

ASSESSMENT AND CERTIFICATION OF SAFETY CRITICAL SOFTWARE

**A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE
in Computer Engineering**

**by
Bengisu UZUN YENİGÜN**

**JULY 2018
İZMİR**

We approve the thesis of **Bengisu UZUN YENİGÜN**

Examining Committee Members:

Assoc. Prof. Dr. Tolga AYAV

Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. M. İ. Can DEDE

Department of Mechanical Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Tuncay ERCAN

Department of Computer Engineering, Yaşar University

10 JULY 2018

Assoc. Prof. Dr. Tolga AYAV

Supervisor, Department of Computer Engineering
İzmir Institute of Technology

Assoc. Prof. Dr. Yusuf Murat ERTEN

Head of the Department of
Computer Engineering

Prof. Dr. Aysun SOFUOĞLU

Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, *Assoc. Prof. Dr. Tolga AYAV* who provided full support at all stages of my thesis process and contributed to my academic development.

I would like to express many thanks my dear professors to significant increase in my knowledge and experience *Assoc. Prof. Dr. M. İ. Can DEDE*, *Assoc. Prof. Dr. Gökhan Kiper*, *Dr. Barbaros Özdemirel*, *Prof. Dr. Enver Tatlıcioğlu*.

I would like to express my endless thanks to my husband *Ufuk YENİGÜN*, my parents and my brother who helped me on this challenging path.

I would like to express my thanks to my friends in *Dependable Components and Systems Research Laboratory* and *IZTECH Robotics Laboratory* and *IZTECH Prototyping Laboratory* for their supports.

Moreover, I would like to thanks all of my colleagues for their warm friendship and desirable behaviors.

Additionally, I would like to thanks my manager at work *İsa TAŞDELEN* for his support and encouraging me. Also, I would like to thanks *Tolga BAŞTÜRK*, *Ali ÇIKIKÇI* and my friends who are in *Department of Air Traffic Management* in *TÜBİTAK*.

This thesis is supported by *The Scientific and Technological Research Council of Turkey* (shortly: *TÜBİTAK* (TR: *Türkiye Bilimsel ve Teknolojik Araştırma Kurumu*))(Project No: *115E725* and *115E726*).

ABSTRACT

ASSESSMENT AND CERTIFICATION OF SAFETY CRITICAL SOFTWARE

Biomedical engineering which consists of combination of various dicipline such as chemical, biology etc. produces remedies to the challenges facing the globalizing world. Therefore, it covers a wide range of area from biosensor to surgical robots. Rapid development of biomedical engineering technology is aimed to research, diagnose and treatment to modern society for providing improved living conditions. Safety critical systems are interested with systems whose failure or malfunction could cause death, serious injury, damage to equipment and environmental harm. NeuRoboScope project is a safety-critical system which navigates the optical-camera system used in the endoscopic pituitary surgery by tracking the surgery tool that the surgeon uses. This thesis gives the information of how we scrutinize NeuRoboScope project depending on standards of medical device which are ISO 14971, IEC 62304, IEC 60601 and DO-178C.

ÖZET

EMNİYET KRİTİK YAZILIMLARIN DEĞERLENDİRİLMESİ VE SERTİFİKASYONU

Çeşitli disiplinlerden oluşan biyomedikal mühendislik, küreselleşen dünyanın karşı karşıya olduğu zorlukların çözüm yollarını üretmektedir. Bu nedenle, biyosensörden cerrahi robotlara kadar geniş bir alanı kapsar. Biyomedikal mühendislik teknolojisinin hızla gelişimi, gelişmiş yaşam koşullarının sağlanması için modern toplumu araştırmak, teşhis etmek ve tedavi etmektir. Emniyet kritik sistemleri, arıza veya ölüm, ciddi yaralanma, ekipmana hasar verme, çevreye zarar verebilecek sistemler ile ilgilenmektedir. NeuRoboScope projesi, cerrahın kullandığı ameliyat aracını izleyerek endoskopik pitituer cerrahide kullanılan optik-kamera sisteminde gezinen, güvenlik açısından kritik bir sistemdir. Bu çalışma, ISO 14971, IEC 62304, IEC 60601 ve DO-178C olan tıbbi cihaz standartlarına bağlı olarak NeuRoboScope projesini nasıl incelediğimizi göstermektedir.

I would like to dedicate this thesis to my husband *Ufuk YENİGÜN*, my parents *Hatice* and *Mustafa UZUN* and also my brother *İsmail Berk UZUN*.

TABLE OF CONTENTS

| | |
|--|------|
| LIST OF FIGURES | xi |
| LIST OF TABLES | xiii |
| LIST OF ABBREVIATIONS | xiv |
| CHAPTER 1. INTRODUCTION | 1 |
| 1.1. Pituitary Tumors and Gland | 1 |
| 1.2. Problem Statement | 2 |
| 1.3. NeuRoboScope | 3 |
| 1.4. Subsystem and Subsystem's Requirements of Surgical System | 4 |
| 1.4.1. Passive Balanced Arm - PEK | 4 |
| 1.4.2. Active Endoscope Holder - KEY | 5 |
| 1.4.3. Communication System - HAS | 7 |
| 1.4.4. Main Control Unit - AKS | 7 |
| 1.4.5. User Interface - KAR | 7 |
| 1.4.5.1. User Input Buttons and Visual Feedback | 7 |
| 1.5. NeuRoboScope Operation of Subcomponents | 8 |
| 1.5.1. Shut Down Mode Scenario | 8 |
| 1.5.2. Manual Mode Scenario | 9 |
| 1.5.3. Passive Mode Scenario | 9 |
| 1.5.3.1. Endoscope Is Held Mode Scenario | 9 |
| 1.5.3.2. Endoscope Is Detached Mode Scenario | 10 |
| 1.5.4. Wait Mode Scenario | 10 |
| 1.5.5. Active Mode Scenario | 11 |
| 1.5.6. Rotation Mode Scenario | 11 |
| 1.5.7. Dolly Mode Scenario | 11 |
| 1.6. Verification and Testing of Safety Critical Software | 12 |
| 1.7. Thesis' Aim and Objectivities | 12 |
| 1.8. Outline of Thesis | 13 |

| | |
|--|----|
| CHAPTER 2. RELATED WORK | 15 |
| CHAPTER 3. SAFETY CRITICAL SOFTWARE | 17 |
| 3.1. Safety Critical Software Life Cycle | 17 |
| 3.2. Safety Critical Software Standards | 18 |
| 3.3. Requirement Based Testing | 28 |
| 3.4. Verification and Validation of Safety Critical Software System | 28 |
| 3.4.1. Model-Based Validation | 29 |
| CHAPTER 4. MODEL BASED REQUIREMENT VERIFICATION | 30 |
| 4.1. Timed Automata..... | 31 |
| 4.2. Timed Automata Specification Language | 32 |
| 4.3. Model Checker Tool Uppaal | 33 |
| 4.4. NeuRoboScope System of Timed Automata | 33 |
| 4.4.1. Preliminary Information | 33 |
| 4.4.1.1. Channels | 34 |
| 4.4.1.2. Variables | 35 |
| 4.4.1.3. Surgeon Endoscope Holds Model..... | 36 |
| 4.4.1.4. Surgeon Control Model..... | 37 |
| 4.4.1.5. Surgeon Endoscope Movement Model | 38 |
| 4.4.1.6. Passive Balanced Arm Model | 39 |
| 4.4.1.7. Active Endoscope Holder Model | 39 |
| 4.4.1.8. Endoscope Region Model | 41 |
| 4.4.1.9. Green Light Model | 41 |
| 4.4.1.10. Red Light Model..... | 43 |
| 4.4.1.11. Battery Status Model | 43 |
| 4.5. Verification of System And Evaluation..... | 43 |
| CHAPTER 5. MODEL BASED MUTATION TESTING | 45 |
| 5.1. Verification With Mutation Testing..... | 45 |
| 5.1.1. Model Mutation NeuRoboScope UPPAAL Model | 45 |
| 5.1.2. Model Mutation Result | 49 |
| 5.1.3. Specification Mutation NeuRoboScope UPPAAL Model | 49 |
| 5.1.4. Specification Mutation Result | 54 |

| | |
|---|----|
| 5.2. Examination Uppaal Outputs | 54 |
| 5.3. Solving State Explosion Problem..... | 57 |
| 5.3.1. Apriori Algorithm | 58 |
| 5.3.2. Prefix Span | 58 |
| 5.3.3. Apriori and Prefix Span Results | 59 |
| 5.3.4. Greedy Heuristic Approach..... | 59 |
| | |
| CHAPTER 6. VERIFICATION WITH BLACK-BOX TESTING | 61 |
| 6.1. Unique MC/DC..... | 62 |
| 6.2. NeuRoboScope Cause Effect Graphs | 63 |
| 6.2.1. Preliminary Information | 63 |
| 6.2.1.1. Safety-Related Cause Nodes | 64 |
| 6.2.1.2. Light-Related Cause Nodes | 64 |
| 6.2.1.3. Safety-Related Effect Nodes | 65 |
| 6.2.1.4. Light-Related Effect Nodes | 65 |
| 6.2.1.5. Safety-Related Internal Nodes | 65 |
| 6.2.1.6. Safety-Related Constraint Nodes | 66 |
| 6.2.1.7. Light-Related Constraint Nodes | 66 |
| 6.2.2. Safety-Related NeuRoboScope Cause Effect Graphic | 66 |
| 6.2.2.1. Safety-Related NeuRoboScope Cause Effect Graphic Mathematical Transformation | 66 |
| 6.2.2.2. Reading of Safety-Related NeuRoboScope Cause Ef- fect Graphic Mathematical Transformation | 68 |
| 6.2.2.3. Unique MC/DC Test Suite of Safety-Related NeuRo- boScope Cause Effect Graphic | 68 |
| 6.2.3. Light-Related NeuRoboScope Cause Effect Graphic | 68 |
| 6.2.3.1. Light-Related NeuRoboScope Cause Effect Graphic Math- ematical Transformation | 69 |
| 6.2.3.2. Reading of Light-Related NeuRoboScope Cause Effect Graphic Mathematical Transformation | 69 |
| 6.2.3.3. Unique MC/DC Test Suite of Light-Related NeuRobo- Scope Cause Effect Graphic | 70 |
| 6.2.4. GraphML Files | 70 |

| | |
|--|----|
| CHAPTER 7. VERIFICATION WITH WHITE-BOX TESTING | 72 |
| CHAPTER 8. IMPLEMENTED TOOL | 75 |
| CHAPTER 9. CONCLUSION AND FUTURE WORK | 77 |
| REFERENCES | 78 |
| APPENDICES | |
| APPENDIX A. SAMPLE SCREENSHOTS OF VERTICALS | 82 |
| APPENDIX B. EXAMPLES OF TEST SUITE PRODUCED BY VERTICALS ... | 88 |
| APPENDIX C. EXAMPLES OF APRIORI AND PREFIX SPAN | 94 |

LIST OF FIGURES

| <u>Figure</u> | <u>Page</u> |
|--|-------------|
| Figure 1.1 Pituitary Tumor | 2 |
| Figure 1.2 Endoscopic Pituitary Surgery | 3 |
| Figure 1.3 Surgery Picture | 4 |
| Figure 1.4 Passive Arm | 5 |
| Figure 1.5 Active Arm | 6 |
| Figure 1.6 Analysis on Top 10 Device Areas in 2022, Market Share & Sales Growth (2016-2022) | 13 |
| Figure 1.7 Methodology | 14 |
| Figure 3.1 Basic Software Life Cycle Steps | 18 |
| Figure 3.2 Programmable Electrical Medical System Relation to IEC 62304 | 20 |
| Figure 3.3 Verification and Validation Under Light of DO-178C and IEC62304 ... | 21 |
| Figure 3.4 Tests To Be Carried Out Under Light of DO-178C | 22 |
| Figure 3.5 Tests To Be Carried Out Under Light IEC-60601-1 | 23 |
| Figure 3.6 Tests To Be Carried Out Under Light of IEC-62304 | 24 |
| Figure 3.7 Tests To Be Carried Out Under Light of 14971 | 26 |
| Figure 3.8 FTA Gates Explanation | 27 |
| Figure 4.1 Sample Example | 32 |
| Figure 4.2 Surgeon Hold Endoscope Model | 34 |
| Figure 4.3 Battery Status Model | 35 |
| Figure 4.4 Surgeon Endoscope Movement Model | 36 |
| Figure 4.5 Surgeon Control Model | 37 |
| Figure 4.6 Endoscope Region Model | 38 |
| Figure 4.7 Active Endoscope Holder Model | 40 |
| Figure 4.8 Green Light Model | 41 |
| Figure 4.9 Passive Balanced Arm Model | 42 |
| Figure 4.10 Red Light Model | 43 |
| Figure 5.1 Change Target Mutant Example | 46 |
| Figure 5.2 Change Selfloop Mutant Example | 47 |
| Figure 5.3 Change Source Mutant Example | 48 |
| Figure 5.4 Change Action Mutant Example | 50 |

| | | |
|------------|---|----|
| Figure 5.5 | Change Guard Mutant Example | 51 |
| Figure 5.6 | Change Invariant Mutant Example | 52 |
| Figure 5.7 | One Path Example | 57 |
| Figure 5.8 | Expected Result Example | 59 |
| Figure 5.9 | An Example of Test Cases | 60 |
| Figure 6.1 | Basic Operations Which Is Used When Creating Cause Effect Graph ... | 61 |
| Figure 6.2 | Constraints which is used when creating cause efect graph | 62 |
| Figure 6.3 | Safety-Related NeuRoboScope Cause Effect Graphic | 68 |
| Figure 6.4 | Light-Related NeuRoboScope Cause Effect Graphic | 70 |

LIST OF TABLES

| <u>Table</u> | | <u>Page</u> |
|--------------|---|-------------|
| Table 3.1 | Level Table of DO-178C | 24 |
| Table 3.2 | The level table according to IEC 62304 and ISO 14971 standard | 25 |
| Table 5.1 | Model Mutations Results | 49 |
| Table 5.2 | Specifications Mutations Results | 54 |
| Table 6.1 | Example Test Set | 63 |
| Table 6.2 | Test Suite For Safety Related NeuRoboScope Requirement | 69 |
| Table 6.3 | Test Suite For NeuRoboScope of Light Requirement | 71 |
| Table 7.1 | Required Coverage Depends on Safety Level | 72 |
| Table 7.2 | Example Algorithm Test Cases | 74 |

LIST OF ABBREVIATIONS

| | |
|----------------|---|
| ACTH..... | Adrenocorticotropic Hormone |
| GH..... | Growth Hormone |
| TÜBİTAK..... | The Scientific and Technological Research Council of Turkey |
| PEK..... | Pasif dEngelenmiş Kol |
| DoF..... | Degrees of Freedom |
| KEY..... | aKtif Endoskop Yönlendirici |
| ASB..... | Ameliyat Sınır Bölgesi Düğmesi |
| HAS..... | HABerleşme Sistemi |
| AKS..... | Ana Kumanda Sistemi |
| KAR..... | Kullanıcı ARayüzü |
| CEK..... | Cerrah Endoskobu Kavradı |
| AGA..... | Ana Güç Anahtarı |
| MKB..... | Manuel Kontrol Butonu |
| ADD..... | Acil Durum Düğmesi |
| VERTICALS..... | Verification and Testing of Safety Critical Software |
| FTA..... | Fault Tree Analysis |
| PRA..... | Probablistic Risk Assesment |
| FMEA..... | Failure Modes and Effects Analysis |
| FMECA..... | Failure Mode, Effects and Critical Analysis |
| HAZOP..... | Hazard and Operability Analysis |
| MBV..... | Model Based Verification |
| CTL..... | Computational Tree Logic |
| TCTL..... | Timed Computational Tree Logic |
| ASO..... | AssociativeShiftOperator |
| CRO..... | ConstantReplacementOperator |
| ENO..... | ExpressionNegationOperator |
| LRO..... | LogicalOperatorReplacement |
| MCO..... | MissingConditionOperator |
| RRO..... | RelationalOperatorReplacement |
| SNO..... | SimpleExpressionNegation |
| STO..... | StuckAtOperator |

VRO VariableReplacementOperator

CHAPTER 1

INTRODUCTION

NeuRoboScope is important tool to used by surgeon during Endoscopic Endonasal Transsphenoidal surgery to heal the pituitary tumors. Pituitary tumors and gland are very common in modern society and affects human life negatively. These are described in Section 1.1. Endoscopic Endonasal Transsphenoidal surgery is one of the most advanced new methods. However, there are some problems faced by surgeons due to long-term use during endoscope use as mentioned Section 1.2. The device of NeuRoboScope is a solution to this problem that surgeons face. The introduction and extensive information about the device are described in the following Sections 1.3, 1.4 and 1.5. Model based requirement verification and model based mutation testing and black box testing with cause effect graphs of our safety critical system have been performed. Under this scope, VERTICALS tool has been developed. It will be extremely easy to produce test suites by specially testing safety critical systems with the this vehicle. It is discussed extensively in the future.

1.1. Pituitary Tumors and Gland

The pituitary gland is a center located at the base of the brain and produces and regulates some hormones/chemicals that influence organs and basic body functions such as Oxytocin, Vasopressin, Adrenocorticotrophic hormone (ACTH) and Growth hormone (GH). The pituitary gland and tumors are shown in Figure 1.1.

The vast majority of tumors that develop from the pituitary gland are benign. It can be seen rarely in malignant tumors showing cancer. Because of the fact that benign tumors press on pituitary gland, endocrine system's is affected directly. For this reason, the tumor may cause mood changes, sexual and reproductive problems, memory loss, eating disorders, unusual hair growth and many other symptoms.

Various hormone tests and MR imaging are performed in the diagnosis of pituitary tumor. The treatment varies according to the size of the tumor. If the detected adenomas do not secrete hormones, give no symptoms, and are not large enough to undergo com-

pression, they are only followed.

Prolactin secreting pituitary tumor (prolactinoma) is treated primarily with medication. Some medicine are used for this purpose. These drugs reduce both the level of prolactin and the size of the tumor. Growth hormone secreting tumors also use these drugs. However, they are not as effective as prolactinomas.

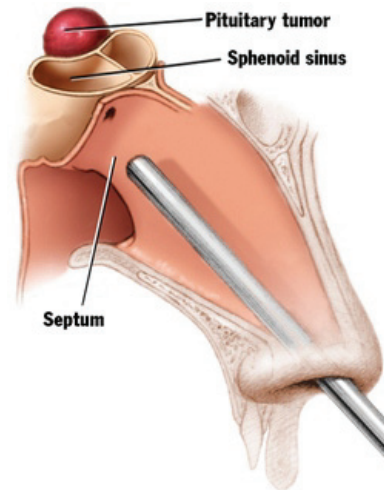


Figure 1.1. Pituitary Tumors (Healty-Inside, 2018)

Surgical treatment (surgery) is the first option in some tumors, but not in others. This depends on whether the tumor responds to its size and drug treatment. Today, pituitary adenoma resections can be performed in three different ways. These are conventional open skull surgeries, microscopic transsphenoidal surgeries and endoscopic endonasal surgery.

In the treatment of pituitary adenomas, radiation therapy may be given along with medication and surgery. However, because of the frequent occurrence of pituitary insufficiency after radiation therapy, this treatment is being used with caution.

1.2. Problem Statement

Endoscopic Endonasal Transsphenoidal surgery is performed through the sphenoid sinus to remove tumours as shown in Figure 1.2. It is the best new emerging method

to allow for minimally invasive and maximally effective procedure for pituitary adenomas.

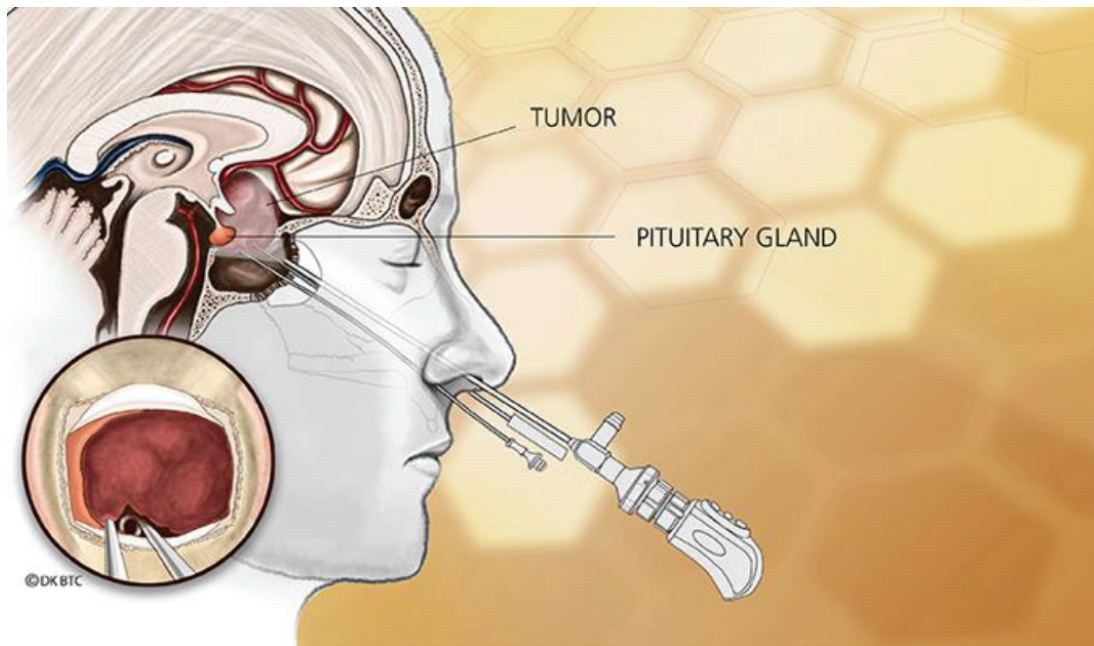


Figure 1.2. Endoscopic Pituitary Surgery (Barkhoudarian, 2015)

Controlling endoscope by one hand during surgery, surgeon operates tools to provide the visual about operating area. One surgery takes usually 2-3 hours(Laws et al., 2016; Almeida et al., 2015; Berker et al., 2013). Reserved hand during surgery for endoscope reduces moving ability to surgeon. Endoscope is not just one equipment throughout surgery time, there have been also several tools such as aspirator, curette, drill and lancet, at the same time. In addition to all, the surgeon gets tired of holding constantly endoscope(Schurr et al., 1999). Figure 1.3 shows commonly visible operating scene.

As it can be seen in Figure 1.3, surgeon's both hands are full. Moreover, surgeon assistants help to aspriate surgery area. NeuRoboScope, which is also called robot assisted endoscope control, gives opportunity to support endoscopic pituitary surgery. It gives surgeon a chance to use efficiently surgeon's both hands during surgery with robot assistant which can be controlled by surgical tools(Wang et al., 1999; Kim et al., 2004).

1.3. NeuRoboScope

Robot-Assisted endoscope Control that can be controlled by the surgical tools which is also called shortly NeuRoboScope is a project supported by TÜBİTAK which is the leading agency for management, funding and conduct of research in Turkey.

1.4. Subsystem and Subsystem's Requirements of Surgical System

Surgical system has five main subsystems. These are Passive Balanced Arm (shortly: PEK (TR: Pasif dEngelenmiş Kol)), Active Endoscope Holder (shortly: KEY (TR: aKtif Endoskop Yönlendirici)), Communication System (shortly: HAS (TR: HAberleşme Sistemi)), Main Control Unit (shortly: AKS (TR: Ana Kumanda Sistemi)) and User Interface (shortly: KAR (TR: Kullanıcı ARayüzü))(Ates et al., 2018).



Figure 1.3. Surgery Picture

The subcomponents of the system and component's requirements details are mentioned in the following sections.

1.4.1. Passive Balanced Arm - PEK

PEK has a balanced mechanical structure that carries the KEY as shown in Figure 1.4. There will be no actuator systems on it. The surgeon will move the KEY to desired position with minimal effort during surgery. When desired position and direction is reached, the movable joints can be mechanically or electromechanically locked to. At each joint, there will be an encoder for position measurement. This component has 6 degrees of freedom structure. Three of them are on the wrist point and the others are on arm structure.



Figure 1.4. Passive Arm

1.4.2. Active Endoscope Holder - KEY

KEY is the one of the main part of the robotic arm which has a 3 DoF parallel kinematic structure 2R1T (2 revolute and 1 translational) mobile platform motion as shown

in Figure 1.5 (Yaşır and Kiper, 2018). It is designed to hold and position the endoscope according to the commands received from the master system. The actuator systems on the endoscope will orient the pivot point of the endoscope along the axis of yawing and pitching and provide linear movement in the direction of advancement. Turning around the pivot point will be guaranteed with mechanical pivoting (Tr: Mekanizma Yapısı) or non-mechanical pivoting. The Surgery Zone Button (Tr: Ameliyat Sınır Bölgesi Düğmesi) (ASB) will be determined by choosing the pivot point. For mechanical pivoting, the

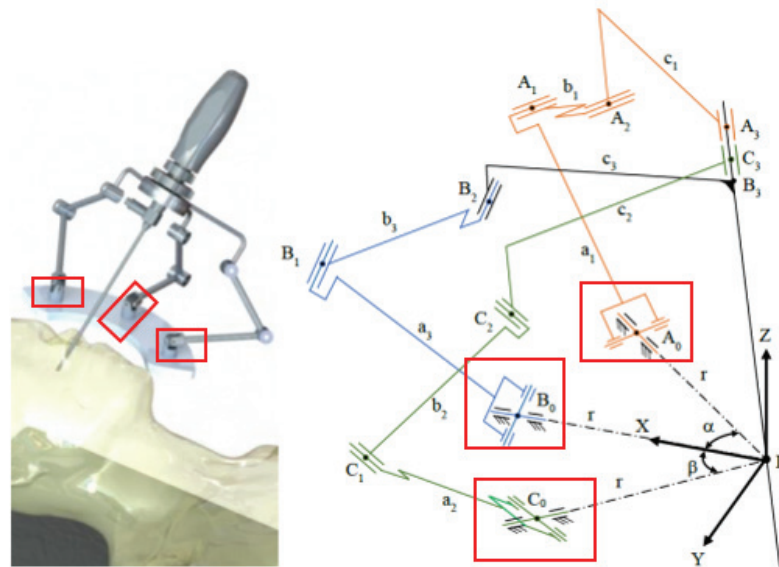


Figure 1.5. Active Arm

pivot point markers must be active when the KEY is locked. They must be inactive when KEY is active. There must be a manual pivot adjustment system. This pivot point is determined by the doctor before surgery and the manual adjustment is performed accordingly. The pivot point is also marked before the start of the operation.

For non-mechanical pivoting, the pivot point is marked prior to the start of the operation and each exit-entry from the operating area. The pivot point in the global axis set is defined by the information from the PEK encoders. The pivot point is ensured by control when KEY is online.

It will only be active in the surgical area. The axes can be braked when requested and in an emergency. With one brake command all axes will be locked. It can be moved backwards active or passive. The work area will be limited to $\mp 20^\circ$ for the pitching and

yawing movements. AKS will be able to drive with the motions to be transmitted via HAS. It will be able to transmit touch information (force feedback) to the AKS via the HAS. The limits of the power levels of the motors are set at 100W.

1.4.3. Communication System - HAS

This system provide bi-directional data transmission at 10hz with Bluetooth Low Energy protocol between KEY and AKS. It is able to provide wireless communication uninterruptedly.

1.4.4. Main Control Unit - AKS

It will allow the surgeon to transmit motion information to the KEY at any time. It must be sterilizable to provide patient. It consists of two units: ring and foot pedal. The touch information from KEY will be visually communicated via KAR, which should not disturb the surgeon's attention.

1.4.5. User Interface - KAR

This is the key system to provide safety and ease of use. It is a light warning system which is used to make a user state notification.

1.4.5.1. User Input Buttons and Visual Feedback

User input buttons are described as follows:

1. *Surgeon Holds Endoscope (shortly: CEK (Tr: Cerrah Endoskobu Kavradı))* : Depending on the way the surgeon presses this key, the operating scenario will change and be transmitted to the surgeon with a LED or vibrating element to be placed on this structure attached to quick-replace. In addition, if the CEK is fully pressed, the surgeon will be able to separate the endoscope from the robot. When the surgeon

wishes to use the endoscope manually, the surgeon will half press CEK to disable the KEY.

2. *Main Power Button (shortly: AGA (Tr: Ana Güç Anahtarı))* : The power to go through the whole system is over this switch. Since all brakes will be de-energized when AGA is off, all joints are locked.
3. *Manual Control Button (shortly: MKB (Tr: Manuel Kontrol Butonu))*: Allows the system to be driven manually. It works semi-correlated with Emergency Button.
4. *Emergency Button (shortly: ADD (Tr: Acil Durum Düğmesi))*: If the ADD is pressed, the MKB is activated and the passive arm releases the necessary brakes so that the surgeon can move the endoscope out of the ASB without removing it from the system.

Visual feedback LEDs are described as follows:

1. *Green Light*: The light is on, KEY is online but not active. When the light is blinking, KEY is active or centering.
2. *Red Light*: The light is on when the centering can not be done because it is very close to the patient.
3. *Yellow Light*: It is on when surgeon excessive force/moment is on the tissue with slave system.

1.5. NeuRoboScope Operation of Subcomponents

In this section, scenarios have been established for the system.

1.5.1. Shut Down Mode Scenario

It is the mode in which the endoscope is shut down.

1. The AGA is in the OFF state and power is not supplied to the system.
2. PEK and KEY are locked.

3. All lights are off.
4. ADD, MKB, CEK, KBE buttons are inactive.
5. AKS, HAS, KAR are closed.
6. The healthcare team should leave the system in this mode before surgery begins and after surgery ends.

1.5.2. Manual Mode Scenario

It is the mode in which the surgeon moves the endoscope manually during surgery.

1. The KEY brakes are active, PEK arm released but wrist locked.
2. To switch to this mode, it must be pressed the MKB button.
3. All the lights are off.
4. This mode is also opposite to ADD. That is, if ADD is pressed, the MKB automatically turns ON and it allows the system to operate manually.

1.5.3. Passive Mode Scenario

It is switched to this mode according to the state of the CEK. The CEK switch has priority over the foot pedal. For this reason, if the CEK key is active, no information will be transmitted from the ring, even if the foot pedal is pressed. According to the activation of the CEK, the passive mode operates in two submodes.

1.5.3.1. Endoscope Is Held Mode Scenario

The surgeon move the endoscope backwards of ASB without detaching the endoscope.

1. CEK is in half-pressed position and LED is active to move robot arm while the endoscope is attached.
2. KEY brakes are locked. B0 of PEK brakes, except 5 prismatic joints.
3. PEK is not balanced, the surgeon is expected to carry an endoscope.
4. To get out of this mode, the surgeon should release CEK close to ASB. After a short wait, the system will center and enter Wait Mode in order to be ready to switch back to Active Mode.

1.5.3.2. Endoscope Is Detached Mode Scenario

It is the mode that the surgeon enters when it separates the endoscope from the system.

1. CEK is in full pressed state.
2. The endoscope is completely disconnected from the system.
3. No action will be taken if the endoscope is not connected to the system. In other words, it is not done when the centering endoscope is not installed. Because we do not have any information on whether or not the endoscope is attached to the system. Since the system can not connect structure of quick-replace pin, surgeon can only see that pin state mechanically.
4. KEY brakes and PEK brakes are completely locked.
5. The surgeon must attach the CEK to the half-pressed position when the endoscope is attached the system. On this count, the Endoscope is entered in the Endoscope is held mode. Before the releasing CEK, it is expected from surgeon to move to a position close to the ASB.
6. When CEK is released, the system is ready to switch to Active Mode.

1.5.4. Wait Mode Scenario

It is the most common mode of surgery.

1. AGA, ADD, CEK are not active.
2. In this mode, the surgeon knows that the system is ready for surgery and that when the foot pedal is pressed, the robot arm will move according to the information transmitted from the ring.
3. KEY and PEK are locked.
4. In the case of entering active mode except for entering from wait mode to active mode, the system is to check if centering is necessary for KEY before returning to wait mode. This is particularly important for situations in which the previous position of the KEY may have reached the limits of the surgical area.

1.5.5. Active Mode Scenario

It is the mode that the system in the wait mode enters by triggering the foot pedal. The KEY brakes are free, the PEK brakes are fixed with a limited resistance force. This resistance force should not be affected by small impacts and should be so high as not to disturb the operation of the KEY and weak enough not to harm the patient under extraordinary conditions. It has two sub modes.

1.5.6. Rotation Mode Scenario

Pressing the pedal once will enable operation in this mode (and held down). Yaw and pitch motions in the ring enable the endoscope to move forward and backward and to the right and left in the plane.

1.5.7. Dolly Mode Scenario

Pedal is operated in this mode by pressing twice (and held down). The spinning motion around its own axis in the ring allows the endoscope to go in or out to its position. CEK brakes are released and PEK brakes are locked.

1.6. Verification and Testing of Safety Critical Software

Verification and Testing of Safety Critical Software is named shortly VERTICALS which is an open-source web-based software system testing tools which provides generating test cases to safety critical systems. VERTICALS is developed under scope of this thesis. It gives an opportunity to establish readable test suites which is created by UPPAAL. UPPAAL(UPPAAL, 2012) is model checking tool based on timed automata model.

1.7. Thesis' Aim and Objectivities

The global medical market has already exceeded 289 trillion dollars. According to 2016 Top Markets Report Medical Devices which is released by U.S. department of commerce of International Trade Administration predicts continually growing table about the next five years.

Producing more new medical device software is inevitable. US Food and Drug Administration reported that %15 medical devices recall related to software error from 2008 to 2012(Alemzadeh et al., 2013).

Safety-critical medical devices must provide specific system abide by standards for saving patient's life, its quality, device's user and device's environment. In this thesis, NeuRoboScope is assessed by scope of ISO 14971, IEC 62304, IEC 60601 and DO178C.

The aim of this thesis is to obtain assessment safety critical software under light of standards such as DO-178C, IEC 60601-1, ISO 14971. NeuRoboscope which is one of the safety critical systems scrutinize with these standard applications.

The objectives of this thesis are:

- Assess safety critical software systems and it's life cycle.

- Verify NeuRoboScope with model-based testing.

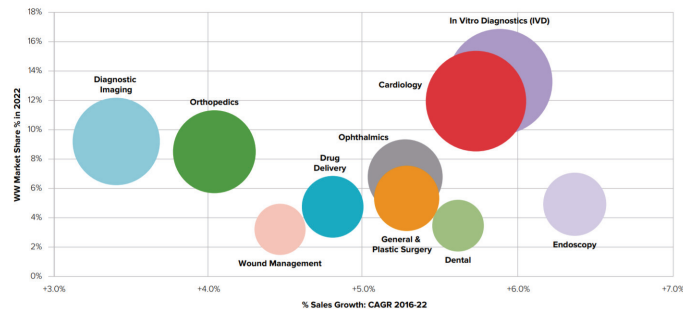


Figure 1.6. Analysis on Top 10 Device Areas in 2022, Market Share & Sales Growth (2016-2022)(EvaluatePharma, 2017)

- Verify NeuRoboScope with mutation model and specification.
- Verify NeuRoboScope with cause effect graph testing.
- Verify NeuRoboScope with white box testing.

1.8. Outline of Thesis

The organization of the thesis as follows :

- Chapter 2 shows related works;
- Chapter 3 "Safety Critical Software" shows how to evaluate the safety critical system under the light of standards;
- Chapter 4 shows model based verification to NeuRoboScope;
- Chapter 5 shows verification of NeuRoboScope with model mutation and specification mutation;
- Chapter 6 shows verification of NeuRoboScope with black box testing;
- Chapter 7 shows verification with white box testing;

- Chapter 8 shows the implemented tool;
- Chapter 9 concludes this thesis. It provides a summary and future research.

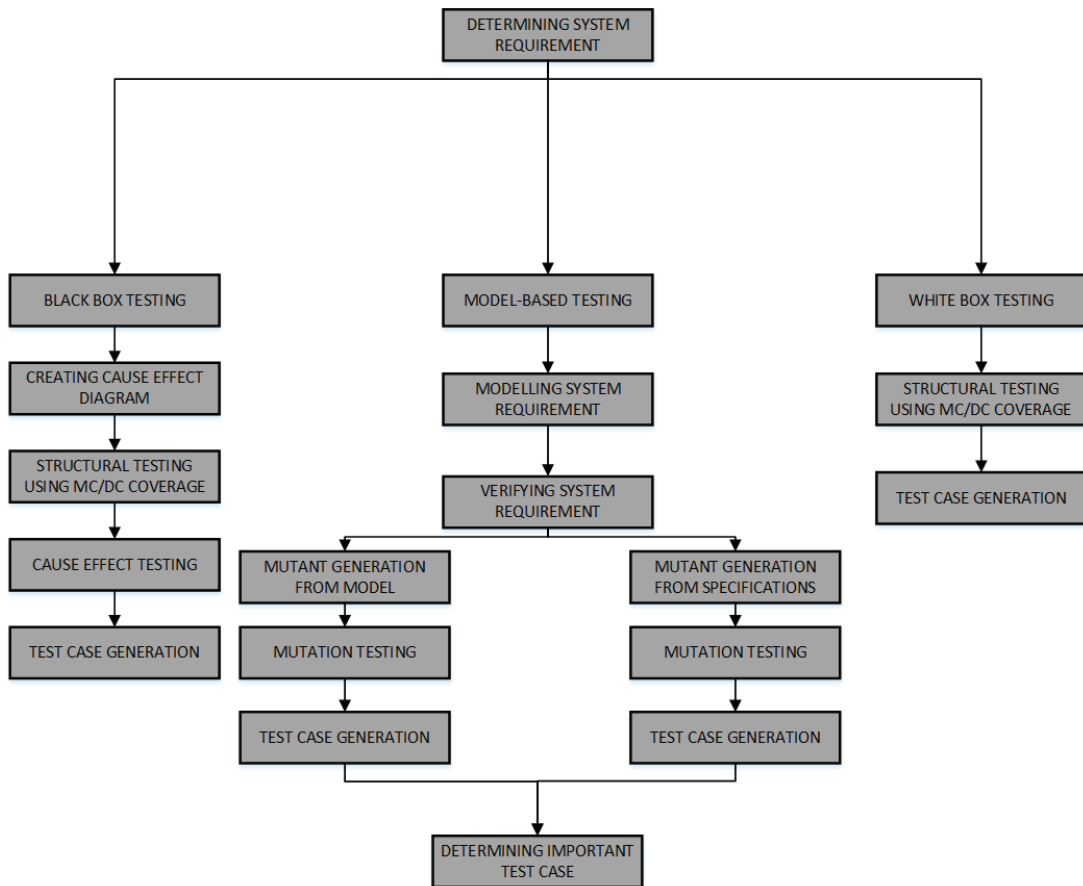


Figure 1.7. Methodology

CHAPTER 2

RELATED WORK

There are many works published till today in this domain. Some of are:

Palanichamy, G. (2010), released a white paper about the risk management process. In this white paper, the risk management process, which required for determining the risks in the production of medical devices and bringing them to an acceptable level, is explained. In addition, FTA and FMEA analyzes are mentioned as inputs to the design.

Alemzadeh et al. (2013) released a study about past medical device accidents which is published by FDA (Food and Drug Administration). According to this article, the FDA classifies medical device accidents as computer-related and non-computer-related. For computer-related accident, the effect of the software has been researched. The software effect has been shown to be a major influence on medical device accidents.

Knight, J. (2002) stated in the article that defective software definitions in safety-critical systems can directly lead to errors. It is stated that the whole system should be analyzed very well.

Hrgarek, Nadica(2012) released a study about common certification and regulatory such as IEC 62304 and ISO 14971. This study represents the relational table of the requirements of IEC 62304 for Classes A, B and C.

Malm T. et al. (2009) research software safety model for safety critical applications, this research includes real-time systems protections.

Dan Y. and Shilong M. (2011) interested verification of requirements for safety-critical software.

Lutz R. R. and Mikulski I. C. (2003) researched requirements discovery during the testing of safety-critical software.

Pei, J et al. (2001) published an article about PrefixSpan which is one of sequential patterns mining algorithm. It solves the problem of large projected databases.

Agrawal, R. and Srikant, R. (1994) published an article about Apriori algorithm to find frequent item sets from a transaction dataset and derive association rules which is most useful data mining algorithm which finding frequent item set from database.

Behrmann, G. et al. (2004) published a tutorial to use UPPAAL tool which is

model checker developed by Uppsala University and Aalborg University.

Fraser, G. et al. (2008) studied about mutation operator for timed automata specifications such as Logical Operator Replacement(LRO) and Relational Operator Replacement(RRO).

Aichernig, B.K. et al. (2013) studied about model based mutation such as change action and change source.

Ayav, T and Belli, F. (2015) published an article that cause effect graph's test cases can be generated by turning graph to boolean expression with MC/DC coverage technique.

Ammann P. et al. (2002) reached model checkers and computational tree logic . And also they used counter examples of model checkers to test cases.

Chilenski and Miller (1994) reached the modified condition/decision coverage (MC/DC) and evaluated the advantages and disadvantages.

Kelly, J et al. (2001) reached MC/DC coverage to generate test cases from boolean expressions. This study shows the types of structural coverage and their criteria.

Chilenski (2001) evaluated three forms of MC/DC with their strengths and weaknesses.

Chang J. R. and Huang C. Y. (2007) propose a method which focuses on all conditions of Boolean expression to practice MC/DC and their developed regression testing tool TASTE(Tool for Automatic Software regression TEsting).

CHAPTER 3

SAFETY CRITICAL SOFTWARE

The term of safety means the condition of being protected from or unlikely to cause danger, risk, or injury. Safety critical also known as a life critical system is a system which failure or malfunction may result in death or serious injury to people, loss or severe damage to equipment/property or environmental harm. Safety and quality are inseparable. To provide safety, quality processes should be carried out.

3.1. Safety Critical Software Life Cycle

In order to develop a safety critical software safe and high quality, it is necessary to carry out the life cycle under light of standards. At each of life cycle producer keeps assurance evidence in order to provide evidence that the system is acceptably safe.

The basic steps of life cycle of a safety-critical software during both production and use includes the following stages:

- **Planning:** It is the first and the most important phase of the life cycle. The basic needs of the project are identified and feasibility studies and planning are carried out during this phase. Project Management Plan, Quality Management Plan, Configuration Management Plan, Integration Plan, Validation and Verification Plan and Safety Plan the most important outputs of this phase.
- **Analysis:** The system will ensure that all requirement is correctly identified and recorded.
- **Design:** High level design (architectural design) and detailed design are realized after determining the requirements for the system.
- **Development:** After the design phase reaches a certain degree, the implementation phase is started.
- **Testing and Evaluation:** The produced system must be tested all steps of life-cycle to reduce the cost and error rate.

- **Implementation and Maintenance:** After all the test steps have been carried out, the product is shipped to the market in implementation phase. The deliverable should include the product, the operating documentation and, if applicable, the version differences in the document. Maintenance phase includes different activities such as problem solutions, infrastructure improvement.

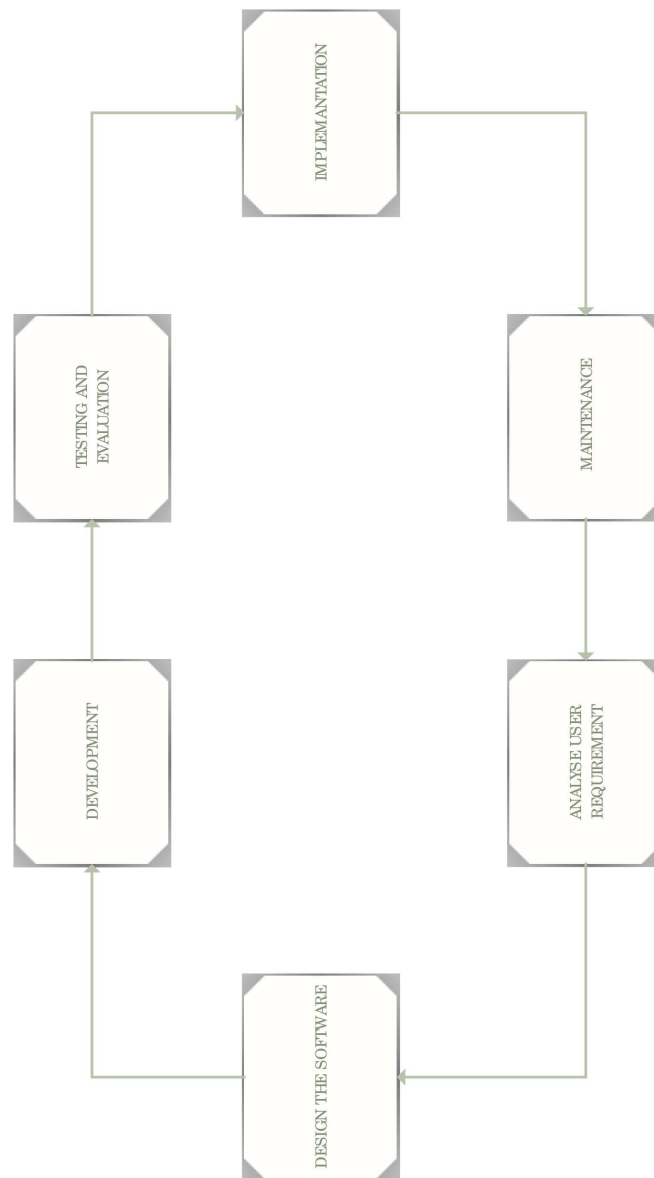


Figure 3.1. Basic Software Life Cycle Steps

3.2. Safety Critical Software Standards

Safety critical system standards are prepared with the lessons learned from the accidents that have taken place until today. Basically their principle is "I can not do it again if I do not write what I do correctly, I make mistakes again if I do not write what I do wrong." Therefore standards ensure that the evidence and outputs of the projects are recorded. The records will be returned in case of an accident to find where the fault originated.

The standards provide the context of the experience, understanding and opinions systematically accepted by the producers as content and scope. Safety-critical projects in the health sector need to specific standards to protect both the patient's life and quality of life as well as the environment. It is necessary to determine, provide and manage the requirements in a systematic approaches(Knight, 2002). In preparation of this device with DO-178C standard, the control of this systematic process is completed. The monitoring process must be completed in order to identify the hazards that may occur with the use of the device, to estimate the risks, to evaluate, to control and to ensure the accuracy of the controls(RTCA, 2011). ISO 14971 standard aims to ensure that the endoscope device completes this process correctly(ISO, 2007). The IEC 62304 (IEC, 2006b) standard is intended to control the software of the endoscopic medical device and the scope of the software life cycle processes. IEC 60601(IEC, 2005) standard aims to investigate the need for basic safety and performance requirements that the device must have.

The verification process in the DO-178C light can be summarized visually in Figure 3.4

Verification and validation scrutinize under light of DO-178C and IEC62304 like a waterflow model in Figure 3.3 to visualise easily however they are support V model to life cycle as shown in Figure 3.2.

We need to verify the system and software with static and dynamic analysis of the standards we have examined in our biomedical device.

Static Analysis: It is used to detect errors and check the features of the software before compilation. This analysis is very important in terms of safety, quality, operating performance, functionality and traceability of the system.

Static Code Analysis Using Overhead: The code should be analyzed by direct eye scan. In terms of safety, the parts that may cause the failure are examined.

Static Code Analysis Using Automated Tools: Code analysis is performed using

automated tools. It should not be forgotten that the earlier the vulnerability is detected, the cheaper and faster it can be fixed.

Requirement Coverage Analysis: In our safety-critical system, static verification aims not only to make the system run smoothly, but also to improve the system by providing the necessary scope of requirements for security.

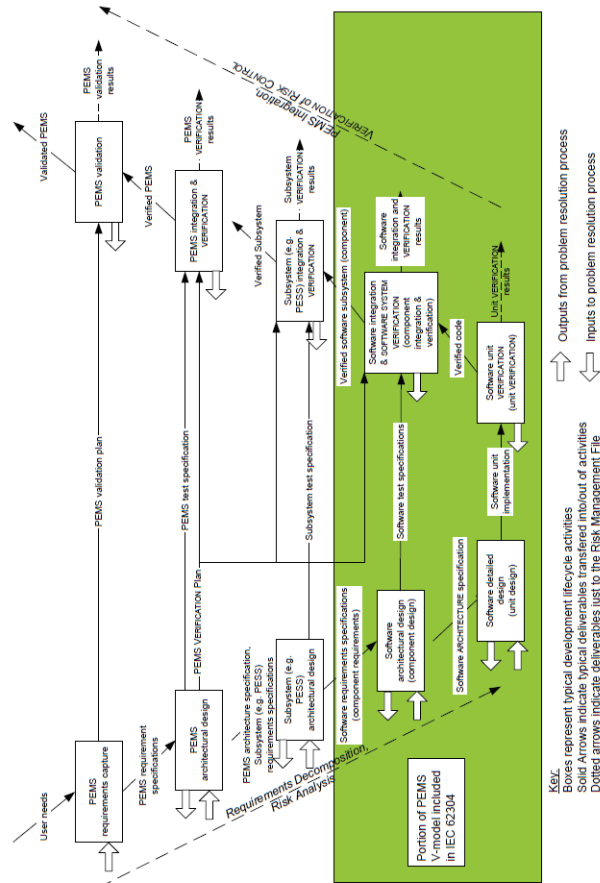


Figure 3.2. Programmable Electrical Medical System Relation to IEC 62304

Test Cases Review: The purpose of testing is to ensure that all requirements can be verified in actual test cases through actual testing. Our main goal is to ensure that all test and all requirements are covered correctly.

Dynamic Analysis: It is used to observe how the system works in the context of changing data and is a test operation using test cases selected from infinite system states to find cases where the system does not fit the expected behavior. Dynamic tests are used to perform tests such as structural coverage analysis, performance, compatibility, and stress

compliance.

Structural Coverage Analysis: Structural coverage analysis tests the extent to which a safety critical software can be run in the test cases in which it is generated. With this analysis, it is aimed to test all the lines of the code.

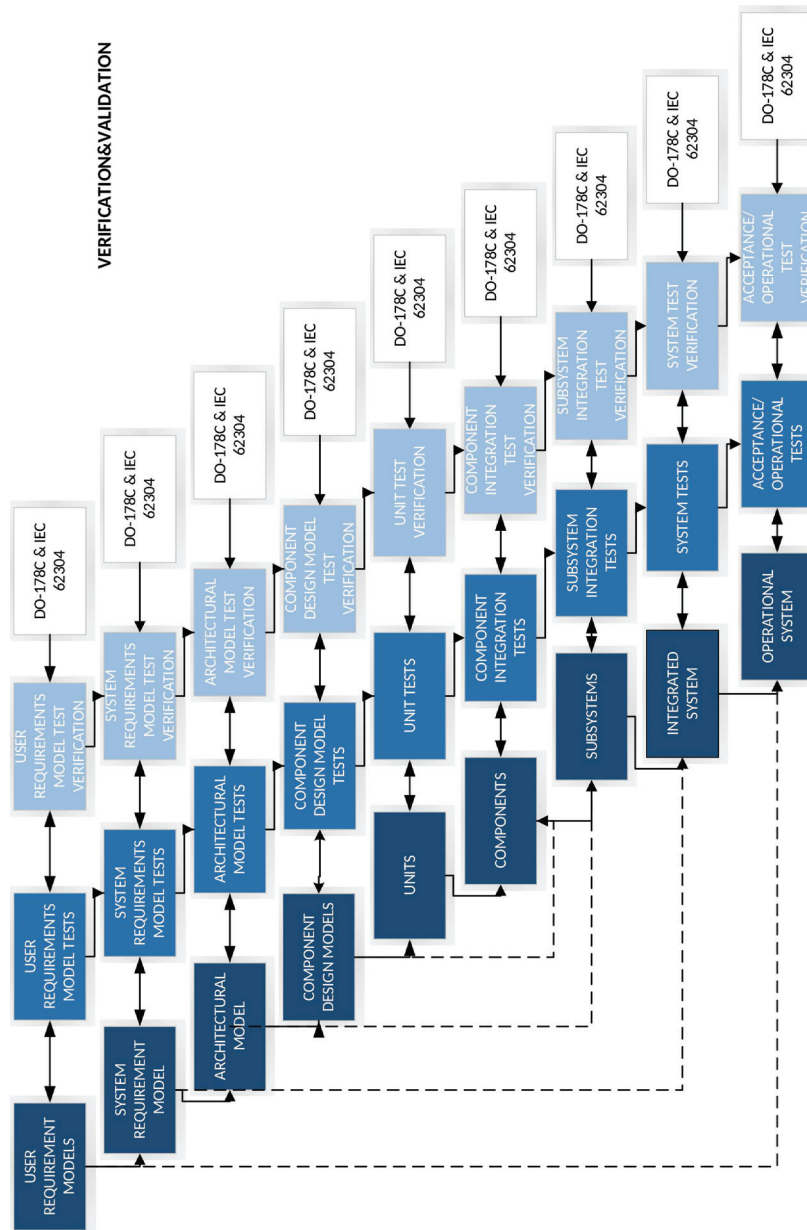


Figure 3.3. Verification and Validation Under Light of DO-178C and IEC62304

Performance Test: Observation of how the system behaves under overloads to measure its performance.

Compatibility Tests: It is intended to be tested to show whether the systems are

The necessary measures are taken to ensure that the safety objective is achieved to reduce the likelihood of unwanted and unacceptable errors to an acceptable level by creating a system report using the analysis methods described above.

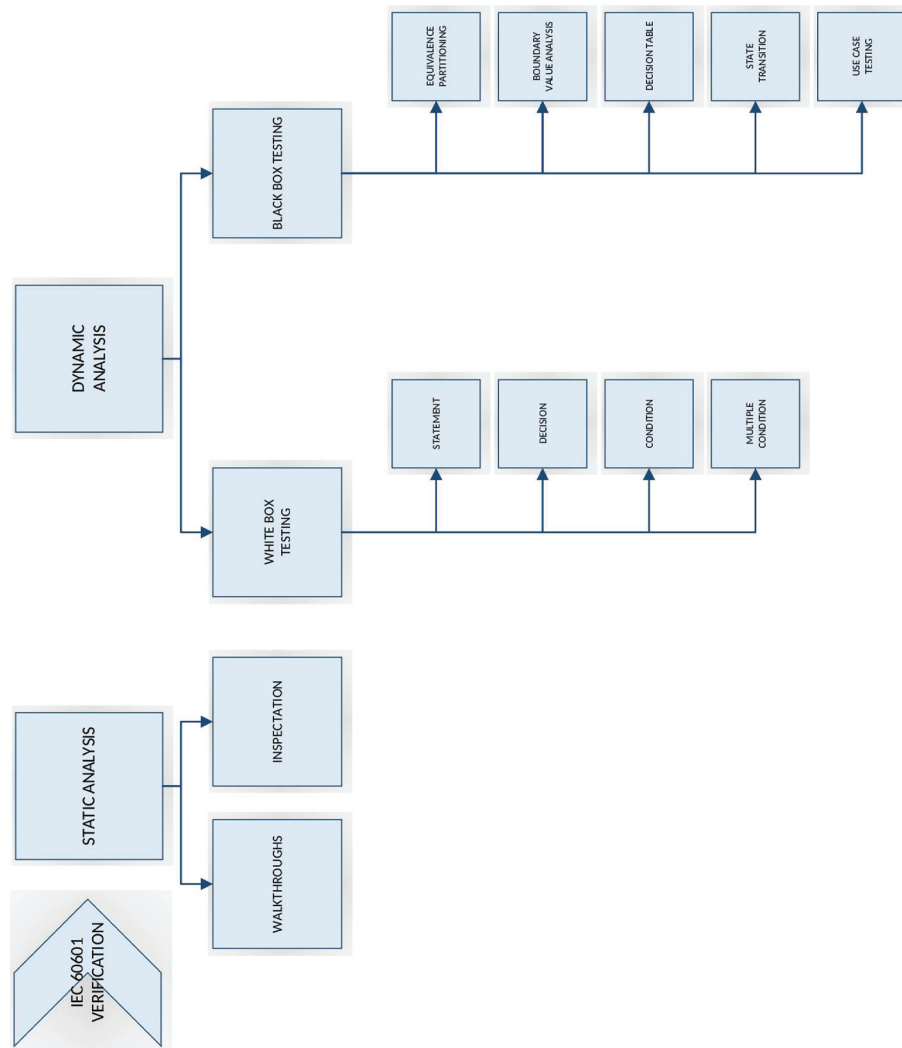


Figure 3.5. Tests To Be Carried Out Under Light IEC-60601-1

If the safety target has not been achieved, appropriate precautions should be taken, such as warning lights, pre-defined and user-reported procedures. ISO 14971 standard states that if there is a risk of serious problems in the system, if the patient’s health is bad, the use of the device depends on the decision of the patient and the physician by the expert and experienced physicians with regard to the health of the patient.

Because of the slightest margin of error of our device can cause fatal or serious injuries to the patient, Our medical device is safety critical. It is Level A according to

DO-178C standard as shown Table 3.1 and Class C according to ISO 62304 as shown Table 3.2.(Hrgarek, 2012).

Table 3.1. Level Table of DO-178C

| Level | Effects of Anamolous Behaviour | Explanation |
|-------|--------------------------------|---|
| A | Catastrophic | Failure cases resulting in death |
| B | Hazardous | Failures that can cause serious injury and death risk |
| C | Major | Situations that cause injury or risk of injury |
| D | Minor | Failure situations that do not put safety in great danger |
| E | No Effect | Fault conditions that do not affect system or user capabilities |

The testing and safety engineering methods under scope of the ISO 14971 can be summarized visually in Figure 3.7, under scope of the IEC-60601-1 in Figure 3.5, under scope of IEC-62304 in Figure 3.6, under scope of in Figure 3.4.



Figure 3.6. Tests To Be Carried Out Under Light of IEC-62304

Safety Engineering: Safety engineering is a type of engineering discipline which assures that engineered systems provide acceptable levels of safety.

Specific procedures and techniques from safety engineering:

1. Probabilistic risk assessment (PRA),
2. Failure modes and effects analysis (FMEA),
3. Fault tree analysis (FTA),
4. Failure mode, effects and critical analysis (FMECA),
5. Hazard and operatibility analysis (HAZOP),

6. Hazard and risk analysis,
7. Cause and effect diagrams.

Fault Tree Analysis (FTA) : FTA is a graph representation of the conditions that cause or contribute to undesirable event (Top event). FTA helps us to minimize risks and drop the faults at all of steps under development(Palanichamy, 2010). We have two types of FTA. The first is qualitative and the second is quantitative. Qualitative gives us to change description failures whereas quantitative gives us to change finding matematically background of probability occurences of failures. In quantitative steps we must determine whether events are dependable or undependable. And also we must find correct relations between each of events using gates. We have so many types of gates. These help finding detailed graph representations Figure 3.8 shows these gates explanation. FTA is top down failure modes and effect analysis where each potential failures's mode may result in an event or fault leading to the product failure. FTA software programs are already available. However, these are specific to use. Therefore if we use the tools we must deeply analyze it. In order to use the fault tree technique efficiently define scope of the analysis, familiarization with design, functions and operation systems, define top event, construction of the fault tree, analysis of the fault tree logic, report analysis, assessment of reliability improvements and trade offs(Malm et al., 2011; Dan and Shilong, 2011; Lutz and Mikulski, 2003).

Table 3.2. The level table according to IEC 62304 and ISO 14971 standard

| Level | Effects of Anamolous Behaviour | Explanation |
|-------|--------------------------------|--|
| A | No Effect | Failure conditions that can be overlooked or rarely seen |
| B | Minor | Failure situations that are not caused by serious injuries |
| C | Major | Failure conditions that cause fatal or serious injuries |

FTA is specifically has multitude of uses as a toll to determine the pertinent logic combination of event leading to the top event and potentially their prioritization and investigate a system under development and anticipate and prevent or mitigate potential cause of undesired top events, analyze a system, determine reliability, identify the major contributors to unreliability and evaluate the design changes, assist probabilistic risk assessment efforts.

We can combine FTA and binary decision diagram qualitative analysis. However it is so expensive and complex. Gates are most suitable for understanding the graphs. OR GATE infers one or more events cause the intermediate events or top event. AND gate infers events with together affect the intermediate events or top event. In my opinion AND gate more useful to avoid non-tolerated faults and do not know faults, however this affect directly complexity and determine requirements of each component independently. Single point of failure and common mode failure must be avoided by FTA at safety critical systems.

Probabilistic Risk Assessment (PRA) :PRA evaluates risks associated with a complex engineered technological entity such as an airliner or nuclear power plant.

PRA answers generally thee basic questions:

- What can go wrong with the studied technological entitiy? or What are the initiators? or What are initiating events that lead to adverse consequence?
- What and how severe are the potential detriments or adverse consequences that the technological entity may be eventually subjected to as a result of the occurence of the initiator?
- How likely to occur are these undesirable consequences or what are their probabilities or frequencies?

Failure Modes and Effects Analysis (FMEA) : FMEA is an inductive analytical methods which may be piece-part level or the functional. Functional FMEA is used to help of functional block diagram in system or equipment item. Piece-part level FMEA is used to determine failure modes for each piece-part components of systems.

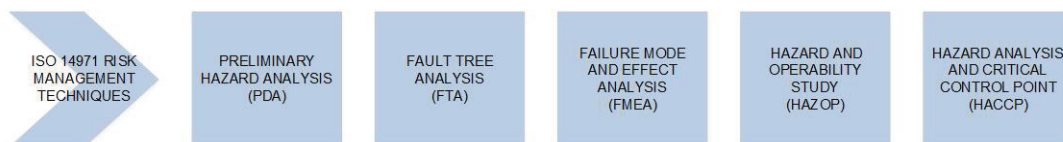


Figure 3.7. Tests To Be Carried Out Under Light of 14971

Failure Mode, Effects and Critical Analysis (FMECA) : FMECA includes FMEA and includes critical analysis. It is used to chart the probability of failure modes against

erating environment and determine failures and malfunctions. Hazard and risk analysis consists five types of classes and five types of frequencies.

Classes are:

1. Catastrophic: multiple deaths,
2. Critical: a death or severe injuries,
3. Marginal: a severe injury,
4. Insignificant: a minor injury,
5. No effects.

Frequency Categories (events/year) are:

1. Probable 0,01,
2. Occasional 0,001,
3. Remote 0,0001,
4. Improbable 0,00001,
5. Incredible 0,000001.

Cause and Effect Diagram: It is used for product design and quality defect prevention and identifying potential factors.

3.3. Requirement Based Testing

A requirement defines what a product is good at and under what conditions it needs to be done. The process of determining, defining, reporting, analyzing, validating and verifying requirements is called requirement engineering. Creating a solution space by understanding the problem space is a very important process for safety critical systems. Requirement-based tests allow verification of how well the system is built.

3.4. Verification and Validation of Safety Critical Software System

The verification and validation system must be applied at every level throughout the life cycle to determine the flaws of the device system, control the suitability of the device for production purposes, and determine the value of the device in the market.

Verification checks whether the product is developed correctly; Validation checks whether the correct product is developed according to user requests.

3.4.1. Model-Based Validation

The model-based validation system is to model a certain part or the whole of the system by abstracting it and to check whether the product produced is the right product for the customer.

CHAPTER 4

MODEL BASED REQUIREMENT VERIFICATION

Model Based Verification (MBV), which is a formal technique to test from abstract models, provides abstraction modelling to a part of system or complete system. Verification is essentially done with comprehensive state-space searching. For instance, a non-formal expression of “The temperature of the reactor had better not exceed about 300°C” is one of the most important features of nuclear reactor’s security system. Verification of this property of nuclear reactor system can be possible with examination of all possible situations and providing for each situation demonstration by the property.

Because MBV is based on comprehensive search, the encountered biggest difficulty is the situation space explosion in very large and complex systems. In this case; instead of comprehensive verification, test data can be created by using counter example produced with MBV.

Because of the fact that this issue is outside our study scope, it will not be further detailed here. In the given following description, Kripke structure is often used to define MBV and clarify time based logic.

Definition (Kripke Structure): structure is 4-tuple defined by $K = (S, S_0, T, L)$. In this definition,

- S is a finite set of state,
- $S_0 \subseteq S$ is a set of initial state,
- $T \subseteq S \times S$ is a transition relation,
- $L: S \Rightarrow 2^{AP}$ corresponding to each state and related atomic propositions that are valid in defining a case the labelling function.

MBV’s strengths are as follows(Baier and Katoen, 2008):

- It is useful in a wide range such as software engineering, embedded systems or hardware design.

- It supports partial verification. As a result of this the system properties can be tested independently of each other.
- It provides information about diagnostic sample and it is quite helpful to debug.

However, the weaknesses of the MBV are as follows:

- More suitable for control-weight applications and is not suitable for data-weighted applications.
- It is only valid for the model of the system, not a real product or prototype system, as a result of this, it is not as valuable as verification of the system itself.

Since NeuRoboScope is a real-time system, the use of timed automata has been deemed appropriate to model. The next section describes timed automata.

4.1. Timed Automata

The timed automaton is especially useful in real-time systems expression and design is a widely used tool. X , is real a finite set of time-valued variables, V show a finite set of values with variable real data changes. The condition \mathcal{K} is defined by:

$$\mathcal{K} := z \odot k \quad | \quad z - y \odot k,$$

such that $z, y \in X$ or $V, k \in \mathbb{N}$ and $\odot \in \{\leq, <, =, >, \geq\}$.

Definition (Timed Automata). A is a tuple $\mathcal{A} = (Q, q_0, X, \Sigma, \delta, I)$ where,

- Q is a finite set of locations(or nodes),
- $q_0 \in Q$ is the initial location,
- X finite set of clocks ,
- Σ is a finite alphabet of actions,
- $\delta \subseteq Q \times 2^{\mathcal{K}} \times \Sigma \times 2^X \times Q$ gives the set of transitions,
- $I : Q \rightarrow 2^{\mathcal{K}}$ assigns invariants to locations.

As an above definition, $\mathcal{A} = (\{A, B\}, A, \{x\}, \emptyset, \{(A, -, x > 3, x = 0, B), (B, -, -, A)\}, \{B : x \leq 5\})$ is a sample automata model shown in Figure 4.1.

In this automata with two transitions between the two positions x , is defined as a clock variable. Initial position A can wait to 5 units of time. Pass from A to B can be done as long as the clock is greater than 3.

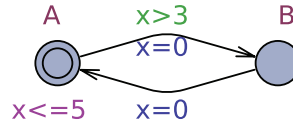


Figure 4.1. Sample Example

In this case, clock can change from A to B while the clock is between 3 and 5 units of time. In the B position, any variable is defined and any condition is defined from B to A at any time. In this case automata can pass from B to A for every time.

4.2. Timed Automata Specification Language

Computational Tree Logic (CTL) is used for model checkers to formal verification which provides checking safety or liveness property of system with tree-like structure of modelling of time Timed Computational Tree Logic (TCTL) quantitative extension of CTL by temporal operators with time intervals which provides verification of safety or liveness or time-bounded liveness system properties.

Definition (Syntax of TCTL): A TCTL formula φ is defined by below grammer:

$$\varphi ::= true \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \exists \mathcal{U}_I \varphi \mid \varphi \forall \mathcal{U}_I \varphi.$$

Where $p \in Pr$ is a atomic condition and clock variables, $I \in \mathcal{I}$ as for φ , an element of set of Interval.

As an above syntax, commonly used in literaturely and given following text(Gabbay et al., 1994):

$\exists \diamond \psi$ (Exists Eventually): In some execution path ψ eventually holds.

$\forall\Diamond\psi$ (Always Eventually): In all execution path ψ eventually hold.

$\exists\Box\psi$ (Exists Globally): ψ held through at least one of path.

$\forall\Box\psi$ (Always Globally): During all execution path ψ holds all path.

$\psi \rightsquigarrow \varphi$ (Always Leads To): In this syntax if ψ provided, φ always holds .

$\psi \rightsquigarrow_{\leq t} \varphi$ (Time Bounded Leads-to): After property of ψ hold, property of φ is to hold within t .

As mentioned above, some properties are about safety critical properties and the others liveness properties. In general concept, one safety critical system has three properties: *Safety Critical Properties* has these properties: “Any bad situation never happen”. For instance, as will be explained later, it is absolutely unwanted behaviour to start centering in surgical region for endoscopic surgery. These type of properties determine reliability of system.

Liveness Properties has these properties: “Something happens eventually”. For instance, surgeon is expected to press control unit active button for movements.

Time Bounded Liveness Properties. In real-time systems, the just only liveness properties are not sufficient and meaningful. Some properties are only meaningful within time limit. Time bounded liveness properties is shown with time bounded operator $\varphi \rightsquigarrow_{\leq t} \psi$.

4.3. Model Checker Tool Uppaal

UPPAAL, which is developed by Uppsala and Aalborg universities, used all around the world from academic projects to business projects. It provides facility to test user defined properties, channel synchronization, structure data types and variables(Behrmann et al., 2004).

4.4. NeuRoboScope System of Timed Automata

In this section NeuRoboScope’s timed automata is presented. Model Checker tool UPPAAL supports communication mechanism from timed automata networks(Uzun et al., 2016).

4.4.1. Preliminary Information

In this subsection, before describing Uppaal Models, brief information about the channels and variables used in system models mentioned as below.

4.4.1.1. Channels

The channels of models as listed below:

- **eyb**: When CEK key is half-pressed, *eyb* channel is synchronized,
- **etb**: When CEK key is fully pressed, *etb* channel is synchronized,
- **eyk**: When CEK key is half released, *eyk* channel is synchronized,
- **etk**: When CEK key is fully released, *etk* channel is synchronized,
- **asb**: When pressing the surgery zone button, *asb* channel is synchronized,
- **kbe**: When main control unit is activated, *kbe* channel is synchronized,

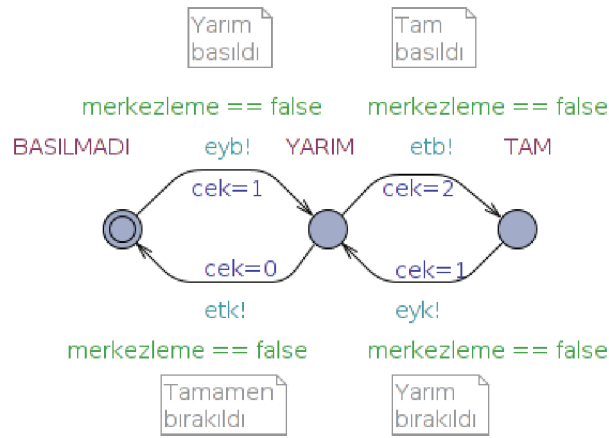


Figure 4.2. Surgeon Hold Endoscope Model

- **kbk**: When main control unit is deactivated, *kbk* channel is synchronized,
- **mkz**: When starting homming, *mkz* channel is synchronized,
- **newpos**: When endoscope is moved new position *newpos* is synchronized,
- **hazir**: When homming finish, *hazir* channel is synchronized,
- **pildolu**: When battery status is full, *pildolu* channel is synchronized,

- **pilaz:** When battery status is low, *pilaz* channel is synchronized,
- **pilbitti:** When battery status is empty, *pilbitti* channel is synchronized,

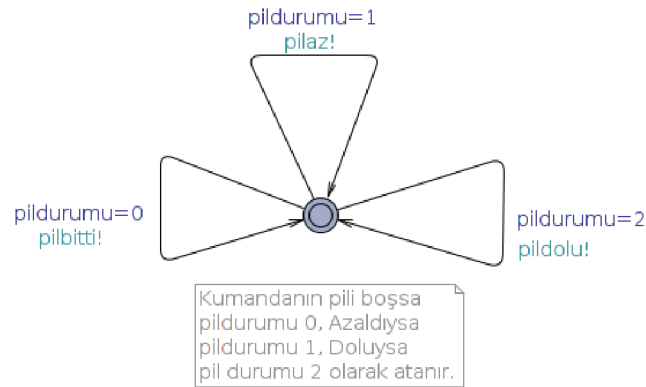


Figure 4.3. Battery Status Model

- **mbk:** When manuel control button is activated, *mbk* channel is synchronized,
- **mkk:** When manuel control button is deactivated, *mkk* channel is synchronized.

4.4.1.2. Variables

The variables of models as listed below:

- ***ameliyat_bolgesi_siniri:*** The surgeon determines the region of operation where the endoscope can move and the region where the centering can be done.
- ***pildurumu:*** Information on whether the battery status is full or else. It is created as 2 when the battery is full, it is 1 when the battery is low, and it is 0 when the battery is empty.
- ***cek:*** CEK key information is kept in the variable *cek*. It is 0 when the CEK key is released, it is 1 half-pressed and 2 is fully-pressed.
- ***enc:*** The encode information stored in the variable *enc*. The point at which the variable is farthest from the patient is determined as 0, and the endpoint of the operative region as 100.

- **pasif_kol_serbest:** Information is kept that the passive balanced arm is released or braked. If the passive balanced arm is released, it is set to true else if the passive balanced arm is not released.
- **aktif_kol_merkezli:** Knowledge of being active endoscope holder centered or not centered. If it is true, the active endoscope holder is centered. Else if it is not centered.

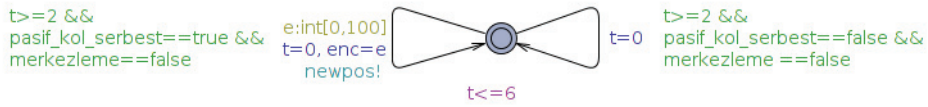


Figure 4.4. Surgeon Endoscope Movement Model

- **manuelkontrolbutonu:** Information is kept that whether the manuel control button is pressed or released. When it is pressed, variable is 1. Otherwise, it will be 0.
- **kumandabirimietkin:** Information is kept that whether the main control unit is active or inactive. When it is active, variable is 1. Otherwise, it will be 0.
- **merkezleme:** Information is kept that whether homming is taking. When homming is taking, variable is true. Otherwise, it will be false.
- **endoskop_iceride:** Information is kept that whether the endoscope is in the surgeon region or not. When it is inside, variable is true. Otherwise, it will be false.

4.4.1.3. Surgeon Endoscope Holds Model

Initially, the CEK key is started to BASILMADI state (Eng: RELEASED) and cek is equal to 0. If the surgeon presses the CEK key halfway, the eyb channel synchronized and state switches to YARIM (Eng: HALF) , and updates cek to 1. In the case of YARIM state, the surgeon can release the CEK key completely, the etk channel synchronized and state switches to BASILMADI state and updates cek to 0; or the surgeon can press the

key more firmly while holding the CEK switch in YARIM state, then the state of TAM (Eng: PRESSED) is passed to the etb channel synchronise, updated cek to 2. The surgeon

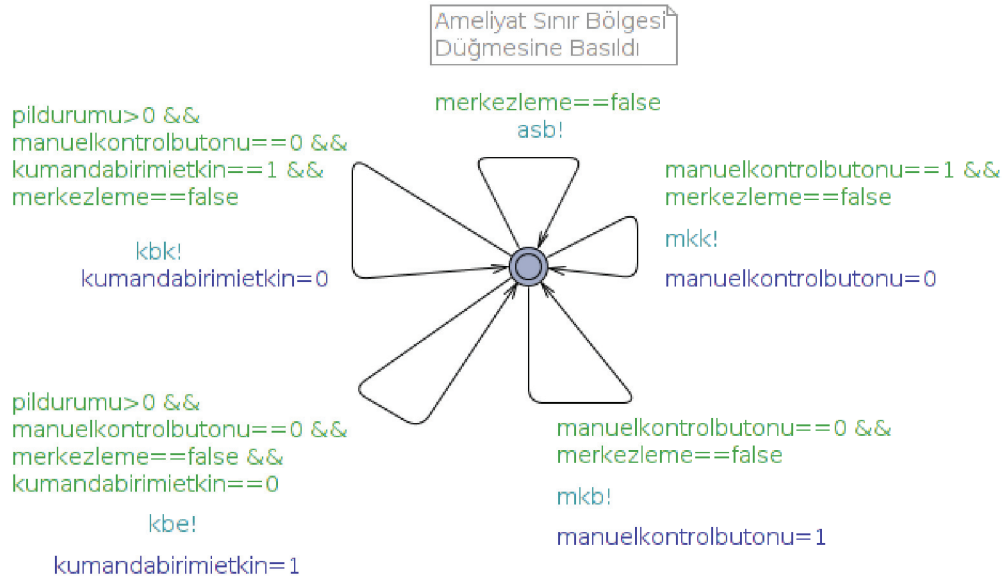


Figure 4.5. Surgeon Control Model

can release the CEK key halfway, in this case the YARIM state is passed, the eyk channel is synchronized and the cek is updated to 1. It is assumed that the CEK switch does not change state when homming is performed. Surgeon endoscope holds model is shown below Figure 4.2.

4.4.1.4. Surgeon Control Model

The surgeon can control the system in 5 different ways. Surgeon endoscope control model is shown below Figure 4.5.

- **Main Control Unit is active (KBE):** Activates the main control unit when the surgeon pushes the activation button of the main control unit when main control unit's battery is low or full. The manual control button is not pressed and no centering is required.
- **Main Control Unit is inactive (KBK):** The surgeon pushes the activation button of the main control unit to remove the activation of the main control unit when the

main control unit's battery is low or full. The manual control button is not pressed and no centering is required.

- **Surgery Zone Button is pressed(ASB):** The surgeon sets the surgery zone by pressing the surgery zone button. It is assumed that the center of surgery is not pressed when the center is operated.

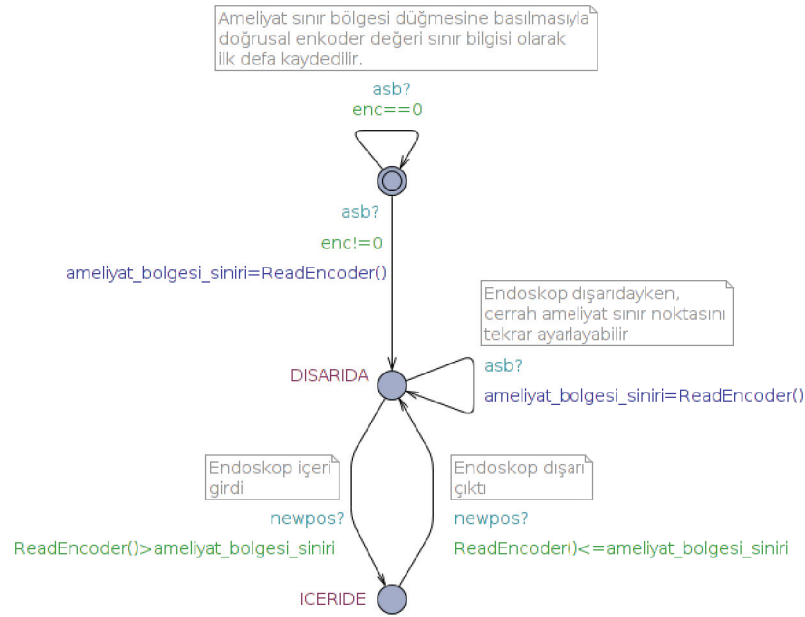


Figure 4.6. Endoscope Region Model

- **Manual Control Button is pressed(MKB):** The surgeon can use the system manually by pressing the manual control button. It is assumed that the manual control button is not pressed while centering is being performed.
- **Manual Control Button is released(MKK):** The surgeon exits manual mode by releasing the manual control button.

4.4.1.5. Surgeon Endoscope Movement Model

If the surgeon pressed the CEK key halfly, the passive balanced arm is released and the surgeon can take the endoscope to the new position within 2-6 seconds. Linear

encoder information is set as 0-100 interval. If the passive balanced arm is not released, it returns to its initial state to wait for it to be released. Surgeon endoscope movement model is shown below Figure 4.4.

4.4.1.6. Passive Balanced Arm Model

There are two type of status exists for passive balanced arm. They are mentioned below list. Passive balanced arm model is shown below Figure 4.9.

- ***Passive Balanced Arm is braked:*** The passive balanced arm is FRENLI (Eng: BRAKED) when system starts. If the CEK key is pressed halfway, eyb? channel synchronized and the time is reset (t=0). After 1 second, etk? or eyk? channel does not synchronized, pasif_kol_serbest becomes true and the passive balanced arm goes to the SERBEST (Eng: RELEASED) status. If the CEK key is pressed, the passive balanced arm remains FRENLI status. If the CEK switch is released halfly within 1 second of the, the passively balanced arm remains FRENLI status. When manual control button is pressed, mkb? synchronized and the passive balanced arm goes to the SERBEST status and the pasif_kol_serbest is updated to true.
- ***Passive Balanced Arm is released:*** When the CEK key is half-pressed or the manual control button is pressed, the passive balanced arm is in the SERBEST status. If the passive balanced arm is in the SERBEST status and the CEK key is not held halfway, passive balanced arm status turns to FRENLI when the mkk? channel is synchronized and the pasif_kol_serbest is updated to false. If the passive balanced arm is in the SERBEST status and the CEK key is pressed fully, the etb? channel is synchronized to FRENLI status and the pasif_kol_serbest is updated to false. If the pasif_kol_serbest is in the SERBEST status and the CEK key is halfly released, the eyk? channel is synchronized to FRENLI status and the pasif_kol_serbest is updated to false.

4.4.1.7. Active Endoscope Holder Model

The system starts with ETKINDEGIL(Eng: INACTIVE) status. And firstly the

system will center in active mode for 4-6 seconds. During the centering, the mkz? channel is synchronized, the system time is reset ($t = 0$), the aktif_kol_merkezli variable is updated to true and the green light is blinking during this time to indicate that the centering has been done. Once the centering is completed, the hazır channel is synchronized and the

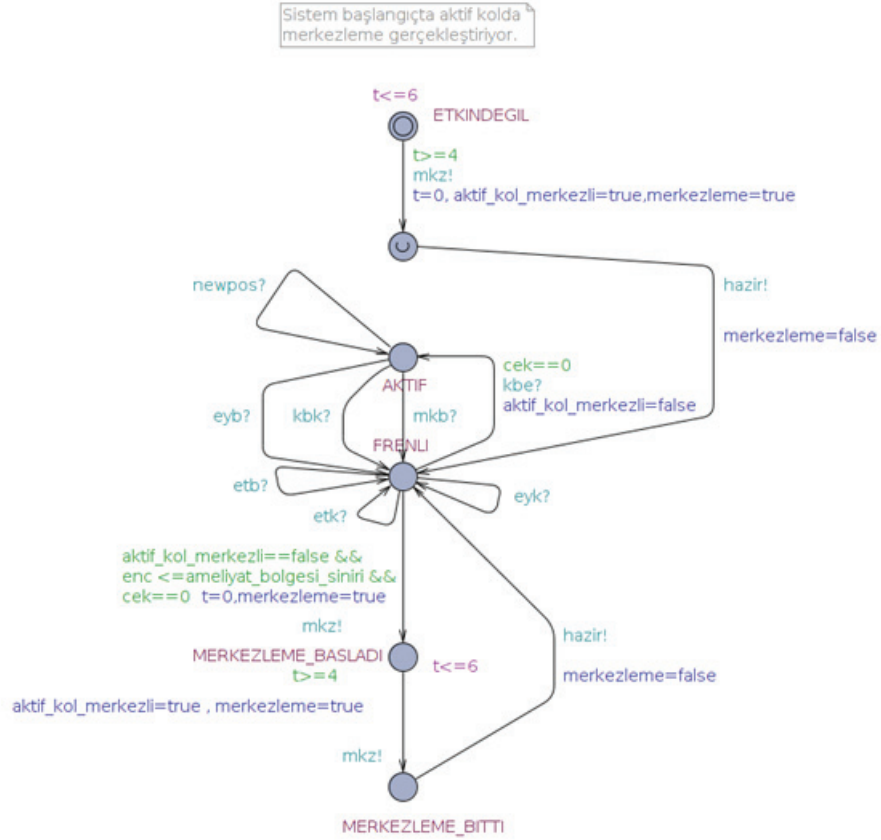


Figure 4.7. Active Endoscope Holder Model

green light is turn to YANIYOR status and active endoscope holder is in FRENLI (Eng: BRAKED) status.

When the active endoscope holder is in the AKTIF (Eng: ACTIVE) status, it will switches to FRENLI (Eng: BRAKED) status with the cek button half-pressed. When the active endoscope holder is in the AKTIF status, it switches to the FRENLI status with the main control unit button pressing.

If the active endoscope holder is FRENLI and the encoder is equal or smaller than ameliyat_bolgesi_siniri, the mkz! channel is synchronized (green light is starting to blink

to indicate that centering is taking) and time updated to 0 ($t = 0$) and merkezleme turns to true. When the system is in 4-6 seconds, centering is performed and the aktif_kol_merkezli variable is updated to true.

By synchronizing the mkz! channel, the green light is continuously blinked. When the centering of the active endoscope holder is finished, hazır! channel is synchronized and the green light turns to YANIYOR status and the active endoscope holder passes to the FRENLI status. Active endoscope holder model is shown below Figure 4.7.

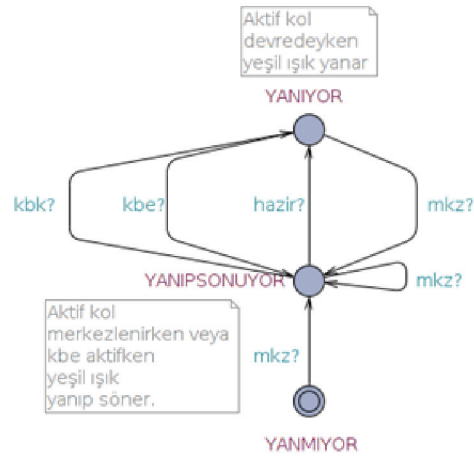


Figure 4.8. Green Light Model

4.4.1.8. Endoscope Region Model

By pressing the surgery zone button, linear encoder value is kept as boundary information. If encoder information is set to 0, the initial state is waited until a new value is given. If the encoder information is different from zero, the surgical region boundary is defined as encoder information and the endoscope is brought to DISARIDA (Eng: OUTSIDE) status. While the endoscope is out, the surgeon can re-adjust the surgical margin or enter the nose to ICERIDE (Eng: INSIDE) status. In case of ICERIDE, only endoscopic encoder information must be larger than ameliyat_bolgesi_siniri. By keeping information in and out of this way, the center of the endoscope is made only outside without harming the patient. Endoscope region model is shown below Figure 4.6.

4.4.1.9. Green Light Model

Initially green light starts in YANMIYOR (Eng: OFF) status. After homming, the green light goes to YANIPSONUYOR (Eng: BLINKING) status until centering starts and ends.

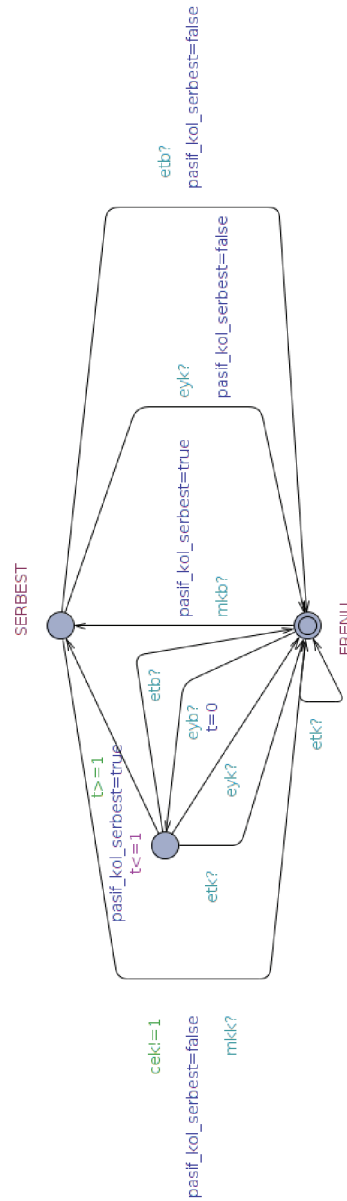


Figure 4.9. Passive Balanced Arm Model

Once homming is complete, the green light goes YANIYOR (Eng: ON) to indicate that the active endoscope holder is active. While the main control unit button is pressed,

the green light switches to YANIPSONUYOR status. Green light model is shown below Figure 4.8.

4.4.1.10. Red Light Model

When the system starts to work, it starts in the YANIP_SONUUYOR (Eng: BLINKING) status because the asb has not been created.

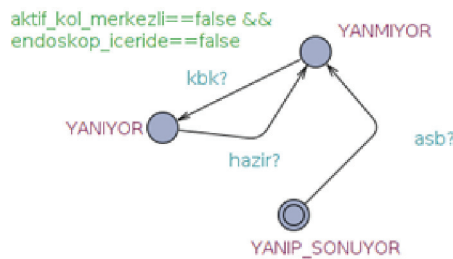


Figure 4.10. Red Light Model

When asb is created, it goes into YANMIYOR(Eng: OFF) status.

It goes YANIYOR (Eng: ON) with kbk synchronisation. it goes to YANMIYOR status with hazir synchronisation. Red light model is shown below Figure 4.10.

4.4.1.11. Battery Status Model

Battery status model is shown in Figure 4.3. The battery status is set to 0 when runs out of battery. In this case the main control unit can not be activated. The battery status is set to 1 when battery life is reduced. When the battery is fully charged, the battery status is set to 2. The main control unit can be activated when the battery is low or full.

4.5. Verification of System And Evaluation

A number of queries have been written to test the safety criticality and liveness properties of the system and a few of the most important ones will be presented. First

query is specific for UPPAAL to check whether system includes deadlock:

$\forall \square \text{ not } \text{deadlock}$

Another important safety critical question is that the endoscope should not start active endoscope holder centering in any situation while in the surgical region:

$\forall \square \text{ not } (\text{enc} > \text{ameliyat_bolgesi_siniri} \text{ and}$

$\text{aKtifEndoskopYonlendirici.MERKEZLEME_BASLADI})$

While the active endoscope holder is active, main control unit must be active:

$\forall \square \text{ not } (\text{aKtifEndoskopYonlendirici.AKTIF} \text{ and } \text{kumandabirimietkin}==\text{false})$

NeuRoboScope's model succeeded successfully from the above questions. Liveness properties of the system should also be tested. For example, there must be a situation where the active endoscope holder is braked:

$\exists \diamond \text{ aKtifEndoskopYonlendirici.FRENLI}$

There is at least one situation where the endoscope's surgical region is inside:

$\exists \diamond \text{ EndoskopBolgesi.ICERIDE}$

Whether or not the active endoscope holder can be centered can be tested with the following question:

$\exists \diamond \text{ aKtifEndoskopYonlendirici.MERKEZLEME_BITTI}$

The queries given in the examples above are formally expressed in the TCTL definition language. The UPPAAL verify receives as an input model to other input queries and tells whether the system verifies the desired properties.

Failure to verifying any properties can mean two things:

- There is an error in the system requirements, and
- There is an error in the model. Therefore, in such a case, going back and looking at both possibilities should be done and continued the process until get in an error-free system / model.

The model-based verification checks whether a system's requirements behave as required.

CHAPTER 5

MODEL BASED MUTATION TESTING

Manufacturers, developers or producers always tend to make mistake. When creating safety critical system, find out where the mistake is made during production is crucial. The model based mutation testing provide what steps we can take to prevent possible fault. Testers expect that test cases are to be enough determine the mutant. The number of mutants killed indicates that the requirements are sufficiently clear. Testers always hope their test technique goes from killing more mutants with less testing.

The NeuRoboScope system tested using mutual analysis as the counter-examples generated by the model checker as the test input (Fraser et al., 2007; Aichernig et al., 2013).

5.1. Verification With Mutation Testing

To verify NeuRoboScope system model, we inject one fault to XML file of generated by UPPAAL. There are two type of mutation methods used as metioned belowed Subsection 5.1.1 and 5.1.3.

5.1.1. Model Mutation NeuRoboScope UPPAAL Model

Model mutation was done relevant with following model mutation operator:

1. **Change action** (μ_{ca}) generates from A a set of $|\Delta_I| (|\Sigma_O|) + |\Delta_O| (|\Sigma_O| - 1)$ mutants, where every mutant changes a single transition in A by replacing the action labeling the transition by a different output label. A TAIIO $M \in \mu_{ca}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta, \delta m = (q, a_m, g, \rho, q'), a_m \in \Sigma_O$ and $a_m \neq a$.

Example of change action has been shown Figure 5.4 which is orginal model is shown Figure 4.7. The places marked with yellow indicate changes.

2. **Change target**(μ_{ct}) generates from A a set of $|\Delta| (|Q| - 1)$ mutants, where every mutant replaces the target location of a transition in A , by another location in A . A TAIO $M \in \mu_{ct}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta$, $\delta m = (q, a, g, \rho, q'_m)$, $q'_m \in Q$ and $q'_m = q'$.

Example of change target has been shown Figure 5.1 which is original model is shown Figure 4.7. The places marked with yellow indicate changes.

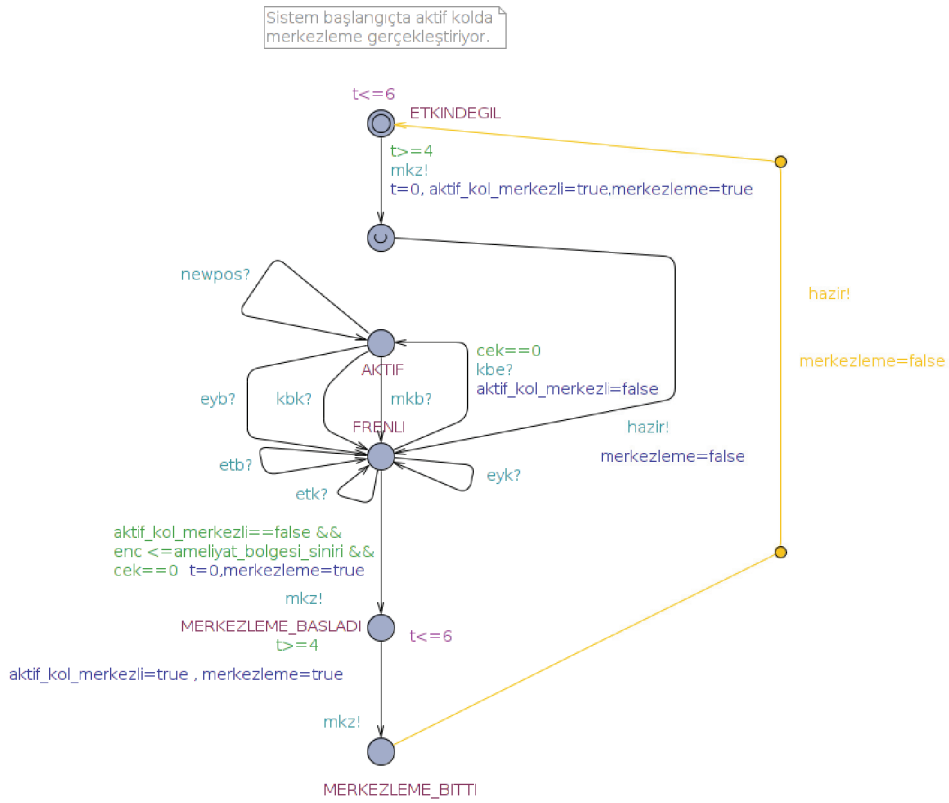


Figure 5.1. Change Target Mutant Example

3. **Change source**(μ_{cs}) generates from A a set of $|\Delta| (|Q| - 1)$ mutants, where every mutant replaces the source location of a transition in A , by another location in A . This expresses an implementation fault where a signal can be triggered from a state where it should be disabled. A TAIO $M \in \mu_{cs}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta$, $\delta m = (q_m, a, g, \rho, q')$, $q_m \in Q$ and $q_m = q$.

Example of change source has been shown Figure 5.3 which is original model is shown Figure 4.7. The places marked with yellow indicate changes.

4. **Self loop** (μ_{sl}) generates from A a set of $|\Delta|$ mutants, where every mutant replaces the target location of a transition in A , by its source location. The created mutants are a subset of both the **Change target** and the **Change source** mutants, where the target and source location are equal. A TAIIO $M \in \mu_{sl}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta$, $\delta m = (q, a, g, \rho, q)$;

Example of self loop has been shown Figure 5.2 which is original model is shown Figure 4.7. The places marked with yellow indicate changes.

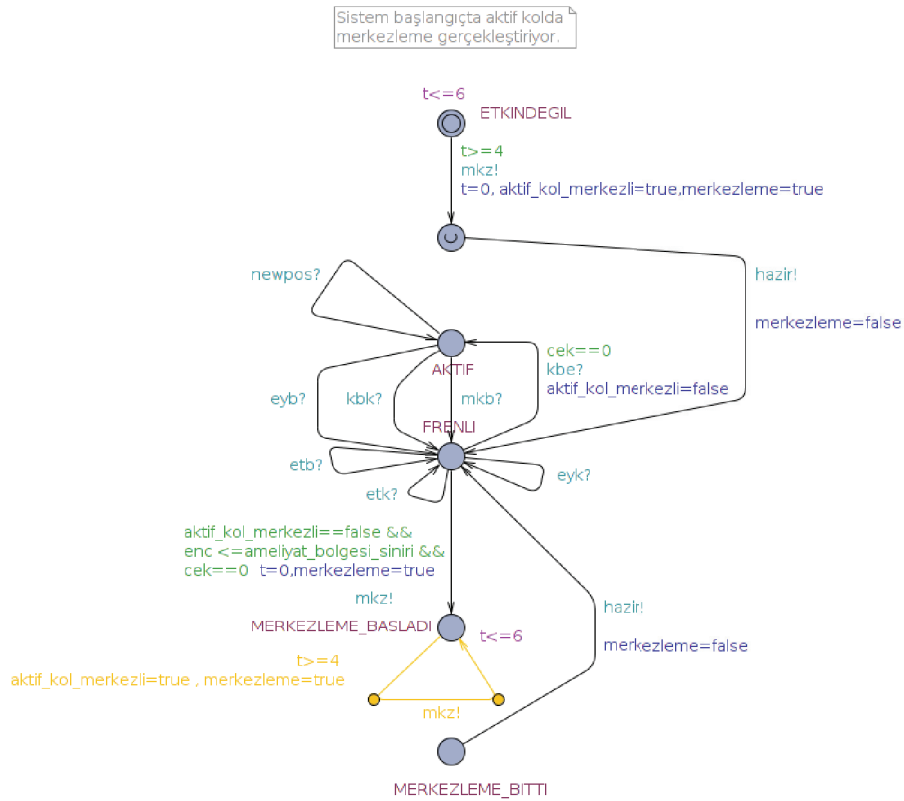


Figure 5.2. Change Selfloop Mutant Example

5. **Change guard** (μ_{cg}) generates from A a set of $4|\mathcal{G}|$ mutants, where every mutant replaces a transition in A with another one which changes the original guard by altering every equality/inequality sign appearing in the guard by another one. This

covers implementation faults with faulty enabling conditions. A TAIIO $M \in \mu_{cg}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta^m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta$, $\delta^m = (q, a, g_m, \rho, q')$, $g = \bigwedge_{i \in I} c_i o_i d_i$, $g_m = \bigwedge_{i \in I} c_i o_i^m d_i$, $o, o_i^m \in \{<, \leq, =, \geq, >\}$, $o_i \neq o_i^m$ for some $i \in I$ and $o_j = o_j^m$ for all $j \neq i$.

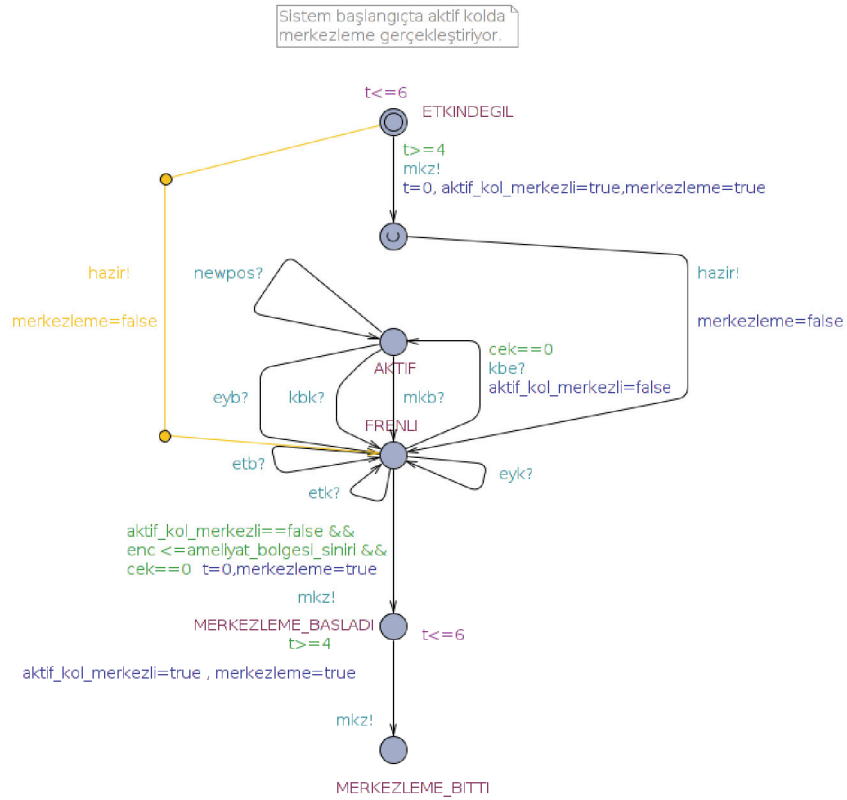


Figure 5.3. Change Source Mutant Example

Example of change guard has been shown Figure 5.5 which is original model is shown Figure 4.7. The places marked with yellow indicate changes.

6. **Change invariant** (μ_{ci}) generates from A a set of $|I|$ mutants, where every mutant replaces the invariant of a location with another invariant with multiplying 2 to the right side of the invariant. This mimics an "off by one"-fault allowing to stay longer in a state than intended. A TAIIO $M \in \mu_{ci}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I_m, \Delta)$ and there exists $q \in Q$, such that $I(q) = \bigwedge_{i \in I} c_i o_i d_i$, $o \in \{<, \leq\}$, $I_m(q) = \bigwedge_{i \in I} c_i o_i^m d_i^m$, $d_i^m = d_i * 2$ for some $i \in I$, $d_j^m = d_j$ for all

$j \neq i$ and $I(q') = I_m(q')$ for all $q' \neq q$.

Example of change invariant has been shown Figure 5.6 which is original model is shown Figure 4.7. The places marked with yellow indicate changes.

7. **Invert Reset**(μ_{ir}) generates from A a set of $|\Delta| |C|$ mutants, where every mutant replaces a transition in A , by another transition with the occurrence of one clock flipped compared to the original set of clocks. This reflects different timing errors, e.g. the incorrect resetting of a timer. A TAIIO $M \in \mu_{ir}(A)$, if M is of the form $(Q, \hat{q}, \Sigma_I, \Sigma_O, C, I, (\Delta \setminus \{\delta\}) \cup \{\delta m\})$, such that $\delta = (q, a, g, \rho, q') \in \Delta$, $\delta m = (q, a, g, \rho_m, q')$, and for some $c \in C$ either $\rho_m = \rho \cup \{c_m\}$ if $c_m \notin \rho$, or $\rho_m = \rho \setminus \{c_m\}$ if $c_m \in \rho$.

5.1.2. Model Mutation Result

According to model mutant types and counts, the Table 5.1 is created after generating mutants from UPPAAL XML file.

Table 5.1. Model Mutations Results

| MUTANT TYPE | MUTANT COUNT |
|--------------------|--------------|
| Change Action | 593 |
| Change Guard | 35 |
| Change Invariant | 4 |
| Change Invertreset | 5 |
| Change Selfloop | 66 |
| Change Source | 88 |
| Change Target | 88 |
| Total | 879 |

5.1.3. Specification Mutation NeuRoboScope UPPAAL Model

Specification mutation was done relevant with following operators:

1. **AssociativeShiftOperator(ASO)**: Change the association between variables

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==false) and aKtifEndoskopYonlendirici.AKTIF

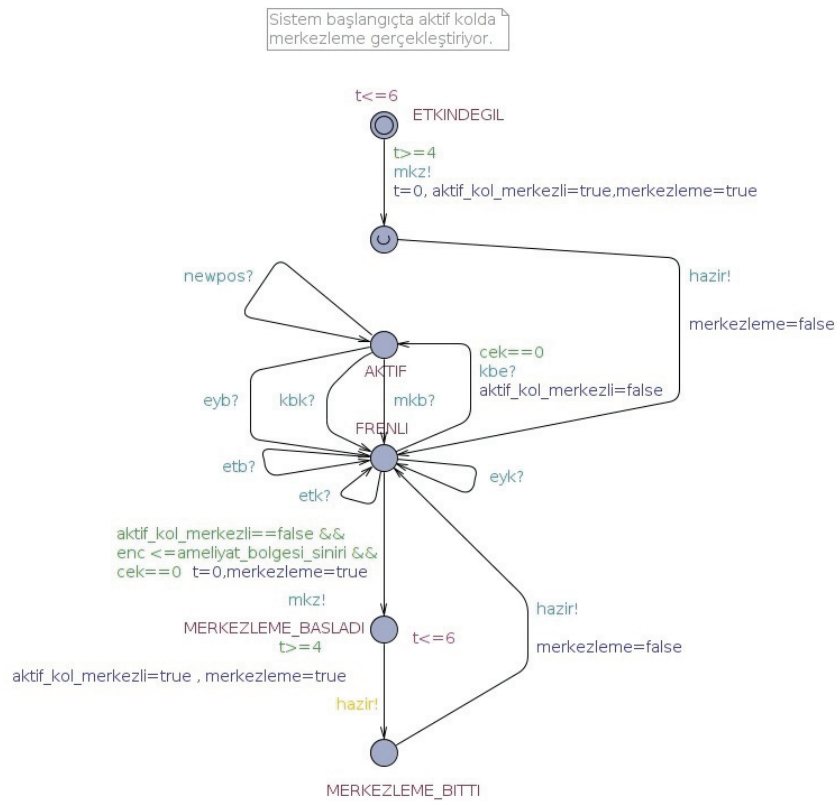


Figure 5.4. Change Action Mutant Example

2. **ConstantReplacementOperator(CRO):** Replace a constant with a syntactically valid different constant

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==**false** and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (PasifEngelenmisKol.FRENLI and aktif_kol_merkezli==true and aKtifEndoskopYonlendirici.AKTIF)

3. **ExpressionNegationOperator(ENO):** Negate sub-expressions.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

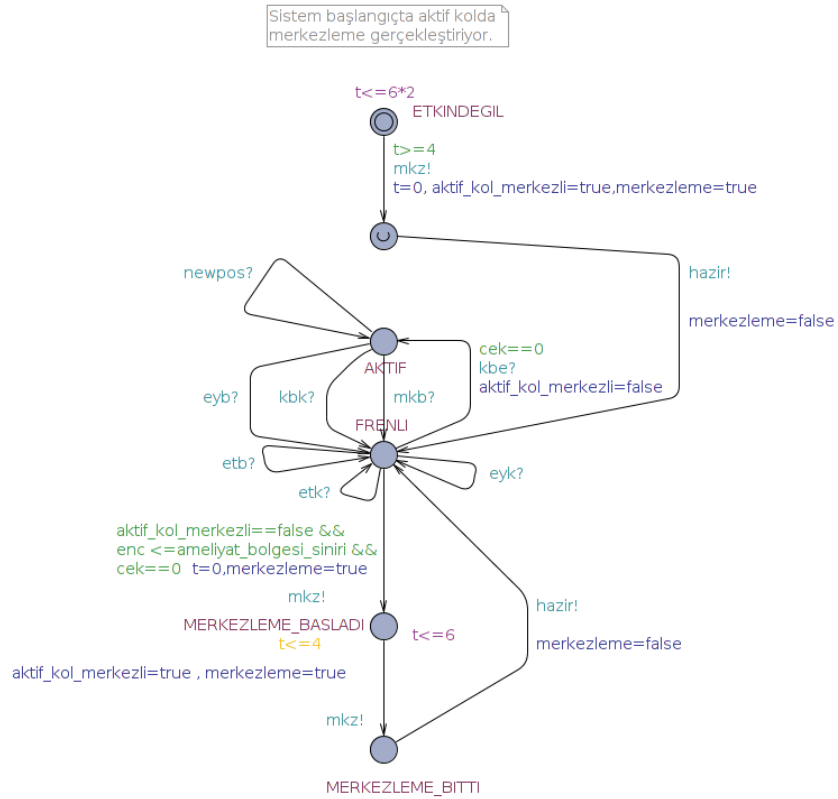


Figure 5.5. Change Guard Mutant Example

$\forall \square$ **not** (PasifEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

4. **LogicalOperatorReplacement(LRO):** Replace a logical operator with another logical operator.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query :

$\exists \diamond$ (PasifEngelenmisKol.FRENLI or aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

5. **MissingConditionOperator(MCO):** Remove a single condition from a decision.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

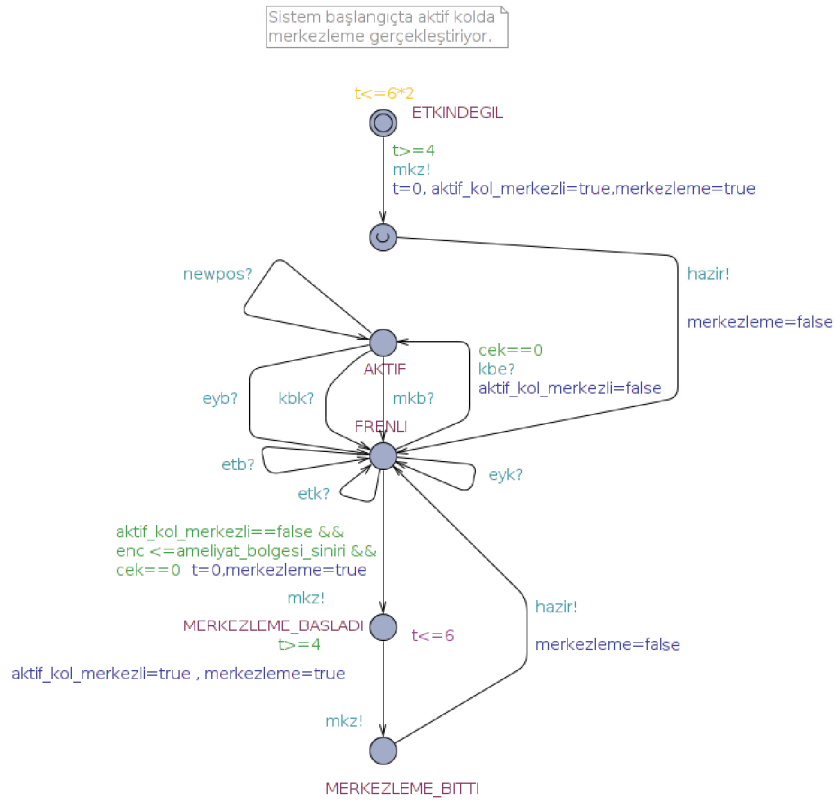


Figure 5.6. Change Invariant Mutant Example

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

6. **RelationalOperatorReplacement(RRO)**: Replace a relational operator with another relational operator

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli!=false and aKtifEndoskopYonlendirici.AKTIF)

7. **SimpleExpressionNegation(SNO)**: Negate an atomic condition in a decision.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (**PasifdEngelenmisKol.FRENLI** and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (**not PasifdEngelenmisKol.FRENLI** and aktif_kol_merkezli==false and aKtifEndoskopYonlendirici.AKTIF)

8. **StuckAtOperator(STO)**: Replace a condition with true or false.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and **aktif_kol_merkezli==false** and aKtifEndoskopYonlendirici.AKTIF)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and **1** and aKtifEndoskopYonlendirici.AKTIF)

9. **VariableReplacementOperator(VRO)**: Replace a variable reference with a reference to another variable of the same type.

Example of NeuRoboScope Model Real Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==false and **aKtifEndoskopYonlendirici.AKTIF**)

Example of NeuRoboScope Model Mutant Injected Query:

$\exists \diamond$ (PasifdEngelenmisKol.FRENLI and aktif_kol_merkezli==false and **EndoskopBolgesi.ICERIDE**)

5.1.4. Specification Mutation Result

According to specification mutant types and counts, the Table 5.2 is created after generating mutants from UPPAAL XML file.

Table 5.2. Specifications Mutations Results

| MUTANT TYPE | MUTANT COUNT |
|-------------|--------------|
| ASO | 9 |
| CRO | 13 |
| ENO | 116 |
| LRO | 18 |
| MCO | 31 |
| RRO | 17 |
| SNO | 22 |
| STO | 46 |
| VRO | 513 |
| Total | 785 |

5.2. Examination Uppaal Outputs

There is possible to create shortest trace, fastest trace or some trace using UPPAAL tool. In this study, we chose fastest way to trace. Therefore we used following terminal code:

```
./verifyta -u -y -t2 $uppalxmlfilename.xml >  
$myfilename.doc 2 > &1
```

Uppaal outputs starts with a STATE and ends with a STATE. A transition links between STATES. If we visualize one trace, we can draw Figure 5.7.

An example UPPAAL output of a State:

```
(CerrahEndoskopKavrama.BASILMADI  
CerrahKontrol..id3  
CerrahEndoskopSurme..id4  
PasiFdEngelenmisKol.SERBEST
```

aKtifEndoskopYonlendirici.ETKINDEGIL

EndoskopBolgesi..id7

KumandaPil..id23

KirmiziIsik.YANIP_SONUUYOR

YesilIsik.YANMIYOR)

$rt \leq 6, rt - CerrahEndoskopSurme.t \leq 0,$

$rt - Pasi fdEngelenmisKol.t \leq 0,$

$rt - aKtifEndoskopYonlendirici.t \leq 0,$

$rt - EndoskopBolgesi.t \leq 0,$

$rt - KumandaPil.t \leq 0,$

$rt - YesilIsik.ty \leq 0,$

$rt - \#tau \leq 0, CerrahEndoskopSurme.t \leq 6,$

$CerrahEndoskopSurme.t - rt \leq 0,$

$CerrahEndoskopSurme.t - Pasi fdEngelenmisKol.t \leq 0,$

$CerrahEndoskopSurme.t - aKtifEndoskopYonlendirici.t \leq 0,$

$CerrahEndoskopSurme.t - EndoskopBolgesi.t \leq 0,$

$CerrahEndoskopSurme.t - KumandaPil.t \leq 0,$

$CerrahEndoskopSurme.t - YesilIsik.ty \leq 0,$

$CerrahEndoskopSurme.t - \#tau \leq 0,$

$Pasi fdEngelenmisKol.t \leq 6,$

$Pasi fdEngelenmisKol.t - rt \leq 0,$

$Pasi fdEngelenmisKol.t - CerrahEndoskopSurme.t \leq 0,$

$Pasi fdEngelenmisKol.t - aKtifEndoskopYonlendirici.t \leq 0,$

$Pasi fdEngelenmisKol.t - EndoskopBolgesi.t \leq 0,$

$Pasi fdEngelenmisKol.t - KumandaPil.t \leq 0,$

$Pasi fdEngelenmisKol.t - YesilIsik.ty \leq 0,$

$Pasi fdEngelenmisKol.t - \#tau \leq 0,$

$aKtifEndoskopYonlendirici.t \leq 6,$

$aKtifEndoskopYonlendirici.t - rt \leq 0,$

$aKtifEndoskopYonlendirici.t - CerrahEndoskopSurme.t \leq 0,$

$aKtifEndoskopYonlendirici.t - Pasi fdEngelenmisKol.t \leq 0,$

$aKtifEndoskopYonlendirici.t - EndoskopBolgesi.t \leq 0,$

$aKtifEndoskopYonlendirici.t - KumandaPil.t \leq 0,$

$aKtifEndoskopYonlendirici.t - YesilIsik.ty \leq 0,$
 $aKtifEndoskopYonlendirici.t - \#tau \leq 0,$
 $EndoskopBolgesi.t \leq 6,$
 $EndoskopBolgesi.t - rt \leq 0,$
 $EndoskopBolgesi.t - CerrahEndoskopSurme.t \leq 0,$
 $EndoskopBolgesi.t - PasifdEngelenmisKol.t \leq 0,$
 $EndoskopBolgesi.t - aKtifEndoskopYonlendirici.t \leq 0,$
 $EndoskopBolgesi.t - KumandaPil.t \leq 0,$
 $EndoskopBolgesi.t - YesilIsik.ty \leq 0,$
 $EndoskopBolgesi.t - \#tau \leq 0,$
 $KumandaPil.t \leq 6, KumandaPil.t - rt \leq 0,$
 $KumandaPil.t - CerrahEndoskopSurme.t \leq 0,$
 $KumandaPil.t - PasifdEngelenmisKol.t \leq 0,$
 $KumandaPil.t - aKtifEndoskopYonlendirici.t \leq 0,$
 $KumandaPil.t - EndoskopBolgesi.t \leq 0,$
 $KumandaPil.t - YesilIsik.ty \leq 0,$
 $KumandaPil.t - \#tau \leq 0,$
 $YesilIsik.ty \leq 6, YesilIsik.ty - rt \leq 0,$
 $YesilIsik.ty - CerrahEndoskopSurme.t \leq 0,$
 $YesilIsik.ty - PasifdEngelenmisKol.t \leq 0,$
 $YesilIsik.ty - aKtifEndoskopYonlendirici.t \leq 0,$
 $YesilIsik.ty - EndoskopBolgesi.t \leq 0,$
 $YesilIsik.ty - KumandaPil.t \leq 0,$
 $YesilIsik.ty - \#tau \leq 0,$
 $\#tau \leq 6, \#tau - rt \leq 0,$
 $\#tau - CerrahEndoskopSurme.t \leq 0,$
 $\#tau - PasifdEngelenmisKol.t \leq 0,$
 $\#tau - aKtifEndoskopYonlendirici.t \leq 0,$
 $\#tau - EndoskopBolgesi.t \leq 0,$
 $\#tau - KumandaPil.t \leq 0,$
 $\#tau - YesilIsik.ty \leq 0$ $ameliyat_bolgesi_siniri = 100$
 $pildurumu = 2$ $cek = 0$ $enc = 0$ $manuelkontrolbutonu = 1$
 $kumandabirimietkin = 0$ $pasif_kol_serbest = 1$
 $merkezleme = 0$ $endoskop_iceride = 0$

$aktif_kol_merkezli = 0$

An example UPPAAL output of a Transitions:

$CerrahKontrol.d3 \rightarrow CerrahKontrol.d3$
 $\{manuelkontrolbutonu == 0 \ \&\& \ merkezleme == false,$
 $mkb!,manuelkontrolbutonu := 1 \}$
 $PasifdEngelenmisKol.FRENLI \rightarrow PasifdEngelenmisKol.SERBEST$
 $\{1,mkb?,pasif_kol_serbest := true\}$

An example UPPAAL output of a Clock:

$rt \geq 6$

So we convert UPPAAL outputs to Excel files. Our three of test suites as shown in Appendix B.

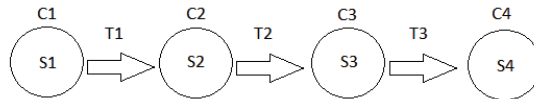


Figure 5.7. One Path Example

We examined all counter example to convert readable format. Figures B.1, B.2 and B.3's first column shows clock of system; second column shows state cases which are initial state, intermediate state, and fail state; third column shows transition and fourth column shows state.

However, there are many trace to test. It takes lot of time search all traces. In literature it called state explosion problem. We try to solve this problem following Sub-sections 5.3, 5.3.1, 5.3.2 and 5.3.4.

After examining all the states,transitions and clocks, there is possible to gave them unique names like S1 (for state), T1(for transition), C1(for clock).

5.3. Solving State Explosion Problem

The state space can be infinite or very large which is one of the most serious problems with model checking in practice is the **state explosion problem**. In terms of limited resources of time and memory, it is impossible to explore the entire state space.

5.3.1. Apriori Algorithm

Apriori algorithm is used for frequent item set mining and association rule learning over transactional databases. We can explain this approach as "bottom up". In this approach, frequent subsets are extended one item at a time (a step called as candidate generation), then group of candidates are tested against the data. If there is no more successful extensions are found, the algorithm terminates (Agrawal and Srikant, 1994).

In the first step, Apriori counts item sets then it generates candidate item sets of length "k" from item sets of length "k-1". Then, the algorithm eliminates the "non frequent" candidates. After from this process, with using non-frequent candidates, the algorithm generates "k+1" items and then the algorithm runs with the same logic which is explained from the first step until now. The algorithm terminates when no further successful extensions are found.

5.3.2. Prefix Span

In this section, we introduce PrefixSpan algorithm which is a pattern-growth method for mining sequential patterns. In this method, the projection is based only on frequent prefixes because any frequent subsequence can always be found by growing a frequent prefix (Pei et al., 2001).

1. Step 1: Find length-1 sequential patterns;
2. Step 2: Divide search space;
3. Step 3: Find subsets of sequential patterns by constructing corresponding projected databases and mine each recursively.

5.3.3. Apriori and Prefix Span Results

We expect that the results of test cases of the model mutants will not fail if the requirements are determined correctly. Therefore, we run our original NeuRoboscope model xml to create 1000 outputs. After that converting excel files, we produced transition from one possible state to another state as shown in Figure C.2 and C.1

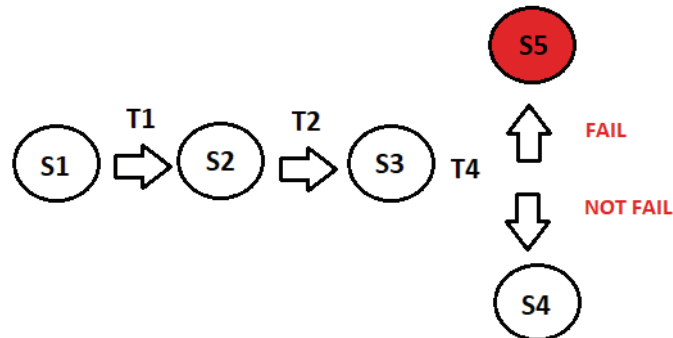


Figure 5.8. Expected Result Example

Figure 5.8 visualize what we expect after an execution test case.

5.3.4. Greedy Heuristic Approach

Greedy Heuristic approach which chooses locally optimal solutions in order to approximate a globally optimal solution.

We used this approach to generate test cases that will decrease number of resets. By help of this, testers can gain a speed-up on the test execution of test cases.

For example Figure 5.9 shows an example of test cases. While applying greedy approach we choose the states with highest number of occurrences from beginning and ending then we combine these two states we continue this process until we cover of all of the test cases.

Result of Figure 5.9 are:

- *E1, E2, E6, E4*

- *E3, E8*
- *E9, E7*
- *E5*

For example, we will read these output from *E1, E2, E6, E4* If there are no faults in the any steps, these testcases can be combined without resetting. So our new test case is *S1, T1, S2, T2, S3, T4, S4, T5, S5, T3, S1, T1, S2, T2, S3, T4, T5, S5, T3, S6, T6, S3.*

In this example, number of test cases drops to four with our combined test cases.

| Test Case ID | Test Case Steps |
|--------------|------------------------------------|
| E1 | (S1),(T1),(S2),(T2),(S3),(T4),(S4) |
| E2 | (S4),(T5),(S5),(T3),(S1),(T1),(S2) |
| E3 | (S1),(T6),(S3),(T4),(S4) |
| E4 | (S4),(T5),(S5),(T3),(S6),(T6),(S3) |
| E5 | (S2),(T2),(S3),(T4),(S4),(T5),(S1) |
| E6 | (S2),(T2),(S3),(T4),(S4) |
| E7 | (S1),(T6),(S7),(T8),(S4) |
| E8 | (S4),(T5),(S5),(T4),(S4) |
| E9 | (S1),(T4),(S4),(T3),(S1) |

Figure 5.9. An Example of Test Cases

CHAPTER 6

VERIFICATION WITH BLACK-BOX TESTING

Black box testing technique is to check whether the requirements is sufficient only through inputs and outputs. In this technique, the system's internal flow is unknown. The cause-effect graph test, one of the dynamic black box testing techniques, has been identified as one of the most suitable techniques for verifying the NeuRoboScope system following the literature review. Cause effect graph test provides the demonstration of logically based relationships of inputs and outputs and the creation of test cases.

The basic operations were used to construct the logical based cause effect graph as in shown Figure 6.1.

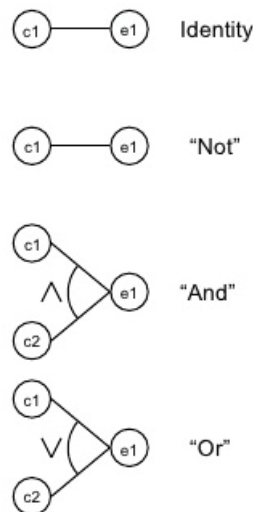


Figure 6.1. Basic Operations Which Is Used When Creating Cause Effect Graph

Some environmental constraints are required while analyzing and graphing system requirements. The following constraints are used in the notation as shown Figure 6.2:

- E (Exclusive-OR) means that, at most, one of a and b can be true.
- I (Inclusive-OR) states that at least one of a and b must be true.

- O states that one and only one of a and b must be 1 (True).
- R states that for a to be 1, b must be 1, in other words, it is impossible for a to be 1 and b to be 0 (False).
- M is used to describe the relations among effects. It states that if effect e is 1, effect f is forced to 0.

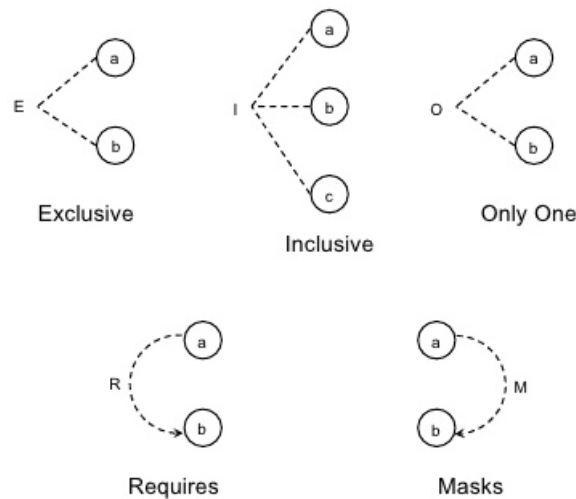


Figure 6.2. Constraints which is used when creating cause effect graph

6.1. Unique MC/DC

Unique MC/DC is an approach of MC/DC techniques. In shortly MC/DC, set of test cases to show that changing the cause's value for each particular changes the outcome of the effects independently from the values of the other effects.(Chilenski and Miller, 1994; Chilenski, 2001)

A test set conforming to MC/DC consists of test cases that guarantee that(Kelly J. et al., 2001):

- Every point of entry and exit in the model has been invoked at least once,

- Every basic condition in a decision in the model has been taken on all possible outcomes at least once, and
- Each basic condition has been shown to independently affect the decisions outcome.

For example $E3 = (C1 \vee C2) \wedge C3$: The truth table is given in Table 6.1.

The independence pairs for the variable C1 is (2, 6) (C1 changes from 0 to 1, the other variables remain the same, the outcome changes from 0 to 1) The independence pair for the variable C2 is (2, 4) and the independence pair for the variable C are (3, 4), (5, 6) and (7, 8).

The test set for MC/DC is a subset of the test cases of the different independence pairs. This subset has to be chosen in such a way that for each variable (at least) one independence pair is covered (in general, there are multiple valid subsets). For the example we have the test set for MC/DC consisting of {2, 3, 4} plus one test case of {5, 6, 7, 8}. The number of test cases for N conditions is $N + 1$.

Table 6.1. Example Test Set

| Test Case Number | C1 | C2 | C3 | E1 |
|------------------|----|----|----|----|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 |

6.2. NeuRoboScope Cause Effect Graphs

To examine the project, the light-related and safety-related graphics were scrutinized separately.

6.2.1. Preliminary Information

Firstly drawing NeuRoboScope cause effect graphics, we analyse the system requirement to determine causes, effects, internal nodes and constraints.

6.2.1.1. Safety-Related Cause Nodes

Safety-Related cause nodes as shown below list:

- C1: CEK is fully pressed.
- C2: CEK is half pressed.
- C3: CEK is released.
- C4: KBE is released.
- C5: KBE is pressed.
- C6: MKB is pressed.
- C7: MKB is released.
- C8: ASB is created.
- C9: Endoscope is outside.

6.2.1.2. Light-Related Cause Nodes

Light-Related cause nodes as shown below list:

- C10: Excessive force/movement on tissue.
- C11: KBE is released.
- C12: KBE is pressed.
- C13: Homming is taking.

- C14: Asb is not created.
- C15: Endoscope is outside.
- C16: Homming is needed.

6.2.1.3. Safety-Related Effect Nodes

Safety-Related effect nodes as shown below list:

- E1: PEK is released.
- E2: KEY is released.
- E3: Homming is taking.

6.2.1.4. Light-Related Effect Nodes

Light-Related effect nodes as shown below list:

- E4: Yellow light is on.
- E5: Green light is on.
- E6: Green light is blinking.
- E7: Red light is blinking.
- E8: Red light is on.

6.2.1.5. Safety-Related Internal Nodes

Safety-Related internal nodes as shown below list:

- I1: OR,

- I2: AND,
- I3: AND,
- I4: AND.

6.2.1.6. Safety-Related Constraint Nodes

Safety-Related constraint nodes as shown below list:

- O: C1,C2,C3,
- O: C4,C5,
- O: C6,C7.

6.2.1.7. Light-Related Constraint Nodes

Light-Related constraint nodes as shown below list:

- O: C11,C12,
- M: E5,E6,
- M: E7,E8.

6.2.2. Safety-Related NeuRoboScope Cause Effect Graphic

After determine safety-related causes, effects and internal nodes, we draw graph with desktop application Gephi(Gephi, 2016) as in shown Figure 6.3.

6.2.2.1. Safety-Related NeuRoboScope Cause Effect Graphic Mathematical Transformation

Effect nodes, Intermediate nodes and Cause nodes mathematical transformation list is as follows:

- $E1 : I1 \& I2$
- $E2 : I3 \& I4$
- $E3 : (C3 \& C7 \& C9)$
- $I1 : (C2 \parallel C6)$
- $I2 : (!C1 \& !C3 \& !C7)$
- $I3 : (C3 \& C5 \& C7 \& C8)$
- $I4 : (!C2 \& !C6 \& !C4 \& !C1)$

The mathematical transformation of the safety-related cause effect graphic resulting from the placement of intermediate nodes is as follows:

- $E1 : (C2 \parallel C6) \& (!C1 \& C3 \& !C7)$
- $E2 : (C3 \& C5 \& C7 \& C8) \& (!C2 \& !C6 \& !C4 \& !C1)$
- $E3 : (C3 \& C7 \& C9)$

6.2.2.2. Reading of Safety-Related NeuRoboScope Cause Effect Graphic Mathematical Transformation

PEK is released when CEK is half pressed or CEK is fully pressed. PEK is braked when CEK is fully pressed or CEK is released or MKB is released. KEY is released when CEK is released, KBE is pressed, MKB is released and ASB is created.

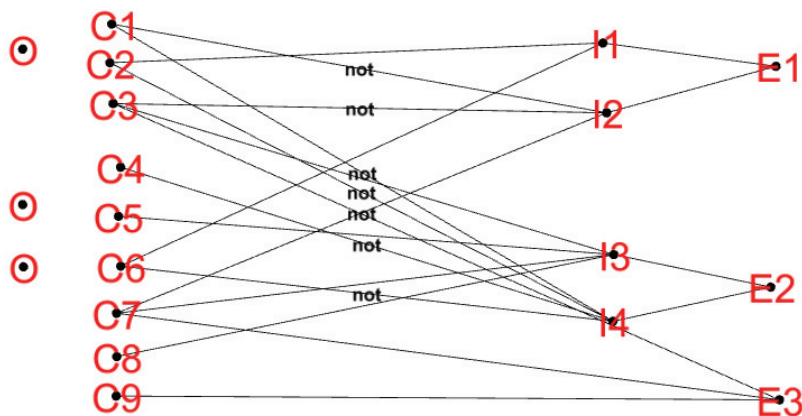


Figure 6.3. Safety-Related NeuRoboScope Cause Effect Graphic

KEY locked when CEK is half pressed or MKB is pressed or KBE is released or CEK is fully pressed. Homming is taking when CEK is released and MKB is released and Endoscope is outside.

6.2.2.3. Unique MC/DC Test Suite of Safety-Related NeuRoboScope Cause Effect Graphic

Using with Safety-Related NeuRoboScope Cause Effect Graphic Unique MC/DC Test Suite is generated as shown in Table 6.2.

Table 6.2. Test Suite For Safety Related NeuRoboScope Requirement

| Test Case Number | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | E1 | E2 | E3 |
|------------------|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 7 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 8 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 9 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

6.2.3. Light-Related NeuRoboScope Cause Effect Graphic

Light-Related NeuRoboScope cause effect graphic is shown Figure 6.4 which is draw with Gephi after determining light-related causes and effects.

6.2.3.1. Light-Related NeuRoboScope Cause Effect Graphic

Mathematical Transformation

Effect and Cause nodes mathematical transformation list is as follows:

- $E4 : C10$
- $E5 : C11$
- $E6 : (C12 \parallel C13)$
- $E7 : C14$
- $E8 : (C15 \& C16)$

6.2.3.2. Reading of Light-Related NeuRoboScope Cause Effect Graphic Mathematical Transformation

Yellow light is on when excessive force/movement is on tissue. Green light is on when KBE is released. Green Light is blinking when KBE is pressed or during homming.



Figure 6.4. Light-Related NeuRoboScope Cause Effect Graphic

Red light is blinking when asb is not created. Red light is on when endoscope is outside or homming is needed.

6.2.3.3. Unique MC/DC Test Suite of Light-Related NeuRoboScope Cause Effect Graphic

Using with Light-Related NeuRoboScope Cause Effect Graphic Unique MC/DC Test Suite is generated as shown in Table 6.3.

Table 6.3. Test Suite For NeuRoboScope of Light Requirement

| Test Case Number | C10 | C11 | C12 | C13 | C14 | C15 | C16 | E4 | E5 | E6 | E7 | E8 |
|------------------|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 7 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

6.2.4. GraphML Files

After drawing graph with Gephi, we exported .grapml file. GraphML file is a type of XML file which attributes mentioned above.

Important nodes attributes to define in gephi:

- grade: Given depth to visualition,
- label: Given name of nodes,
- type: Type of node such as CAUSE, EFFECT,
- relation: Relation between nodes such as AND, OR,

Important edge attributes to define in gephi:

- source: ID for start node of edge,
- target: ID for end node of edge,
- edgeid: Specific ID of edge,
- neg: Negation of edge.

CHAPTER 7

VERIFICATION WITH WHITE-BOX TESTING

The most general description of this test technique is that looking inside the code. The tester checks that the input-output, code design, and whether the code work correctly.

This testing type provide more comprehensive tests to development stage and optimize the codes and also gives change to removing extra lines of codes. And also it is possible to find out fault and error from codes before user interface tests. These tests include testing the system's source code with elements such as statements, control flows, and so on.

To make this test technique, the coding information must be at a upper level. Because the test will be done directly at the code level, not at the user interface level.

Table 7.1. Required Coverage Depends on Safety Level

| | Level A | Level B | Level C | Level D |
|-------------------------------------|---------|---------|---------|---------|
| Statement Coverage | x | x | x | |
| Decision Coverage | x | x | | |
| Modified Condition/DecisionCoverage | x | | | |

Advantages of White Box Testing: It forces the test developer to reason carefully about implementation. It reveals errors in hidden code. Besides it spots the dead code or other issues with respect to best programming practices.

Disadvantages of White Box Testing: It is expensive and spend so many time and money. It does not determine all possibility errors. It may miss few lines accidentally . For using this technique, tester should know in-depth knowledge about the programming language to perform white box testing.

There are several sub technique for white box testing:

Statement Coverage: This technique is used for exercising all programming statement with minimal test.

Branch Coverage: This technique is running number of tests to ensure that all branches are tested at least once.

Path Coverage: This technique related to testing all possible paths which means

that each statement and branch is covered.

Modified Condition/Decision Coverage: In particular, it is an important way to test at a sufficient level for critical software in terms of safety. As a result, if a choice can be made, there is a technique for testing all possible decisions that contribute to that option(Chang and Huang, 2007).

Since NeuRoboScope is Level A as previously mentioned Section 3.2, the mc / dc coverage and unique mc/dc coverage scope has been chosen to examine the device as can be seen from Table 7.1.

$$MC/DC = Decision\ coverage + additional\ requirements$$

1. Every point of entry and exit in the model has been invoked at least once,
2. Every condition in a decision in the program has taken all possible outcomes at least once,
3. Each condition in a decision has been shown to independently affect that decision's outcome.

With all other conditions constant, changing the truth value of the condition changes the result of the decision.

Algorithm 1 Example Algorithm For MC/DC Coverage

Input:

A: One of Example Input;

B: One of Example Input

Output: C: Example Output

```
1: if  $A > 10$  and  $B == 0$  then  
2:    $C = A + B$ ;  
3: end if
```

You can use test suite for MC/DC coverage:

Test Suite: (T, T) , (T, F) , (F, F)

Unique MC/DC: $A > 10$ and $B == 0$: The truth table is given in Table 7.2 for Algorithm 1 .

The independence pairs for the variable $A > 10$ is (1, 3) ($A > 10$ changes from T to F, the other variables remain the same, the outcome changes from T to F) The independence pair for the variable $B == 0$ is (1, 2) ($B == 0$ changes from T to F, the other variables remain the same, the outcome changes from T to F)

The test set for unique MC/DC is a subset of the test cases of the different independence pairs. This subset has to be chosen in such a way that for each variable (at least) one independence pair is covered (in general, there are multiple valid subsets).

Table 7.2. Example Algorithm Test Cases

| Test Case No | A>10 | B==0 | A>10 and B==0 |
|--------------|------|------|---------------|
| 1 | T | T | T |
| 2 | T | F | F |
| 3 | F | T | F |
| 4 | F | F | F |

For the example we have the test set for unique MC/DC consisting of {1, 2, 3}. The number of test cases for N conditions is $N + 1$.

CHAPTER 8

IMPLEMENTED TOOL

VERTICALS which is an open-source web-based software system testing tools which provides generating test cases to safety critical systems.

VERTICALS generate test cases with different testing methods:

1. Cause Effect Graph Testing,
2. Model Based Testing.

The basic steps to generate results with using VERTICALS are as follows:

1. Select preference testing methods from menu,
2. Select the file users want to analyse,
3. Wait for producing test cases,
4. Save the results.

VERTICALS includes home page, cause effect testing page, model mutation page, specification page, test suite generation page and also help page and their figures added in Appendix A.

Home pages give information aims and purpose of tool and creators as shown Figure A.1.

Cause Effect Testing pages produce test cases from graphML file as shown Figure A.2. To produce test cases from cause effect graph, users can upload graphML file with browse and press "Generate Test Cases" button. After generating test cases saving Unique MC/DC Tests cases and all test cases is possible. Example graphML file also downloadable from tool.

Model based testing examined three menus. One of them is model mutation menu which is used for generating model mutants as shown Figure A.3. First of all users select model mutation from menu. After this step users select a XML file which is generated by UPPAAL and then press "Generate Mutants" button to produce mutants then save the mutants. The other menu is specification mutation which is used for generating specification

mutants as shown Figure A.4. It includes the same steps as the above-mentioned menu. As well as these menus, there is test suite generation menu as shown Figure A.5. After selecting test suite generation menu, users can be able to upload csv file of id if they want convert readable format to UPPAAL model ids. This step can be skipped by users. Then users select the file a zip file which is consists of UPPAAL verification result documents and press "Unzip Files" button to for generating test suites and then save the results.

Help page also gives information how to use VERTICALS as shown Figure A.6.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In this thesis, the model of the NeuRoboScope system is created then model based verification of NeuRoboScope system was carried out. In the system, which is developed in this project while these entries are gathered with the help of a control arm in the literature, a movement system is systematically created so that the movement information does not unnecessarily effort during the surgery and does not take surgeon's sight from the surgical region. The system is modeled using with timed automata, the created model is verified against to system properties and system verifies all queries. Model verification studies will be performed immediately after the design is implemented in the system. Although the model based testing can not replace observing the device, model based verification provide the speed, quickness and assumptions of the experiments, and lightening of the material burden. Finally, it is planned to use the model checker in the test phase when the system is complete.

NeuRoboScope Uppaal model mutants have been produced and from these mutants the sufficient test cases for the system have been made readable and ready for testing.

The requirements of the device were drawn from the gephi tool using cause effect graph and graph exported file of .graphml. Unique MC / DC test suite were produced with VERTICALS which convert graph relations to boolean expression to create test cases.

Since a device at Level A needs to be examined with the MC / DC coverage technique, white box testing technique with Unique MC / DC coverage has been added to generate test cases to VERTICALS.

In order to use our VERTICALS tool in the sector, it will be opened on the internet and it codes will be shared.

REFERENCES

- Agrawal, R. and R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, San Francisco, CA, USA, pp. 487–499. Morgan Kaufmann Publishers Inc.
- Aichernig, B. K., F. Lorber, and D. Nickovic (2013). Time for mutants model based mutation testing with timed automata. pp. 20–21.
- Alemzadeh, H., R. K. Iyer, Z. Kalbarczyk, and J. Raman (2013, July). Analysis of safety-critical computer failures in medical devices. *IEEE Security Privacy* 11(4), 14–26.
- Almeida, J., L. De Albuquerque, M. Dal Fabbro, M. Sampaio, R. Medina, M. Chacon, and J. Gondim (2015, November). Endoscopic skull base surgery: evaluation of current clinical outcomes. *Journal of neurosurgical sciences*.
- Ates, G., R. Majani, and M. Dede (2018). Design of teleoperation scheme with a wearable master for minimally invasive surgery.
- Baier, C. and J.-P. Katoen (2008). *Principles of Model Checking*. The MIT Press.
- Barkhoudarian, G. (2015). Endoscopic pituitary surgery. <https://atlasofscience.org/the-resilient-pituitary-gland/>. Online; Accessed: 2018-07-17.
- Behrmann, G., R. David, and K. G. Larsen (2004). A tutorial on uppaal. pp. 200–236. Springer.
- Berker, M., I. Isikay, D. Berker, M. Bayraktar, and A. Gürlek (2013). Early promising results for the endoscopic surgical treatment of cushings disease. *Neurosurgical Review* 37, 105–114.
- Chang, J. and C. Huang (2007, July). A study of enhanced mc/dc coverage criterion for

software testing. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, Volume 1, pp. 457–464.

Chilenski, J. J. (2001). An investigation of three forms of the modified condition decision coverage (mcdc) criterion. Technical report, Office of Aviation Research.

Chilenski, J. J. and S. P. Miller (1994, Sept). Applicability of modified condition/decision coverage to software testing. *Software Engineering Journal* 9(5), 193–200.

Dan, Y. and M. Shilong (2011, June). A method of analysis and verification for safety-critical software based on modelling and testing. In *2011 Fifth International Conference on Secure Software Integration and Reliability Improvement - Companion*, pp. 3–4.

EvaluatePharma (2017). World Preview 2017 , Outlook to 2022. (September).

Fraser, G., F. Wotawa, and P. E. Ammann (2007). Testing with model checkers: a survey. *Software Testing, Verification and Reliability* 19(3), 215–261.

Gabbay, D. M., I. Hodkinson, and M. Reynolds (1994). *Temporal Logic (Vol. 1): Mathematical Foundations and Computational Aspects*. New York, NY, USA: Oxford University Press, Inc.

Gephi (2016). Gephi: The open graph viz platform. <https://gephi.org/>. Online; Accessed: 2016-08-04.

Healty-Inside (2018). Pituitary tumors. <https://www.healthy-inside.com/health/hormonal-disorders/pituitary-gland-tumors/>. Online; Accessed: 2018-07-17.

Hrgarek, N. (2012, June). Certification and regulatory challenges in medical device software development. pp. 40–43.

IEC (2005). IEC 60601-1:Medical electrical equipment Part 1: General requirements for basic safety and essential performance.

- IEC (2006a). IEC 61025: Fault tree analysis.
- IEC (2006b). IEC 62304:2006:Medical device software – Software life cycle processes.
- ISO (2007). ISO 14971:2007 Medical devices Application of risk management to medical devices.
- Kelly J., H., V. Dan S., C. John J., and R. Leanna K. (2001). A practical tutorial on modified condition/decision coverage. Technical report.
- Kim, J., Y.-J. Lee, S.-Y. Ko, D.-S. Kwon, and W.-J. Lee (2004, Sept). Compact camera assistant robot for minimally invasive surgery: Kalar. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Volume 3, pp. 2587–2592 vol.3.
- Knight, J. C. (2002, May). Safety critical systems: challenges and directions. In *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, pp. 547–550.
- Laws, E. R., J. J.-M. Wong, T. R. S. Smith, K. M. de los Reyes, L. Aglio, A. J. Thorne, D. J. Cote, F. Esposito, P. Cappabianca, and A. A. Gawande (2016). A checklist for endonasal transsphenoidal anterior skull base surgery. *Journal of neurosurgery* 124 6, 1634–9.
- Lutz, R. R. and I. C. Mikulski (2003). Requirements discovery during the testing of safety-critical software. In *Proceedings of the 25th International Conference on Software Engineering, ICSE '03*, Washington, DC, USA, pp. 578–583. IEEE Computer Society.
- Malm, T., M. Vuori, J. Rauhamäki, T. Vepsäläinen, J. Koskinen, J. Seppälä, H. Virtanen, and M. Hietikko (2011). Safety-critical software in machinery applications.
- Palanichamy, G. (2010). Basic Principles of Risk Management for Medical Device Design.

- Pei, J., J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu (2001, April). Prefixspan,: mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings 17th International Conference on Data Engineering*, pp. 215–224.
- RTCA (2011). RTCA DO-178C : Software Considerations in Airborne Systems and Equipment Certification. Washington. RTCA Inc.
- Schurr, M. O., A. Arezzo, B. Neisius, H. Rininsland, H. U. Hilzinger, J. Dorn, K. Roth, and G. F. Buess (1999). Trocar and instrument positioning system tiska. an assist device for endoscopic solo surgery. *Surgical endoscopy* 13 5, 528–531.
- UPPAAL (2012). Uppaal. <http://www.uppaal.org>. Online; Accessed: 2016-03-17.
- Uzun, B., T. Ayav, M. Berker, B. Ozdemirel, G. Kiper, O. Maarroof, M. Dede, I. Isikay, c. Hanalioglu, E. Tatlıcioğlu, O. Işıtman, G. Ateş, and A. Yaşır (2016, 09). Neurobo-scope sisteminin model tabanlı sınaması. pp. 347–352.
- Wang, Y., D. R. Laby, K. P.and Uecker, A. A. Mangaser, and M. Ghodoussi (1999). Automated Endoscope System For Optimal Positioning Patent No: US005878193A.
- Yaşır, A. and G. Kiper (2018). Structural Synthesis of 2R1T Type Mechanisms for Minimally Invasive Surgery Applications. pp. 31–38.

APPENDIX A

SAMPLE SCREENSHOTS OF VERTICALS

VERTICALS (Verification and Testing of Safety Critical Software)

Home Cause Effect Graph Testing Model Mutation Specification Mutation Test Suite Generation Help

VERTICALS (Verification and Testing of Safety Critical Software) is an open-source web-based software system testing tool which provides generating test cases to safety critical software systems

VERTICALS generate test cases with different testing methods:

1. Cause Effect Graph Testing
2. Model Based Testing

Using VERTICALS

The steps for generating results are as follows:

1. Select preference testing methods from menu
2. Select the file you want to analyze
3. Wait for producing test cases
4. Save the results

Project Team:
Bengisu Uzun Yenigün, bengisuuzun@tubitak.gov.tr
Tolga Ayav, tolgaayav@yte.edu.tr

Dependable Components And System Lab
Department of Computer Engineering
Izmir Institute of Technology
Urla, Izmir, Turkey



Figure A.1. Sample Screenshot of VERTICALS Home Page

VERTICALS (Verification and Testing of Safety Critical Software)

Home Cause Effect Graph Testing Model Mutation Specification Mutation Test Suite Generation Help

Choose GRAHPML File
Browse... No file selected

GENERATE TEST CASES

↓ DOWNLOAD UNIQUE MC/DC TEST CASES

↓ DOWNLOAD ALL TEST CASES

↓ DOWNLOAD EXAMPLE GRAHPML

Reminder: Modified Condition/Decision Coverage, MC/DC
Modified Condition/Decision Coverage (MC/DC) is one of coverage analysis method to provide sufficient test cases for Safety Critical Software Systems.
Basically MC/DC is:

1. Every point of entry and exit in the program has been invoked at least once
2. Every condition in a decision in the program has taken all possible outcomes at least once
3. Every decision in the program has taken all possible outcomes at least once, each condition in a decision has been shown to independently affect that decision's outcome. A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible condition

Figure A.2. Sample Screenshot of VERTICALS Cause Effect Graph Testing Page



Figure A.3. Sample Screenshot of VERTICALS Model Mutation Page



Figure A.4. Sample Screenshot of VERTICALS Specification Mutation Page

VERTICALS (Verification and Testing of Safety Critical Software)

Home Cause Effect Graph Testing Model Mutation Specification Mutation Test Suite Generation Help

Choose ID File
Browse... No file selected

Upload Zip file
Browse... No file selected
Unzip Files

Generate Test Suites
Download Test Suites

Figure A.5. Sample Screenshot of VERTICALS Test Suite Generation Page

VERTICALS (Verification and Testing of Safety Critical Software)

[Home](#)

[Cause Effect Graph Testing](#)

[Model Mutation](#)

[Specification Mutation](#)

[Test Suite Generation](#)

[Help](#)

[How to Use Cause Effect Testing Tool](#)

[How to Use Model Mutation Tool](#)

[How to Use Specification Mutation Tool](#)

[Documentation](#)

The steps for creating mutants are as follows:

The steps for verification with model mutation:

1. Select Model Mutation from Menu.
2. Select the file a XML file which is generated by UPPAAL.
3. Press Generate Mutants button.
4. Wait for producing mutants.
5. Save the mutants.

Figure A.6. Sample Screenshot of VERTICALS Help Page

APPENDIX B

EXAMPLES OF TEST SUITE PRODUCED BY VERTICALS

| | | |
|-------|--------------------|--|
| rt<=6 | Initial State | <pre>(CerrahEndoskopKavramaBASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngelenniskolFRENL1 aktifEndoskopYonlendiriciLETKINDEGIL EndoskopBolgesi_id7 KumandaPili_id23 Kirmizisk_YANIP_SONUyor Yesilisk_YANMIYOR) rt<=6, rt.CerrahEndoskopSurme.t<=0, rt.PasifEngelenniskol.t<=0, rt.aktifEndoskopYonlendirici.t<=0, rt.EndoskopBolgesi.t<=0, rt- KumandaPili.t<=0, rt.Yesilisk.ty<=0, rt.#au<=0, CerrahEndoskopSurme.t<=6, CerrahEndoskopSurme.t-rt<=0, CerrahEndoskopSurme.t- PasifEngelenniskol.t<=0, CerrahEndoskopSurme.t-aktifEndoskopYonlendirici.t<=0, CerrahEndoskopSurme.t-EndoskopBolgesi.t<=0, CerrahEndoskopSurme.t-KumandaPili.t<=0, CerrahEndoskopSurme.t-Yesilisk.ty<=0, CerrahEndoskopSurme.t-#au<=0, PasifEngelenniskol.t- PasifEngelenniskol.t-rt<=0, PasifEngelenniskol.t-EndoskopBolgesi.t<=0, PasifEngelenniskol.t-EndoskopBolgesi.t-rt<=0, PasifEngelenniskol.t-EndoskopBolgesi.t-#au<=0, PasifEngelenniskol.t-EndoskopBolgesi.t-#au-rt<=0, PasifEngelenniskol.t- aktifEndoskopYonlendirici.t-rt<=0, aktifEndoskopYonlendirici.t-EndoskopBolgesi.t<=6, aktifEndoskopYonlendirici.t-rt<=0, aktifEndoskopYonlendirici.t-CerrahEndoskopSurme.t<=0, aktifEndoskopYonlendirici.t-KumandaPili.t<=0, aktifEndoskopYonlendirici.t- Yesilisk.ty<=0, aktifEndoskopYonlendirici.t-#au<=0, EndoskopBolgesi.t<=6, EndoskopBolgesi.t-rt<=0, EndoskopBolgesi.t- CerrahEndoskopSurme.t<=0, EndoskopBolgesi.t-PasifEngelenniskol.t<=0, EndoskopBolgesi.t-aktifEndoskopYonlendirici.t<=0, EndoskopBolgesi.t-KumandaPili.t<=0, EndoskopBolgesi.t-Yesilisk.ty<=0, EndoskopBolgesi.t-#au<=0, KumandaPili.t<=6, KumandaPili.t-rt<=0, KumandaPili.t-CerrahEndoskopSurme.t<=0, KumandaPili.t-PasifEngelenniskol.t<=0, KumandaPili.t-aktifEndoskopYonlendirici.t<=0, KumandaPili.t-EndoskopBolgesi.t<=0, KumandaPili.t-Yesilisk.ty<=0, KumandaPili.t-#au<=0, Yesilisk.ty<=6, Yesilisk.ty-rt<=0, Yesilisk.ty- CerrahEndoskopSurme.t<=0, Yesilisk.ty-PasifEngelenniskol.t<=0, Yesilisk.ty-aktifEndoskopYonlendirici.t<=0, Yesilisk.ty- EndoskopBolgesi.t<=0, Yesilisk.ty-KumandaPili.t<=0, Yesilisk.ty-#au<=0, #au-rt<=0, #au-CerrahEndoskopSurme.t<=0, #au- PasifEngelenniskol.t<=0, #au-aktifEndoskopYonlendirici.t<=0, #au-EndoskopBolgesi.t<=0, #au-KumandaPili.t<=0, #au-Yesilisk.ty<=0 ameliyat_bolgesi:=0 aktif_kol:=0 merkezkol:=0 pikturnu:=2 cek=0 eno=0 manuelkontrolbutonu=0 kumandabrimmekin=0 pasif_kol_serbest=0 merkezkolme=0 merkezkol:=0 aktif_kol:=0 merkezkol:=0</pre> |
| | | <pre>(CerrahEndoskopKavramaBASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngelenniskolSERBEST aktifEndoskopYonlendiriciLETKINDEGIL EndoskopBolgesi_id7 KumandaPili_id23 Kirmizisk_YANIP_SONUyor Yesilisk_YANMIYOR) rt<=6, rt.CerrahEndoskopSurme.t<=0, rt.PasifEngelenniskol.t<=0, rt.aktifEndoskopYonlendirici.t<=0, rt.EndoskopBolgesi.t<=0, rt- KumandaPili.t<=0, rt-Yesilisk.ty<=0, rt-#au<=0, CerrahEndoskopSurme.t<=6, CerrahEndoskopSurme.t-rt<=0, CerrahEndoskopSurme.t- PasifEngelenniskol.t<=0, CerrahEndoskopSurme.t-aktifEndoskopYonlendirici.t<=0, CerrahEndoskopSurme.t-EndoskopBolgesi.t<=0, CerrahEndoskopSurme.t-KumandaPili.t<=0, CerrahEndoskopSurme.t-Yesilisk.ty<=0, CerrahEndoskopSurme.t-#au<=0, PasifEngelenniskol.t- PasifEngelenniskol.t-rt<=0, PasifEngelenniskol.t-EndoskopBolgesi.t<=0, PasifEngelenniskol.t-KumandaPili.t<=0, PasifEngelenniskol.t-Yesilisk.ty<=0, PasifEngelenniskol.t-#au<=0, aktifEndoskopYonlendirici.t<=6, aktifEndoskopYonlendirici.t- aktifEndoskopYonlendirici.t-CerrahEndoskopSurme.t<=0, aktifEndoskopYonlendirici.t-KumandaPili.t<=0, aktifEndoskopYonlendirici.t- Yesilisk.ty<=0, aktifEndoskopYonlendirici.t-#au<=0, EndoskopBolgesi.t<=6, EndoskopBolgesi.t-rt<=0, EndoskopBolgesi.t- CerrahEndoskopSurme.t<=0, EndoskopBolgesi.t-PasifEngelenniskol.t<=0, EndoskopBolgesi.t-aktifEndoskopYonlendirici.t<=0, EndoskopBolgesi.t-KumandaPili.t<=0, EndoskopBolgesi.t-Yesilisk.ty<=0, EndoskopBolgesi.t-#au<=0, KumandaPili.t<=6, KumandaPili.t-rt<=0, KumandaPili.t-CerrahEndoskopSurme.t<=0, KumandaPili.t-PasifEngelenniskol.t<=0, KumandaPili.t-aktifEndoskopYonlendirici.t<=0, KumandaPili.t-EndoskopBolgesi.t<=0, KumandaPili.t-Yesilisk.ty<=0, KumandaPili.t-#au<=0, Yesilisk.ty<=6, Yesilisk.ty- CerrahEndoskopSurme.t<=0, Yesilisk.ty-PasifEngelenniskol.t<=0, Yesilisk.ty-aktifEndoskopYonlendirici.t<=0, Yesilisk.ty- EndoskopBolgesi.t<=0, Yesilisk.ty-KumandaPili.t<=0, Yesilisk.ty-#au<=0, #au-rt<=0, #au-CerrahEndoskopSurme.t<=0, #au- PasifEngelenniskol.t<=0, #au-aktifEndoskopYonlendirici.t<=0, #au-EndoskopBolgesi.t<=0, #au-KumandaPili.t<=0, #au-Yesilisk.ty<=0 ameliyat_bolgesi:=0 aktif_kol:=0 merkezkol:=0 pikturnu:=2 cek=0 eno=0 manuelkontrolbutonu=1 kumandabrimmekin=0 pasif_kol_serbest=1 merkezkolme=0 merkezkol:=0 aktif_kol:=0 merkezkol:=0</pre> |
| rt<=6 | Intermediate State | <pre>CerrahKontrol_id3 CerrahKontrol_id3 (manuelkontrolbutonu := 0 && merkezkolme == false, mkbl manuelkontrolbutonu := 1 PasifEngelenniskol.FRE) NLI- >PasifEngelenniskol.SE RBEST (1, mkB7, pasif_kol_serbest := true)</pre> |

Figure B.1. First Picture of Example 1 of Test Suite produced by VERTICALS

| | | | |
|------------------------------|-------------------------------|--|---|
| <pre>rp>=2 rt<=6</pre> | <pre>Intermediate State</pre> | <pre>CerrahEndoskopSurme_ d4- >CerrahEndoskopSurme _id4 { t >= 2 && pasif_kol_serbest == true && merkezeleme == false newposl, t = 0, enc = 1 }</pre> | <pre>(CerrahEndoskopKavramaBASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngellemisKolSERBEST akTifEndoskopYonlendiriciETKINDEGIL EndoskopBolgesi_id7 KumandaPili_id23 KirmizilikYANIP_SONU YOR Yesilisk_YANMIYOR) rp>=2, PasifEngellemisKol t<=2, akTifEndoskopYonlendirici t>=2, EndoskopBolgesi t>=2, KumandaPili t>=2, Yesilisk ty=>2, #au>=2, rt<=6, rt-CerrahEndoskopSurme t<=6, rt-PasifEngellemisKol t<=0, rt-akTifEndoskopYonlendirici t<=0, rt-EndoskopBolgesi t<=0, rt- KumandaPili t<=0, rt-Yesilisk ty<=0, rt-#au<=0, CerrahEndoskopSurme t<=4, CerrahEndoskopSurme t-rt<=2, CerrahEndoskopSurme t- PasifEngellemisKol t<=2, CerrahEndoskopSurme t-akTifEndoskopYonlendirici t<=2, CerrahEndoskopSurme t-EndoskopBolgesi t<=2, CerrahEndoskopSurme t-KumandaPili t<=2, CerrahEndoskopSurme t-Yesilisk ty<=2, CerrahEndoskopSurme t-#au<=2, PasifEngellemisKol t<=6, PasifEngellemisKol t-rt<=0, PasifEngellemisKol t-CerrahEndoskopSurme t<=6, PasifEngellemisKol t- akTifEndoskopYonlendirici t<=0, PasifEngellemisKol t-EndoskopBolgesi t<=0, PasifEngellemisKol t-KumandaPili t<=0, PasifEngellemisKol t-Yesilisk ty<=0, PasifEngellemisKol t-#au<=0, akTifEndoskopYonlendirici t<=6, akTifEndoskopYonlendirici t- akTifEndoskopYonlendirici t-CerrahEndoskopSurme t<=6, akTifEndoskopYonlendirici t-PasifEngellemisKol t<=0, akTifEndoskopYonlendirici t-EndoskopBolgesi t<=0, akTifEndoskopYonlendirici t-KumandaPili t<=0, akTifEndoskopYonlendirici t- Yesilisk ty<=0, akTifEndoskopYonlendirici t-#au<=0, EndoskopBolgesi t<=6, EndoskopBolgesi t-akTifEndoskopYonlendirici t<=0, CerrahEndoskopSurme t<=6, EndoskopBolgesi t-PasifEngellemisKol t<=0, EndoskopBolgesi t-#au<=0, KumandaPili t<=6, KumandaPili t-rt<=0, KumandaPili t-CerrahEndoskopSurme t<=6, KumandaPili t-PasifEngellemisKol t<=0, KumandaPili t-akTifEndoskopYonlendirici t<=0, Yesilisk ty-CerrahEndoskopSurme t<=0, KumandaPili t-Yesilisk ty<=0, KumandaPili t-#au<=0, Yesilisk ty-akTifEndoskopYonlendirici t<=0, Yesilisk ty- EndoskopBolgesi t<=0, Yesilisk ty-KumandaPili t<=0, Yesilisk ty-#au<=0, #au-rt<=0, #au-rt<=0, #au-CerrahEndoskopSurme t<=6, #au- PasifEngellemisKol t<=0, #au-akTifEndoskopYonlendirici t<=0, #au-EndoskopBolgesi t<=0, #au-KumandaPili t<=0, #au-Yesilisk ty<=0 ameliyat_bolgesi_smirt=100 pidurumu=2 cek=0 enc=1 manuelkontroltonu=1 kumandabirimmetkn=0 pasif_kol_serbest=1 merkezeleme=0 endoskop_tceride=0 aktif_kol_merkez=0</pre> |
| <pre>rp>=2 rt<=6</pre> | <pre>Intermediate State</pre> | <pre>CerrahKontrol_id3- >CerrahKontrol_id3 { merkezeleme == false, asbl, 1 } EndoskopBolgesi_id7- >EndoskopBolgesiDISAR IDA (enc = 0, asb? ameliyat_bolgesi_smirt = ReadEncoder()) KirmizilikYANIP_SONU YOR >Kirmizilik_YANMIYOR { 1, asb?, 1 }</pre> | <pre>(CerrahEndoskopKavramaBASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngellemisKolSERBEST akTifEndoskopYonlendiriciETKINDEGIL EndoskopBolgesiDISARIDA KumandaPili_id23 Kirmizilik_YANMIYOR Yesilisk_YANMIYOR) rp>=2, PasifEngellemisKol t<=2, akTifEndoskopYonlendirici t>=2, EndoskopBolgesi t>=2, KumandaPili t>=2, Yesilisk ty=>2, #au>=2, rt<=6, rt-CerrahEndoskopSurme t<=6, rt-PasifEngellemisKol t<=0, rt-akTifEndoskopYonlendirici t<=0, rt-EndoskopBolgesi t<=0, rt- KumandaPili t<=0, rt-Yesilisk ty<=0, rt-#au<=0, CerrahEndoskopSurme t<=4, CerrahEndoskopSurme t-rt<=2, CerrahEndoskopSurme t- PasifEngellemisKol t<=2, CerrahEndoskopSurme t-akTifEndoskopYonlendirici t<=2, CerrahEndoskopSurme t-EndoskopBolgesi t<=2, CerrahEndoskopSurme t-KumandaPili t<=2, CerrahEndoskopSurme t-Yesilisk ty<=2, CerrahEndoskopSurme t-#au<=2, PasifEngellemisKol t<=6, PasifEngellemisKol t-rt<=0, PasifEngellemisKol t-CerrahEndoskopSurme t<=6, PasifEngellemisKol t- akTifEndoskopYonlendirici t<=0, PasifEngellemisKol t-EndoskopBolgesi t<=0, PasifEngellemisKol t-KumandaPili t<=0, PasifEngellemisKol t-Yesilisk ty<=0, PasifEngellemisKol t-#au<=0, akTifEndoskopYonlendirici t<=6, akTifEndoskopYonlendirici t- akTifEndoskopYonlendirici t-CerrahEndoskopSurme t<=6, akTifEndoskopYonlendirici t-PasifEngellemisKol t<=0, Yesilisk ty<=0, akTifEndoskopYonlendirici t-EndoskopBolgesi t<=0, akTifEndoskopYonlendirici t-KumandaPili t<=0, akTifEndoskopYonlendirici t- Yesilisk ty<=0, akTifEndoskopYonlendirici t-#au<=0, EndoskopBolgesi t<=6, EndoskopBolgesi t-akTifEndoskopYonlendirici t<=0, EndoskopBolgesi t-KumandaPili t<=0, EndoskopBolgesi t-Yesilisk ty<=0, EndoskopBolgesi t-#au<=0, KumandaPili t<=6, KumandaPili t-rt<=0, KumandaPili t-CerrahEndoskopSurme t<=6, KumandaPili t-PasifEngellemisKol t<=0, KumandaPili t-akTifEndoskopYonlendirici t<=0, KumandaPili t-EndoskopBolgesi t<=0, KumandaPili t-Yesilisk ty<=0, KumandaPili t-#au<=0, Yesilisk ty-akTifEndoskopYonlendirici t<=0, Yesilisk ty- CerrahEndoskopSurme t<=6, Yesilisk ty-PasifEngellemisKol t<=0, Yesilisk ty-EndoskopBolgesi t<=0, Yesilisk ty-EndoskopBolgesi t<=0, Yesilisk ty- EndoskopBolgesi t<=0, Yesilisk ty-KumandaPili t<=0, Yesilisk ty-#au<=0, #au-rt<=0, #au-rt<=0, #au-CerrahEndoskopSurme t<=6, #au- PasifEngellemisKol t<=0, #au-akTifEndoskopYonlendirici t<=0, #au-EndoskopBolgesi t<=0, #au-KumandaPili t<=0, #au-Yesilisk ty<=0 ameliyat_bolgesi_smirt=100 pidurumu=2 cek=0 enc=1 manuelkontroltonu=1 kumandabirimmetkn=0 pasif_kol_serbest=1 merkezeleme=0</pre> |

Figure B.2. Second Picture of Example 1 of Test Suite produced by VERTICALS

| | | | |
|--------------------------------|-------------------------|--|---|
| <pre> rp>=2 rt<=6 </pre> | <pre> Fail State </pre> | <pre> CerrahEndoskopSurme_1 id_ >CerrahEndoskopSurme_ _id4 { t>= 2 && pasif_kol_serbest == true && merkezieme == false, newposi, t : 0, enc : 2 } </pre> | <pre> endoskop_iceride=0 aktif_kol_merkezi=0 (CerrahEndoskopKavrama.BASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngelenmsKol.SERBEST aktifEndoskopYonlendirici.ETKINDEGIL EndoskopBolgesi.DISARIDA KumandaPili_id23 Kirmizisik.YANMIYOR Yesilisk.YANMIYOR) rp>=4, PasifEngelenmsKol.t>=4, aktifEndoskopYonlendirici.t>=4, EndoskopBolgesi.t>=4, KumandaPili.t>=4, Yesilisk.ty>=4, #tau>=4, rt<=6, rt-CerrahEndoskopSurme.t<=6, rt-PasifEngelenmsKol.t<=0, rt-aktifEndoskopYonlendirici.t<=0, rt-EndoskopBolgesi.t<=0, rt- KumandaPili.t<=0, rt-Yesilisk.ty<=0, rt-#tau<=0, CerrahEndoskopSurme.t<=2, CerrahEndoskopYonlendirici.t<=0, CerrahEndoskopSurme.t-rt<=4, CerrahEndoskopSurme.t- PasifEngelenmsKol.t<=4, CerrahEndoskopSurme.t-aktifEndoskopYonlendirici.t<=4, CerrahEndoskopSurme.t-EndoskopBolgesi.t<=4, CerrahEndoskopSurme.t-KumandaPili.t<=4, CerrahEndoskopSurme.t-Yesilisk.ty<=4, CerrahEndoskopSurme.t-#tau<=4, PasifEngelenmsKol.t<=6, PasifEngelenmsKol.t-rt<=0, PasifEngelenmsKol.t-CerrahEndoskopSurme.t<=6, PasifEngelenmsKol.t- aktifEndoskopYonlendirici.t<=0, PasifEngelenmsKol.t-EndoskopBolgesi.t<=0, PasifEngelenmsKol.t-KumandaPili.t<=0, aktifEndoskopYonlendirici.t<=0, PasifEngelenmsKol.t-#tau<=0, aktifEndoskopYonlendirici.t-PasifEngelenmsKol.t<=0, aktifEndoskopYonlendirici.t-CerrahEndoskopSurme.t<=6, aktifEndoskopYonlendirici.t-KumandaPili.t<=0, aktifEndoskopYonlendirici.t- Yesilisk.ty<=0, aktifEndoskopYonlendirici.t-#tau<=0, EndoskopBolgesi.t<=6, EndoskopBolgesi.t-rt<=0, EndoskopBolgesi.t- EndoskopSurme.t<=6, EndoskopBolgesi.t-PasifEngelenmsKol.t<=0, EndoskopBolgesi.t-aktifEndoskopYonlendirici.t<=0, KumandaPili.t-CerrahEndoskopSurme.t<=6, KumandaPili.t-PasifEngelenmsKol.t<=0, KumandaPili.t-aktifEndoskopYonlendirici.t<=0, KumandaPili.t-EndoskopBolgesi.t<=0, KumandaPili.t-#tau<=0, Yesilisk.ty<=6, Yesilisk.ty-rt<=0, Yesilisk.ty- CerrahEndoskopSurme.t<=6, Yesilisk.ty-PasifEngelenmsKol.t<=0, Yesilisk.ty-aktifEndoskopYonlendirici.t<=0, Yesilisk.ty- EndoskopBolgesi.t<=0, Yesilisk.ty-KumandaPili.t<=0, Yesilisk.ty-#tau<=0, #tau<=6, #tau<=0, #tau-CerrahEndoskopSurme.t<=6, #au- PasifEngelenmsKol.t<=0, #tau-aktifEndoskopYonlendirici.t<=0, #tau-EndoskopBolgesi.t<=0, #tau-KumandaPili.t<=0, #tau-Yesilisk.ty<=0 amalyal_bolgesi_snni=t pidurunu-2 cek=0 enc=2 manuekontrolonu=1 Kumandabirimekin=0 pasif_kol_serbest=1 merkezieme=0 endoskop_iceride=0 aktif_kol_merkezi=0 </pre> |
|--------------------------------|-------------------------|--|---|

Figure B.3. Third Picture of Example 1 of Test Suite produced by VERTICALS

| | | |
|---|--|---|
| <pre> ndirici.ETKINDEGIL- >aktifEndoskopYonle endiinci_id11 {t>= 4, mikzi, t:=0, aktir_kol_merkezleme = true, merkezleme = true} Yesilisk.YANMIYOR - >Yesilisk.YANIPSO NUYOR {1, mikz?, 1 } </pre> | <pre> PasifEngelenmisKolt>=4, EndoskopBolgesi t>=4, Kumandapil t>=4, Yesilisk ty=>4, #au>=4, rt=<=6, rt-CerrahEndoskopSurme k=<=6, rt- PasifEngelenmisKolt<=0, rt-aktifEndoskopYonlendirici t=<=6, rt-EndoskopBolgesi t=<=0, rt-Kumandapil t=<=0, rt-Yesilisk ty=<=0, rt-#au<=0, CerrahEndoskopSurme k<=4, CerrahEndoskopSurme t-EndoskopBolgesi t=<=2, CerrahEndoskopSurme t-PasifEngelenmisKolt<=2, CerrahEndoskopSurme t- aktifEndoskopYonlendirici t=<=4, CerrahEndoskopSurme t-EndoskopBolgesi t=<=2, CerrahEndoskopSurme t-Kumandapil t=<=2, CerrahEndoskopSurme t-Yesilisk ty<=2, PasifEngelenmisKolt-EndoskopBolgesi t=<=6, PasifEngelenmisKolt-rt<=0, PasifEngelenmisKolt-CerrahEndoskopSurme t=<=6, PasifEngelenmisKolt-rt-EndoskopYonlendirici t=<=6, PasifEngelenmisKolt- EndoskopBolgesi t=<=0, PasifEngelenmisKolt-Kumandapil t=<=0, PasifEngelenmisKolt-Yesilisk ty=<=0, PasifEngelenmisKolt-#au<=0, aktifEndoskopYonlendirici t=<=0, aktifEndoskopYonlendirici t-rt<=4, aktifEndoskopYonlendirici t-CerrahEndoskopSurme t=<=0, aktifEndoskopYonlendirici t-PasifEngelenmisKolt k=<=4, aktifEndoskopYonlendirici t-EndoskopBolgesi t=<=4, aktifEndoskopYonlendirici t- Kumandapil t=<=4, aktifEndoskopYonlendirici t-Yesilisk ty=<=4, aktifEndoskopYonlendirici t-EndoskopBolgesi t=<=4, aktifEndoskopYonlendirici t- rt<=0, EndoskopBolgesi t-CerrahEndoskopSurme t=<=6, EndoskopBolgesi t-PasifEngelenmisKolt k=<=0, EndoskopBolgesi t=<=0, EndoskopBolgesi t- Kumandapil t=<=6, Kumandapil t-rt<=0, Kumandapil t-CerrahEndoskopSurme t=<=6, Kumandapil t-PasifEngelenmisKolt k=<=0, Kumandapil t- aktifEndoskopYonlendirici k=<=6, Kumandapil t-EndoskopBolgesi t=<=0, Kumandapil t-Yesilisk ty<=0, Kumandapil t-#au<=0, Yesilisk ty=<=6, Yesilisk ty-rt<=0, Yesilisk ty-CerrahEndoskopSurme t=<=6, Yesilisk ty-PasifEngelenmisKolt k=<=0, Yesilisk ty-aktifEndoskopYonlendirici t=<=6, Yesilisk ty-EndoskopBolgesi t=<=0, Yesilisk ty-Kumandapil t=<=0, Yesilisk ty-#au<=0, #au-rt<=0, #au-CerrahEndoskopSurme t=<=6, #au-PasifEngelenmisKolt k=<=0, #au-aktifEndoskopYonlendirici t=<=6, #au-EndoskopBolgesi t=<=0, #au-Kumandapil t=<=0, #au-Yesilisk ty=<=0 ameilyat_bolgesi_snrir=100 plidurume=2 cek=0 ent<=0 manuelkontrolbutonu=0 kumandabimmetikn=0 pasif_kol_serbest=0 merkezleme=1 endoskop_iceride=0 aktif_kol_merkezleme=1 </pre> | <pre> ndirici.ETKINDEGIL- >aktifEndoskopYonle endiinci_id11 {t>= 4, mikzi, t:=0, aktir_kol_merkezleme = true, merkezleme = true} Yesilisk.YANMIYOR - >Yesilisk.YANIPSO NUYOR {1, mikz?, 1 } </pre> |
| <pre> (CerrahEndoskopKavranma BASILMADI) CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifEngelenmisKolt.FRENLI aktifEndoskopYonlendirici.FRENLI EndoskopBolgesi_id7 Kumandapil_id23 Kimizisik.YANIP_SONU.YOR.Yesilisk.YANIVOR) rt>=4, PasifEngelenmisKolt>=4, EndoskopBolgesi t>=4, Kumandapil t>=4, Yesilisk ty>=4, #au>=4, rt<=12, rt-CerrahEndoskopSurme k=<=6, rt- PasifEngelenmisKolt<=0, rt-aktifEndoskopYonlendirici t=<=6, rt-EndoskopBolgesi t=<=0, rt-Kumandapil t=<=0, rt-Yesilisk ty=<=0, rt-#au<=0, CerrahEndoskopSurme t=<=6, CerrahEndoskopSurme t-rt<=2, CerrahEndoskopSurme t-PasifEngelenmisKolt k=<=2, CerrahEndoskopSurme t- aktifEndoskopYonlendirici t=<=4, CerrahEndoskopSurme t-EndoskopBolgesi t=<=2, CerrahEndoskopSurme t-Kumandapil t=<=2, CerrahEndoskopYonlendirici t-Yesilisk ty=<=2, CerrahEndoskopSurme t-#au<=2, PasifEngelenmisKolt-rt<=0, PasifEngelenmisKolt-CerrahEndoskopSurme t=<=6, PasifEngelenmisKolt-LaktifEndoskopYonlendirici t=<=6, PasifEngelenmisKolt- EndoskopBolgesi t=<=0, PasifEngelenmisKolt-Kumandapil t=<=0, PasifEngelenmisKolt-Yesilisk ty=<=0, PasifEngelenmisKolt- aktifEndoskopYonlendirici t=<=6, aktifEndoskopYonlendirici t-rt<=4, aktifEndoskopYonlendirici t-CerrahEndoskopSurme t=<=0, aktifEndoskopYonlendirici t-PasifEngelenmisKolt k=<=4, aktifEndoskopYonlendirici t-EndoskopBolgesi t=<=4, aktifEndoskopYonlendirici t- Kumandapil t=<=4, aktifEndoskopYonlendirici t-Yesilisk ty=<=4, aktifEndoskopYonlendirici t-#au<=4, EndoskopBolgesi k=<=12, EndoskopBolgesi t- rt=<=0, EndoskopBolgesi t-CerrahEndoskopSurme t=<=6, EndoskopBolgesi t-PasifEngelenmisKolt k=<=0, EndoskopBolgesi t=<=0, EndoskopBolgesi t- Kumandapil t=<=12, Kumandapil t-rt=<=0, Kumandapil t-CerrahEndoskopSurme k=<=6, Kumandapil t-PasifEngelenmisKolt k=<=0, Kumandapil t- aktifEndoskopYonlendirici k=<=6, Kumandapil t-EndoskopBolgesi k=<=0, Kumandapil t-Yesilisk ty=<=0, Kumandapil t-#au<=0, Yesilisk ty=<=12, Yesilisk ty-rt=<=0, Yesilisk ty-CerrahEndoskopSurme t=<=6, Yesilisk ty-PasifEngelenmisKolt k=<=0, Yesilisk ty-aktifEndoskopYonlendirici t=<=6, Yesilisk ty-EndoskopBolgesi t=<=0, Yesilisk ty-Kumandapil t=<=0, Yesilisk ty-#au<=12, #au-rt=<=0, #au-CerrahEndoskopSurme k=<=6, #au-PasifEngelenmisKolt k=<=0, #au-aktifEndoskopYonlendirici t=<=6, #au-EndoskopBolgesi t=<=0, #au-Kumandapil t=<=0, #au-Yesilisk ty=<=0 ameilyat_bolgesi_snrir=100 plidurume=2 cek=0 ent<=0 manuelkontrolbutonu=0 kumandabimmetikn=0 pasif_kol_serbest=0 merkezleme=0 endoskop_iceride=0 aktif_kol_merkezleme=1 </pre> | <pre> aktifEndoskopYonle ndirici_id11- >aktifEndoskopYonle endiinci.FRENLI [1, hazir, merkezleme := false} Yesilisk.YANIPSON UYOR- >Yesilisk.YANIVOR [1, hazir?, 1} </pre> | <pre> Fail State rt>=4 rt<=6 </pre> |

Figure B.5. Second Picture of Example 2 of Test Suite produced by VERTICALS

| | | |
|-------|--------------------|--|
| rt<=6 | Initial/Fail State | <p>(CerrahEndoskopKavrama.BASILMADI CerrahKontrol_id3 CerrahEndoskopSurme_id4 PasifdEngelenmisKol.FRENLI aktifEndoskopYonlendirici.ETKINDEGIL EndoskopBolgesi_id7 KumandaPil_id23 Kirmizilik.YANIP_SONUYYOR Yesillik.YANMIYYOR) rt<=6, rt-CerrahEndoskopSurme.t<=0, rt-PasifdEngelenmisKol.t<=0, rt-aktifEndoskopYonlendirici.t<=0, rt-EndoskopBolgesi.t<=0, rt-KumandaPil.t<=0, rt-Yesillik.ty<=0, rt-#tau<=0, CerrahEndoskopSurme.t<=6, CerrahEndoskopSurme.t-rt<=0, CerrahEndoskopSurmet-PasifdEngelenmisKol.t<=0, CerrahEndoskopSurme.t-aktifEndoskopYonlendirici.t<=0, CerrahEndoskopSurme.t-EndoskopBolgesi.t<=0, CerrahEndoskopSurme.t-KumandaPil.t<=0, CerrahEndoskopSurme.t-#tau<=0, PasifdEngelenmisKol.t<=6, PasifdEngelenmisKol.t-rt<=0, PasifdEngelenmisKol.t-CerrahEndoskopSurme.t-#tau<=0, CerrahEndoskopSurme.t-rt<=0, PasifdEngelenmisKol.t-aktifEndoskopYonlendirici.t-#tau<=0, EndoskopBolgesi.t<=6, EndoskopBolgesi.t-rt<=0, EndoskopBolgesi.t-CerrahEndoskopSurme.t<=0, EndoskopBolgesi.t-PasifdEngelenmisKol.t<=0, EndoskopBolgesi.t-KumandaPil.t<=0, EndoskopBolgesi.t-#tau<=6, KumandaPil.t-rt<=0, KumandaPil.t-CerrahEndoskopSurme.t<=0, KumandaPil.t-PasifdEngelenmisKol.t<=0, KumandaPil.t-aktifEndoskopYonlendirici.t<=0, KumandaPil.t-#tau<=0, Yesillik.ty<=6, Yesillik.ty-rt<=0, Yesillik.ty-CerrahEndoskopSurme.t<=0, Yesillik.ty-PasifdEngelenmisKol.t<=0, Yesillik.ty-aktifEndoskopYonlendirici.t<=0, Yesillik.ty-EndoskopBolgesi.t<=0, Yesillik.ty-KumandaPil.t<=0, Yesillik.ty-#tau<=6, #tau-rt<=0, #tau-CerrahEndoskopSurme.t<=0, #tau-PasifdEngelenmisKol.t<=0, #tau-aktifEndoskopYonlendirici.t<=0, #tau-EndoskopBolgesi.t<=0, #tau-KumandaPil.t<=0, #tau-Yesillik.ty<=0 ameliyat_bolgesi_siniri=100 pildurumu=2 cek=0 enc=0 manuekontrolbutonu=0 kumandabirimietkin=0 pasif_kol_serbest=0 merkezieme=0 endoskop_iceride=0 aktif_kol_merkezli=0</p> |
|-------|--------------------|--|

Figure B.6. First Picture of Example 3 of Test Suite produced by VERTICALS

APPENDIX C

EXAMPLES OF APRIORI AND PREFIX SPAN

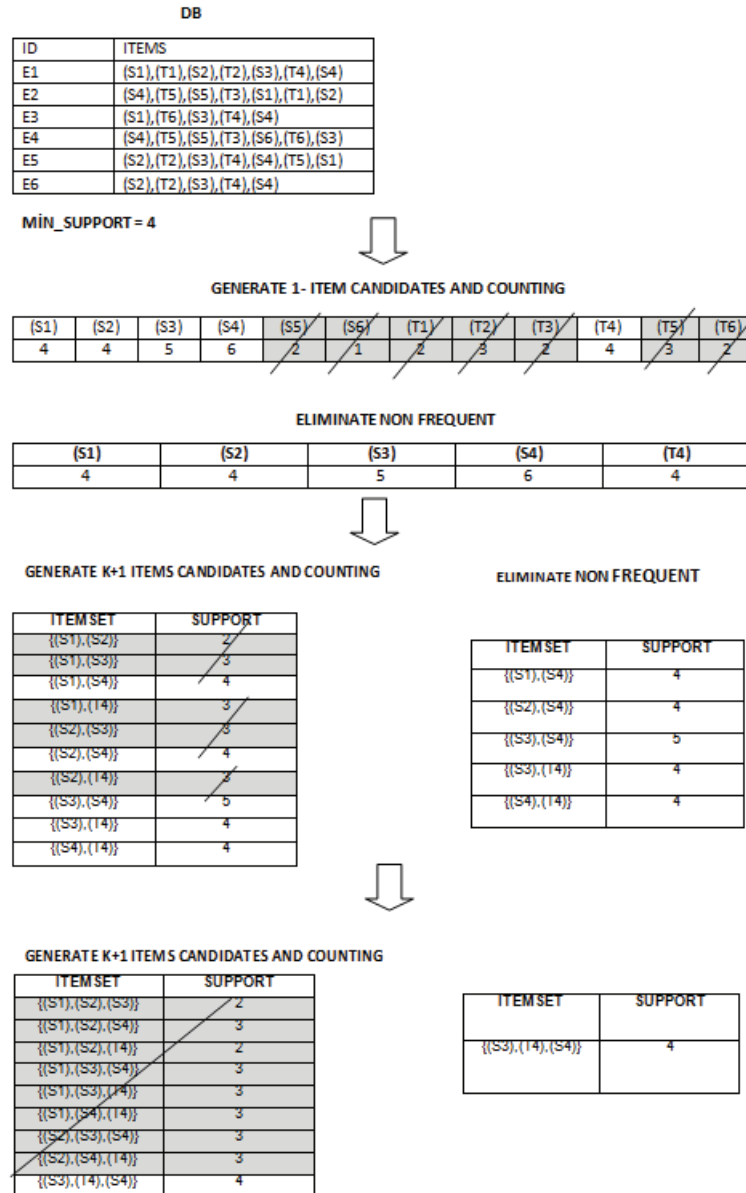


Figure C.1. Apriori Example

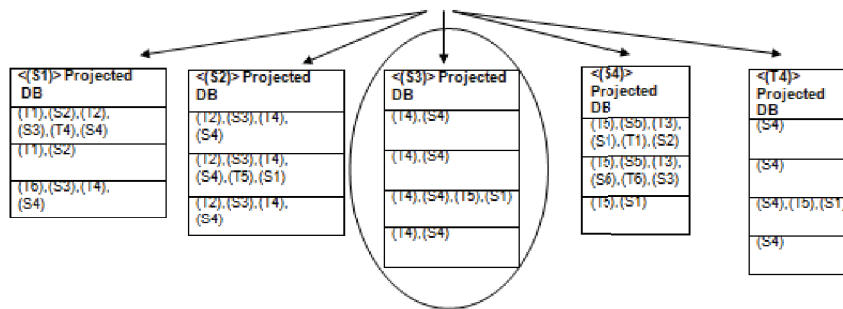
| DB | |
|----|------------------------------------|
| ID | SEQUENCE |
| E1 | (S1),(T1),(S2),(T2),(S3),(T4),(S4) |
| E2 | (S4),(T5),(S5),(T3),(S1),(T1),(S2) |
| E3 | (S1),(T6),(S3),(T4),(S4) |
| E4 | (S4),(T5),(S5),(T3),(S6),(T6),(S3) |
| E5 | (S2),(T2),(S3),(T4),(S4),(T5),(S1) |
| E6 | (S2),(T2),(S3),(T4),(S4) |

MIN_SUPPORT = 4

Find Length-1 Sequential Patterns

| (S1) | (S2) | (S3) | (S4) | (S5) | (S6) | (T1) | (T2) | (T3) | (T4) | (T5) | (T6) |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 4 | 4 | 5 | 6 | 2 | 1 | 2 | 3 | 2 | 4 | 3 | 2 |

PREFIX



Find Subsets of Sequential Patterns

| (S1) | (S4) | (T4) | (T5) |
|------|------|------|------|
| 1 | 4 | 1 | 1 |

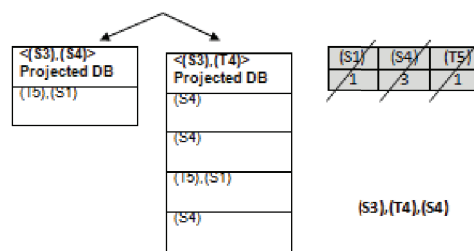


Figure C.2. Prefix Span Example