Bone-FixES: An External Skeletal Fixation Expert System

W.D. Potter[1,3], D. N. Aron[2], T. Wang[1], and G. Greene[3]
(long paper)
[1]Artificial Intelligence Center
[2]College of Veterinary Medicine
[3]Department of Computer Science
University of Georgia, Athens, GA, 30605
(Contact W.D. Potter at potter@cs.uga.edu)

Abstract

Bone-FixES is an expert system currently under development to aid student and clinical practitioner decision making for construction of external skeletal fixation devices. External skeletal fixation devices (also called fixators) are used to secure broken bone segments during the healing process. For teaching, Bone-FixES will be used to provide feedback to students given a clinical scenario and asked to construct an appropriate fixator. The current system analyzes patient factors and fixator construction to assess the quality of the student's solution to the scenario. In practice, Bone-FixES will be used to assist veterinarians with fixator construction via analysis and recommendations. The development of Bone-FixES follows the tradition knowledge engineering approach of rapid prototyping within a prolog environment using a simple development shell.

Keywords: Expert System, External Skeletal Fixation

1. Introduction

The Bone Fixation Expert System (Bone-FixES) is an expert system currently under development to aid student and clinical practitioner decision making for construction of external skeletal fixation (ESF) devices. External skeletal fixation devices (also called fixators) are used to secure broken bone segments during the healing process. A fixator provides support while an animal's fractured limb heals yet allows the animal to move around with only moderate inconvenience. Typically, animals are not as cooperative as humans when following instructions to stay off a broken leg until healing occurs. Consequently, a device that provides sufficient support is needed instead of a simple cast used to hold a limb stable.

Designing and constructing a fixator requires extensive knowledge of physiology and mechanical engineering. For example, the optimal placement of securing pins and connecting rods from a mechanical point of view may not be optimal from the biological point of view. This concept of patient-device balance is paramount for students and practitioners, and the prime motivation for Bone-FixES as a teaching tool. Bone-FixES will be used to provide feedback to students who are given a clinical scenario and asked to design an appropriate fixator. The current system analyzes patient factors and fixator design to assess the quality of the student's solution to the scenario. An assessment is given along with recommendations, for example the fixator may be rated as only mildly satisfactory due to the student's use of smooth pins instead of threaded pins.

In clinical practice, Bone-FixES will be used to assist veterinarians with fixator design via analysis and recommendations. It will be used as an assistant to help the veterinarian with any and all aspects of the design. In addition to giving recommendations on fixator design, Bone-FixES will also provide an analysis of the patient's suitability via the widely accepted Fracture Patient Assessment Score (FPAS) that combines bone condition, cause of fracture, and type of patient to derive the FPAS.

The development of Bone-FixES follows the tradition knowledge engineering approach of rapid prototyping within a prolog environment using a simple development shell. In the following sections, we present brief overviews of expert systems and external skeletal fixation. These are followed by brief details of the development environment and Bone-FixES. A brief discussion with future directions follows.

2. Expert Systems

The field of Artificial Intelligence (AI) has experienced a rapid growth of research activity and general interest in expert systems. This is due to the successful development of knowledge-based systems (KBS), sometimes known as expert systems (ES) (Stefik et al. 1982, Duda and Shortliffe 1983, Hayes-Roth 1984). A KBS is a computer program that applies AI techniques to real-world problems that are currently solved by human experts. The distinction between a KBS and an ES is well defined by Duda and Shortliffe (Duda and Shortliffe 1983, page 267):

> "The phrase ’knowledge-based system’ is often preferred to ’expert system,’ since there are no uniquely qualified human experts for a large number of AI applications; however, both phrases are sufficiently vague that the latter can be applied to almost any program at all. While usage is far from uniform, we shall define a knowledge-based system as an AI program whose performance depends more on the explicit presence of a large body of knowledge than on the possession of ingenious computational procedures; by expert system we mean a knowledge-based system whose performance is intended to rival that of human experts."

Typically, knowledge-based systems have three major components: the user interface, the knowledge base, and the inference engine. The user interface, which usually entails over 40% of the design and coding effort in KBS development (Smith 1984), provides the user or knowledge engineer with a means for interacting with the system. This is especially true with Bone-FixES due to the finite element analysis feature built into the user interface. The knowledge base contains the domain specific and general control problem-solving knowledge. This knowledge is maintained in some sort of knowledge representation scheme, such as logical predicates, frames, semantic networks, production rules, or some combination of these (Barr and Feigenbaum 1982, Stefik 1995). The inference engine embodies the problem-solving control strategy. The control strategy generally follows either a forward-chaining approach or a backward-chaining approach. The forward-chaining or data-driven approach attempts to derive new knowledge from an existing knowledge set by finding rules whose conditions are satisfied and adding their conclusions to the knowledge set. The backward-chaining or goal-directed approach attempts to satisfy the selected goal by iteratively scanning the knowledge base for sub-goals that support the selected goal and then attempting to satisfy the sub-goals. Eventually, Bone-FixES will have a slightly atypical architecture including a knowledge base, a backward chaining inference engine, a user interface with finite element analysis feature, and a transparent (i.e., built into the inference engine) web session blackboard manager (this will allow Bone-FixES to be accessed via the internet).

3. External Skeletal Fixation

Bone-FixES is an expert system for analyzing and recommending external skeletal fixation (ESF) devices used to support fractured limbs. An ESF system is made up of a collection of transfixation pins, an external frame (rods and clamps), and in some cases a special pin called an intramedullary pin. Transfixation pins are used to pierce soft tissue and attach to segments of bone. A pin may be either threaded or smooth, and may anchor to the bone (a half pin) or extend past the soft tissue on the other side of the bone (a full pin). The external portion of the pins is connected via connecting rods and clamps to hold the structure in a fixed position. The intramedullary pin, when used, is placed within the bone running along the interior of the bone from one end to the other. Depending on the number of connecting rods, the number of different planes and whether the pins are located on one side or both sides (e.g., full pins transfixing the bone and extending past the soft tissue on both sides), a fixator can be categorized as Type I, Type II, or Type III. Typically, a Type I fixator is unilateral (using half pins) and uniplanar (all pins and connecting rods in the same spatial plane). A Type I fixator can also be biplanar. A Type II fixator is bilateral with full pins extending on both sides of the limb with a connecting rod securing each side. A Type III fixator uses both a bilateral and biplanar configuration. Finally, these three basic types may be combined if a special case warrants.

The structural support provided by the various types of fixators increases as the type number increases (i.e., Type III is stronger than Type II which is stronger than Type I). Within a specific fixator type, increasing the number of pins improves overall strength. Also, once the pins are affixed to the bone segments, fixator strength increases as the distance between the connecting rod and bone decreases. Connecting rod support increases when pins are located closer together along the axis of the bone.

However, the patient factors into the fixator equation predominately. It may be the case that the fracture and bone length leave little room for more than two pins per bone segment. Using a bilateral biplanar fixator may simply be out of the question due to fracture location and limb involved. Another patient factor affecting fixator design is age. Older animals have more brittle bones than younger animals and brittle bones may not receive multiple pin configurations well. Fracture location also determines fixator design due to close proximity of other soft tissue of the animal. This reflects on the convenience of use and mobility the animal may have during the healing process.

A special measure has been established in the clinical community, called the Fracture Patient Assessment Score (FPAS), to help veterinarians with fixator design. The FPAS is composed of three terms corresponding to mechanical, biological, and clinical factors of the patient. Each factor ranges from one to ten and is combined to give an overall FPAS from one to ten. Mechanical factors relate to the skeleton itself and have many facets including size of the animal, number and type of limb fractured, type of fracture, and amount of load the fracture can support. Note that load sharing quickens the healing process. Biological factors include length of time to install the fixator on the patient, patient health, and amount of energy exerted to cause the fracture. The third term deals with clinical factors relating to the patient. Examples of clinical factors include the personality of the animal, such as very mild-mannered or very active, and whether or not the animal accepts the fixator well. An additional factor within the clinical term deals with the patient's owner. An involved owner capable of handling the patient during the healing process and able to manage the patient's healing process figures heavily into the FPAS and ultimately into the fixator design.

4. Expert Systems Development

The following are the steps used in developing the Bone-FixES prototype. These include problem identification, knowledge base design, and knowledge base validation (Stefik 1995).

An expert system requires a precise domain. The domain must be well organized and well understood. The selected application within the domain will need to require expert knowledge in order to solve specific problems within that domain. Otherwise, there would be no need for an expert system; a standard algorithmic search scheme would be more suitable in that case. This means that the types of problems encountered within the domain should exhibit combinatorially explosive solution spaces when using an exhaustive search scheme to find a solution. For example, in the diagnosis domain, as the number of disorders (diseases) increases linearly, the number of possible diagnoses increases exponentially (i.e., there are exponentially more combinations of diagnoses to consider). This type of growth in the total number of solutions is called combinatorial explosion. Clearly, external skeletal fixation construction is an appropriate domain for expert systems development.

The knowledge base is the core component of any expert system since it contains the knowledge acquired from an expert in the field (Dr. Aron is the expert providing Bone-FixES knowledge). Typically, a knowledge engineer is responsible for working with an expert to build the knowledge base for the system. The knowledge engineer must perform a detailed analysis of the inference process and develop the prototype knowledge base. The tasks involved in developing the Bone-FixES knowledge base include knowledge acquisition, knowledge representation, knowledge programming, and knowledge refinement.

The objective of knowledge acquisition is to obtain facts and rules from the expert that will allow the system to draw expert level conclusions. The process of knowledge acquisition is very time-consuming and difficult, especially if the knowledge engineer is unfamiliar with the domain. In our case, there are several knowledge engineers working on the project and among them we have sound biological knowledge.

After the acquisition of the knowledge begins, a prototype system implementation is begun to test the early stages of the system. This process involves encoding the expert knowledge into the proper format for the computer. Representing and encoding the facts and relationships that constitute knowledge is the next step in the system implementation. There are many established approaches of representing knowledge, for example, semantic networks, rules and logic expressions. "If - Then" style rules are, however, widely used because they are easy to understand and enhance. They facilitate the addition of an explanation facility early in the development process or as an add-on later in the development. Bone-FixES employs a typical rule-based knowledge representation within a prolog environment with the XSHELL development tool (Covington, et al 1997).

During the knowledge programming stage, we first design an overall framework and systematic representation scheme based on the rules derived from the expert. The knowledge is assembled into an organized rule base for the inference engine to interpret and use. The process involves coding facts, rules, objects and relationships found in the domain in the programming language for the system.

5. Bone-FixES – the Expert System

The Bone-FixES knowledge base currently includes over 75 rules for deciding the quality of a fixator design. At the top of the rule hierarchy are nine rules that conclude an overall fixator score similar to the FPAS discussed earlier. The overall score combines a mechanical fixator assessment with the FPAS to give the student or veterinarian either an assessment of a fixator design or a fixator design recommendation. The system is coded in prolog and currently interacts with the user via a menu based front end. We use the XSHELL approach for handling various queries to the user such as menus, yes-no prompts, and fill-in-the-blank queries. XSHELL provides the mechanism to handle menus and the other types of queries through a series of simple parameter specification rules. Query prompts, possible responses, query type, and fact header are coded within a common structure for easy development and maintenance. For example:

```
xkb_menu(fracture_type, [‘What type of fracture does the patient have?’],
    [‘simple fracture’, 2 or 3 cortical fragments w/o fissures’, ‘comminuted fracture’],
    ‘Please select fracture type:’).
```

This structure shows the internal identifier associated with this rule (fracture_type), the list containing the query string, the possible menu options, and the string containing the instructions. Each item in the menu list is automatically numbered to ease user selection. Once the option is selected, the fact is asserted into the knowledge base. If the inference process (backward chaining) is trying to establish a particular overall score for example and some conditional in the rule fails, the next rule that matches is fired. If the next rule needs a particular item of information that was queried earlier, then the system simply recalls that item from the fact base instead of prompting the user to enter it again.

The rules at the top of the hierarchy are used to guide the inference process through the determination of the fixator score. Components of these top-level rules include the FPAS score, the fixator type, the pin type for each pin, the pin design, the pin distribution, the clamp type, and the number and type of connecting rods. The pin design, as mentioned earlier, includes number of pins, pin diameter, pin placement relative to other pins and bone ends, and pin position with respect to the bone thickness. There are other characteristics that are considered as well which are not currently included. One example is the type of soft tissue through which a pin is being placed. If there are many blood vessels and/or nerves in the immediate area, then this location is considered a poor location to insert a pin.. The following are examples of rules taken directly from the Bone-FixES knowledge base.

```
pin_design(ok) :-
        pin_number(ok),             % check the number of pins in each bone segment
        pin_diameter(ok),           % check pin diameters relative to bone diameter
        pin_position(ok),           % check pin positions relative to bone thickness
        pin_placement(ok).          % check pin placements along each bone segment

pin_placement(not_ok) :-
        parm(distributed, y_n, no),         % are pins evenly spaced along the bone
        parm(avoided, y_n, no).             % do the pins avoid critical soft tissue
```

In the pin_design rule, the design is determined to be alright (ok), if all of the following conditions are satisfied:

1. the number of pins in each bone segment is ok (range should be between 2 and 4)
2. each pin’s diameter relative to the bone at the pin insertion point should be about 25%
3. each pin should traverse the bone at its thickest diameter
4. pins should be equally spaced along the bone segments and at least 2cm from the bone ends and fracture point, plus avoid critical soft tissue.

Notice that this is just one pin_design rule. Currently, there are fifteen other pin_design rules that indicate various design situations such as designs that use excessively large pins or designs that use too few pins per segment. In scoring the fixator, pin design is vital for successful healing. Bone-FixES strives to achieve a balance between pin invasiveness and strength. Ideally the fixator should secure the bone segments and provide just the exact amount of support to allow healing yet not so much to negate any load sharing which aids healing.

In the pin_placement rule, two query prompt conditions receiving negative responses from the user indicate that the pin placement is not very good. The parm predicates invoke yes/no prompts to the user (unless the identified facts already exist in the knowledge base). The first prompt asks about the distribution of pins along the bone segments. Pins should be equally spaced and should be at least 2cm from either the fracture site or the bone end. The second prompt asks whether or not the pins have avoided vital soft tissue. It is very important to avoid inserting pins through soft tissue areas containing numerous blood vessels and nerves. Responding negatively to each prompt satisfies these two conditions that in turn satisfy the poor pin placement conclusion.

6. Future directions

Our plans are to continue to evolve the knowledge base by adding additional rules and revising the existing rules. As we move toward the “volkswagen” version of the prototype, our immediate goal is to integrate the prolog knowledge system with the already operational finite element analysis front end. The front end uses a visual three-dimensional imaging tool for fixator design. That is, the user can indicate the bone type and see the bone image on the screen. Then the user can design their fixator by selecting from a collection of pin types, connecting rods, and clamps. The location of the pins, insertion angle, pin plane, and anchor depth are simulated using the front end’s point and click facility while rotating the bone as needed in three dimensional space. Another feature of the front end is the mechanical analysis that can be performed on the design. The finite element analysis shows pictorially how the fixator design will hold up under various static and dynamic forces.

We already have the communication mechanism designed for passing inputs from the front end to the knowledge engine for the inference process (further analysis and recommendation). Additional information needed by the inference process may be obtained by sending messages to the front end for display to the user. The assessment and recommendations are then sent to the user. We plan to incorporate an extensive explanation facility within the near future that can be used for instructional purposes and recommendation justification. Of course, our ultimate goals are to incorporate Bone-FixES within the veterinary medicine curriculum for use by students and to incorporate the system into your local vet clinic to improve the care provided for fracture management.

References

Barr, A. and E.A. Feigenbaum, eds. (1982) The Handbook of Artificial Intelligence. Vol. 1 & 2. William Kaufmann, Inc., Los Altos, CA.

Benders, J. and F. Manders (1993) Expert systems and organizational decision-making. Information and Management. 25: 207-231.

Bobrow, D.G., S. Mittal and M. Stefik (1986) Expert systems: perils and promises. Communication of the ACM. 29: 880-894.

Covington, M.A, D. Nute, and A. Vellino (1997) Prolog Programming In Depth. Prentice Hall. Upper Saddle River, New Jersey.

Duda, R.O. and E.H. Shortliffe (1983) Expert systems research. Science. 220(4594): 261-268.

Hayes-Roth, F. (1984) Knowledge based expert systems. IEEE COMPUTER. 17(10): 263-273.

Holsapple, C.W. and A.B. Winston (1996) Decision Support Systems – A Knowledge-Based Approach. West Publishing Company. New York.

Smith, R. (1984) On the development of commercial expert systems. The AI Magazine. 5(3): 61-73.

Stefik, M. (1995) Introduction to Knowledge Systems. Morgan Kaufmann, California.

Stefik, M., J. Aikins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth, and E. Sacerdoti (1982) The organization of expert systems: a tutorial. Artificial Intelligence. 18: 135-173.

Turban, E. (1993) Decision Support and Expert Systems: Management Support Systems. 3rd ed., Macmillan, New York.