Design of an Expert System for Ergonomic Assessment

Shen Fan

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

April 2002

0-612-68452-0

ABSTRACT

Design of an Expert System for Ergonomic Assessment

Shen Fan

As a branch of artificial intelligence, expert systems have achieved a tremendous development in the past two decades and have gained mature implementations in many industrial fields. Ergonomics is playing an important role in the modern society. Applying the technology of expert systems in ergonomics makes it possible for people to use the expertise of ergonomic experts to finish their work correctly and efficiently without the assistance of experts.

This thesis presents the design of a rule-based expert system that is used to analyze and evaluate the working conditions by ergonomic standards. CLIPS (C Language Integrated Production System) is the tool used to develop this expert system. The knowledge base of this expert system is acquired from OSHA's (Occupational Safety and Health Administration) ergonomic standards and Washington State's Ergonomics Rules.

Our expert system has five subcomponents construct this expert system: management component, general OSHA questions component, Visual Display Terminal analysis component, Lifting calculation component and In-depth work analysis component. The management component controls the flow of the other subcomponents whereas the other subcomponents identify hazards of musculoskeletal disorders and evaluate ergonomic conditions in the workplace. The power of our expert system is shown by doing a step-by-step in-depth work analysis as an example.

Acknowledgment

I am honored to express my gratitude to Dr. Chen and Dr. Gouw, my supervisors for their continuous support towards the success of this thesis. I deeply appreciate their invaluable advices and encouragement that enabled me to reach this education level.

I would like to thank my friend Mr. Ming Li for his continuous assistance and encouragement throughout this work.

I am deeply thankful to my parents and my brother. Without their love and encouragement, this work would not have been possible. I also wish to thank my cousin who gave me many valuable recommendations during my study.

I am also thankful to many of my friends who have supported me technically, morally and spiritually.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview of Ergonomics

1.1.1 Introduction

Ergonomics is the application of scientific principles, methods, and data drawn from a variety of disciplines to the development of engineering systems in which people play a significant role. Ergonomics focuses on the human as the most important component of our technological systems and it concerns the characteristics of the people involved, particularly of their dimensions, their capabilities, and their limitations.

1.1.2 Historical Review of Ergonomics

We can trace the history of ergonomics back to ancient time. A fundamental activity of human beings was that they could use objects found in the environment, such as tools or weapons. This is the difference between humans and other primates. Purposeful shaping of tools, creating finished products from raw materials, fitting clothes and making shelters were early and fundamental "ergonomic" activities.

Early in human civilization time, people began to do some research about the human body. About 400 B.C. Hippocrates described a scheme of four body types. Over the centuries, more exact information accumulated into specialized disciplines. From the 15th to 17th centuries, gifted persons such as Leonardo da Vinci could still master all the existing knowledge of anatomy, physiology, and equipment design [1].

Although there were some activities about human factors, ergonomics emerged as a subject in the 20th century.

In the early 1900s, in the United Kingdom, the Industrial Fatigue Research Board considered theoretical and practical aspects of the human at work. In 1913, Rubner founded a Work Physiology Institute in Germany. In the United States, Benedict and Cathcard described the efficiencies of muscular work. In the 1920s, the Americans established the Harvard Fatigue Laboratory [1].

In the first half of the 20th century, two distinct approaches to studying human characteristics had developed. One was concerned mainly with physiological and physical properties of humans, the other with psychological and social traits. The former was studied mainly in Europe and the latter in North America [1].

In North America, during the time of the Second World War, people began to study human factors as part of "man-machine systems". On January 13 and 14, 1950, a group of British researchers met in Cambridge, England. The term "ergonomics", from the Greek words "ergon" meaning "work" and "nomos" meaning "law" was proposed and formally accepted as the name of the new society at its council meeting on February 16, 1950 [1].

1.1.3 The Current Status of Ergonomics

Since the birth of ergonomics, more than half a century has passed. Ergonomics has had significant developments both in developed countries and developing countries. But the position of ergonomics in those countries is different.

- **Developed Countries**

Ergonomics has become an "identified profession" in the Western world. In North America and Europe, people implement the principles of ergonomics in many fields. They established social administrations and organizations, such as American Industrial Hygiene Association (AIHA), Human Factors and Ergonomics Society, and Occupational Safety and Health Agency (OSHA). Many universities have specific faculty for ergonomics research and education, such as Michigan University (USA), and Concordia University (Canada). Some enterprises hire specific duty persons to be responsible for ergonomic programs, such as CAE Montreal, Canada. In the market, people can find more and more industrial products embedded with ergonomics design. In our daily life, the implementation of ergonomics brings us many conveniences. Ergonomic furniture, ergonomic computer keyboards, computer monitors, traffic guides, comfortable car seats, proper office temperature, etc. These reduce the rate of working accidents, increase working efficiency, bring benefits to people not only physically but also psychologically.

- **Developing Countries**

Ergonomic ideas did not emerge early in developing countries. Interest in ergonomics among developing countries followed belatedly after its introduction in the developed countries. In 1983, the first international conference on the ergonomics in developing countries was held at Lulea University, Sweden [2]. In 1985, the International Symposium on Ergonomics in Developing Countries was held in Jakarta, Indonesia. More than 300 participants from 20 countries, mostly from developing countries attended the 3 day-meeting [2]. With the great scale of import of advanced facilities and frequent technical and research exchanges with developed countries, the concept of ergonomics came to the developing countries. China is a typical example. It reached its peak of import during the 1980s. Many advanced technologies and facilities were introduced from Western world. From then on, some enterprises improved their safety management systems and commercial advertisements spread the concept of ergonomics among consumers. There are special training programs of ergonomics offered. Some companies improved their product design with ergonomic ideas.

Generally speaking, although the interest and applications of ergonomics are growing in developing countries, the result is far from satisfactory. Human factors research is very limited in developing countries. Ergonomics is taught in very few universities, and to a very limited extent. According to statistics, the rate of accidents and injuries at work in developing countries is more than 10 times [2] than that in developed countries.

Due to economic situations, demography, tradition, religion, political systems and other cultural and social conditions, technology transfer from developed countries has not led to significant improvement of ergonomics in developing countries. Poor working conditions, the absence of effective preventive programs for work-related injuries and discomfort in many workplaces in developing countries have resulted in high sickness and accident rates, causing not only enormous human suffering, but also loss of productivity and product quality.

When technology is transferred from developed countries to developing countries, due to the reasons of cost, lack of demand and lack of knowledge by the recipient countries, companies very seldom take the initiative to adapt their technology to the conditions of the recipient's countries. The rapid changes in many developing countries do not allow for the establishment of appropriate and compatible social and technical infrastructure to deal with the consequent problems.

In some cases, the old technology and worn-out machinery are still used in many developing countries. This machinery creates many work environment and production problems. On the other hand, due to the economic constraints in some countries, national policies even encourage the importation of second-hand technology through tax reduction measures [2].

- **Ergonomics in China**

In China, the early work on ergonomics can be traced to the 1930s. Professor Li Chen, Honorary President of Hangzhou University was a pioneer in this field [3].

During the 1980s, many world famous ergonomists visited China. Frequent lecture exchanges and ergonomics training programs helped China to develop its ergonomics. In 1989, China founded its ergonomic organization, the Chinese Ergonomics Society. In July 1991, this society became a member of the International Ergonomics Association (IEA) [2].

1.1.4 Work-Related Musculoskeletal Disorders (MSDs)

As defined by the International Health Commission, work-related musculoskeletal disorders (MSDs) [4] are disorders and diseases of the musculoskeletal system that have work related causal determinants. They are a group of painful disorders of muscles, tendons, and nerves. Examples are carpal tunnel syndrome, tendonitis, thoracic outlet syndrome, and tension neck syndrome. Frequent and repetitive activities, or activities with awkward postures cause these painful disorders during work or at rest.

Almost all human activities in the work place require the use of the arms and hands. Therefore, most MSDs affect the hands, wrists, elbows, neck, and shoulders. Work using the legs can lead to MSD of the legs, hips, ankles, and feet. Repetitive activities may also cause some back problems.

The National Institute for Occupational Safety and Health (NIOSH) in the US and a scientific committee of the American National Research Council have published a critical review of the epidemiological evidence for MSDs. Both reports established a strong positive relationship between the occurrence of musculoskeletal disorders and the performance of work, with high levels of exposure to work risk factors [4].

1.2 Overview of Expert Systems

1.2.1 Introduction

As defined by Professor Edward Feigenbaum of Stanford University, an expert system is "... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution." [5]. That is, an expert system is a computer system. This type of system emulates the decision-making ability of a human expert.

Since expert systems usually have no algorithmic solution and rely on inferences to achieve a reasonable solution, people usually make expert system programs in a very different way from conventional programs.

People use expert systems to explain to the user the line of reasoning that led to the solution of a problem or the desired advice.

1.2.2 Historical Review of Expert Systems

Artificial Intelligence (AI) has many branches concerned with speech, vision, robotics, natural language understanding, and learning. Expert systems is one of the AI branches.

The development of expert systems technology can be traced back in 1940s. In early 1943, people obtained post-production rules. The Markov Algorithm for controlling rule execution was invented in 1954; in 1958, LISP AI language appeared; in 1970, work on PROLOG began. The research in expert systems gained substantial growth in the 1970s [5]. The early expert systems were mostly in the field of medical diagnosis. The best-known expert system in medicine is MYCIN, developed in 1973 at Stanford University [5]. In 1977, the OPS expert system shell emerged. In 1979, people implemented Rete Algorithm for fast pattern matching and commercialization of AI began [5].

By the 1980s, new companies started to bring expert systems out of the university laboratory and produced commercial products. Some powerful new software such as the Automated Reasoning Tool (ART) and Rulemaster were introduced by

Knowledge Engineering Tool. In 1985, NASA introduced CLIPS [5].

1.2.3 Knowledge Based Expert Systems

Expert systems, a branch of AI, became a commercially viable solution approach to real-life problems in the beginning of 1980s. Expert systems make extensive use of specialized knowledge to solve problems at the level of a human expert.

The knowledge in expert systems is usually from books, scientific publications, standards, and experts in the field. The inference engine of the expert system draws its conclusions from the knowledge base. These conclusions are the expert systems' response to the user's queries for expertise.

1.2.4 Advantages and Weaknesses of Expert Systems

Expert systems have some unique advantages.

- Reduced cost. Although expert systems are still expensive to build and maintain, they are inexpensive to operate.
- Increased availability. An expert system is the mass production of human expertise. It can be easily distributed in a number of copies, whereas training a new human expert is much more time-consuming and expensive.
- Fast response. Depending on the software and hardware used, it responds faster than a human.
- Steady, increased reliability. Expert systems are likely to perform tasks more consistently than human experts. An expert system will handle similar situations in the same way and make comparable recommendations, whereas humans are influenced by various effects. An expert system can provide permanent documentation of the decision process.

- Intelligent tutor. An expert system may act as an intelligent tutor by letting a student run sample programs and explaining the system's reasoning.
- Easy to improve and modify. In knowledge-based expert systems, for example, people can update the system by adding new rules without changing the basic structure [5].

While expert systems have many advantages, they also have some weaknesses. A practical limitation of many expert systems today is lack of causal knowledge. That is, expert systems do not really understand the underlying causes and effects in a system. It is comparatively easier to program expert systems with shallow knowledge based on empirical and heuristic knowledge than with deep knowledge based on the basic structure, function, and behavior of objects. Another problem with expert systems today is that their expertise is limited to the knowledge domain that the systems know about. Unlike people, typical expert systems cannot generalize their knowledge by using analogy to reason about new situations. Although rule induction helps, only limited types of knowledge can be put into an expert system this way [5].

1.3 Summary

In this Chapter, ergonomics and expert systems were briefly introduced. These two disciplines emerged in 20th century. The introduction in this chapter presents a bird's eye-view of the two subjects.

- **Research Motivations**

Ergonomics has everything to do with people's lives. Utilizing ergonomics principles in today's society becomes more necessary and important. With more than a half-century of development, ergonomics researchers have created a number of standards and rules to guide people's behavior in work. Those standards and rules may stay in different sources, some are in government documents, and some are within ergonomics experts' personal knowledge to be extracted. When people want to apply ergonomics principles in their work, such as workplace design, they may have to check the documents or consult experts. This works sometimes, can be very complex and expensive. In addition, assistance of experts may not be readily available.

- **Research Contributions**

In this research, we set up a rule-based expert system. In this expert system, we stored ergonomics knowledge in the format of rules. The knowledge base focuses on factors that generate Musculoskeletel Disorder (MSD) hazards. Through the dialogs between the user and the computer, the system draws a solution heuristically. Through the inference, the system works as an expert to consult the user. It notifies the user about the existence of MSD hazards in his/her workplace design efficiently. Therefore, it helps him/her to avoid MSD risk factors.

- **Thesis Organization**

In Chapter 2, we present a detailed literature survey in ergonomics and expert systems. We will review some conventional methods in ergonomics research and implementation of expert systems. In Chapter 3, we present a further introduction of the specific problem and set up the foundation of our system. In Chapter 4, we will address an in-depth analysis of the problem and detailed descriptions of our system. Finally in Chapter 5, we draw the conclusion and make recommendations for future research.

Chapter 2

Literature Review

2.1 Introduction

In the past half-century, numerous scientists contributed much to the research of ergonomics. Research in this field is also becoming more and more specific. Meanwhile, with the assistance of fast development of computer science in hardware and software, the technology of expert systems has matured. The implementation of this technology has spread in many domains. In this chapter, by the literature review, we attempted to find out a way to implement expert system technologies in the area of ergonomics.

2.2. Ergonomics

2.2.1 The Size of the Problem

Although the research and implementation of human factor engineering is improving, people are still facing severe problems of ergonomics around the world. For example, Hilderbrandt [6] carried out a survey in 12 branches of Dutch agriculture. Through a sample population of 2580 male employees and employers, with response rate of 49%, a total of 75% of the employees and 71% of the employers reported musculoskeletal symptoms during the past 12 months. Lower back pain was most prevalent, with a one-year prevalence rate of 51% among employees and 47% among employers, followed by symptoms of the neck-shoulder and knees. Neck-shoulder symptoms have one-year prevalence rate of 35% among employees and 30% among employers. Knee symptoms have a one-year prevalence rate of 22% among employees and 17% among employers.

Work-related MSDs are recognized as leading causes of significant human suffering, loss of productivity, and economic burdens on society. Workers with severe MSDs often face permanent disability that prevents them from returning to their jobs or handling simple, everyday tasks like combing their hair, picking up a baby, or pushing a shopping cart. In the last decade, it is reported that MSDs are the largest single job-related injury and illness problem in the United States, consistently accounting for 34% of all reported injuries and illness [4]. Through OSHA's (Occupational Safety and Healthy Administration, USA) report in 1997, employers are annually paying more than $ (US) 15 billion for MSDs. Other expenses associated with work-related MSDs, such as the costs of training new workers, may increase this total to $ (US) 45 billion a year [4].

2.2.2 Study of Musculoskeletal Disorders (MSDs)

MSDs have been studied extensively through field studies. In order to find a correlation between certain activities and specific MSDs, recently computers have been used to help to design proper work environments and evaluate potential MSDs.

Saldana *et al.* [7] conducted a musculoskeletal discomfort survey to assess musculoskeletal discomforts among rural mail carriers in two post offices. By means of a computerized Discomfort Assessment System (DAS), they collected data about musculoskeletal discomfort directly from the workforce. Through the survey, they determined the number of participating employees who were experiencing some kind of musculoskeletal problem, found out subtasks associated with discomfort, determined the body areas most affected by different subtasks and investigated the patterns of discomfort that occurred with time.

Ranney *et al.* [8] found that in highly repetitive jobs, musculoskeletal disorders in the upper limb are potentially work-related. Through research, they suggested that muscle tissue is highly vulnerable to overuse. Stressors that affect muscle tissue, such as static loading, should be studied in the forearms as well as the shoulder.

Chaffin and Page [9] introduced three case studies about infrequent lifting of varied size boxes from near floor level. The cases included symmetric, sagittal plane lifting using a freestyle posture, a similar lift but with freestyle and squat lift postures, and an asymmetric load lifting task. The study result presented the relation of postural effects on biomechanical and psychophysical weight-lifting limits.

With the popular implementation of computers, scientists began to pay more and more attention to jobs in the office, especially that have something to do with visual display terminals (VDT) [10]. Among office staff with repetitive work using VDTs, musculolsketetal neck-and shoulder disorders are common. To date, office work is mostly mixture of VDT and other tasks. Office workers are spending more and more time at the keyboard. Through studying upper-arm elevation during office work, people found that, although people receive other office work-tasks besides VDT work, they don't achieve an improvement in arm postures or in neck-and-shoulder disorders. In order to decrease such disorders, people need to apply more extensive changes in arm postures provided by new work tasks [10].

By examining 260 VDT workers, Bergqvist *et al.* [10] found several factors related to the individual to be important in relation to musculoskeletal problems. Those factors include age, children at home for women, and stomach-related stress reactions. Limited rest break opportunity appeared to be a major factor for several muscular problems. Limited or extensive peak contacts were also associated with musculoskeletal problems. Neck and shoulder problems were associated with posture factors. Arm/hand problems were associated with hand and keyboard position and non-use of lower arm support. The findings provide examples of important factors for musculoskeletal problems of VDT operators.

With more than a half-century development, people have summarized some effective research methodologies for ergonomics research. The most basic methodology is statistical analysis and experimentation. Through the experiments, researchers acquire data for analysis. Analyzing and summarizing these data generates conclusions.

When investigating the factors that influence musculoskeletal disorders among VDT workers, researchers gathered a group of 260 VDT workers [10]. Seventy-six percent of the workers are women. Their common types of VDT jobs involved extensive numeral input, data acquisition, sometimes with limited numerical and text input and extensive word processing. Subsequently, researchers conducted a worksite investigation at the work place with data on each individual's most common work situation. The data covered 88% of the 260 VDT users. Ninety-seven percent of the 260 VDT workers participated a physiotherapeutic examination. The physiotherapeutic examination included short case history of aches, stiffness, tiredness, paresis, and numbness in the neck, shoulders, elbows, and hands. Meanwhile, the subjects accepted active range of motion measurement by using a compass goniometer for the neck, and functional tests for shoulder joints, elbows, and hands. Muscle function tests for isometric manual resistance were performed with the subject in a sitting posture. Muscles and muscular attachments in the neck, shoulders, and arms were palpated, and pain noted [10]. Also, investigators selected some diagnostic endpoints, such as tension neck syndrome diagnosis, cervical diagnoses, shoulder diagnoses and arm/hand diagnoses. Combining with the effect measurements, people analyzed all individual, organizational and ergonomic factors to draw conclusions.

With fast technology development, people began to apply new facilities to aid their research work. When probing the effects of a negative slope keyboard system and full motion forearm supports, Hedge and Powers [11] applied video-motion analysis and computer data processing. While subjects were typing the keyboard during the experiment, researchers used an ultrasonic sensor coupled to a microprocessor mounted to the side of the computer monitor to measure their body to screen distance continuously. At the same time, they used one video camera to record wrist posture and another video camera to record vertical hand/wrist/forearm positions. Videotapes were encoded automatically with a time base and frame numbers, and then digitized using a personal computer connected to a videotape player capable of playing each tape through frame by frame with stable images. The video-motion software tracked the frame-by-frame digitizing and after a few frames constructed a predicted pattern of movement. Prior to analysis, raw data files were also processed by the software using an interpolating algorithm to smooth out any discrepancies in digitizing accuracy [11]. These tasks were used to evaluate MSDs.

In recent years, computer technology began to occupy a more and more important position in ergonomics research. Saldana *et al.* [7] collected data about musculoskeletal discomfort directly from the workforce by means of a computerized discomfort assessment system (DAS). By this way, they aimed to determine subtasks associated with discomfort, the body areas most affected by different subtasks and to investigate the patterns of discomfort that occurred with time.

Biomechanical modeling can be used to analyze the situation of people's body while working. In Chaffin and Page's work, they applied computerized biomechanical models. The models provided them a complete understanding of the pathophysiology of a person's spinal column and supporting structures, thereby provided a means to predict the risk of tissue trauma in given lifting situations. The computer model they used is an integration of two sub-models, the whole-body kinetic model and the 3D Torso Model [9]. The whole-body kinetic model is a 12-link representation of major body segments. It uses Newtonian static analysis to compute the 3D moments and external forces. The 3D Torso Model provides a geometric description of the bones, muscles and ligaments of the human torso when placed in a variety of postures and for people of different anthropometry. The results of these geometric studies provide a systematic method for predicting torso muscle and ligament lengths, and the corresponding moment arms. The two models have integrated (with fixed, normal, geometric properties) into a PC software package referred to as 3DSSPP, 3D Static Strength Prediction Program. It allows an analyst to rapidly simulate a variety of manual materials handling activities and predict static spinal compression forces, as well as population strength, body balance, and foot traction requirements [9].

In designing a workstation, people use computer-aided human modeling programs to analyze human-fit to the workstation components. There are some representative programs to illustrate the current state of development. Those programs are CYBERMAN, COMBIMAN, CREW, JACK, SAMMIE and MANNEQUIN [12]. The programs differ considerably in terms of system requirements, operating characteristics, applicability and various ergonomic evaluation functions available in the human modeling programs [12].

Several checklists have been designed to evaluate how much workers are at risk of developing MSDs. These include:

(1) Basic Screening Tool [4],

(2) VDT Workstation checklist [4],

(3) The Job Strain Index [13],

(4) The NIOSH lifting equation [14],

(5) The UAW-GM checklist [15],

(6) The applicable ACGIH threshold limit values for physical agents [16],

(7) The Rapid Entire Body Assessment (REBA) [17],

(8) The Rapid Upper Limb Assessment (RULA) [18],

(9) Appendix B to the final Washington State ergonomics standard (WAC 296-05174) [19],

(10) Snook & Ciriello's tables of maximum weights and forces for manual material handling tasks [20].

These checklists need to be used by experts to interpret the results and recommend appropriate actions. A computer-based expert system could be designed to help with this task. We use some of the checklists mentioned above in our expert system, where they are discussed in detail.

2.3 Expert Systems

2.3.1 Application of Expert Systems

Expert System technology is a tool to assist people to solve problems. Based on the characteristic of the expert systems, the methodologies of developing expert systems are different from writing conventional computer programs. This section reviews a number of expert systems, which have been developed.

Lee and Decker [21] developed a knowledge-based expert system, Design Script. Engineering design requires cooperation among engineering designers to complete a design. Revisions of a design are time consuming, especially if designers work at a distance and have different design description formats. In order to reduce the design cycle, a sharable design description in the engineering community is very necessary. Furthermore, it will be ideal if the description can be electronically transportable. Design Script [21] is a conceptual model of the design process that is based on hierarchical design structure. It shows how to capture design knowledge and integrate data and tools into a knowledge based design system [21].

In today's manufacturing, industrial robots are playing an increasing role in improving production and manufacturing processes. Because there is a very wide range of robot models from numerous vendors, and the purchase of a robot often involves a large capital outlay, selecting a proper robot is very important. Nour *et al.* [22] developed a prototype rule-based expert system for the intelligent selection of robots for manufacturing operations. In this system, technical decisions, economic decisions and acquisition decisions work together as a 3-stage model. In this 3-stage selection process, each stage feeds its output into the next stage until a final robot combination is generated. Therefore, the user of the system can get an optimized solution of the selection of the robot.

For Multiple Sclerosis patients, the prescription of wheelchairs involves many complicated factors such as ambulation status, length of diagnosis, funding sources etc. However, very few experts exist in this area. For this reason, researchers developed an expert system to assist medical therapists to make wheelchair selection decisions [23]. The system also serves as a diagnostic, classification, prescription and training tool in the field of Multiple Sclerosis.

When preparing a medical malpractice lawsuit, an attorney must identify the relevant facts and use them to decide first if the case has merit. Usually, the attorney consults a medical expert to evaluate the client's medical records and to advise the attorney. The problem is both for attorneys and clients; medical experts are both expensive and relatively scarce. The problem of determining fault is tedious and time consuming, and the caseload is growing. Lewandowski developed [24] an expert system called Expert Witness, which will solve this problem. The output of the system is a narrative transcript containing important data, immediate conclusions from the data and overall of the case. The attorney and medical expert usually use the case to make decisions about whether and how to proceed with the case.

2.3.2 Knowledge Acquisition

Knowledge acquisition is the process of extracting, structuring, and organizing knowledge from several knowledge sources [25]. The knowledge sources are usually human experts so that the problem-solving expertise can be captured and transformed into a computer-readable form. Knowledge is the most important component of expert systems. Without explicitly represented knowledge, an expert system is no more than a computer program.

There are three primary concerns of the knowledge acquisition task: the involvement of appropriate human resources; the employment of proper techniques to elicit knowledge; and a structured approach to performing the knowledge acquisition task [5].

In the famous MYCIN system, the rules are acquired from the experts. For example, the first 150 rules were determined over several months of meetings during which the collaborators discussed representative case histories [26]. Those rules were coded into LISP by hand and provide the core knowledge base. With the ability to acquire new knowledge of MYCIN, once an expert has determined what information is needed by the program, he/she will enter new rules into the system. Once new rules are acquired from the experts, they immediately become available for use by the system.

In the expert system Wheelchair_Advisor [23], people collected information both from the therapist's standpoints and the patient's standpoints. Meanwhile, they stored the current models of wheelchairs in the database. Additionally, the designers considered the therapists' years of the experience of prescription of numerous of wheelchairs as part of the system's knowledge. To obtain this knowledge, the designers and knowledge engineers interviewed experts in the field. Insurance letters and other prescription forms supplied the knowledge engineers with the missing links in the pieces of knowledge gained from the interviews. A third way for building the knowledge base is prototyping. The interview went side by side with an actual prototype developed to foster better communication between the expert and knowledge engineers. This helped offset some of the limitations of the interviewing process. Each subsequent version of the prototype provided a chance for the expert to "endorse" the knowledge engineers' interpretation of the knowledge supplied in the previous interview. At times the expert would clarify a previous answer and supply a new one; thus it became clear that the prototype helped correct errors in communication and misinterpretations, then build a better expert system.

2.3.3 Expert System Languages and Tools

Popular languages for building expert systems include LISP, PROLOG, C and C++. In the United States, LISP was the language of choice. Though powerful in its symbolic processing capability, it was found to be difficult to master. Researchers in the United Kingdom and Japan adopted PROLOG for developing intelligent programs. Based in a formal well-understood logic, PROLOG offers a language to develop exact deductive programs. People began to use C or C++ to develop expert systems only recently. CLIPS is a powerful tool for rule based expert systems first used by NASA in 1985 [5].

Ortiz and Hasan introduced expert system technologies for space shuttle decision support [27]. They presented preliminary results of some ongoing software development projects aimed at exploiting CLIPS technology in NASA Johnson Space Center (NASA/JSC). In the Mission Control Center (MCC) in NASA/JSC, there are networked workstations for acquiring and sharing data through NASA/JSC-developed Information Sharing Protocol (ISP). Their paper outlined some approaches taken to integrate CLIPS and ISP in order to permit the development of intelligent data analysis applications.

The expert system for intelligent selection of robots for manufacturing operations developed by Nour, is a rule-based expert system using the CLIPS expert system shell. Lee and Decker used CLIPS with a Windows NT graphical user interface for their expert system [21].

Jung and Biegel designed and expert system for Intelligent Individualized Instruction (I3) [28]. CLIPS modules and classes were utilized for modular design and inter module communications. CLIPS facts and rules were used to represent system components in the knowledge base. CLIPS provides an inference mechanism to allow the I3 system to solve problems [28].

We also found evidence of CLIPS application in the medical field. Expert Witness, an expert system with over 600 CLIPS rules, was designed with the integration of C and CLIPS.

2.3.4 The Problem Domain and Implementation

- **MYCIN**

Many expert systems are in the form of rule-based systems. Among the tremendous fields of expert systems applications, the implementation in the diagnosis of an illness offered a successful contribution. MYCIN expert system is a remarkable one. This expert system is a milestone for the development of AI for several reasons. First, it demonstrated the use of AI for rule-world problems. Second, MYCIN applied some new concepts such as an explanation facility, automatic acquisition of knowledge and intelligent tutoring. These techniques are found in many expert systems today. Third, MYCIN demonstrated the feasibility of an expert system shell [5].

Shortliffe [26] introduced the famous rule-based expert system-MYCIN in his book. MYCIN's task is to assist with the decisions involved in the selection of appropriate therapy for patients with infections. It contains considerable medical expertise. The system can show us the rules and assumptions upon which each decision is based.

- **System Organization**

The work on MYCIN started early in 1972. This expert system has approximately 200 decision rules [26]. These rules construct the knowledge base of MYCIN system. The rules are not explicitly linked in a decision tree or reasoning network. In this way, the system knowledge is modular and manipulable. However, rules are subject to categorization in accordance with the context-types for which they are most appropriately invoked. In contrast to knowledge base, MYCIN uses two kinds of data as its "data base". Information about the patient under consideration is termed "patient data". These data are entered by the physician in response to computer-generated questions during the consultation. Another type of data is called "dynamic data". Those data are created by MYCIN during consultation.

The program itself consists of three subcomponents. Subprogram 1 is the Construction System. It asks questions, makes conclusions and gives advice. In this subcomponent, each probability statement provides information in an explicit rule format: $P(h|e)=X$, meaning that if e is known to be true then conclude that h is true with probability X. The rules in MYCIN system consist of a PREMISE, and ACTION, and sometimes an ELSE clause. The rules are stored as a LISP data structure and categorized by context.

Subprogram 2 is the Explanation System. It answers questions from the user and attempts to explain its advice. Subprogram 2 expects the user to guide the interaction.

Subprogram 3 is the Rule Acquisition System. This module permits experts to teach MYCIN new decision rules or to alter pre-existing rules that are judged to be inadequate or incorrect.

- **Acceptability Criteria**

Modularity to insure straightforward modification

In MYCIN, people stored all information in decision rules. These rules are coded in LISP internally, but can be translated into an English language version for communication with the user. Since the knowledge base is constructed by clusters of rules and there is no explicit relations from one rule to the other in the system, it is easy to accomplish modification of the system.

Ability to acquire new knowledge from experts

It is easy to acquire new knowledge from the experts. For example, once an expert has determined what information is needed by the program, he/she enters the Rule-Acquisition System and enters a new rule.

Ability to understand questions

MYCIN answers questions about its decisions by retrieving and printing out the relevant rules. The system has the ability to understand questions. Physician may ask (1) informational questions, (2) questions about the deductions of the current consultation, (3) general questions about any of MYCIN's judgmental rules, (4) to explain questions in retrospect and (5) to ask for confirmation of one's own decision rules [26].

Ability to explain decisions

MYCIN has two additional explanatory capabilities. One of these allows the user to enter "QA" in response to any question that is asked by MYCIN. The second capability permits the user to demand that MYCIN justifies any question that is asked. Whenever a question generated by MYCIN puzzles the physician, he/she enters the word RULE. Then the program responds by printing out the translation of the decision rule that has generated the current question. Also, MYCIN has the WHY option. It provides a more detailed and conversational explanation of the program's reasoning.

- **Medical Expert Systems Programmed in CLIPS**

The medical field was one of the first testing grounds for Expert System technology. MYCIN is often cited as a classic expert system in this field. Besides MYCIN, there are some other famous medical expert systems such as NURSExpert, CENTAUR, DIAGNOSER, MEDI and GUIDON, MEDICS, and DiagFH [23]. Two important systems: Wheelchair_Advisor [23] and Expert Witness [24] are implemented in CLIPS.

- **An Expert System for Wheelchair Selection - Wheelchair_Advisor**

Wheelchair_Advisor is a prototype system programmed in CLIPS. It serves as a diagnosis, classification, prescription, and training tool in the Multiple Sclerosis (MS) field.

When building the system, the designers considered the patient's needs and constraints. They stored the data about a particular patient and provided the data on line or by using an input text file. By examination, the expert system searches the database of the wheelchairs and finds out the proper wheelchair. IF/THEN rules are used for this task.

The wheelchair database contains a list of wheelchairs with different features. There is an explanation facility to explain the reasoning of the system to the user and there is a solution set module to give the recommendations of the expert system.

- **An Expert System for Developing Expert Medical Testimony - Expert Witness**

Expert Witness in an expert system designed to assist attorneys and medical experts in determining the merit of medical malpractice claims in the area of obstetrics. This system is a narrative transcript. It contains important data, immediate conclusions from the data, and overall conclusions of the case that the attorney and medical expert use to make decisions. The transcript may also contain directives for gathering additional information needed for the case.

This system is implemented using over 600 CLIPS rules together with a C-based user interface [24]. It builds up a patient file cyclically. Within each cycle there are two stages, data collection and data inference. For data collection, once all known information is provided, the inference phase begins. The system analyzes the known data and draws conclusions. When there is more information needed, additional data are suggested in the transcript. If the medical expert wants the system to obtain more information, the next cycle of data collection/inference process works and it allows direct entry of any additional information, and produces a more complete narration.

The inference part of the system is written in CLIPS 4.3 [24]. The 600 rules constitute the knowledge base. The basic architecture is an elaboration of the heuristic classification model [24]. Based on CLIPS rule matching, the system determines the first level of solutions in the form of direct conclusions in the narrative transcript. Based on initial level, additional reasoning is performed to produce the next level of conclusions. Expert Witness uses a mix of reasoning methods. It uses some data to strengthen conclusions and some data to weaken conclusions. Since using basic CLIPS is adequate for the conservative reasoning, it does not use certainty factors or other approximate reasoning methods. Factoring of the rule base into a number of independent subsystems determines the first level of conclusions. After the first level of conclusions have been made, the second level of conclusions come out by using a rule base.

2.3.5 Coding Strategy and Inference Method

Since the birth of Artificial Intelligence (AI), people are applying expert systems in a number of fields. Coding expert system programs is somewhat different from the work of conventional programs. Expert systems may use different coding strategies and methods of inference to process questions.

In the Intelligent Individualized Instruction (I3) system [28], the designers used CLIPS to build the system. For easy development, maintenance and possible expansion modular design was used to build the system. CLIPS modules and classes were utilized for modular design and inter module communication.

In the Wheelchair_Advisor expert system [23], the designers used interactive approach to develop the system. In the prototype of the system, the patient's needs and constraints are considered first. These data can be provided on line or by an input text file. In the input text file, the data about a particular patient is stored. Implementing a number of IF/THEN rules can accomplish this task. The result of this examination is set as a template of facts about the patient in question. Then the search module uses the facts to search the wheelchair database to find the appropriate wheelchair(s).

Expert Witness [24], an expert system for developing expert medical testimony, builds up a patient file cyclically. Within each cycle there are two stages: data collection and data inference. For data collection, once all known information is provided, the inference phase begins. The system analyzes the known data and draw conclusions. When more information is needed, additional data are suggested in the transcript.

The inference part of the system is written in CLIPS 4.3 [24]. The 600 rules constitute the knowledge base. The basic architecture is an elaboration of the heuristic classification model [24]. Based on CLIPS rule matching, the system determines the first level of solutions in the form of direct conclusions in the narrative transcript. Based on initial level, additional reasoning is performed to produce the next level of conclusions. Expert Witness uses a mix of reasoning methods. Since using basic CLIPS is adequate for the conservative reasoning, it does not use certainty factors or other approximate reasoning methods. Factoring of the rule base into a number of independent subsystems determines the first level of conclusions. After the first level of conclusions have been made, the second level of conclusions come out by using a rule base. It uses forward chaining production and matching.

2.4 Summary

In the past decades, work-related musculoskeletal disorders (MSDs) are problems that most ergonomists are interested in. Research in MSD has achieved great success.

MYCIN is a classical expert system in the medical field. CLIPS is a relatively new tool for expert systems development. It has been implemented in many fields to build expert systems. In this chapter, we reviewed some successful examples in medical field and several valuable techniques to implement this tool.

Concerning the current state of ergonomics and the problems of MSD, we realized it is very important and necessary to conduct research in this area. In addition, the techniques of expert systems provide us a new way to attempt solve problems of

MSDs. The records of successful implementation of CLIPS predict the possibility of successful application of CLIPS in our system.

Chapter 3

Expert Systems and Ergonomic System Design

3.1 Introduction

A number of factors must be considered in the developing of an expert system. These factors include problem selection, expert systems tool selection, cost, and benefits. Both the managerial and technical aspects must be considered in building a successful system.

3.2 The Expert System Approach

3.2.1 Problem Selection

- **The Problem**

Before building an expert system, we must select an appropriate problem. Like any software projects, there are a number of decisions that should be made before major commitment of people, resources and time to develop the system.

As we discussed in the previous chapters, work-related musculoskeletal disorder (MSD) is a very important problem in ergonomics workplace design. MSDs occur most often when the physical demands of work exceed worker capacity and cause preventable wear on the body. Symptoms include pain, motor weakness, sensory deficits and restricted ranges of motion. For example, applying excessive force, lifting heavy loads, working in awkward postures or performing certain repetitive motions over extended periods of time can lead to injury.

- **The Importance to Solve the Problem**

In 1997, more than 626,000 lost workdays were reported in the US due to MSD injuries and illnesses [4]. A critical review by NOISH (National Institute for Occupational Safety and Health, USA) shows more than 600 epidemiological studies addressing the effects of exposure to workplace risk factors [4]. According to OSHA's (Occupational Safety and Health Administration) report, more than 2000 articles on work-related MSDs and work place risk factors have been published [4].

If we properly follow ergonomic principles in workplace design, we can achieve significant benefits. A 1997 General Accounting Office report on companies with ergonomics programs presented many successful cases [4]. Well-managed ergonomics programs have achieved significant reductions in the severity and number of work-related MSDs in the past 30 years. These programs have also improved productivity and employee morale and reduce employee turnover and absenteeism [4].

Much evidence strongly supports two basic conclusions: (1) There is a positive relationship between work-related MSDs and employee exposure to workplace risk factors, and (2) ergonomics programs and specific ergonomic interventions can substantially reduce the number and severity of these injuries [4].

- **Possible Methods to Solve the Problem**

Ergonomics experiments have been the primary method of study for solving the problem. Through experiments, researchers develop ergonomic principles to obey in order to reduce the risk of MSDs.

Ergonomics experts provide guidelines for workplace design. Training, education, and ergonomics programs are all effective methods to reduce MSD risks. There are a lot of accessible sources for people to get information about ergonomics. For example, there are many publications; informational materials and training courses are available from OSHA through Regional Offices (USA), OSHA-sponsored educational centers, OSHA's state consultation programs, etc.

3.2.2 OSHA's Ergonomics Standard

- **Introduction**

On November 14, 2000, OSHA published its ergonomics standard, which took effect on January 16, 2001. The requirements of OSHA's Ergonomics rules apply to all general industry employers in the US, about 102 million workers at 6.1 million worksites [4]. On March 20, 2001 President George Bush repealed this standard. "Today I have signed into law S.J. Res. 6, a measure that repeals an unduly burdensome and overly broad regulation dealing with ergonomics ... The safety and health of our Nation's workforce is a priority for my Administration. Together we will pursue a comprehensive approach to ergonomics that addresses the concerns surrounding the ergonomics rule repealed today. We will work with the Congress, the business community, and our Nation's workers to address this important issue." [29] Although this standard is no longer in effect, it does contain valuable information, which can be used to analyze a work environment and recommend actions to be taken to improve the work environment.

The purpose of OSHA's standard is to reduce the number and severity of MSDs caused by occupational exposure to ergonomic risk factors (also called "ergonomic stress") on the job. The standard required employers to implement an ergonomics program to address risk factors in jobs that pose an MSD hazard to the employees in those jobs.

In this standard, OSHA used substantial experience with ergonomics programs, experience of private firms and insurance companies, and results of research studies conducted during the past 30 years. Those experiences clearly show that:

(1) Ergonomics programs are an effective way to reduce occupational MSDs;

(2) Ergonomics programs have consistently achieved that objective;

(3) OSHA's standard is consistent with these programs;

(4) The standard has firm ground in the OSH Act and OSHA policies and experience [4].

The standard provides employers with tools that consist of checklists of the ergonomic risks as follows [4]:

1. **Repetition** - high repetition rate for the same movements for at least 2 hours at a time; or, using a high repetition device (e.g. keyboard, mouse) fore more than 4 hours a day.

2. **Force** -any lift of more than 333 Newton (75 pounds); any pushing/pulling of more than 88 Newton (20 pounds) of initial force for more than 2 hours per day.

3. **Posture** - repeated working in a deviated body posture (e.g. bent neck, back, wrists, arms above the head etc.) for more than 2 hours per day.

4. **Contact stress** - applying contact force with a body part more than 10 times per hour for more than 2 hours per day (e.g. using the hand or knee as a hammer).

5. **Vibration** - using hand tools with high vibration levels for more than 30 minutes

per day; using hand tools with moderate vibration levels for more than 2 hours per day.

OSHA predicted that over the first 10 years of the standard's implementation, more than 3 million lost workdays due to MSDs will be prevented in general industry. In addition to better safety and health for workers, the standard was expected to save employers money, improve product quality, and reduce employee turnover and absenteeism [4].

- **Scope of Coverage**

This standard was developed to apply to general industry employment, which means all employment except for railroads and employment covered by OSHA's agriculture, construction, and maritime standards. This standard did not cover general industry work performed incidentally to or in support of construction, maritime, or agricultural employment or railroad operations. Although this standard applied to general industries, its coverage was further limited to general industry manufacturing jobs, manual handling jobs, and jobs with MSDs.

This standard covered MSDs affecting the neck, shoulder, elbow, forearm, wrist, hand, back, knee, ankle, and foot as well as abdominal hernias. It did not cover eye disorders, even when associated with jobs involving computer monitors.

- **Applications**

The rules in this standard incorporate a two-stage action trigger. A job meets the action trigger in the final standard based on two criteria. The first is what has been called the "single-incident trigger" [4]. Under this criterion, an employee working in the job must have incurred either a work-related MSD severe enough to result in a work restriction, medical treatment beyond first aid, or MSD signs or symptoms lasting at least seven consecutive days after being reported to the employer [4]. The second step of the action trigger must only be addressed after an MSD incident occurs. It is based on the employee's exposures to ergonomic risk factors. If the employee is exposed to one or more of the risk factors described in the Basic Screening Tool for longer than the time listed for that risk factor, then the job meets the screen. If a job does not meet the requirements of the Action Trigger, no further employer intervention is required [4].

3.2.3 A Rule Based Expert System

The ergonomics checklists mentioned before can best be represented by rules. Fortunately, among expert systems, rule-based expert systems are the most frequently used. The knowledge is stored in the expert system in the form of rules. They are the essential elements and represent the knowledge in this type of expert systems. The rules are sequenced in the succession of logical thinking (*if ... then*) and may point at a jump in the sequence (*else -> go to).* In an expert system the rules may have very complex relations.

3.3 Motivations

There is strong evidence of the positive relationship between work-related MSDs and employee exposure to work place risk factors. Meanwhile, it is believable that ergonomics programs and specific ergonomic interventions can substantially reduce the number and severity of these injuries.

In recent years, both employers and employees have become very aware of the connection between workplace risk factors and MSDs. People are pursuing effective ways to apply ergonomics principles in their work in order to reduce lost and obtain benefits. From the lessons in proceeding chapters, we learned that employers with effective, well-managed ergonomics programs can achieve significant reductions in the severity and number of MSDs. OSHA's standard provided a very good ergonomics program for this aim in workplace design. However, the question is how to properly and correctly apply the standard? Although OSHA has adopted many efforts to let its users utilize the standard properly, the correct and efficient application is still a very hard task. For example, the description and explanation of the rules in the Federal Register is over 600 pages thick, to check the needed information from it is a very time-costing job. Very possibly, he/she needs help from an ergonomic expert while looking up the document. However, in some certain cases, experts may not be available when needed.

In view of such a complicated mission, sophisticated computer technology and modern developed expert systems open a gate to us to solve the problem. In order to apply the standard in an efficient, correct and suitable way, it is very worthy of adopting a rule-based expert system for ergonomics workplace design.

3.4 Size of the System

The objective of this research is to develop a rule-based expert system for ergonomics workplace analysis and design. The knowledge source is the rules from part of OSHA's standard. It uses basic structure and functions of expert system techniques. The system runs on microcomputers. The number of rules in this system is about 100.

3.5 Tools Selection

It is extremely important to choose a right expert system building tool. For solving simple problems, it is feasible to develop an expert system by using a language such as LISP or Prolog. For more complex systems, it is usually advantageous to use an expert system building shell. Those building tools provide a predefined structure for knowledge storage and retrieval in the application domain. Using these tools can significantly reduce development time and effort.

3.5.1 CLIPS

The C Language Integrated Production System (CLIPS) was invented at NASA's Johnson Space Center in 1985 [30]. Although LISP was one of the main expert systems languages at that time, it still had problems, such as: low availability on a wide variety of conventional computers, high cost of state-of-the-art tools and hardware, and poor integration with other languages.

From mid 1980s to the late 1990s, CLIPS had developed over 10 versions. Originally, the primary representation methodology in CLIPS was a forward chaining rule-based language on the Rete algorithm. The programming paradigms of the current version of CLIPS are: procedural programming, as found in languages such as C and Ada; and object-oriented programming as found in languages such as the Common Lisp Object System and Smalltalk. CLIPS supports the newly developed and/or enhanced X Windows, MS-DOS, and Macintosh interfaces. It is a fully integrated object/rule pattern matching and supports features for rule-based software engineering. It has C++ compatibility and functions for profiling performance.

3.5.2 Language Characteristics and Inference Engine

CLIPS is not only a computer language designed for writing expert systems but also a complete environment for developing expert systems. It includes features such as an integrated editor and a debugging tool.

A program written in CLIPS may consist of rules, facts, and objects. The inference engine decides which rule should be executed and when it should be executed. A rule-based expert system written in CLIPS is a data-driven program where the facts, and objects, are the data that stimulate execution via the inference engine.

CLIPS differs from procedural languages such as Pascal, Ada, BASIC, FORTRAN, and C. In procedural languages, execution can proceed without data. However, in CLIPS, data are required to cause the execution of rules.

CLIPS is designed to facilitate the development of software to model human knowledge or expertise. There are three ways to represent knowledge in CLIPS:

- Rules primarily intend for heuristic knowledge based on experience.
- Deffunctions and generic functions primarily intend for procedural knowledge.
- Object-oriented programming, also primarily intended for procedural knowledge. CLIPS supports five generally accepted features of object-oriented programming: classes, message-handlers, abstraction, encapsulation, inheritance, and polymorphism. Rules may pattern match on objects and facts.

An expert system designer can develop software using only rules, only objects, or a mixture of objects and rules.

3.6 Advantages and Limitations

3.6.1 Advantages

- **High Efficiency**

High efficiency is an advantage of an expert system. Today's microcomputers can have a high CPU speed of 1G. The running speed of normal size of expert system on such computers is not a problem. Rule-based expert systems infer by the way of communication with the user. To the explicit questions asked by the computer, the users only need to reply by answering "yes" or "no" through the keyboard. In a very short time, people can get distinct answer about a specific problem.

- **High Reliability**

When rules are coded into the expert system, the knowledge base about MSD analysis is relatively steady. Humans are sometimes influenced by various effects, but computers work properly in most time. Under normal conditions, the expert system follows the coded sequence to work and does not make any mistakes. It performs tasks more consistently than human experts.

- **High Availability**

With the help of this expert system, it will be very convenient to implement OSHA's standard for ergonomics workplace design without assistance of an expert. Also, the system could be copied and distributed to many other computers. People can access the system to consult about MSD problems anywhere, anytime.

- **Ease of Development and Modification**

Without changing the basic structure, we can add new rules to enlarge the system. The characteristics of CLIPS tell us that writing new rules into the system is an easy job. With the good compatibility of CLIPS, we can also enrich the functions of the system by integrating to other programming languages.

- **Cost Reduction**

This expert system will bring many cost reduction. First of all, some tasks such as consulting to the ergonomics experts, training in ergonomics, learning the Federal Register for OSHA's standard are eliminated. Second, this expert system is built and can run on microcomputers. Further, the CLIPS expert system tool is provided for free.

3.6.2 Limitations

Although there are many advantages for applying this expert system, there stays some limitations.

- **Knowledge Limitations**

The knowledge of this expert system is limited in the domain of the coded rules. This expert system has no ability to generate rules. It can only provide information from coded rules by the fixed sequence.

- **Function Limitations**

The function of the system is limited by the programming way. It can only work by following the fixed procedure no matter how complex or simple the problem is.

There is no shortcut for inference. Besides, CLIPS also has function limitations. For example, it does not have the function to treat graphs. Therefore, if we don't integrate CLIPS with other programming languages, the system can only provide information in the form of words. This is not a major limitation, as several people have developed graphical interfaces to work with CLIPS. [5]

3.7 Summary

In this chapter, we are assessing the ergonomic design of workplaces by the assistance of an expert system. From the study of OSHA's standard, we determined the domain of the knowledge base in the coming expert system and estimated the size of the system. Through researching the application of OSHA's standard, we obtained an initial idea about the basic structure and imagined the fundamental consulting method in our system. Meanwhile, we realized that to apply OSHA's ergonomics rules in an efficient way is very important. This realization motivated us to build a rule-based expert system for rules application. After a careful selection, we concluded that CLIPS is a feasible expert system tool for our research. Furthermore, we analyzed the advantages and limitations about implementing an expert system for our task. Therefore, the preparation to build a rule-based expert system is accomplished.

Chapter 4
The Expert System Development and Problem Solving

4.1 Introduction

Through the discussion in the Chapter 3, we presented that a rule-based expert system is suitable for us to solve problems of ergonomics workplace design. The inference method for this expert system will be forward chaining. We decided to use OSHA's rules as our knowledge source.

In this chapter, we introduce the complete procedure about how to apply the expert system for ergonomics work place design. Meanwhile, we will explain sub-components of this expert system in detail.

4.2 Knowledge Acquisition

4.2.1 Introduction

In this expert system, OSHA's standard is our knowledge source. In particular, all the rules come from OSHA's Federal Register.

After extracting, structuring, and organizing knowledge from those knowledge sources, we will extract the rules and transform them into a computer-readable form.

4.2.2 The Key Elements of the Rules and Their Application

After carefully studying the rules, we found that the rules have some key elements which should be followed during application.

- The rules apply to employers with "caution zone jobs". Doing the caution zone jobs, an employee's work includes physical risk factors specified in the rules. Caution zone jobs are not prohibited and they may not be hazardous. If there are no physical risk factors above the levels listed in the rule, then the employer is not subject to the rule and no action is required.
- Employers with caution zone jobs must ensure that employees working in or supervising these jobs receive ergonomics awareness education. These employers also must analyze the caution zone jobs to determine if they have hazards. If the analysis finds that work-related MSD hazards are present in the caution zone job, then the employer must reduce hazard of the jobs below the hazardous level or to the degree technologically and economically feasible.
- Employers may choose their own method and criteria for identifying and reducing work-related MSD hazards or may use some OSHA specified criteria.
- Employers must provide and encourage employee's participation.
- Employers may continue to use effective methods of reducing work-related MSD hazards.

In developing the expert system, we let the software imitate this human decision process. The purpose of this expert system is to consult the users and guide them to utilize the ergonomics rules correctly and efficiently.

We obtained our rules, thus our knowledge source, from the Basic Screening Tool and VDT Workstation Checklist of Table W-1 and Appendix D-2 of OSHA's Ergonomics standard [4], as well as Appendix B to the final Washington State ergonomics standard (WAC 296-05174) [19]. Appendix D-2 is a simple checklist to assess the physical activities and layout of workstations with a VDT (video display terminal).

- **Basic Screening Tool**

The Basic Screening Tool checklist was developed to evaluate a range of risk factors in manufacturing jobs (see Appendix A of this thesis). The checklist uses check (v) and star (*) to indicate whether certain activities and conditions are present for less than or more than one-third of the production cycle. The number of checks and stars, in conjunction with the report of an MSD, is used to determine if the job requires further investigation or control action.

As can be seen, the areas of the body this checklist addresses are: Hand/wrists, Forearms/elbows, Shoulders, Neck, Back/Trunk and Legs/knees. This checklist evaluates the following risk factors: Force (including manual handling), Repetition, Awkward Postures (including Static Postures), Vibration, and Contact stress.

- **Washington State Appendix B**

The Washington State Appendix B was developed to determine if jobs that were in the Washington State "caution zone" pose an MSD hazard to employees (see Appendix A of this thesis). The checklist shows physical risk factors and lists duration (from 2 to 6 hours) by body part. If the work activities or conditions apply, the job poses an MSD hazard.

Areas of body that Washington State Appendix B covers are: Shoulders, Neck, Back, Trunk, Knees, Forearms, Wrists, Hands, and Elbows. Risk factors it evaluated: Awkward postures, Force (including manual lifting and high hand force), Repetition, Contact Stress and Vibration.

Examples of jobs that can be analyzed using the Washington State Appendix B include: patient lifting and transfer, assembly and production work, janitorial and maintenance, meatpacking, working in a restaurant, grocery cashier, telephone operator, keyboarding, and solid waste handling.

- **VDT Workstation Analysis**

MSDs associated with computer use are reported in a wide range of industries (e.g., telecommunication, telephone, banking, insurance, catalog and telephone sales, customer service, package delivery service, newspaper) and in businesses of all sizes, including very small establishments. The VDT checklist provides these businesses with an easy and quick way to identify and control hazards in a large number of jobs.

OSHA designed this checklist after considering many examples of computer workstation checklists in the record. The checklist is designed to provide employers with a simple way to identify the five risk factors this standard covers, as the most commonly occur in computer work and workstations.

The checklist provides clear and specific guidance in how the employer can provide or adjust a computer workstation to comply with the standard.

The function of the VDT checklist is to determine if the computer workstation and layout address the risk factors most commonly found in VDT jobs. The analyst using this checklist would talk with and observe the worker(s) while they are at the computer workstation.

OSHA VDT checklist is more flexible than some other checklists in the record because it is risk factor-based rather than equipment-based. In equipment-based checklists, employers get a passing score only if they have purchased and installed particular equipment at each computer workstation. OSHA' s risk factor-based checklist, however, gives employers the flexibility of deciding how to best control the identified hazards. For example, an equipment-based checklist asks employers whether they have provided adjustable height tables, chairs and monitor risers. Since it is possible to have an adjustable table that has been set to the wrong height, the equipment based checklist is insufficient. A risk factor-based checklist, on the other hand, asks questions like: are the employees' heads and necks in straight rather than awkward positions. This means that we also evaluate whether the equipment is used or set properly.

4.2.3 The Rules Analysis

OSHA recommended Ergonomic Job Hazard Tools in Appendix D-1 and Appendix D-2 in its Ergonomics Standard. In this OSHA's standard, the term "MSD incident" means either a Musculoskeletal Disorder that is work-related and involves a work restriction, or requires medical treatment beyond first aid, or involves MSD signs or symptoms that are work-related and persist for 7 or more consecutive days after the employee reports them to the employer [4]. An MSD is a disorder of the soft tissues, specifically of the muscles, nerves, tendons, ligaments, joints, cartilage, blood vessels and spinal discs that is not caused by a slip, trip, fall, or motor vehicle accident [4]. The rules in this standard covers MSDs affecting the neck, shoulder, elbow, forearm, wrist, hand, back, knee, ankle, and foot as well as abdominal hernias. It does not, however, cover eye disorders, even when associated with jobs involving computer monitors. According to OSHA's description, "work-related" means a workplace exposure caused or contributed to a MSD incident or significantly aggravated a pre-existing MSD [4]. The MSD symptoms include pain, numbness, tingling, burning, cramping, and stiffness.

- **Duration**

The Basic Screening Tool contains specific definitions of the risk factors and exposure durations that define a job requiring further analysis. Basically, there are five risk factors covered in the final rule. Those factors are repetition, force, awkward postures, contact stress and vibration. In our system, vibration is not considered. The rules are summarized in charts. In the chart, repetition includes separate description for keyboarding and mouse use. Force is broken down into lifting, pushing/pulling, and pinching and gripping unsupported objects of specified weights. Awkward postures are defined by specific postures.

Each job or task activity also includes a duration and frequency limit. In selecting the duration limit for the risk factors, OSHA based its decision on balancing scientific evidence and the need for the screening tool to be clear. For many items in the chart, the standard chooses 2 hours per day as an exposure duration to trigger job hazard analysis. This is based on relevant epidemiological data contained in the rulemaking record [4]. Many studies in the epidemiological literature clearly demonstrate that the incidence of MSDs increases with increased duration of exposure to certain risk factors or a combination of them [4].

There were also studies that showed increased risk of MSDs associated with exposures of less than 2 hours daily [4]. In using this 2-hour cut point, OSHA does not intend to imply that all workers will experience significant adverse effects after 2 hours or more of exposure. Rather, OSHA is using this cut point in the criteria to give employers guidance. For repetitive motion other than use of a keyboard or mouse, the rule triggers the standard only if the exposure occurs for more than 2 consecutive hours in a workday, as opposed to more than two hours total per day.

The screening tool departs from the 2-hour duration criterion for a few items. These include the following: For the use of keyboard and mouse in a steady manner, the duration is set at 4 hours total per work day; for lifting, the screen sets weight and frequency criteria.

- **Lifting**

The chart contains weight limits and limits on the number of times per day the weight can be lifted. OSHA specifies weight limits lifted from below the knee, above the shoulder and at arm's length. The limits specified are as follows: lifting more than 334 Newton (75 pounds) at any one time; more than 245 Newton (55 pounds) for more than 10 times per day; or more than 111 Newton (25 pounds) below knees; above shoulders, or at arms' length for more than 25 times per day [4].

OSHA finds that heavy, frequent or awkward lifting increase the risks for MSDs. Washington State also has used similar data to support its "caution zone job criteria" for lifting.

- **Pinching or Gripping**

For performing activities such as pinching or gripping unsupported objects, the chart specifies weights of 9 Newton (2 pounds) or more per hand for pinching and 45 Newton (10 pounds) or more per hand for gripping. This is based on research results [4] that increased risks of carpal tunnel syndrome, thumb disorders, shoulder disorders, and nerve abnormalities among workers repetitively pinching objects in the range of 9 Newton (2 pounds) or gripping objects in the range of 45 Newton (10 pounds) [4].

- **Contact Stress**

For contact stress, OSHA has specified a frequency of 10 times per hour when using hand or knee as a hammer. Studies have shown increased risk in MSDs among workers using their hand or knee as a hammer [4]. However, few data are available that quantify the frequency of exposure. Washington Sate chose a value of 10 times per hour for their "caution zone job" criteria.

4.2.4 Constructing Expect System Rules Based on OSHA Standard

The rules in this expert system are from OSHA standard's Table W-1, Appendix D-2 and Washington State Appendix B. In those documents, the rules can be divided into two categories, rules described by words and rules described in procedures.

- **Rules Described by Words**

Rules in the Basic Screening Tool Checklist and VDT Workstation Checklist are in form of items. In each item, a sentence tells the user what should do according to ergonomics principles. Comparing to the actual situation, the user answers yes or no. For example, a question on seating in Appendix D-2, VDT Workstation Checklist asks: "Backrest provides support for employee's lower back?" The user should check the situation and answer yes or no. According to the rule, if the answer is yes, there is no MSD hazard, if the answer is no, there is a MSD hazard. In the expert system, we transfer these rules in the form of dialogs. The working procedure of a rule in the system is shown below:

System asks: "Does the backrest provide support for employee's lower back?"

if the answer is yes

then

there is no MSD hazard risk

    then

    ask the next question

if the answer is no

then there is a MSD hazard risk

    then add one in the MSD calculator

        then ask the next question.

After the rules are organized in this way, they are ready to be coded in the program.

- **Rules Described in Procedures**

An example where a standard presents rules in the form of procedures is checking heavy, frequent or awkward lifting, A total of 5 steps are used to determine the existence of an MSD. Step 1 is to find out the actual weight of objects that the employee lifts; step 2 is to determine the unadjusted weight limit; step 3 is to find the limit reduction modifier; step 4 is to calculate the weight limit and step 5 is to determine whether a MSD hazard exists. For this section, we follow the 5 steps to build the rules. The basic rule structure is like this:

The system acquires information from the user and calculates the Weight Limit. This work is from step 1 to step 4. During this procedure, there are many "*if – then*" rules. For example, in step 2, determine the unadjusted weight limit. From the graph shown in the standard, one of the rules we extracted is: While lifting, if the employee's hand is above shoulder and in the mid range (18cm ~ 30cm) from the body, then based on the standard, the unadjusted weight limit is 15 kg. There are similar rules in step 3 and step 4. In the standard, the units are in English system, we transfer them into metric system. Step 5 determines whether there is an MSD hazard. The rule is shown like below:

If the Actual Weight lifted is greater than the Weight Limit calculated,

then

the lifting is a MSD hazard, it must be reduced below the hazard level

else

the lifting is not a MSD hazard

4.3 The Structure of Our Expert System

4.3.1 Overview of the System

Our expert system runs in a CLIPS environment. It is constructed using two sets of files, batch files and function files. The batch files work as interface to connect every sub-component of the system. The structure of the file organization is shown by the flow diagram in Figure 4.1

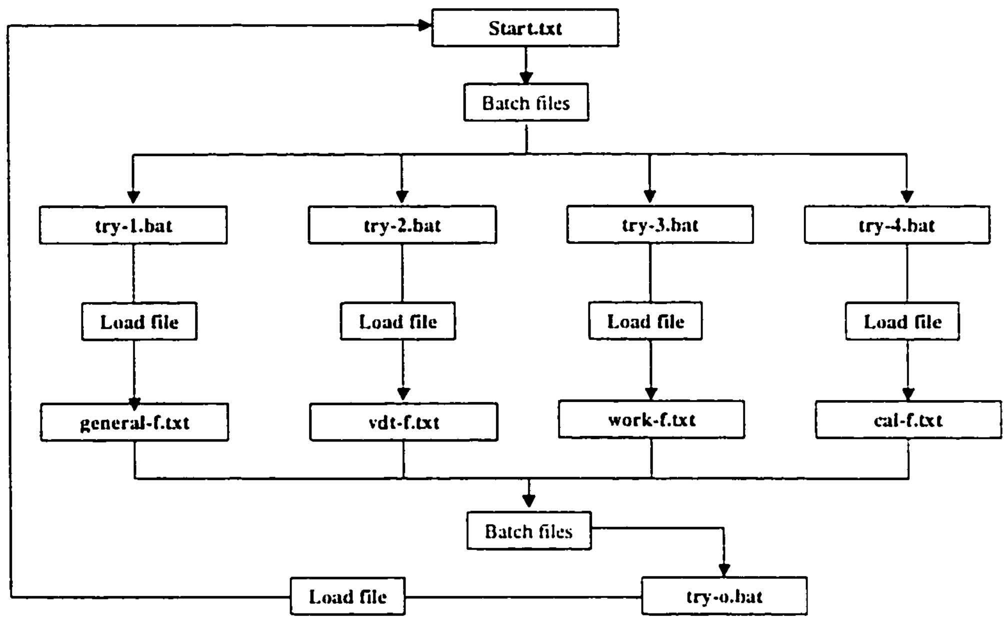


Figure 4.1 Files Organization

When the program is executed, the user needs to answer questions posed by the expert system. The answers can be yes or no, or take a multiple-choice selection, or enter data. Most of the answers to the questions from Basic Screening Tool Checklist and the VDT Workstation Checklist are yes or no. By typing "*y*" or "*n*", the user tells the system the answer, the system executes the rules to generate response.

Multiple-choice menus require the user to select a function. For example, as shown in Figure 4.2, the computer asks the user to make a choice:

****

Welcome to Knowledge-Based Expert System
Basic Screening Tool
You need only review risk factors for those areas of
the body affected by the MSD incident

****

1. Neck/Shoulder;

2. Hand/Wrist/Arm;

3. Back/Trunk/Hip;

4. Leg/Knee/Ankle.

Please select the number to check the part of your body (1/2/3/4) and press Enter

Figure 4.2 Basic Screening Tool Menu

Sometimes, the system needs data from the user. Those data include lifting weight, bending degree and working hours. While the question appears on the screen, the user types the number on the keyboard to inform the system.

This system has 5 sub-systems. These are Management Component, General OSHA rules application, VDT check, Work Shop In-depth Analysis and Lift Limit Calculation. In the following sections, we will introduce each sub-system and other necessary programs respectively.

4.3.2 Management Sub-system

The management component is to manage the process of the whole system. Its main functions include loading a file to start a new subcomponent, switching from one component to another, or leaving the system. The relationship between the management component and other subsystems is shown in Figure 4.3.

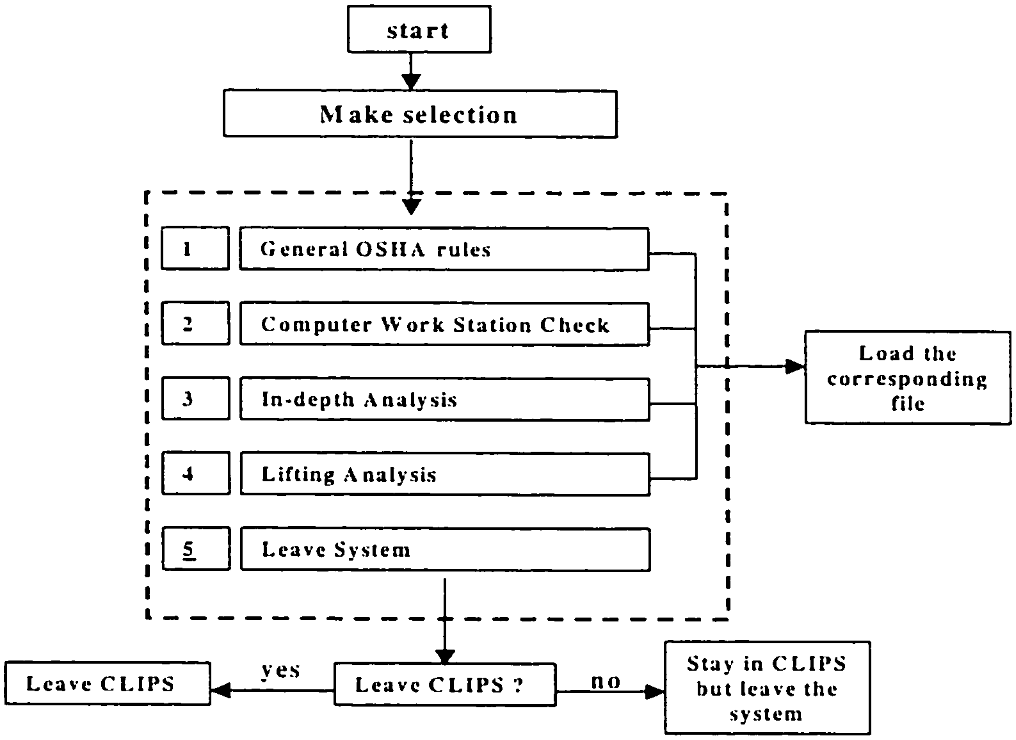


Figure 4.3 Management Sub-system

- **Start**

Whenever the user starts to consult the system, he/she always starts from the file start.txt. In CLIPS running environment, the user should load the file start.txt to let it run. Then it shows a multiple-choice menu on the screen as shown in Figure 4.4.

```
************************************************************************************
*
*       Welcome to Knowledge-Based Expert System
*
*
************************************************************************************
Please select the number of the corresponding item to run the system:

-----------------------------------------------------------------
.
.       1. General OSHA rules
.
.       2. VDT Work Station Check
.
.       3. In-depth Analysis about Work Shop
.
.       4. Lift-limit calculation
.
.       5. Leave System
.
-----------------------------------------------------------------
```

Figure 4.4 Start Menu

From this menu, the user can select options 1, 2, 3, 4 or 5. Option 1 leads to evaluation of working conditions against OSHA rules based on the Basic Screening Tool Checklist. Option 2 leads to VDT Check based on the rules from the VDT Workstation Checklist. Option 3 leads to Work Shop In-depth Analysis based on the rules from Washington State Appendix B. Option 4 leads to Lift-limit Calculation. Option 5 leads the user to leave the system.

- **Batch files**

The batch files connect all the subsystems. In the system, there are 5 batch files: try-0.bat, try-1.bat, try-2.bat, try-3.bat and try-4.bat. Try-0.bat loads the file start.txt, therefore to trigger the whole system. Try-1.bat loads the file general-f.txt. This program applies rules from the Basic Screening Tool checklist. Try-2.bat is to execute file vdt-f.txt. Programs in vdt-f.txt apply rules from Appendix D-2 of OSHA's standard. It checks whether VDT design obeys OSHA regularities. Try-3.bat loads the file work-f.txt. Work-f.txt provides in-depth analysis on workplace design. Try-4.bat loads the file cal-f.txt coming into force. Cal-f.txt executes lifting formula. The relation between batch files and the subsystems is shown in Figure 4.1.

4.3.3 General Rules Sub-system

Rules are extracted from OSHA's Basic Screening Tool checklist (see Appendix A in this thesis) and coded in general-f.txt. They are organized as shown in Figure 4.5, following the format of OSHA's Basic Screening Tool checklist.

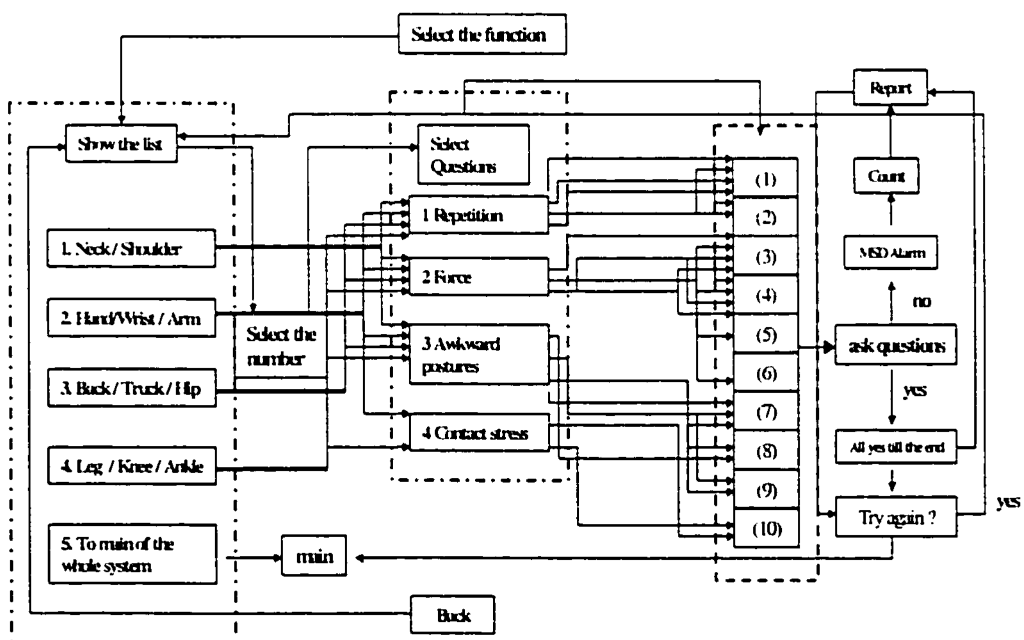


Figure 4.5 Logic Diagram of the General Questions

OSHA's checklists check the situation by asking several questions. If the answer to any one of questions is yes, the system identifies that working condition has a work-related MSD hazard risk.

When this subsystem is executed, the user needs to select the body part to check, as shown in Figure 4.2.

Once the body part is identified, the system will ask further questions to analyze the situation for four basic MSD risk factors. Those factors are: repetition, force, awkward postures and contact stress. For example, when checking Back/Trunk/Hip, the system will ask the user whether he/she has kneeling or squatting for more than 2 hours total per day. If the answer is yes, then the system determines that the user has a MSD hazard.

When it is found the possibility of MSD hazard in the workplace, an alarm will alert the user. The system starts to count the quantity of such risk factors. The number of the risk factors indicates the MSD risk level of the whole workplace design. In the end, the user will be given the information whether MSD hazards exist and the level of the hazards.

When the questions are answered, the user may enter other subsystems to start a new session or leave the system completely.

4.3.4 VDT Workstation Sub-system

Knowledge in this subsystem comes from Appendix D-2 of OSHA's standard (see Appendix A in this thesis). All relevant rules are coded in VDT-f.txt. Rules in this file are organized as shown in Figure 4.6.

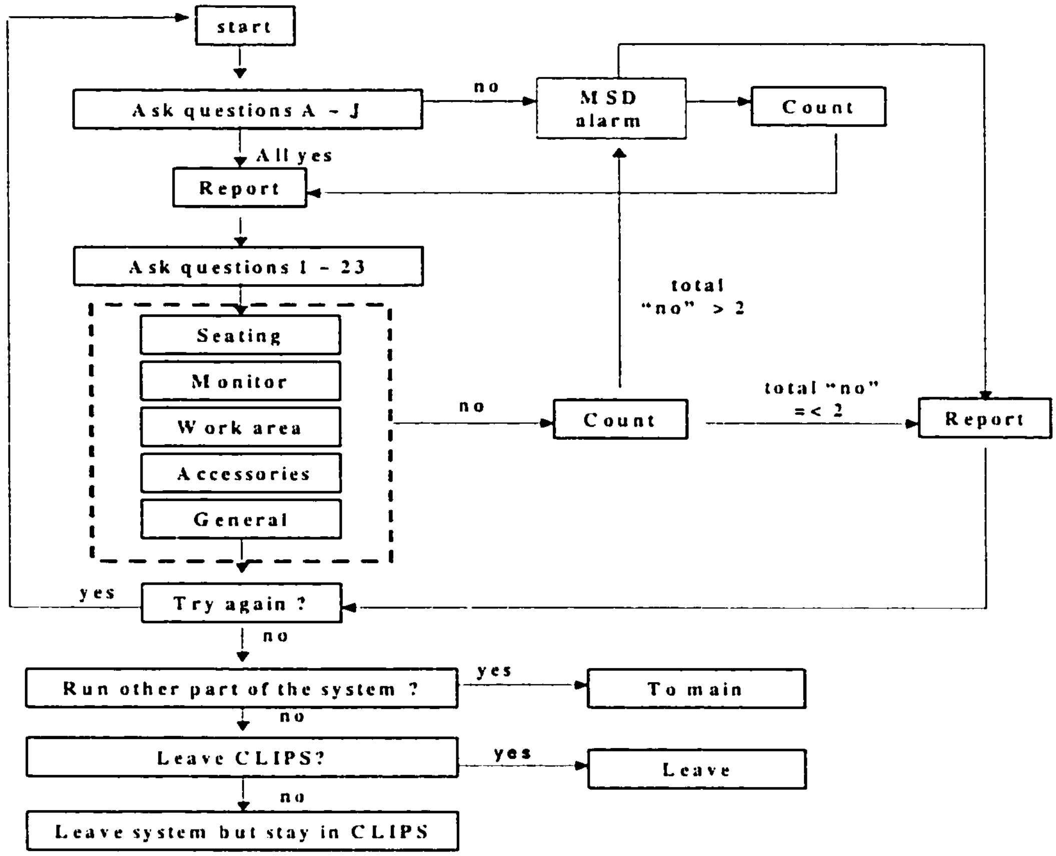


Figure 4.6 VDT Workstation Analysis

When this subsystem is executed, the user will be asked to answer certain questions. According to OSHA's VDT Workstation Checklist, there are two sets of questions. Questions in the first set are critical questions, question A to J in the VDT checklist. These questions are used to check whether the workstation is designed or arranged properly for VDT jobs. If one of the answers from the user shows that the design does not follow these rules, there is a MSD hazard. For example, question A ask whether the user's head and neck are about upright. Answer "yes" means the working condition is acceptable, whereas answer "no" means a MSD hazard exists. In question E, the rule asks whether the worker's upper arms and elbows are close to body (not extended outward). If the answer is "yes", there is no MSD hazard, if the answer is "no", the operation violates the rule and a MSD hazard exists.

Questions in the second set are from Question 1 to 23 in the VDT checklist. Those are not so critical as the questions in the first set. They are related to VDT workstation seating, keyboard/input device, monitor, work area, accessories and other general aspects. For example, question 7 asks whether input device (mouse or trackball) is located right next to keyboard so it can be operated without reaching. Answer "yes" accords to the rule whereas answer "no" violates the rule. According to the standard, if two or more answers violate these rules, there is a MSD hazard. The more rules are violated, the more serious the problem of MSD hazards is. After the section is complete, the user can start a new section or leave the system.

4.3.5 In-depth Analysis System

The In-depth Workshop Analysis subsystem is the most complicated part of the expert system. The knowledge source is from Washington State Appendix B [19]. This sub-system is divided into five blocks: caution zone select block, work situation check block, further situation check block, analyze and report block and, system interface block, as shown in Figure 4.7.

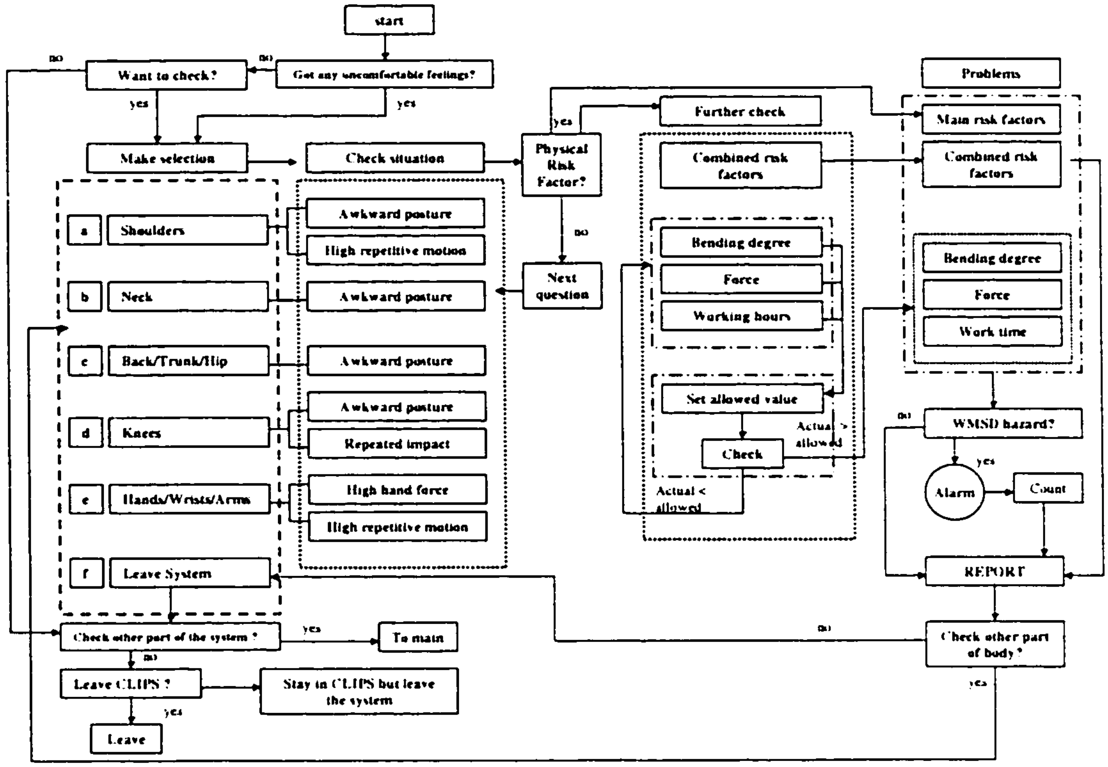


Figure 4.7 In-depth Analysis System

According to Washington State Appendix B, MSDs happen because of several main factors. These are awkward posture, high hand force, high repetitive motion and repeated impact. Each main factor may be constructed from several physical risk factors and their duration. For example, working with hands above the head, with the neck or back bent forward, squatting or kneeling are physical risk factors of awkward posture. Pinching or gripping heavy objects are physical factors of high hand force. Using the same motion with little or no variation every few seconds or intensive keying are physical factors of highly repetitive motion. Using a hand or knee as a hammer more than once per minute are physical risk factors of repeated impact.

Combined risk factors include highly repetitive motion, degree of bending, flexion or extension, highly repetitive motion and ulnar deviation. Each combined factor has its explicit level in the standard.

In this subsystem, each part of human body is considered separately and the system checks whether there is a physical risk factor of MSD. If the physical risk factor exists, the system will check further items such as the duration and combined risk factors. If there is no physical risk factor for the current part, the system will check another part of the human body.

- **Caution Zone Select Block**

In Washington Appendix B, the "caution zone" assessment refers to different parts of human bodies. According to Washington State Appendix B, the "caution zones" are shoulders, neck, back/trunk/hip, knees and hands/wrists/arms. When the "caution zone" select block is executed, the system will ask the user whether there are any uncomfortable feelings. If the answer is "yes", the user will be asked to indicate specific area by entering different options. This process is shown by the two menus in Figures 4.8 and 4.9.

CLIPS> (run)

```
***********************************************************

        Welcome to Knowledge-Based Expert System

          In-depth-analysis about MSD

***********************************************************

 Do you have any of the following uncomfortable feelings (yes/no)?
 ===================================================
      * Painful joints
      * Pain, tingling or numbness in hands or feet
      * Shooting or stabbing pains in arms or legs
      * Swelling or inflammation
      * Burning sensation
      * Pain in wrists, shoulders, forearms, knees
      * Stiffness
  ===================================================

 Do you have any of the above uncomfortable feelings (yes/no)?
```

Figure 4. 8 In-depth Analysis of Uncomfortable Feelings

```
 ______________________________________________

    a. Shoulders;

    b. Neck;

    c. Back/Trunk/Hip;

    d. Knees;

    e. Hands/Wrists/Arms;

    f. MAIN MENU of the system

 ______________________________________________

Please select a, b, c, d, e or f and then press Enter to choose the part of your
check.
```

Figure 4.9 Choose Body Part

- **Physical Risk Factor Check Block**

For shoulders, awkward posture and high repetitive motion are main factors to generate MSD hazards. For neck, back, trunk and hip, the main MSD risk factor is awkward posture. For knees, MSD main risk factors are awkward posture and repeated impact. For hands, wrists and arms, MSD main risk factors are high hand force and high repetitive motion. When this block executes, it checks whether physical risk factors for those main factors exist. If a physical risk factor exists, this block goes to the "further situation check block" for further check.

- **Further Situation Check Block**

This block conducts further analysis of the situation. It checks whether there are any combined risk factors. If there are, it will check further whether some levels of the combined risk factors are below a critical value in the standard. The combined risk factors are bending degree, force and the work duration. When the analysis is complete, the system will enter the next block, the "Analysis and Report Block".

- **Analysis and Report Block**

If the system determines that there is an MSD hazard, it will give an alarm and present a detailed report on the screen. It reports concerned main factors, combined factors and working hours. In reporting bending, force and working hours, the expert system identifies the off-range factors and presents correct values. It will also report the total number of MSD risk factors involved at the final stage of the section.

- **System Interface Block**

After the system has presented a report, it will ask whether the user wants to check other part of the human body. If the answer yes, the system will execute other blocks; if the answer is no, the user will leave the system.

4.3.6 Lift Weight Calculation Sub-system

The Lift Weight Calculation sub-system calculates the lift weight limits and decides whether the employee's lifting weight has MSD hazard risks. The organization of this subsystem is shown in Figure 4.10.

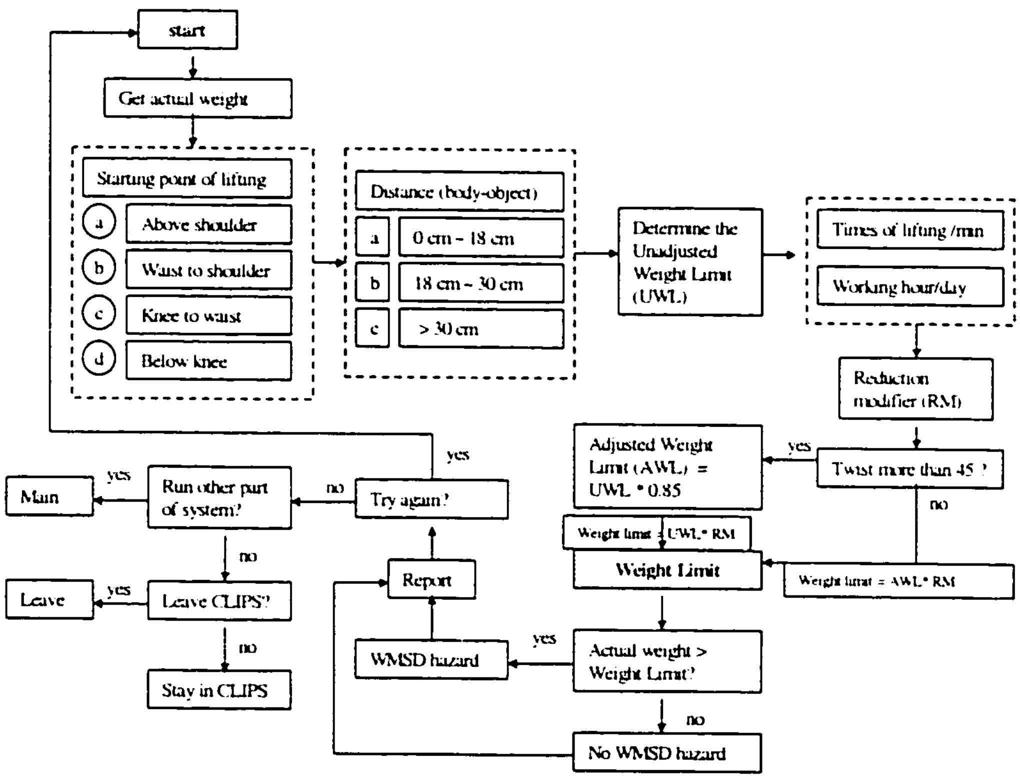


Figure 4.10 Weight Limit Calculation

The knowledge source of this section is from Washington State Appendix B: WAC 296-62-05174 Appendix B: Criteria for analyzing and reducing MSD hazards for employers who choose the Specific Performance Approach [19]. According to the regulations in the standard, the analysis is divided into 5 steps:

- **Step 1**

  Find out the actual weight of object that the employee lifts. When the system executes this step, the system asks the user the actual lifting weight and stores it.

- **Step 2**

  Determine the unadjusted weight limit. The system asks where the employee's hands are and when they begin to lift or lower the object. Based on this position, the system determines the unadjusted weight limit.

- **Step 3**

  Find the limit reduction modifier. At this step, the system has to find out the number of lifts per minute and the total number of hours per day spent lifting. From these two factors, system evaluates the limit reduction modifier.

- **Step 4**

  Calculate the weight limit. Using the information obtained from Steps 2 and 3, and taking account if there is twisting, the system calculates the weight limit.

- **Step 5**

  Make decision. The "expert" in the system compares the weight limit calculated in Step 4 with the actual weight lifted from Step 1 to determine whether there is an MSD hazard.

- **Lift Position Identifier and Evaluation of the Unadjusted Weight Limit**

In the standard, as shown below in Figure 4.11, there is a graph that illustrates the position of the employee's hands when he/she begins to lift or lower the object. There are four levels for vertical positions: above shoulder, waist to shoulder, knee to waist and below knee. Three horizontal ranges: near (0~18cm), mid-range (18 ~ 30cm) and extended (> 30cm). Each combination of vertical position and horizontal range determines an unadjusted weight limit.

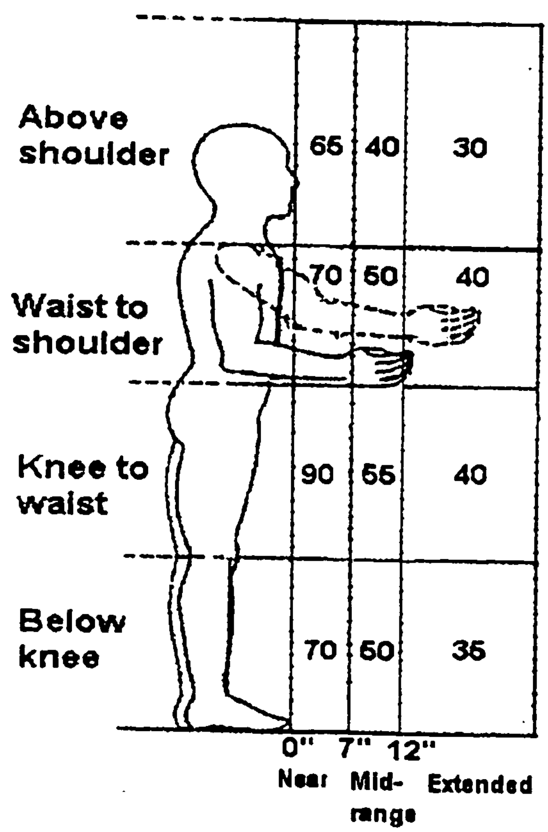


Figure 4.11 Unadjusted Weight Limit [19]

The system executes this function in 3 steps, finds the vertical level first, and then gets the horizontal range. From the result of the previous steps, the system determines the unadjusted weight limit. When the lift position identifier executes, the system will present the user several choices as shown in Figures 4.12 and 4.13.

Where are your hands when you begin to lift or lower the object?

- **a. Above shoulder**
- **b. Waist to shoulder**
- **c. Knee to waist**
- **d. Below knee**

Figure 4.12 Choose Position –1

From options a, b, c, d, the user tells the system the vertical level.

What is the distance between the object and your body?

- **a. 0 cm - 18 cm**
- **b. 18 cm - 30 cm**
- **c. > 30 cm**

Figure 4.13 Choose Position –2

From options a, b, c, the user tells the system the horizontal range. Then, according to the actual lifting position, the expert system finds the unadjusted weight limit. At this point, the system completes the Step 2 of the analysis.

- **Limit Reduction Modifier**

Limit reduction modifier executes Step 3 in the standard. At this step, as shown in Figure 4.14, the standard provides a table for the user to find the limit reduction modifier. In the table, there are times of lift per minute and hours of lifts per day. The numbers inside the table are modifiers for the weight limit reduction.

How many lifts per minute?	**For how many hours per day**		
	1 hr or less	1 hr to 2 hrs	2 hrs or more
1 lift every 2-5 mins.	1.0	0.95	0.85
1 lift every min	0.95	0.9	0.75
2-3 lifts every min	0.9	0.85	0.65
4-5 lifts every min	0.85	0.7	0.45
6-7 lifts every min	0.75	0.5	0.25
8-9 lifts every min	0.6	0.35	0.15
10+ lifts every min	0.3	0.2	0.0

Note: For lifting done less than once every five minutes, use 1.0

Figure 4.14 Limit Reduction Modifier [19]

In this subsystem, the limit reduction modifier finding work is separated in two steps. By selecting the choices shown in Figure 4.15, the user tells the system the number of lifts per minute. In Figure 4.16, the user tells the system the total lifting hours per day.

All the limit reduction modifiers are stored in the system in the form of rules. The rules form is like: *if the number of lifts per minute is x and the number of hours per day is y, then the modifier is z.* For example, if there are 2-3 lifts every minute, and there are 2.5 hours lift per day, then the limit reduction modifier is 0.65. The rule for this in the system is:

If 2-3 lifts every minute

and

2.5 hours of lift per day

then

the limit reduction modifier is 0.65.

In this way, the system finds the corresponding limit reduction modifier.

How many lifts do you have per minute?

a. **less than once every 5 mins**

b. **1 lift every 2-5 mins**

c. **1 lift every min**

d. **2-3 lifts every min**

e. **4-5 lifts every min**

f. **6-7 lifts every min**

g. **8-9 lifts every min**

h. **10+ lifts every min**

Figure 4.15 Choose Lifting Frequency

For how many hours per day?

a. **1 hour or less**

b. **1 hour to 2 hours**

c. **2 hours or more**

Figure 4.16 Choose Lifting Hours

- **Weight Limit Calculator**

This calculator executes Step 4. The weight limit is calculated from the unadjusted weight limit and the limit reduction identifier. According to the regulations in the standard [19], if the employee twists more than 45 degrees while lifting, reduce the Unadjusted Weight Limit obtained in Step 2 is multiplied by 0.85. So, the system asks the user if there is any twisting of more than 45 degrees while lifting. If the answer is yes, the weight limit is calculated as follows:

(Weight limit) = ((Unadjusted Weight) 0.85) * (Reduction Modifier)*

If the answer is no, the weight limit is:

*(Weight limit) = (Unadjusted Weight) * (Reduction Modifier)*

- **Determinator and Report**

In this step, the weight limit calculated in the previous step is compared to the weight actually lifted. If the actual weight is heavier than the weight limit, there is a MSD hazard and a report is generated.

4.4 Coding Strategy and Control Techniques

4.4.1 Pattern Matching

CLIPS provides different formats for knowledge representation. In our system, we use rules to represent knowledge. A rule has one or more conditions. If the conditions are satisfied, actions will be taken. Rules are executed (or fired) based on the existence or non-existence of facts or instances of user-defined classes. The inference engine of CLIPS attempts to match the conditions of the rules of the current system state and take actions.

After this rule base is built and the fact-list is prepared, CLIPS is ready to execute rules. In a conventional computer program, the sequence of operations is defined explicitly in the algorithm. In expert systems, the program flow does not need to be defined explicitly.

Rules are presented in the format of the "*defrule*" structure, provided by CLIPS. The syntax is like:

```
(defrule <rule-name> [<comment>]
   [<declaration>]              ; Rule properties
   <conditional-elements>       ; Left-Hand Side (LHS)
   =>
   <action>)                    ; Right-Hand Side (RHS)
```

The LHS contains a series of conditional elements (CEs). These CEs typically consist of pattern conditional elements to be matched against pattern entities. An implicit and conditional element always surrounds the patterns on the LHS. The RHS contains a list of actions to be performed. The arrow (=>) separates the LHS from the RHS. There is no limit to the number of conditional elements or actions a rule may have. When all conditional elements on the LHS are satisfied, the expert system performs actions sequentially.

If no CEs are on the LHS, the inference engine uses the pattern CE (initial-fact) automatically. If no actions are on the RHS, the rule can be activated and fired but nothing will happen.

When a program is running, the inference engine searches and fires rules one by one to accomplish all the functions the expert system has. An example of the rules is given below:

```
(defrule check-bending-degree-0
(declare (salience 15))
?bending-degree <- (degree ?s-degree)
=>
(retract ?bending-degree)
(printout t " What is your bending degree by this posture?" crlf crlf)
(bind ?a-degree(read))
(assert    (actual-d ?a-degree)
           (standard-d ?s-degree)))
```

In the above example, the rule is given a name *check-bending-degree-0*, then a priority of execution (*15*). Next we set under what condition the rule is executed. If the condition is set, the rule asks from further information and sets the condition elements. In this case, there are the actual number of bending degrees together with the number of degrees specified in the ergonomics standard which was provided in another rule not shown here. The expert system can now use this condition element pair as conditions for other rules.

The expert system will use the just acquired actual bending degree and early stored standard bending degree as conditions for other rules.

4.4.2 Control Techniques

CLIPS provides various techniques to control the rule execution process. These techniques can by used effectively to make the program more powerful and efficient. In this system, the main control techniques used are input techniques, salience and pattern logical OR, AND and NOT.

- **Input Techniques**

We use the ***read*** function to input information. This function allows computer to read information from the keyboard. Read function requires a carriage return before it will read the token entered. The communication between the user and the computer is established. The application of this function frequently appears in our system. In the following example which is similar to the one presented in the previous section, we can see a typical application of this function.

```
(defrule check-work-time-0
(declare (salience 10))
?duration <- (hour ?s-time)
=>
(retract ?duration)
(printout t " How many hours do you work by this posture per day?" crlf crlf)
(bind ?a-time(read))
(assert    (actual-t ?a-time)
           (standard-t ?s-time)))
```

At the LHS of this rule, we have *(defrule check-work-time-0 , (declare (salience 10))* and *?duration <- (hour ?s-time). defrule check-work-time-0* defines a rule. *Check-work-time-0* is the name of the rule. *(declare (salience 10))* means this rule has a priority to execute. The priority level is 10. *?duration <- (hour ?s-time)* is to store the fact *(hour?s-time)* into the address named *?duration.* In the fact *(hour?s-time), ?s-time* is a variable stands for standard working time. The value of this variable is set by a previous program according to the current working condition.

When the expert system sends the fact *(hour ?s-time)* to the address *?duration*, the program runs to the RHS, by the command *(retract ?duration)*, it retracts fact *(hour ?s-time)*. Then the system prints out the question *"How many hours do you work by this posture per day?"* on the screen to ask the user. At this time, command *(bind ?a-time(read))* works. The function of it is to give a value to the variable *?a-time*. The command *read* is used to let the system acquire that value from the user through the keyboard. After that, the system stores the data by *(assert (actual-t ?a-time))* to let the system know that actual work time by the fact *(actual-t ?a-time)*. Therefore, the communication between the system and the user is established. Meanwhile, by asserting the fact *(standard-t ?s-time)*, the program tells the system the standard working at this posture. Facts *(actual-t ?time)* and *(standard-t ?s-time)* will work as the condition elements of another rule to execute.

- **Salience**

CLIPS provides a direct way of control through the command *salience*. In the inference system, the agenda is a list of tasks which need to be executed. The most recent activation is placed at the top of the agenda and is the one first to fire. *Salience* allows more important rules to stay at the top of the agenda regardless of when the rules were added to the agenda. Lower salience values have a lower position on the agenda than higher salience values.

We use salience to force rules to fire in a sequential fashion. In the previous example, the *check-work-time-0* rule has salience 10, it has a higher priority to fire than rules with salience 0. It is very useful in loops. For example, whenever the system meets the fact *(hour ?s-time)*, it calls this rule again and again to check the working time.

- **The Pattern Logical OR, AND, NOT**

CLIPS provides the capability of specifying an explicit logical AND, OR and NOT condition on the LHS. Those logical conditions have an important role in our system. With the logical AND condition, a rule will not be triggered unless all of the patterns are true. With the logical OR condition, a rule is executed if only one of the patterns is true. Sometimes it is useful to be able to pattern match against the absence of a particular fact in the fact-list. CLIPS allows the specification of the absence of a fact as pattern on the LHS using the logical NOT. Logical conditions are very helpful to make a compact program, especially when rules become complicated.

4.4.3 System Operation & Running Examples

We are running CLIPS in a Windows environment. As shown in Figure 4.1, every time, the expert system starts from the file start.txt. By the command (*load*), we load the file **start-f.txt** as shown in Figure 4.17.

```
CLIPS> (load "C:/My Documents/Final program/Start-f.txt")
Defining deffunction: ask-question
Defining deffunction: yes-or-no-p
Defining deffacts: whole-system-start
Defining defrule: welcome +j
Defining defrule: main +j
Defining defrule: get-item +j
Defining defrule: leave-system +j
TRUE
```

Figure 4.17 Load Start-f.txt

When loading the file, CLIPS "tells" the user "where it is" and "what it is doing" by displaying "Defining ..." on the screen. The **+j** in the output indicates that a **join** is being added. In the last line of Figure 4.17, "TURE" means the program is loaded successfully. Then with the command *(reset)* (Figure 4.18), all activations are removed from the agenda, and all facts from the fact-list.

```
CLIPS> (reset)
```

Figure 4.18 Reset

At this moment, the program is ready to execute. The command ***(run)*** triggers the system.

```
CLIPS> (run)
```

Figure 4.19 Run

The user now gets the menu indicated in Figure 4.4, which is repeated below:

```
*******************************************************************************

      Welcome to Knowledge-Based Expert System

*******************************************************************************
Please select the number of the corresponding item to run the system:

------------------------------------------------------------
.
.      1. General OSHA rules
.
.      2. VDT Work Station Check
.
.      3. In-depth Analysis about Work Shop
.
.      4. Lift-limit calculation
.
.      5. Leave System
.
------------------------------------------------------------
```

Figure 4.4 Start Menu

In this example, we choose option 3, which loads the file **work-f.txt** and the following information appears on the screen as shown in Figure 4.20:

```
CLIPS> (load "C:/My Documents/Final program/Work-f.txt")
Defining deffunction: ask-question
Defining deffunction: yes-or-no-p
Defining deffacts: start-general-questions
Defining defrule: welcome +j
Defining defrule: unfortable-feeling +j
Defining defrule: continue +j
Defining defrule: count-first-no +j+j+j+j
Defining defrule: count-next-no +j+j+j
Defining defrule: count-number-of-no +j+j+j
Defining defrule: risk-factor-no +j+j+j
Defining defrule: start-general-questions +j
Defining defrule: number-selection +j
Defining defrule: check-work-time-0 +j
Defining defrule: check-work-time-1 +j+j
```

```
Defining defrule: check-bending-degree-0 +j
Defining defrule: check-bending-degree-1 +j+j
Defining defrule: check-force +j
Defining defrule: check-other-part +j
Defining defrule: check-again +j
Defining defrule: leave-system +j
Defining defrule: main-manu +j
Defining defrule: physical-risk-factor-report +j+j
Defining defrule: main-factor-report +j+j
Defining defrule: combined-factor-report +j+j
Defining defrule: no-combinded-factor =j+j
Defining defrule: main-factor-report-connect-to-degree +j+j
Defining defrule: bending-degree-report-big +j+j+j+j
Defining defrule: bending-degree-report-small =j+j+j+j
Defining defrule: no-bending-degree =j+j+j
Defining defrule: work-time-report-long +j+j+j+j
Defining defrule: work-time-report-short =j+j+j+j
Defining defrule: high-hand-force-flexion +j+j
Defining defrule: high-hand-force-extention =j+j
Defining defrule: high-hand-force-ulnar =j+j
Defining defrule: no-other-risk-factors +j+j
Defining defrule: WMSD-harzard-report +j+j+j
Defining defrule: WMSD-no-over-time =j=j+j
Defining defrule: WMSD-no-PRF =j+j+j
Defining defrule: WMSD-no-PRF =j+j+j
Defining defrule: WMSD-state-conclusions +j
Defining defrule: check-the-body-part +j
Defining defrule: awkward-posture-shoulders-1 +j+j
Defining defrule: awkward-posture-shoulders-2 =j+j
Defining defrule: awkward-posture-neck-1 +j+j
Defining defrule: awkward-posture-back-1 +j+j
Defining defrule: awkward-posture-back-2 =j+j
Defining defrule: awkward-knees-squatting +j+j
Defining defrule: awkward-knees-kneeling =j+j
Defining defrule: knees-repeated-impact =j+j
Defining defrule: high-hand-force-1-1 +j+j
Defining defrule: high-hand-force-1-2 =j+j
Defining defrule: high-hand-force-2-1 =j+j
Defining defrule: high-hand-force-2-2 =j+j
Defining defrule: high-repetitive-motion-shoulder =j+j
Defining defrule: high-repetitive-motion-hands =j+j
Defining defrule: high-force-exertion +j
Defining defrule: high-repetitive-motion-4 =j+j
Defining defrule: problem-upper-body-9 =j+j
TRUE
CLIPS>
```

Figure 4.20 Load Work-f.txt

Then, by command *(run)*, we have the following information as shown in Figure 4.21.

Welcome to Knowledge-Based Expert System

In-depth-analysis about MSD

Do you have any of the following uncomfortable feelings (yes/no)?

- **Painful joints**
- **Pain, tingling or numbness in hands or feet**
- **Shooting or stabbing pains in arms or legs**
- **Swelling or inflammation**
- **Burning sensation**
- **Pain in wrists, shoulders, forearms, knees**
- **Stiffness**

Do you have any of the above uncomfortable feelings (yes/no)?

Figure 4.21 Uncomfortable Feelings

By answering "y", we get the menu in Figure 4.22.

---

a. Shoulders;

b. Neck;

c. Back/Trunk/Hip;

d. Knees;

e. Hands/Wrists/Arms;

f. MAIN MENU of the system

Please select a,b,c,d,e or f and then press Enter to choose the part of your check.

Figure 4.22 Check Body Part Menu

If we now choose **a**, the conversation continues like shown below in Figure 4.23,

a
Let's check the Shoulders part.

Do you work with your hand(s) above the head or the elbow(s) above the shoulders
(yes/no)?
y

How many hours do you work by this posture ?

6
The working time by this posture is too long.

The actual working time is 6 hours

The working time should be shorter than 4 hours.

====================================

Alert: Here is a MSD hazard!

====================================

Figure 4.23 MSD Alert

Figure 4.24 shows how the expert system analyzes the user input of the actual working condition and gives feedback to the user explaining

1. What is wrong,
2. What can be done to remedy it, and
3. Warning that there is a real possibility the user develops a Musculoskelotal Disorder.

Do you repetitively raise your hand(s) above the head or the elbows above the shoulder(s)more often than once per minute (yes/no)?

y
The potential physical risk factor of WMSD is (awkward posture at shoulders)

How many hours do you work by this posture per day?

5
The working time by this posture is too long.

The actual working time is 5 hours

The working time should be shorter than 4 hours.

=====================================

Alert: Here is a MSD hazard!

=====================================

Figure 4.24 Another MSD Alert

The expert system continues to ask more questions to find out if there are any other ergonomic problems. This is shown in Figure 4.25.

Do you use the same motion with little or no variation every few seconds (excluding keying activities) (yes/no)?

n
If you have checked your body, do you want to know how much MSD hazards you have (yes/no)?

y
You have 2 MSD risk factors!

Do you want to check other part of your body (yes/no)?
n
Do you want to leave CLIPS (yes/no)?

n
Do you want to see other part of our system (yes/no)?
n
CLIPS>

Figure 4.25 More Questions

The actual CLIPS program which executes the example given above is listed in section 5 of Appendix B of this thesis. The other CLIPS programs (see Figure 4.1) developed for our expert system are listed in sections 1, 2, 3, 4 of Appendix B of this thesis.

4.5 Summary

In this chapter, we introduced the research work in detail. Through the introduction, we know how the rules were extracted. By analyzing, we understand the meaning of the rules, the fields they cover and their application way in expert systems. We explained the whole procedure of constructing the system and described every component of this system in detail. In the end, we illustrated how this expert system works. It is clear that the intention of solving the ergonomics problems by the technique of rule-based expert system is feasible and the work is successful.

Chapter 5

Conclusions and Recommendations for Future Work

5.1 General

People began to apply simple ergonomic principles in their life from the ancient time. But as a distinct discipline, compared to some other scientific subjects, ergonomics is relatively new. Work-related musculoskeletal disorders can cause severe problems that influence people's health. It is also one of the key topics that many ergonomics experts have studied. They have many methods to probe problems in ergonomics. In recent years, the computer has become an important assistant to help researchers to process data in their experiments. Artificial intelligence and expert systems were first developed in the 1960s. From then on, especially since the 1980s, this sophisticated technology has matured and now is playing an important role in many fields of society. The application of expert systems in ergonomics is new. In this research, we have established a prototype of a rule-based expert system that can be used to evaluate musculoskeletal disorders.

5.2 Highlights of This Research

In the previous chapters, we have discussed the application of a rule-based expert system in ergonomics workplace design. In Chapter 1, we reviewed the history of ergonomics and expert systems, introduced the development of the two subjects and presented the current problems in the application of ergonomics. Next, we discussed work-related musculoskeletal disorders (MSDs) and examined the advantages and weakness of expert systems. In Chapter 2, the literature review presented shows us the seriousness and size of ergonomic problems in the society. By studying current research papers in expert systems applications, we acquired basic knowledge and learned some fundamental techniques about how to develop a rule-based expert system. In Chapter 3, we analyzed the problem in detail. We realized the importance to solve the MSD problem in ergonomics workplace design and understood that rule-based expert system is a sophisticated technique to help us to solve the problem effectively and efficiently. We decided to use OSHA standard as our knowledge source and CLIPS as our expert system-developing tool. In Chapter 4, we analyzed the rules in detail and discussed how to code them into an expert system. Finally, we introduced every component of our system, presented functions of each of them and gave an example how our expert system can be used to analyze if a worker is at risk of developing MSD(s) and what can be done to avoid them.

5.3 Conclusions and Future Research

MSD is one of the serious ergonomic problems that many people are suffering from. Ergonomic researchers have contributed a lot of work in MSD research. To our understanding, preventing the occurrence of MSD is rather more important than seeking medical attention when MSD exists. However, to correctly and efficiently identify the MSD hazard requires ergonomic expertise which is usually grasped by ergonomic experts. When experts are not available, expert systems can help people to evaluate and analyze MSD hazards and give recommended actions if MSD hazards exist. Our system focuses on MSD hazard identification with limited recommendations for corrective actions. It provides a fast and convenient way for applying ergonomic rules.

From designing this system, we learned, first, the evolutionary prototyping in designing expert systems is proven to be superior to conventional system development life cycle. Prototyping presents a more efficient way to design a system. Second, it is evident from this research that other similar medical sub-domains might be good candidates for the application of the expert system technology. Third, CLIPS was found to be a flexible, powerful, and intuitive development environment for this application.

Basically, the future research work for this expert system should be focused on two areas. One is the ergonomic knowledge base, the other is expert system building technologies. In the Federal Register, OSHA not only provides checklists to identify MSD hazards, but also, it tells people what actions should be taken when an MSD hazard exists. Therefore, in the next step, more work should be done to let the system have richer and better recommendations if an MSD hazard exists.

With an enlarged knowledge base and complicated functions, the programming techniques should be improved by integrating the functions of other programming languages. This will allow us to build a more powerful, real-world expert system for ergonomic applications in work place design.

References

1 K.H.E. Kroemer, H.B. Kroemer and K.E. Kroemer-Elbert, *Ergonomics - How to Design for Ease and Efficiency* 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 2001

2 H. Shahnavaz, "Making Ergonomics a World-Wide Concept", *Ergonomics*, vol. 39, No. 12, 1996, 1391-1402

3 R. Shen, "Establishing Ergonomics in the People's Republic of China" *Ergonomics*, vol. 38, No.8, 1995, 1661-1666

4 "Rules and Regulations", *Federal Register*, Vol. 65, No. 220, 2000, 68262-68870, http://www.osha-slc.gov/FedReg_osha_pdf/FED20001114.pdf. (current April, 2002)

5 J.C. Giarratano, *Expert Systems Principles and Programming*, Boyd & Fraser, Boston, MA, 1989

6 V.H. Hilderbrandt, "Musculoskeletal Symptoms and Workload in 12 Branches of Dutch agriculture", *Ergonomics*, 1995, Vol. 38, No. 12, 2576-2587

7 N. Saldana, G. D. Herrin, T. J. Armstrong and A. Franzblau, "A Computerized Method for Assessment of Musculoskeletal Discomfort in the Workforce: A Tool for Surveillance", *Ergonomics*, 1994, Vol. 37, No.6, 1097-1112

8 D. Ranney, R. Wells and A. Moore, "Upper Limb Musculoskeletal Disorders in Highly Repetitive Industries: Precise Anatomical Physical Findings", *Ergonomics*, 1995, Vol. 38, No. 7, 1408-1423

9 D.B. Chaffin and G.B. Page, "Postural Effects on Biomechanical and Psycho-physical Weight-lifting Limits", *Ergonomics*, 1994, Vol. 37, No. 4, 663-676

10 U. Bergqvist, E. Wolgast, B. Nilsson, and M. Voss, "Musculoskeletal Disorders Among Visual Display Terminal Workers: Individual, Ergonomic, and Work Organizational Factors", *Ergonomics*, 1995, Vol. 38, No. 4, 763-776

11 A. Hedge and J.R. Powers, "Wrist Postures while Keyboarding: Effects of A Negative Slope Keyboard Support System and Full Motion Forearm Supports", *Ergonomics*, 1995, vol. 38, No. 3, 508-517

12 B. Das and A.K. Sengupta, "Computer-aid Human Modeling Programs for Workstation Design", *Ergonomics*, 1995, Vol. 38, No. 9, 1958-1972

13 J.S. Moore and A. Garg, "The Strain Index: A Proposed Method to Analyze Jobs for Risk of Distal Upper Extremity Disorders", 1995, *American Industrial Hygiene Association Journal*, 1995, Vol. 56, No. 5, 443-458

14 T.R. Waters, V. Putz-Anderson and A. Garg, "Applications Manual for the Revised NIOSH Lifting Equation", *National Institute for Occupational Safety and Health*, 1994, Publication No. 94-110

15 "UAW-GM Ergonomics Risk Factor Checklist RFC2", *United Auto Workers-General Motors Center for Human Resources, Health and Safety Center*, 1998

16 "2000 Threshold Limit Values for Chemical Substances and Physical Agents and Biological Exposure Indices", *American Conference of Governmental Industrial Hygienists*, 2000, 117-133

17 S. Hignett and L. McAtamney, "Rapid Entire Body Assessment (REBA)", *Applied Ergonomics*, 2000, Vol. 31, 201-205

18 L. McAtamney and E.N. Corlett, "RULA: A Survey Method for the Investigation of Work-Related Upper Limb Disorders", *Applied Ergonomics*, 1993, Vol. 24, 91-99

19 *Washington State's Ergonomics Rule*, http://www.lni.wa.gov/wisha/ergo/ergorule.htm (current April, 2002)

20 S.H. Snook and V.M. Ciriello, "The Design of Manual Handling Tasks: Revised Tables of Maximum Acceptable Weights and Forces", *Ergonomics*, 1991, Vol. 39, No. 9, 1197-1213

21 D.H. Lee and D.R. Decker, "Collaborative Engineering-Design Support System", *Third Conference on CLIPS Proceedings (Electronic Version)*, pp. 253-259, 1994, http://www.ghg.net/clips/download/documentation/3CCP.pdf, (current April, 2002)

22 M.A. Nour, F.O. Offodile and G.R. Madey, "A CLIPS-Based Expert System for The Evaluation and Selection of Robots", *Third Conference on CLIPS Proceedings (Electronic Version)*, 1994, pp. 23-33, http://www.ghg.net/clips/download/documentation/3CCP.pdf, (current April, 2002)

23 G.R. Madey, S.A. Alaraini, M.A. Nour and C.A. Bhansin, "On the Development of An Expert System for Wheelchair Selection", *Third Conference on CLIPS Proceedings (Electronic Version)*, 1994, pp. 5-17, http://www.ghg.net/clips/download/documentation/3CCP.pdf, (current April, 2002)

24 R. Lewandowski, D. Perkins, and D. Leasure, "Expert Witness - A System for Developing Expert Medical Testimony", *Third Conference on CLIPS Proceedings (Electronic Version)*, 1994, pp. 19-22, http://www.ghg.net/clips/download/documentation/3CCP.pdf, (current April, 2002)

25 J. Liebowitz, *The Handbook of Applied Expert Systems*, CRC Press, Boca

Raton, FL, 1998

26 E.H. Shortliffe, *Computer-based Medical Consultations, MYCIN*, Elsevier, New York, NY, 1976

27 C.J. Ortiz, D.A. Hasan "Expert System Technologies for Space Shuttle Decision Support: Two Case Studies", *Third Conference on CLIPS, Proceedings (Electronic Version)*, 1994, pp. 165-173, http://www.ghg.net/clips/download/documentation/3CCP.pdf (current April, 2002)

28 N. Jung and J. Biegel, "Knowledge Based Translation and Problem Solving in An Intelligent Individualized Instruction System", *Third Conference on CLIPS Proceedings (Electronic Version)*, 1994, pp. 217-223, http://www.ghg.net/clips/download/documentation/3CCP.pdf, (current April, 2002)

29 "Statement by the President", http://www.whitehouse.gov/news/releases/2001/03/20010321.html (current April, 2002)

30 *CLIPS Basic Programming Guide*, Version 6.10, 1998, http://www.ghg.net/clips/download/documentation/bpg.pdf, (current April, 2002)

Appendix A Checklists

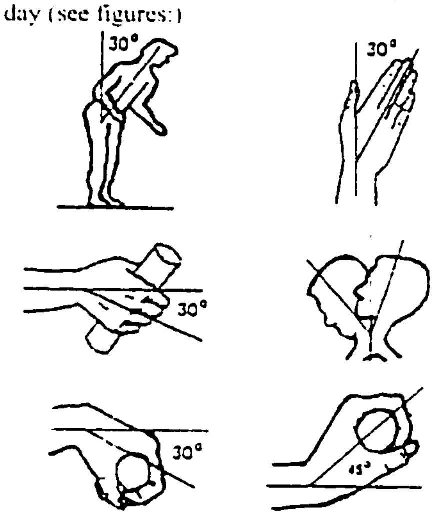
Table W-1 - Basic Screening Tool

You need only review risk factors for those areas of the body affected by the MSD incident.

Risk Factors This Standard Covers	Performing job or tasks that involve:	Body Part Associated With Musculo Skeletal Disorder (MSD) Incident: Neck/ Shoulder	Hand/ Wrist/ Arm	Back/ Trunk/ Hip	Leg/ Knee/ Ankle
Repetition	(1) Repeating the same motions every few seconds or repeating a cycle of motions involving the affected body part more than 2 consecutive hours in a workday;	√	√	√	√
	(2) Using an input device, such as a keyboard and/or mouse, in a steady manner for more than 4 hours total in a workday	√	√		
Force	(3) Lifting more than 75 pounds at any one time, more than 55 pounds more than 10 times per day; or more than 25 pounds below the knees, above the shoulders, or at arms' length more than 25 times per day,	√	√	√	√
	(4) Pushing pulling with more than 20 pounds of initial force (e.g. equivalent of pushing a 65 pound box across a tile floor or pushing a shopping cart with five 40 pound bags of dog food) for more than 2 hours total per day;	√	√	√	√
	(5) Pinching an unsupported object weighing 2 or more pounds per hand, or use of an equivalent pinching force (e.g. holding a small binder clip open), for more than 2 hours per day;		√		
	(6) Gripping an unsupported object weighing 10 pounds or more per hand, or use of an equivalent gripping force (e.g. crushing the sides of an aluminum soda can with one hand), for more than 2 hours per day.		√		

Table W-1 - Basic Screening Tool - continued

You need only review risk factors for those areas of the body affected by the MSD incident.

		Body Part Associated With Musculo Skeletal Disorder (MSD) Incident			
Risk Factors This Standard Covers	**Performing job or tasks that involve:**	**Neck/ Shoulder**	**Hand/ Wrist/ Arm**	**Back/ Trunk/ Hip**	**Leg/ Knee/ Ankle**
Awkward Postures	(7) Repeatedly raising or working with the hand(s) above the head or the elbow(s) above the shoulder for more than 2 hours total per day;	√	√	√	
	(8) Kneeling or squatting for more than 2 hours total per day;			√	√
	(9) Working with the back, neck or wrists bent or twisted for more than 2 hours total per day (see figures:) 	√	√	√	
Contact Stress	(10) Using the hand or knee as a hammer more than 10 times per hour for more than 2 hours total per day.		√		√
Vibration	(11) Using vibrating tools or equipment that typically have high vibration levels (such as chainsaws, jack hammers, percussive tools, riveting or chipping hammers) for more than 30 minutes total per day;	√	√	√	
	(12) Using tools or equipment that typically have moderate vibration levels (such as jig saws, grinders, or sanders) for more than 2 hours total per day.	√	√		

Appendix D-2 to §1910.900: VDT Workstation Checklist

Using this checklist is one, but not the only, way an employer can comply with the requirement to identify, analyze and control MSD hazards in VDT tasks. This checklist does not require that employees assume specific working postures in order for the employer to be in compliance. Rather, employers will be judged to be in compliance with paragraph (k) and (m) of OSHA's standard if they provide the employee with a VDT workstation is arranged or designed in a way that would pass this checklist.

If employee exposure does not meet the levels indicated by the Basic Screening Tool, you may STOP HERE.

WORKING CONDITIONS **The workstation is designed or arranged for doing VDT tasks so it allows the employee's . . .**	Y	N
A. **Head and neck** to be about upright (not bent down/back).		
B. **Head, neck and trunk** to face forward (not twisted).		
C. **Trunk** to be about perpendicular to floor (not leaning forward/backward)		
D. **Shoulders and upper arms** to be about perpendicular to floor (not stretched forward) and relaxed (not elevated).		
E. **Upper arms and elbows** to be close to body (not extended outward)		
F. **Forearms, wrists, and hands** to be straight and parallel to floor (not pointing up/down).		
G. **Wrists and hands** to be straight (not bent up/down or sideways toward little finger).		
H. **Thighs** to be about parallel to floor and **lower legs** to be about perpendicular to floor		
I. **Feet** to rest flat on floor or be supported by a stable footrest.		
J. VDT tasks to be organized in a way that allows employee to vary VDT tasks with other work activities, or to take micro-breaks or recovery pauses while at the VDT workstation.		
SEATING **The chair . . .**	**Y**	**N**
1. **Backrest** provides support for employee's lower back (lumbar area).		
2. **Seat width and depth** accommodate specific employee (seatpan not too big/small)		
3. **Seat front** does not press against the back of employee's knees and lower legs (seatpan not too long)		
4. **Seat** has cushioning and is rounded/ has "waterfall" front (no sharp edge).		
5. **Armrests** support both forearms while employee performs VDT tasks and do not interfere with movement.		

KEYBOARD/INPUT DEVICE The keyboard/input device is designed or arranged for doing VDT tasks so that . . .	Y	N
6. **Keyboard/input device platform(s)** is stable and large enough to hold keyboard and input device		
7. **Input device** (mouse or trackball) is located right next to keyboard so it can be operated without reaching.		
8. **Input device** is easy to activate and shape/size fits hand of specific employee (not too big/small).		
9. **Wrists and hands** do not rest on sharp or hard edge.		

MONITOR The monitor is designed or arranged for VDT tasks so that . . .	Y	N
10 .**Top line** of screen is at or below eye level so employee is able to read it without bending head or neck down back. (For employees with bifocals/trifocals, see next item.)		
11. **Employee with bifocals/trifocals** is able to read screen without bending head or neck backward.		
12. **Monitor distance** allows employee to read screen without leaning head, neck or trunk forward/backward.		
13. **Monitor position** is directly in front of employee so employee does not have to twist head or neck.		
14. **No glare** (e.g., from windows, lights) is present on the screen which might cause employee to assume an awkward posture to read screen		

WORK AREA The work area is designed or arranged for doing VDT tasks so that . . .	Y	N
15. **Thighs** have clearance space between chair and VDT table/keyboard platform (thighs not trapped).		
16. **Legs and feet** have clearance space under VDT table so employee is able to get close enough to keyboard/input device.		

ACCESSORIES	Y	N
17. **Document holder**, if provided, is stable and large enough to hold documents that are used.		
18. **Document holder**, if provided, is placed at about the same height and distance as monitor screen so there is little head movement when employee looks from document to screen.		
19. **Wrist rest**, if provided, is padded and free of sharp and square edges.		
20. **Wrist rest**, if provided, allows employee to keep forearms, wrists and hands straight and parallel to ground when using keyboard/input device.		
21. **Telephone** can be used with head upright (not bent) and shoulders relaxed (not elevated) if employee does VDT tasks at the same time.		

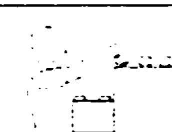
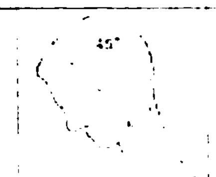
GENERAL	Y	N
22. Workstation and equipment have sufficient adjustability so that the employee is able to be in a safe working posture and to make occasional changes in posture while performing VDT tasks.		
23. VDT Workstation, equipment and accessories are maintained in serviceable condition and function properly.		
PASSING SCORE = "YES" answer on all "working postures" items (A-J) and no more than two "NO" answers on remainder of checklist (1-23).		

Job	Date / /
Notes	Analyst(s)

Reading across the page, determine if any of the conditions are present in the work activities. For many of the risk factors, two conditions are presented, which are the indicators for Caution (a lower level of risk) and Hazard (a higher level of risk). Most of the conditions are based on duration. If the lower threshold condition is not met, no box is checked. If the lower condition is met but the higher is not, then Caution is checked. If the higher condition is met (generally a longer period of time), then Hazard is checked.

If only Caution boxes are checked, the risk is present but immediate action (further analysis or interventions) are not recommended. It is worthwhile to continue to monitor Caution level jobs for changes that might increase the risk and for injuries or symptoms that may occur.

If one or more Hazard boxes are checked, a work-related musculoskeletal disorder (WMSD) hazard exists, and further action is recommended.

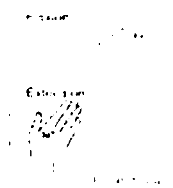
Awkward Posture				Check (✓) as applicable
Body Part	**Physical Risk Factor**	**Duration**	**Visual Aid**	
Shoulders	Working with the hand(s) above the head or the elbow(s) above the shoulder(s)	More than 2 hours total per day		Caution ❑
		More than 4 hours total per day		Hazard ❑
	Repetitively raising the hand(s) above the head or the elbow(s) above the shoulder(s) more than once per minute	More than 2 hours total per day		Caution ❑
		More than 4 hours total per day		Hazard ❑
Neck	Working with the neck bent more than 45° (without support or the ability to vary posture)	More than 2 hours total per day		Caution ❑
		More than 4 hours total per day		Hazard ❑

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

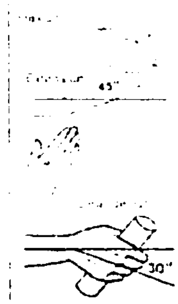
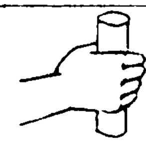
Awkward Posture (continued)				Check (✓) as applicable
Body Part	**Physical Risk Factor**	**Duration**	**Visual Aid**	
Back	Working with the back bent forward more than 30° (without support, or the ability to vary posture)	More than 2 hours total per day More than 4 hours total per day		Caution ❑ Hazard ❑
	Working with the back bent forward more than 45° (without support or the ability to vary posture)	More than 2 hours total per day		Hazard ❑
Knees	Squatting	More than 2 hours total per day More than 4 hours total per day		Caution ❑ Hazard ❑
	[illegible]	More than 2 hours total per day More than 4 hours total per day		Caution ❑ Hazard ❑

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form.

High Hand Force – Pinch

Body Part	Physical Risk Factor	Combined with	Duration	Visual Aid	Check (✓) as applicable
Arms, wrists, hands	Pinching an unsupported object(s) weighing 2 or more pounds per hand, or pinching with a force of 4 or more pounds per hand (comparable to pinching half a ream of paper)	Highly repetitive motion	More than 3 hours total per day		Hazard ❑
		Wrists bent in flexion 30° or more, or in extension 45° or more, or in ulnar deviation 30° or more	More than 3 hours total per day		Hazard ❑
		No other risk factors	More than 2 hours total per day		Caution ❑
			More than 4 hours total per day		Hazard ❑

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

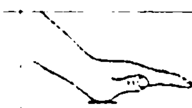
High Hand Force – Grasp					Check (✓) as applicable
Body Part	**Physical Risk Factor**	**Combined with**	**Duration**	**Visual Aid**	
Arms, wrists, hands	Gripping an unsupported object(s) weighing 10 or more pounds per hand, or gripping with a force of 10 pounds or more per hand (comparable to clamping light duty automotive jumper cables onto a battery)	Highly repetitive motion	More than 3 hours total per day		Hazard ☐
		Wrists bent in flexion 30° or more, or in extension 45° or more, or in ulnar deviation 30° or more	More than 3 hours total per day		Hazard ☐
		No other risk factors	More than 2 hours total per day		Caution ☐
			More than 4 hours total per day		Hazard ☐

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

Highly Repetitive Motion

Body Part	Physical Risk Factor	Combined with	Duration	Check (✓) as applicable
Neck, shoulders, elbows, wrists, hands	Using the same motion with little or no variation every few seconds (excluding keying activities)	No other risk factors	More than 2 hours total per day	Caution ❑
			More than 6 hours total per day	Hazard ❑
	Using the same motion with little or no variation every few seconds (excluding keying activities)	Wrists bent in flexion 30° or more, or in extension 45° or more, or in ulnar deviation 30° or more **AND** High, forceful exertions with the hand(s)	More than 2 hours total per day	Hazard ❑
	Intensive keying	Awkward posture, including wrists bent in flexion 30° or more, or in extension 45° or more, or in ulnar deviation 30° or more	More than 4 hours total per day	Hazard ❑
		No other risk factors	More than 4 hours total per day	Caution ❑
			More than 7 hours total per day	Hazard ❑

Repeated Impact

Body Part	Physical Risk Factor	Duration	Visual Aid	Check (✓) as applicable
Hands	Using the hand (heel/base of palm) as a hammer more than 10 times per hr	More than 2 hours total per day		Caution ❑
	Using the hand (heel/base of palm) as a hammer more than 60 times per hr			Hazard ❑
Knees	Using the knee as a hammer more than 10 times per hour	More than 2 hours total per day		Caution ❑
	Using the knee as a hammer more than 60 times per hour			Hazard ❑

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

Heavy, Frequent or Awkward Lifting

Body Part	Physical Risk Factor	Combined with	Duration	Check ✓ as applicable
Back and shoulders	Lifting 75 or more pounds	No other risk factors	One or more times per day	Caution ❑
	Lifting 55 or more pounds	No other risk factors	More than 10 times per day	Caution ❑
	Lifting more than 10 pounds	More than 2 times per minute	More than 2 hours total per day	Caution ❑
	Lifting more than 25 pounds	Above the shoulders Below the knees At arm's length	More than 25 times per day	Caution ❑
	WISHA Lifting Analysis – Perform if any Caution condition exists Actual Weight is greater than the Weight Limit (See separate work sheet)			Hazard ❑

Moderate to High Hand-Arm Vibration

Body Part	Physical Risk Factor	Duration	Check ✓, as applicable
Hands, wrists, and elbows	Using impact wrenches, carpet strippers, chain saws, percussive tools (jack hammers, scalers, riveting or chipping hammers) or other hand tools that typically have high vibration levels	More than 30 minutes total per day	Caution ❑
	Using grinders, sanders, jig saws or other hand tools that typically have moderate vibration levels	More than 2 hours total per day	Caution ❑
	WISHA HAV Analysis – Perform if any Caution condition exists Actual exposure time is greater than the Hazard Level Exposure Time (See separate work sheet)		Hazard ❑

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version includes some format changes, inclusion of caution zones and revisions to lifting and vibration sections. See http://www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

Job	Date / /
Notes	Analyst(s)

The lifting analysis on the following page is performed when one or more of the Caution Level job risk factors in the following checklist is present. This checklist is taken from the adapted WISHA checklist.

Heavy, Frequent or Awkward Lifting

Body Part	Physical Risk Factor	Combined with	Duration	Check (✓) as applicable
Back and shoulders	Lifting 75 or more pounds	No other risk factors	One or more times per day	Caution ❑
	Lifting 55 or more pounds	No other risk factors	More than 10 times per day	Caution ❑
	Lifting more than 10 pounds	More than 2 times per minute	More than 2 hours total per day	Caution ❑
	Lifting more than 25 pounds	Above the shoulders Below the knees At arm's length	More than 25 times per day	Caution ❑
	WISHA Lifting Analysis – Perform if any Caution condition exists Actual Weight (Step 1) is greater than the Weight Limit (Step 4) (See separate work sheet)			Hazard ❑

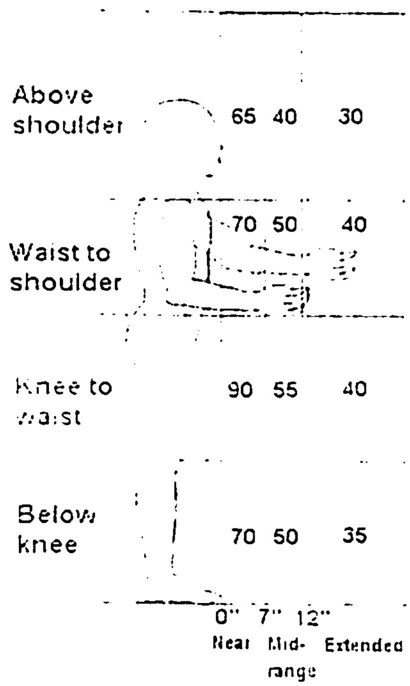
Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version focuses on the lifting section. See www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form

This analysis pertains to jobs where employees lift 10 lbs. or more.

Step 1 **Find out the actual weight of objects that the employee lifts.**

Actual Weight = ________ lbs.

Step 2 **Determine the Unadjusted Weight Limit.** Where are the employee's hands when they begin to lift or lower the object? Mark that spot on the diagram below. The number in that box is the Unadjusted Weight Limit in pounds



Unadjusted Weight Limit: ______ lbs.

Step 3 **Find the Limit Reduction Modifier.** Find out how many times the employee lifts per minute and the total number of hours per day spent lifting. Use this information to look up the Limit Reduction Modifier in the table below

How many lifts per minute?	For how many hours per day? 1 hr or less	1 hr to 2 hrs	2 hrs or more
1 lift every 2-5 mins	1.0	0.95	0.85
1 lift every min	0.95	0.9	0.75
2-3 lifts every min	0.9	0.85	0.65
4-5 lifts every min	0.85	0.7	0.45
6-7 lifts every min	0.75	0.5	0.25
8-9 lifts every min	0.6	0.35	0.15
10+ lifts every min	0.3	0.2	0.0

Note: For lifting done less than once every five minutes, use 1.0

Limit Reduction Modifier: ______

Step 4 **Calculate the Weight Limit.** Start by copying the Unadjusted Weight Limit from Step 2

Unadjusted Weight Limit: = ___ lbs.

If the employee twists more than 45 degrees while lifting, reduce the Unadjusted Weight Limit by multiplying by 0.85. Otherwise use the Unadjusted Weight Limit

Twisting Adjustment: = ___

Adjusted Weight Limit: = ___ lbs

Multiply the Adjusted Weight Limit by the Limit Reduction Modifier from Step 3 to get the Weight Limit **X**

Limit Reduction Modifier: ______

Weight Limit: = ___ lbs

Step 5 **Is this a hazard?** Compare the Weight Limit calculated in Step 4 with the Actual Weight lifted from Step 1. If the Actual Weight lifted is greater than the Weight Limit calculated, then the lifting is a WMSD hazard

Note: If the job involves lifts of objects with a number of different weights and/or from a number of different locations, use Steps 1 through 5 above to:

1. Analyze the two worst case lifts -- the heaviest object lifted and the lift done in the most awkward posture
2. Analyze the most commonly performed lift. In Step 3, use the frequency and duration for all of the lifting done in a typical

Adapted from State of Washington Department of Labor and Industries Ergonomics Rule
See http://www.lni.wa.gov/wisha/ergo/ergorule.htm
This version focuses on the lifting section. See www.hsc.usf.edu/~tbernard/ergotools for electronic copy of form.

Appendix B Programs

Start.txt

```
;==========================================================
;
; Expert System for Ergonomics - Beginning Part of the
;
;               Whole System
;
;     To Start the System, Just Reset and Run
;
;==========================================================

;==================
;;* DEFFUNCTIONS *
;==================

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question crlf crlf)
  (bind ?answer (read))
  (if (lexemep ?answer)
     then (bind ?answer (lowcase ?answer)))
  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question crlf crlf)
    (bind ?answer (read))
    (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer))))
  ?answer)

(deffunction yes-or-no-p (?question)        ; yes-or-no-p can have more arguments than
one
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
     then TRUE
     else FALSE))

   (deffacts whole-system-start
   (start))

(defrule welcome
?w <-(start)
=>
(retract ?w)
```

```
(format t "%n****************************************************** %n")
(format t "*                                                      * %n")
(format t "*       Welcome to Knowledge-Based Expert System       * %n")
(format t "*                                                      * %n")
(format t "*                                                      * %n")
(format t "****************************************************** %n")
(assert (main manu)))

(defrule main
?main-m <- (main manu)
=>
(retract ?main-m)
(printout t "Please select the number of the corresponding item to run the system: "rlf crlf
crlf)

(format t "%n------------------------------------------------------------   %n")
(format t ".                                                      .       %n")
(format t ".       1. General OSHA rules                          .       %n")
(format t ".                                                      .       %n")
(format t ".       2. VDT Work Station Check                              %n")
(format t ".                                                             .%n")
(format t ".       3. In-depth Analysis about Work Shop                   %n")
(format t ".                                                              %n")
(format t ".       4. Lift-limit calculation                              %n")
(format t ".                                                              %n")
(format t ".       5. Leave System                                        %n")
(format t ".                                                      .       %n")
(format t ".                                                      .       %n")
(format t "------------------------------------------------------------   %n")
(assert (get item)))

(defrule get-item
?item <- (get item)
=>
(retract ?item)
(bind ?f(read))
(if (eq 1 ?f)
  then (batch "C:/CLIPS/program/notepad/final/try-1.bat")
   else (if (eq 2 ?f)
      then (batch "C:/CLIPS/program/notepad/final/try-2.bat")
      else (if (eq 3 ?f)
          then (batch "C:/CLIPS/program/notepad/final/try-3.bat")
        else (if (eq 4 ?f)
           then (batch "C:/CLIPS/program/notepad/final/try-4.bat")
            else (if (eq 5 ?f)
               then (assert (leave))else
```

```
(assert (get item))))))))

(defrule leave-system
?L<- (leave)
=>
(retract ?L)
(if (yes-or-no-p "Do you also want to leave CLIPS(yes/no)? ")
then
(exit)
else))
```

Batch Files

Try-0.bat

```
(clear)
(load "C:/CLIPS/program/notepad/final/start-f.txt")
(reset)
(run)
```

Try-1.bat

```
(clear)
(load "C:/CLIPS/program/notepad/final/general-f.txt")
(reset)
(run)
```

Try-2.bat

```
(clear)
(load "C:/CLIPS/program/notepad/final/vdt-f.txt")
(reset)
(run)
```

Try-3.bat

```
(clear)
(load "C:/CLIPS/program/notepad/final/work-f.txt")
(reset)
(run)
```

Try-4.bat

```
(clear)
(load "C:/CLIPS/program/notepad/final/cal-f.txt")
(reset)
(run)
```

General-f.txt

```
;;;*******************************************************
;;;*              Ergonomic Expert System                *
;;;*                                                     *
;;;*      Part Three - In-depth Analysis for Work Shop   *
;;;*                                                     *
;;;*    This expert system diagnoses some simple problems *
;;;*         about the MSD in Work Shop                  *
;;;*                                                     *
;;;*                                                     *
;;;*              CLIPS Version 6.0                      *
;;;*                                                     *
;;;*      To execute, merely load, reset and run.        *
;;;*******************************************************


;;****************
;;* DEFFUNCTIONS *
;;****************

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question crlf crlf)
  (bind ?answer (read))
  (if (lexemep ?answer)
     then (bind ?answer (lowcase ?answer)))
  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question crlf crlf)
    (bind ?answer (read))
    (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer))))
  ?answer)

(deffunction yes-or-no-p (?question)       ; yes-or-no-p can have more arguments than
one
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
     then TRUE
     else FALSE))

;;;************
;;;*   START  *
;;;************
  (deffacts start-general-questions
  (in-depth-analysis))
```

```
(defrule welcome
?come <- (in-depth-analysis)
=>

(retract ?come)
(format t "%n ************************************************** %n")
(format t  " *                                                  * %n")
(format t  " *       Welcome to Knowledge-Based Expert System           * %n")
(format t  " *                                                          * %n")
(format t  " *                                                  * %n")
(format t  " *         In-depth-analysis about MSD                      * %n")
(format t  " *                                                          * %n")
(format t  " ************************************************** %n")
(assert (uncomfortable-feeling)))

(defrule unfortable-feeling
?poor-feeling <- (uncomfortable-feeling)
=>
(retract ?poor-feeling)
(format t "%n                                               %n")
(format t  "   Do you have any of the following uncomfortable feelings (yes/no)?
%n")
(format t  "  ==================================================
       %n")
(format t  "     * Painful joints
       %n")
(format t  "                                                             %n")
(format t  "     * Pain, tingling or numberness in hands or feet
       %n")
(format t  "
       %n")
(format t  "     * Shooting or stabbing pains in arms or legs            %n")
(format t  "
       %n")
(format t  "     * Swelling or inflammation                              %n")
(format t  "                                                             %n")
(format t  "     * Burning sensation
       %n")
(format t  "                                                             %n")
(format t  "     * Pain in wrists, shoulders, forearms, knees
       %n")
(format t  "                                                             %n")
(format t  "     * Stiffness
       %n")
(format t  "
==================================================  %n")
```

```
(format t "    Do you have any of the above uncomfortable feelings (yes/no)?
%n")
(format t "                                                          %n")
(if (yes-or-no-p "" )
then
(assert (list))
else (assert (continue))))

(defrule continue
?continue <- (continue)
=>
(retract ?continue)
(if (yes-or-no-p "Would you like to know something about how to avoid the above
problems (yes/no)? ")
then
(assert (list))
else
(assert (leave system))))     ; Leave the progam


;;;*****************************************************
;;;* COUNTING THE NUMBER OF WMSD HAZARD FACTORS     *
;;;*                                                   *
;;;*****************************************************

(defrule count-first-no
(declare (salience 10))
(not (c ?))
(not (number ?))
?a1 <- (count)
=>
(retract ?a1)
(assert (start count number of WMSD))
(assert (number 0))
(assert (c 1)))

(defrule count-next-no
(declare (salience 10))
(c ?c)
(number ?number)
?a2 <- (count)
=>
(retract ?a2)
(assert (start count number of WMSD)))

(defrule count-number-of-no
```

```
(declare (salience 10))
?f1 <- (start count number of WMSD)
?f2 <- (number ?number)
?f3 <- (c ?c)
=>
(retract ?f1 ?f2 ?f3)
    (assert (next-question))
    (assert (number =(+ 1 ?number)))
    (assert (c =(+ 1 ?c)))
    (printout t ?number crlf))

;;;*************************************
;;; PRINT THE NO OF WMSD RISK FACTORS
;;;*************************************

(defrule risk-factor-no
(declare (salience 9))
?WMSD<-(have WMSD-risk)
?end <- (print WMSD)
?risk-factor <- (number ?number)
=>
(retract ?end ?risk-factor ?WMSD)
(printout t "You have " ?number " WMSD risk factors!" crlf crlf)
(assert (again)))                         ; go to select continue or not

;;;****************************************
;;; START THE ASK THE GENERAL QUESTIONS
;;;****************************************

(defrule start-general-questions
?print-list <- (list)
=>
(retract ?print-list)
(format t "%n     ______________________________________________
%n")
(format t "                                    %n")
(format t "     a. Shoulders;                  %n")
(format t "                                    %n")
(format t "     b. Neck;                           %n")
(format t "                                        %n")
(format t "     c. Back/Trunk/Hip;                 %n")
(format t "                                        %n")
(format t "     d. Knees;                          %n")
(format t "                                        %n")
(format t "     e. Hands/Wrists/Arms;              %n")
(format t "                                        %n")
```

```
(format t "                                              %n")
(format t "      f. MAIN MANU  of the system       %n")
(format t "                                              %n")
(format t "  ______________________________ %n")
(assert (next-step)))


(defrule number-selection
?select <- (next-step)
=>
(retract ?select)
(printout t "Please select a,b,c,d,e or f and then press Enter to choose the part of your
check." crlf crlf)
(bind ?input(read))
(if (eq a ?input)
  then (assert (part Shoulders)
             (problem shoulder 1))
   else (if (eq b ?input)
       then (assert (part Neck)
                    (problem neck 1))
       else (if (eq c ?input)
           then (assert (part Back/Trunk/Hip)
                        (problem back 1))
         else (if (eq d ?input)
             then (assert (part Knees)
                          (problem knees 1))
              else (if (eq e ?input)
                  then (assert (part Hands/Wrists/Arms)
                               (problem hands 1))
                          else (if (eq f ?input)
                          then (assert (main manu))

else
(assert (next-step)))))))))  ; if the user doesn't assert 1 or 2 or 3 or 4 or 5,
                      ; it keeps ask the same question

;;==================
;; Check work time
;;==================

(defrule check-work-time-0
(declare (salience 10))                    ; check time after angle
?duration <- (hour ?s-time)
=>
(retract ?duration)
(printout t " How many hours do you work by this posture per day?" crlf crlf)
```

```
(bind ?a-time(read))
(assert    (actual-t ?a-time)
        (standard-t ?s-time)))

(defrule check-work-time-1
(declare (salience 10))
?a-t <- (actual-t ?a-time)
?s-t <- (standard-t ?s-time)
=>
(retract ?a-t ?s-t)
(if ( > ?a-time 24)
then
    (printout t "Are you on earth? One day has only 24 hours, idiot!" crlf crlf)
    (assert (hour ?s-time))
else
    (if (< ?a-time 0)
    then
        (printout t "Hi buddy, are you sleeping? Time cannot be minus!" crlf crlf)
        (assert (hour ?s-time))
    else
                (assert (a-t ?a-time)
                        (s-t ?s-time))
            (if (>= ?a-time ?s-time)
            then
                (assert (work-time long))
               else
                (assert (work-time short))))))

;;=====================
;; Check bending degree
;;=====================

(defrule check-bending-degree-0
(declare (salience 15))
?bending-degree <- (degree ?s-degree)
=>
(retract ?bending-degree)
(printout t " What is your bending degree by this posture?" crlf crlf)
(bind ?a-degree(read))
(assert    (actual-d ?a-degree)
        (standard-d ?s-degree)))                              ;tell system the standard
bending degree
                                                              ;and the actual bending
degree
```

```
(defrule check-bending-degree-1
(declare (salience 15))
?a-d <- (actual-d ?a-degree)
?s-d <- (standard-d ?s-degree)
=>
(retract ?a-d ?s-d)
(if (> ?a-degree 90)
then
    (printout t "Hi buddy, are you an acrobat?" crlf crlf)
    (assert (degree ?s-degree))
else
    (if (< ?a-degree 0)
    then
        (printout t " We use extention and flexion to judge the direction of the degree.
please reenter degree." crlf crlf)
        (assert (degree ?s-degree))
    else
          (assert (a-d ?a-degree)
                  (s-d ?s-degree))
        (if (>= ?a-degree ?s-degree)
        then
            (assert (combined-factor bending-degree too high)
                        (bending degree big))
        else
            (assert (combind-factor bending-degree small)
                                (bending degree small))))))

;;==========
;; Check force
;;==========
(defrule check-force
(declare (salience 15))
?Force <- (force ?s-force)
=>
(printout t " How much is the force (kg)?" crlf crlf)
(bind ?a-force(read))
(assert    (actual-f ?a-force)
        (standard-f ?s-force))                          ;tell system the standard force
                                                        ;and the actual force

(if (>= ?a-force ?s-force)
then
        (retract ?Force)
    (assert (main-factor high hand force))

else
```

```
        (retract ?Force)))

;****************************
;check other part of your body?
;****************************

(defrule check-other-part                 ; it doesn't need a higher salience
?check<-(check other part)
=>
(retract ?check)
(if (yes-or-no-p "If you have checked your body, do you want to know how much
WMSD hazards you have(yes/no)?")
then
(assert (print WMSD)
     (check again))
else
(assert (check again))))

(defrule check-again
?check-a<-(check again)
=>
(retract ?check-a)
(if (yes-or-no-p "Do you want to check other part of your body(yes/no)?")
then
(assert (list))
else
(assert (leave system))))

;;;*******************
;;;leave the system
;;;*******************

(defrule leave-system
?leave <- (leave system)
=>
(retract ?leave)
(if (yes-or-no-p "Do you want to leave CLIPS (yes/no)?")
then
(exit)
else
(if (yes-or-no-p "Do you want to see other part of our system
(yes/no)?" )
        then
        (assert (main manu))
        else
```

```
        (printout t "See you, sweetie! Your choice makes you leave the system, I'll miss
you!" crlf))))

;;================================
;; go to main manu of the system
;;================================

(defrule main-manu
?main <- (main manu)
=>
(retract ?main)
(batch "C:/CLIPS/program/notepad/final/try-0.bat"))

;****************************************
;*                  REPORT              *
;****************************************

;;=================================
;; Awkward posture  -  to report duration
;;=================================

(defrule physical-risk-factor-report
(declare (salience 10))
?R-0 <- (REPORT awk)
?main <- (physical-f $?physical)
=>
(retract ?R-0 ?main)
(printout t " The potential physical risk factor of WMSD is " ?physical  crlf crlf)
(assert (report duration)))

;;______________________________________________________

;;==============================
;;  Main problem factors Report
;;==============================

(defrule main-factor-report
(declare (salience 10))
?R-1 <- (REPORT main)
?main <- (main-factor $?main-factor)
=>
(retract ?R-1 ?main)
(printout t " The main risky factor of WMSD is " ?main-factor  crlf crlf)
(assert (report combined factor)))

;;============================================================
```

```
;; combined problem factors report - it's only for combined factor "with high repetitive
motion"
;;============================================================

(defrule combined-factor-report
(declare (salience 10))
?R-2 <- (report combined factor)
?c-factor <- (combined-factor $?combined)
=>
(retract ?R-2 ?c-factor )
(printout t "Meanwhile, you have combined WMSD risky factor " ?combined crlf crlf)
(assert (report duration)))

;;================================
;; When there is no combine factor
;;================================

(defrule no-combinded-factor
(declare (salience 10))
?R-2 <- (report combined factor)
(not (combined-factor $?combined))
=>
(retract ?R-2)
(assert (report duration)))

;;_______________________________________
;;This section is for degree report

;;==================
;; Main factor report
;;==================

(defrule main-factor-report-connect-to-degree
(declare (salience 10))
?R-1 <- (REPORT with degree)
?main <- (main-factor $?main-factor)
=>
(retract ?R-1 ?main)
(printout t " The main risky factor of WMSD is " ?main-factor  crlf crlf)
(assert (report combined degree)))

;;===============
;; Degree report
;;===============

(defrule bending-degree-report-big
```

```
(declare (salience 10))
?R-4-1<- (report combined degree)
?big <- (bending-degree big)
?standard-degree <- (s-d ?s-degree)
?actual-degree <- (a-d ?a-degree)
=>
(retract ?R-4-1 ?big ?standard-degree ?actual-degree )
(printout t "The bending degree by this posture is too big." crlf crlf)
(printout t "The acutal bending degree is " ?a-degree " degree" crlf crlf)
(printout t "The bending degree should be smaller than " ?s-degree " degree." crlf crlf)
(assert (report duration)))

(defrule bending-degree-report-small
(declare (salience 10))
?R-4-1<- (report combined degree)
?small <- (bending-degree small)
?standard-degree <- (s-d ?s-degree)
?actual-degree <- (a-d ?a-degree)
=>
(retract ?R-4-1 ?small ?standard-degree ?actual-degree )
(printout t "The bending degree by this posture is not too big." crlf crlf)
(printout t "The acutal bending degree is " ?a-degree " degree" crlf crlf)
(printout t "The bending degree should be smaller than " ?s-degree " degree." crlf crlf)
(assert (report duration)))

;;========================
;; When no degree dievation
;;========================

(defrule no-bending-degree
(declare (salience 10))
?R-4-1<- (report combined degree)
(not(bending-degree big))
(not(bending-degree small))
=>
(retract ?R-4-1)
(assert (report duration)))

;;______________________________________________________

;;================================
;; work time report -    from every part
;;================================


(defrule work-time-report-long
```

```
(declare (salience 10))
?R-3-1<- (report duration)
?long <- (work-time long)
?standard-time <- (s-t ?s-time)
?actual-time <- (a-t ?a-time)
=>
(retract ?R-3-1  ?long ?standard-time ?actual-time )
(printout t "The working time by this posture is too long." crlf crlf)
(printout t "The acutal working time is " ?a-time " hours" crlf crlf)
(printout t "The working time should be shorter than " ?s-time " hours." crlf crlf)
(assert (work time long)))

(defrule work-time-report-short
(declare (salience 10))
?R-3-1<- (report duration)
?short <- (work-time short)
?standard-time <- (s-t ?s-time)
?actual-time <- (a-t ?a-time)
=>
(retract ?R-3-1 ?short ?standard-time ?actual-time )
(printout t "The working time by this posture is okay." crlf crlf)
(printout t "The acutal working time is " ?a-time " hours" crlf crlf)
(printout t "The working time should be shorter than " ?s-time " hours." crlf crlf))

;;The upper part is for the report to the section of high hand force motion
;;_____________________________________________________________

;;============
;; flexion 30
;;============

(defrule high-hand-force-flexion
(declare (salience 20))
?t <- (h ?s-time)
?problem <- (flexion)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in flextion (yes/no)?")
then
        (printout t "Bend wrist in flextion." crlf crlf)
        (assert (degree 30)
                (hour ?s-time))         ; if has angle, then tell system the working time
else
        (assert (extention))))          ;  if no flexion, then check extention
```

```
;;============
;; extenion 45
;;============

(defrule high-hand-force-extention
(declare (salience 20))
?t <- (h ?s-time)
?problem <- (extention)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in extention (yes/no)?")
then
        (printout t " Bend wrist in extention. " crlf crlf)
        (assert (degree 45)
                (hour ?s-time))

else
        (assert (ulnar))))        ; if no extention, then check if there is ulnar


;;=================
;; ulnar deviation
;;=================

(defrule high-hand-force-ulnar
(declare (salience 20))
?t <- (h ?s-time)
?problem <- (ulnar)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in ulnar deviation (yes/no)?")

then
        (printout t "Bend wrist in ulnar deviation. " crlf crlf)
        (assert (degree 30)
                (hour ?s-time))

else
        (assert (no other risk factors))))

;;==============================================
;; When no other riks factors with gripping or pinching
;;==============================================

(defrule no-other-risk-factors
```

```
(declare (salience 20))
?no-risk-t <- (no-other-risk ?s-time)
?problem <- (no other risk factors)
=>
(retract ?problem ?no-risk-t)
(assert (hour ?s-time)
        (REPORT main)
        (report WMSD)))

;;====================
;; WMSD hazard report
;;====================

(defrule WMSD-harzard-report
?S-R <- ( report WMSD)                    ; tell the system to report
?PRF <- (physical risk factor)
?overtime <- (work time long)
=>
(retract ?S-R ?PRF ?overtime)
(assert ( WMSD alert)))

(defrule WMSD-no-over-time
?S-R <- ( report WMSD)
?PRF <- (physical risk factor)
(not(work time long))
=>
(retract ?S-R ?PRF))

(defrule WMSD-no-PRF
?S-R <- ( report WMSD)
?overtime <- (work time long)
(not(physical risk factor))
=>
(retract ?S-R ?overtime))

;;;************************
;;;* WMSD Alert            *
;;;************************

(defrule WMSD-state-conclusions ""
  (declare (salience 10))
  ?alert <- (WMSD alert)
  =>
  (retract ?alert)
  (assert (count))
  (printout t "==================================" crlf crlf)
```

```
  (printout t "  Alert:   Here is a  WMSD hazard !"   crlf crlf)
  (printout t "====================================" crlf)
  (assert (have WMSD-risk)))

;;*********************************************
;;                    QUERY RULES
;;*********************************************

;;;=====================
;;;Check the body part
;;;=====================

(defrule check-the-body-part
?body <- (part ?part)
=>
(assert (body ?part))
(retract ?body)
(printout t "Let's check the "  ?part " part." crlf crlf))

 ;;*****************************
;;  PROBLEM  ANALYSIS
 ;;*****************************

;;================================
;; About awkward postures - Shoulders
;;================================

(defrule awkward-posture-shoulders-1
(body Shoulders)
?problem <- (problem shoulder 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your hand(s) above the head or the elbow(s) above
the shoulders (yes/no)? ")
then
        (assert (physical-f awkward posture at shoulders)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)                              ; output a report
                (problem shoulder 2))

else
        (assert  (problem shoulder 2))))

(defrule awkward-posture-shoulders-2
```

```
(body Shoulders)
?problem <- (problem shoulder 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you repetitively raise your hand(s) above the head or the elbows
above the shoulder(s)
more often than once per minute (yes/no)? ")
then
        (assert (physical-f awkward posture at shoulders)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)                                   ; output a report
                (problem shoulder high repetitive motion))

else
        (assert (problem shoulder high repetitive motion))))

;;====================
;;Awkward Posture Neck
;;====================

(defrule awkward-posture-neck-1
(body Neck)
?problem<- (problem neck 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your neck bending without support or ability to vary
posture(yes/no)?")
then
        (assert (physical-f awkward posture at neck)
                (physical risk factor)
                (degree 45)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (check other part))
else
        (assert (check other part))))

;;=========================
;; Awkward Posture Back
;;=========================

(defrule awkward-posture-back-1
(body Back/Trunk/Hip)
```

```
?problem <- (problem back 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your back bent forward(without support, or the
ability to very posture) (yes/no)? ")
then
        (assert (physical-f awkward posture at back)
                (physical risk factor)
                (degree 30)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem back 2))

else
(assert (problem back 2))))

(defrule awkward-posture-back-2
(body Back/Trunk/Hip)
?problem <- (problem back 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your back bent forward near 45 degree (without
support or the ability to vary posture)(yes/no)? ")
        then
        (assert (physical-f awkward posture at back)
                (physical risk factor)
                (degree 45)
                (hour 2)
                (REPORT awk)
                (report WMSD)
                (check other part))
        else
        (assert (check other part))))          ; finish back part and check other part

;;;=====================
;;; Awkward Posture Knees
;;;=====================

(defrule awkward-knees-squatting
(body Knees)
?problem <- (problem knees 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you squat while working(yes/no)? ")
        then
```

```
        (assert (physical-f awkward posture at knees by squatting)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem knees 2))
        else
        (assert (problem knees 2))))

(defrule awkward-knees-kneeling
(body Knees)
?problem <- (problem knees 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have kneeling while working(yes/no)? ")
        then
        (assert (physical-f awkward posture at knees by kneeling)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem knees 3))
        else
        (assert (problem knees 3))))

;;===================
;; Repeated Impact Knees
;;===================

(defrule knees-repeated-impact
(body Knees)
?problem <- (problem knees 3)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the knee as a hammer more than once per minute (yes/no)?
")
        then
        (assert (physical-f repeated impact at knees by use knee as a hammer)
                (physical risk factor)
                (hour 2)
                (REPORT awk)
                (report WMSD)
                (check other part))
        else
        (assert (check other part))))
```

```
;;=========================
;; High Hand Force - Pinching
;;========================= =

(defrule high-hand-force-1-1
(body Hands/Wrists/Arms)
?problem <- (problem hands 1)
=>
(retract ?problem)
(if(yes-or-no-p "Do you need to pinch an unsupported object(s) while working
(yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)
                (main-factor pinching an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 0.7)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 1-2))
        else
        (assert (flexion)
                (force 0.7)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 1-2)))           ;check flexion -> extention -> ulnar-> no
                                                ; other risk factors

else
(assert (problem hands 1-2))))

(defrule high-hand-force-1-2
(body Hands/Wrists/Arms)
?problem <- (problem hands 1-2)
=>
(retract ?problem)
(if(yes-or-no-p "Do you just pinch an object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)
                (main-factor pinching an object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
```

```
        then
        (assert (combined-factor highly repetitive motion)
                (force 1.5)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-1))
        else
        (assert (flexion)
                (force 1.5)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-1))) ; check degrees starting from flexion
else
        (assert (problem hands 2-1))))          ;go to gripping

;;======================
;; High Hand Force - Gripping
;;======================

(defrule high-hand-force-2-1
(body Hands/Wrists/Arms)
?problem <- (problem hands 2-1)
=>
(retract ?problem)
(if(yes-or-no-p "Do you need to grip an unsupported object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)       ; when there is no other factors,
                                        ; the system know the working time is 4 hours

                (main-factor gripping an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-2))
        else
        (assert (flexion)
                (force 4)
                (h 3)
                (REPORT main)
```

```
            (report WMSD)
            (problem hands 2-2)))        ;check flexion -> extention -> ulnar

else
(assert (problem hands 2-2))))

(defrule high-hand-force-2-2
(body Hands/Wrists/Arms)
?problem <- (problem hands 2-2)
=>
(retract ?problem)
(if(yes-or-no-p "Do you just need to grip an object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4 hours)
                (main-factor gripping an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hand high repetitive motion))
        else
        (assert (flexion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hand high repetitive motion)))
else
        (assert (problem hand high repetitive motion))))

;;==============================
;; Highly Repetitve Motion - shoulders
;;==============================

(defrule high-repetitive-motion-shoulder
(body Shoulders)
?problem <- (problem shoulder high repetitive motion)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the same motion with little or no variation every few
seconds (excluding keying activities) (yes/no)? ")
then
```

```
        (assert (main-factor higly repetitive motion at shoulders)
                (physical risk factor)
                (hour 6)
                (REPORT main)
                (report WMSD)
                (check other part))
else
        (assert (check other part))))

;;===========================
;; Highly Repetitve Motion - hands
;;===========================

(defrule high-repetitive-motion-hands
(body Hands/Wrists/Arms)
?problem <- (problem hand high repetitive motion)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the same motion with little or no variation every few
seconds (excluding keying activities) (yes/no)? ")
then

        (assert (main-factor higly repetitive motion at shoulders)
                (no-other-risk 6)
                (physical risk factor)
                (h 2)
                (flexion)
                (REPORT main)
                (report WMSD)
                (high force exertions))
else
(assert (intensive keying))))   ; go to intensive keying

(defrule high-force-exertion
?problem <- (high force exertions)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have high, forcefl exertions with hands(yes/no)? ")
then
        (assert (combined-factor high, forcefl exertions with hands)
                (h 2)
                (report combined factor)
                (intensive keying))
else
        (assert (intensive keying)))) ; to check intensive keying
```

```
(defrule high-repetitive-motion-4
(or     (body Neck)
        (body Hands/Wrists/Arms))
?problem <- (intensive keying)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have intensive keying(yes/no)? ")
then
        (assert (main-factor intensive keying)
                (physical risk factor)
                (no-other-risk 7)
                (flexion)
                (h 4)
                (REPORT main)
                (report WMSD)
                (repeated impact hands))
else
        (assert (check other part)
                (repeated impact hands))))

(defrule problem-upper-body-9
(body Hands/Wrists/Arms)
?problem <- (repeated impact hands)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the hand as a hammer more than once per minute (yes/no)?
")
        then
        (assert (main-factor repeated impact)
                (physical risk factor)
                (hour 2)
                (REPORT main)
                (repot WMSD)
                (check other part))
        else
        (assert (check other part))))          ; finish the hands part and go to other part
```

VDT-f.txt

```
;;;*************************************************************
;;;*  Ergonomic Expert System      Part Two - Questions for VDT           *
;;;*                                                                      *
;;;*    This expert system diagnoses some simple                         *
;;;*    problems with a general questions                                *
;;;*                                                                      *
;;;*    CLIPS Version 6.0                                                 *
;;;*                                                                      *
;;;*    To execute, merely load, reset and run.                          *
;;;*************************************************************

;;;***************
;;;*  START   *
;;;***************

  (deffacts start
  (start))

(defrule start
(start)
=>
(format t "%n*************************************************%n")
(format t  "*                                                    * %n")
(format t  "*       Welcome to Knowledge-Based Expert System         * %n")
(format t  "*               VDT Workstation Check                    * %n")
(format t  "*  If employee exposure does not meet the levels indicated     * %n")
(format t  "*       by the Basic Screening tool, you may STOP HERE       * %n")
(format t  "*                                                    * %n")
(format t  "************************************************** %n")
(assert (question A)))

;;****************
;;* DEFFUNCTIONS *
;;****************

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question crlf crlf)
  (bind ?answer (read))
  (if (lexemep ?answer)
     then (bind ?answer (lowcase ?answer)))
  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question crlf crlf)
```

```
    (bind ?answer (read))
    (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer))))
  ?answer)

(deffunction yes-or-no-p (?question)
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
     then TRUE
     else FALSE))


;;;*************************
;;;* WMSD STATE RULES FIRST *
;;;*************************

(defrule normal-state-conclusions ""
  (declare (salience 10))
  ?b <- (body-state normal)
  =>
  (retract ?b)
  (assert (no WMSD-risk )))

(defrule WMSD-state-conclusions ""
  (declare (salience 10))
  ?b <- (body-state unsatisfactory)
  =>
  (retract ?b)
  (assert (have WMSD-risk ))
  (printout t " Alarm: You have WMSD risk !" crlf crlf))


;;;*************************
;;;* COUNTING THE NUMBER OF *
;;;*  "NO" ANSWERS        *
;;;*************************

(defrule count-first-no
(declare (salience 10))
(not (c ?))
(not (number ?))
?a1 <- (count)
=>
(retract ?a1)
(assert (start count number of no))
(assert (number 0))
```

```
(assert (c 1)))

(defrule count-next-no
(declare (salience 10))
(c ?c)
(number ?number)
?a2 <- (count)
=>
(retract ?a2)
(assert (start count number of no)))


(defrule count-number-of-no
(declare (salience 10))
?f1 <- (start count number of no)
?f2 <- (number ?number)
?f3 <- (c ?c)
=>
(retract ?f1 ?f2 ?f3)
    (assert (next-question))
    (assert (number =(+ 1 ?number)))
    (assert (c =(+ 1 ?c)))
    (printout t ?number crlf))

;;;*************************
;;;  PRINT NO FOR 1ST PART
;;;*************************
(defrule printout-out
(declare (salience 9))
?first-end <- (first-end)
?print <- (number ?number)
=>
(retract ?first-end ?print)
(if (> ?number 0)
then
(printout t "The total no you have: " ?number crlf)
(assert (question 0))
else))

;;;***************************
;;; PRINT IF ALL YES 1st PART
;;;***************************

(defrule all-yes
(declare (salience 9))
(not (have WMSD-risk))
```

```
?first-end <- (first-end)
=>
(retract ?first-end)
(printout t "All yes for this part! " crlf)
(assert (question 0)))


;;;***********************************
;;;*   COUNTING THE NUMBER OF        *
;;;*  "NO" ANSWERS IN THE 2nd PART    *
;;;***********************************

(defrule calculate-first-no
(declare (salience 6))
(not (m ?m))
(not (n ?n))
?c-1 <- (calculate)
=>
(retract ?c-1)
(assert (calculate number of no))
(assert (n 0))
(assert (m 1)))

(defrule calculate-next-no
(declare (salience 6))
(m ?m)
(n ?n)
?c-2 <- (calculate)
=>
(retract ?c-2)
(assert (calculate number of no)))

(defrule calculate-number-of-no
(declare (salience 6))
?w1 <- (calculate number of no)
?w2 <- (n ?n)
?w3 <- (m ?m)
=>
(retract ?w1 ?w2 ?w3)
    (assert (next-question))
    (assert (n =(+ 1 ?n)))
    (assert (m =(+ 1 ?m)))
    (printout t ?n crlf)
    (if (>= ?n 1)
    then
    (assert (WMSD-risk-2nd))
```

```
    (printout t "Alarm: You have WMSD hazard!" crlf crlf)
    else))


;;;*****************************
;;; PRINT NO FOR THE 2nd PART
;;;*****************************

(defrule printout-out-2nd
(declare (salience 5))
?second-end <- (second-end)
?print <- (n ?n)
=>
(retract ?second-end ?print)
(if (> ?n 0)
then
(printout t "The total no you have: " ?n crlf)
(assert (question-next))
else))

;;;**********************************
;;; PRINT IF ALL YES FOR THE 2ND PART
;;;**********************************

(defrule all-yes-second
(declare (salience 5))
(not (WMSD-risk-2nd))
?second-end <- (second-end)
=>
(retract ?second-end)
(printout t "All yes for this part! " crlf crlf)
(assert (question-next)))

;;;***************
;;;* QUERY RULES *
;;;***************
;;;===================
;;; The first question
;;;===================

(defrule question-A ""
  ?question <- (question A)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your head and neck about upright (yes/no)? ")
```

```
  then (assert (question B))

  else (assert (count))
      (assert (question B))
      (assert (body-state unsatisfactory))))

;;;========================
;;; The following quesitons
;;;========================

(defrule question-B ""
  ?question <- (question B)
  =>
  (retract ?question)
  (if (yes-or-no-p "Do your head, neck and trunk face forward (yes/no)? ")

  then (assert (question C))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question C))))

(defrule question-C ""
  ?question <- (question C)
  =>
  (retract ?question)
  (if (yes-or-no-p "Is your trunk about perpendiculaar to floor (yes/no)? ")

  then (assert (question D))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question D))))

(defrule question-D ""
  ?question <- (question D)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your shoulders and upper arms about perpendiculaar to floor
(yes/no)? ")

  then (assert (question E))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question E))))
```

```
(defrule question-E ""
  ?question <- (question E)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your upper arms and elbows close to body (yes/no)? ")

  then (assert (question F))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question F))))

(defrule question-F ""
  ?question <- (question F)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your forearms, wrists and hands straight and parallel to floor
(yes/no)? ")

  then (assert (question G))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question G))))

(defrule question-G ""
  ?question <- (question G)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your upper arms and elbows close to body (yes/no)? ")

  then (assert (question H))
  else
      (assert (body-state unsatisfactory))
      (assert (count))
      (assert (question H))))

(defrule question-H ""
  ?question <- (question H)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your thighs about parallel to floor and
lower legs about perpendicular to floor (yes/no)? ")

  then (assert (question I))
```

```
  else
     (assert (body-state unsatisfactory))
     (assert (count))
     (assert (question I))))

(defrule question-I ""
  ?question <- (question I)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are your feet to rest flat on floor or supported by a stable footrest
(yes/no)? ")

  then (assert (question J))
  else
     (assert (body-state unsatisfactory))
     (assert (count))
     (assert (question J))))

(defrule question-J ""
  ?question <- (question J)
  =>
  (retract ?question)
  (if (yes-or-no-p "Are VDT tasks organized in a way taht allows employee to vary VDT
tasks with
other work activities, or to take micro-breaks ro recovery pauses while at
the VDT workstation.(yes/no)? ")

  then (assert (first-end))
  else
     (assert (body-state unsatisfactory))
     (assert (count)
          (have WMSD-risk))
     (assert (first-end))))


;;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;;
;;              CONTINUE TO THE SECOND PART
;;
;;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


(defrule question-0 ""
  ?question <- (question 0)
  =>
  (retract ?question)
```

```
 (printout t "Let's come to check some further conditions." crlf crlf crlf)
 (assert (question 1)))

;;;*********************
;;;*      SEATING
;;;*     The chair...
;;;*********************

(defrule question-1 ""
  ?question <- (question 1)
  =>
  (retract ?question)
  (if (yes-or-no-p "Backrest provides support for employee's lower back (lumbar area)
(yes/no)? ")

  then (assert (question 2))
  else
     (assert (calculate))
     (assert (question 2))))


(defrule question-2 ""
  ?question <- (question 2)
  =>
  (retract ?question)
  (if (yes-or-no-p "Seat width and epth accommodate specific employee
(seatpan not too big/small) (yes/no)? ")

  then (assert (question 3))
  else (assert (calculate))
     (assert (question 3))))

(defrule question-3 ""
  ?question <- (question 3)
  =>
  (retract ?question)
  (if (yes-or-no-p "Seat front does not press against the back of employee's
knees and lower legs (yes/no)? ")

  then (assert (question 4))
  else (assert (calculate))
     (assert (question 4))))

(defrule question-4 ""
  ?question <- (question 4)
  =>
```

```
  (retract ?question)
  (if (yes-or-no-p "Seat has cushioning and is rounded/has (waterfall-front/no sharp edge)
(yes/no)? ")

  then (assert (question 5))
  else (assert (calculate))
     (assert (question 5))))

(defrule question-5 ""
  ?question <- (question 5)
  =>
  (retract ?question)
  (if (yes-or-no-p "Armrests support both forearms while employee performs VDT tasks
and
do not interfere with movement(yes/no)? ")

  then (assert (question 6))
  else (assert (calculate))
     (assert (question 6))))

;;;**********************
;;; KEYBOARD/INPUT DEVIE
;;;**********************

(defrule question-6 ""
  ?question <- (question 6)
  =>
  (retract ?question)
  (if (yes-or-no-p "Keyboard/input device platform(s) is stable and large
enough to hold keyboard and input device(yes/no)? ")

  then (assert (question 7))
  else (assert (calculate))
     (assert (question 7))))

(defrule question-7 ""
  ?question <- (question 7)
  =>
  (retract ?question)
  (if (yes-or-no-p "Input device (mouse or trackball) is stable and large enough
to hold keyboard and input device. (yes/no)? ")

  then (assert (question 8))
  else (assert (calculate))
     (assert (question 8))))
```

```
(defrule question-8 ""
  ?question <- (question 8)
  =>
  (retract ?question)
  (if (yes-or-no-p "Input device is easy to activate and shape/size fits hand of
specific employee (not too big/small) (yes/no)? ")

  then (assert (question 9))
  else (assert (calculate))
     (assert (question 9))))

(defrule question-9 ""
  ?question <- (question 9)
  =>
  (retract ?question)
  (if (yes-or-no-p "Wrists and hands do not rest on sharp or hard edge (yes/no)? ")

  then (assert (question 10))
  else (assert (calculate))
     (assert (question 10))))

;;;************************
;;;     MONITOR
;;;************************

(defrule question-10 ""
  ?question <- (question 10)
  =>
  (retract ?question)
  (if (yes-or-no-p "Top line of screen is at or below eye level so employee
is able to read it without bending head or neck down/back.(yes/no)? ")

  then (assert (question 11))
  else (assert (calculate))
     (assert (question 11))))

(defrule question-11 ""
  ?question <- (question 11)
  =>
  (retract ?question)
  (if (yes-or-no-p "Employee with bbifocals/trifocals is able to read screen without
bending head or neck backward(yes/no)? ")

  then (assert (question 12))
  else (assert (calculate))
     (assert (question 12))))
```

```
(defrule question-12 ""
  ?question <- (question 12)
  =>
  (retract ?question)
  (if (yes-or-no-p "Monitor distance allows employee to read screen without leaning
head,
neck or trunk forward/backward(yes/no)? ")

  then (assert (question 13))
  else (assert (calculate))
     (assert (question 13))))

(defrule question-13 ""
  ?question <- (question 13)
  =>
  (retract ?question)
  (if (yes-or-no-p "Monitor position is drectly in fornt of employee so employee
does not have to twist head or neck.(yes/no)? ")

  then (assert (question 14))
  else (assert (calculate))
     (assert (question 14))))

(defrule question-14 ""
  ?question <- (question 14)
  =>
  (retract ?question)
  (if (yes-or-no-p "No glare (e.g., from windows, lights) is present on the screen
which might cause employee to assume an awkward posture to read screen.(yes/no)? ")

  then (assert (question 15))
  else (assert (calculate))
     (assert (question 15))))

;;;********************
;;;   WORK AREA
;;;********************

(defrule question-15 ""
  ?question <- (question 15)
  =>
  (retract ?question)
  (if (yes-or-no-p "Thighs have clearance space between chair and VDT
```

```
table/keyboard platform (thihgs not rapped).(yes/no)? ")

  then (assert (question 16))
  else (assert (calculate))
     (assert (question 16))))

(defrule question-16 ""
  ?question <- (question 16)
  =>
  (retract ?question)
  (if (yes-or-no-p "Legs and feet have clearance space under VDT table
so employee is able to get close enough to keyboard/input device.(yes/no)? ")

  then (assert (question 17))
  else (assert (calculate))
     (assert (question 17))))

;;;********************
;;;   ACCESSORIES
;;;********************

(defrule question-17 ""
  ?question <- (question 17)
  =>
  (retract ?question)
  (if (yes-or-no-p "Document holder, if provided, is stable and large enough to hold
documents that are used.(yes/no)? ")

  then (assert (question 18))
  else (assert (calculate))
     (assert (question 18))))

(defrule question-18 ""
  ?question <- (question 18)
  =>
  (retract ?question)
  (if (yes-or-no-p "Document holder, if provided, is placed at about the same height
and distance as monitor screen so there is little head movement when employee looks
from document to scree.(yes/no)? ")

  then (assert (question 19))
  else (assert (calculate))
     (assert (question 19))))

 (defrule question-19 ""
  ?question <- (question 19)
```

```
  =>
  (retract ?question)
  (if (yes-or-no-p "Wrist rest, if provided, is padded and free of sharp and square
edges.(yes/no)? ")

  then (assert (question 20))
  else (assert (calculate))
     (assert (question 20))))

(defrule question-20 ""
  ?question <- (question 20)
  =>
  (retract ?question)
  (if (yes-or-no-p "Wrist rest, if provided, allows employee to keep forearms, wrists and
hands
straight aand parallel to ground when using keyboard/input device.(yes/no)? ")

  then (assert (question 21))
  else (assert (calculate))
     (assert (question 21))))

(defrule question-21 ""
  ?question <- (question 21)
  =>
  (retract ?question)
  (if (yes-or-no-p "Wrist rest, if provided, allows employee to keep forearms, wrists and
hands
straight aand parallel to ground when using keyboard/input device.(yes/no)? ")

  then (assert (question 22))
  else (assert (calculate))
     (assert (question 22))))

;;;********************
;;;   GENERAL
;;;********************

(defrule question-22 ""
  ?question <- (question 22)
  =>
  (retract ?question)
  (if (yes-or-no-p "Workstation and equipment ahve sufficient adjustability so taht
the employee is able to be in a save working posture and to make occasional
changes in posture while performing VDT tasks..(yes/no)? ")

  then (assert (question 23))
```

```
  else (assert (calculate))
      (assert (question 23))))

(defrule question-23 ""
  ?question <- (question 23)
  =>
  (retract ?question)
  (if (yes-or-no-p "VDT Workstation, equipment and accessories are maintained in
serviceable condition and function properly.(yes/no)? ")

  then (assert (second-end))
  else (assert (calculate))
      (assert (second-end))))

;;;*********************************
;;; CONNECTION TO THE NEXT SECTION
;;;*********************************

(defrule next-step
?end <- (question-next)
=>
(retract ?end)

(if (yes-or-no-p "Would you like to try again (yes/no)?")
then
(reset)
(run)
else
(assert (check other part))))

;=======================
; Check other part
;=======================

(defrule to-main
?main<- (check other part)
=>
(retract ?main)
(if (yes-or-no-p " Do you want to run other part of the system (yes/no)?" )
then (batch "C:/CLIPS/program/notepad/final/try-0.bat")
else
(assert (leave system))))

;;==================
;; leave system
;;==================
```

```
(defrule leave-system
?leave <- (leave syetem)
=>
(retract ?leave)
(if (yes-or-no-p " Leave CLIPS (yes/no)?" )
then
(exit)
else))
```

Work-f.txt

```
;;;*************************************************************
;;;*                    Ergonomic Expert System                  *
;;;*                                                             *
;;;*        Part Three - In-depth Analysis for Work Shop         *
;;;*                                                             *
;;;*     This expert system diagnoses some simple problems       *
;;;*                about the MSD in Work Shop                   *
;;;*                    CLIPS Version 6.0                        *
;;;*                                                             *
;;;*          To execute, merely load, reset and run.            *
;;;*************************************************************

;;*****************
;;* DEFFUNCTIONS *
;;*****************

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question crlf crlf)
  (bind ?answer (read))
  (if (lexemep ?answer)
     then (bind ?answer (lowcase ?answer)))
  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question crlf crlf)
    (bind ?answer (read))
    (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer))))
  ?answer)
(deffunction yes-or-no-p (?question)        ; yes-or-no-p can have more arguments than
one
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
     then TRUE
     else FALSE))

;;;***************
;;;*   START     *
;;;***************
   (deffacts start-general-questions
   (in-depth-analysis))

(defrule welcome
?come <- (in-depth-analysis)
=>
```

```
(retract ?come)
(format t "%n ********************************************%n")
(format t  " *                                            * %n")
(format t  " *      Welcome to Knowledge-Based Expert System      * %n")
(format t  " *                                            * %n")
(format t  " *                                            * %n")
(format t  " *        In-depth-analysis about MSD          * %n")
(format t  " *                                            * %n")
(format t  " ******************************************** %n")
(assert (uncomfortable-feeling)))


(defrule unfortable-feeling
?poor-feeling <- (uncomfortable-feeling)
=>
(retract ?poor-feeling)
(format t "%n                                                    %n")
(format t  "   Do you have any of the following uncomfortable feelings (yes/no)?
%n")
(format t  "  ============================================  %n")
(format t  "     * Painful joints                                 %n")
(format t  "                                                      %n")
(format t  "     * Pain, tingling or numberness in hands or feet  %n")
(format t  "                                                      %n")
(format t  "     * Shooting or stabbing pains in arms or legs     %n")
(format t  "                                                      %n")
(format t  "     * Swelling or inflammation                       %n")
(format t  "                                                      %n")
(format t  "     * Burning sensation                              %n")
(format t  "                                                      %n")
(format t  "     * Pain in wrists, shoulders, forearms, knees     %n")
(format t  "                                                      %n")
(format t  "     * Stiffness                                      %n")
(format t  "  ============================================  %n")
(format t  "   Do you have any of the above uncomfortable feelings (yes/no)?
%n")
(format t  "                                              %n")
(if (yes-or-no-p "" )
then
(assert (list))
else (assert (continue))))

(defrule continue
?continue <- (continue)
=>
(retract ?continue)
```

```
(if (yes-or-no-p "Would you like to know something about how to avoid the above
problems (yes/no)? ")
then
(assert (list))
else
(assert (leave system))))      ; Leave the program

;;;***************************************************
;;;* COUNTING THE NUMBER OF WMSD HAZARD FACTORS  *
;;;*                                                *
;;;***************************************************


(defrule count-first-no
(declare (salience 10))
(not (c ?))
(not (number ?))
?a1 <- (count)
=>
(retract ?a1)
(assert (start count number of WMSD))
(assert (number 0))
(assert (c 1)))

(defrule count-next-no
(declare (salience 10))
(c ?c)
(number ?number)
?a2 <- (count)
=>
(retract ?a2)
(assert (start count number of WMSD)))

(defrule count-number-of-no
(declare (salience 10))
?f1 <- (start count number of WMSD)
?f2 <- (number ?number)
?f3 <- (c ?c)
=>
(retract ?f1 ?f2 ?f3)
    (assert (next-question))
    (assert (number =(+ 1 ?number)))
    (assert (c =(+ 1 ?c)))
    (printout t ?number crlf))
```

```
;;;************************************
;;; PRINT THE NO OF WMSD RISK FACTORS
;;;************************************

(defrule risk-factor-no
(declare (salience 9))
?WMSD<-(have WMSD-risk)
?end <- (print WMSD)
?risk-factor <- (number ?number)
=>
(retract ?end ?risk-factor ?WMSD)
(printout t "You have " ?number " WMSD risk factors!" crlf crlf)
(assert (again)))                        ; go to select continue or not

;;;************************************
;;; START THE ASK THE GENERAL QUESTIONS
;;;************************************


(defrule start-general-questions
?print-list <- (list)
=>
(retract ?print-list)
(format t "%n    ___________________________________ %n")
(format t "                                        %n")
(format t "      a. Shoulders;                     %n")
(format t "                                        %n")
(format t "      b. Neck;                          %n")
(format t "                                        %n")
(format t "      c. Back/Trunk/Hip;                %n")
(format t "                                        %n")
(format t "      d. Knees;                         %n")
(format t "                                        %n")
(format t "      e. Hands/Wrists/Arms;             %n")
(format t "                                        %n")
(format t "                                        %n")
(format t "      f. MAIN MANU  of the system       %n")
(format t "                                        %n")
(format t "   ___________________________________ %n")
(assert (next-step)))

(defrule number-selection
?select <- (next-step)
=>
(retract ?select)
```

```
(printout t "Please select a,b,c,d,e or f and then press Enter to choose the part of your
check." crlf crlf)
(bind ?input(read))
(if (eq a ?input)
  then (assert (part Shoulders)
              (problem shoulder 1))
   else (if (eq b ?input)
       then (assert (part Neck)
                   (problem neck 1))
       else (if (eq c ?input)
           then (assert (part Back/Trunk/Hip)
                       (problem back 1))
         else (if (eq d ?input)
            then (assert (part Knees)
                        (problem knees 1))
             else (if (eq e ?input)
                then (assert (part Hands/Wrists/Arms)
                            (problem hands 1))
                       else (if (eq f ?input)
                       then (assert (main manu))

else
(assert (next-step)))))))))  ; if the user doesn't assert 1 or 2 or 3 or 4 or 5,
                      ; it keeps ask the same question

;;==================
;; Check work time
;;==================

(defrule check-work-time-0
(declare (salience 10))                        ; check time after angle
?duration <- (hour ?s-time)
=>
(retract ?duration)
(printout t " How many hours do you work by this posture per day?" crlf crlf)
(bind ?a-time(read))
(assert    (actual-t ?a-time)
        (standard-t ?s-time)))

(defrule check-work-time-1
(declare (salience 10))
?a-t <- (actual-t ?a-time)
?s-t <- (standard-t ?s-time)
=>
(retract ?a-t ?s-t)
(if ( > ?a-time 24)
```

```
then
    (printout t "Are you on earth? One day has only 24 hours, idiot!" crlf crlf)
    (assert (hour ?s-time))

else
    (if (< ?a-time 0)
    then
        (printout t "Hi buddy, are you sleeping? Time cannot be minus!" crlf crlf)
        (assert (hour ?s-time))
    else
                (assert (a-t ?a-time)
                        (s-t ?s-time))

            (if (>= ?a-time ?s-time)
            then
                (assert (work-time long))
               else
                (assert (work-time short))))))

;;======================
;; Check bending degree
;;======================


(defrule check-bending-degree-0
(declare (salience 15))
?bending-degree <- (degree ?s-degree)
=>
(retract ?bending-degree)
(printout t " What is your bending degree by this posture?" crlf crlf)
(bind ?a-degree(read))
(assert    (actual-d ?a-degree)
        (standard-d ?s-degree)))                    ;tell system the standard bending degree
                                                    ;and the actual bending degree

(defrule check-bending-degree-1
(declare (salience 15))
?a-d <- (actual-d ?a-degree)
?s-d <- (standard-d ?s-degree)
=>
(retract ?a-d ?s-d)
(if (> ?a-degree 90)
then
    (printout t "Hi buddy, are you an acrobat?" crlf crlf)
    (assert (degree ?s-degree))
else
```

```
    (if (< ?a-degree 0)
    then
        (printout t " We use extention and flexion to judge the direction of the degree,
please reenter degree." crlf crlf)
        (assert (degree ?s-degree))
    else

          (assert (a-d ?a-degree)
                 (s-d ?s-degree))

        (if (>= ?a-degree ?s-degree)
        then
            (assert (combined-factor bending-degree too high)
                      (bending degree big))
        else
            (assert (combind-factor bending-degree small)
                              (bending degree small))))))

;;=================
;; Check force
;;=================
(defrule check-force
(declare (salience 15))
?Force <- (force ?s-force)
=>

(printout t " How much is the force (kg)?" crlf crlf)
(bind ?a-force(read))
(assert    (actual-f ?a-force)
       (standard-f ?s-force))                          ;tell system the standard force
                                                       ;and the actual force

(if (>= ?a-force ?s-force)
then
       (retract ?Force)
   (assert (main-factor high hand force))

else
       (retract ?Force)))

;*********************************
;check other part of your body?
;*********************************

(defrule check-other-part              ; it doesn't need a higher salience
?check<-(check other part)
```

```
=>
(retract ?check)
(if (yes-or-no-p "If you have checked your body, do you want to know how much
WMSD hazards you have(yes/no)?")
then
(assert (print WMSD)
     (check again))
else
(assert (check again))))

(defrule check-again
?check-a<-(check again)
=>
(retract ?check-a)
(if (yes-or-no-p "Do you want to check other part of your body(yes/no)?")
then
(assert (list))
else
(assert (leave system))))

;;;*******************
;;;leave the system
;;;*******************

(defrule leave-system
?leave <- (leave system)
=>
(retract ?leave)
(if (yes-or-no-p "Do you want to leave CLIPS (yes/no)?")
then
(exit)
else
(if (yes-or-no-p "Do you want to see other part of our system
(yes/no)?" )
        then
        (assert (main manu))
        else
        (printout t "See you, sweetie! Your choice makes you leave the system, I'll miss
you!" crlf))))

;;==============================
;; go to main manu of the system
;;==============================

(defrule main-manu
?main <- (main manu)
```

```
=>
(retract ?main)
(batch "C:/CLIPS/program/notepad/final/try-0.bat"))

;******************************************
;*                    REPORT                    *
;******************************************

;;========================================
;; Awkward posture  -  to report duration
;;========================================

(defrule physical-risk-factor-report
(declare (salience 10))
?R-0 <- (REPORT awk)
?main <- (physical-f $?physical)
=>
(retract ?R-0 ?main)
(printout t " The potential physical risk factor of WMSD is " ?physical  crlf crlf)
(assert (report duration)))

;;==============================
;;  Main problem factors Report
;;==============================

(defrule main-factor-report
(declare (salience 10))
?R-1 <- (REPORT main)
?main <- (main-factor $?main-factor)
=>
(retract ?R-1 ?main)
(printout t " The main risky factor of WMSD is " ?main-factor  crlf crlf)
(assert (report combined factor)))

;;=========================================================
;; combined problem factors report - it's only for combined factor "with high repetitive
motion"
;;=========================================================

(defrule combined-factor-report
(declare (salience 10))
?R-2 <- (report combined factor)
?c-factor <- (combined-factor $?combined)
=>
(retract ?R-2 ?c-factor )
(printout t "Meanwhile, you have combined WMSD risky factor " ?combined crlf crlf)
```

```
(assert (report duration)))

;;================================
;; When there is no combine factor
;;================================

(defrule no-combinded-factor
(declare (salience 10))
?R-2 <- (report combined factor)
(not (combined-factor $?combined))
=>
(retract ?R-2)
(assert (report duration)))

;;____________________________________________________________

;;This section is for degree report


;;==================
;; Main factor report
;;==================

(defrule main-factor-report-connect-to-degree
(declare (salience 10))
?R-1 <- (REPORT with degree)
?main <- (main-factor $?main-factor)
=>
(retract ?R-1 ?main)
(printout t " The main risky factor of WMSD is " ?main-factor  crlf crlf)
(assert (report combined degree)))

;;===========================
;; Degree report
;;===========================

(defrule bending-degree-report-big
(declare (salience 10))
?R-4-1<- (report combined degree)
?big <- (bending-degree big)
?standard-degree <- (s-d ?s-degree)
?actual-degree <- (a-d ?a-degree)
=>
(retract ?R-4-1 ?big ?standard-degree ?actual-degree )
(printout t "The bending degree by this posture is too big." crlf crlf)
(printout t "The acutal bending degree is " ?a-degree " degree" crlf crlf)
```

```
(printout t "The bending degree should be smaller than " ?s-degree " degree." crlf crlf)
(assert (report duration)))


(defrule bending-degree-report-small
(declare (salience 10))
?R-4-1<- (report combined degree)
?small <- (bending-degree small)
?standard-degree <- (s-d ?s-degree)
?actual-degree <- (a-d ?a-degree)
=>
(retract ?R-4-1 ?small ?standard-degree ?actual-degree )
(printout t "The bending degree by this posture is not too big." crlf crlf)
(printout t "The acutal bending degree is " ?a-degree " degree" crlf crlf)
(printout t "The bending degree should be smaller than " ?s-degree " degree." crlf crlf)
(assert (report duration)))

;;==========================
;; When no degree dievation
;;==========================

(defrule no-bending-degree
(declare (salience 10))
?R-4-1<- (report combined degree)
(not(bending-degree big))
(not(bending-degree small))
=>
(retract ?R-4-1)
(assert (report duration)))

;;____________________________________________________________________

;;=======================================
;; work time report -   from every part
;;=======================================

(defrule work-time-report-long
(declare (salience 10))
?R-3-1<- (report duration)
?long <- (work-time long)
?standard-time <- (s-t ?s-time)
?actual-time <- (a-t ?a-time)
=>
(retract ?R-3-1  ?long ?standard-time ?actual-time )
(printout t "The working time by this posture is too long." crlf crlf)
(printout t "The acutal working time is " ?a-time " hours" crlf crlf)
```

```
(printout t "The working time should be shorter than " ?s-time " hours." crlf crlf)
(assert (work time long)))


(defrule work-time-report-short
(declare (salience 10))
?R-3-1<- (report duration)
?short <- (work-time short)
?standard-time <- (s-t ?s-time)
?actual-time <- (a-t ?a-time)
=>
(retract ?R-3-1 ?short ?standard-time ?actual-time )
(printout t "The working time by this posture is okay." crlf crlf)
(printout t "The acutal working time is " ?a-time " hours" crlf crlf)
(printout t "The working time should be shorter than " ?s-time " hours." crlf crlf))

;;The upper part is for the report to the section of high hand force motion
;;_____________________________________________________________________________


;;============
;; flexion 30
;;============

(defrule high-hand-force-flexion
(declare (salience 20))
?t <- (h ?s-time)
?problem <- (flexion)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in flexion (yes/no)?")
then
        (printout t "Bend wrist in flexion." crlf crlf)
        (assert (degree 30)
                (hour ?s-time))                 ; if has angle, then tell system the working
time
else
        (assert (extention)))) ;          if no flexion, then check extention

;;=============
;; extenion 45
;;=============

(defrule high-hand-force-extention
(declare (salience 20))
?t <- (h ?s-time)
```

```
?problem <- (extention)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in extention (yes/no)?")
then
        (printout t " Bend wrist in extention. " crlf crlf)
        (assert (degree 45)
                (hour ?s-time))

else
        (assert (ulnar))))        ; if no extention, then check if there is ulnar

;;==================
;; ulnar deviation
;;==================

(defrule high-hand-force-ulnar
(declare (salience 20))
?t <- (h ?s-time)
?problem <- (ulnar)
=>
(retract ?problem ?t)
(if(yes-or-no-p "Do you bend your wrist in ulnar deviation (yes/no)?")

then
        (printout t "Bend wrist in ulnar deviation. " crlf crlf)
        (assert (degree 30)
                (hour ?s-time))

else
        (assert (no other risk factors))))

;;=========================================
;; When no other riks factors with gripping or pinching
;;=========================================

(defrule no-other-risk-factors
(declare (salience 20))
?no-risk-t <- (no-other-risk ?s-time)
?problem <- (no other risk factors)
=>
(retract ?problem ?no-risk-t)
(assert (hour ?s-time)
        (REPORT main)
        (report WMSD)))
```

```
;;=====================
;; WMSD hazard report
;;=====================

(defrule WMSD-harzard-report
?S-R <- ( report WMSD)                    ; tell the system to report
?PRF <- (physical risk factor)
?overtime <- (work time long)
=>
(retract ?S-R ?PRF ?overtime)
(assert ( WMSD alert)))

(defrule WMSD-no-over-time
?S-R <- ( report WMSD)
?PRF <- (physical risk factor)
(not(work time long))
=>
(retract ?S-R ?PRF))

(defrule WMSD-no-PRF
?S-R <- ( report WMSD)
?overtime <- (work time long)
(not(physical risk factor))
=>
(retract ?S-R ?overtime))

;;;*************************
;;;* WMSD Alert            *
;;;*************************

(defrule WMSD-state-conclusions ""
  (declare (salience 10))
  ?alert <- (WMSD alert)
  =>
  (retract ?alert)
  (assert (count))
  (printout t "===================================" crlf crlf)
  (printout t "  Alert:  Here is a  WMSD hazard !"   crlf crlf)
  (printout t "==================================" crlf)
  (assert (have WMSD-risk)))

;;*******************************************************
;;                    QUERY RULES
;;*******************************************************

;;;=====================
```

```
;;;Check the body part
;;;====================

(defrule check-the-body-part
?body <- (part ?part)
=>
(assert (body ?part))
(retract ?body)
(printout t "Let's check the "  ?part " part." crlf crlf))

;;****************************
;;  PROBLEM  ANALYSIS
;;****************************

;;==================================
;; About awkward postures - Shoulders
;;==================================

(defrule awkward-posture-shoulders-1
(body Shoulders)
?problem <- (problem shoulder 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your hand(s) above the head or the elbow(s) above
the shoulders (yes/no)? ")
then
        (assert (physical-f awkward posture at shoulders)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)                                   ; output a report
                (problem shoulder 2))

else
        (assert  (problem shoulder 2))))

(defrule awkward-posture-shoulders-2
(body Shoulders)
?problem <- (problem shoulder 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you repetitively raise your hand(s) above the head or the elbows
above the shoulder(s)
more often than once per minute (yes/no)? ")
then
        (assert (physical-f awkward posture at shoulders)
```

```
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)                          ; output a report
                (problem shoulder high repetitive motion))

else
        (assert (problem shoulder high repetitive motion))))

;;==================================
;;Awkward Posture Neck
;;==================================

(defrule awkward-posture-neck-1
(body Neck)
?problem<- (problem neck 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your neck bending without support or ability to vary
posture(yes/no)?")
then
        (assert (physical-f awkward posture at neck)
                (physical risk factor)
                (degree 45)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (check other part))
else
        (assert (check other part))))

;;===========================
;; Awkward Posture Back
;;===========================

(defrule awkward-posture-back-1
(body Back/Trunk/Hip)
?problem <- (problem back 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your back bent forward(without support, or the
ability to very posture) (yes/no)? ")
then
        (assert (physical-f awkward posture at back)
                (physical risk factor)
                (degree 30)
```

```
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem back 2))
else
(assert (problem back 2))))

(defrule awkward-posture-back-2
(body Back/Trunk/Hip)
?problem <- (problem back 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you work with your back bent forward near 45 degree (without
support or the ability to vary posture)(yes/no)? ")
        then
        (assert (physical-f awkward posture at back)
                (physical risk factor)
                (degree 45)
                (hour 2)
                (REPORT awk)
                (report WMSD)
                (check other part))
        else
        (assert (check other part))))           ; finish back part and check other part

;;;================================
;;; Awkward Posture Knees
;;;================================

(defrule awkward-knees-squatting
(body Knees)
?problem <- (problem knees 1)
=>
(retract ?problem)
(if (yes-or-no-p "Do you squat while working(yes/no)? ")
        then
        (assert (physical-f awkward posture at knees by squatting)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem knees 2))
        else
        (assert (problem knees 2))))

(defrule awkward-knees-kneeling
```

```
(body Knees)
?problem <- (problem knees 2)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have kneeling while working(yes/no)? ")
        then
        (assert (physical-f awkward posture at knees by kneeling)
                (physical risk factor)
                (hour 4)
                (REPORT awk)
                (report WMSD)
                (problem knees 3))
        else
        (assert  (problem knees 3))))

;;==========================
;; Repeated Impact Knees
;;==========================

(defrule knees-repeated-impact
(body Knees)
?problem <- (problem knees 3)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the knee as a hammer more than once per minute (yes/no)?
")
        then
        (assert (physical-f repeated impact at knees by use knee as a hammer)
                (physical risk factor)
                (hour 2)
                (REPORT awk)
                (report WMSD)
                (check other part))
        else
        (assert  (check other part))))


;;========================================
;; High Hand Force - Pinching
;;========================================

(defrule high-hand-force-1-1
(body Hands/Wrists/Arms)
?problem <- (problem hands 1)
=>
(retract ?problem)
```

```
(if(yes-or-no-p "Do you need to pinch an unsupported object(s) while working
(yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)
                (main-factor pinching an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 0.7)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 1-2))
        else
        (assert (flexion)
                (force 0.7)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 1-2)))          ;check flexion -> extention -> ulnar-> no
other
                                               ; ->risk factors
else
(assert (problem hands 1-2))))


(defrule high-hand-force-1-2
(body Hands/Wrists/Arms)
?problem <- (problem hands 1-2)
=>
(retract ?problem)
(if(yes-or-no-p "Do you just pinch an object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)
                (main-factor pinching an object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 1.5)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-1))
        else
```

```
        (assert (flexion)
                (force 1.5)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-1))) ; check degrees starting from flexion
else
        (assert (problem hands 2-1))))          ;go to gripping


;;==============================
;; High Hand Force - Gripping
;;==============================


(defrule high-hand-force-2-1
(body Hands/Wrists/Arms)
?problem <- (problem hands 2-1)
=>
(retract ?problem)
(if(yes-or-no-p "Do you need to grip an unsupported object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4)       ; when there is no other factors,
                                        ; the system know the working time is 4 hours

                (main-factor gripping an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-2))
        else
        (assert (flexion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hands 2-2)))           ;check flexion -> extention -> ulnar

else
(assert (problem hands 2-2))))
```

```
(defrule high-hand-force-2-2
(body Hands/Wrists/Arms)
?problem <- (problem hands 2-2)
=>
(retract ?problem)
(if(yes-or-no-p "Do you just need to grip an object(s) while working (yes/no)?")
then
        (assert (physical risk factor)
                (no-other-risk 4 hours)
                (main-factor gripping an unsupported object while working))
        (if(yes-or-no-p "Do you have highly repetitve motion meanwhile (yes/no)?")
        then
        (assert (combined-factor highly repetitive motion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hand high repetitive motion))
        else
        (assert (flexion)
                (force 4)
                (h 3)
                (REPORT main)
                (report WMSD)
                (problem hand high repetitive motion)))
else
        (assert (problem hand high repetitive motion))))


;;=======================================
;; Highly Repetitve Motion - shoulders
;;=======================================


(defrule high-repetitive-motion-shoulder
(body Shoulders)
?problem <- (problem shoulder high repetitive motion)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the same motion with little or no variation every few
seconds (excluding keying activities) (yes/no)? ")
then

        (assert (main-factor higly repetitive motion at shoulders)
                (physical risk factor)
```

```
                (hour 6)
                (REPORT main)
                (report WMSD)
                (check other part))
else
        (assert (check other part))))


;;=========================================
;; Highly Repetitve Motion - hands
;;=========================================

(defrule high-repetitive-motion-hands
(body Hands/Wrists/Arms)
?problem <- (problem hand high repetitive motion)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the same motion with little or no variation every few
seconds (excluding keying activities) (yes/no)? ")
then

        (assert (main-factor higly repetitive motion at shoulders)
                (no-other-risk 6)
                (physical risk factor)
                (h 2)
                (flexion)
                (REPORT main)
                (report WMSD)
                (high force exertions))
else
(assert (intensive keying))))    ; go to intensive keying

(defrule high-force-exertion
?problem <- (high force exertions)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have high, forcefl exertions with hands(yes/no)? ")
then
        (assert (combined-factor high, forcefl exertions with hands)
                (h 2)
                (report combined factor)
                (intensive keying))
else
        (assert (intensive keying)))) ; to check intensive keying
```

```
(defrule high-repetitive-motion-4
(or     (body Neck)
        (body Hands/Wrists/Arms))
?problem <- (intensive keying)
=>
(retract ?problem)
(if (yes-or-no-p "Do you have intensive keying(yes/no)? ")
then
        (assert (main-factor intensive keying)
                (physical risk factor)
                (no-other-risk 7)
                (flexion)
                (h 4)
                (REPORT main)
                (report WMSD)
                (repeated impact hands))
else
        (assert (check other part)
                (repeated impact hands))))

(defrule problem-upper-body-9
(body Hands/Wrists/Arms)
?problem <- (repeated impact hands)
=>
(retract ?problem)
(if (yes-or-no-p "Do you use the hand as a hammer more than once per minute (yes/no)?
")
        then
        (assert (main-factor repeated impact)
                (physical risk factor)
                (hour 2)
                (REPORT main)
                (repot WMSD)
                (check other part))
        else
        (assert  (check other part))))          ; finish the hands part and go to other part
```

Cal-f.txt

```
;;*************************************************
;;*                                               *
;;*    Program for calculating the lifting weight *
;;*    CLIPS version 6.0                          *
;;*                                               *
;;*                                               *
;;*************************************************

;;****************
;;* DEFFUNCTIONS *
;;****************

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question crlf crlf)
  (bind ?answer (read))
  (if (lexemep ?answer)
     then (bind ?answer (lowcase ?answer)))
  (while (not (member ?answer ?allowed-values)) do
    (printout t ?question crlf crlf)
    (bind ?answer (read))
    (if (lexemep ?answer)
       then (bind ?answer (lowcase ?answer))))
  ?answer)

(deffunction yes-or-no-p (?question)        ; yes-or-no-p can have more arguments than
one
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
     then TRUE
     else FALSE))

;;;============================
;;; Start to calculate    step 1
;;;============================

(deffacts start
(calculation start))
(defrule start-calculate-lift-weight
?calculate <-(calculation start)
=>
(retract ?calculate)
(printout t
"=========================================================="
crlf crlf)
```

```
(printout t "Let's check whether your lifting weight has any WMSD hazard." crlf crlf)
(printout t
"============================================================"
crlf crlf)
(assert (step 1)))

(defrule step-1
?step <- (step 1)
=>
(retract ?step)
(printout t "What is the actual lifting weight you are going to lift (kg)?" crlf crlf)
(bind ?actual-weight (read))
(assert (a-w ?actual-weight))
(assert (step 2)))

(defrule step-2
?step <- (step 2)
=>
(retract ?step)
(printout t "Where are your hands when you begin to lift or lower the object?" crlf crlf)

(format t "%n                                          %n")
(format t "      a. Above shoulder.                          %n")
(format t "                                       %n")
(format t "      b. Waist to shoulder                        %n")
(format t "                                       %n")
(format t "      c. Knee to waist                          %n")
(format t "                                       %n")
(format t "      d. Below knee                             %n")
(format t "                                       %n")
(assert (select letter)))

(defrule letter-selection
?select <- (select letter)
=>
(retract ?select)
(bind ?v(read))
(if (eq a ?v)
  then (assert (level above shoulder)
               (cal shoulder))
   else (if (eq b ?v)
       then (assert (level waist to shoulder)
                    (cal waist))
       else (if (eq c ?v)
           then (assert (level knee to waist)
                        (cal knee-waist))
```

```
        else (if (eq d ?v)
           then (assert (level below knee)
                        (cal below-knee))

else
(assert (step 2)))))))



;=====================================================
; Step 2 - find the ditance between the body and the object
;=====================================================

(defrule above-shoulder
?le <- (level $?L)
?cl <- (cal $?C)
=>
(retract ?le ?cl)
(printout t "What is the distance between the object and your body?" crlf crlf)
(printout t "      a.  0 cm - 18 cm   " crlf crlf)
(printout t "      b. 18 cm - 30 cm   " crlf crlf)
(printout t "      c.      > 30 cm   " crlf crlf)
(assert (lel $?L)
      (calcu $?C)))


;;===================================
;; above shoulder range
;;===================================


(defrule above-shoulder-select
?le <- (lel above shoulder)
?cl <- (calcu shoulder)

=>
(retract ?le ?cl)
(bind ?distance(read))
(if (eq a ?distance)
  then (assert (unaj-w-limit 24)
             (step 3))

   else (if (eq b ?distance)
       then (assert (unaj-w-limit 15)
                   (step 3))
```

```
        else (if (eq c ?distance)
            then (assert (unaj-w-limit 11)
                           (step 3))

else
(assert (level above shoulder)
        (cal shoulder))))))

;========================================
; Unadjusted weight limit waist to shoulder
;========================================

(defrule waist-to-shoulder
?le <- (lel waist to shoulder)
?cl <- (calcu waist)

=>
(retract ?le ?cl)
(bind ?distance(read))
(if (eq a ?distance)
  then (assert (unaj-w-limit 26)
              (step 3))

   else (if (eq b ?distance)
       then (assert (unaj-w-limit 19)
                      (step 3))

       else (if (eq c ?distance)
           then (assert (unaj-w-limit 15)
                          (step 3))

else
(assert (level waist to shoulder)
        (cal waist))))))


;========================================
; Unadjusted weight limit knee to waist                all the units are in kg
;========================================

(defrule knee-to-waist-select
?le <- (lel knee to waist)
?cl <- (calcu knee-waist)
```

```
=>
(retract ?le ?cl)
(bind ?distance(read))
(if (eq a ?distance)
  then (assert (unaj-w-limit 34)
              (step 3))

   else (if (eq b ?distance)
       then (assert (unaj-w-limit 21)
                   (step 3))

       else (if (eq c ?distance)
           then (assert (unaj-w-limit 15)
                       (step 3))

else
(assert (level waist to shoulder)
        (cal waist))))))

;==========================================
; Unadjusted weight limit below knee
;==========================================


(defrule below-knee-select
?le <- (lel below knee)
?cl <- (calcu below-knee)
=>
(retract ?le ?cl)
(bind ?distance(read))
(if (eq a ?distance)
  then (assert (unaj-w-limit 26)
              (step 3))

   else (if (eq b ?distance)
       then (assert (unaj-w-limit 19)
                   (step 3))

       else (if (eq c ?distance)
           then (assert (unaj-w-limit 13)
                       (step 3))

else
(assert (level above shoulder)
        (cal shoulder))))))
```

```
;========================================
; Step 3 Find the limit reducion modifier
;========================================


(defrule find-modifier-1
?step <- (step 3)
=>
(retract ?step)
(printout t "How many lifts do you have per minute?" crlf crlf)
(format t "%n                                           %n")
(format t "      a.  less than once every 5 mins    %n")
(format t "                                          %n")
(format t "      b.  1 lift every 2-5 mins;         %n")
(format t "                                          %n")
(format t "      c.  1 lift every  min;             %n")
(format t "                                          %n")
(format t "      d.  2-3 lifts every  min;          %n")
(format t "                                          %n")
(format t "      e.  4-5 lifts every  min;          %n")
(format t "                                          %n")
(format t "      f.  6-7 lifts every  min;          %n")
(format t "                                          %n")
(format t "      g.  8-9 lifts every  min;          %n")
(format t "                                          %n")
(format t "      h.  10+ lifts every  min;          %n")
(format t "                                          %n")
(assert (step 3-1)))


(defrule find-modifier-2
?step<- (step 3-1)
=>
(retract ?step)
(bind ?mod-1(read))

(if (eq a ?mod-1)
then (assert (less than 1)
(step 3-2))

else (if (eq b ?mod-1)
then (assert (lift every 2-5 min 1)
(step 3-2))

else (if (eq c ?mod-1)
then (assert (lift every min 1)
```

```
(step 3-2))

else (if (eq d ?mod-1)
then (assert ( lifts every min 2-3)
(step 3-2))

else (if (eq e ?mod-1)
then (assert (lifts every  min 4-5)
(step 3-2))

else (if (eq f ?mod-1)
then (assert (lifts every  min 6-7)
(step 3-2))

else (if (eq g ?mod-1)
then (assert (lifts every  min 8-9 )
(step 3-2))

else (if (eq h ?mod-1)
then (assert (lifts every  min 10+ )
(step 3-2))

else
(assert (step 3))))))))))


(defrule find-modifier-3
?step <- (step 3-2)
=>
(retract ?step)
(printout t "For how many hours per day?" crlf crlf)
(format t "%n                                %n")
(format t  "        a.  1 hour or less        %n")
(format t  "                                  %n")
(format t  "        b.  1 hour to 2 hours;    %n")
(format t  "                                  %n")
(format t  "        c.  2 hours or more;      %n")
(format t  "                                  %n")
(assert (step 3-3)))


(defrule find-modifier-4
?step <- (step 3-3)
=>
(retract ?step)
(bind ?mod-2(read))
```

```
(if (eq a ?mod-2)
        then (assert (colum 1))

        else (if (eq b ?mod-2)
                then (assert (colum 2))

                else (if (eq c ?mod-2)
                        then (assert (colum 3))

else
(assert (step 3-2))))))

;=========================
; Get the exact modifier
;=========================

;============================================
; less than once very five minutes use 1.0
;============================================


(defrule get-modifier-1-1
?m-1 <- (less than 1)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 1.0)
(step 4)))

(defrule get-modifier-1-2
?m-1 <- (less than 1)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 1.0)
(step 4)))

(defrule get-modifier-1-3
?m-1 <- (less than 1)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 1.0)
(step 4)))

;=====================
```

```
; 1 lift / 2-5 mins.
;===================


(defrule get-modifier-2-1
?m-1 <- (lift every 2-5 min 1)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 1.0)
(step 4)))

(defrule get-modifier-2-2
?m-1 <- (lift every 2-5 min 1)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.95)
(step 4)))

(defrule get-modifier-2-3
?m-1 <- (lift every 2-5 min 1)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.85)
(step 4)))

;===========================
; 1 lift every min
;===========================

(defrule get-modifier-3-1
?m-1 <- (lift every min 1)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.95)
(step 4)))

(defrule get-modifier-3-2
?m-1 <- (lift every min 1)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
```

```
(assert (modifier 0.9)
(step 4)))

(defrule get-modifier-3-3
?m-1 <- (lift every min 1)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.75)
(step 4)))


;===========================
; 2-3 lifts / min
;===========================

(defrule get-modifier-4-1
?m-1 <- (lifts every min 2-3)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.9)
(step 4)))

(defrule get-modifier-4-2
?m-1 <- (lifts every min 2-3)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.85)
(step 4)))

(defrule get-modifier-4-3
?m-1 <- (lifts every min 2-3)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.65)
(step 4)))

;=======================
; 4-5 lifts/min
;=======================

(defrule get-modifier-5-1
?m-1 <- (lifts every  min 4-5)
```

```
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.85)
(step 4)))

(defrule get-modifier-5-2
?m-1 <- (lifts every  min 4-5)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.7)
(step 4)))

(defrule get-modifier-5-3
?m-1 <- (lifts every  min 4-5)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.45)
(step 4)))

;====================
; 6-7 lifts/min
;====================

(defrule get-modifier-6-1
?m-1 <- (lifts every  min 6-7)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.75)
(step 4)))

(defrule get-modifier-6-2
?m-1 <- (lifts every  min 6-7)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.5)
(step 4)))

(defrule get-modifier-6-3
?m-1 <- (lifts every  min 6-7)
?m-2 <- (colum 3)
=>
```

```
(retract ?m-1 ?m-2)
(assert (modifier 0.25)
(step 4)))

;======================
; 8-9 lifts/min
;======================

(defrule get-modifier-7-1
?m-1 <- (lifts every  min 8-9)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.6)
(step 4)))

(defrule get-modifier-7-2
?m-1 <- (lifts every  min 8-9)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.35)
(step 4)))



(defrule get-modifier-7-3
?m-1 <- (lifts every  min 8-9)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.15)
(step 4)))

;=================
; 10+ lifts / min
;=================

(defrule get-modifier-8-1
?m-1 <- (lifts every  min 10+)
?m-2 <- (colum 1)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.3)
(step 4)))
```

```
(defrule get-modifier-8-2
?m-1 <- (lifts every  min 10+)
?m-2 <- (colum 2)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.2)
(step 4)))

(defrule get-modifier-8-3
?m-1 <- (lifts every  min 10+)
?m-2 <- (colum 3)
=>
(retract ?m-1 ?m-2)
(assert (modifier 0.0)
(step 4)))

;=====================================
; Step 4  Calculate the Weight Limit
;=====================================

(defrule twist-or-not
?actual <- (unaj-w-limit ?uwl)
?step <- (step 4)
=>
(retract ?step ?actual)
(if (yes-or-no-p "Do you twist more than 45 degree while lifting (yes/no)? ")
then
        (bind ?a-weight(* 0.85 ?uwl))
        (bind ?a-weight(+ 0.5 ?a-weight))
        (bind ?a-weight(integer ?a-weight))
        (assert (adjust-w-l ?a-weight))
        (printout t "Since you are twisting while lifting, your adjusted weight limit is : "
?a-weight " kg." crlf crlf)
        (assert (step 4-1))
else

(assert (a-w ?uwl)
        (step 4-1))))

;============================================================
==
; calculate weight limit - adjusted - limit reducion modifier * adjusted weight limit
;============================================================
==

(defrule weight-limit-adjusted
```

```
?mod<-(modifier ?m)
?adj<-(adjust-w-l ?a-weight)
?step <- (step 4-1)
=>
(retract ?step ?mod ?adj)
(bind ?w-l (* ?m ?a-weight))
(bind ?w-l (+ 0.5 ?w-l))
(bind ?w-l (integer ?w-l))
(assert (weight-limit ?w-l))
(printout t "Your weight limit is " ?w-l " kg." crlf)
(assert (step 5)))

;=====================================
; calculate weight limit - unadjusted
;=====================================

(defrule weight-limit-unadjusted
?mod<-(modifier ?m)
?adj<-(a-w ?uwl)
?step <- (step 4-1)
=>
(retract  ?mod ?adj ?step)
(bind ?w-l (* ?m ?uwl))
(bind ?w-l(+ 0.5 ?w-l))
(bind ?w-l(integer ?w-l))
(printout t "Your weight limit is " ?w-l " kg." crlf)
(assert (weight-limit ?w-l)
      (step 5)))

;;==============================
;;  Step 5  Is this a hazard?
;;==============================

(defrule hazard-or-not
?step <- (step 5)
?weight-l <- (weight-limit ?w-l)
?weight-a <- (a-w ?actual-weight)
=>
(retract ?step ?weight-l ?weight-a)
(if (> ?actual-weight ?w-l)
then
(printout t "This lifting is a WMSD hazard ! " crlf crlf)
(assert (try again))
else
(printout t "This lifting has no WMSD hazard." crlf crlf)
(assert (try again))))
```

```
;;=====================
;; try again ?
;;=====================

(defrule try-again
?try <- (try again)
=>
(retract ?try)
(if (yes-or-no-p "Try again (yes/no)? ")
then (assert (calculation start))
else
(assert (other part of the system))))

;;========================
;; Want to see other part?
;;========================

(defrule other-part
?other-p <- (other part of the system)
=>
(retract ?other-p)
(if (yes-or-no-p "Do you want to run other part of the system (yes/no)? ")
then
(batch "C:/CLIPS/program/notepad/final/try-0.bat")
else
(assert (leave system))))

;;===================
;; leave the system
;;===================

(defrule leave-system
?leave <- (leave system)
=>
(retract ?leave)
(if (yes-or-no-p "Do you want to leave CLIPS(yes/no)?")
then
(exit)
else))
```