Proof of Ammunition, Acceptance, Reproof and

1.1 Rejection Criteria for Ammunition

1.1.1 By

1.1.2 Muhammad Hashim

Submitted to the Faculty of Computer Science,

National University of Science and Technology, Rawalpindi,

In partial fulfillment for the requirement of MS Degree.

MARCH 2000

TABLE OF CONTENTS

Chapter 1 **page**

Introduction

1.1 Introduction 1

1.2 Area of Research 1

1.3 Objectives 2

1.4 BACKGROUND 2

Chapter 2

Background

2.1 Introduction 5

2.2 Existing system 5

2.3 Flaws in the existing system 7

2.4 Why Developing Intelligent system for PARCS 8

2.5 Definitions 10

2.5.1 Ammunition 10

2.5.2 Types of small Arm Ammunition 13

2.6 Components 17

2.6.1 Criticality of components 18

2.6.2 Tests of components 19

2.6.3 Propellants 21

2.6.4 Proofing 22

2.6.5 External ballistics 24

2.6.6 Sentencing 25

2.7 Intelligent system 27

2.7.1 Definition 27

2.7.2 Difference between intelligent and

Non intelligent system 27

2.7.3 Components of Intelligent system 27

2.7.4 knowledge based expert sys 28

2.7.4.1 Experts Systems 29

2.7.4.2 Design of Rule Based Experts System 29

2.7.4.3 Pattern Matching 31

2.7.4.4 Selection of a Problem 32

2.7.4.5 Production system, rule based expert system 33

2.7.4.6 The knowledge Engineering process 37

2.8 Summary 38

Chapter 3

Conceptualization

3.1 Introduction 40

3.2 Analysis of modules of Existing system 41

3.3 The conceptual model of expert system 44

3.4 Programming in LISP 44

3.4.1 Conditional and predicates 45

3.4.2 Functions and lists 47

3.4.3 List as a recursive structure 48

3.4.4 Functional programming ,side effects 51

3.4.5 Data types in LISP 51

3.5 PARCS 52

3.5.1 Architecture 52

3.5.2 Data Object 56

3.5.3 Quantitative Data Object 56

3.6 Summary 56

Chapter 4

Implementation

4.1 Introduction 57

4.2 Source code of PARCS 57

4.2.1 Types of files 57

4.3 Advantages and limitations of Rules 59

4.3.1 Rules in PARCS 60

4.3.2 Filtration process 62

4.3.3 Data collection methodology 62

4.3.4 Working memory 63

4.4 Rule based inference mechanism of PARCS 64

4.5 User guide

4.5.1 Editing of data objects 79

4.6 Characteristics of expert system PARCS 79

4.6.1 Knowledge engineering tools 79

4.6.2 Security of data 80

4.6.3 Interface 82

4.7 Summary 82

Chapter 5

Conclusion

5.1 Introduction 84

5.1.1 New areas 84

5.2 Limitations of the system 85

5.3 Future research prospects 86

5.3.1 New computing architecture 86

5.3.2 New vision 87

5.3.3 Integration with relational database 87

5.3.4 Multimedia facility 88

5.4 Achievements 88

5.5 Project summary 89

5.6 Conclusion 91

Abstract

The aim of this research was to develop an intelligent system to carry out proofing of Ammunition (Acceptance, Rejection, and reproof system) which will be able to check the performance of Ammunition according to the specifications given by the manufacturer. Using Artificial intelligence (AI) techniques, the designing procedure has been made flexible so as to closely resemble the procedure followed by a human mind to design the same proofing system of Ammunition. The intelligent system developed in this project not only design the proofing procedures but also provides facility of complete old record to judge the performance of the particular Ammunition. The second part of the project facilitates the user to design the system himself and can also check the system so designed. This system has been developed in Franz Allegro LISP for windows version 3.02. Many features of Allegro LISP has been explored which includes:

- Rule Based Inference Mechanism.
- Pattern Matching.

The knowledge based of the system has been developed and production rules have been utilized for inferencing. The system is equipped with "graphical user interface (GUI). The rationalization of deeply rooted knowledge based is in such a way that existing Proof schedule Acception and Rejection computer system (PARCS) interface having Manus, Dialogue

boxes, widgets of various types, buttons, Menu-bars remains functional in the new environment. The existing Knowledge Representing (KR) formulas have been successfully implemented in the modified package. The overall response time of the system is reduced because of inbuilt features of LISP. It is hoped that this implementation will fulfill the need of user for the coming couple of years without any major modification and also provides a platform for the future expansion, like for example the software can be implemented at manufacturing stage in factories and also inspections carried out at different levels. Then LAN can link all the stages.

Acknowledgments

My thanks to….

Almighty Allah for his blessings upon me, and giving me the wisdom, knowledge and Understanding to accomplish this research work..

Dr. Tasnim. M. Shah my supervisor, for his guidance, encouragement, dedication, and with this research, without which I would not have been able to complete this thesis.

Major Akhtar Iqbal Major Fazal, Mr. Zarar Javed for serving on my guidance committee and providing me with their valuable suggestions and helpful comments about my project work.

Computer Science dept especially Lt. Col Wasim Suleman, My friends Major Inam Surri, Mohammed Umair khan for their moral support and encouragement.

My parents, my wife and children, for their prolonged patience, lively encouragement and continued prayers for the successful completion of this work.

Military College of Signals and National University of Sciences and Technology For building an institution of highly qualified instructional staff and provision of advanced research facilities

CHAPTER 1

Introduction

2.1 Introduction

This chapter introduces the proposed research area, the goal of thesis and its general background. The first section presents the area of research of the thesis. The second section covers the major goal of the thesis. The third section describe the background of the intelligent system for design and the last section presents the outline of the thesis.

2.2 Area of Research

The general area of research being addressed is, knowledge based approach to design the intelligent system. The system being developed is an intelligent system for designing the proofing of Ammunition (Proof schedule, Acceptance, Rejection, Computer system). This area requires a complete comprehension of the design procedures used for designing the proofing system. As the project belongs to Artificial intelligence (AI) area, the rule based expert system approach is employed to implement the design procedure. A rule based knowledge representation technique employing common LISP has been used for knowledge-based engineering. The inference mechanism is implemented with the help of another knowledge representation technique called pattern matching.

2.3 Objectives

The major goal of this research is to develop an intelligent system for proofing of Ammunition (Proof schedule Acceptance, Rejection Computer system). The system presents a user-friendly graphical user interface which helps the user to give the specifications of Ammunition to be discussed and the system at its own comes out with a complete solution of problem. The user now can run this system using the specifications and can check the performance of Ammunition. Following objectives have to be achieved.

- Carry out Proof Scheduling required for the decision of Ammunition.
- Decision about acceptance and rejection of Ammunition.
- Defects and Life of Ammunition.
- Implementation of the expert system for acceptance and rejection of Ammunition.

2.4 Background

Knowledge based system have been used to make a system to check the performance of Ammunitions at different stages. Which will reduce the design and implementation time of a proofing considerably like MYCIN, POEMS. The system PARCS (Proof schedule, Acceptance and Rejection computer system) carries out inspection at some stages using constraint propagation in Knowledge Representation (KR) which is a special feature of knowledge representation .Due to the facility of constant modification open to inspections all stages, heuristic in using knowledge to obtain solutions of expert systems, the utilization of this thesis for representing knowledge and carrying out the inspection/ performance results in a very logical solution. The thesis is divided into five chapters. The first chapter presents the general over view of the area of research, the goal of the thesis, which includes the major objective of research and background. The second chapter presents background that includes domain knowledge of procedures for Ammunition proofing its, types, propellants and different tests carried out, testing environment and procedures to check the performance of the Ammunition. It also gives a brief account of some previously developed system. The third chapter presents the conceptualization of the project. It depicts the thought process followed during the whole project. The first section describes the analysis of design procedure, conceptual modal of PARCS, concept of structuring the knowledge base, the inference mechanism and finally the main algorithm of the system. Next section describes the different modules of the problem to be used for the designed system. The summary of the chapter is given at the end. The fourth chapter describes the implementation strategy adopted during the development of the system. Firstly the source code of the system, architecture of the system, the graphical user interface (GUI) are explained. The next section describes how knowledge base has been structured. The inference mechanism using production rule and characteristics of the PARCS is also described in next section. The fifth chapter summarizes the entire thesis, by discussing the achievements made during the development of the system.

CHAPTER 2

Background

3.1 Introduction

In the present era of highly advanced and fast moving technology, computer has taken the key role in all spheres of life. It has not only simplified the highly complex management functions but also has infused efficiency, speed and accuracy to the level of utmost surprise. In the modern society, use of computer as a problem solver has been accepted as a rule instead of being an exception, may it be an aerospace complex, a medical concern or a tuition center for basic education. The engineering was one of the testing grounds for expert systems technology and has often been cited as one of the great break through in expert systems. Other examples of expert system include MYCIN, NURSExpert, CENTAUR, DIAGNOSER, MEDI and GUIDON, MEDICS and DiagFH to mention only a few [Davis 76].

3.2 Existing procedure

Ammunitions are composed of variety of components and importance of each one of them can not be overemphasized. For example a fuze, which is supposed to be safe in all respect once being handled by own troops, is required to function without fail when it lands in enemy area. HE, filled in the shells, should be insensitive to normal misbehavior by own troops but once reaches in the enemy terrain, should be lethal to provide appropriate fragmentation of shell. Similarly the propellant behind the projectile should be able to propell it to the just desired range, not more not less. All these parameters, an Ammunition is supposed to fulfill, and can not be achieved unless it is tested at different stages of production. The Ammunition which is declared fit through manufacturing Phase- I is distributed in lots each lot is tested for second phase through the selection of random sample After deploying the OP's survey party, weapon and equipment test fire is conducted for each sample Data of each fire is collected through survey party and OP's Behavior of each bullet of the sample is observed and reading is noted by the staff It is very important that each entry of each bullet of each sample is the deciding factors for the whole lots All the entries are made by the staff manually for each sample at the site

All the data collected by above sources is compiled by the proof officer/staff manually. A very complicated procedure is being adopted to calculate PE by using formula for each bullet of the sample. This all is manual. Proof papers for these samples are prepared. These are sent to

IDA’s (Inspection Depot Ammunition), which are loc at difficult plus, wah, Sanjwal, Gadwal etc.

The staff at these IDA’s enters the data records on different form’s comparing the records with the predetermined data by consulting a very large no of manual’s books, old records. Initially prepared results are forwarded for existing. In case of correct results, the lot is accepted. In case of incorrect results, lot is rejected, in case of deviation, the lot in reproofed. In same cases the reproofed lots are double reproofed.

3.3 Flaws in the existing system

The existing system was developed long time before. With the rapid changes in science/technology and changing requirements of inspection/tests it should have been modified, therefore the existing system have some flaws which are :-

- Chances of manipulation while selecting proof sample from a production lot.
- If in case this ammo have been already tested then a small mistake intentionally or unintentionally can pass or reject the ammo basing on previous results
- Application of firing data is prone to be faulty due to human error
- Working out standard deviation (SD) probable error (PE) is manual

- Calculation of PE is all manual mostly by the lower staff
- Filling of data in proof paper is manually and so enhances the chances of mistake
- Complete data collected by the survey party is not communicated to IDA’s
- Record keeping of a complex firing data requires deliberate efforts
- Finding and old record require a colossal efforts that too with the help of experience staff.
- Security of record remains doubtful.

3.4 Why Developing Intelligent System for PARCS?

As it is said “a friend in need is a friend in deed” is a saying, which is generally used for human friends, but it would not be unfair to say the same for Ammunition and Weapons. The most beloved friends of the soldier during heat of the battle are the Armaments. To ensure their trustworthiness during peace and war, they are required to go through different tests and inspections called “Proofing”.

Ammunitions are composed of variety of components and importance of each one of them cannot be overemphasized. For example a fuze, which is supposed to be safe in all respect once being handled by own troops, is required to function without fail when it lands in enemy area. HE, filled in the shells, should be insensitive to normal misbehavior by own troops but once reaches in the enemy terrain, should be lethal to provide appropriate fragmentation of shell. Similarly the propellant behind the projectile should be able to propell it to the just desired range, not more not less. All these parameters,an Ammunition is supposed to fulfill, and can not be achieved unless it is tested at different stages of production.

From the very earliest moments in the modern history of the computer, scientists have dreamed of creating an electronic brain. The scientists and Engineers alike were captivated by the potential; such a technology might have in industry. It was also thought that with the power of these machines, they would become perfect Engineers/experts in a box. Now systems are present, which allow the user to make final decision with full help from the system. FOX proposes a set of criteria for decision support systems [FOX 92]: -

- **Robustness**: A system's performance should not be dependent on ideal circumstances only, and should decline gracefully in non-ideal circumstances.
- **Flexibility**: System should be able to handle a number of tasks.

- **Accountability:** System should be able to explain each step, action or recommendations. This can be used for detecting faults in the knowledge base and help in debugging it.
- **Soundness**: System should have a well-developed and justified knowledge base. Different facts of the domain are properly incorporated and stored in the system knowledge base.

3.5 Definitions

3.5.1 Ammunition

Ammunition charged with explosive, propellant, pyrotechnic, initiating composition or a device, nuclear, biological or chemical material for use in connection with defense or defense including demolitions. It includes Ammunition used for training, ceremonial or non-operational purposes metal, or mild steel coated with gilding metal, which contains a lead nose and a copper tube, behind it, which contains the tracer composition, and its priming composition.

- **Arming**

  Changing the conditions of Ammunition from its safe storage and transportation mode to a state of readiness for initiation.

- **Augmenting cartridge**

  Propellant charge for a mortar bomb normally consists of several augmenting cartridges, any number of which may be removed to vary the charge weight. The primary cartridge initiates them.

- **Ballistic Cap**

  A cap fitted to the nose of projectile to improve its ballistic properties.

- **Ballistics**

  The science of the motion of projectiles acted upon by propellants, gravity and aerodynamic forces.

- **Barrel**

  The tube of a gun through which the projectile is fired.

- **Shot**

  A solid projectile fired from ordnance. It may include a discarding sabot.

- **Small Arms Ammunition (SAA)**

  Ammunition of caliber’s up to 20mm.

- **Projectile**

  An object capable of being propelled by a force, normally from a gun, and continuing in motion by virtue of its kinetic energy.

- **Proof**

  The evaluation of Ammunition performance by analysis of the results of monitored firing of randomly selected components or rounds.

- **Main Charge**

  Which is provided to accomplish the end result in ammunition, i.e. bursting a casing to produce blast and fragmentation, splitting a canister to dispense sub-missiles or producing The charge other effects for which it may be designed.

- **Mine**

  In land warfare, Ammunition which is designed to be detonated by the action of its target, by the passage of time, or by controlled means, and which is normally designed to damage or destroy vehicles or incapacitate personnel.

- **Misfire**

  Failure to fire or explode properly

- **Missile**

  An object, which is projected or propelled towards a selected point.

- **Mortar**

  A low pressure, smooth bore weapon, which transmits the recoil and firing to the ground through a base plate.

- **Warhead**

  That part of a missile, projectile, torpedo, rocket or other Ammunition which contains either the nuclear or thermonuclear system, high explosive system, chemical or biological agents to inflict damage.

- **Yaw**

  The angle between the longitudinal axis of a projectile at any moment and the tangent to the trajectory in the corresponding point of flight of the projectile.

3.5.2 Types of Small Arms Ammunition

- **Service Types (Ball)**

  This is the normal service bullet for general use. The bullet may be solid lead; with or without a light tip, and cased in cupro-nickel or gilding metal; or a mild steel core in an envelope.

- **Armour Piercing**

  For use against soft-skinned and lightly armoured vehicles, and other targets which would defeat ball ammunition, the bullet consists, in small caliber rounds, of a mild steel envelope containing a hardened steel core, and a combined tip and sleeve.

- **Tracer**

  A tracer round is so called because it leaves a visible trace, so that the trajectory can be seen, to enable the aim to be corrected. In the smaller caliber's, the bullet is made with an envelope of cupro-nickel, gliding metal, or mild steel coated with gilding metal, which contains a lead nose and a copper tube, behind it, which contains the tracer composition, and its priming composition.

- **Incendiary**

  Incendiary bullets are used against targets, such as petrol tanks, which are inflammable or explosive. They are based either on ball bullets or on armour piercing bullets.

- **Rifle Grenade Cartridge**

  Made similarly as the blank round. The mouth of the American manufactured cartridge is crimped, but, in the case of British manufactured cartridges a shellacked wad closes the mouth. It is used for discharge a grenade from a rifle.

- **Proof Ammunition**

  The bullet is as for ball Ammunition, but the cartridge has an adjusted charge in order to obtain a higher chamber pressure for the proof of new weapons. This is done either by filling to a higher charge weight than used for service rounds, or by filling with a“hotter”propellant. They are sometimes known as “overcharge”, “high pressure” or “test” rounds. Proof Ammunition is not issued to units.

- **Standard Ammunition**

  During the proof of the small arm Ammunition (SAA) a number of rounds from each lot of Ammunition are fired for velocity and pressure. Both the velocity and the pressure obtained with any cartridge vary with atmospheric conditions, the state of wear in the barrel and breech mechanism, and other conditions. In order to obtain consistent results, it is necessary to eliminate these variations. This is done by firing, both for velocity and for pressure, in comparison with Ammunition of known ballistics which is kept for comparative purposes, and which is known as “standard Ammunition”.

- **Practice Rounds**

  These may be made up of ball Ammunition or of a soft metal bullet, with a reduced charge or they may be specially manufactured, cheap rounds. Practice Ammunition manufactured as such should not be confused with service ammunition, which has been related to practice on account of age or defects.

- **Drill Cartridge**

  This is a non-explosive round used for instructional purposes in loading and unloading weapons. It consists of a case, with an empty cap chamber and either a rejected ball bullet, secured by two canellures, with a wooded distances a piece to support it.

- **Dummy or Inspection round**

  This is also non-explosive round, but it is carefully manufactured in weight, shape and balance to represent the counterpart live Ammunition. Armourers and gun makers for gauging purposes use it. It consists of a white metal case with an empty cap chamber. There are no flash holes or anvil.

- **Blank Cartridges**

  Used in training for simulating rifle or machine gun fire, and for firing salutes.

3.6 Components

- **Cartridge Case**

  The brass cartridge case is manufactured in one piece by several drawing and machining operations, which give it the final shape and dimensions specified.

- **Cap Chamber**

  There are several firing systems in use. In the British system, a cap chamber is formed in the base of the cartridge case, and is connected to the interior by two or three fire holes. In the center an anvil is formed integral with the base, on which the cap composition is crushed by the blow of the striker.

- **Primers**

  The primer is one of the most important components of a complete cartridge. It is also the most dangerous. If a primer misfires, the cartridge is useless. In the next instant your enemy may kill you, or your game escape, or you may lose the shooting match. The primer cup must be soft enough to be easily indented by the firing pin without puncturing, but at the same time it must hard enough to maintain its seating in the cartridge case and to resist rupture or blow back under the pressure of the propellant gases.

3.6.1 Criticality of Components

Smallest components, in simple Ammunition, go through many critical tests in order to ensure that when it reaches own troops, it possesses all the basic requirement of Ammunition. The complete list of tests is quite lengthy, however some much important tests are explained here, safety, function, durability, etc

An axial compressive stress, due to the setback of the material of the shell and filling on firing, A radial compressive stress due to the pressure developed in the filling on set back, on the assumption that the filling behaves as a liquid, A circumferential tensile stress, due to this radial compressive stress. A further circumferencial tensile stress, due to centrifugal force, caused by the spinning of the shell. A torsional stress, about a longitudinal axis, due to the angular acceleration of the shell. In the design, the problem is to combine all these stresses into a single equivalent stress and thus to determine the yield or proof stress of the steel to be used for the walls. There are some other tests through which the Ammunition is passed.

3.6.2 Tests of Components

- **Water Tightness**

  The watertightness test is carried out to ensure that water will not penetrate inside the case, and damage the propellant or cap composition.

- **Break-up**

  The object of the break-up test is to examine in detail the components of a small proportion of the rounds produced, so as to ensure that they comply with the specification. The majority of components are examined before being assembled, but one round is broken up, the complete round is weighted, and examined for visible case and cap defects.

- **Hardness**

  In order to keep a check on the physical properties of the materials used in the components of the round-i.e. the brass of the cartridge case and the steel of A.P. cores-five rounds from each date of work are tested for hardness. A Vickers diamond hardness-testing machine is used. The hardness of brass cases is measured using the 5-kg. Load.

- **The mercurous nitrate test**

  This test is carried out to find if there are any residual stresses in the brass of cartridge cases which are liable to cause early “season cracking” in the cases. Five rounds from each date of work are subjected to this test.

- **The drop test**

  Detonation proof and sealing proof are carried out on lots of fuzes from normal production runs. During the design stage, and for the first lot of a new run of production the tests described here are carried out, to ensure, in the first case, that the fuze is satisfactory, and in the second case, that the methods of manufacturing to be used give the results which can be expected from the design of the fuze.

- **The side slap test**

  When guns and mortars become badly worn, there is a tendency for projectiles to “slap” the “sides” of the bore as they travel up it. This phenomenon is known as side slap, and it may introduce great lateral accelerations into the fuze.

- **The jolting test**

  This test, carried out at the design stage, is aimed at ensuring that the fuze, as designed, will be safe to handle and will be in a fit condition to function satisfactory, after being subjected to the worst shocks it is likely to encounter during transport over rough ground.

- **Spinning test**

  The shutters of fuzes, which are armed by centrifugal force, are tested during manufacture to ensure that the specified rate of spin does, in fact, open the shutter. The shutter is placed in a suitable machine, provided with a tachometer, and is spun. The arming of the shutter is seen, and the rate of spin at which this occurred is noted.

3.6.3 PROPELLANTS

Until the end of the 19th century, and the introduction of nitrocellulose as a propellant, gunpowder was used universally as the propellant for all small arms Ammunition, since it was, at that time, the only known explosive.

- **The double based propellants**

  Which are used in SAA are sometimes called NG powders, then from the more common NC powders, the name given to single based SAA propellants. Propellant Mk. 1, originally known as Cordite, Mk. 1, is used for some purposes in small arms cartridges. Thus, proof rounds, in which a surplus of chemical energy is required to be loaded into the normal cartridge, sometimes use this very “hot” propellant.

- **Single Base Propellant**

  Ballistite A, one of the earliest “smokeless powders”, has similar properties to the double based propellants, but is rather easier to ignite, and burns faster and more regularly at low pressures. It is also highly corrosive, and not very stable, especially in hot climates. It is therefore only used nowadays, where no other propellant would be suitable, such as in mortar secondary and rifle grenade cartridges, where the low pressure preclude the use of double based types.

- **Propellant Shapes**

  The shapes most commonly used are short tubes, used mainly in rifle and machine gun cartridges, spherical grains, and thin flakes, used mainly in pistol and machine carbine rounds.

3.6.4 Proofing

To ensure that the Ammunition when reaches in the hands of troops is reliable, free of any defect and is safe, the inspection of Ammunition is carried out at all stages. The conditions of production i.e. dimension, processes and functions are laid out by the designer. The manufacturer contracts to provide the user with Ammunition of laid down quality. In the final terms of delivery the user carries out the tests/inspection of Ammunition as per laid down conditions.

- **Proof Samples**

  As with all Ammunition components, a method of proof by statistical sampling has to be used, since proof is essentially a destructive test. Further, since the sample, which would be necessary to achieve sentencing of individual lots would be uneconomic, a deferred method of sentencing is adopted, which reduces the sample per lot, without holding up proof sentencing so long as the results are satisfactory. The drawback of this system of course, is that, when a defect occurs in one lot, the sentencing of subsequent lots is automatically delayed.

- **Material Inspection**

  The producer is required to satisfy the inspector that specified materials are used for different types of components. The inspector may accept the certificate of the producer or arrange to get them tested under his own arrangements. Only specified materials are allowed to be used.

- **Dimension Checks**

  All components are checked as per respective drawings. The Inspector specially checks functional dimensions. Following are the different components, which are checked. Only correct components/stores are allowed to be accepted, Cap, Bullet, Case, Cap filled, complete round.

- **Processes Checks**

  Different components are required to have different surface/body treatment, which are checked during the inspection.

3.6.5 External Ballistics

Once the projectile has left the gun and the influence of the emerging gases, the part of the flight known as external ballistics begins. There are a number of factors, which affect the motion of projectile; some associated with the projectile itself and others with the atmosphere through which the projectile is moving. The properties of the projectile, which enter into the problem, are its mass, caliber, shape and axial spin rate. The relevant properties of the atmosphere are air density, temperature, static pressure, viscosity and wind speed and direction. The effects of these are made manifest through the projectile properties introduced above.

These notes will deal first of all with the atmosphere and its properties, followed by a conceptual introduction to the important properties of the air from which it is composed. The influence of these properties on the flight of a projectile will then be dealt with in some detail. Air pressure, temperature, density and viscosity all vary with altitude. These changes in the physical properties of the atmosphere effect the resistance of the air to the passage of a projectile and hence its range. Since the trajectories of small arm Ammunition, Artillery projectiles and ballistic missiles normally have peak altitudes of 50m, 20km,and 600km respectively. Projectiles in these different roles will experience significantly different environments in flight. For example, the density of air decreases the height above sea level increases and consequently the range of the projectile increases as the trajectory gets higher and higher. This is exploited by inter-continental ballistic missiles, which travel for most of their flight in the upper atmosphere where there is practically no air resistance at all. Atmospheric conditions vary from place to place and from time time. For comparative performance assessment standard atmosphere is required.

Essentially, there are three separate contributions to the drag force on an object moving through the atmosphere, fluid medium due to which the motion of the bullet is restricted. Pressure DragYaw-dependent drag

3.6.6 Sentencing

The final sentencing of Ammunition and Ammunition components is the responsibility of Chief Inspector of Ammunitions. While sentencing, the sentencing officer must abide by the rules laid down in the relevant sentencing schedule. Any failure to do so is a breach of contract, either with the user, if he accepts Ammunition, which is sub-standard, or with the manufacture, if he rejects Ammunition which is upto the required standard.

It is, therefore, vitally important that the sentencing officer should thoroughly understand the various proof schedules, and should consult a higher authority if ever he is in any doubt as to their application in a particular case. On completion of the proof firing, copies of the proof papers, on which all relevant information has been entered at the proof range, are sent to the sentencing officer, who enters on the papers the sentence he awards. Once a lot of any Ammunition/fuzes has been sentenced as serviceable, as a result of proof, it is issued, in the case of an empty lot, to the filling factory, and, in the case of filled lot, to an Ordnance Depot for issue to units, or to an assembly line, for insertion into a shell or bomb. A lot, which has been sentenced to rejection, on the other hand, will never be issued, but will be destroyed, or, if possible, rectified. The material of a rejected lot, however, remains the property of the manufacturer, and great care must be taken to ensure that no rejected lot of components is assembled to otherwise serviceable Ammunition.

There are four sentences that can be passed namely Serviceable, Reproof, Double Reproof and Rejected. The first implies that the lot or series may go forward to the service in the case of filled stores, or to the net stage of manufacture. Stores, which are to reproof and double reproof, means that the procedure for proof must be repeated. And new sample will be selected and fired before final sentence can be passed.

3.7 Intelligent System

3.7.1 Definition

It is an intelligent computer program that uses knowledge and inference mechanism to solve problems that are difficult enough to require a significant human expertise for their solution.

3.7.2 Difference between the Intelligent and Non-Intelligent System

A standard computer program can only solve problems for which it is specifically designed. This is not only time consuming, but other parts of the program may be adversely affected in the process and errors may result. Artificial Intelligence (AI) as its name implies, really enable a computer to think. By simplifying the way programs are put together,(AI)Artificial Intelligence imitates the basic human learning process by which new information is absorbed and made available for future reference. The human mind can incorporate new knowledge without changing the way the mind works or disturbing all other facts that is already stored in the brain, an AI program is very much the same way.

3.7.3 Components of Intelligent System

The intelligent system concept is divided into:-

- Knowledge base
- Inference mechanism

The knowledge base is unique to a particular domain. Which is some times called the working memory or temporary data storage, contains "declarative knowledge about the particular problem being solved and the current state of affairs in the attempt to solve the problem". There are several ways to represent this data like first order predicate logic.

Inferencing is a process by which new facts are derived from the known or assumed facts. When we reach at a goal, we are not only solving an immediate problem but are also acquiring new knowledge at the same time.

3.7.4 Knowledge Base.

The first principle of knowledge engineering is that the problem solving power exhibited by an intelligent agent's performance is primarily the consequence of its knowledge base, and only secondarily a consequence of the inference method employed. Expert systems must be knowledge-rich even if they are methods-poor. AI has focused its attentions almost exclusively on the development of clever inference method almost any inference method will do. The power resides in the knowledge.

3.7.4.1 Expert System

An expert system is a knowledge-based program that provides “expert quality” solutions to problems in a specific domain. Generally, its knowledge is extracted from human experts in the domain and it attempts to emulate their methodology and performance. As with skilled humans, expert systems tend to be specialists, focusing on a narrow set of problems. Also, like humans, their knowledge is both theoretical and practical, having been perfected through experience in the domain. Unlike a human being, however, current programs cannot learn from their own experience; their knowledge must be extracted from humans and encoded in a formal language. Open to inspection, both in presenting intermediate steps and in answering questions about the solution process. Easily modified, both in adding and in deleting skills from the knowledge base. Heuristic, in using knowledge to obtain solutions.

3.7.4.2 Design of Rule Based Expert Systems.

The most important modules shown in figure. 2.1 that makes up a rule-based expert system. The user interacts with the expert system through a user interface that makes access more comfortable for the human and hides much of the system complexity. Expert system employs a variety of interface styles, including question-and-answer, menu-driven, natural language, or graphics interfaces.

The program must keep track of case specific data, the facts, conclusions, and other relevant information of the case under consideration. This includes the data given in a problem instance, partial conclusions, confidence measures of conclusions, and dead ends in the search process. This information is separate from the general knowledge base.

The explanation subsystem allows the program to explain its reasoning to the user. These explanations include justifications for the system's conclusions.

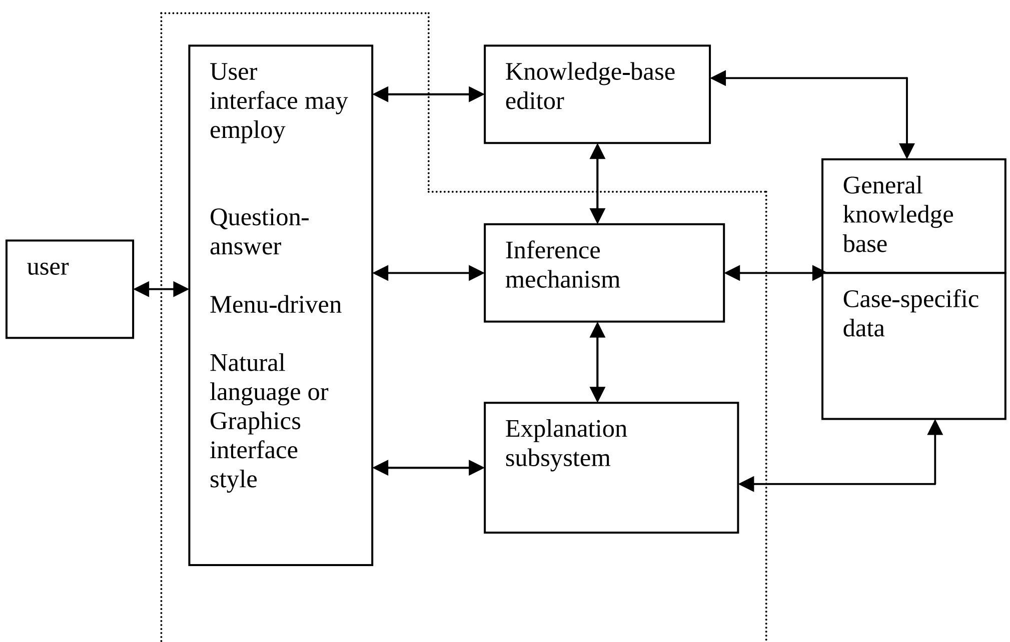


Fig:2.1 Architecture of a typical Expert system.

Many systems also include a knowledge-base editor, Knowledge-base editors can access the explanation subsystem and help the programmer locate bugs in the program's performance. The heart of the expert system is the general knowledge base, which contains the problem-solving knowledge of the particular applications in a rule based expert system This knowledge is represented in the form of if then rule. The inference mechanism applies the knowledge to the solution of actual problems. It is the interpreter for the knowledge base. In the production system, the inference engine performs the recognize-act control cycle.

Expert systems are built by progressive approximations, with the program's mistakes leading to corrections or additions to the knowledge base. In a sense, the knowledge base is "grown" rather than constructed

3.7.4.3 Pattern Matching

The heart of this retrieval system is a function called match, which takes as arguments two s-expressions and returns if the expressions match. Matching requires that both expressions have the same structure, as well as having identical atoms in corresponding positions. In addition, match allows the inclusion of variables, in s-expressions. Variables are allowed to match with any s-expression, either a list or an atom, but do not save binding, as with full unification.

Match is used to define a function called get-matches, which takes as arguments two s-expressions. The first argument is a pattern to be matched against elements of the second-s-expression, which must be a list, get-matches returns a list of the elements of the list that match the first argument. The heart of the system is the match function, a predicate that determines whether or not two s-expressions containing variables actually match. Match is based on the idea that two lists match if and only if their respective cars and Cdr. This suggests a car-Cdr. recursive scheme for the algorithm. The recursion terminates when either of the arguments is atomic. If both patterns are the same atom or of the patterns is a variable atom, which can match with anything, then termination is with a successful match: otherwise, the match will fail. Notice that if either of the patterns is a variable, the other pattern may or may not be atomic variables may match with s-expressions of arbitrary complexity.

3.7.4.4 Selecting a Problem for Expert System Development

Expert systems tend to involve a considerable investment in money and human effort. Attempts to solve a problem that is too complex, poorly understood, or otherwise unsuited to the available technology, can lead to costly and embarrassing failures. Researchers have developed an informal set of guidelines for determining whether a problem is appropriate for expert system solution. The need for the solution justifies the cost and effort of building an expert system. Human expertise is not available in all situations where it is needed. The problem domain is well structured and does not require common sense reasoning.

The problem may not be solved using traditional computing methods. Cooperative and articulate experts exist. The problem is of proper size and scope.

3.7.4.5 Production Systems, Rules, and the Expert System Architecture

The architecture of rule-based expert systems may be understood in terms of the production system model for problem solving. In fact, the parallel between the two entities is more than a simple analogy; the production system was the intellectual precursor of modern expert system architecture. This is not surprising when Newell and Simon began developing the production system model; their goal was to find a way to model human problem solving.

Let consider the expert system architecture in Figure 2.2, 2.3, 2.4 given as a production system, the knowledge based is the set of production rules. The expertise of the problem area is represented by the production. In a rule-based system, these condition-action pairs are represented as rules, with the premises of the rules (the if portion) corresponding to the condition. Case specific data are kept in the working memory. Finally, the inference engine is the recognize-act cycle of the production system. This control may be either data driven or goal driven. In a goal driven expert system the goal expression is initially placed in working memory. The system matches the rule conclusions with the goals, selecting one rule and placing its premises in the working memory.

As a more detailed example of goal driven problem solving, we create a small expert system for diagnosing automotive problems:

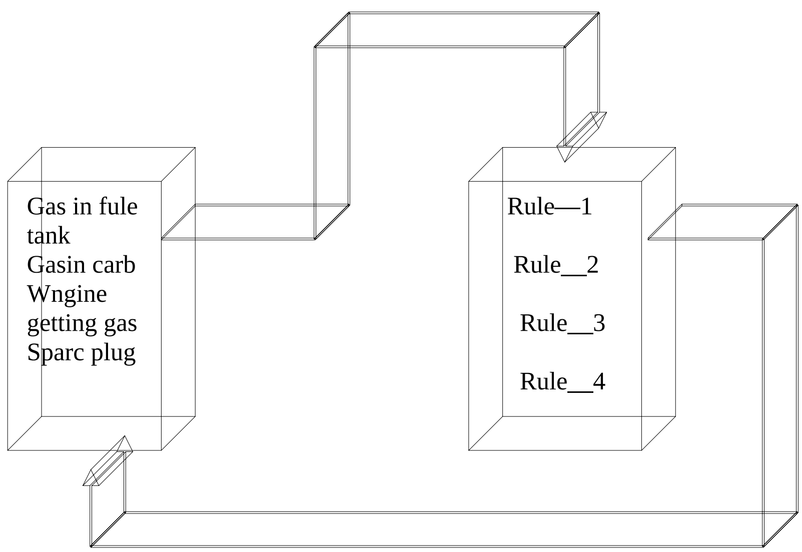


Fig:2.2The production system after Rule-4 has fired.

Rule 1: if

The engine is getting gas,

and the engine will turn over,

then

the problem is spark plugs,

working memory Rule

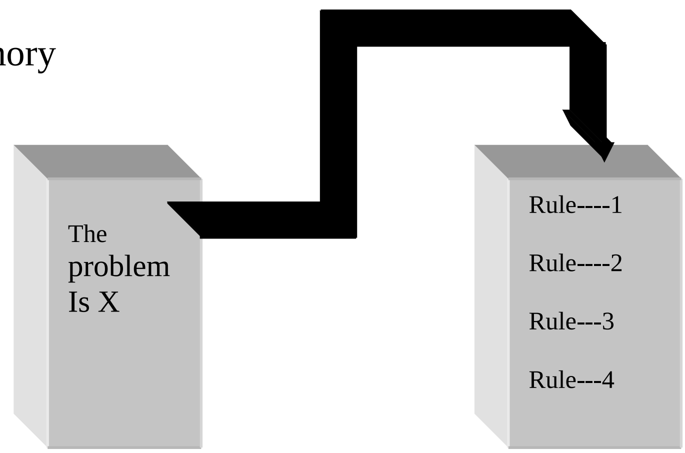


Fig: 2.3 The production system at the start of consultation in the car diagnostic example.

Working memory Rule

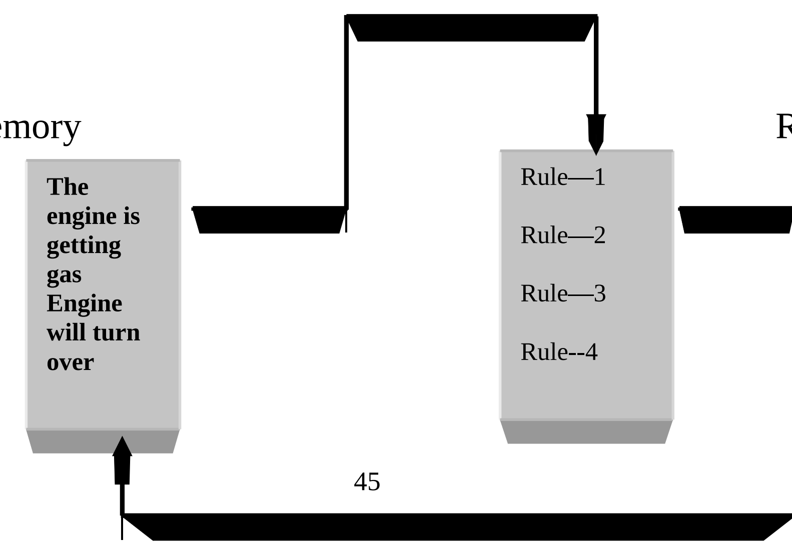


Fig 2.4

After Rule

1 has fired

Rule 2 ; if

The engine does not turn over, and

the lights do not come on

then

the problem is battery or cables.

Rule 3 : if

The engine does not turn over, and

the lights do come on

then

the problem is the starter motor.

Rule 4 : if

There is gas in the fuel tank, and

there is gas in the carburetor

then

the engine is getting gas.

To run this knowledge base under goal-directed control regime, place the top-level goal, the problem is X. in working memory as in Figure X is a variable that can match with any phrase; it will become bound to the solution when the problem is solved.

There rules match with the expression in working memory; rule 1, rule 2, and rule 3. If we resolve conflicts in favor of the lowest-numbered rule, then rule 1 will fire. The cause X to be bound to the value spark plugs and the premises of rule 1 to be placed in the working memory as the system has thus consigned to explore the possible hypothesis that the spark plugs are bad.Note that there are two premises to rule 1, both of which must be satisfied to prove the conclusion true. These are, and, branches of the search graph representing a decomposition of the problem into two sub-problems. Then rule 4is fired, whose conclusion matches with "engine is getting gas, ": causing its premises to be placed in working memory as in Figure At this point, there are three entries in working memory that do no match with any rule conclusions. Our expert system wills equerry the user directly about these sub-goals. If the user confirms all three of these as true, the expert system will have successfully determined that the car will not start because the spark plugs are bad.

3.7.4.6 The knowledge Engineering Process

The primary people involved in building an expert system are the knowledge engineer, the domain expert, and the end user. The knowledge engineer is the (AI) Artificial Intelligence language and representation expert. His or her main task is to select the software and hardware tools for the project, help extract the necessary knowledge from the domain expert, and implement that knowledge in a correct and efficient knowledge base. The knowledge engineer may initially be ignorant of the application domain.

The domain expert provides the knowledge of the problem area. The domain expert is generally someone who has worked in the domain area and understands its problem-solving techniques, such as using shortcuts, handling imprecise data, evaluating partial solutions, and all the other skills to the knowledge engineer. As in most applications, the end user determines the major design constraints. Unless the user is happy, the development effort is by and large wasted. The skills and needs of the user must be considered throughout the design cycle.Expert systems are built by progressive approximations with the program mistakes leading to corrections or additions to the knowledge base.

3.8 Summary

This chapter gives an overview of an expert system, explaining criteria for developing Proof Schedule Acceptance and Rejection Computer system (PARCS).In the first section the criteria of developing the Expert system for PARCS have been described .In next section expert system ,the background of the Ammunition ,parts of Ammunitions, its fillings external agents and different tests have been discussed. The expert system PARCS in view of the AI (Artificial intelligence) using production rules and inference mechanism have been discussed.This chapter also covers the Proofing diagnostic pakage.The chapter is summarized at the end.

CHAPTER 3

Conceptualization

4.1 Introduction

A conceptual design of an expert system PARCS is similar to an architectural sketch of a building. It gives us a general idea of how the system will look like and how it is going to solve the problem. The design shows the general capabilities of the system, the required resources, and any other information that is necessary for detailed design.

Computer aided design tools are very widely being utilized in the different fields of engineering. These tools allow users before using analysis programs, to estimate different solutions, which corresponds to the user's specifications. It is also based on the parallel search of solutions to respect the philosophy of design, as seen by the expert. To take into account the development of manufacturing technologies and constraints (like economic criteria), design tool is easy to maintain and to adapt the future requirements of design. In this chapter the conceptual model of the PARCS will be discussed as conceived on the conclusion of the thought process.

In the first section the existing proofing procedures has been discussed. The proposed model of the system is discussed in the next section. This section also describes the graphical user interface, concept of knowledge base structuring the inference mechanism using rules. Next section describes design strategy adopted to design the Ammunition proofing criteria.

4.2 Analysis of Modules of Existing system

Designing the system which carries the proofing of Ammunition is basically the collection of information's, specifications and readings from different stages, which are connected in some particular fashion to perform a specified task. There are mainly three major stages of the performing the proofing of the Ammunition, which are:

- Phase-1 (Manufacturing stage)
- Phase-2 (Firing stage)
- Phase-3 (Sentencing stage)

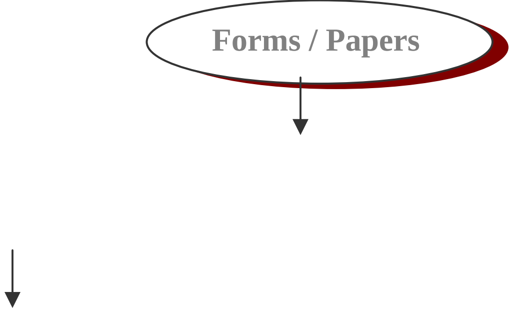
The major difference of between the three stages is very clear. If we try to simplify the design procedure, we end up with a summary as follows:-

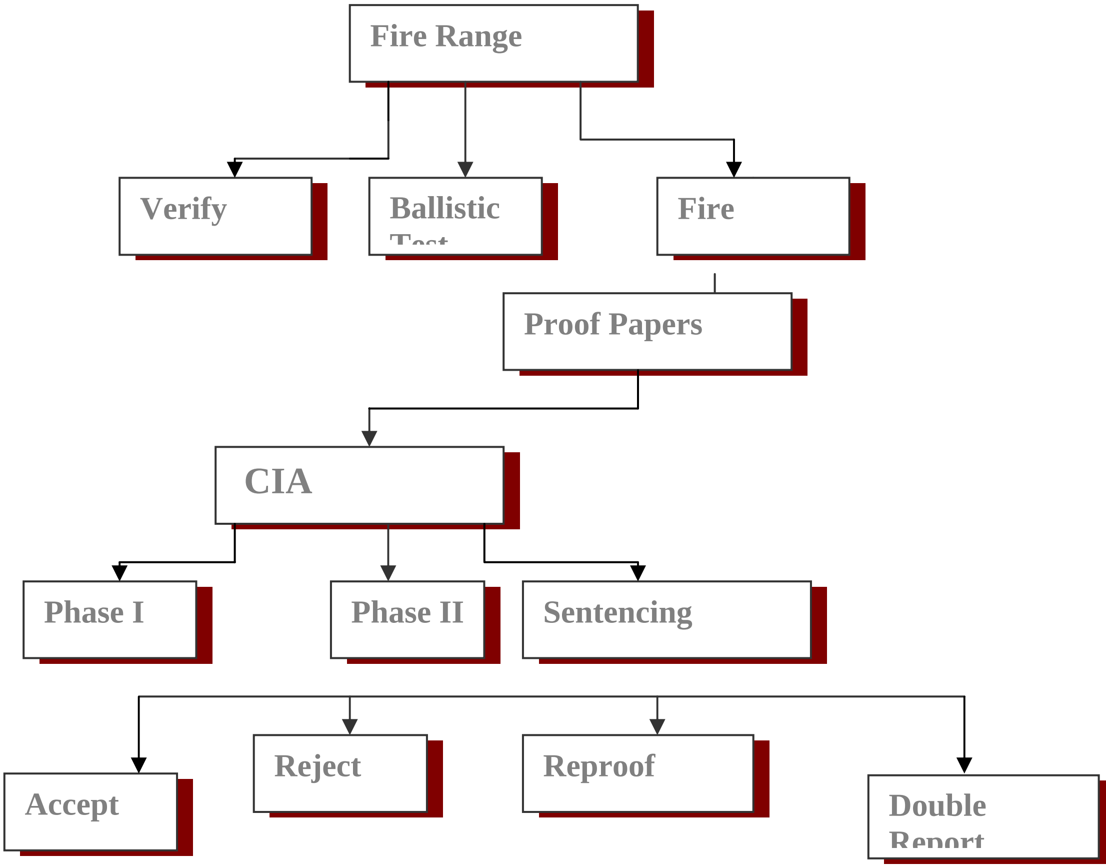
Manufacturing stage :In this stage the information's are collected from the manufacturing stage and from the testing of each component/part/material of the Ammunition. When these initial observations are ready, propellant is filled in cases. Now after getting all the information’s from each stage of manufacturing of small components of empty cases, these are linked to get a decision about its correctness. These cases are then filled with propellants. The Ammunition is now ready for fire. The sampling and loting is carried out in order to carry out the fire by keeping in view the external conditions (atmospheric).

Fire Observation: At this stage the Ammunition is tested to check its battle worthiness, which is the core issue. Fire is conducted on the range. Proof officer will collect the necessary information’s from all the source points like i.e. environmental conditions (temperature, wind direction and pressure etc) Armaments behavior (muzzle velocity etc) correctness of the fire, in some cases PE (probable error) formula is also used to see the fire behavior.

Sentencing stage: All the above information is linked to see the overall behavior of the Ammunition. All the data is collected, evaluated and finally the tested Ammunition in sentenced.

All the collected data is sent to the sentencing authority for sentencing of the checked Ammunition..


Forms / Papers


Fire Range
Verify
Ballistic
Fire
Proof Papers
CIA
Phase I
Phase II
Sentencing
Accept
Reject
Reproof
Double

4.3 The Conceptual Model of Expert System PARCS

The conceptual model of this project has following components of the model in the expert system PARCS.

- Knowledge Base Representation
- Knowledge Acquisition Subsystem
- Inference mechanism.
- User Interface

There are various schemes for representing knowledge in a computer, however keeping in view the disadvantages of multiple knowledge representation; it has been decided to use production rules for representing the knowledge. Production rules will be used for inference mechanism of the expert system PARCS.

4.4 Programming in LISP: Creating New Functions

All modern LISP dialects support a large number of built-in functions, including: A full range of arithmetic functions, supporting both integers and real numbers, A variety of looping and program control functions, List manipulation and other data structuring functions, Input/output functions, Forms for the control of function evaluation, Functions for the control of the environment and operating system. LISP includes too many functions to list. In LISP,the program is made by defining new functions in the LISP environment, constructing programs from this already rich repertoire of built-in functions. These new functions are defined in terms of existing functions using defun, which is short for define function. Once a function is defined it may be used in the same fashion as functions that are built into the language. Suppose, for example, the user would like to define a function called square that takes a single argument and returns the square of that argument. Square may be created by having LISP evaluate the following expression;

```
(defun square(x)
(* xx))
```

The first argument to defun is the name of the function being defined; the second is a list of the formal parameters for that function. Which must all be symbolic atoms, the remaining arguments are zero or more s-expressions, which constitute the body of the new function. A newly defined function may be used just like any built-in function.

4.4.1 Conditionals and Predicates

LISP branching is based on function evaluation; control functions perform tests and, depending on the results, selectively evaluate alternative forms. Consider, for example, the following definition of the absolute value function using cond (note that LISP has a built in function, abs, that computes absolute value):

```
(defun absolute-value(x)
(cond((<x0  (-x))
((>=0)  x)))
```

cond takes as arguments a number of condition-action pairs.

These pairs may be arbitrary s-expressions, and each pair is enclosed in parenthesis. Like defun, cond does not evaluate all of its arguments. Instead, it evaluates the conditions in order until one of them returns a non-nil (for “true”) value. When this occurs, the associated action is evaluated and this result is returned as the value of the cond expression. None of the other actions and none of the subsequent conditions are evaluated. If all of the conditions evaluate to nil (for “false”), the cond returns nil.

```
(define absolute-value(x)
(cond ((< x O) (- X))
(t    x)))        if X is less than o, retun-x
;otherwise, retun x unchanged
```

Although any valuable s-expressions may be used as the conditions of a cond, generally these are a particular kind of LISP function called a predicate. A predicate is simply a function that returns a value of either true of false depending on whether or not its arguments possess some property.

4.4.2 Functions, Lists

Although the preceding sections introduced in LISP syntax and demonstrated a few useful LISP functions, they did so in the context of simple arithmetic examples. The real power of LISP, however, is in symbolic computing and is based on the use of lists to construct arbitrarily complex data structures of symbolic and numeric atoms, along with the forms needed for manipulating them. As well as the naturalness of data abstraction techniques in LISP, with a simple data base example.

Generally, AI programs use large amounts of varied knowledge about problem domains. The data structures used to represent this knowledge, such as semantic networks, and humans generally find it easier to relate to this knowledge in terms of its meaning rather than the particular syntax of its internal representation. Therefore, data abstraction techniques, always-good computer science, are essential tools for the AI programmer, because of the case with which LISP supports the definition of new functions; it is an ideal language for data abstraction.

4.4.3 Lists as Recursive Structures

Uptill now list has been used to implement access functions for records in a simple database. Because all records were of a determinate length, these functions were sufficient to access the fields of records. However, these functions are not adequate for performing operations on lists of unknown length, such as searching through an unspecified number of employee records. To do this, list is scanned iteratively or recursively, terminating when certain conditions are met (e.g. the desired record is found) or the lists exhausted. Although LISP includes a number of constructs for doing explicit iteration, the fundamentally recursive structure of lists makes recursion a natural vehicle for list manipulation. The basic functions for accessing the components of lists are car and Cdr. Car takes a single argument, which must be a list, and returns the first element of that list. Cdr also takes a single argument, which must be list, and returns that list with its first argument removed

(Car ‘(a b c))

a

(cdr “(a b c))

(b c)

(Car ‘((a b) (c d)))

((c d))

(car (cdr ‘(a b c d)))

b

the way in which car and cdr operate suggests a recursive approach to manipulating list structures:

To perform an operation on each of the elements of a list:

- If the list is empty, quit.
- Perform the operation on the first element of the list and recur on the remainder of the list.

In addition to the functions car and cdr, LISP provides a number of functions for constructing lists. One of these, list, which takes as arguments any number of s-expressions, evaluates them, and returns a list of the results. Cons takes two s-expressions as arguments, evaluates them, and returns a list whose car is the value of the first argument and whose cdr is the value of the second.

(Cons 1’(2 3 4))

(1 2 3 4)

(Cons ‘(a b)’ (c d e))

((a b) c d e)

cons bears an inverse relationship to car and cdr in that the car of the value returned by a cons form is always the first argument to the cons and the cdr of the value returned by a cons form is always the second argument to that form.

(car (cons 1 ‘(2 3 4)))

1

(cdr (cons 1 ‘(2 3 4)))

(2 3 4)

3.4.4 Nested Lists, Structure

Although both cons and append may be used to combine smaller lists into a single list, It is important to note the difference between these two functions. If cons is called with two lists as arguments, it makes the first of these a new first element of the second list, whereas append returns a list whose elements are the elements of the two arguments:

>(cons ‘ (1 2)’(3 4))

((12) 3 4)

>(append ‘ (1 2) ’ (3 4))

(1 2 3 4)

It is important to distinguish between the lists (1 2 3 4) and ((1 2) 3 4), which have fundamentally different structures.

4.4.4 Functional Programming, Side Effects

LISP is based on the theory of recursive functions; early LISP was the first example of a functional or appreciative programming language. An important aspect of purely functional languages is the lack of any side effects as a result of function execution. This means that the value returned by a function call depends only on the function definition and the value of the parameters in the call. Although LISP is based on mathematical functions it is possible to define LISP forms that violate this important property. Consider the following LISP interaction.

```
>( f 4 )
5
```

Note that f does not behave as a true function in that its output is not determined solely by the calling parameter; each time it is called with 4, it returns a different value. Execution of the function creates a side effect that influences the behavior of future calls.

4.4.5 LISP Data Types in Common

LISP provides the user with a number of built-in data types. These include integers, floating-point numbers, strings, and characters. LISP also includes such structured types as arrays, hash tables, sets, and structures. All of these types include the appropriate operations on the type and predicates for testing whether an object is a member of the type. For example, lists are supported by such functions as list, which identifies an object as a list, null, which identifies the empty list, and constructors and accessories such as list, nth, car and cdr.

4.5 PARCS

4.5.1 Architecture

PARCS draw a major influence from the conceptual model of medical decision-making proposed by Bailey [Bailey 89]. Bailey's conceptual model includes three main stages. These stages closely relate to the thinking process of PARCS carrying out the inspection of the Ammunition. The MDSS proposed by Bailey is briefly explained shown in figure, keeping in view the PARCS requirements. Data Abstraction. It identifies the stage when the inspecting officer is collecting the initial observations, the initial specifications & Data of the Ammunition and converts them into their qualitative representation in the thought process.

Diagnosis Candidate Generation. In this stage the data collected from the above stage is evaluated and single or multiple likely solutions are generated at each stage

Investigative and therapeutic action. This stage narrows the gap between likely decisions by asking detail queries from the proof officer and consulting various available information's after fire tests in order to reach the most likely decision. Now coming on the architecture of PARCS which is briefly explained in figure, the data qualifier stage deals with the initial inspection of Ammunition and handles different stage inspection and tests of the Ammunition, propellants. It converts these findings into respective qualitative representation, similar to the data abstraction stage of Bailey's MDSS model. The Ammunition proof data collected is then evaluated by the proof officer/inspection after storing it in the knowledge base first, so that related features also get instantiated. The inspecting officer also acquires the data from physical inspection at the spot, and produces single or multiple likely decisions at the stage after the filtration process. Investigative Query Handler narrows the gap between likely decisions by requesting detailed information of the Ammunition obtained after the fire of the Ammunition. The ranges including various investigative tests in order to reach the most likely decision for final sentencing

Inference mechanism. Propagates the current data through the knowledge base hierarchy, so those features (from data objects) also get instantiated. The initial inspection data before fire is accessed by the inspector in order to instantiated their behavior, then all the collected data before the conducting of proof fire and data collected after the fire is compared, evaluated to reach at the most likely decision. At the top level, the main process collects the likelihood results, sorts and passes them to filter the data. The filter filters out the data/info whose likelihood is below the threshold value (cutoff) set by the user.

Similarly PARCS model is divided into three major parts. Which are before fire inspection, fire observations, and after fire observations. These major portions further consist of many knowledge source points, called the data objects. Fist stage comprises of three parts, which are manufacturing stage, filling of the propellant stage and inspections after filling.

First Stage .In the first stage the bullets are manufactured after carrying out the detail laboratory tests of the component of the Ammunition. All the tests are carried out keeping in view the technical and military requirement/specifications. Now the parts/components are assembled and empty round is now ready. The empty round is tested & checked its pressure sustainability .The empty bullet in now filled with propellants according to requirement of the Military. After filling the round it is handed over to the inspector's of Ammunition to check its stage inspection, visual inspection, its dimensions are checked in detail. Now of the Ammunition is correct & fulfil the technical requirements it is now handed over for fire test

Second Stage. In this stage accepted Ammunition from stage-1 in lots of different size with different requirements. If it is production lot its size in different, if it is lot then its lot size in different Lot. Sampling is carried out with respect to requirement. So samples from each lot are collected and are brought to the fire range to test the Ammunition’s battle worthiness, its reliability and its correctness. Before sending it to the range all the Ammunition is packed according to the laid down procedures keeping in view the environmental conditions. The proof officer conducts the fire, select the weapon which is correct in all respect and reading measuring instruments are placed at the fire ranges. Fire officer is responsible to collect all type of fire information’s/observations, to see the behavior of fire, behavior of propellant, etc .

Third Stage .In this stage all the observations of first stage and fire stage are put up to the sentencing authority for their evaluation and sentencing. Here all the test information’s are evaluated and filtered. The inspector uses the formulas if necessary to get the probable error of the fire. And now the subject Ammunition is discussed in detail, checked its allowable limits, then all deduction’s are put up to the sentencing authority, who again after checking the readings and observation’s accepts the Ammunition if it in correct, rejects the Ammunition if it is out of the limits and the Ammunition is said to be reproof if there are chances to get the same Ammunition within the limits again.

4.5.2 Data Objects

The data objects represent the set of features provided by the manufacturer responsible for creating the knowledge base. The data objects having qualitative or quantitative values are named accordingly. The qualitative data objects represent qualitative feature values.

4.5.3 Quantitative Data Objects

Quantitative data objects can take numerical as well as qualitative values. Objects like probable error, standard Deviation, come under this group, since they can get numerical values as well as qualitative values.

4.6 Summary

In this chapter design procedure of the system carrying the proof schedule rejection, acceptance & reproof of ammunition have been discussed. Then keeping in view the AI (artificial intelligence) requirements of the system, a conceptual model of the system PARC has been presented. Each component of this model has been discussed in detail. In the next section, the design strategy has been described to show that how the system will be actually implemented.

CHAPTER 4

Implementation

5.1 Introduction

This chapter describes the implementation details of the system being developed. In the first section, the architecture of the system keeping in view the source code is discussed, which includes the different modules of the system. Limitations and advantages of the system with respect to limitations of the rules and interface limitations have been discussed. Data collection methodology, and the filtration of irrelevant data is discussed in the next section. In the last section characteristics of knowledge based PARCS system have been discussed .The whole chapter is summarized at the end.

5.2 Source Code of PARCS

It is necessary to know and understand the source code files of present designed system for proper implementation. The original package involves over three hundred LISP files, which are loaded during run time. Few important files have been explained in subsequent paragraphs.

5.2.1 Types of Files in PARCS

PARCS contain following types of files.

- **Builder Files**
- **Lisp files.**

Builder files: having the extension bill these files contain the source code for graphical user infuse of PARCS. All the forms and windows of PARCS are designed by system engineer and created by LISP.LISP generates most of the code. This is one of the main reasons for selecting lisp as a tool for this system.

Lisp Files: These files comprise the system. Logically these can be divided into following categories.

Data files: These contain the data of the system. All the records are kept in these files in the form of db-insert commands. When these files are loaded, the data is transferred to the working memory from where other functions can access it.

Rule files: These files contain the definition of rules. When these files are loaded all the rules are defined. Check rules and insert rules are two major types.

Function files: These files contain definitions of all the functions being used in this system. When these files are loaded, functions become defined. These files have the extension also.

Master file contains the Master Knowledge Base in the form, which is understood by the inspector, user and manufacturer. It contains details of pre-fire requirements and initial observation in plain text form. In fact PARCS maintains two separate parts of the knowledge base: one is the master knowledge base, which contains all the validated objects. Second is the extension knowledge base, which contains all the newly entered data. The advantage of keeping two separate knowledge bases are that new objects added can be tested by loading extensions knowledge base without disturbing the certified master knowledge base. Once tested the object, can be added to the master knowledge base. The other issue to be addressed is that the senior Ammunition experts can test knowledge addition, when they get time, before its addition to the master knowledge base.

The object file contains data objects in a form, which is understood by the system. The plain text of Master knowledge base file, having different constraints and restrictions, have been converted to a structured layout.

5.3 Advantages and Limitation of rules

Firstly, there is no guarantee that your program will perform in the way that you expect unless the rules have been carefully written with the conflict resolution strategy used by the interpreter kept constantly in mind.

Secondly, representing knowledge is an unordered and unstructured set of rules have certain disadvantages which, taken together, probably outweigh the advantage that one can easily add another rule to the set and let the conflict resolution sort out when it should fire. Finally, although production rules seem well suited to encoding empirical associations between situations and actions of the general form, if these conditions hold, then do this, they appear to be less effective as a means of expressing more subtle forms of knowledge which can be used to reason about the fundamental nature and causes of interesting phenomena. Production rules in the service of knowledge-based programming suffer from the shortsightedness as well. The major advantages of rules are as following:

- Interference and explanations are easily derived.
- Modifications and maintenance are relatively easy
- Uncertainty is easily combined with rules
- Each rule is usually independent of other

5.3.1 Rules in PARCS

The system is able to add new rules to the knowledge base and can also correct the existing rules. A small window displays the rule numbers of all the rules presently in the knowledge base. The user selects the desired rule number and clicks on the display button; the appropriate rule will be displayed. To make the system more users friendly and elaborative, rules are displayed (and can be entered as well) by parts i.e. for rule priority a separate window is opened. Where the window shows priorities form zero to the required numbers, the user has only to click on the desired number and the system will automatically select that number as priority for that particular rule. Similarly a separate window is opened for the pattern and for the s-expression (body) of the rule. After entering the new rule or modifying the existing one. The user can save and ultimately load this rule in the knowledge base. A button named as "clear" on the toolkit, reinitializes the rule editor, hence the user can re-examine the knowledge base rules, edit them and enter new ones. Another advantage of this system is that, once a rule has been entered or edited, it checks for data consistency and does not allow duplicate rule names. The major limitations of rule representation are as follows:

- Complex knowledge requires many (thousands of) rules. This may create problems in both using the system and maintaining it.
- Builders like rules: they try to enforce all knowledge into rules rather than looking for more appropriate representations.

- Systems with many rules may have a search limitation in the control program. Some programs have difficulty in evaluating rule-based systems and making inferences.

5.3.2 Filtration Process

After collecting all the support features for the particular Ammunition, the likelihood evaluation sub-process calls the compute likelihood rule. This rule further applies filtration rules to that Ammunition support, in order to eliminate it (filter it out) if support is below the given cut off. If we start raising this cutoff value, more filtration will be carried out, which may have been filtered out previously. This cut off value allows the user to vary the focus on the different choices generated by the inference mechanism. The user can relax the focus to see different alternatives, which have only weak support, or tighten the focus to only see those alternatives with a high level of support.

5.3.3 Data collection Methodology

There may be instances where complete object data is not entered into the system, which is essentially required by the inspectors to reach to likely decision. The system has a fixed set of features. Which are either supplied by the user or asked by the expert inspectors. When the user forgets to provide some of the required features, and requests the system to run the inference mechanism, the system comes back to indicate that there is some missing information and asks the user whether he will provide the further information or he just wants to run the system with the data provided so far.

5.3.4 The Working Memory

The most basic function of the working memory (WM) is to hold data, in the form of object-attribute-value triples. This data is used by the interpreter to drive the rules in the sense that the presence or absence of data elements in the working memory will trigger some rule, by satisfying their activation patterns.An example will make this clear. If the working memory is a list containing the following triples:

WM= ((Alam age 36) (Alam employment none))

Then at the next cycle the interpreter will look to see which rules in production memory have conditions, which are capable of being satisfied. If a condition contains no variables, then it is satisfied only if an identical expression is present in working memory. If a condition contains one or more variables, i.e. if it is a pattern, then it is satisfied only if there exists an expression in working memory which matches" it in a way that is consistent with the way in which other conditions in the some rule have already been matched. In this context, a simple match is just an assignment of constants to variables, which, if applied as sub situation, would make the pattern identical to the expression that it matched against. Thus, (Alam age 36) satisfies the condition (“Person age” number) with substitution ‘Alam” for person and 36 for age.

5.4 RULE BASED INFERENCE MECHANISM OF PARC

One of the main reasons that LISP is being used in the field of Artificial intelligence is that intelligent inference mechanism can be built on the bases of Rules. MYCIN is the most common example of the expert system, which contains the Rule based inference mechanism.

PARCS also has the Rule based inference mechanism. These Rules interact with other rules through the working memory. Their mechanism is closely related /like the human intelligence system.The figure 4.1 shown below is the simple brief architecture of the inference mechanism of PARCS.Insert Rules are defined when the data files are loaded. When a user selects any Ammunition, the insert Rule of that Ammunition is fired. It inserts the name of Ammunition, its type of proof, and all its related material in the working memory. So that system exactly knows which Ammunition is selected. After the selection of the Ammunition, the user has the two options available with him, if he wants to check its previous record, the corresponding functions will do this task. If the user wants to do the proof scheduling the Rule based inference mechanism is activated. Now it is the duty of check Rules to make the decision and insert the Rule in the working memory. If the Ammunition needs PE (probable error), the system will give the message to the user for the calculation of the PE. If on the other hand Ammunition does not need PE, the system knows it. It will not allow the user to enter the PE.

Then step by step other Rules are fired.(fired means executed , it is special term used for the Rules.)They check all the conditions one by one. They don't allow the user to enter the wrong data and the readings and proceeds forward. At the end the observations made by the proof officer are entered .The system makes the decision of the Acception, Rejection and Reproof of the Ammunitions.

There are about 120 Rules in this system. Some Rules are associated with only to their respective Ammunition like insert rules. They are only activated when that particular Ammunition is selected. Some Rules are general i.e. they are not concerned with that Ammunition. They can handle all the Ammunitions. Name of the Ammunition from the working memory, get the final sentence from the sentencing authority from the form and result from the working memory. It combines all this data and saves it in the record of that Ammunition in the form of db-insert command on the record- file. It also displays the message to the user that the record is added to the system.

Consider the other Rule of this system Rule 1-1-3, it is specific for each Ammunition. Ammunition two has its Rule 2-1-3 and next as Rule 3-1-3 and so on. First digit indicates the number of the Ammunition, second is the category and 1 is for proof scheduling. The third digit is name of the Rule, is the number of the Rule, which is 3 in our current example of the Rule. This Rule check the Ballistic data of the Ammunition .It gets the current data from the user and match it with the actual data for that Ammunition and make the decision accordingly. Thus the inference mechanism of the PARCS works simila rly as the human mind does. It makes use of the different Rules for its decision making.

"There are now in the world machines that think"Herbert Simon.

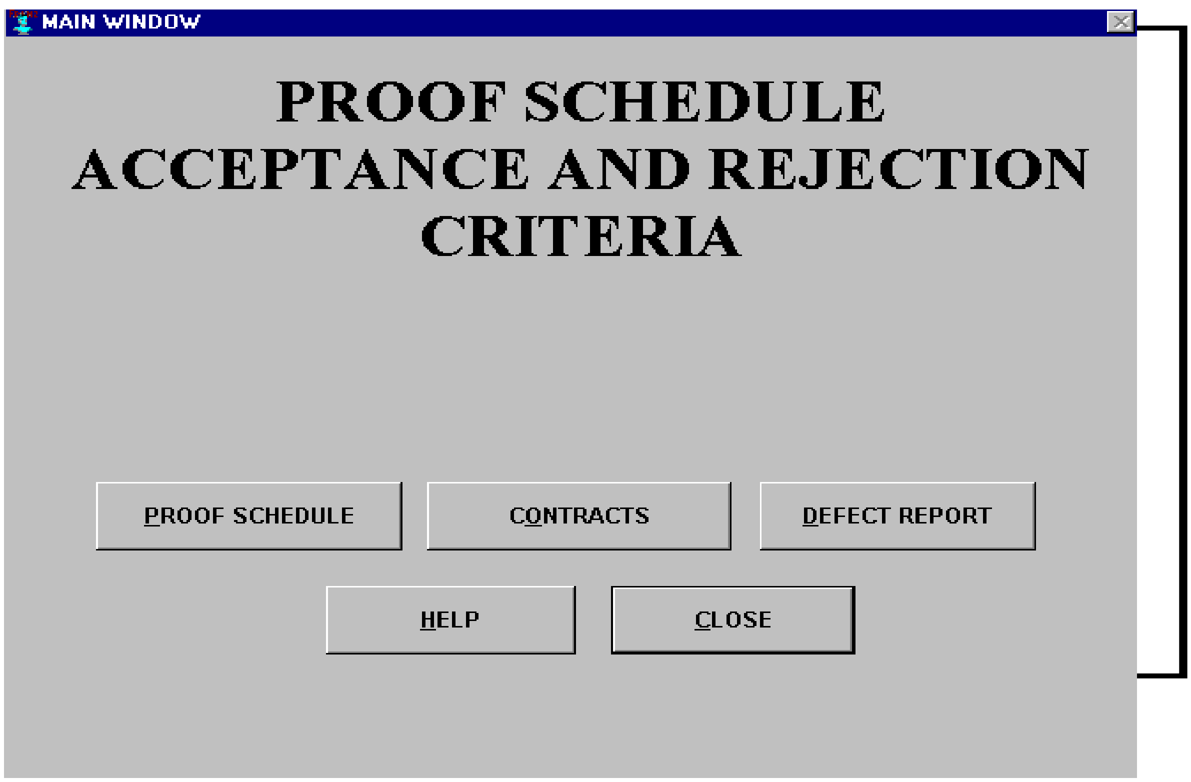
5.4.1 WORKING

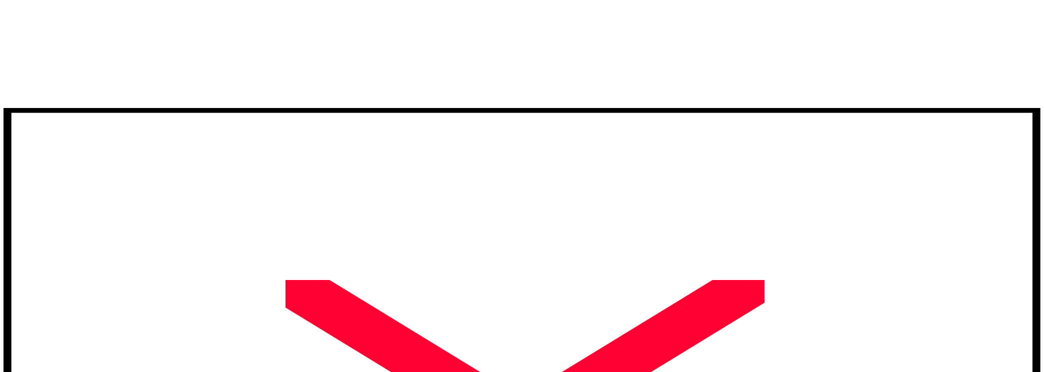
The carrying of the proof scheduling, acceptance, reproofing, and the rejection of the Ammunition PARCS is carried out in the following steps, which will give the result by using the expert quality .Working of the PARCS is shown by the fig. 4.2 also:

- **Step I**

User load, Code lines written in this file are executed which load (pre-defined function to load a file) all the files of the systems. In this way all the functions and rules of the system are defined (Move from hard disk to main memory of the computer)

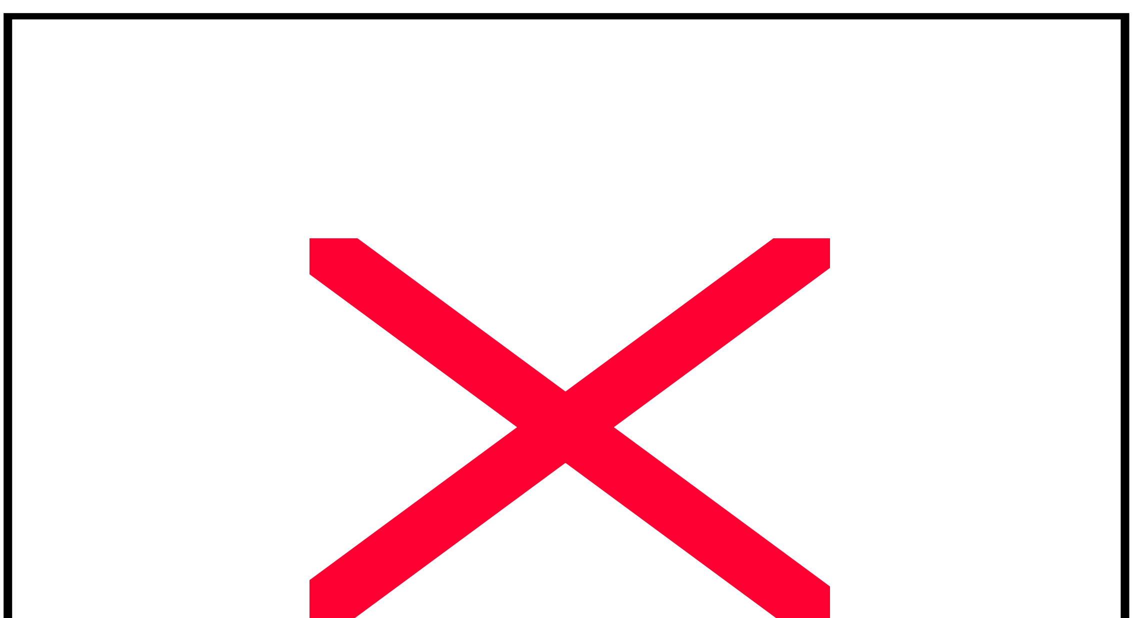
Second last line of this file i.e. open , will open the main window of the system. Last line i.e. (close * system –loading *) closes the blue window (wait window) of the system.





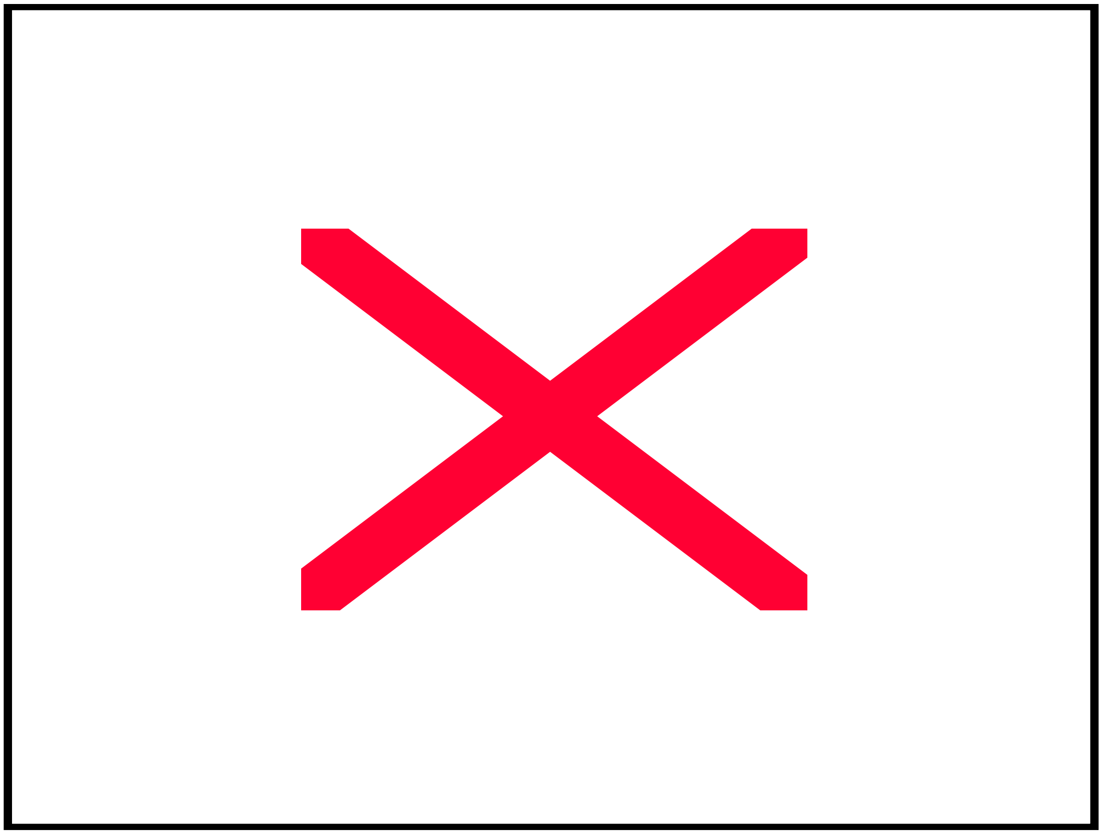
- **Step 2**

If we click any one of the five buttons of main window the specific window is opened. Suppose we click proof schedule button code of the button (Main attribute – event handler – set value function) is executed which is simply ammunition. It opens the ammunition window.



- **Step 3**

Now we are on the ammunition window. We can select any of the ammo. Suppose we select motor – 60mm – HE, open proof 1 function is executed (Db-insert (ammo Mortar – Bomb – 60mm HE- filled)). This line insert in the working memory the name of ammo which is Mortor HE and its type. (Proof 1) this line opened the next window which is * proof 1 * which contain proof schedule, enter code word show record end enter PE buttons (Set – dialogue – item – value (widget: Static-text – file * proof 1*) " Mortar 60mm HE") this line display the name of ammo on proof 1 window. Now we are on proof 1 window



- **Step 4**

  PE Calculation

  Button simply opens the PE form on which PE (L) and PE (R) can be entered.

Record Table

This buttons will get the all-previous record of Mortar Bomb from working memory and display it on a window. This record was entered in working memory when record file. lsp was loaded with all other files of the systems.

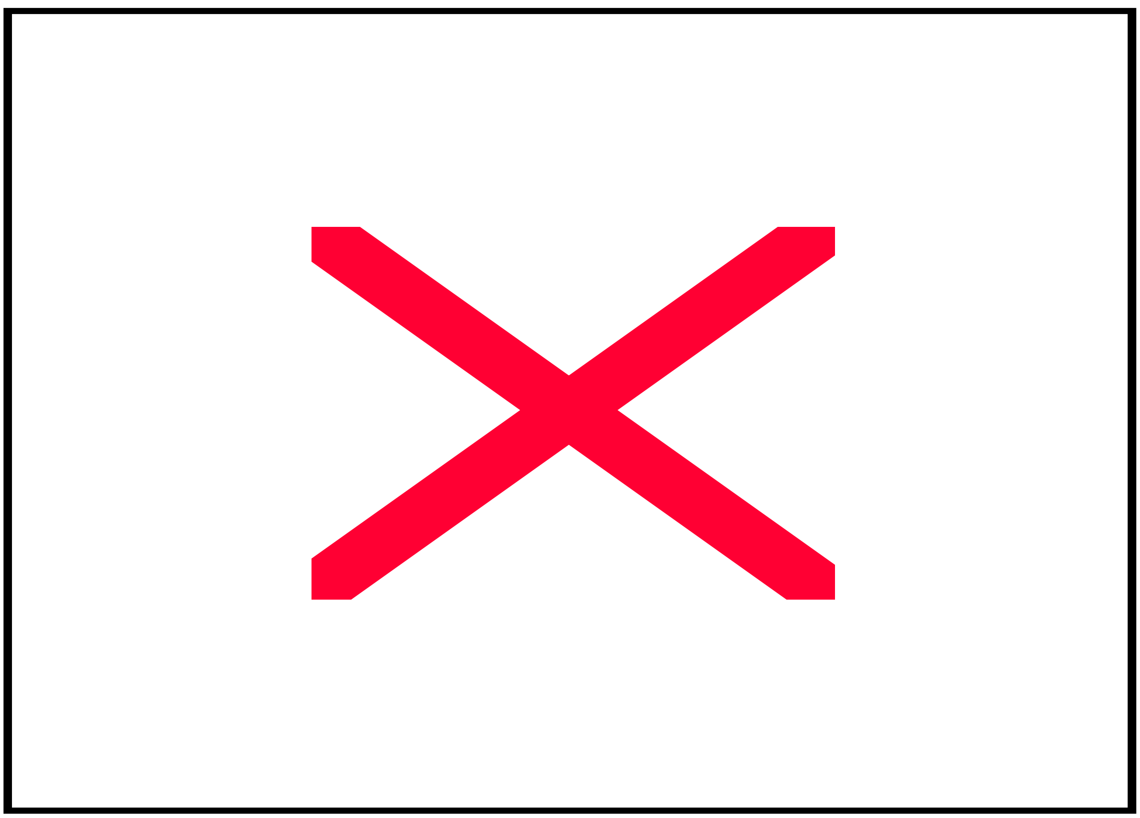
Enter Code Word

This button of the code word from user and it it matches with the actual code word it do (db-insert (code word entered OK))

Enter Data For Proof

When the Button is clicked, openinit function is executed (called) it check if the code word is not entered, it pop-up the message to enter the codeword and exit. After checking the codeword it checks the PE is entered or not, if not entered it pop-up the message if PE is entered and is within limits it means Ammunition is not yet rejected. This function opens the with window of the Ammunition. In this case (db-find (PE range !x !y)) will return nil indicating that Mortar Bomb 60mm HE have no range of PE or simply it does not need Pe so it will open the next window by following the first path in cond.

(Cond ((DB-FIND (Ammo Mortar Bomb 60mm HE – Filled)) (Opeinti)). This will open the * initob *. Now we are on * initob * window.



Step 5

Now we click on button its event handler function, which is open-vardata/, is executed. Its code is as follow:- (Db-insert (Rule-1-1-2). This live insert (Rule-1-1-2) is working memory, which is the pattern of Rule-1-1-2, so (if (Run) (Varda 1)

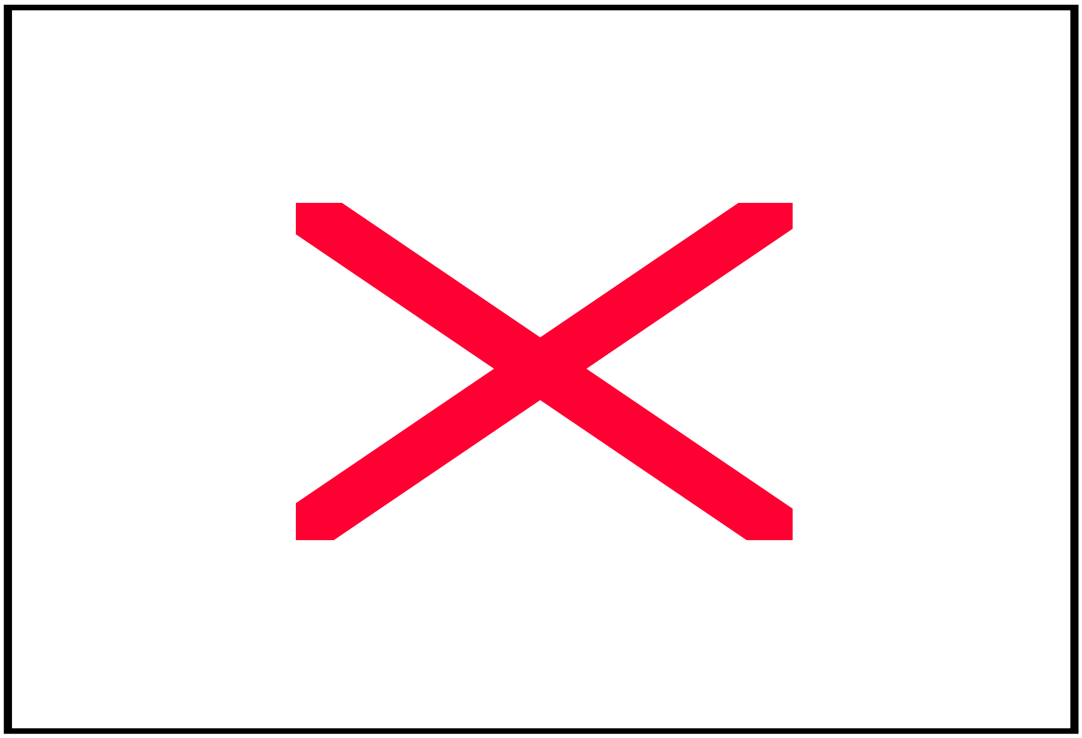
Run with execute the rule-1-1-2 if this rule return in. this function will do noting if this

rule return true then part of if which is (vardata 1) is executed which simple open the new Window wardata1, Rule-1-1-2 will delete its pattern.(and box 1 box 2 box 3 box 4 box 5 box 6 box 7) will check if the all are true, if not true it will display the pop-up-message and return will suppose all check-boxes of initial observation window are ok, so we Move the next window which is *vardata 1).

Step 6

On this window user enter the ballistic data. When user press enters button. Its event handler set value function, which is open-observations function, is executed. Its code is (DB-Insert (Rule-1-1-3))

It will insert (Rule-1-1-3) in W.M which is pattern of Rule-1-1-3, so run will execute this rule. This rule will get the ballistic data of the Ammunition. From W.M. and get the current ballistic data from the user it matches these two data, for any mismatch pop-up message dialogue is displayed. If all variable data matched. True is returned from this rule, true is returned from (run) and then-part (initob-1-2) is executed. It simply open the next window. Initob-1-2

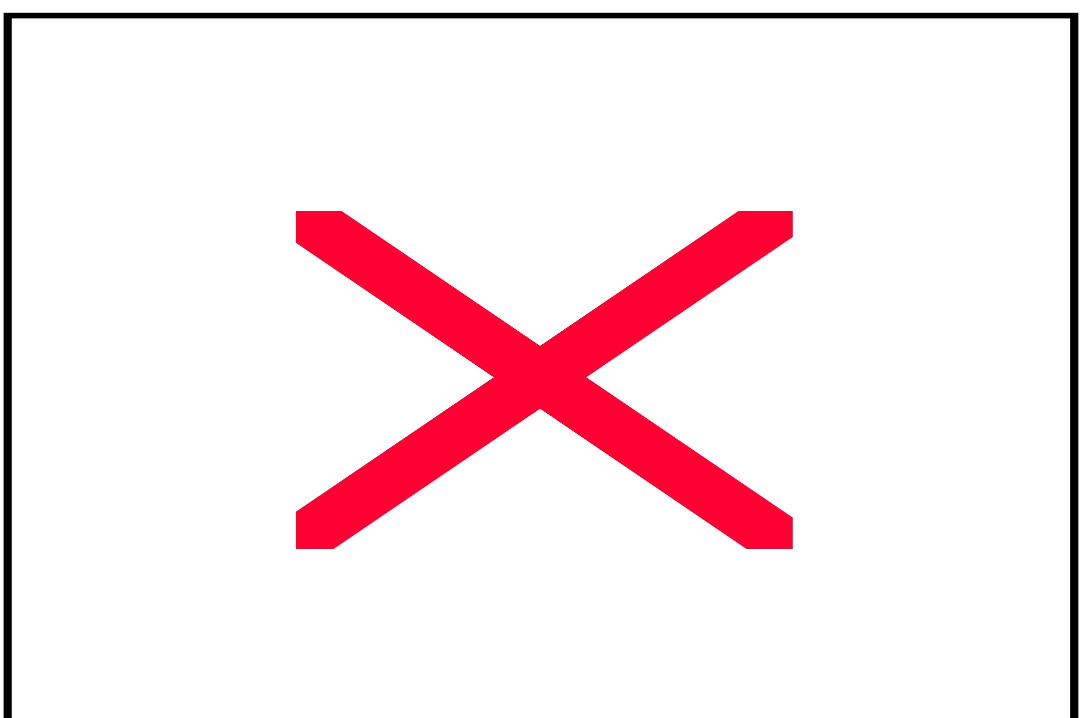


Step 7

Now we are on method of proof window its OK button will simply check that all should be true etc. it use rule-1-1-4 (exactly similar to rule-1-1-2. If all are OK then we enter the final observation windows.

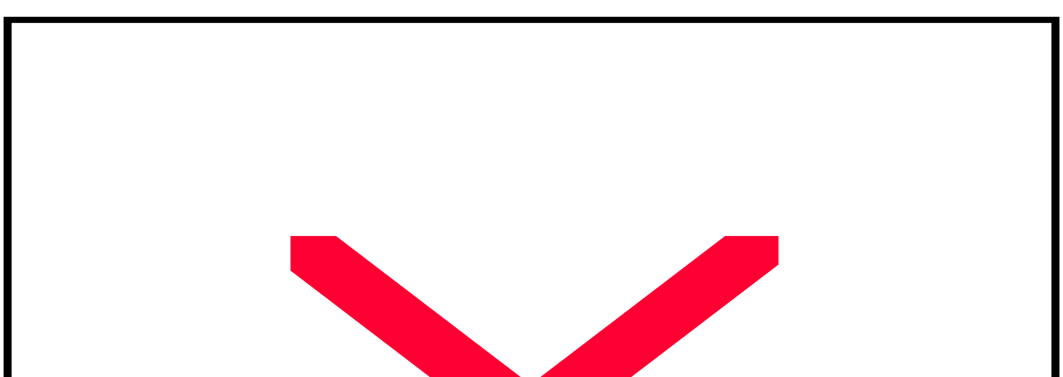
Step 8

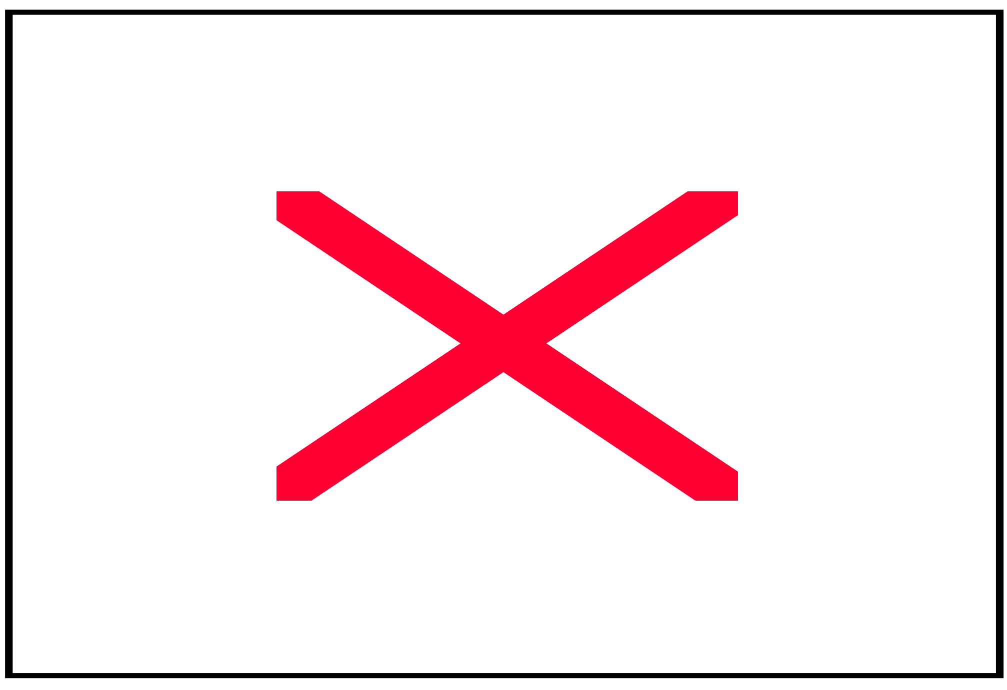
After entering the final observations officer will press enter button its function show-results will be executed. First of all is insert (Rule-1-1-5) is W.M. and then call (Run), Run will execute Rule 1-1-5. This rule gets the observations from editable texts entered by proof officer. Match these observations with the Acception and Rejection criteria of the ammunition and insert the result in W.M by (db-insert (result reproof " Reproof due to one blind etc')). After the (run) committed (due to which result is entered in W.M.). Condition is checked if the result is fined by db-find (result) command will open the result window. then (set-dialog-item-value (widget : result-reason result) reason) will set the reason of result on result window. Then it deletes all the data of ammo by db-delete commands. Then this function closes all the opened windows of Mortar Bomb 60mm HE except the results window.



Step 9

Now we are on result window officer will enter the sentence and press the ok button .Rule-1-1-6 is called by entering its pattern and calling (Run) this rule will get result, sentence, and date from the result window open 2 * Record-file * which is d; \\hashim\\record-file.Isp” write or print the db-insert command of new record on it and them close the * record-file * . After entering data by Rule-1-1-6 this function give pop up message that record is entered. Then it delete pe (if required) result and name of ammo from W.M. close * proof 1 * and * result * window and we again return back to Ammunition window repeat the same process form other Ammunition.





5.4.2 Editing of Data Objects

The GUI contains show, enter and edit options for knowledge base objects (initial specifications, after fire reading & sentencing) along with data validation options. The expert enters the detail first time and subsequently the inspector's user can enter this data through the knowledge elicitation toolkit. Knowledge elicitation option can be selected from PARCS main menu. The toolkit menu allows the user to show an object, enter a new object and edit and object. The entered objects can be qualitative data-object. Quantitative-data-object, all is evaluated before & after fire and then the Ammunition will be sentenced.

5.5 Characteristics of Expert System PARCS

5.5.1 Knowledge Engineering Tools

An expert system PARCS provides knowledge engineering tools to assist users who don't have the assistance of expert knowledge engineers. Such tools would also be useful to knowledge engineers reviewing their own work. Relevant tools are by [Citrenbaum 87]:

- Allow the user to first gather and assert knowledge and then shape it while providing feedback regarding its structure and interrelationships.

- Provide knowledge consistency and complete checking with a high quality explanation facility to indicate the cause of any inconsistencies found.
- Develop rules if/then decision trees from a set of examples (inductive knowledge engineering).
- Optimize query sequences in rule-based systems. The user would only be required to enter the query; the PARCS would work out the need and optimum rule sequences.
- Provide reasonable defaults for all options. Ideally these defaults would be provided both the system level and separately for different classes with their own defaults could include the areas of diagnostics classification.
- Provide built-in, high-level domain expertise in certain basic areas to guide the users.
- Interpret a reasoning audit trail maintained by the inference mechanism and suggest ways to reach desired conclusions more quickly and surely.

5.5.2 Security of PARCS

- Top level Password
- No access to System Data
- Password check for proof officer

Security is of prime importance in this system. Before entering the system you have to enter the password of the respective log in name. The system engineer can only create new login name. There is no restriction on its count i.e. system Engineer can credit as many log-in name as he wanted. If the password entered is not correct pop up message is displayed. If wrong quitted. Next time it would be slow. Two or three failures will stop the system i.e. it would be halted. Thus there is zero percent chance that any illegal user can open the password. The facility of changing the password is also available. Officer can change the password permanently on his desire. There is a special check on proof scheduling which need a code word. Proof officer can change code. If correct code is not entered the system will automatically turned off.

- **Data Hiding**; main power of PARCS is its data hiding mechanism. There are rules named insert rules, which contains the correct data for each Ammunition.The data of each Ammunition is invisible to user. It can not be changed. User can enter the current or fresh data, which is matching with the correct data. If it passes the system proceeds otherwise it remain still there. It is not possible to have any Ammunitions lot accepted with wrong data. Rule based inference mechanism keep track of it.Thus we have two security layers first the security on system with the help of passwords and code words. Secondary the security of the data related with proof scheduling, acception and rejection.

5.5.3 Interface

An ideal interface should [Citrenbaum87]:

Provide standard default features enabling expert systems to be constructed with minimum effort.

Provide guidance. A very simple user interface may be suitable for new users or domain expert with no expert system development experience (or interest). The interface should provide guidance. Either explicates or implicitly, through the model of the word and default values it presents. The PARCS put together an interesting expert system without requiring the user to make decisions that require knowledge of the internal system operations.

Deal with graphics. Some user interfaces deal easily with graphics. Especially image capture and display

5.6 Summary

The implementation details of the ideas conceived in the previous chapter have been given in this chapter. After discussing the overall architecture of the system developed, the details of the graphical user interface is given along with the interface diagrams as a first step to acquaint the user with the system. The chapter also included details discussion on few of the main objectives of the project like, execution of formulas to get the probable error of fire, getting the information for acceptance, rejection and reproof of Ammunition and final sentencing.

Chapter 5

Conclusion

6.1 Introduction

This chapter summarizes the whole project and assesses the accomplishments of the project. It also describes the original contribution and research done in the course of the project and finally it recommends the future work to be incorporated in this project. It discusses the limitations and shortfalls of the system with reasons and suggestions of improvement. A detail study of PARCS system was essentially required for its implementation in the field. The process of rationalizing and developing a new knowledge representation for PARCS was a rewarding educational experience. During the process, many new areas were explored which were of great educational value. Some are highlighted below.

6.1.1 New Areas

- Understanding of the subject field and its working.
- Understanding the strategy for writing interpreter.
- Facilitated for quick decisions and it has made possible for the inspectors of Ammunition to inspect very correctly and error free.

- Enhanced efficiency of the troops due to correct selection of weapon and correct Ammunition.
- Packing the raw information into knowledge representation using AI techniques.
- Structured programming style and its implementation.
- Understanding coding, re-coding the advanced level LISP functions written in the PARCS code.
- Reducing the normal working labor
- Self-storage of data.
- Easy availability of record stored.

6.2 Limitations of the system

In the conversion process of the knowledge base of PARCS, a number of system/implementations level limitations were encountered. These problems, although successfully dealt with, present a very broad future work in the respective domains.

. Current system of PARC has following deficiencies: -

Difficulty in capturing 'deep' knowledge of problems domain, for example in MYCIN dev in 1970's, it could not understand the real knowledge of human physiology. Similarly PARCs lacks in real complete knowledge of it decides with the given knowledge.

Inability to provide deep explanations. as the PARCS such deep knowledge of problems domain, so it is unable to explain, why a certain approach was taken.

Difficulty in verification’s. Though the correctness of any large computer systems are particularly difficult to verify. This is a sessions problem as this technology in being used in critical applications such as air traffic control., nuclear reactor operations & weapon system.

Lack of robustness & flexibility. If PARCS like any other expect systems, cannot follow/assume a new approach to solve the problems like human kind.

Little learning from experience. Like other expert systems are completed, its performance cannot improve with out further attention from its program.

Inspite of these limitations, expert system’s are proving their value is a number of important applications like PARCs will prove its value & it is hoped that the above limitations will only encourage the new student to process & enhance this important project.

6.3 Future Research Work

6.3.1 New Computing Architecture

A new corporate computing paradigm, referred to as multi-tier client-server, has recently arisen to address the challenges associated with knowledge-based information processing. This architecture consists of a database server layer (containing various applications for processing and filtering information) and a "thin" client layer. The layers are connected together via the corporate LAN, Intranet, extranet, or the Internet. The best utility of PARCS is visualized in this Multi-tier Client-Server Architecture. Thus PARCS should incorporate dynamic querying system and web browser interface as a pilot project.

6.3.2 New Vision

Vision technology has taken a boost with the enhancements of processor speeds and graphic improvements. The results of PARCS system are displayed in a surrealistic or simulated manner. A work in this direction can not only enhances the present PARCS system but will also have a profound effect on the other aspects like education and graphics etc.

6.3.3 Integration with Relational Databases

Now a days the object oriented languages are incorporating relational database systems for the raw data. Integrating PARCS with the existing databases like Oracle, DB-II etc with a view to compare the processing time can form a major research domain.

6.3.4 Multimedia Facility

This facility if incorporated, will definitely enhance the implementation phase of PARCS to Army/civil manufacturing organizations.

6.4 Achievement

The designed software PARCS provides some facilities, which are:

- Understanding of the subject field & its working
- Facilitation for quick and correct decision which are error free
- Enhancement in the troops confidence
- Increase efficiency in staff work of Inspectorate and professional efficiency of troops is definitely enhanced many time due to avail of correct ammo
- Easy availability of old record
- Reducing the labor in consulting the book
- Self storage of data
- Security
- Understanding coding the advance level LISP functions written in the PARCs code.
- To introduce awareness about the technology of future. i.e. computer.
- Reliability

- Cost effectiveness

6.5 Project summary

This decision support system has been rationalized into an Expert system PARCS that can be used for military and industrial control applications. Like a medical expert system “POEMS” (Post Operative Expert Medical System) was developed for providing assistance to doctors to reach a diagnostic decision [Sarwar 92]. The current model collects the data by letting the user enter relevant data. This data is then used by the system to diagnose the faults/complications.

The main contribution of this research is to extend the knowledge base representation of the existing system to handle production rules. The benefits of knowledge representation have been exploited in this system because frames were difficult to program, and difficult to infer, whereas the production rules provide a totally effective representation facility. The major inadequacies of frames are in areas that are effectively handled by production rules.

A great deal of success, has been achieved by production rule language. This improvement has provided tremendous flexibility in knowledge representation because this capability makes it easier for the domain expert to construct and understand rules. And for the system designer to control when and for what purpose particular collection of rules are to be used by the system. Finally, a graphical user interface has been developed in Franz Common LISP which allows the user to enter required data and also provides the facility to the user to browse and edit the production rules. This means that user has the facility to modify and improve upon the knowledge base of the system. This facility provides enormous versatility to the system, making it fulfill the criteria of a respectable expert system PARCS.

The system was developed from the conceptual model. The mechanism of the PARCS has been kept as simple and transparent as possible, which in turn reduces the conceptual problems, which user may have in using the system. The knowledge representation is done in the production rules. This choice despite the limitations of scarifying the goal of uniformity in favour of exploiting the benefits of multiple knowledge representation seemed to be a good choice. The rule base used was developed from the elementary single pattern matcher rule based system. Certain changes were incorporated to develop it to handle the complex knowledge.

6.6 Conclusion

This work has introduced many new concrete concepts towards the knowledge base systems and has eliminated the shortcomings of older systems. Due to the quick sentencing a very vast area of research has been opened up for the successors to experiment and bring new concepts for the expert systems. The system can easily fulfill the needs of future developers and researchers.

Bibliography

[Biondo 90] Samuel J. Biondo, “Fundamentals of Expert Systems Technology; Principles and concepts”, Albex, Norwood, NJ 1990.

[Clancey 86] Clancey A. R., “From Guidon to NEOMYCIN and HERACLES in twenty short lessons”, AI Magazine, 1986

[Davis 76] Randall Davis, “Mwetal Level Knowledge in the Construction and maintenance of large Knowledge Bases”, Computer Science Department, Stanford University, Stanford, CA 94305, U.S.A 1976

[Franz 94] Allegro CL for Widows, Volume I, II and III, Version 2.0, document number D-U-00-PCO-06-4026-2-4, Berkeley U>S>A, 1994.

[Garcia 92] Oscar N. Garcia and Yi-Tzuu Chieu “knowledge Based System fundamentals and Tools”, IEEE Computer Society Press, 1992.

[Goldman 80] Benette J. S. and Goldman D, “CLOT: a Knowledge Based consultant for bleeding disorder”, technical report, Computer Science department Stanford University Memo HPP-80-7, 1980

[Hollander 81] Hollander C. R. , “DART: an Expert system for computer fault diagnosis”, proc IJCAI-81 pp 843-845, 1981.

[Nilssson 80] N. Nilsson, “ Principle of Artificial Intelligence”, Tioga, Palo Alto, California 1980.

{Noyes 92] James L. Noyes, “Artificial intelligence with Commmon Lisp”, DC Health and Company 1992.

[Parsaye 88] Kanran Parsaye and Mark Chingnell, “Expert Systems for experts”, Prentice-Hall, USA, 1988

[Rusmus 96] Daniel W. Rusmus, “Developing AI applications” object magazine June 1996.

[Sarwar 92] M.J. Sarwar, Post Operative Expert Medical System, Ph.D. thesis. University of Leeds, 1992

[turban 92] Efrahim Turban, “Expert System and applied Artificial Intelligence”, Macmilln Publishing, 1992.