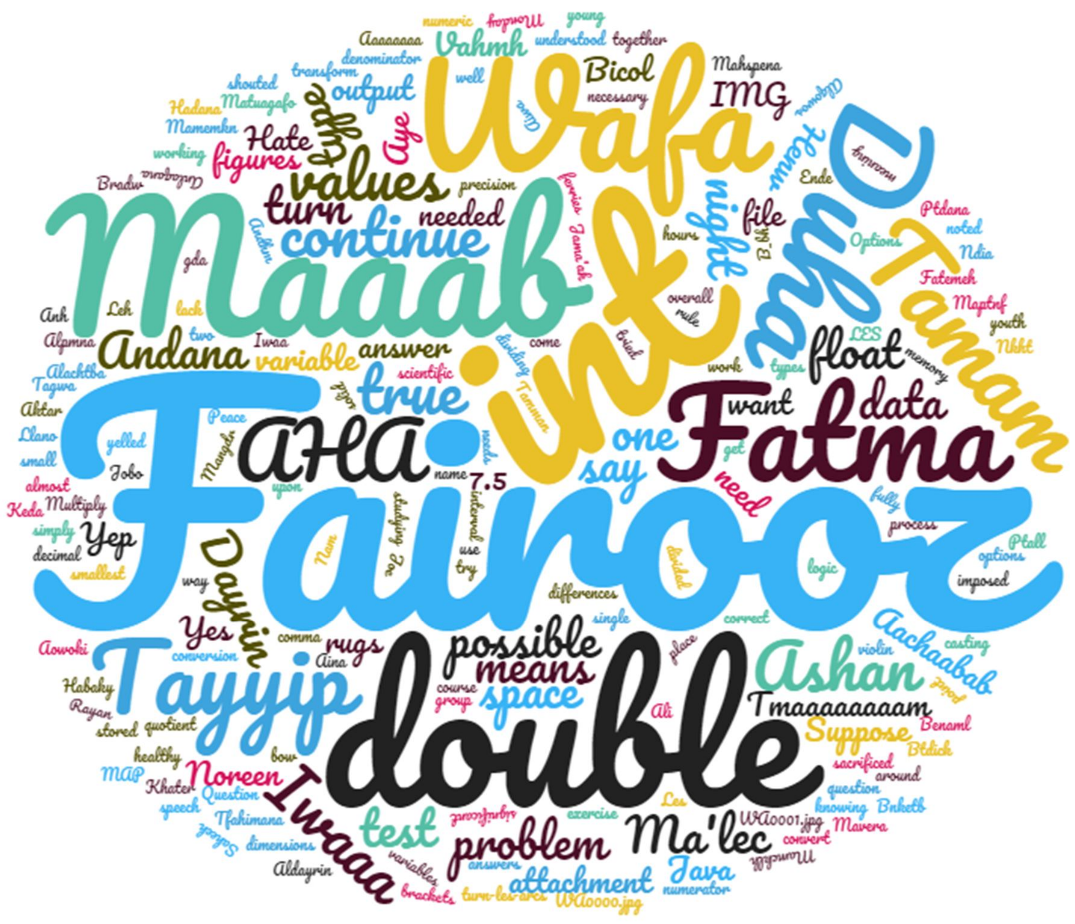


Figure 7-5: Word cloud of the fifth session's chat

Table 7-5: Students and their participation frequency on the fifth visit

Student	Comments	Student	Comments
Fairooz	20	Noreen	2
Maaab	11	Fatemeh	1
Wafa	9	Tagwa	1
Fatma	8	Rayan	1
Duha	8	Ali	1

Sixth session 14/08/2015 ‹8:43 PM - 14/08/2015 ‹10:58 PM (12 students)

Figure 7-6: Word cloud of the sixth session's chat

Table 7-6: Students and their participation frequency on the sixth visit

Student	Comments	Student	Comments
Zakaria	13	Lena	3
Walaa	9	Abdulla	2
Wisal	9	Aisha	2
MAltayb	4	Maaab	2
Duha	4	Zacharias	1
Fatma	3	Muhammad	1

Additional 6 sessions were conducted on 15-18-20-21-23-25 of August and the trend was similar to the sessions presented.

Summary:

The online sessions have a positive impact on the formal classes. From the chatting sessions, it is found that the chatting session is engaging to students in nature and it gives them the feeling that their tutors are understanding and approachable.It doesn’t seem that the students’ level and programming abilities affect the interaction, while there are occasional private messages that show that students approve what is in the group.
Participation rates were good; only 4 out of 67 students were not in the group.

On the first day a student left the group, after addressing him from his colleagues and the lecturer he joined back and the reasons were:

- “He said he doesn’t like groups in general”
- “He was not connected at the time of the session, and when he got connected loads of messages arrived.”

There is a drawback from these scheduled sessions as it is not easy to get information out of it. A more modular way of communicating with teaching staff should be provided in DrSUST. Modular meant that chat messages should be divided/grouped according to its topic.

7.4 DrSUST 1.0

The first iteration of the system was launched for the students who are studying a programming course from Computer Science College (74 students) at SUST on the first semester of the academic year 2015-2016. The system was launched on 8 Feb which is 3 weeks before their final exams.

A fixed logins were created for all the students and it has been advertised to the students that they could use it. The outcome: Only 27% of the students tried the system

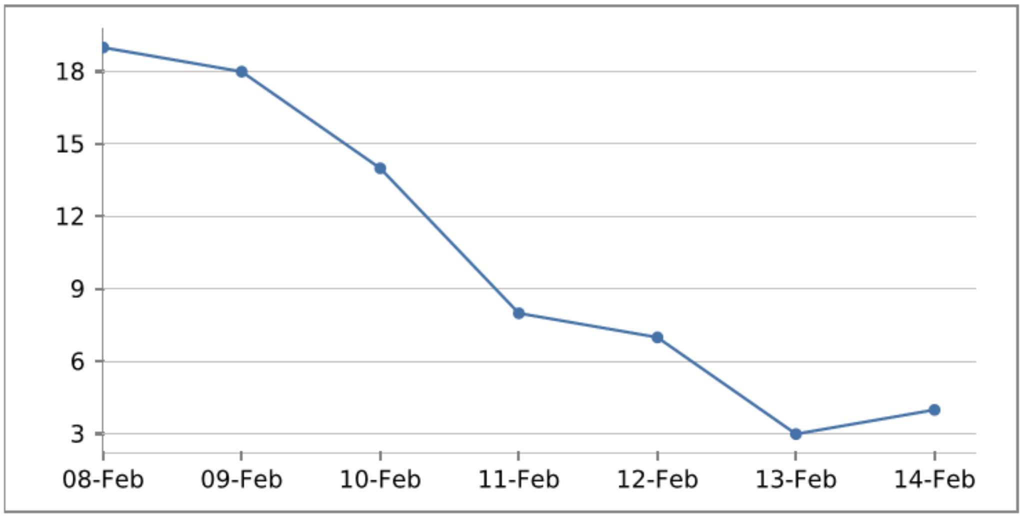


Figure 7-7: Users' visits on the first week of lunching the system

Initial engagement was realized followed by rapid drop on the system usage.

7.4.1 Adaptive Tutor specification and usage data:

The design of this version of the system was as follows:

1- All the teaching materials were in Arabic
2- Limited navigation was allowed based on the navigation pointer.

The usage data showed:

1- Students tend to read and attempt quizzes while there are no attempts of solving programming exercises.
2- Some students stayed on a quiz for long duration with no attempts to solve it in several sessions. The causes of no quiz submission might be because:
   a. In this version of the system students were required to finish a whole quiz before submitting; no immediate feedback on the attempted questions.
   b. Students are writing quiz questions down so that they can solve it with their colleagues.
3- Why these students are not attempting exercises?
   a. Not confident about their programming abilities
   b. No marks awarded for submissions

c. The exercises were easy; students have already done all sorts of practices in the course when they started using the system. While quizzes seemed to have challenged students which made them try to solve it.

4- The system provided material sequencing with a pointer indicating their reached level on reading. Some students tried to access the pages directly by copying links from their colleagues.

An interview was conducted with two of the students who have used the system; and the outcome was that some students prefer to study in English and some prefer Arabic. Also, students are interested in having an immediate feedback and help in the attempted quizzes.

7.5 DrSUST 2.0

This time students were allowed to create their own logins and also senior students were invited to test the system.

7.5.1 Adaptive Tutor specification and usage data:

The design of this version of the system differs from DrSUST 1.0 in the following:

1- Preferred learning language was set to be one of the attributes in student model and hence it will define which part of the system to navigate.
2- The quizzes were divided and submission is done in question base, also students can skip some questions and try them later: positive feedback was received on the questions design.

The outcomes of the usage date showed that: There is still low trial of programming exercises. While the submissions to individual quiz answers were attempted.

More comments were received from users at this stage as follow:

<u>Positive:</u>

“Beautiful summary! – Is that all?” {Student wishing to proceed beyond the given level}.

“I like it!”.

“Good!”.

“Huge effort is obvious in implementing the quizzes sections.”

<u>Neutral:</u>

“Nice work as a start, hope it became more like hackerrank, gj”.

“Continue working on it .... We need it”.

“This is similar to traditional tutoring”.

<u>Negative:</u>

“Some Arabic terms need to be standardized regarding the translation.”

“The design is simple.”

“The Arabic transition arrows are confusing.”

“Limited navigation transition should be a recommendation not forced”

“Some of the quiz questions were not covered in the given topic summary.”

The following table provides a summary to individual students’ visit to DrSUST 2.0 and their dwell time with the system.

Table 7-7: Summary of visits to DrSUST 2.0

St_no	visits	Duration / min	clicks	Topics, quizzes, exercises
112	2 visits {21-8 (8 min)-18-8(3min)}	11	43	{6 , 0, 0}
113	1 visit {12-10 (3 min)}	3	19	{1 , 1, 0}
114 {En + Ar}	1 visit {12-10 (2 min)}	2	6	{2 , 0, 0}
115 {En + Ar}	1 visit {12-10 (86 min)}	86	20	{5 , 0, 0}
116	1 visit {12-10 (6 min)}	6	21	{2 , 1, 0}
117	1 visit {12-10 (7 min)}	7	12	{3 , 0, 0}
118	1 visit {12-10 (1 min)}	1	2	{1 , 0, 0}
119	1 visit {12-10 (1 min)}	1	16	{2 , 2, 0}
120 {En + Ar} All q2	1 visit {14-10 (97 min)}	97	180	{8 , 1, 0}
This students was the student having the most positive comment: “Beautiful summary! – is that all?”				
121 {En + Ar}	1 visit {14-10 (274 min)}	274	32	{7 , 0, 0}
122 All q3+q4	1 visit {14-10 (68 min)}	68	133	{4 , 2, 0}
123	1 visit {15-10 (1 min)}	1	8	{3 , 0, 0}
124	1 visit {15-10 (24 min)}	24	52	{6 , 1, 0}
125 All q4	1 visit 22-1-2017 {11}	11	52	{5 , 1, 0}
Total	**15 visits**	**592 minutes**	**596 clicks**	

The student no 120 gave the most positive comments on the system. Although this student was not the one who stayed in the system longer, but he was the user with the higher number of clicks or transition in the system.

The following table gives the count of individual visits to each of the concepts.

Table 7-8: Students' visits counts

Student ➔	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**	**120**	**121**	**122**	**123**	**124**	**125**
Concept														
'intro.php'	5	3	4	8	8	8	2	2	5	4	4	5	9	2
'datatype.php'	6		3	6		2		2	2	6	4	2	8	4
controlstatements.php'	6			4		2			2	6	2	2	4	2
loops.php'	6			2					5	4	2		4	2
methods.php'	6								4	8			4	2
arrays.php'	8								6	4			2	

The following figure shows the trace of students' visits.

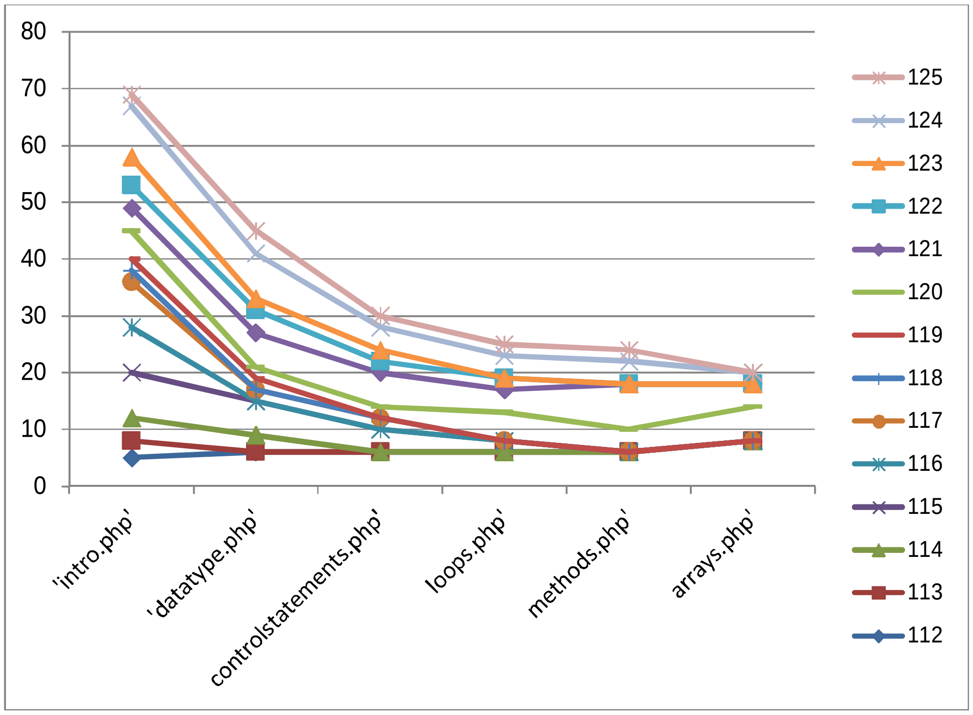


Figure 7-8: Students' visits transition graph

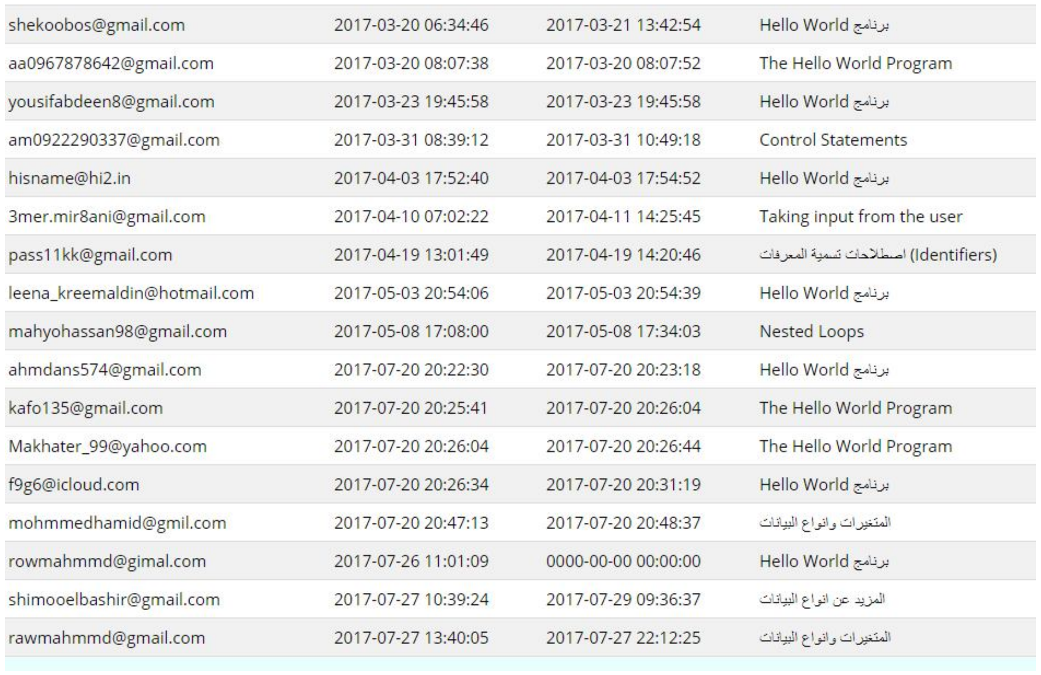
From the log data, some study trend in navigating the system appeared on some of the students' usage of the system:

a) Students finished visiting all the lecture notes before attempting to solve any question or try in programming exercise. After finishing the first round of reading the related materials the student will go back from the beginning to try other activities. Thus restricting the navigation on the web site to one level at a time is not applicable for students who wish to revisit some of the previously studied concepts. While the expected behavior was that students will read a particular concept note and try the corresponding exercises and quizzes before going to advanced levels.
b) Some students visit all the lecture notes regardless of the language. They read the topic in Arabic and later will cover the same topic in English. Which means that student's preferred language should remain a dynamic attribute that changes based on the student's interest.

7.6 DrSUST3.0

Further improvements were as follows: revising of the consistency of English materials and it is Arabic translation was performed. Modification on the ease of navigation was also provided. And comments were allowed to be grouped into public and instructor comments. Public comments made on individual topics could be seen by everyone, while comments to instructor were meant to be seen only by the sender and the receiver.

The current users started the course on May-2017. The students covered 2 concepts in class out of 6 concepts. The following table shows their login details and the current location in the study. It was found that 2 of them have proceeded beyond the course level.



shekoobos@gmail.com	2017-03-20 06:34:46	2017-03-21 13:42:54	برنامج Hello World
aa0967878642@gmail.com	2017-03-20 08:07:38	2017-03-20 08:07:52	The Hello World Program
yousifabdeen8@gmail.com	2017-03-23 19:45:58	2017-03-23 19:45:58	برنامج Hello World
am0922290337@gmail.com	2017-03-31 08:39:12	2017-03-31 10:49:18	Control Statements
hisname@hi2.in	2017-04-03 17:52:40	2017-04-03 17:54:52	برنامج Hello World
3mer.mir8ani@gmail.com	2017-04-10 07:02:22	2017-04-11 14:25:45	Taking input from the user
pass11kk@gmail.com	2017-04-19 13:01:49	2017-04-19 14:20:46	اصطلاحات تسمية المعرفات (Identifiers)
leena_kreemaldin@hotmail.com	2017-05-03 20:54:06	2017-05-03 20:54:39	برنامج Hello World
mahyohassan98@gmail.com	2017-05-08 17:08:00	2017-05-08 17:34:03	Nested Loops
ahmdans574@gmail.com	2017-07-20 20:22:30	2017-07-20 20:23:18	برنامج Hello World
kafo135@gmail.com	2017-07-20 20:25:41	2017-07-20 20:26:04	The Hello World Program
Makhater_99@yahoo.com	2017-07-20 20:26:04	2017-07-20 20:26:44	The Hello World Program
f9g6@icloud.com	2017-07-20 20:26:34	2017-07-20 20:31:19	برنامج Hello World
mohmmedhamid@gmil.com	2017-07-20 20:47:13	2017-07-20 20:48:37	المتغيرات وانواع البيانات
rowmahmmd@gimal.com	2017-07-26 11:01:09	0000-00-00 00:00:00	برنامج Hello World
shimooelbashir@gmail.com	2017-07-27 10:39:24	2017-07-29 09:36:37	المزيد عن انواع البيانات
rawmahmmd@gmail.com	2017-07-27 13:40:05	2017-07-27 22:12:25	المتغيرات وانواع البيانات

Figure 7-9: DrSust 3.0 users.

The following table provides a summary to individual students' visit to DrSUST 3.0 and their dwell time with the system.

Table 7-9: Summary of visits to DrSUST 3.0

St_no	Visits	Duration / minutes	Clicks	Topics, quizzes, exercises
7	2	4	5	{2 , 1, 0}
8	1	1	2	{1,0,0 }
9	1	1	1	{1 , 1, 0}
10	1	10	6	{2 , 0, 0}
11	1	2	11	{1 , 0, 0}
12	2	6	12	{2 , 1, 0}
13	1	79	75	{2 , 1, 0}
14	1	1	2	{1 , 0, 0}
15	1	26	96	{3 , 1, 0}
16	1	1	2	{1,0,0 }
17	1	1	2	{1,0,0 }
18	1	1	2	{1,0,0 }
19	1	6	10	{1,1,0 }
20	1	1	3	{2,0,0 }
22	11	324	403	{6,5,2 }
This student was the only student who submitted exercises.				
23	4	40	25	{1,3,0 }
24	1	45	24	{1,1,0 }
25	1	1	4	{1,1,0 }
Total	**33 visits**	**550 min**	**683 clicks**	

From the above table, more visits were made to this version of the system and more clicks were observed while the overall time on the site was slightly less than the previous version.
This version of the system was launched at the beginning of the 2nd semester of the academic year 2016-2017 and hence the students had time to write code.

7.7 Future Improvements

Intended enhancement for the adaptive system would be in the form of:

1- Learning through testing (Surprising place to learn).
2- Randomized sequence of quiz questions (memorizing effect).
3- Produce and include additional materials in the system for novice students (e.g. Videos).

7.8 Conclusion

The design of the first iteration of DrSUST had the following attributes: (1) All the teaching materials were in Arabic. (2) Limited navigation was allowed based on the navigation pointer.

The design of the second iteration of DrSUST had the following attributes: (1) Preferred learning language was set to be one of the attributes in student model and hence it will define which part of the system to navigate. And (3) The quizzes were divided and submission is done in question base, also students can skip some questions and try them later: positive feedback was received on the questions design.

Further improvements in the third iteration of DrSUST were as follows: revising of the consistency of English materials and it is Arabic translation was performed. Modification on the ease of navigation was also provided.

From the results it can be seen that students have different styles in covering the syllabus. Some tend to scan most of the lecture notes as a first round in studying the syllabus before attempting any quizzes or exercises (holistic) while some students will study and perform additional activities before moving to advanced topics (serialist).

From the results, what is happening in the class affect the students' behavior toward the system! When using the system is totally optional and not a requirement of the course, the time of the academic year in which the system is lunched matters. Near to exams time students prefer to do reading of materials and solving quizzes. Running the system at different times of the course will produce different engagement behavior.

CHAPTER EIGHT: CONCLUSION

8.1 Introduction

Programming is a skill that needs to be developed through intensive practice; students need to allocate time for performing the related learning tasks. Students can allocate time for learning if they are engaged with the subject. This from both literature and experience seems to be missed by many of programming learners. Students' engagement is affected by many factors, including timely feedback and scaffolding.

It is difficult to meet the needs of the increasing number of students with their tutors. The aim of this work is to investigate the current technologies for personalized learning and to implement a solution for increasing engagement for novice programming learners. Also the research aims to find the attributes for the selected technology that could be adjusted to increase students' engagement in programming learning.

- In this research, the research questions were:
  - What are the attributes of students' disengagement in learning programming at SUST?
  - How can adaptive tutoring systems increase students' engagement?
  - How can students' engagement be measured?

8.2 Thesis Contribution

As presented in section 2.2.4 (Educational Practices that Contribute to Student Engagement), in order to improve programming learning several factors should be considered. In addition, the attitude of both students and staff could be modified to result in a higher engagement level. Engagement in learning leads to desirable learning outcomes.

From the literature the following: learning strategy, collaboration, self-efficacy, time availability, interest, support, scaffolding, availability of devices and resources and comfort level were identified as the leading attributes that affect engagement in programming. A good Human tutor helps in engagement since he can give timely feedback that will reduce frustration and help student to understand the topics better.

Also a good tutor will force students to dedicate time and keep moving, forces students to collaborate and find ways that will engage and excite students. Unfortunately, one to one tutoring is not feasible in most of the current higher education settings. This is fairly applicable for a small number of students. And usually in larger classes, lecturers might not be able to identify students' knowledge gaps since many students are passive and there is no way to converse with each of them.

Several IT solutions exist that could help in providing a quality learning experience and improve engagement. Such as: (1) recommender system, (2) adaptive systems and (3) intelligent tutoring systems that can help in increasing engagement in programming learning.

From the range of available technologies for tutoring adaptive tutors are easy to implement and it overlay with the research objectives. Personalized contents based on students' attributes will be further investigated in this research.

Three artifacts were produced in this work to meet the research objectives:

Firstly, the attributes contributing to students' engagement from three sources: literature (Literature Review on Students' Engagement), students' quantitative and qualitative surveys (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT), and from evaluating the usage of the designed solution.Secondly, overall technical details of the suggested solution and the design decisions were presented in chapter five (CHAPTER FIVE: THE MODEL). And the final artifact was presented in chapter six (CHAPTER SIX: DR SUST) that is the design of the adaptive system DrSUST. There were three cycles in developing this technical solution and in this chapter listed the distinctive features of every system's iteration.

The last objective was to evaluate students' engagement in the designed solution. Hence, chapter seven (CHAPTER SEVEN: RESULTS and ANALYSIS) presented the evaluation of the students' engagement obtained from each of the systems' iteration and analysis of the findings. To monitor students' engagement while using the adaptive system, analyzing web system logs was performed. From the literature the following measurement are used to measure users' engagement: Click-through rates, Number of page views, Time spent on site or dwell time, Frequency of return visits (during single or multiple sessions), Time between visits or absence time, Number of tasks, Reading amount.

In this work, personalized learning in the form of adaptive system was implemented following the design science research process steps shown in Design Science Research (DSR). Since the research involves a real problem, current situation needs to be studied in addition to covering the literature. Chapter 4 covered the aspect of working with people to identify their view of the current situation. Chapter 4 presented finding of students quantitative and qualitative surveys (CHAPTER FOUR: CASE STUDY - PROGRAMMING LEARNING NEEDED SUPPORT) from which the finding was incorporated into the designed solution.

From the questionnaires it is found that: Students need flexibility when taking a programming course. Flexibility means to allow multiple learning path options and providing various range of materials and finding an assessment scheme that allow students to make up for poor or missing assignments. It will be good practice if students could get multiple chances to improve their scores which will give them a reason to continue working hard. Frustration can occur when students don’t get timely feedback or help when faced with difficulties and students can’t follow the materials taught timely and they feel left behind. In addition, some students feel frustrated when they are faced with complicated assignments and projects that can’t be divided into smaller problems. Also, students pointed out that it is useful to share coding solutions and the different ways of solving a problem. Students mentioned that it is good to have a feeling of competition in solving problems. Disengagement can also happen to good students who feel that the subject is easy and boring.

CHAPTER FIVE: THE MODEL, provided the technical details of the suggested technical solution and the design decisions obtained from usage data that were used in each of the systems iterations.

Adaptive systems differ in their implementation based on the aspects of the design that need to be emphasized and improved in the system. In this work the emphasis was on increasing students’ engagement while learning programming. This was reflected in the design by involving the students early on in the implementation of the solution and studying their current situation and the aspects that they needed to be provided to achieve better engagement.

There were three iterations in developing this technical solution. Attributes that represents students were added incrementally for each of the system's iterations.

The design of the first iteration of DrSUST had the following attributes: (1) All the teaching materials were in Arabic. (2) Limited navigation was allowed based on the navigation pointer.

The design of the second iteration of DrSUST had the following attributes: (1) preferred learning language was set to be one of the attributes in student model and hence it will define which part of the system to navigate. And (2) the quizzes were divided and submission is done in question base, also students can skip some questions and try them later: positive feedback was received on the questions design.

Further improvements in the third iteration of DrSUST were as follows: revising of the consistency of English materials and it is Arabic translation was performed. Modification on the ease of navigation was also provided.

From the results it can be seen that students have different styles in covering the syllabus. Some tend to scan most of the lecture notes as a first round in studying the syllabus before attempting any quizzes or exercises (holist) while some students will study and perform additional activities before moving to advanced topics (serialist).

8.3 Future work

The developed system aimed to make students practice programming without frustration or fear of marking. Scaffolding in programming was performed manually in this version of the system. Automating the scaffolding by providing hints/ help and generating hints can be performed using machine learning techniques. And thus this system can be used to serve a larger set of users.

Currently, accessing the system is not part of the programming course. That is to say, doing the assignments and getting the feedback is optional. A suitable method of integrating this site as part of the Java programming course could lead to more practice and commitment on using it and benefiting from the services.

8.4 Conclusion

Adaptive systems differ in their implementation based on the aspects of the design that need to be emphasized and improved. In this work the emphasis was on increasing students' engagement while learning programming. This was reflected in the design by involving the students early on in the implementation of the solution, studying their current situation and the aspects that they needed to be provided to achieve better engagement. There were three iterations in developing this technical solution. Attributes that represents students were added incrementally for each of the system's iterations. Requirements were perceived from students' needs and personalized systems survey and attributes that can increase engagement were identified. In building the system students' engagement attributes were divided to generic attributes which were used as guidelines when implementing the system and attributes that are part of the students' model i.e. personalized attributes. **Generic attributes** are attributes that affect the general design of the system i.e.: Simplified explanation, Quizzes with immediate feedback, Anonymous Assistant, Devices used for accessing the system and Chatting help and support from lecturers and colleagues. The **personalized attributes** are attributes that represents features of individual students and they were: Contents language, Controlled navigation vs. free navigation, Unlocking exercises' solutions, Level, Q_level, Ex_level, and Learning Styles (holist/serialist). The attributes that reflect students' engagement were: making comments, click rates, overall coverage of materials and the duration in using the system.

The online adaptive system DrSUST, that help students adaptively to learn the introductory programming course is built and made accessible to students via several versions. Requirements were perceived from students' needs and personalized systems survey and several factors that can increase engagement were identified. Engagement of students with the system was assessed and measure using the metrics discussed in chapter seven. Log data concerning students' behavior is made available for further study (navigation data set).

The objectives were met as presented by the three artifacts: Firstly: the construct language (Engagement attributes) from literature, from the results of the survey, and from some of the attributes defining students' engagement at SUST in learning programming were identified by analyzing log data/ in the developed online adaptive system that was available to students. Secondly: the model- shows general guidelines for implementing and engaging adaptive system for learning and the model. And finally: the instantiation (DrSust) - implementation of the model was performed in several iterations.

REFERENCES

Alsaggaf, W. A. (2013). A constructivist, mobile and principled approach to the learning and teaching of programming (Doctoral dissertation, RMIT University).

Anderson, J.R.; Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). "Cognitive tutors: Lessons learned". The Journal of the Learning Sciences 4: 167–207.

Australasian survey of student engagement (AUSSE), 2009, http://www.tedi.uq.edu.au/australasian-survey-of-student-engagement, Accessed 1 August 2014

Baccarani, C. (2014, December). Is the professor still useful in the age of the Internet?.In Liverpool (2014): 17th Toulon-Verona Conference" Excellence in Services".

Baig, F., QUALITY ASSESSMENT OF COFALE: AN AUTHORING TOOL FOR ADAPTIVE EDUCATIONAL HYPERMEDIA SYSTEMS., Journal of Quality and Technology Management Volume X, Issue II, December 2014, Page 67 – 80

Barata, Gabriel, Sandra Gama, Joaquim Jorge, and Daniel Gonçalves."Engaging engeneering students with gamification." In Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on, pp. 1-8. IEEE, 2013.

Bennedsen, J., Caspersen, M. E., andKölling, M. (Eds.). (2008). Reflections on the teaching of programming: methods and implementations (Vol. 4821). Springer.

Bo Liu, INTELLIGENT TUTORING SYSTEM AND ITS APPLICATION

Brown, L. and Howard, A.M., 2014. Assessment of Engagement for Intelligent Educational Agents: A Pilot Study with Middle School Students.

Brusilovsky, P. and Millán, E., 2007, January. User models for adaptive hypermedia and adaptive educational systems. In The adaptive web (pp. 3-53).Springer-Verlag.

De Oliveira, M.C.F., Turine, M.A.S. and Masiero, P.C., 2001. A statechart-based model for hypermedia applications. ACM Transactions on Information Systems (TOIS), 19(1), pp.28-52.

Dixson, M.D., 2012. Creating effective student engagement in online courses: What do students find engaging?. Journal of the Scholarship of Teaching and Learning, 10(2), pp.1-13.

Dobashi, K. Approaches to the approximate tabulation of the duration of student visits when working with digitized course materials within the class. The International Journal of E-Learning and Educational Technologies in the Digital Media. 2015; 1(4): 197-202.

Drigas, Athanasios S., KaterinaArgyri, and John Vrettaros. "Decade review (1999-2009): Artificial intelligence techniques in student modeling." Best Practices for the Knowledge Society.Knowledge, Learning, Development and Technology for All.Springer Berlin Heidelberg, 2009.552-564.

Dupret, G. and Lalmas, M., 2013, February. Absence time and user engagement: evaluating ranking functions. In Proceedings of the sixth ACM international conference on Web search and data mining (pp. 173-182).ACM.

Goldingay, S. and Land, C., 2014.Emotion: The" E" in Engagement in Online Distance Education in Social Work. Journal of Open, Flexible and Distance Learning, 18(1), pp.58-72.

Goldberg, B., and Hoffman, M. (2015). Adaptive Course Flow and Sequencing through the Engine for Management of Adaptive Pedagogy (EMAP). In Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice (p. 61).

Gomes, A., and Mendes, A. J. (2007, June).An environment to improve programming education.In Proceedings of the 2007 international conference on Computer systems and technologies (p. 88).ACM.

Guenther, C. L., and Miller, R. L. (2011).Factors that promote student engagement. Promoting student engagement, 1, 10-17.

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course.Computers and Education, 54(4), 1127-1136.

Hevner, A.R., 2007. A three cycle view of design science research. Scandinavian journal of information systems, 19(2), p.4.

Hosseini, R., Vihavainen, A., and Brusilovsky, P. (2014).Exploring problem solving paths in a Java programming course.In Psychology of Programming Interest Group Annual Conference 2014 (p. 65).

Jenkins, T. (2002, August).On the difficulty of learning to program.InProceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (Vol. 4, pp. 53-58

Kapp, K. M. (2012). The gamification of learning and instruction: game-based methods and strategies for training and education. John Wiley and Sons

Kinzie, J. (2010). Student engagement and learning: Experiences that matter. In Taking Stock: Research on Teaching and Learning in Higher Education (139-153).

Knutov, E., De Bra, P. and Pechenizkiy, M., 2011. Generic adaptation framework: a process-oriented perspective. Journal of Digital Information,12(1).

Koch, N., 1998, October.Towards a methodology for adaptive hypermedia systems development.In In Adaptivität und Benutzermodellierung in interaktiven Software systemen, Proceedings of the 6th Workshop.

Koedinger, K., and Tanner, M. (7). Things You Should Know About Intelligent Tutoring Systems,(July 2013).

Konecki, M., and Petrlić, M. (2014, January).Main problems of programming novices and the right course of action.In Proceedings of the 25th Central European Conference on Information and Intelligent Systems (pp. 116-123).

Kristen Bourgault ,"Gamification in Education: Epic Win, or Epic Fail?" , - Online Teaching Strategies, 2012, Accessed July 7 2014. http://www.digitalpedagog.org/?p=1416

Kuechler, B. and Vaishnavi, V., 2008. On theory development in design science research: anatomy of a research project. European Journal of Information Systems, 17(5), pp.489-504.

Labaj, M. and Bieliková, M., 2014.Utilization of Exercise Difficulty Rating by Students for Recommendation. In New Horizons in Web Based Learning (pp. 13-22). Springer International Publishing.

Lalmas, M., O'Brien, H. and Yom-Tov, E., 2014.Measuring user engagement. Synthesis Lectures on Information Concepts, Retrieval, and Services, 6(4), pp.1-132.

Liang, Y.D., 2009. Introduction to Java programming: brief version. pearson prentice hall.

Li, N., Cohen, W., Koedinger, K. R., and Matsuda, N. (2010, June).A machine learning approach for automatic student model discovery.In Educational Data Mining 2011.

Lowyck, J., 2014. Bridging learning theories and technology-enhanced environments: A critical appraisal of its history.In Handbook of research on educational communications and technology (pp. 3-20).Springer New York.

Mahajan, R. and Mahajan, V., 2014. Real Time Analysis of Attributes Of An Indian E-Learning Site. The International Journal of E-Learning and Educational Technologies in the Digital Media, 1(2), pp.109-114.

Martin, B. and Ivan, J., 2013. Adaptive System Framework. ADAPTIVE 2013, p.20.

Mezhoudi, N., Perez Medina, J.L. and Vanderdonckt, J., 2015. Agile method in the support of UI Context-Aware Adaptation.In 2nd World Congress on Computer Applications and Information Systems WCCAIS'2015.

MIT, MITOPENCOURSEWARE, viewed 17 October 2013, http://ocw.mit.edu

Muntean, C. I. (2011, October). Raising engagement in e-learning through gamification. In Proc. 6th International Conference on Virtual Learning ICVL(pp. 323-329).

Nicholson, S. (2012).A user-centered theoretical framework for meaningful gamification. Games+ Learning+ Society, 8(1).

Nikula, U., Gotel, O., and Kasurinen, J. (2011). A motivation guided holistic rehabilitation of the first programming course. ACM Transactions on Computing Education (TOCE), 11(4), 24.

Norman, D.A, Draper, S. W. (1986). User-Centered System Design: New Perspectives on Human Computer Interaction.

Offermann, P., Levina, O., Schönherr, M. and Bub, U., 2009, May. Outline of a design science research process. In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (p. 7).ACM.

Oxman, S., Wong, W., and Innovations, D. V. X. (2014). White Paper: Adaptive Learning Systems. Snapwiz.Integrated Education Solutions, 1.

Özmen, B., and Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. Turkish Online Journal of Qualitative Inquiry, 5(3).

Perkins, D. N., Hancock, C., Hobbs, R., Martin, F., and Simmons, R. (1986). "Conditions of learning in novice programmers". Journal of Educational Computing Research, 2(1), 37-55.

Retalis, S. and Papasalouros, A., 2005. Designing and generating educational adaptive hypermedia applications. Educational Technology & Society, 8(3), pp.26-35.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. Computer Science Education, 20(1), 37-71.

Robins, A., Rountree, J., and Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer Science Education, 13(2), 137-172.

Rodríguez, V. and Ayala, G., 2012.Adaptivity and Adaptability of Learning Object’s Interface. International Journal of Computer Applications, 37(1), pp.6-13.

Rogerson, C., and Scott, E. (2010). The fear factor: How it affects students learning to program in a tertiary environment. Journal of Information Technology Education: Research, 9(1), 147-171.

Sakthivel, E., Kannan, K. S., and Arumugam, S. (2013). Optimized Evaluation of Students Performances Using Fuzzy Logic.International Journal of Scientific & Engineering Research, Volume 4, Issue 9.

Scott, M. J., andGhinea, G. (2013, April).Educating programmers: A reflection on barriers to deliberate practice.In Proc. 2nd HEA Conf. on Learning and Teaching in STEM Disciplines (p. 028P).

Sottilare, R., Graesser, A., Hu, X., and Holden, H. (2013). Design recommendations for intelligent tutoring systems.

Thomsen, B. (2008). Using On-Line Tutorials in Introductory IT Courses.InReflections on the Teaching of Programming (pp. 68-74).Springer Berlin Heidelberg.

Trowler, V. (2010).Student engagement literature review. York: Higher Education Academy.

Vaishnavi, V. and Kuechler, W. (2004). “Design Science Research in Information Systems,” January 20, 2004; last updated: October 23, 2013. URL:http://www.desrist.org/design-research-in-information-systems/

VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist. 46(4), 197-221 (2011)

Villanueva, S. (2011). "Lessons learned on promoting student engagement in a virtual versus traditional classroom setting". In R. L. Miller, E. Amsel, B. Beins, K. Kowalewski, and B. Peden (Eds.), Promoting student engagement, Volume 1: Programs, techniques, and opportunities. – (Villanueva, S , 2011)

Von Alan, R. Hevner., March, S.T., Park, J. and Ram, S., 2004. Design science in information systems research. MIS quarterly, 28(1), pp.75-105.

ZatarainCabada, R., Barron Estrada, M.L., Gonzalez Hernandez, F. and OramasBustillos, R., 2015, July.An Affective Learning Environment for Java.In Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on (pp. 350-354).IEEE.

Appendix A: Students Needs in programming learning Survey

Questionnaire about students' activities and the best resources for studying programming fundamentals

استبيان حول نشاطات الطلاب وافضل المصادر لدراسة مادة اساسيات البرمجه

This questionnaire is part of the activities to collect data that will be used to design a system to help the students of programming fundamentals in their study. The questionnaire aims to identify difficult aspects of the subject and to figure out what could be done to serve students and obtain better communication with them. The questionnaire is also designed to obtain information from students who studied the subject before about the resources that helped them to study and prepare for the subject; this will help us give recommendations and choices for future students. It is expected that the system will be available for the students studying Programming fundamental in July semester of the academic year 2014-2015.

هذا الاستبيان هو جزء من مرحلة تجميع بيانات لتصميم نظام لمساعدة طلبة اساسيات البرمجه في الدراسه. يهدف الاستبيان الى التعرف على اوجه الصعوبه في الماده والمناطق التي يمكن ان يقوم النظام فيها بخدمة الطلبه والاساتذه للوصول الى تواصل اكبر. كما يرجى ايضا من خلال الاستبيان الوصول الى اقتراحات من الطلاب حول افضل المصادر التي ساعدتهم على التحضير والدراسه وتنويع المصادر المتاحه في الكليه بالنسبه للدراسين المستقبليين. من المتوقع ان يكون النظام متوفرا بالنسبه لطلاب الفصل الدراسي الثاني للعام الدراسي 2014 – 2015.

This questionnaire will give initial vision for the system developers, although more collaboration is needed from our students who wish to help future students of the course through suggestions and sharing of the resources. Students who are willing to participate in further questionnaire can communicate with us further on the email nejood.test@gmail.com. A web site will be announced later for discussions and sharing. The information in the questionnaire will only be used for developing the future system and it is not used to evaluate subject/lecturer.

هذا الاستبيان يمكن ان يقدم رؤيه اوليه بالنسبه لمطوري النظام ولكن مشاركه وتعاون مستقبلي اكبر سيكون مرجوا من طلابنا الراغبين في مساعدة الطلاب المستقبليين في المقرر من خلال مواصلة التحاور واقتراح مجموعه مفصله من المصادر لدراسة مادة اساسيات البرمجه. كما نرحب بحراره بالمشاركات بتسجيل تمارين من قبل الطلاب في مواضيع اساسيات البرمجه المختلفه. على الطلاب الراغبين التواصل مبدئيا على الايميل nejood.test@gmail.com وسنعلن لاحقا عن موقع للتواصل اجتماعيا للنقاش مع الطلاب الراغبين. البيانات التي ستجمع لن تستخدم الا كمصدر لبناء نظام للطلاب المستقبليين وليس لها علاقه بتقييم تدريس الماده.

1 Never ابدا	2 Sometimes احيانا	3 Often غالبا	4 Very often كثيرا جدا	**In your experience at your institution during the current academic year, about how often have you done each of the following?** **حسب اداءك في هذا الفصل الدراسي, كيف تصنف مشاركتك في كل من النشاطات ادناه؟**
				Asked questions or contributed to discussions in class or online سألت اسئله او قمت بالمشاركه في الصف
				Sought advice from academic staff قمت بأخذ النصح من احد الاسأتذه
				Made a class or online presentation قمت بتقديم عرض/شرح في احد الصفوف
				Worked hard to master difficult content عملت جاهدا لاجادة مواضيع صعبه
				Prepared two or more drafts of an assignment before handing it in قمت بالعمل عل اكثر نسخه للواجبات قبل التسليم النهائي (**إعادة الواجب**)
				Used library resources on campus or online قمت بإستخدام موارد المكتبه
				Worked on an essay or assignment that required integrating ideas or information from various sources قمت بالاستعانه بأكثر من مصدر لتجميع افكار لمساعدتك في حل واجب
				Came to class having completed readings or assignments حضرت الصف وانت دارس و مؤدي للواجبات
				Kept up to date with your studies كنت متابع للدروس (الدراسه اولا بأول)
				Worked with other students on projects during class قمت بالعمل مع طلاب اخرين في المشاريع او خلال الدراسه
				Worked with other students outside class to prepare assignments قمت بالعمل مع طلاب اخرين خارج الصف للتحضير للواجبات
				Put together ideas or concepts from different subjects when completing assignments or during class discussions قمت بالاستفاده من مفاهيم وافكار من مواد مختلفه عند اداء الواجبات او خلال المناقشات في الصف
				Tutored or taught other university students (paid or voluntary) قمت بتدريس طلاب جامعيين اخرين (تطوعا او مقابل دفع)
				Used an online learning system to discuss or complete an assignment
				Used email or a forum to communicate with teaching staff استخدمت الايميل او غيره من الوسائل للتواصل مع الاساتذه
				Discussed your grades or assignments with teaching staff قمت بمناقشة درجاتك وادائك في الماده مع الاساتذه
				Talked about your career plans with teaching staff or advisors قمت بمناقشة مواضيع مختلفه مع الاساتذه في غير وقت الصف
				Talked about your career plans with teaching staff or advisors - Discussed ideas from your readings or classes with teaching staff

1 Never ابدا	2 Sometimes احيانا	3 Often غالبا	4 Very often كثيرا جدا	**In your experience at your institution during the current academic year, about how often have you done each of the following?** **حسب اداءك في هذا الفصل الدراسي, كيف تصنف مشاركتك في كل من النشاطات ادناه؟**
				outside class
				Received prompt written or oral feedback from teachers/tutors on your academic performance تم اخطارك بتقييمك في الواجبات سواء شفهيا او بارجاع الاوراق
				Worked harder than you thought you could to meet a teacher's/tutor's standards or expectations قمت بمجهود اكبر مما توقعت انك تستطيع لتغطية محتويات المقرر
				Worked with teaching staff on activities other than coursework (e.g. students, family members, co-workers, etc.) قمت بمناقشة مواضيع مختلفه متعلقه بالماده مع اخرين خارج الصف (مثلا: زملائك, افراد اسرتك)

1 Very little القليل	2 Some البعض	3 Quite a bit كثير	4 Very much كثيرا جدا	**During the current academic year, how much has your coursework emphasised the following intellectual activities?** **من خلال دراستك لمادة اساسيات البرمجه, ماكانت كمية النشاطات العقليه ادناه خلال اداءك لأعمال السنه؟**
				- Memorising facts, ideas or methods from your subjects and readings تذكر حقائق, اجراءات, وافكار من الماده والدراسه
				Analysing the basic elements of an idea, experience or theory, such as examining a particular case or situation in depth and considering its components تحليل العناصر الاوليه للأفكار, التجارب او النظريات
				Synthesising and organising ideas, information or experiences into new, more complex interpretations and relationships ترتيب ووضع الافكار, المعلومات, والخبرات في نماذج متقدمه
				Applying theories or concepts to practical problems or in new situations تطبيق المفاهيم في مسائل عمليه وحل مسائل جديده

5 More than 6 اكثر من 6	4 5 to 6 5 الى 6	3 3 to 4 3 الى 4	2 1 to 2 1 الى 2	1 None صفر	**In a typical week, how many exercises, lab reports, problem sets and tutorial questions do you complete?** **كم عدد التمارين, والمسائل وواجبات المعمل التي تقوم بإكمالها في الاسابيع العاديه بحيث تتصف بالتالي؟**
					Number of pieces of work that take one hour or less to complete عدد الواجبات التي تتطلب ساعه او اقل لحلها
					Number of pieces of work that take more than one hour to complete عدد الواجبات التي تتطلب اكثر من ساعه لحلها

5 More than 20 اكثر من 20	4 11 to 20 11 الى 20	3 5 to 10 5 الى 10	2 1 to 4 1 الى 4	1None صفر	**During the current academic year, about how much reading and writing have you done?** **ماكمية نشاطات القراءه والكتابه التي قمت بها خلال دراستك للكورس**
					Number of assigned textbooks, books or book-length packs of subject readings عدد المراجع, الكتب وملخصات المواضيع المقرره
					Number of books read on your own (not assigned) for personal enjoyment or academic enrichment عدد الكتب المتعلقه بالماده التي قمت بدراستها دون ان تكون مطلوبه

7 Very much الكثير	6	5	4	3	2	1 Very little القليل	
							Which box best represents the extent to which your examinations during the current academic year have challenged you to do your best work? اي درجات التحدي جعلك التقييم في الماده تقدم افضل ماعندك من اداء (المجهود المطلوب)

1 Unhelpful, inconsiderate, rigid 2 2 3 3 4 4 5 5 6 6 7 Helpful, considerate, flexible ○ غير متوفرين, غير متعاونين, وغير متفهمين ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ متوفرين, متعاونين, ومتفهمين	Which of these boxes best represent the quality of your relationships with people at your institution? لأي مدى تقيم علاقتك بالاساتذه القائمين بتدريس مادة اساسيات البرمجه؟

8 Over 30 اكثر من 30	7 26 to 30 26 الى 30	6 21 to 25 21 الى 25	5 16 to 20 16 الى 20	4 11 to 15 11 الى 15	3 6 to 10 6 الى 10	2 1 to 5 1 الى 5	1 None صفر	About how many hours do you spend in a typical seven-day week doing each of the following? Leave blank if the item does not apply. **كم تقدر عدد الساعات التي تقضيها اسبوعيا خلال دراستك للكورس في كل من النشاطات ادناه؟**
								Preparing for class (e.g. studying, reading, writing, doing homework or lab work, analysing data, rehearsing and other academic activities) التحضير للصف (مثلا: الدراسه, القراءه, الكتابه, اداء الواجبات, واجبات المعمل وغيرها من النشاطات الاكاديميه)
								Working for pay off campus العمل مقابل المال خارج الجامعه
								Participating in extracurricular activities المشاركه في نشاطات اخرى في الجامعه غير الدراسه
								Relaxing and socialising (e.g. watching TV, partying, etc.) الاسترخاء والتواصل الاجتماعي
								Providing care for dependents living with you (e.g. parents, children, spouse, etc.) قضاء وقت لمساعده افراد الاسره (مثلا: الوالدين, الاطفال, الازواج)
								Managing personal business (e.g. housework, shopping exercise, health needs, etc.) اداء اعمال شخصيه (اعمال منزل, التسوق, التمارين وغيرها)
								- Travelling to campus (e.g. driving, walking, etc.) التنقل من السكن الى الجامعه
								- Being on campus, including time spent in class التواجد في الجامعه عموما بما فيها اوقات المحاضرات المختلفه
								Being on campus, excluding time spent in class التواجد في الجامعه باستثناء وقت المحاضرات

The upper sections were selected from – Australasian survey of student engagement – >below is our own

من افضل 5%	6	5	4	3	2	سيئ	
							How do you rank your level on the subject ماهو تقديرك لمستواك في الماده

ممتعه وبها تحدي	ممتعه وليست صعبه	مملة وصعبه	مملة وسهله	
				How did you find studying the subject كيف كانت دراستك للماده

اكثر من 75	75-71	70-66	65-61	60-56	55-45	رسوب	
							What was your score on the subject ماهي الدرجه التي احرزتها في الماده

Online courses كورسات اونلاين	Private tutoring شرح من جهات اخرى	Study online مصادر اونلاين	Lab practice التطبيق في المعمل	Study with others الدراسه مع الزملاء	books الكتب	lectures المحاضرات	
							What are your resources for studying the subject ماهي مصادرك لدراسة البرمجه (يمكن اختيار اكثر من خيار)
							What is the best resource ماهو افضل المصادر بالنسبه لك

كثير	6	5	4	3	2	قليل	
							الى اي مدى تعتمد على الدراسه اونلاين؟

- Did you regret studying **computer related specialization? 1-yes 2- no**
- هل ندمت على دخولك لتخصص متعلق بدراسة الحاسوب والبرمجه؟ 1- نعم 2- لا
- Would you cha**nge the field of your study if that was possible? 1-yes 2- no**
- هل ستقوم بتغيير التخصص اذا توفرت لك فرصه للاختيار مره اخرى؟ 1- نعم 2- لا
- Are you male or female?
- هل انت ذكر ام انثى؟ 1- ذكر 2- انثى
- Where do you live during the academic year?
- اين تسكن اثناء دراستك بالجامعه

....

- Could you practice java programming outside university?
- هل كان بإمكانك تطبيق برامج لغة الجافا خارج الجامعه؟
  - نعم: امتلك جهاز yes I have personal computer

- نعم: يوجد جهاز مشترك في المنزل we have shared computer at home
- نعم: يمكنني التطبيق في اجهزة الاصدقاء I can use my friends computers
- نعم: كنت اقوم بالدراسه مع زملاء يمتلكون اجهزه I used to study with friends having computers
- لا no

- Do you think it is necessary to practice programming out of university labs?
- هل تعتقد انه من الضروري القدره على تطبيق برامج الجافا خارج الجامعه؟
  - لا (الوقت في معامل الجامعه كافي) no (lab time is enough)
  - نعم yes
- Will it be useful if you could write and test java programs on mobiles?
- هل سيكون من المفيد تمكنك من كتابة برامج الجافا واختبارها في الموبايل؟

1- لا no 2- لحد ما sort of 3- مفيد useful 4- ضروره necessary

- Was it possible for you to dedicate more time in studying java programming?
- هل كان بإمكانك اعطاء دراسة مادة البرمجه المزيد من الوقت للدراسه
  - نعم yes
  - لا no

- In what activity would you use the extra time?
- في ماذاكنت ستستعمل الوقت الاضافي؟

....
....

- Did you receive advices from others when you started learning the course?
- هل قدم لك احد النصح عند البدء في دراسة الكورس؟ وماهي؟

....
....

- Were the advices useful?
- هل كانت النصائح مفيده؟

....
....

Did you receive assistance in studying the course from anyone beside the course staff? 1- yes 2-no

هل كنت تحصل على المساعده في دراسة الماده من غير الاساتذه؟ 1- نعم 2- لا

- Who helped you?
- ممن كنت تحصل على المساعده؟

……

……

- What advice will you give for future students?
- ماهي افضل نصيحه يمكنك تفيدمها للدارسين لمادة اساسيات البرمجه المستقبلين؟

……

……

- What do you suggest to make studying the course more engaging?
- ماهي التعديلات التي تقترحها لجعل دراسة الكورس اكثر جاذبيه

……

……

……

……

……

……

……

- Would you like to communicate and provide further feedback? 1- yes 2- no
- هل لديك الرغبه في مواصلة المشاركه في عملية تقديم الاقتراحات ومحتويات النظام مستقبلا؟ 1- نعم 2- لا
- If you would like to continue collaborating with us, please write your email
- اذا كان لديك الرغبه في مواصلة المشاركه يرجى كتابة ايميلك للتواصل مع المتطورين

……

APPENDIX B: PUBLISHED PAPERS

1. International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering, 2015

Eltegani, N. and Butgereit, L., 2015, September.Attributes of students engagement in fundamental programming learning. In (ICCNEEE), 2015 International Conference on (pp. 101-106). IEEE.

2.International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169

Nejood Eltegani, "Students' Perception about Fundamental Programming Course Teaching and Learning", September 17 Volume 5 Issue 9 , International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 06 – 10

3.The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM)

Nejood Eltegani**, Laurie Butgereit** , "**DRSUST: ADAPTIVE TUTORING SYSTEM TO INCREASE STUDENTS' ENGAGEMENT IN PROGRAMMING LEARNING": 2017 in The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM) Vol. 3, No 1.**