

A KNOWLEDGE-BASED SYSTEM FOR HOMINID FOSSILS

By

ROBERT D. COOPER

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2004

Copyright 2004

by

Robert D. Cooper

To my wife, Jessica

ACKNOWLEDGMENTS

I would like to thank Dr. Douglas D. Dankel II for being a cochair on my committee and providing great advice and help in the development of this thesis. I would like to thank Dr. Gerhard X. Ritter for being a cochair on my committee and Dr. Beverly Sanders for serving on my committee. I would also like to thank Dr. John S. Krigbaum, Dr. Susan D. deFrance, and Laurie Kauffman of the University of Florida Anthropology Department for aiding me with the biological anthropology portions of this thesis. Most of all, I would like to thank my parents for their love and support throughout my education.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
ONE INTRODUCTION	1
A Knowledge-based System in Human Evolution	2
Species Identifier and the Use of Certainty Factors	2
Thesis Organization	3
TWO BACKGROUND INFORMATION	4
Background Information on Human Evolution	4
Certainty Factors	8
Previous Rules-based Systems	11
Mycin	11
XCON	11
Expert Systems in Anthropology	12
Anthropologist	12
ForDisc	14
Conclusion	14
THREE WALK THROUGH OF SPECIES IDENTIFIER	15
Description of the Graphical User Interface	16
Image Panel	16
Current Conclusions Table	17
Questions Panel	18
A Walk-Through of Species Identifier's Execution	20
Age and Location Questions	20
Questions about Dentition	21
Cranial Questions	25

Species Identifier's Conclusion	33
Conclusion	35
FOUR SPECIES IDENTIFIER DESIGN.....	36
Design of the Species Identifier Engine	36
Fact Structure.....	36
Rule Format	37
Rete Algorithm	38
Format.....	39
Implementation of Certainty Factors.....	40
Implementation of the Engine	42
Step one: check facts	43
Step two: the <i>get-check</i> rule	44
Step three: the <i>ask-question</i> rule	45
Step four: the <i>assemble-GUI</i> function.....	46
Step five: check input for correctness	47
Step six: <i>answer</i> rules.....	49
Step seven: combine certainty factors	50
Implementation of the GUI.....	51
Screen Layout.....	52
Functionality.....	53
Design of the Question Panel	53
Question text	53
Answer input	54
Certainty slider	56
Current conclusions table	57
Image panel	58
Conclusion	58
FIVE CONCLUSION.....	59
Species Identifier Results	59
Future Work.....	60
Summary	61
GLOSSARY	62
LIST OF REFERENCES	64
BIOGRAPHICAL SKETCH	66

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1. List of genus <i>Australopithecus</i> and <i>Ardipithecus</i> characteristics	7
2-2. List of genus <i>Homo</i> characteristics.....	8

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1. A chronology of hominids recognized by Species Identifier.	6
2-2. The form used by Anthropologist to estimate age based on dentition	12
2-3. This form uses cranial measurements to determine the gender of skeletal remains	13
2-4. Selecting your expert in Anthropologist	14
3-1. The first question asked by Species Identifier.	15
3-2. The image panel of Species Identifier.	16
3-3. The current conclusions table of Species Identifier.	17
3-4. The questions panel of Species Identifier.	18
3-5. The slider feature allows users to indicate their confidence in an answer.	19
3-6. Pressing the Help button creates a window providing more information on the question being asked.	19
3-7. An example of an <i>H. neandertalensis</i> skull.	20
3-8. The user indicates that the specimen was found in Europe with 100% certainty.	21
3-9. Species Identifier asks the user about the number of cusps on the lower front premolar.	22
3-10. The application asks if the size of the molars is large or small.	23
3-11. Species Identifier asks about the presence of shovel-shaped incisors.	24
3-12. Species Identifier asks the user to determine the volume of the brain case.	25
3-13. Species Identifier asks the user about the cranial capacity of the specimen in general terms.	26
3-14. The user is asked to determine the degree of facial prognathism.	27

3-15. The application asks if the skull possesses a brow ridge.	28
3-16. Species Identifier asks if there is any sagittal keeling along the top of the skull.....	30
3-17 Since Species Identifier suspects <i>H. neandertalensis</i> could be a conclusion, it asks the user a question specific to that species.....	31
3-18. The application asks if the skull features an occipital bun.....	32
3-19. Species Identifier asks about the slope of the forehead.	33
3-20. Species Identifier displays its conclusion to the examination.....	34
4-1. An example rule in JESS.....	38
4-2. A diagram of the Rete network created to execute an <i>ask-question</i> rule.	39
4-3. The function that calculates certainty factors.....	41
4-4. The source code for the <i>combine-values</i> rule.....	42
4-5. A flowchart of Species Identifier's engine.....	42
4-6. An example of a rule asserting a <i>check</i> fact.....	43
4-7. An example of asserting an <i>ask</i> fact.....	44
4-8. A <i>question</i> fact used by Species Identifier.....	45
4-9. An example of an <i>answer</i> fact.....	48
4-10. An example of how an <i>answer</i> rule appears in the source code.	49

Abstract of Thesis
Presented to the Graduate School of the University of Florida
in Partial Fulfillment of the Requirements for the
Degree of Master of Science

A KNOWLEDGE-BASED SYSTEM FOR HOMINID FOSSILS

By

Robert D. Cooper

May 2004

Chair: Douglas D. Dankel II

Cochair: Gerhard X. Ritter

Major Department: Computer and Information Sciences and Engineering

Determining the species of a set of hominid fossils can be an inexact science for biological anthropologists. Species Identifier is a knowledge-based system for hominid fossils. This application aids in species identification by asking the user a sequence of questions about the existence of certain fossil characteristics. Based on the answers provided, Species Identifier decides which known hominid species, if any, the fossil specimen represents. Although similar applications such as FORDISC are used in the field of forensic anthropology, no knowledge-based software currently exists in biological anthropology.

Species Identifier is developed using Java Expert System Shell (JESS), which is a language similar in syntax to CLISP. JESS was created with Java and allows a developer to easily integrate Java classes and objects into an expert system. This feature allows the use of Java Swing to create Species Identifier's graphical user interface.

Species Identifier implements certainty factors in order to determine the most likely species of a hominid fossil specimen. Certainty factors allow Species Identifier to emulate the body of knowledge a biological anthropologist uses when examining a specimen by assigning various scores to a skeletal trait observed by the user. The accumulation of these scores leads to the final species identification made by the application.

This thesis presents the development of Species Identifier and the implementation of certainty factors using Java Expert System Shell.

CHAPTER ONE INTRODUCTION

Biological anthropology is the study of the evolution of modern humans from their primate ancestry. The process of identifying species from fossil remains and the process of discovering new species from fossil remains can be an inexact science. New discoveries in the field of biological anthropology can prove difficult to classify because of the fragmentary nature of fossil evidence.

Classifying a fossil set or even declaring a new species can lead to great debate and controversy. For example, when fossil remains of what would be the new species *Australopithecus garhi* were uncovered, researchers were unsure if they were looking at one, two, or even more species [GRO04]. Although many accept this species designation, until more evidence is uncovered, skepticism about the new species designation will remain.

There have even been great hoaxes in the field. In 1912, the Piltdown skull was uncovered in England and declared to be the missing link [STE96]. It was not until 1953 that the skull was proven to be a hoax [STE96]. Through the use of more modern dating techniques the age of the skull fragments were found to not match. The jaw of the fossil set was finally determined to be a modified orangutan jaw [STE96].

Many times, determining the species of a fossil set is exceedingly difficult. A software tool to help researchers determine the species of a fossil set, especially in difficult cases, could aid in preventing misclassifications. At a minimum, such a system

provides biological anthropologists with an easy method of acquiring multiple opinions about the classification of a fossil set.

A Knowledge-based System in Human Evolution

A knowledge-based system, which can make decisions to supplement the knowledge of a biological anthropologist, could prove to be an extremely valuable tool. Currently, no such software exists for biological anthropologists. In a related field—forensic anthropology—users have found knowledge-based systems to be an important tool in determining the race, gender, and age of human remains where these factors are not immediately apparent. Such systems are said to be accurate enough to be used in international tribunals for investigating war crimes [COU04].

Species Identifier and the Use of Certainty Factors

The application developed in this thesis is called Species Identifier. The purpose of the application is to interview a user while the user is examining a set of fossils and determine which species the fossils represent. Species Identifier aims to model the decision making process a biological anthropologist uses to classify a set of fossils.

The researcher examines the fossil evidence looking for various characteristics. The characteristics the researcher attempts to identify can be subjective. Researchers use their body of knowledge about hominid fossils to answer questions where there are no quantifiable traits. For example, one characteristic a researcher may attempt to identify is the size of the molars. The molars may be large or small with no exact measurement that determines which is which. However the researcher, knowing how molars of various sizes typically appear, will be able to properly classify the size of the molars.

Species Identifier models this process through the implementation of certainty factors. Certainty factors allow an expert system to emulate the body of knowledge a

biological anthropologist uses when investigating fossils. Species Identifier accumulates evidence from each question that enhances or weakens the measure of belief in each species. At the end of the interview, the species that has the highest confidence is declared as Species Identifier's conclusion.

Thesis Organization

The thesis is organized into five chapters. Chapter Two discusses background information pertaining to biological anthropology, certainty factors, and previously developed rules-based systems. Chapter Three is a walk-through of Species Identifier, providing a view of its execution. Chapter Four provides a detailed analysis of the implementation of the certainty factor engine utilized by Species Identifier, as well as an analysis of the development of the graphical user interface (GUI). Chapter Five gives suggestions about extensions that can be added to Species Identifier and concludes this thesis.

CHAPTER TWO BACKGROUND INFORMATION

Species Identifier uses certainty factors to determine the species represented by a set of fossils. Using an interview format, the system asks the user a series of questions to provide information on the characteristics of the fossils. Some questions gather information common to all the hominid species and are asked in all interviews, while other questions are only asked if the certainty score of a specific species is above a set level. These questions are asked to strengthen or weaken the argument for that species. This chapter provides a brief look at background information on human evolution, certainty factors, some example rule-based systems, and previously developed applications in the field of anthropology.

Background Information on Human Evolution

The practice of classifying species of hominid fossils can be a difficult and sometimes controversial task. In many instances the specimens uncovered in excavations are scarce and fragmentary. Based on just part of a mandible and some teeth, biological anthropologist Dr. Robert Broom declared a new species of hominid called *Parathropus crassidens*, which most researchers now agree is actually *Australopithecus robustus* [GRO04].

Skeletal features used to distinguish species in the fossil record include dental patterns such as the shapes and sizes of the molars or canines, the shape of the tooth rows, and the thickness of the tooth enamel. Researchers also look at cranial features such as the presence and shape of brow ridges, the estimated brain size, and the presence of

cresting along various parts of the skull. Post cranial features used to help classify a specimen include stature, placement of the thumb, and the presence of a precision grip.

Species Identifier covers all commonly recognized species of hominids. The twelve identifiable species are *Ardipithecus ramidus*, *Australopithecus anamensis*, *A. afarensis*, *A. africanus*, *A. aethiopicus*, *A. boisei*, *A. robustus*, *Homo habilis*, *H. erectus*, *H. heidelbergensis*, *H. neandertalensis*, and *H. sapiens*.

There is much debate in the field of biological anthropology about species classification. Many of the species covered by this system are commonly split into two or more species by anthropologists to better indicate the sometimes wide variation of characteristics seen within a species. For example, *H. erectus* is sometimes split into two species: *H. erectus* and *H. ergaster*, to better indicate the wide variation between fossils of *H. erectus* found in Africa and in Asia.

Other species are sometimes classified under a completely different genus. For example, the robust species of Australopithecines are occasionally classified under the genus *Paranthropus* since their characteristics diverge drastically from other species of Australopithecines. Even though many of these classifications are widely accepted, this system only uses the most conservative classifications of hominids.

The species of hominids represented in the fossil record date as far back as 4.4 million years ago although researchers believe the origin of hominids likely occurred almost two million years earlier [STE96]. Figure 2-1 shows a timeline of the hominids, which details, in general, the morphological relationships between each species. As the figure indicates, many gaps exist in the current evolutionary tree. The diagram begins 4.4 million years ago with genus *Ardipithecus*, which is the oldest hominid to be discovered.

Ardipithecus is considered a sister genus to *Australopithecus* that may have evolved in some other direction. *Australopithecus anamensis* appears in the fossil record as early as 4.2 million years ago and is followed by *A. afarensis* 3.9 million years ago [STE96].

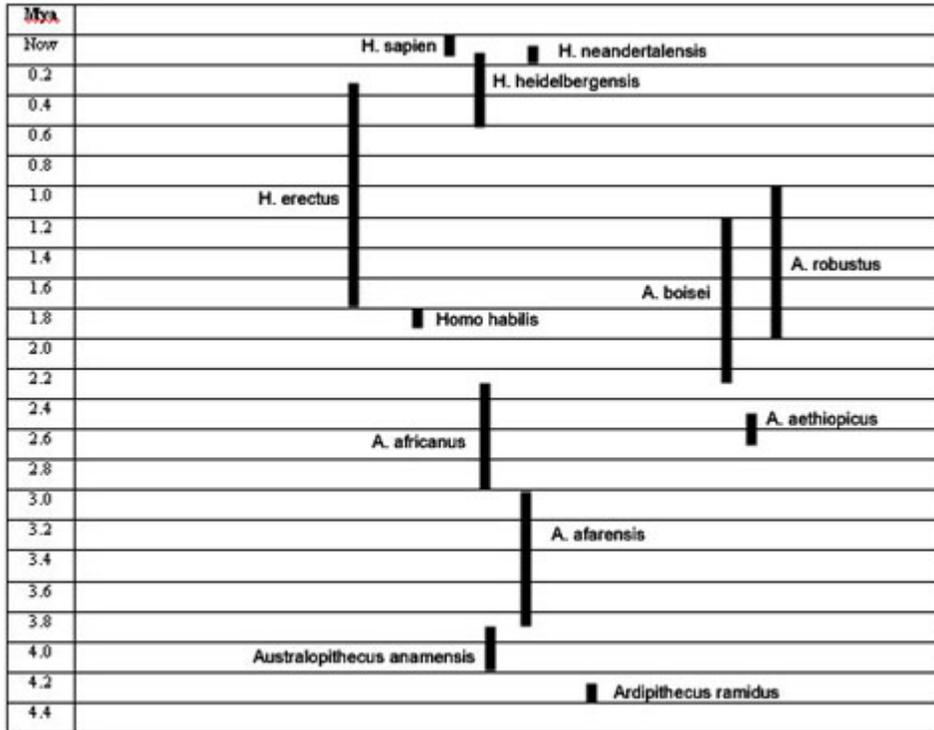


Figure 2-1. A chronology of hominids recognized by Species Identifier. Adapted from [INS04]

All species of genus *Australopithecus* and *H. habilis* are found only in Africa. Fossils of other species of genus *Homo* are found all over the Old World, while only *H. sapien* fossils have been located in the American continents.

Table 2-1 represents the main characteristics of genus *Ardipithecus* and genus *Australopithecus* identifiable by the system. Species of genus *Australopithecus* and genus *Ardipithecus* exhibit apelike features such as smaller brains, massive jaws, and cresting along the skull, which was used to attach large chewing muscles. They also possess modern features such as bipedality and sometimes a precision grip, which might allow for tool use. While no direct evidence of tool use has been directly associated with genus

Australopithecus, evidence of tool manufacture does exist as far back as 2.6 million years ago. This date indicates Australopithecines may have created and used tools [STE96].

Table 2-1. List of genus *Australopithecus* and *Ardipithecus* characteristics*

Name	Age (million years ago)	Location	Some Features
<i>Ardipithecus ramidus</i>	4.4	East Africa	Small incisors relative to molars, large canines, thin tooth enamel relative to <i>A. afarensis</i> , single cusp on lower premolar, post canine megadontia
<i>Australopithecus anamensis</i>	4.2 – 3.9	East Africa	Large and elongated canines, thick tooth enamel, body size ranges from ~3'5" to 4'11" depending on gender, straight tibia shaft, bone thickening on proximal and distal ends of tibia
<i>A. afarensis</i>	3.9 – 3.0	East Africa	Brain size 380 – 500 cm ³ , projecting face, body size ranges from ~3'5" to 4'11", lower premolar has two cusps, diastema present, some postorbital constriction, relatively long arms, supraorbital torus
<i>A. aethiopicus</i>	2.7 – 2.5	East Africa	Brain size ~410 cm ³ , sagittal crest on top of skull, wide and flat face, extreme facial prognathism, marked postorbital constriction, small incisors and canines, robust jaw
<i>A. africanus</i>	3.0 – 2.3	South Africa	Brain size 435 – 530 cm ³ , some postorbital constriction, small canines with no diastema, projecting face, lower premolar has two cusps, broad nasal aperture, relatively long arms, short and wide iliac blade
<i>A. robustus</i>	2.0 – 1.0	South Africa	Brain size 530 cm ³ , body size from 3'7" to 4'4", sagittal crest, robust chewing muscles, marked postorbital constriction, wide and flat face, decreased facial prognathism, robust jaw, large premolar and molars, lower premolar has two cusps
<i>A. boisei</i>	2.3 – 1.2	East Africa	Brain size 410 – 530 cm ³ , body size from 4'1" – 4'6", robust chewing muscles, marked postorbital constriction, sagittal crest on mid-brain case, decreased facial prognathism, large premolar and molars, small incisors and canines

*Table adapted from [INS04].

Genus *Homo* began appearing around two million years ago. *Homo habilis*, whose name means "handy man" in reference to their use of tools, is the first acknowledged

species of genus *Homo*. A clear trend towards larger brains and a reduction of apelike traits, such as brow ridges and facial prognathism, is seen as genus *Homo* matures into anatomically modern *Homo sapiens*. Species Identifier recognizes five species listed in Table 2-2 from this genus.

Table 2-2. List of genus *Homo* characteristics*

<i>Homo habilis</i>	1.9 – 1.8	East Africa	Brain size 500 – 800 cm ³ , body size from 3'11" – 5'2", thin cranial bones, slight facial prognathism, reduced postorbital constriction, light jaw, small teeth, molars longer than wider, hand similar to modern humans, used stone tools
<i>H. erectus</i>	1.8 – 0.3	Africa, Asia, Europe	Brain size 750 – 1250 cm ³ , thick cranial bones, robust skull, slight facial prognathism, reduced postorbital constriction, small teeth, robust jaw, modern hand, long legs, barrel shaped chest, double arched supraorbital torus
<i>H. neandertalensis</i>	0.15 - 0.03	Europe, Asia	Brain size 1300 – 1750 cm ³ , Body size 5'1" – 5'5", thin cranial bones, rounded skull vault, occipital bun, continuous and rounded brow ridge, no postorbital constriction, incisors somewhat large, wide nasal aperture, muscular and robust body, long and low brain case, lack canine fossa
<i>H. heidelbergensis</i>	0.6 – 0.1	Africa, Europe, Asia	Brain size 1100 – 1400 cm ³ , body size 5'2" – 5'9", thick cranial bones, higher skull than <i>H. erectus</i> , slight to no postorbital constriction, reduced cranial robusticity, robust mandible, large and broad face, thick brow ridge, modern body proportions, lack canine fossa
<i>H. sapiens</i>	0.1 – present	Worldwide	Brain size 1000 – 1700 cm ³ , lightly built skull, rounded skull vault, slight facial prognathism, no pronounced brow ridges, small teeth, low braincase, smaller nasal aperture, presence of canine fossa

*Table adapted from [INS04].

Certainty Factors

Species Identifier relies on certainty factors to develop conclusions about the species of a set of fossils. Certainty factors were developed to address problems inherent within the Bayesian method [GON93]. Among these inadequacies, which are present in

the problem of classifying human fossils, is the lack of large quantities of data. For example, there are no firm statistics that can state that 98% of *A. robustus* exhibit a sagittal crest on top of their skull. It is one feature among many that identifies them, but hardly enough fossil evidence exists to produce accurate statistics such as this.

Certainty factors determine the solution to the problem by weighing evidence then, with knowledge represented in an *if-then* rule format, provide a mechanism for the system to explain its logic to the user. They measure the degree of confidence in a hypothesis from -1.0 to 1.0. Negative numbers represent the measure of disbelief in a hypothesis and positive numbers represent the measure of belief in a hypothesis [ART04]. “Confidence measures correspond to the informal evaluations that human experts attach to their conclusions, such as ‘it is probably true,’ ‘it is almost certainly true’ or ‘it is highly unlikely’” [LUG02, pp. 320].

Certainty factors are useful in situations where there is a lack of large quantities of data, a need for the system to explain its reasoning, and a need to balance positive and negative information. The system could identify for instance, “because this fossil is 100,000 year old only three species of genus *Homo* are known to have existed in this time period: *H. sapiens*, *H. neandertalensis*, and *H. heidelbergensis*. The fossil is from France where three species of genus *Homo* are known to have existed in that area and at that time period: *H. sapiens*, *H. neandertalensis*, and *H. heidelbergensis*. It has a brain size of 1500 cm³ which is larger than *H. heidelbergensis* but matches the brain sizes known to have been possessed by *H. sapiens* and *H. neandertalensis*. The fossil exhibits robust bones and was heavily muscled which is a feature of *H. neandertalensis*, as well as an occipital bun on the back of the skull which is a feature known only to be exhibited by *H.*

neandertalensis. The skull also features a continuous and rounded brow ridge, which is a feature of *H. neandertalensis* and not *H. sapiens* who have no pronounced brow ridge. The skull shows no postorbital constriction, which is a feature of *H. neandertalensis* and *H. sapiens*. As a result, it is with 99.7% confidence that this fossil set is classified as *H. neandertalensis*.”

The rules in a system using certainty factors have the following format:

IF EVIDENCE
 THEN HYPOTHESIS (CF)

Where EVIDENCE is a characteristic of the fossil that is observed by the user that leads Species Identifier to believe the resulting HYPOTHESIS with a confidence indicated by the value of CF [GON93].

As Species Identifier interviews the user, many rules may execute that indicate the same hypothesis. These rules' certainty factors are combined for each species. The value of the system's certainty about each species will approach -1 or 1 depending on whether the fossil evidence strengthens or weakens the case for each species.

Certainty factors provide a mechanism for combining the results of multiple rules deriving the same hypothesis. This mechanism allows the same conclusion derived by numerous rules with low certainties to be combined to form a strong conclusion.

Certainty factors are combined using the following equations [GON93]:

- $CF_{\text{combined}}(CF_1, CF_2) = CF_1 + CF_2 * (1 - CF_1)$, if CF_1 and $CF_2 > 0$
- $CF_{\text{combined}}(CF_1, CF_2) = -CF_{\text{combined}}(-CF_1, -CF_2)$, if CF_1 and $CF_2 < 0$
- $CF_{\text{combined}}(CF_1, CF_2) = (CF_1 + CF_2)/(1 - \text{MIN}(|CF_1|, |CF_2|))$, otherwise.

The first equation is used when the CF of the current species and the CF of the rule just executed are both positive. The second equation is used when the CF of both are

negative. The third equation is used if either the CF of the current species or the CF of the newly executed rule is negative and the other is positive.

Previous Rules-based Systems

Mycin

Mycin was an early expert system developed at Stanford, which used certainty factors in its implementation. It provided an expert diagnosis of the blood infections bacteremia and meningitis. The Mycin system simulated having an expert consultant specializing in blood infections helping to diagnose a patient [LAN04].

The Bayesian approach to uncertainty was found to be inadequate for use in Mycin since its developers needed a way for the system to explain how Mycin came to its particular conclusion, which the Bayesian model does not provide. The way in which physicians gather information and come to a conclusion is also different from an implementation of the Bayesian model [GON93]. For these reasons certainty factors were developed and successfully implemented in Mycin.

XCON

Another rule-based expert system is XCON, which was developed in the early 1980's by Digital Equipment Corporation. XCON helped customers configure VAX computer systems by removing or adding components to make certain the configuration was correct.

XCON replaced technical editors who reportedly were only accurate with their configurations 65% of the time (or inaccurate 35% of the time) [INF04]. XCON on the other hand, only produced systems where the configuration needed to be reevaluated in 10% of the cases [AGO04].

The system contained 6,000 rules for over 20,000 different components. By the late 1980's, the number of rules had grown to over 18,000, which needed a large staff of programmers to maintain. In spite of this, XCON was a commercial success for Digital Equipment Corporation since it produced annual revenue of over \$40 million [INF04].

Expert Systems in Anthropology

Expert systems are not common to the field of anthropology. One sub-discipline of anthropology having some examples of expert systems is forensics. "The forensic anthropologist can be of help in resolving problems of identity and assessing trauma that has occurred. However, the accuracy and speed of one's conclusions are dictated to a large degree by the amount and quality of evidence at hand" [RHI99, pp. 47].

Forensic anthropologists are typically given a set of skeletal remains and attempt to determine factors such as the age, race, sex, and stature of the individual. The results of the analysis are compared with missing person files to see if there are any close matches.

Anthropologist

Anthropologist is a forensics expert system designed by Meister Dmitry. The user of this system enters forensic evidence into forms from which the system can determine the race, sex, and age of a set of skeletal remains.

UPPER JAW RIGHT SIDE		
1	Corona is grey	1.2
	Caries	1.2
2	Split off	0.44

Figure 2-2. The form used by Anthropologist to estimate age based on dentition

Anthropologist has been tested at the Perm Regional Bureau of Forensic Examination [DMI04]. On the form shown in Figure 2-2, users enter dental information while the system updates its estimation of the age of the skeletal remains in real time. Anthropologist includes many different subprograms to use in a forensic analysis. These include programs that can determine the gender of a vertebrae, skull, or scapula by various measurements or observations that are to be made by the user in a forensics laboratory. Figure 2-3 is a screenshot of the form used to determine the gender of a skull. The possible conclusions it can reach are reliable male/female, probable male/female, or indefinite.

The screenshot shows a window titled "Reporter - Parameters of the diagnostic sizes of a skull at a determinatic". The window contains a table with 12 rows of measurements and their corresponding input fields. The table is organized into columns for "Reliable Man 0", "Reliable Woman 0", "Indefinite 0", "Probable Man 0", and "Probable Woman 0".

	Reliable Man 0	Reliable Woman 0	Indefinite 0	Probable Man 0	Probable Woman 0
1. Longitudinal diameter					
2. Cross diameter					
3. High-altitude diameter					
4. Length of the basis of a skull					
5. The least width of a forehead					
6. Width of the basis of a skull					
7. Width of a nape					
8. Mammiform width					
9. A circle of a skull					
10. Sagittal chord					
11. A frontal chord					
12. A parietal chord					

Figure 2-3. This form uses cranial measurements to determine the gender of skeletal remains

The Anthropologist software has the option to use certain forensic experts' opinions. The software comes with the evaluation techniques of John Samson with the option of adding your own evaluations techniques or the techniques of another expert.

Sex	Male
Enclosure N	
Statement N	
Expert	Samson John

Figure 2-4. Selecting your expert in Anthropologist

ForDisc

ForDisc is another application used in forensic anthropology to estimate factors such as the sex and race of human remains. ForDisc uses the University of Tennessee's Forensic Databank, which contains data on over 1,400 individuals. Users of this application input cranial measurements, which are then compared to the measurements in the databank. The system uses discriminant functions to estimate the sex and race of the human remains in question.

In one study, the accuracy of ForDisc was tested using the Hamann-Todd Collection of human remains. ForDisc was used to determine the race and gender of 100 individuals selected at random and was correct in its classification of 81% of the individuals [SAN04].

Conclusion

This chapter discussed some background information on human evolution as well as the problem of classifying hominid fossils. It also described certainty factors, which are used by Species Identifier to model the confidence with which a biological anthropologist would classify a specimen. This chapter also explored some previous examples of rules-based systems in general, as well as expert systems developed for forensic anthropologists. The next chapter is a walkthrough of Species Identifier.

CHAPTER THREE WALK THROUGH OF SPECIES IDENTIFIER

This chapter provides a walk through of Species Identifier. This application identifies twelve species of hominids asking up to 39 different questions depending on what species the system thinks is the most likely solution. The example walkthrough presented in this chapter uses characteristics specific to the species *Homo neandertalensis*. Figure 3-1 shows the system at startup. It first asks about the estimated age of the fossils. An age of 100,000 years has been entered with a certainty factor of 20%.

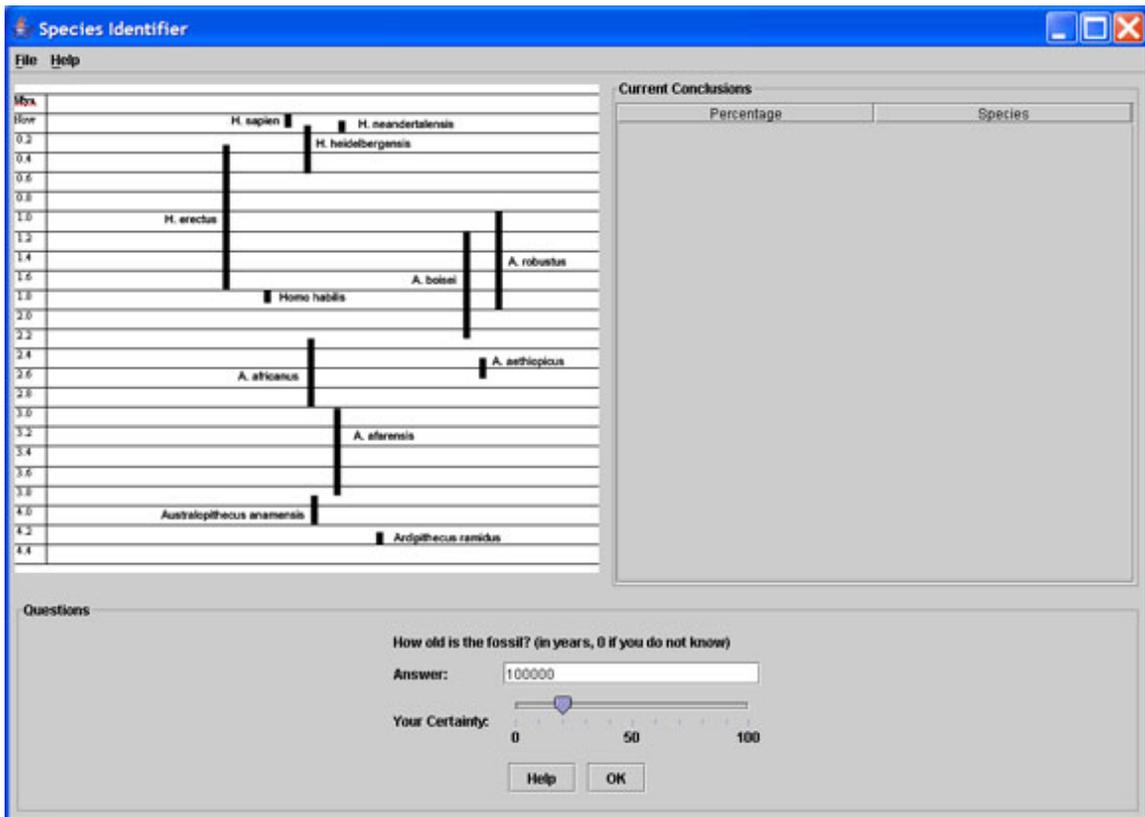


Figure 3-1. The first question asked by Species Identifier.

Description of the Graphical User Interface

The Graphical User Interface (GUI) consists of three main panels: the image panel, the current conclusions table, and the questions panel. Each panel is discussed in detail below.

Image Panel

The image panel is in the top left of the application and displays a picture that is related to the current question being asked. For example, when Species Identifier asks, “How many cusps are on the anterior lower premolar,” the accompanying picture acts as a visual guide for the user. Figure 3-2 is a screenshot of the image that appears with this question, identifying to the user which tooth they should be analyzing.

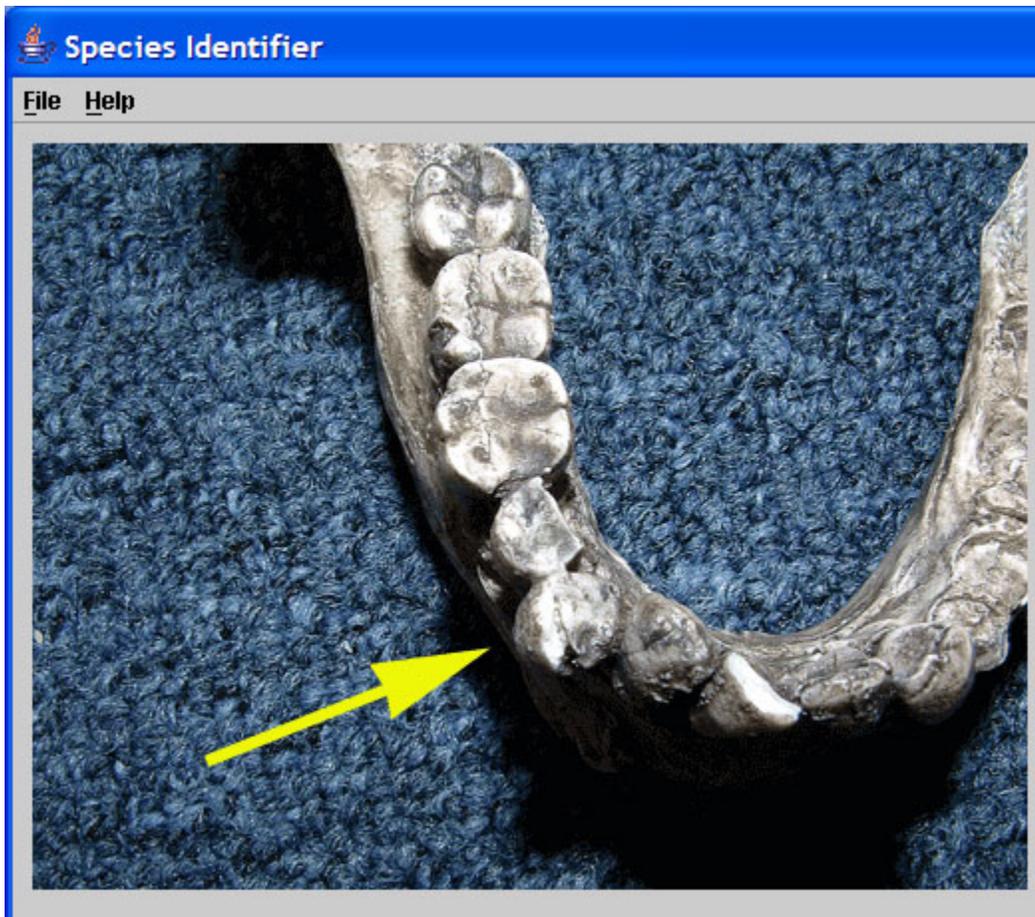


Figure 3-2. The image panel of Species Identifier.

Current Conclusions Table

On the top right of the GUI is the current conclusions table. This table represents the current confidence score of each species still being considered. A species is still being considered as a conclusion if its confidence score is above negative 100%. The species are listed in descending order starting with the species the system currently considers the most likely conclusion. Figure 3-3 shows an example of the current conclusions table after the user has been asked about the age of the fossils. Based on the answer provided to this question, the system currently believes—with a confidence of 12%—that the fossils represent either *H. heidelbergensis* or *H. neandertalensis*.

Current Conclusions	
Percentage	Species
12.0	Homo neandertalensis
12.0	Homo heidelbergensis
8.0	Homo sapien
2.0	Homo erectus
-19.8	Australopithecus robustus

Figure 3-3. The current conclusions table of Species Identifier.

Questions Panel

On the lower portion of the GUI is the questions panel. Figure 3-4 shows the questions panel in its entirety. This panel displays the text of the current question being asked, a location for the answer to the question, a slider to indicate the user's certainty in their answer, a help button, and an OK button to submit the answer.

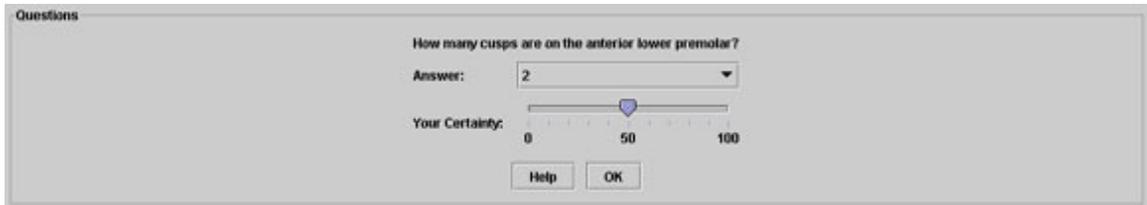


Figure 3-4. The questions panel of Species Identifier.

There are two different mechanisms provided by the GUI for the user to enter answers. The first type is an input box. The input box is provided for answers that require numerical input of more than four possible numbers. The second type is a pull-down menu. The pull-down menu allows the user to select from a set of valid answers. Both types typically allow an answer of “Don’t Know.” If the user answers with “Don’t Know,” the confidence scores of each species being considered remain unchanged.

A user may not be certain about a particular answer. In this case, the slider allows the user to indicate how certain they are in their answer to the question. Figure 3-5 shows an example of the slider. The user can select a certainty factor from 0-100 with zero indicating they are absolutely uncertain about their answer and 100 indicating they are absolutely confident in the accuracy of their answer. The application divides this number by 100 to convert it to a value between 0.0 and 1.0, which can be factored into an existing certainty factor score.

Some questions do not require a certainty factor to be indicated. For example, a certainty slider is not present when the application asks if portions of the skull are

present. In these cases, the slider is grayed-out and not selectable by the user to avoid any confusion.

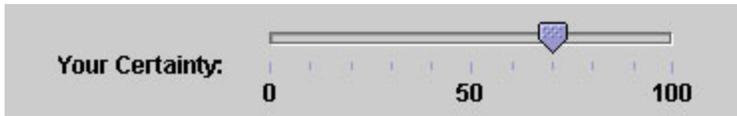


Figure 3-5. The slider feature allows users to indicate their confidence in an answer.

The Help button is provided to give the user a more detailed description of what the question is asking. Figure 3-6 shows the window that is created when the Help button is pressed. The help window typically provides definitions for some of the vocabulary in the question, a description of the image in the image panel, and some background information on the question being asked.

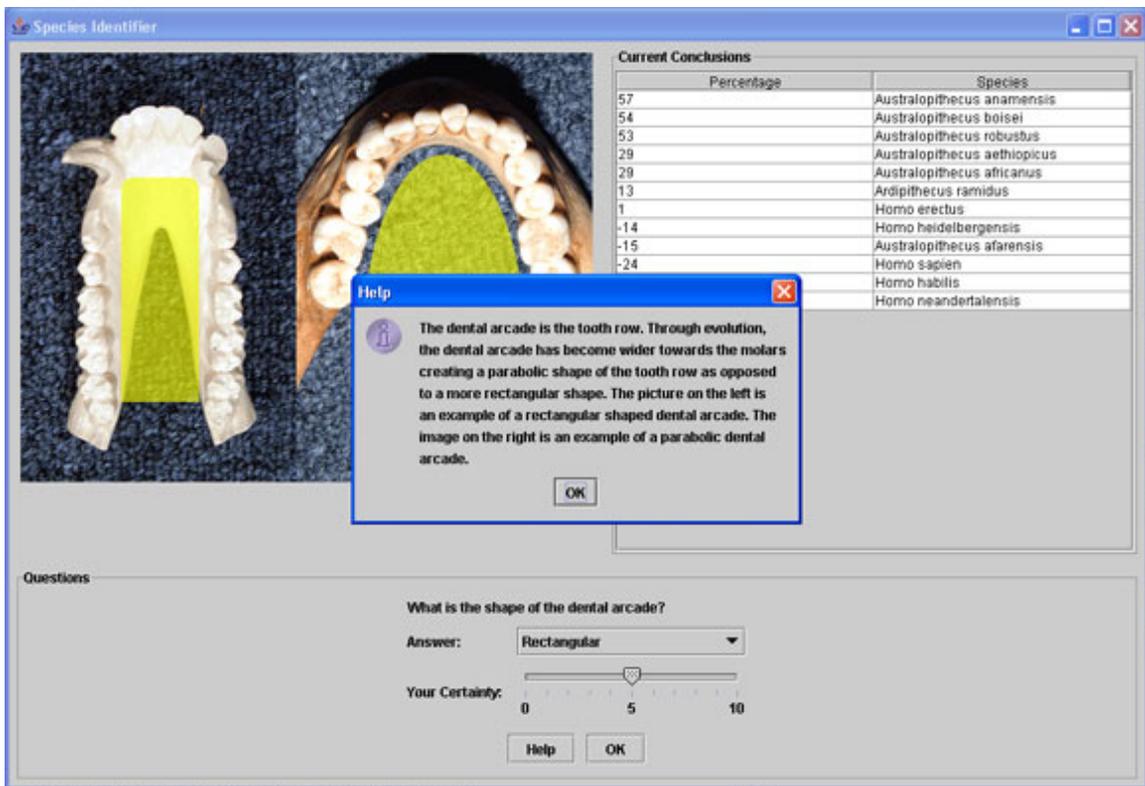


Figure 3-6. Pressing the Help button creates a window providing more information on the question being asked.

A Walk-Through of Species Identifier's Execution

A hypothetical example is used in this chapter to demonstrate Species Identifier's execution. The user is examining the skull of *H. neandertalensis* assembled from fragments found in Europe. This skull is in relatively good shape with most of the cranium still intact and the mandible present. Figure 3-7 is a collection of several images of an *H. neandertalensis* skull.



Figure 3-7. An example of an *H. neandertalensis* skull.

Age and Location Questions

The first question that Species Identifier asks is the age of the fossil. In this case, the user uses relative dating. Relative dating uses locally associated faunal or floral materials to date fossils to obtain an idea of the age of the fossil remains [CON97]. Using this dating technique, the estimated age is approximately 100,000 years old. In Figure 3-1, the estimated age is entered into the system with a confidence of 20% since relative dating was used.

The next question the system asks concerns the location of the fossil. Figure 3-8 shows the screen where the user enters the location from which this fossil was excavated. Since this fossil was found in Europe, the item on the pull-down menu indicating “Europe” is selected with a certainty of 100%.

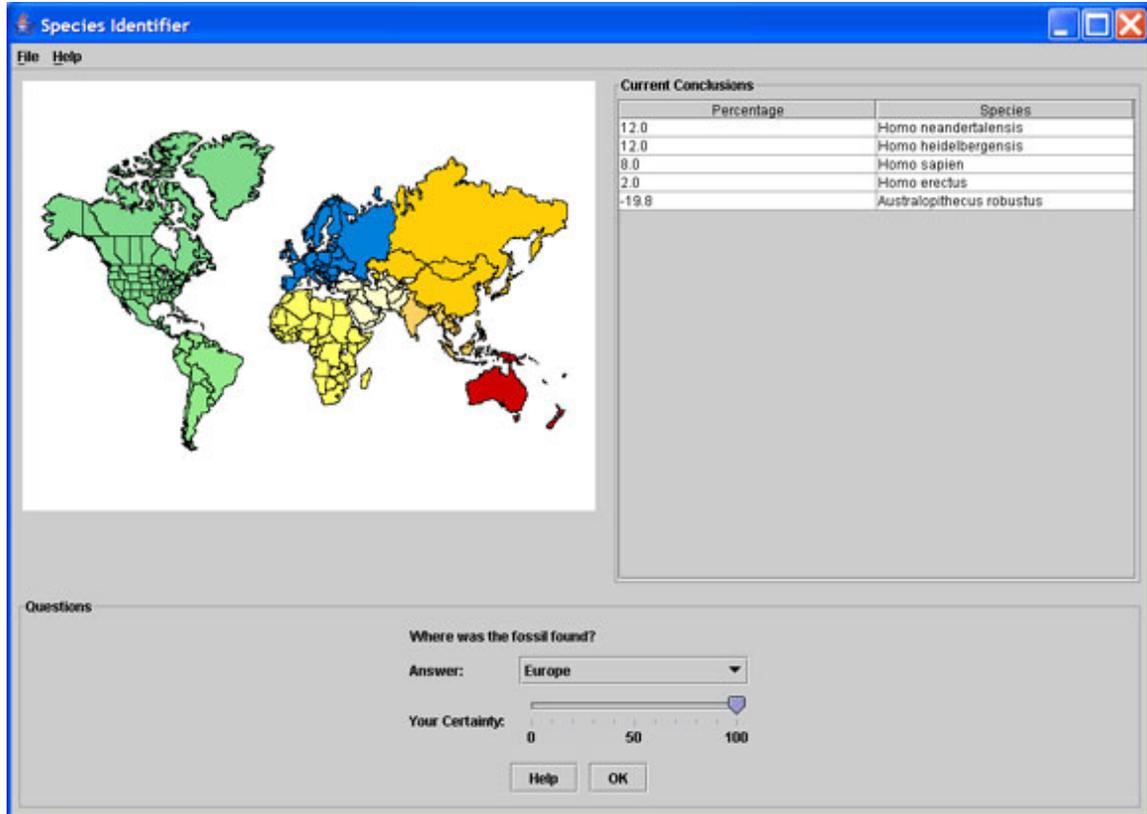


Figure 3-8. The user indicates that the specimen was found in Europe with 100% certainty.

Questions about Dentition

At this point, based on the age and location questions, Species Identifier believes the specimen could represent *H. sapien*, *H. erectus*, *H. heidelbergensis*, or *H. neandertalensis*, with almost the same certainty for each species. Following the question about location, Species Identifier asks if portions of the skull present. If portions are available, the system asks if there are dental remains available for examination.

The following series of dental questions and cranial questions attempt to provide more differentiation between the possible conclusions. Figure 3-9 shows Species Identifier inquiring about the number of cusps on the anterior lower premolar. This question is used to determine if the species is hominid or some other type of primate. The user answers there are two cusps with a certainty of 50%.

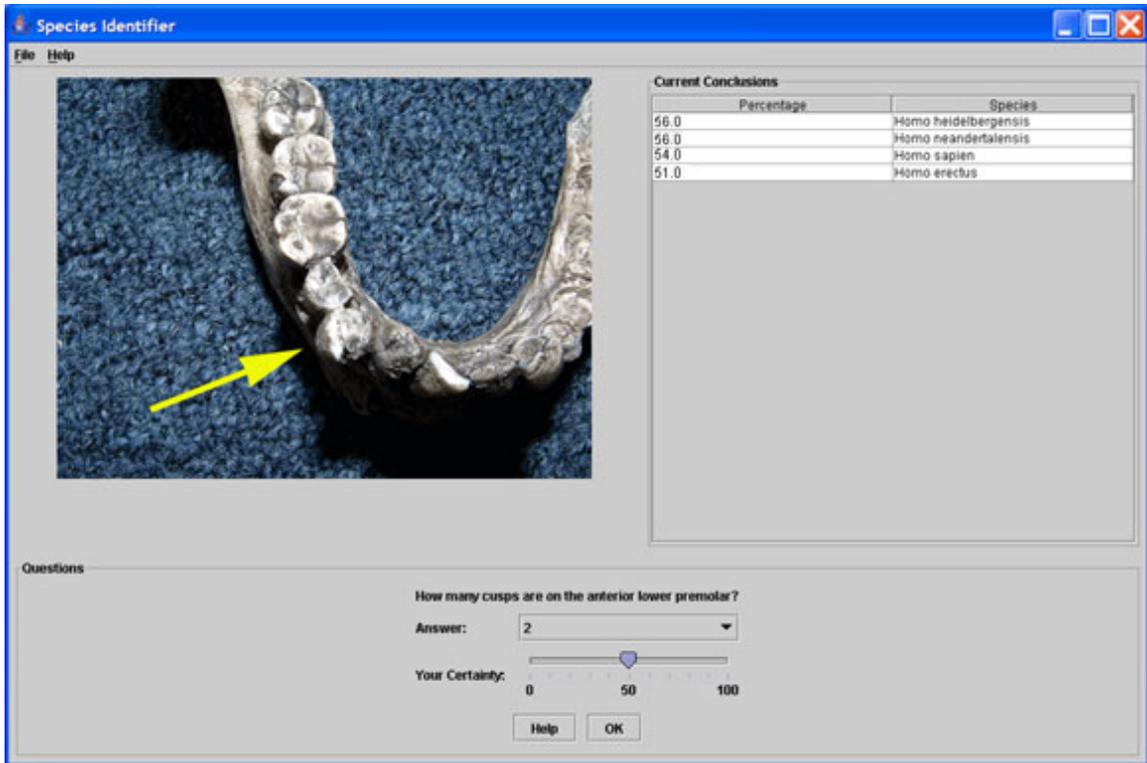


Figure 3-9. Species Identifier asks the user about the number of cusps on the lower front premolar.

Next, Species Identifier asks the user about the size of the canines and whether they are larger or similar in size to other teeth present in the dentition. This question is asked to determine if the species could possibly be an early hominid or other primate because more modern hominids have canines similar in size to their other teeth. Species Identifier also asks about the size of the molars. Robust species of genus *Australopithecus* possessed very large, flat molars, possibly used to grind tough, fibrous vegetation [STE96]. Therefore, this question helps determine if the specimen could possibly be in this classification.

Figure 3-10 shows Species Identifier asking about the size of the molars. An answer of “Small” is given, with a certainty of 70%. If this question is answered “Large,” Species Identifier will then ask if the fossil exhibits postcanine megadontia, where the

molars and premolars are much larger than the incisors, to further differentiate its conclusion.

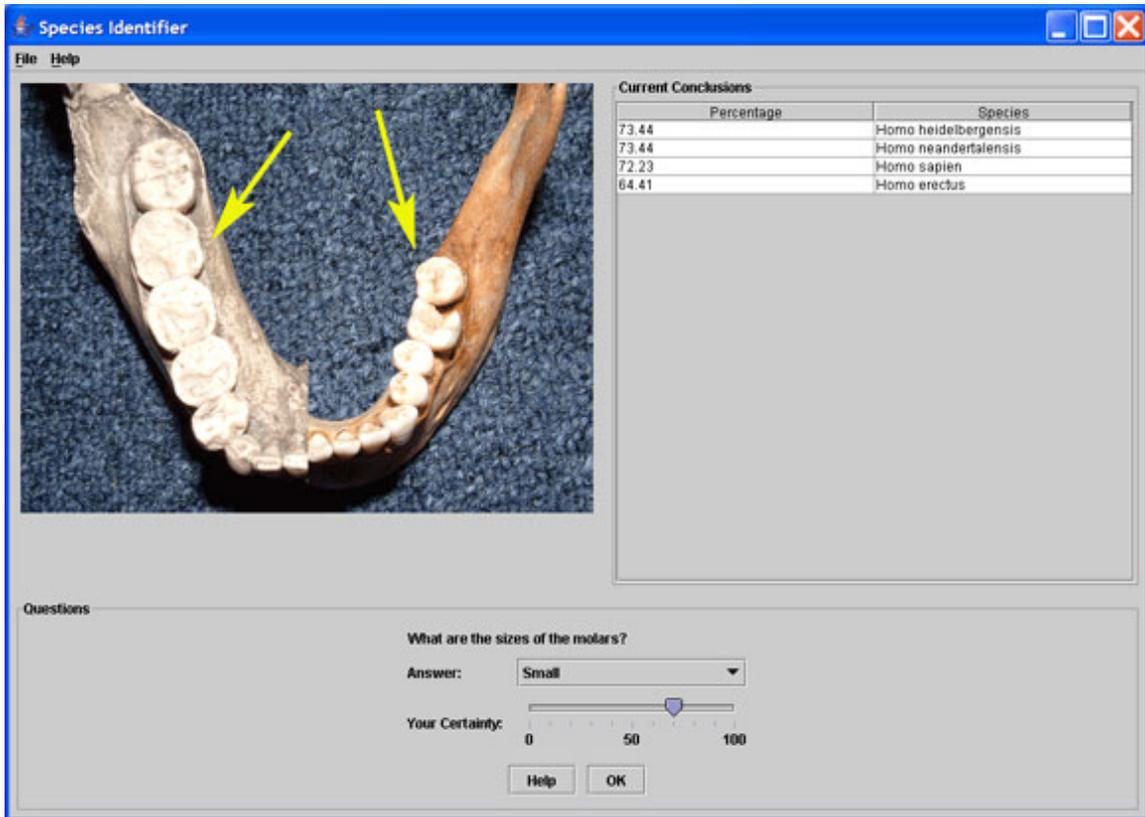


Figure 3-10. The application asks if the size of the molars is large or small.

Species Identifier then asks the user about the thickness of the tooth enamel. Tooth enamel is the hard, mineralized, outer layer of the tooth [STE96]. A biological anthropologist would use a scanning electron microscope to determine the thickness of tooth enamel [STE96]. In this example, we do not have access to a scanning electron microscope, so an answer of “Don’t Know” is entered. Answers to questions where the user does not know the answer are not factored into the conclusion.

Next, Species Identifier examines a series of questions triggered by the current scores of each species. This strengthens or weakens the score of a specific species it

suspects. In the example, only one triggered question is asked since most of the triggered questions in the dentition flowchart are associated with older species of hominids.

Because the score of *H. erectus* is currently above 70%, the system asks about the presence of shovel-shaped incisors. Shovel-shaped incisors are a trait common to *H. erectus*. “Shovel-shaped incisors are incisors that have a scooped-out shape on the tongue side of the tooth” [STE96, pp. 491]. Figure 3-11 shows the system asking the question about shovel-shaped incisors. This specimen does not seem to possess such a trait, so an answer of “No” is indicated with a certainty of 80%. With the triggered questions completed, Species Identifier has concluded the dentition section of the interview.

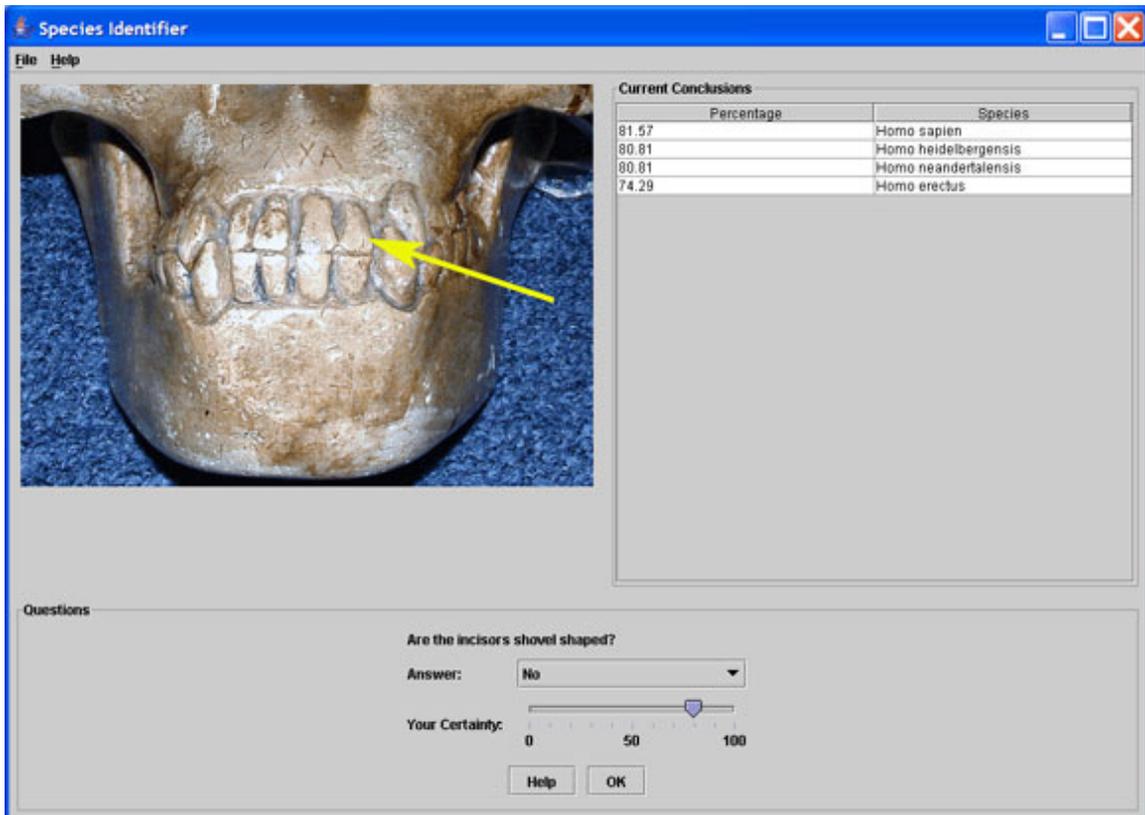


Figure 3-11. Species Identifier asks about the presence of shovel-shaped incisors.

Cranial Questions

Because the user previously indicated that portions of the skull are present, the interview now begins on the cranial questions. The last question about shovel-shaped incisors was able to further differentiate the other three species being considered from *H. erectus*, whose certainty score was reduced from 74.29% to 66.17%. The series of cranial questions will provide even more differentiation for these species.

The first of the cranial questions concerns the estimated brain size in cubic centimeters. Determining this measurement usually involves creating a cast of the brain case and measuring its volume [STE96]. In this example, we do not have the ability to create a cast, so an answer of “0” is provided in the input box to indicate that we do not know the answer.

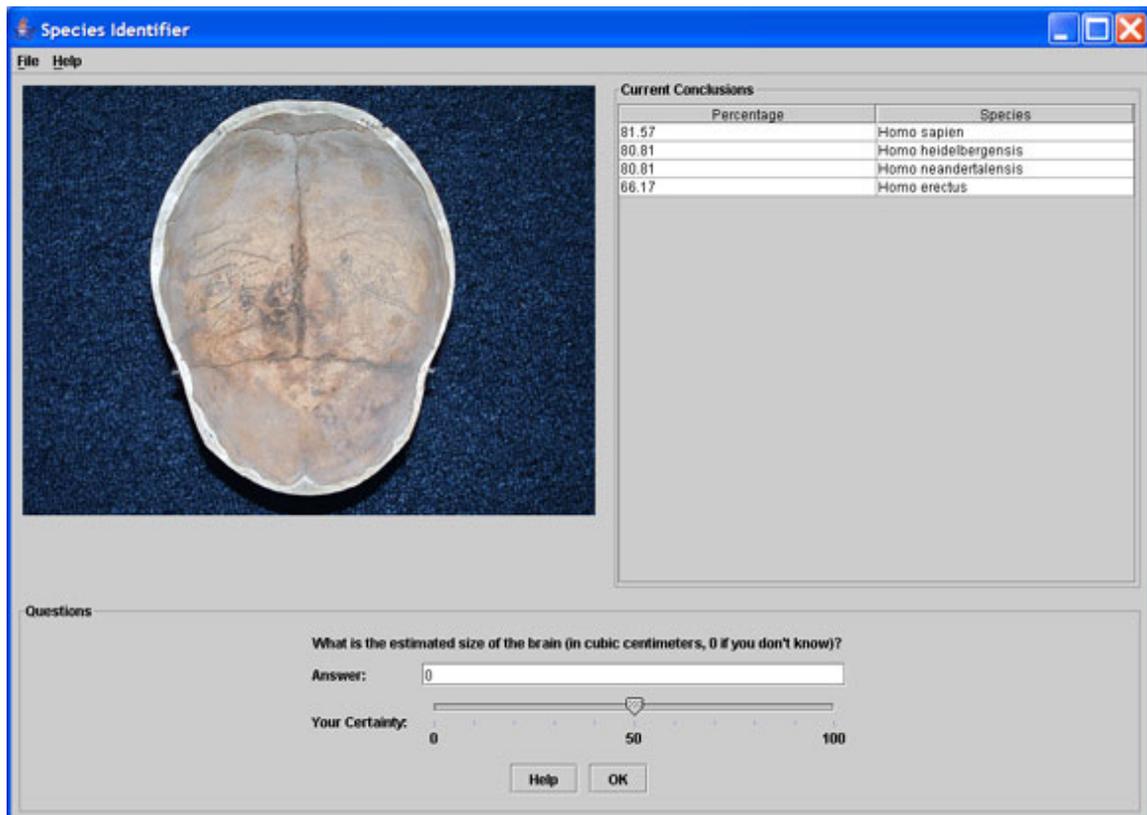


Figure 3-12. Species Identifier asks the user to determine the volume of the brain case.

Since a “Don’t Know” answer was given to the previous question, Species Identifier tries to obtain an idea about the size of the brain by asking the user to estimate the size of the brain in the general terms of “Small,” “Moderate,” etc. Figure 3-13 shows the system asking the more generalized version of the brain-size question. In this example, it is obvious the brain case is quite large when compared to other species of hominids, so an answer of “Large” is indicated with a certainty of 80%.

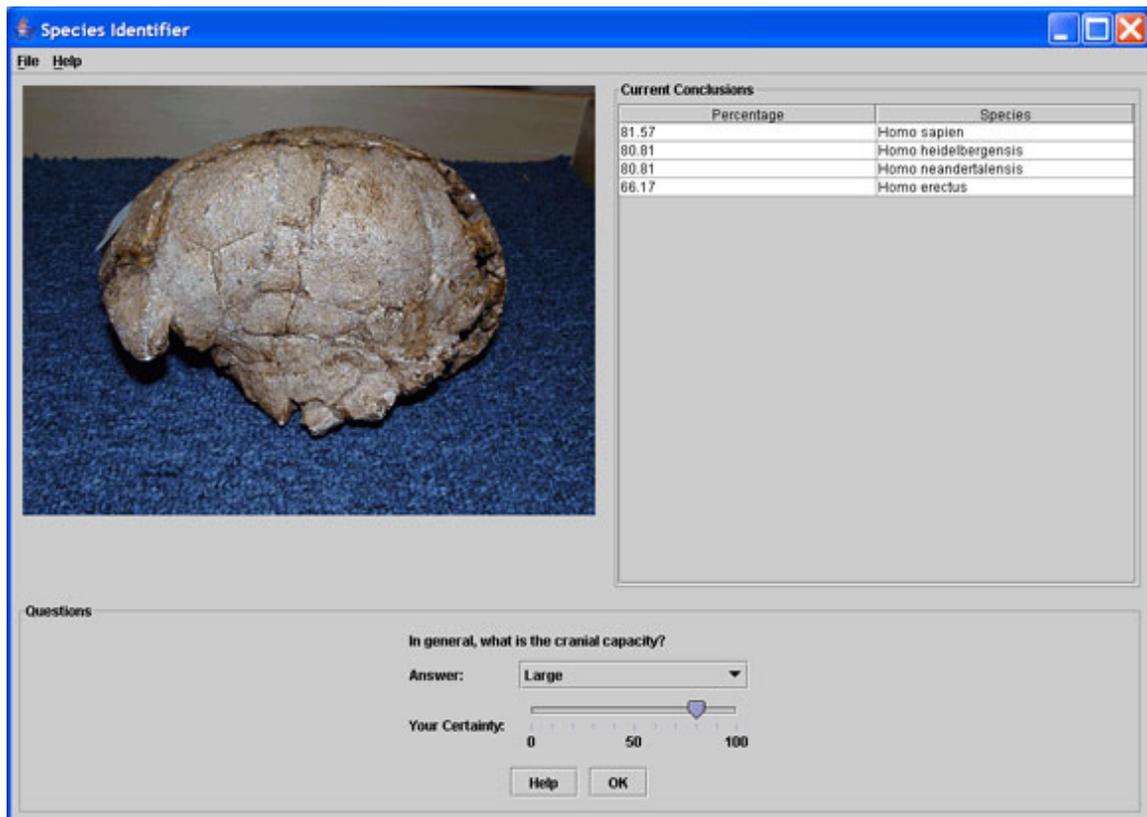


Figure 3-13. Species Identifier asks the user about the cranial capacity of the specimen in general terms.

The next question concerns the amount of facial prognathism exhibited by the skull. Figure 3-14 is a screenshot of the question referring to facial prognathism. Facial prognathism refers to the jutting forward of the facial region [CON97]. The amount of facial protrusion has generally decreased through the evolution of hominids. In this

example, the amount of facial prognathism is noted to be slight; therefore, an answer of “Slight” is indicated with a certainty of 90%.

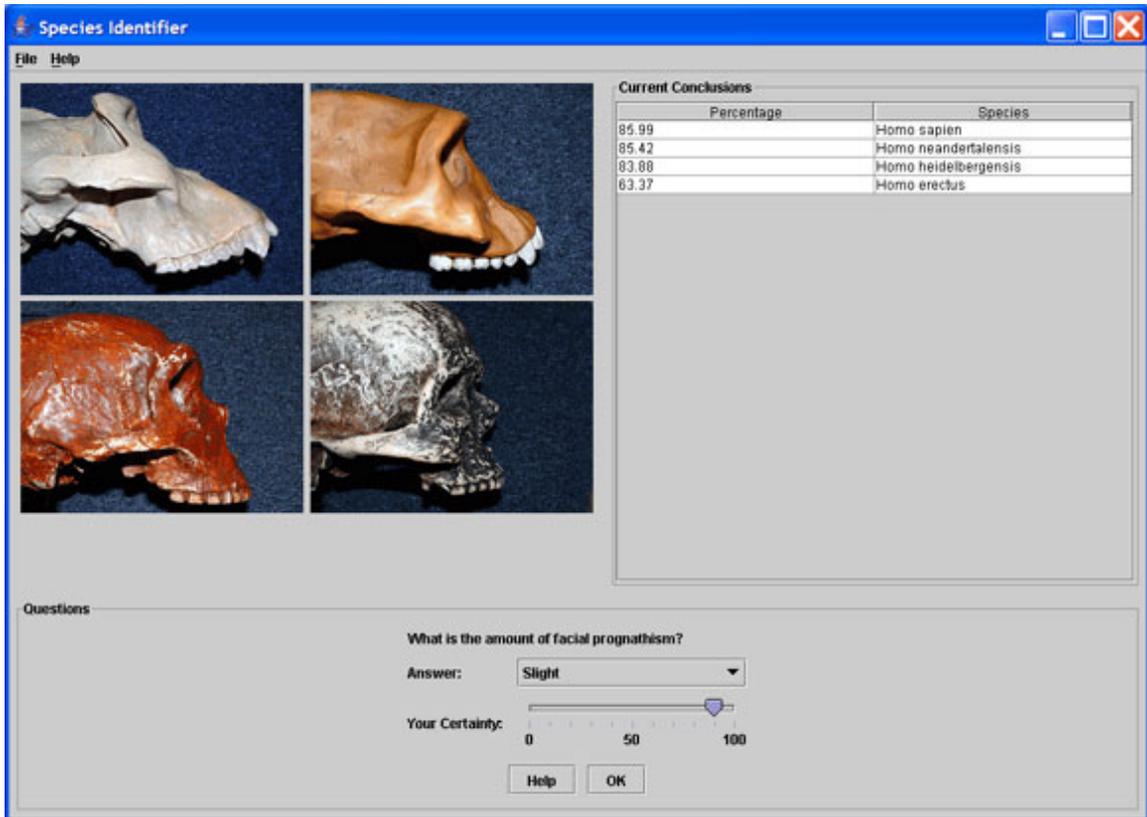


Figure 3-14. The user is asked to determine the degree of facial prognathism.

Species Identifier then asks about the presence of a sagittal crest, which is a ridge of bone along the top of the skull that was used to connect large chewing muscles to the skull. This question is used mainly to differentiate between genus *Homo* and robust species of genus *Australopithecus*. No known species of genus *Homo* possesses this more apelike trait. For this question, the user indicates that this specimen does not seem to possess a sagittal crest with a certainty of 70%.

After this question, Species Identifier asks the user to examine the amount of postorbital constriction. Postorbital constriction is the amount that the front portion of the brain case closes in behind the orbital sockets. This trait is a decent indicator of the size

of the brain since a smaller brain size has a greater amount of postorbital constriction. This skull appears to have no postorbital constriction, so an answer of “None” is given with a certainty of 70%.

In Figure 3-15 the presence of a supraorbital torus is asked. A supraorbital torus is a thick ridge of bone going across the brow of the skull [STE96]. This question is a good general indicator of if the fossil remains are *Homo sapiens*, which do not possess brow ridges. However, the example skull possesses brow ridges. This is indicated with a certainty of 80%. Since the skull does have a brow ridge, this triggers a question about the shape and thickness of the brow ridge. This helps to further differentiate between the species of hominids that do possess a brow ridge. Since the user is not sure of the shape, the user answers “Don’t Know.”

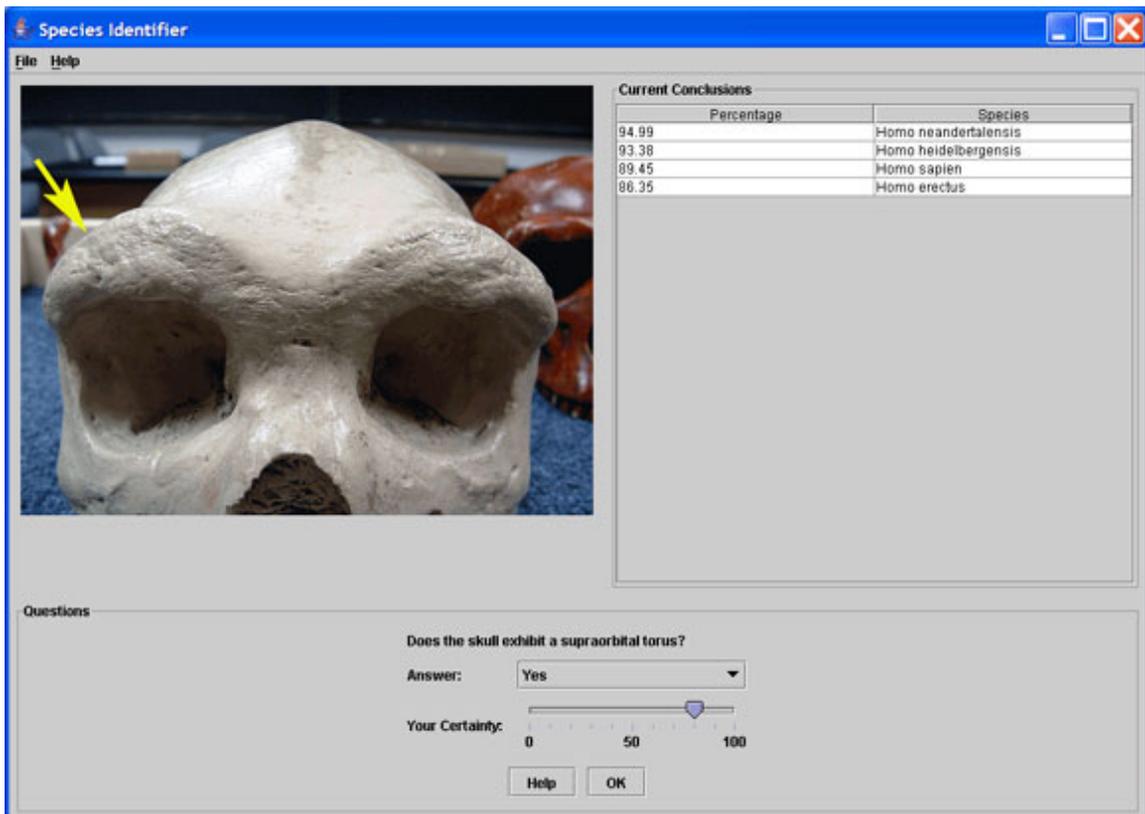


Figure 3-15. The application asks if the skull possesses a brow ridge.

A question about the thickness of the cranial bones is next asked to differentiate between the various species of genus *Homo*. The cranial bones of this skull are thin compared to other species of genus *Homo*, so an answer of “Thin” is entered with a certainty of 80%.

The interview now enters the triggered question phase of execution. Since no species of genus *Australopithecus* is being considered, no questions specific to any species in that genus are activated. *Homo erectus* is one possible conclusion with a current certainty score of 89.05%, so questions unique to this species are activated. The first triggered question asks whether the skull exhibits a supraorbital sulcus.

A supraorbital sulcus is a depression between the brow ridge and the forehead [STE96]. There is no indication of a supraorbital sulcus in the example skull, so an answer of “No” is given with a certainty of 80%. This answer lowers the score of *Homo erectus* from 89.05% to 81.75% as well as lowering the score of *Homo heidelbergensis* from 93.38% to 90.11% since these two species are known to possess this feature. The scores of *Homo neandertalensis* and *Homo sapiens* are unaffected by this answer since this triggered question is not specific to these two species.

The next triggered question specific to *Homo erectus* asks if the skull has sagittal keeling, which is a small ridge of bone running along the top of the skull. This ridge of bone is not nearly as pronounced as the sagittal cresting in some species of genus *Australopithecus*. Figure 3-16 shows a screenshot of the application asking this question. Examining the skull shows no ridge of bone along the top of the skull, therefore an answer of “No” is given. This answer lowers the score of *H. erectus* from 82% to 67% and has no affect on the scores of any other species being considered.

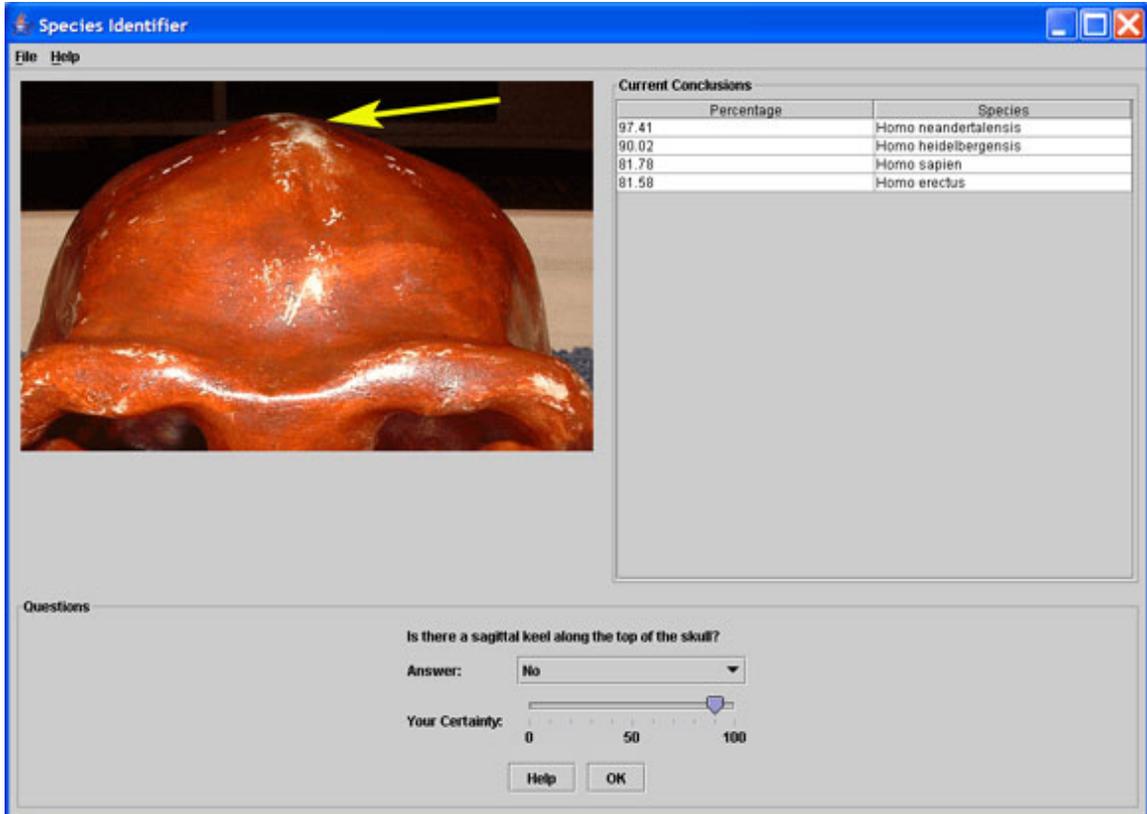


Figure 3-16. Species Identifier asks if there is any sagittal keeling along the top of the skull.

The last question specific to *H. erectus* asks if the skull possesses an occipital torus. An occipital torus is a horizontal ridge of bone on the rear of the brain case [STE96]. The occipital of the skull being investigated seems to exhibit no pronounced ridge; therefore, “No” is entered as the answer with a certainty of 90%. This answer lowers the certainty score of *H. erectus* even further to 44.19%.

Since *H. neandertalensis* is now believed to be the conclusion with a certainty score of 97.41%, questions specific to this species are asked. Figure 3-17 shows a screenshot of the first triggered question for *H. neandertalensis*, which asks for the general width of the nasal aperture. *H. neandertalensis* skulls exhibit a very wide nasal aperture compared to contemporary hominid species. The example skull features a wide nasal aperture; therefore, an answer of “Wide” with a certainty of 90% is entered.

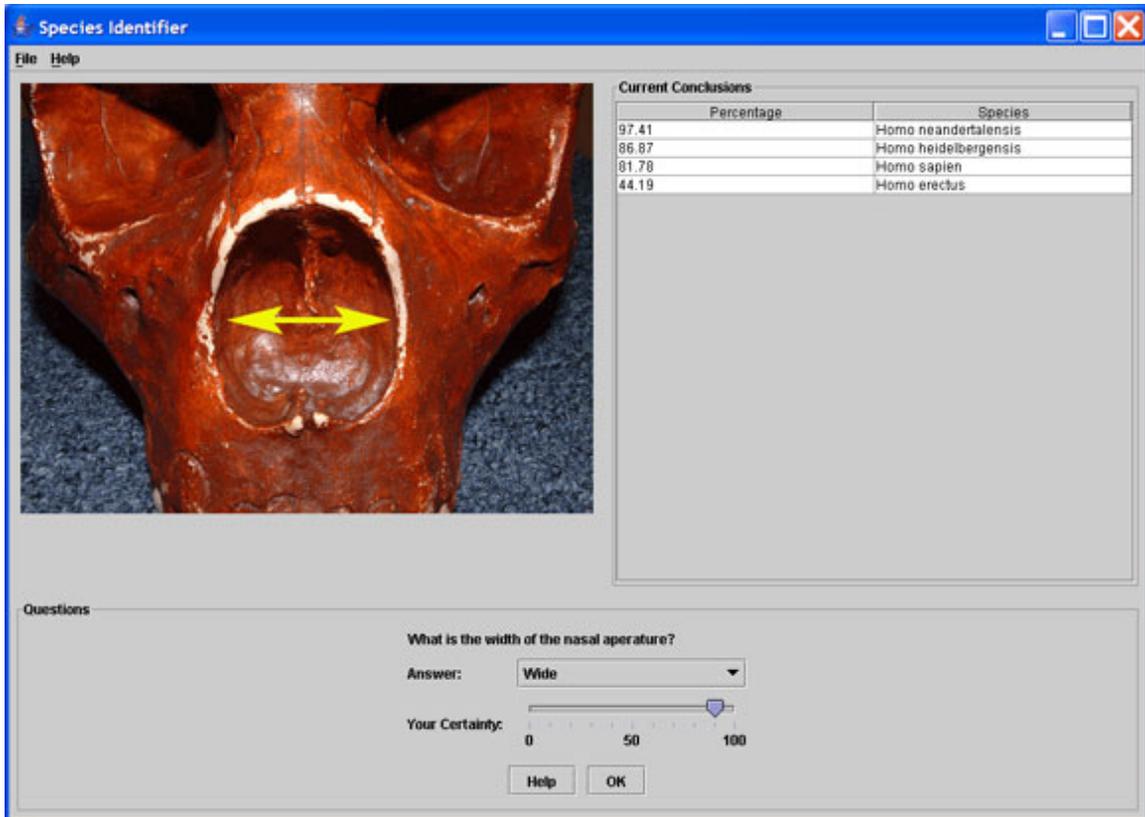


Figure 3-17 Since Species Identifier suspects *H. neandertalensis* could be a conclusion, it asks the user a question specific to that species.

The answer of “Wide” to this question causes Species Identifier to be more confident in the conclusion of *H. neandertalensis* raising its score slightly from 97.41% to 97.92%. Another question specific to *H. neandertalensis* is triggered as the system asks if the skull possesses an occipital bun, a feature unique to Neanderthals. Figure 3-18 displays an image of this question. An occipital bun appears as a mound of bone on the rear of the brain case. This differs from an occipital torus in that it is not a ridge and is more rounded in appearance. The example skull seems to feature an occipital bun and is indicated with a certainty of 80%. This answer increases the applications confidence in *H. neandertalensis* from 97.92% to 99.41%. This question also helped differentiate the confidence in *H. neandertalensis* even further from its contemporaries by decreasing the

certainty score of *H. heidelbergensis* from 86.87% to 82.88% and decreasing the certainty score of *H. sapien* from 81.78% to 69.55%.

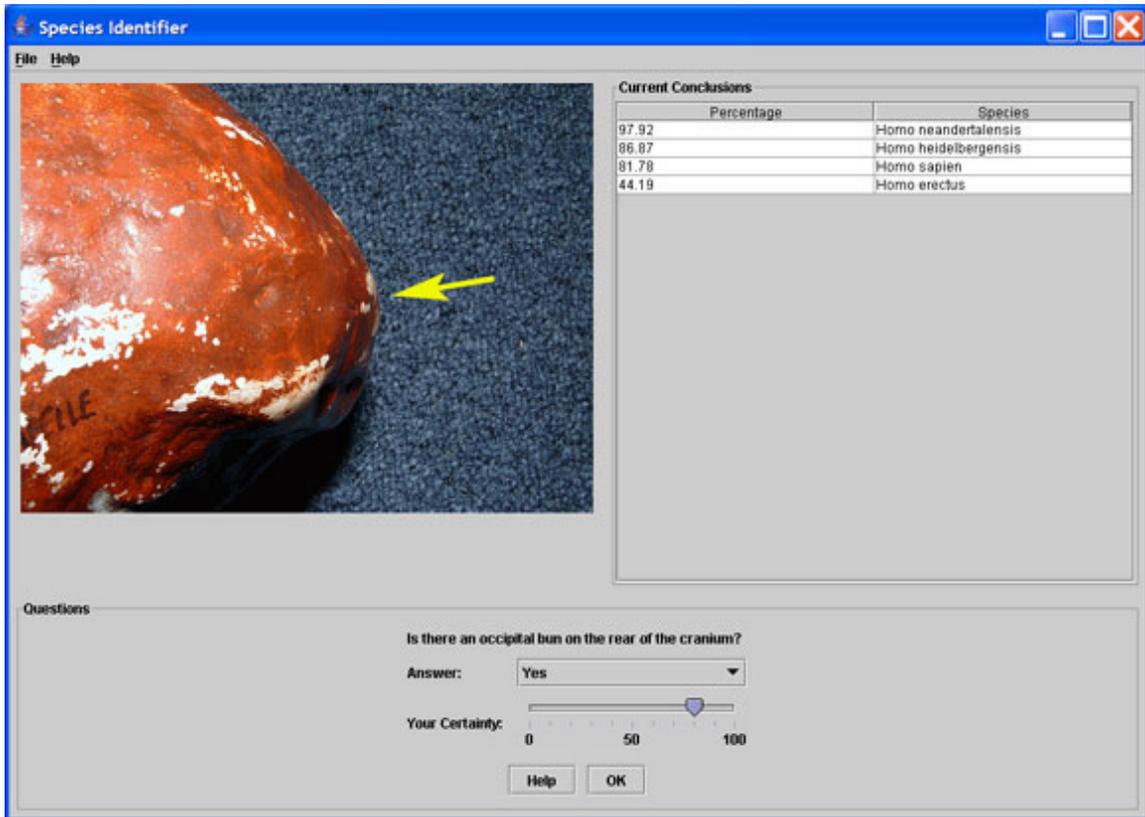


Figure 3-18. The application asks if the skull features an occipital bun.

With the triggered questions for *H. neandertalensis* complete, Species Identifier determines if asking questions specific to *H. sapiens*, which currently has a confidence that rounds up to 70%, should be performed. Since this confidence is at a high enough level, the *H. sapiens* specific questions are activated. The first question regards the presence of a projecting chin on the mandible. This is a feature exclusive to *H. sapiens* among genus *Homo*. A look at the skull seems to reveal no chin, so “No” is selected from the pull down menu with a certainty of 60%. This answer decreases the certainty score of *H. sapien* to 52.54% while making Species Identifier 99.52% confident that the specimen in question is *H. neandertalensis*. The application does not stop at this point.

Figure 3-19 shows the next question in the *H. sapiens* series of triggered questions, which concerns the slope of the forehead. Anatomically, modern humans feature a vertically high forehead. Other species of hominids have a relatively low slope to their forehead. The skull in this example reveals a relatively low forehead, so an answer of “Low” with a certainty of 80% is entered. This answer drastically lowered the measure of belief Species Identifier had in *H. sapien* from 52.54% to 8.51%. This answer also has the effect of slightly strengthening the certainty scores for *H. heidelbergensis* and *H. erectus*.

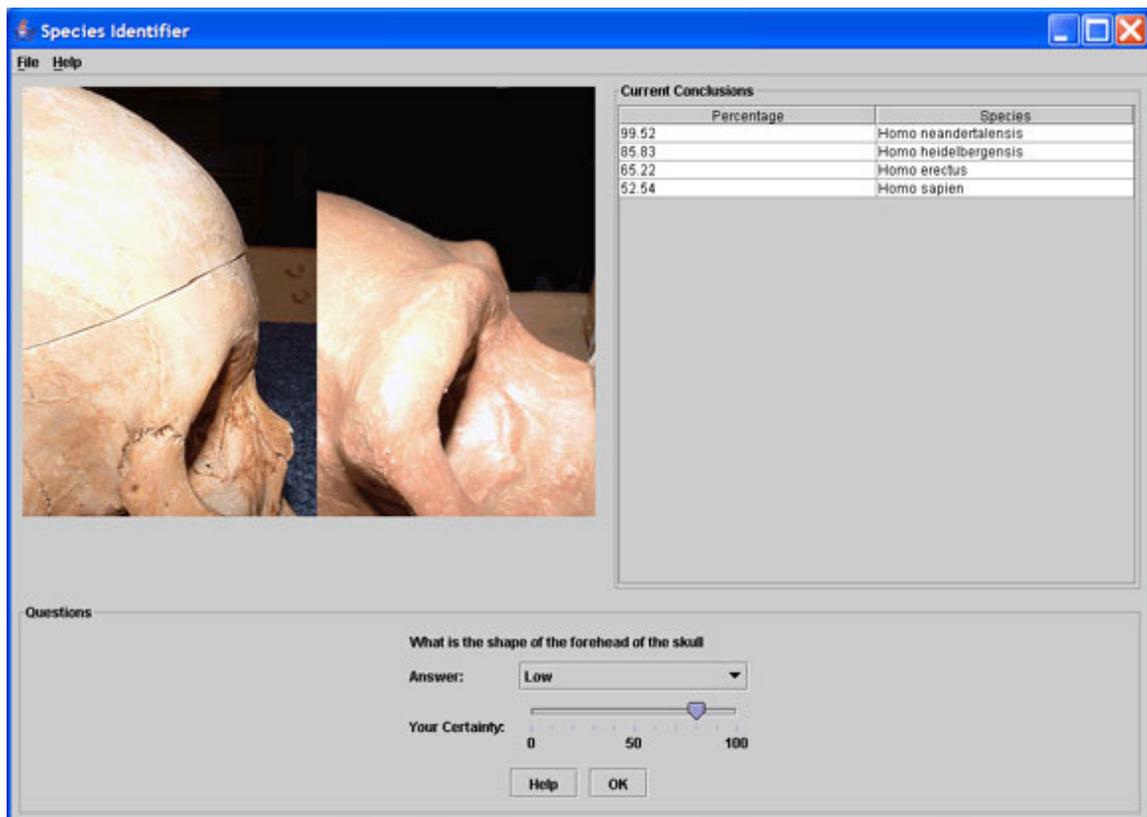


Figure 3-19. Species Identifier asks about the slope of the forehead.

Species Identifier’s Conclusion

Species Identifier has now asked all of the triggered questions for the cranial section, the last section to be examined, so it selects the species with the highest certainty

value as its conclusion. Figure 3-18 shows that Species Identifier is 99.56% confident the fossil remains examined represent the species *H. neandertalensis*. The conclusion function also searches for other species within a factor of 20% of the winning certainty value. In this case, *H. heidelbergensis* also had a high measure of belief of 87.09%. This is identified in the conclusion window shown in Figure 3-20.

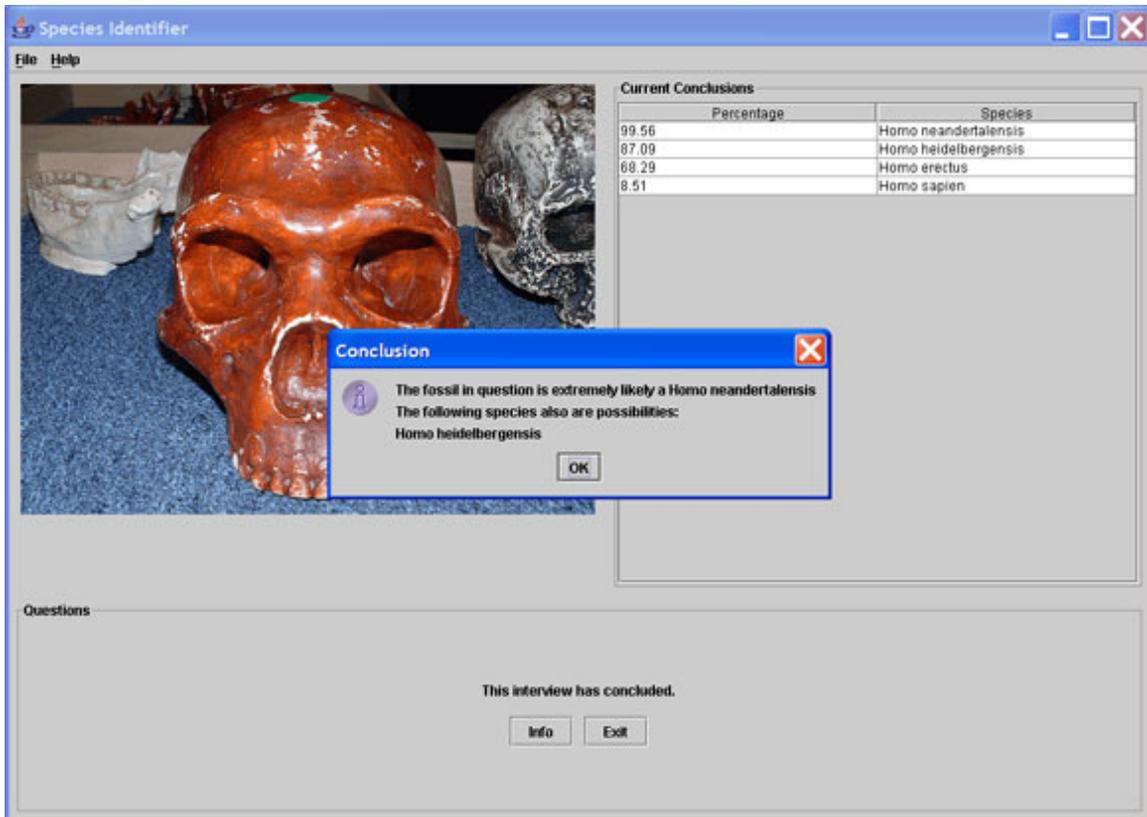


Figure 3-20. Species Identifier displays its conclusion to the examination.

The conclusion screen's image panel features a picture of the skull of the species that the system determined is the most likely solution. The final screen also includes an Info button that displays more background information on the highest scoring species as well as an Exit button to end the program.

Conclusion

This chapter provided an explanation of the controls on the graphical user interface (GUI) of Species Identifier. It also provided a walk-through of the system's execution as it proceeded through an examination of a fossil. The next chapter provides a detailed explanation of Species Identifier's implementation.

CHAPTER FOUR SPECIES IDENTIFIER DESIGN

Species Identifier is an application that identifies the species of a set of hominid fossils. It uses certainty factors to model the decision making process of a biological anthropologist. The implementation of Species Identifier involved three separate phases:

1. The investigation and collection of knowledge in human evolution
2. The development of the Species Identifier engine
3. The integration of the engine with the GUI

Phase one involved the reviewing of knowledge gained from previously taken courses in biological anthropology and current research using websites relevant to the topic. This chapter discusses in detail phases two and three of the implementation.

Design of the Species Identifier Engine

Species Identifier was created using the Java Expert System Shell (JESS) 6.1. JESS, a shell written entirely in Java, is similar in syntax and semantics to CLIPS. One advantage that JESS possesses over CLIPS is that it was written in Java. This allows all Java objects and functions to be easily used since they already built into JESS. This integration with Java makes JESS a potentially powerful tool for developing a rule-based system with a command line interface, a GUI using Java Swing, or an embedded system in a web application [FRI03].

Fact Structure

There are two types of facts used in Species Identifier: ordered and unordered. Ordered facts are simple short and flat lists of information [FRI03]. Ordered facts are

created using the *assert* function. Unordered facts contain named fields for each piece of information. Unordered facts are more structured than ordered facts since they do not rely simply on the order in which the information appears in the fact [FRI03].

An unordered fact is defined using the *deftemplate* function. A fact definition includes the name of the fact along the names and types of slots the fact contains. A slot contains a piece or pieces of information depending on whether the slot is declared as a “slot,” for a single piece of information, or a “multislot,” when a slot can contain multiple pieces of information.

An example of an unordered fact used by Species Identifier is the *ask* fact. The *ask* fact is a simple, one element fact used to indicate to the application that a question is to be displayed to the user. An *ask* fact appears as:

(ask (slot id)).

In the *ask* fact, the value in the id slot is identical to the id of the question to be asked.

One example of an ordered fact used by Species Identifier is the *check* fact. A *check* fact is used to make sure the conditions exist for a question to be asked. The *check* fact’s template is as follows:

(check (slot id) (slot trigger)).

The *check* fact has two slots—id and trigger. The id slot is used to identify the question that has been requested. The id slot of the *check* fact and the id slot of the *question* fact, described later in this chapter, are identical.

Rule Format

Species Identifier is implemented using forward chaining rules. A forward chaining rule is similar to an *if-then* statement in a programming language, except the *then* portion of each rule is executed whenever the *if* portion is matched [GON93]. Figure 4-1 shows

an example of how a rule appears in JESS. A rule in JESS is declared using the *defrule* function. This is followed by the name of the rule and a set of conditions or patterns that must be satisfied for the rule's then portion to execute. The "=>" symbol is interpreted as "then." Each statement after the "=>" symbol is executed only if the conditions before the symbol are satisfied. The rule in Figure 4-1 is fired whenever an *ask* fact is asserted into the fact base. Its execution calls the *ask-question* function and passes the id of the *ask* fact as an argument. Variables identifiers have a "?" in front of the variable name. In the patterns of a rule, variables are assigned information from a fact's slot and can be used to match information within other facts or rules patterns. In Figure 4-1, the element that the *ask* fact contains is stored into the identifier "?id." This variable is then passed to the *ask-question* function as an argument.

```
(defrule rule-1
  "An example of a rule in JESS"
  (ask ?id)
  =>
  (ask-question ?id))
```

Figure 4-1. An example rule in JESS.

Rete Algorithm

JESS implements the Rete algorithm to quickly and efficiently perform pattern matching. The Rete algorithm remembers past testing results throughout the execution of the program, so that only newly asserted facts are tested against the rules [FRI03]. Figure 4-2 illustrates the internal representation of the *get-check* and *ask-question* rules. The execution of these two rules starts when *check* fact is first asserted, which causes the *get-check* rule to fire. The *get-check* rule asserts an *ask* fact. The *question* facts are asserted at

the beginning of the program. If the *ask* fact's id slot matches a *question* fact's id slot, the *ask-question* rule is then fired.

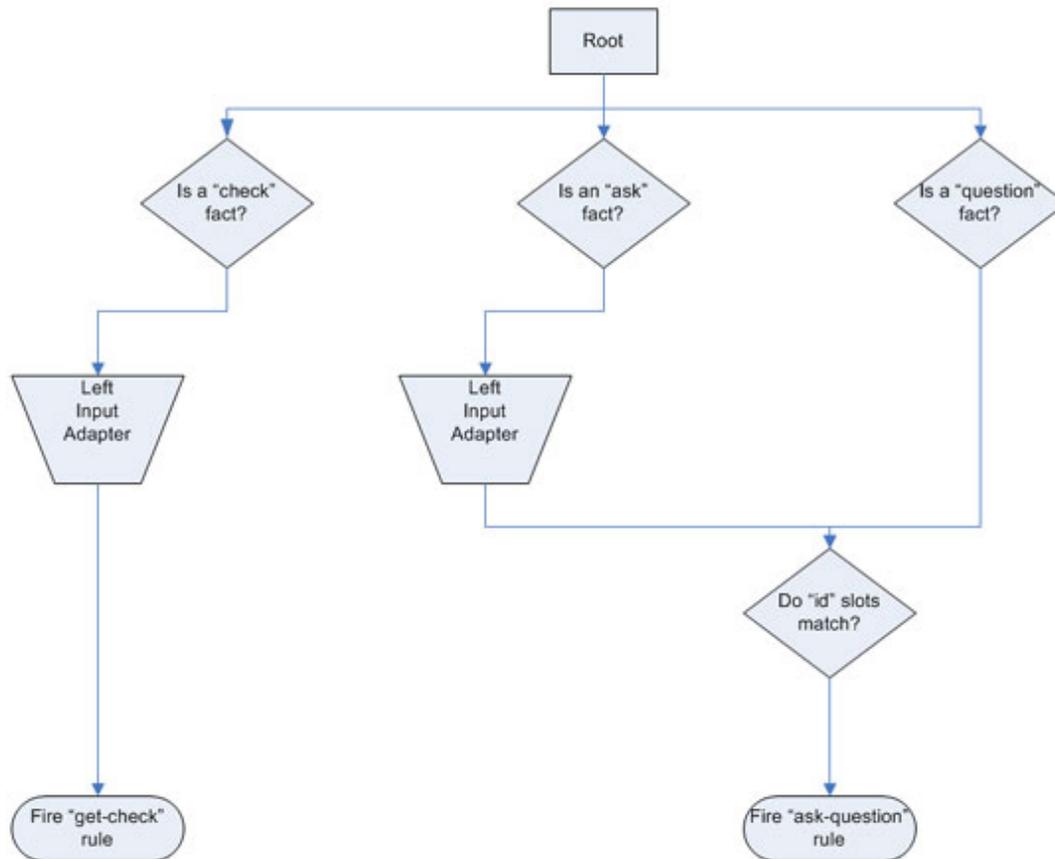


Figure 4-2. A diagram of the Rete network created to execute an *ask-question* rule.

Format

Species Identifier uses an interview format where the system asks the user one question at a time. Another option was to use a single screen form. For use in a laboratory setting, a form could be a better interface for Species Identifier. The form would have all of the general questions asked on one screen, so the user could examine the general traits in whatever order is convenient. Once these values are entered on the form and the form is submitted, the system could ask the user follow-up species-specific questions on a new form, which was constructed based upon the original set of answers.

However, the interview format is more appropriate for several reasons. The interview format better replicates how a biological anthropologist would examine a fossil set. There also are a few practical reasons why an interview format was chosen over a form. Because of the certainty factor sliders on the GUI, putting multiple questions on a single screen form could become confusing for the user. A certainty factor slider would be required for each question creating a form with many controls, all equally accessible. The user could become overwhelmed by the number of controls and have trouble determining which slider goes to which question [JOH00].

Also, the form itself would appear more cluttered than a screen having one question. Displaying one question per screen allows Species Identifier to easily provide assistance to the user. This is accomplished using images as visual aids and the Help button to provide a detailed description about the question being asked.

Implementation of Certainty Factors

The decision to use an interview format meant that the application must present one question to the user at a time, wait for an answer, check the validity of the answer, and match the answer against a proper rule. Once the rule executed, the certainty factor derived by the rule for a species had to then be combined with the existing certainty factor for that species.

Species Identifier implements certainty factors through the use of two rules: *combine-values* and *combine-values-init*. Figure 4-3 shows how the certainty factor equations are implemented in JESS with the *certainty-factor* function. Figure 4-4 displays the source code for the *combine-values* rule, which combines the new certainty factors of each species with the old certainty factor to create an updated score. A *type-human* fact holds the certainty factor value for a species computed from user input for a question. A

type-human-final fact holds the current total certainty factor value for a species. When a *type-human* fact is asserted into the fact base and a *type-human-final* fact already exists in the fact base, the certainty factors for both facts need to be combined. This rule retracts both the *type-human* and the *type-human-final* facts from the fact base. The certainty factors for both facts are passed to the *certainty-factor* function as arguments. The *certainty-factor* function combines the two values using the certainty factor equations for combining values discussed in Chapter Two.

```
;; Calculates the certainty factor
(defun certainty-factor (?cf1 ?cf2)
  "Calculates the current certainty factor"
  (if (and (> ?cf1 0) (> ?cf2 0)) then
    (return (+ ?cf1 (* ?cf2 (- 1 ?cf1))))
    else
    (if (and (< ?cf1 0) (< ?cf2 0)) then
      (bind ?cf1 (abs ?cf1))
      (bind ?cf2 (abs ?cf2))
      (return (- 0 (+ ?cf1 (* ?cf2 (- 1 ?cf1)))))
      else
      (return (/ (+ ?cf1 ?cf2) (- 1 (min (abs ?cf1) (abs ?cf2)))))))
```

Figure 4-3. The function that calculates certainty factors.

The application has two questions that can eliminate a species completely from consideration. These are the questions about the age of the fossil and about location where the fossil was found. If a fossil is found to be an age outside of a set range, its certainty factor value is initialized to -1.0, which means the application is absolutely certain that the fossils do not represent that species.

This certainty factor value, -1.0, will never be able to increase and is ignored by the combining rules. Because of this, the age and location ranges for eliminating species are generous. For example, a species is only eliminated if the age of the species is determined to be over one million years outside its currently known age range.

```

:: Updates the old certainty factor according to new information
(defrule combine-values
  "Combines certainty factors for each rule"
  (declare (auto-focus TRUE))
  :: Get the original certainty factor for the species
  ?old <- (type-human-final (genus ?g)
                        (species ?s)
                        (CF ?cf1))
  :: Get the new certainty factor for the species
  ?new <- (type-human (genus ?g1&:(eq ?g1 ?g))
          (species ?s1&:(eq ?s1 ?s))
          (CF ?cf2))
  :: Ignore if this is an eliminated species
  (not (eq cf2 -1))
  =>
  (retract ?new ?old)

  :: place the species fact back into working memory with the new CF value
  (assert (type-human-final (genus ?g)
                          (species ?s)
                          (CF (certainty-factor ?cf1 ?cf2) ))))

```

Figure 4-4. The source code for the *combine-values* rule.

Implementation of the Engine

Species Identifier's interview engine involves seven major steps. Figure 4-5 details the seven step interact for each question asked during the interview. In this figure, the ovals identify asserted facts, the circles identify fired rules, and the rectangles identify functions that are called. The assembled GUI screen is in the square.

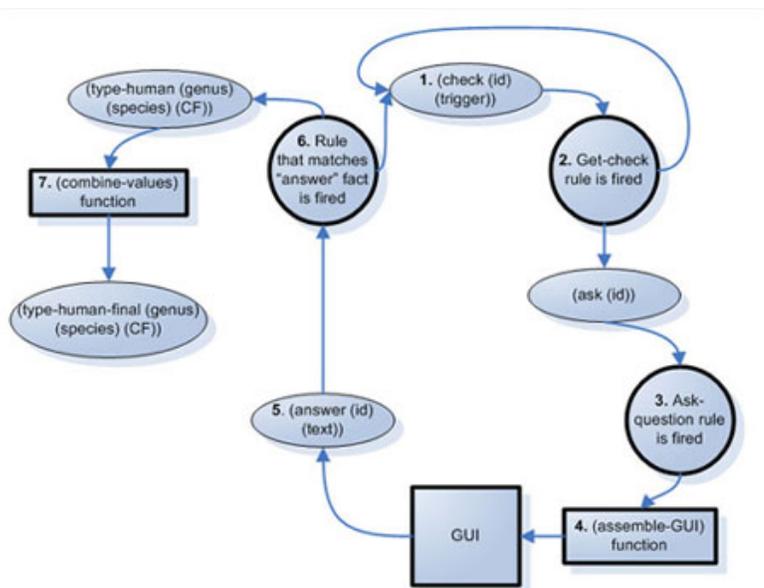


Figure 4-5. A flowchart of Species Identifier's engine.

Step one: check facts

The first step of the system is to assert a fact into the fact base telling Species Identifier that a new question is to be displayed to the user. This is accomplished using a *check* fact. See Figure 4-6.

```
;;; THICKNESS OF ENAMEL ;;;
(defrule MAIN::thickness-tooth-enamel
  "Determine the thickness of the tooth enamel"
  (answer (id tooth-enamel) (text ?text)
    (certainty ?certainty))
  =>
  (assert (check (id dental-arcade) (trigger no)))
```

Figure 4-6. An example of a rule asserting a *check* fact.

The highlighted line of code in Figure 4-6 asserts a *check* fact into the fact base. This *check* fact begins the process of asking the question with the id “dental-arcade.” Certain conditions may need to exist for this question to be asked as indicated by the trigger slot. The trigger slot can be a value of “yes” or “no.” The trigger slot is used to identify a triggered question. If a question is a triggered question, it is specific to one or a small group of hominids. The specific species must have a certainty factor score above a set level for the Species Identifier to ask a question. Because the *check* fact in Figure 4-6 has a value of “no” for triggered, the question will be asked regardless of any conditions that may exist.

Check facts are used to control the flow of questions asked by the system. After the user answers a question, a *check* fact for the next question in the progression of the flowchart is asserted into the fact base. If the *check* fact was a triggered question that did not match the requirements for the question to be asked, another *check* fact is asserted so the flow of the program can continue.

Step two: the *get-check* rule

The second step of the interview engine is the execution of the *get-check* rule. A *get-check* rule is fired when a *check* fact is asserted. *Get-check* then retracts the *check* fact from the fact base and verifies the bounds associated with the id of the *check* fact. If the *check* fact's trigger slot was set to "no," then *get-check* asserts an *ask* fact telling the system to display the text of the question. See Figure 4-7.

```
(defrule get-check
  "Gets the check token for the next question in the interview"
  ?fact <- (check (id ?id) (trigger ?trigger))
  =>
  (retract ?fact)

  ;; If the question is not a triggered question
  (if (or (eq ?trigger no)
          (eq ?*debug* TRUE)) then
      (assert (ask ?id))
```

Figure 4-7. An example of asserting an *ask* fact.

In Figure 4-7, the *get-check* rule matches a *check* fact that has been asserted into the fact base. The *check* fact is assigned to the variable "?fact," which is retracted from the fact base. If the trigger slot of the *check* fact was set to "yes," the *get-check* rule must check the certainty scores of the species associated with this question to make sure that it is worthwhile to ask. The *get-check* rule first determines the id of the requested question. Then, it searches for the certainty factor for the species specific to this question to see if the species' score is high enough for the question to be activated. This is accomplished by activating queries to the fact base. "...queries are used to search the working memory under direct program control. A rule is activated once for each matching set of facts, whereas a query gives you a java.util.Iterator of all the matches" [FRI03, pp. 128].

There is a query defined for each of the twelve species that the application can recognize, which returns the fact associated with one of those species. The *get-check* rule

calls a function named *get-CF* with the identity of the species passed as an argument. The *get-CF* function then calls a query defined for that species. The fact associated with a species contains a slot with the current certainty factor score for that species. This value is returned to the caller. If the certainty factor value is above a specified level, then an *ask* fact is asserted into the fact base and the question is displayed to the user. If the certainty factor value is below the specified level, then the *check* fact for the next question in the flowchart is asserted and the process starts over.

Step three: the *ask-question* rule

The *ask-question* rule is fired when an *ask* fact is asserted and there exists a *question* fact with a matching id slot. Species Identifier represents each question it can ask as a *question* fact. Figure 4-8 illustrates a *question* fact used in this application.

```
(question (id dental-arcade)
          (type multistring)
          (text "What is the shape of the dental arcade?")
          (valid "Rectangular"
                "Parabolic"
                "Don't Know")
          (certainty yes)
          (image "dental-arcade.gif"))
```

Figure 4-8. A *question* fact used by Species Identifier.

This *question* fact contains information for a question asking about the shape of the dental arcade. The id slot is used to identify the question and to match the fact with an *ask* fact. In Figure 4-8, the id slot contains the value “dental-arcade”, which refers to this *question* fact. The type slot identifies the type of question this fact represents: number (for questions requiring numerical input) or multistring (for multiple choice questions). In this case, the type slot contains the value “multistring.” This value indicates to the GUI that an option menu is used to display the possible answers. The text slot contains the text

of the question that is displayed to the user. The valid slot lists all the valid answers applicable to this question if it is a multiple-choice question. The valid answers for the *question* fact in Figure 4-8 are “Rectangular,” “Parabolic,” and “Don’t Know.” The certainty slot identifies whether the question requires a certainty value be entered. In Figure 4-8, the certainty slot has a value of “yes,” so the answer to this question is factored into the conclusion. Finally, the image slot is used to hold the filename of the image, “dental-arcade.gif,” which will be displayed in the GUI.

All of the *question* facts are asserted at the beginning of the application when a reset command is issued. The *ask* fact acts as a token for the *ask-question* rule. Any time the *ask* fact is asserted with an id matching a *question* fact’s id, the rule fires for that particular question.

Once the *ask-question* rule has fired, it retracts the *ask* fact from the fact base and calls the *assemble-GUI* function discussed in the next section. The rule then calls the *waitForActivations* function, which halts the program until the working memory has been altered by some other thread [FRI03]. This causes the application to pause and wait for the user to enter information into the GUI screen assembled in the next step.

Step four: the *assemble-GUI* function

For each question, the *assemble-GUI* function creates the screen where the user can read the question and enter an answer. The details of the implementation of the GUI are discussed later in this chapter. The application passes all of the information contained in a *question* fact to the *assemble-GUI* function as arguments. This information includes: the text of the question, the image file, the type, the list of valid answers, and the certainty factor if required by the answer.

The *assemble-GUI* function begins by clearing each of the three panels of the GUI. The image panel is redrawn using the image filename passed as an argument. The current conclusions table is assembled next by calling the *assemble-conclusions* function.

Based on the Boolean value passed to *assemble-GUI*, the function determines if this question requires a certainty factor slider for the answer. If the value is false, then the *assemble-GUI* function grays-out the certainty factor slider so that the user cannot select it. If the value is true, then the function allows the user to modify the value represented by the slider.

Next, the *assemble-GUI* function creates the questions panel. The function uses the type argument to determine if the answer type is numeric, requiring an input box, or if the type is multiple-choice, requiring a dropdown menu of options. The text of the question is placed in a label using the value of the variable text passed to this function. The questions panel is assembled by placing the question text on the first line, the method to input an answer on the next line, and the certainty factor slider on the line below that. Finally, the Help and OK buttons are placed on the last line. The application returns control to the *ask-question* rule where it calls the *waitForActivations* function. Species Identifier now waits for the user to input an answer and click the OK button before continuing.

Step five: check input for correctness

Only questions requiring numerical input are checked for validity. Multiple choice questions only list valid input as possible answers so the validity of answers for these types of questions need not be checked. Once the user presses the OK button to enter the answer, it is checked for validity by the *input handler* function if needed.

The numerical input answers are first checked to ensure they are a number. If the input is valid, an *answer* fact is asserted. Figure 4-9 illustrates an *answer* fact used in the application.

```
(deffunction read-input (?EVENT)
  (bind ?answer (?*jAnswers* getSelectedItem))
  (bind ?certainty (/ (integer (?*slider* getValue)) 100))
  (assert (answer (id ?*answer-id*) (text ?answer) (certainty ?certainty))))
```

Figure 4-9. An example of an *answer* fact.

The id slot matches the id slots of the previous *check*, *ask*, and *question* facts. In Figure 4-9, the id is determined from the global variable “?*answer-id*,” which contains the id value for the current question. The answer the user supplies is held in string or numerical form, depending on the question, in the text slot. In Figure 4-8, the *read-input* function expects the user to select an answer from an option menu. The *getSelectedItem* function returns the answer that has been selected from the option menu. If the answer did not require a certainty factor value, the value held by the CF slot is simply “nil.” If the question did require a certainty factor value, the value on the certainty factor slider is recorded by the *input handler* function. This value is divided by 100 to convert it into a number from 0.0 to 1.0. This number is now suitable to be combined with previous certainty factors using the certainty factor formulas.

If the input is determined to be invalid by the input handler, a small dialog window is created to inform the user that they have entered an invalid answer. Once the user presses the OK button on this window, the *ask* fact associated with this question is reasserted by the input handler and the cycle starts again at step three. If the input is valid, the *answer* fact is asserted as shown in the highlighted region of Figure 4-8, and a rule matching this answer is fired.

If the answer requires numerical input, then the number entered must be checked to ensure its falls within the bounds of valid answers. For example, Species Identifier only recognizes species of hominids as far back as 4.5 million years ago. If the user enters an age outside this range, then the system will not be able to recognize the species. Rules that match this *answer* fact check the bounds and fire if the numerical answer is invalid. The rule uses a dialogue window to notify the user that the numerical answer is invalid. The rule then reasserts an *ask* fact and the question is asked again starting at step three.

Step six: *answer* rules

Figure 4-10 shows a rule that asks a question about the existence of a sagittal keel. The rule fires if an *answer* fact with the id “sagittal-keel” is asserted into the fact base.

```

;;; SAGITTAL KEEL ;;; (if suspect erectus)
(defrule sagittal-keel
  (declare (auto-focus TRUE))
  (answer (id sagittal-keel) (text ?text)
          (certainty ?certainty))
  =>
  (assert (check (id occipital-torus) (trigger no)))

  ;; If there is a sagittal keel
  (if (eq ?text "Yes") then
      (assert (type-human (genus Homo) (species erectus)
                          (CF (* 0.5 ?certainty))))))

  ;; If there is not a sagittal keel
  (if (eq ?text "No") then
      (assert (type-human (genus Homo) (species erectus)
                          (CF (* -0.5 ?certainty))))))

```

Figure 4-10. An example of how an *answer* rule appears in the source code.

This engine uses two types of *answer* rules. The first type is for numerical input. These rules fire if an *answer* fact with a certain id is asserted and if the value in the text slot matches a certain range. The second type of rule is for the string input used by the multiple-choice questions. These rules fire if an *answer* fact with a certain id is asserted.

Once this type of rule fires, *if-else* statements are used to determine what actions to take based on the string input.

Next, the *answer* rule asserts *type-human* facts for all species to whom this question applies. A *type-human* fact appears as:

(type-human (slot genus) (slot species) (slot CF)).

The genus slot holds the genus of the species. This can be either “Ardipithecus,” “Australopithecus,” or “Homo.” The species slot holds the actual species name. The CF slot contains the certainty factor value associated with this rule. This CF value represents the belief in this species given the evidence presented from the answer [GON93]. The CF slot’s value is the certainty from 0.0 to 1.0 that the user entered using the slider multiplied by the certainty value associated with the rule:

(User certainty) * (Rule certainty).

In Figure 4-10, the *answer* rule asserts a *type-human* fact for *Homo erectus* with a certainty factor of 0.50 times the user’s certainty if the answer is “Yes” or a certainty factor of -0.50 times the user’s certainty if the answer is “No.” All of the *type-human* facts that were asserted by the *answer* rule are now combined with the previous certainty factor values existing for each species. This is triggered by the assertion of a *type-human* fact to update the certainty factors for each species, then a *check* fact for the next question is asserted.

Step seven: combine certainty factors

The rules for combining certainty factors assert a fact into the fact base called *type-human-final*. This fact appears as:

(type-human-final (slot genus) (slot species) (slot CF)).

The slots for genus and species are identical to the slots of the same name for the *type-human facts*. The CF slot holds the current certainty factor value for a species. The *type-human-final* fact is the final result of the interview engine.

There are two rules for combining certainty factors. The first rule, *combine-values-init*, executes if a *type-human* fact is asserted into the fact base and if there is no *type-human-final* fact already in the fact base for the given species. The first rule initializes the certainty factor of the new *type-human-final* fact to be the exact value in the *type-human* fact. The second rule, *combine-values*, factors the certainty value in the *type-human* fact into the existing *type-human-final* fact. This is accomplished by calling the *certainty-factor* function which implements the certainty factor formulas described in Chapter Two. Both rules retract the *type-human* fact upon a match.

Once the new certainty value has been calculated, the *type-human-final* fact is reasserted into the fact base with the updated value. Species Identifier now begins the process of asking the next question in the sequence through the *get-check* rule, which fires when the new *check* fact is asserted in step six.

Implementation of the GUI

A graphical user interface was chosen because it makes the system easier to use. Since JESS has the ability to call Java functions and classes, Swing was used to create the GUI. The GUI consists of three main components:

1. an image for each question Species Identifier asks,
2. a display of each species' current certainty score, and
3. a method for users to read a question and enter an answer along with their certainty in that answer.

The GUI uses group boxes to separate each of these three components. Of the three components, only the third component is needed for the application to function properly. The first two components are to aid the user in utilizing the application. The first component is included to help the user better understand the question. The second component displays to the user what the system is thinking after each question and shows how each question affects this thinking.

Screen Layout

The layout of the screen is designed to keep the user focused on the question panel of the screen. If the user needs help answering the question, they can look to the top left portion of the GUI for a visual aid or press the Help button. If the user is interested in what Species Identifier is currently thinking, they can look to the top right portion of the screen to examine the current conclusions table. This layout was chosen so the user does not need to move their eyes to different parts of the screen unless the user needs more information. A user could very easily use the entire program without glancing at any portion of the GUI other than the question panel.

A menu bar with “File” and “Help” menu items is located along the top following the design principle that any application should display a menu bar to avoid being confused with a dialog window [JOH00]. The OK and Help buttons are placed at the bottom of the window and are centered. This flow follows the design principle that any command buttons that affect the entire window should be located at the bottom-center of the window [GAL97]. The OK and Help buttons were placed in the questions panel to limit the amount of cursor travel required for each question. For each question, the user only needs to keep the cursor in the question panel unless they want to restart the

program or exit. This follows the design principle to minimize the amount of cursor movement required [GAL97].

Functionality

When designing a GUI, first consider the purpose of the application [JOH00]. Species Identifier's purpose is to interview a user and assess the user's answers to determine the species of a set of hominid fossils. The application accomplishes this by asking the user a sequence of questions. The user must be able to view the questions as well as answer them. The user must also be able to indicate their certainty in the answer given.

Only information in the task domain that interests the users should be displayed in the GUI [JOH00]. The target audience of this application is a biological anthropologist or student working in a lab. This audience is not concerned with concepts of more interest to the field of computer science such as the current status of the fact base or which rule is firing. For this reason, only the details of the application that would be useful in a laboratory setting—the question, an image, and current conclusions—were included in the GUI.

Design of the Question Panel

The most crucial portion of the GUI is the question panel. If this part of the GUI is incomprehensible to the user, then the entire application is unusable. The question panel must include the question text, a method to answer the question, and a method of indicating certainty in the answer.

Question text

The first design element that needed to be implemented was the display of the question text. One of the bloopers listed in Jeff Johnson's book GUI Bloopers is to use a

text field for read-only data. Using a text field to display read-only data leads to user confusion since they will not know if the value in the text field needs to be altered [JOH00]. Read-only data should be displayed using a label, which is not editable, so there is no confusion about what items need to be set before clicking the OK button to answer the question [JOH00]. For this reason, the text for each question is displayed using a label. This is implemented through the use of the Swing class JLabel. The JLabel class creates a read-only label, which does not react to input and cannot obtain keyboard focus.

Answer input

On the line below the question text, a method of entering an answer is provided. There are two types of answers that can be entered for each question: a number or a string. Most of the questions Species Identifier asks require string input. The methods for providing users a way enter an answer include a text field and an option menu. The options are to inform the users of valid answers and require them to type in the answers, allow users to enter any answer and try to match their input with valid answers, or provide an option menu where the users can select among valid answers.

“Text fields should only be used when the data is unstructured, free-form text” [JOH00, pp. 127]. In this application, only a small set of acceptable answers exists for each question. For example, when a question asks “In general, what is the cranial capacity,” the only answers allowed are “Small,” “Moderate,” “Slightly Enlarged,” “Large,” or “Don’t Know.” These answers are broad enough to cover all possible sizes of a hominid brain and detailed enough to provide information helpful in determining a conclusion. If the user does not believe their answer matches any answer provided, they

can select what they believe is the closest answer and express their uncertainty in the answer using the certainty slider.

For answers requiring string input, an option menu containing valid answers was chosen over a text-field where users must enter their answers. This prevents the application from having to perform error checking for string answers and from having to attempt to match user input with answers the application recognizes. Allowing free-form text answers could prove to be frustrating to the user if the user must type in an answer for each question. User frustration could rise if they enter data the application does not recognize and must retype the answer.

Multiple-choice answers solve each of these problems by only requiring the user to click on an answer from a set of valid choices. An option menu uses the same amount of space on the GUI no matter how many options populate the list. Because there are a different number of possible answers for each question, an option menu is preferable over a set of radio buttons because it does not alter the layout of the GUI for each question.

Text fields are provided for answering questions requiring numerical input where the range of possible numerical answers is greater than a few numbers. For example, the question about the age of the fossils can accept answers from zero to 4.5 million. Multiple-choice answers are not provided for these answers since the number of possible answers is extremely large—they can be considered free-form text.

For multiple-choice answers, the Swing class `JComboBox` is implemented. The option menu is populated with answers using values in the valid slot of the *question* fact. For numerical input, the Swing class `JInput` is used.

Certainty slider

On the line below the user's input is a slider for users to indicate their certainty in the answer entered. Among the methods considered for allowing a user to indicate certainty were a text field, an option menu, a set of radio buttons, and a slider. Users indicate their certainty on a scale of 0-100. This number is translated internally to a number between 0.0 and 1.0 that represents the confidence that a user has in an answer. The numbers are displayed on a different scale since users are more use to indicating certainty of 0-100% as opposed to a scale of 0.0 to 1.0.

A text field is not used to save the user from having to manually type a number for each question requiring a certainty value. Error checking is another reason a text field is not used. The number of characters would need to be limited to three to allow a number from 0-100. This creates a possibility that the user can enter a number outside this range, forcing the user to reenter the certainty value. A text field should be used when there is a great range of values that can be entered and when the use of a selection list is not possible [GAL97]. In the case of displaying certainty values, the range of values is limited and a selection list could possibly be used.

Radio buttons should only be used when the number of possible values is between two and eight [GAL97]. The number of possible values for indicating certainty is far outside this range. It is possible to allow users to only select from among the multiples of ten in the range of 0-100. However, this still results in eleven possible values, which is still outside the range of two through eight. An option menu was not used because the number of values listed would require the user to scroll through the list to find the proper value.

A slider is used to display certainty values because they are designed for cases in which users need to select a value from among a finite set of continuous values [GAL97]. The slider is labeled with the values of 0, 50, and 100 to mark the low, intermediate, and high values. The Java Swing class `JSlider` is used to implement the slider on this GUI.

There are some questions that do not require a user to input certainty. These are questions whose only possible answers are “Yes” or “No,” and whose answer has no effect on the conclusion of the system. For example, the question that asks, “Are portions of the skull available” only exists for Species Identifier to determine if it needs to ask cranial questions. The answer of this question has no direct effect on the certainty scores of each species, so the slider is grayed-out prohibiting users from clicking on it.

Current conclusions table

The current conclusions table is created using a table inserted into a scrolling pane. The elements of the table are listed in descending order. An insertion sort algorithm is used to sort the species on the table. At most, there are twelve species listed on the table. An insertion sort algorithm is extremely fast when the number of elements to be sorted is small.

The current conclusions table is implemented using the Java Swing class `JTable`. `JTable` requires a multidimensional object to represent the columns and rows of the table. JESS, however, does not allow the use of multidimensional arrays—it automatically flattens any multidimensional array into a linear list [FRI03]. To compensate for the lack of a multidimensional array, the class `java.util.Vector` is used to create a vector of vectors holding the columns and rows of the table.

Image panel

Applying an image to the Java Swing class JLabel creates the image panel. The image is determined for each question from the GIF image file named in the “image” slot of the *question* fact that matched the current question.

Conclusion

This chapter described the implementation details of Species Identifier’s engine and the design principles used to create the GUI. The next chapter examines how Species Identifier could be extended in the future and concludes this thesis.

CHAPTER FIVE CONCLUSION

This thesis examines the development of a knowledge-based application that attempts to identify the species of a set of hominid fossils. This application uses certainty factors to model the decision making process that a biological anthropologist uses when examining fossils in a laboratory setting.

Chapter One introduced the problem. Chapter Two provided background information on biological anthropology and certainty factors. This chapter also examined previously developed rule-based systems as well as similar applications that exist in the field of forensic anthropology. Chapter Three explained the controls on the GUI and provided a walk-through of the application's execution. Chapter Four provided the implementation details of the Species Identifier engine and the GUI.

Species Identifier Results

Species Identifier can identify twelve species of hominids based on a fossil set. The application successfully implements certainty factors, a feature not automatically available within JESS, as a method of representing human uncertainty in observations of characteristics among fossils. When a biological anthropologist is examining a set of fossils, there are many instances where the anthropologist is not absolutely certain in an observation. There are also instances where the anthropologist does not have enough fossil evidence to provide an answer with any certainty. Species Identifier is able to model all of these instances through the use of the certainty slider on the GUI, which allows users to indicate their certainty in an answer on a scale of 0-100%. Following the

interview, the species with the highest certainty value is displayed as the conclusion. Other species that scored within a factor of 20% of the final score are also displayed as possible conclusions.

Species Identifier uses a GUI to display the questions and allow users to enter their answers. This GUI provides users with graphical information. This graphical information includes an image that serves as a visual aid for each question and a current conclusions table that displays the potential conclusions the application is considering at each set of execution. Besides aesthetic purposes, the GUI also allows users to easily indicate their answers by use of an option menu for most questions. This saves the user from having to manually type each answer, input errors, and provides a guide for what characteristics the user should observe for each question.

Future Work

There are many ways Species Identifier can be extended for future work. One extension would simply be the addition of more species-specific questions. The addition of these questions would aid in providing greater differentiation between species as well as more confident conclusions. Additionally, questions pertaining to gender should be implemented. Many hominid species exhibit some degree of sexual dimorphism. Being able to determine the gender of the fossils would enable Species Identifier to identify more information about a fossil set and possibly increase its conclusion accuracy.

Another addition in this area would be the addition of postcranial questions. Efforts were made to add these types of questions. However, fossil evidence for postcranial remains is fragmentary at best for many of the older species that this application can recognize. For example, most of the fossil evidence for *Ardipithecus ramidus* is dental remains [KRE04]. Barring future discoveries, adding postcranial questions general to all

species could require some amount speculation for some species. Adding species-specific postcranial questions could produce more accurate results than attempting to add general questions.

A feature in the Anthropologist application discussed in Chapter Two that would be extremely useful in Species Identifier is the ability to choose a different expert for an interview. There are many different schools of thought in biological anthropology. Capturing the knowledge of various researchers allows the application to offer multiple opinions on a set of fossils. When examining a set of fossil remains, a researcher could use Species Identifier to get different opinions from various experts. For example, if a researcher believes they may have discovered a new species, they could use Species Identifier to quickly get an idea of what other experts would classify the fossil as before declaring a new species.

Finally, Species Identifier can be extended by including a feature where the application can explain the reasoning it used to come its conclusion. This feature would give users better insight into how the application derived its results. Some insight can already be gained from monitoring the current conclusions table, but a detailed description at the end of the interview would likely be more informative.

Summary

Species Identifier models the decision making process used to determine the species of hominid fossils. This application would be useful as an instruction tool in biological anthropology classes and laboratories. It could also prove to be a valuable tool to a biological anthropologist by aiding in species identification. With some extensions, it can also be used to quickly provide additional opinions on species identification.

APPENDIX GLOSSARY

Anterior: Towards the front portion.

Anterior Pillars: Vertical columns of bone on the sides of the nasal opening. These columns help the specimen withstand stresses caused from chewing.

Bilophodont: Four cusps on the grinding surface of the molar.

C/P3 Shearing Complex: A feature where the canines grind against the premolars during normal use. This grinding acts to keep the canines sharpened.

Canine: Pointed tooth used for piercing.

Canine Fossa: A depression on the maxilla caused by the canine root.

Crest: A ridge of bone used to attach muscles.

Cusps: An elevation on the surface of a tooth.

Diastema: A space between teeth for long teeth on the opposite jaw. In hominids this space allows the mouth to shut when large canines are present.

Dental Arcade: The row of teeth. In *H. sapiens* this row forms a parabolic shape. In older hominid species this row is more rectangular in shape.

Enamel: The hard, mineralized, outer layer of a tooth.

Facial Prognathism: Forward projection of the facial region.

Incisor: Flat, frontal teeth used for cutting.

Molar: Rear tooth used for grinding.

Nasal Aperature: The nasal opening.

Occipital Bun: A mound of bone on the back of the skull.

Occipital Torus: A ridge of bone on the back of the skull.

Orthognathism: No forward projection of the facial region.

Palate: The roof of the mouth.

Postcanine Megadontia: Characteristic where the molars and premolars of a mandible are much larger than the anterior teeth. Such large molars are used to grind tough, fibrous vegetation.

Posterior: Towards the back portion.

Postorbital Constriction: The narrowing in of the skull behind the orbitals.

Premolar: Somewhat flat tooth lying in between the canine and the molars

Sagittal Crest: Ridge of bone running along the midline of the skull. It connects large muscles from the skull to the jaw to produce great chewing force.

Sagittal Keel: A small ridge of bone running along the midline of the skull. Not as pronounced as a sagittal crest.

Shovel-shaped Incisors: Incisors with a scooped out shape on the back side of the tooth.

Supraorbital Sulcus: A depression between the brow ridge and the forehead.

Supraorbital Torus: A ridge of bone running along the brow of the skull.

Y-5 Molar Cusp Pattern: Molars which exhibit five raised cusps on the chewing surface. The space in between the cusps forms a "Y" shape.

Zygomastics: The cheek bones.

LIST OF REFERENCES

- [ART04] Artificial Intelligence Group. Mycin.
<http://www.computing.surrey.ac.uk/research/ai/PROFILE/mycin.html>, Accessed January, 2004.
- [AGO04] Agogino, A. Expert Systems in Mechanical Engineering. Introduction to Expert Systems.
<http://best.me.berkeley.edu/~aagogino/me290m/s99/Week2/week2.html>, Accessed January, 2004.
- [CON97] Conroy, G. *Reconstructing Human Origins: A Modern Synthesis*. New York City, New York: W. W. Norton & Company, 1997.
- [COU04] Court TV's Criminal Library: Criminal Minds and Methods. Profile of Dr. Bill Bass, Founder of the Body Farm.
http://www.crimelibrary.com/criminal_mind/forensics/bill_bass/5.html?sect=21, Accessed January, 2004.
- [DMI04] Dmitry, M. Forensic Medicine Anthropologist Programs.
<http://www.geocities.com/anthropolog1/>, Accessed January, 2004.
- [FRI03] Friedman-Hill, E. *JESS in Action: Rule-Based Systems in Java*. Greenwich, Connecticut: Manning Publications, 2003.
- [GAL97] Galitz, W. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. New York City, New York: John Wiley & Sons, 1997.
- [GON93] Gonzalez, A. and Dankel, D. *The Engineering of Knowledge-Based Systems: Theory and Practice*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [GRO04] Groves, C. *Australopithecus garhi: A New Found Link*.
http://home.austarnet.com.au/stear/cg_australopithecus_garhi.htm, Accessed January, 2004.
- [INF04] Information Systems: A Management Perspective. Information Systems—Useful Cases.
<http://www.prenhall.com/divisions/bp/app/alter/student/useful/ch12dec.html>, Accessed January, 2004.
- [INS04] Institute of Human Origins. Becoming Human: Paleoanthropology, Evolution, and Human Origins. <http://www.becominghuman.org/>, Accessed January, 2004.

- [JOH00] Johnson, J. *GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers*. San Francisco, California: Morgan Kaufmann Publishers, 2000.
- [KRE04] Kreger, C. A Look at Modern Human Origins. Australopithecus/Paranthropus robustus. <http://www.modernhumanorigins.com/robustus.html>, Accessed January, 2004.
- [LAN04] Landsbergen, D. Introduction to Expert Systems: Mycin. <http://ppm.ohio-state.edu/ppm/~landsbergen/classes/ITP/ES.pdf>, Accessed January, 2004.
- [LUG02] Luger, G. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley, 2002.
- [RHI99] Rhine, S. *Bone Voyage: A Journey in Forensic Anthropology*. Albuquerque, New Mexico: University of New Mexico Press, 1999.
- [SAN04] Sanders, J. A Test of the Postcranial Discriminant Functions of FORDISC 2.0 Using the Hamann-Todd Collection. <http://archlab.uindy.edu/SandersJL.html>, Accessed January, 2004.
- [STE96] Stein, P. and Rowe, B. *Physical Anthropology*. New York: McGraw-Hill, 1996.

BIOGRAPHICAL SKETCH

Robert D. Cooper was born in Tampa, Florida, on September 2, 1978 where he was raised to love the Tampa Bay Buccaneers and the Florida Gators. He graduated from Robinson High School where he played football and ran on the cross country team.

He came to the University of Florida with the purpose of earning a bachelor's degree and a J.D. in law. Months before completing his undergraduate education, he developed a passion for computer programming and decided to study computer and information science instead of attending law school at the University of Florida's Frederic G. Levin College of Law.

He graduated with a bachelor's degree in anthropology in spring 2001 and married his high school sweetheart in Las Vegas, Nevada. As a postbaccalaureate student, he completed the prerequisite courses for admission into the computer and information science graduate program at the University of Florida in Fall 2002.