Computers Math. Applic Vol. 20, No 9/10, pp. 125–140, 1990
Printed in Great Britain.

0097-4943/90 $3.00 + 0.00

DIAGNOSING JAUNDICE EXPERT SYSTEM

L. Ü. YALÇINALP and L. STERLING
Department of Computer Engineering and Science, Case Western Reserve University, Cleveland, OH 44106, U.S.A.

Abstract—DIJEST (DIagnosing Jaundice Expert SysTem) is a medical expert system which produces a differential diagnosis of a patient presenting with jaundice. DIJEST is written in Prolog, and illustrates the use of the language for clearly expressing knowledge. Specifically, the expert system contains explicit declarative knowledge of anatomy and physiology which is used by clinicians when diagnosing obstructive jaundice. The inference engine matches patient records against expected manifestations of symptoms in diseases. Novel in DIJEST is the uncertainty reasoning scheme, using contribution and absence factors, which places equal importance to symptoms present, absent and unknown in the patient's medical record. Domain specific reasoning and domain specific knowledge are clearly separated from general inference capabilities and knowledge representation schemes. DIJEST has performed well in preliminary tests, being particularly impressive for patients with multiple diseases.

1. INTRODUCTION

Research in medical expert systems, a major application area of AI, has led to the development of, and experimentation with, new schemes for representing knowledge. No universal tool or technique has emerged. Each research group has its own style affected by the problem domain and the medical expertise being used.

This paper describes a new medical expert system, DIJEST (DIagnosing Jaundice Expert SysTem), which is concerned with the differential diagnosis of patients with obstructive jaundice. DIJEST evolved with its major objective to explore present knowledge representation techniques and to introduce a declarative style for modelling clinical problem solving. Subsequently another issue became critical, namely the modelling of uncertainty reasoning during the many stages of consultation and diagnosis of a disease.

DIJEST has yet another frame based scheme for knowledge representation and reasoning with uncertainty. It is developed using Prolog. We are able to combine different knowledge representation techniques in a single framework due to the flexibility of Prolog in the design of different data structures for the system. Specifically features of frame-based and rule-based representations were integrated with a new calculus for uncertainty reasoning. General medical knowledge about the domain was easily represented in Prolog declaratively. It also enabled a clear representation of inference as a specialized interpreter handling the data structures.

Our scheme for uncertainty reasoning is novel due to a strong dependence on the interpretation of present and absent data, information that is *known to exist*, *known not to exist* and information that is *unknown* at the time of consultation with the program. This was a constraint imposed by our medical experts. It allows context-dependent evaluation of the patient data. The scheme uses contribution and absence factors which are attached to particular manifestations of a disease. These factors constitute a numerical representation which complements the qualitative descriptions in DIJEST. These qualitative descriptions mirror the medical experts' definition of the characteristics of manifestations.

Our work has been influenced by the famous medical expert systems, MYCIN, Internist and PIP. The representation of knowledge using frames is similar to PIP's [1]. The concept of contribution and absence factors evolved from investigation of the confidence factors of MYCIN [2] and the evoking strengths and frequencies in Internist [3]. In addition, representing common-sense knowledge in DIJEST is affected by the representation *of properties* in Internist [4] and the use of *logical decision criteria* in PIP [5].

The basic system was designed to be expanded and enhanced to incorporate the stages of clinical reasoning during the course of a patient's treatment. DIJEST has been tested on sample patient cases taken from medical textbooks and patient records. It performs at an acceptable level

according to our experts. An interesting feature is its handling of multiple diseases contributing to the jaundice.

The paper is organized as follows. After a brief overview of DIJESTs scope, we present DIJESTs architecture and the multi-layered knowledge representation in the system. The next section describes the uncertainty reasoning mechanism which underlies the modelling of diagnosis, followed by our conclusions. We emphasize in this account how Prolog can be used to develop an expert system.

2. SCOPE OF DIJEST

2.1. The problem of obstructive jaundice

Jaundice is the yellow pigmentation of the skin or sceleras by bilirubin. This in turn is a result of elevated levels of bilirubin in the blood stream [6]. There are several reasons for this elevation.

Most of the bilirubin is derived from the catabolism of hemoglobin present in the red blood cells. The bilirubin is transformed into bile and the liver plays a central role in this metabolism of the bile pigments. The derangements of this metabolism cause several diseases which have jaundice as a common *symptom*.

The elevation of the bilirubin might be related to pathogenetic mechanisms or disease processes. We are concerned about a subset of these diseases which cause *obstructive jaundice*. This is jaundice due to the mechanical obstruction of the biliary radicles or functional factors that cause impaired hepatic excretion of bilirubin into bile.

Figure 1 is a simplified diagram of the organs that are related to the flow of bile to the intestine after its excretion from the liver. The enlargement of any organ near the bile ducts can block the flow of bile, thereby causing obstructive jaundice. The principal examples are inflammation of the gallbladder, liver or pancreas, a tumor or a cystlike mass in the head of the pancreas. Obstructive jaundice can also be caused by gallstones leaving the gallbladder, lodging in the bile ducts and blocking the flow. Diagnosing the most common causes of obstructive jaundice as mentioned above is the primary focus of DIJEST. Specifically, DIJEST considers viral hepatitis, alcoholic hepatitis, cirrhosis, cholecystitis, choledocholithiasis, pancreatitis, pancreatic cancer and pancreatic pseudocyst. Hepatitis is the inflammation of the liver. We are concerned with two types of hepatitis, one is caused by excessive consumption of alcohol and the other by virus. Cirrhosis is the chronic irreversible injury of the liver. Cholecystitis is the inflammation of the gallbladder where choledocholithiasis refers to the obstruction of the bile duct by gallstone(s). Pancreatitis is inflammation of the pancreas. Pancreatic cancer refers to a cancerous growth, while pancreatic pseudo cyst refers to the cystlike masses at the head of the pancreas.

It is critical to differentiate the mechanism that causes the obstruction and the site of the obstruction in clinical practice. Therefore, DIJEST is designed to produce possible diagnosis by

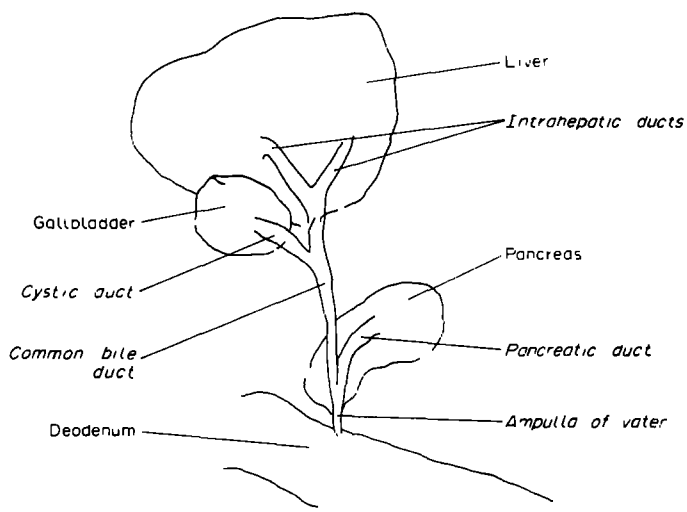

Fig. 1. The anatomy of organs participating in the bile flow.

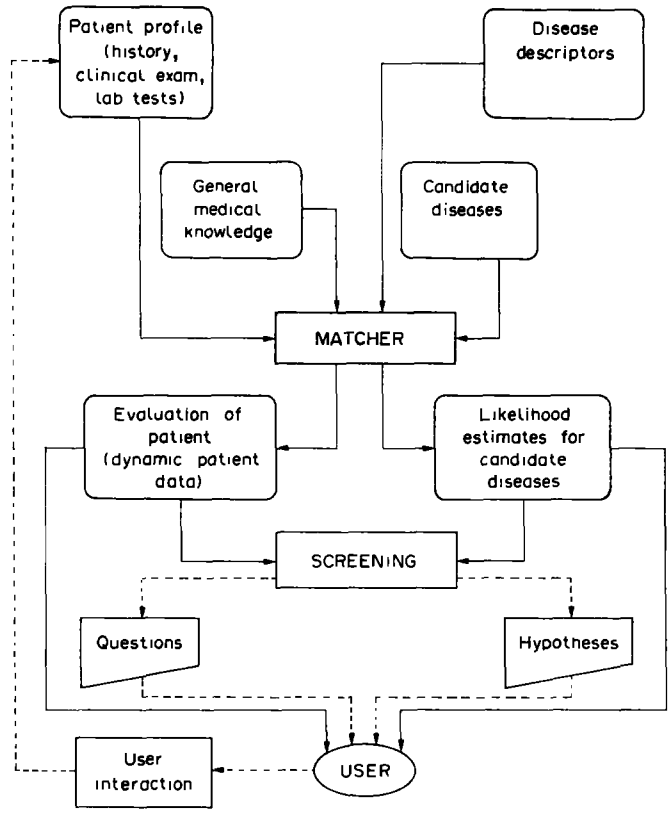

Fig. 2. The architecture of DIJEST.

indicating the likelihood of each of these diseases and the differentiating factors that leads to the diagnosis.

2.2. Architecture of DIJEST

DIJESTs system structure as initially planned is shown in Fig. 2. The system is constructed around its most important component, the specialized interpreter that we call the MATCHER. In this section we describe the function of the system components.

The boxes above the MATCHER indicate the knowledge used by DIJEST. Diseases are represented by disease descriptors. The candidate diseases are the list of diseases that are to be considered for differential diagnosis. The patient profile consists of all the knowledge related to a particular patient.

The MATCHER analyzes the patient producing an evaluation of the patient and likelihood estimates for candidate diseases. The MATCHER evaluates the current mixture of known, uncertain, partially satisfied and unknown findings of a patient with respect to the candidate diseases. The evaluation of each patient includes any contradictory evidence and suggestions for additional tests that should be performed. The results will provide feedback for the next stage of diagnosis. More details of the MATCHER are given in Section 4.

A full evaluation of the output of the MATCHER was intended to be considered by the *screening* process. Currently the patient profile is examined for the manifestations expected by the disease descriptors. The screening process would also evaluate significant patient data that is not explained by the differential diagnosis.

3. KNOWLEDGE REPRESENTATION IN DIJEST

The knowledge base of DIJEST consists of medical knowledge about jaundice and information about the patient. Knowledge of jaundice is divided into descriptions of the diseases which cause

```
disease_history(not_appl,
        disease_desc(choledocholithiasis,
        history(not_appl,       %previous illness
                not_appl,       %previous tests
                not_appl,       %exposed_to
                not_appl,       %family_background
                [               %symptoms
                (pain,[site(abdomen,5),
                                severity(none_to_severe,1),
                                continuity(intermittent,5),
                                duration(short,3),
                                coupled_by(nausea,1),
                                coupled_by(vomiting,1),
                                threshold(13)],
                                contribution_absence_factors(0.9,-0.1)),
                normalization_factor(0.9)],
                not_appl,       %observations
                not_appl,       %drug_use
                not_appl        %surgery
                ),
        clinical(
                [(jaundice,[pace(fast,2),
                                pace(medium_slow,1.3),
                                threshold(1.3)],
                                contribution_absence_factors(0.6,-2.0)),
                 (gallbladder,present,contribution_absence_factors(0.9,-2.0)),
                 (tender_abdomen,[site(upper_quadrant,5),
                                    condition(attacks,5),
                                    threshold(10)],
                                    contribution_absence_factors(0.7,-2.0)),
                 normalization_factor(2.2)]
                ),
        labtests(
                [(obstructive_tests,disease_related,contribution_absence_factors(0.7,-2.0)),
                (gallbladder, gallstones(present),contribution_absence_factors(0.8,0.2)),
                (common_bile_duct, obstruction(present),contribution_absence_factors(0.9,-2.0)),
                (common_bile_duct, dilatation(abnormal),contribution_absence_factors(0.9,-2.0)),
                normalization_factor(2.7))]
                ).
```

Fig. 3. Disease descriptor of choledocholithiasis.

jaundice, and general medical knowledge. This section describes the significant points in our representation.

3.1. Disease descriptors

The diseases that have jaundice as a common symptom are represented individually in DIJEST by *disease descriptors* or *DDs*. A DD describes the prototypical characteristics of a patient who has the disease and it is represented by a framelike structure in Prolog. Figure 3 shows the disease descriptor for choledocholithiasis.

Each disease descriptor is a quadruple indexed by the disease name. The expected characteristics related to the *history*, *clinical examination* and the *laboratory tests* are the components of this structure. We will refer to those three components as *contexts*. Each context consists of a number of *slots* that show the logical subdivision within that context. For example, the history context has eight slots showing previous diseases, previous tests, environmental and clinical factors which suggest the disease if the patient has been exposed to them, facts that are related to family

background, expected previous symptoms, expected physical observations, usage of particular drugs and previous abdominal surgery, respectively. The contexts for both clinical examination and laboratory tests have only one slot. Slots that are not applicable for a specific disease are shown by *no_appl* as illustrated by Fig. 3.

Each slot consists of specific characteristics related to the slot. They are called *elements*. They are indicated by slanted uppercase letters in Fig. 3. An element is either a *single* characteristic or a *disjunction* of characteristics.

A single characteristic is called a *key tuple* and is indexed by a *key*. A key is the smallest component of this layered structure and it can be either a *direct* key or an *extractable* key. A key tuple consists of *key attributes*, and a pair containing a *contribution factor* and an *absence factor*. The key attributes of a key tuple show the characteristics of a key which are related to the disease. They are defined with respect to the key type. The contribution and absence factors are related to the uncertainty reasoning handled by the MATCHER and will be discussed in the next section. They are represented as *contribution_absence_factors*(*CF,AF*) for clarity in Fig. 3.

The type of a key determines how the diagnosis is handled by DIJEST. The different key types are known by the MATCHER and handled differently. They are described below.

I. Direct keys. A simple concept or a finding is represented by a direct key. The key attributes of a direct key are given by a list of *qualitative* characteristics defining the key. For each attribute, a number showing the importance of this qualitative description with respect to the key is given as an integer between 1 and 5. It is called the *significance value* of the attribute. The qualitative descriptions along with significance values describe the concept or finding fully. For example, *pain* is a direct key, but its severity, duration or location differs from one disease to another as well as their relative significance for defining the pain. In Fig. 3, the respective attribute values of pain for choledocholithiasis are shown. For example, a patient is expected to have pain with intermittent continuity. Since intermittent pain is an important indicator for choledocholithiasis, it is given a significance value of 5. Every direct key is defined with a *threshold* that is used by the MATCHER for uncertainty reasoning.

II. Extractable keys. An extractable key represents a medical concept that can not be described as a simple finding. The concept needs to be extracted from the patient data. In order to simplify their representation, they are shown as a single key in the disease descriptors. They are divided into four categories to simplify the diagnosis:

(i) Some of the keys represent anatomical or physiological states or concepts. The patient is expected to be in or have such states if he is likely to have the disease. For example, *gallbladder* is a key in the laboratory tests context, and the presence of gallstones is its defining attribute as shown in Fig. 3. The value of this attribute is used as an aid for determining the site, or the exact condition for such a key. Since there may be many tests that would determine whether the patient has the specified state, this representation allows the MATCHER to determine the necessary diagnostic tests that would indicate the condition given in this key.

(ii) The names of blood tests are used as keys in DIJEST to show the tests required in the diagnosis, for example bilirubin, amylase and sgot. The analysis of the results of a particular blood test is disease dependent. The same blood test may indicate different likelihoods of the presence of a disease for different diseases. *Possibility distribution curves* are used to represent those ranges of results of blood tests. They are separate from the disease descriptors and were provided by our medical experts. How they are used with respect to the disease descriptors will be covered in the next section.

(iii) Some keys refer to a collection of simple or complex findings. The individual findings in the collection might not be very significant on their own or affect the

diagnostic process. However, their combination constitutes a medical concept and should be considered as a composite finding. We call such keys *compound* keys. The collection of individual findings are represented in separate tables for each compound key. Compound keys are represented as a combination of direct or extractable keys, but the contribution and absence factors are defined for the composite meaning. *Prodrome* is an example of a compound key which is used for diagnosing hepatitis. Figure 3 does not contain an example of a compound key.

(iv) Some keys represent rules that DIJEST has to activate in order to check the presence of a disease. They are used by the MATCHER to evaluate the patient data that are related to different contexts or to compare the tests results. For example, *inflammation* and *obstructive tests* are two rule names. The latter is shown in Fig. 3. The key attributes for this kind of key are not used since the concept to which they refer is embedded in the rules.

3.2. Patient data

The information about a patient is given as input to DIJEST. All the information related to a patient is indexed by a unique patient number. It is given in four different frames, analogous to the contexts in the disease descriptors. We refer to the information about the patient as the *patient profile*.

1. *Patient ID record.* Consists of identification information.
2. *Medical history.* This frame is similar to the history context of a DD. An example frame for a patient is shown in Fig. 4. It consists of six different slots corresponding to the first six slots in a DD. Drug and surgery information are represented as separate slots in the knowledge base if the patient has relevant data.

```
medical_history(10001,
        % previous diseases
                [(jaundice,[occurrence(negative)]),
                 (alcoholism,[occurrence(negative)])],
        % previous_tests
                [(wbc,[date(2,2,1987),site(blood),result(10200)])],
        %exposed to
                [(hepatotoxins,[exp_to(negative)]),
                 (jaundiced_people,[exp_to(negative)])],
        %family background
                [(jaundice,[occurrence(negative)])],
        %symptoms
                [(pain,[date(15,1,1987),
                        site(abdomen),
                        severity(severe),
                        continuity(intermittent),
                        duration(6, days),
                        coupled_by(nausea),
                        coupled_by(vomiting)]),
                 (nausea,[date(15,1,1987)]),
                 (vomiting,[date(15,1,1987),cause(large_dinner)]),
                 (intolerance_fatty_foods,[reaction(negative)])],
        %observations
                [(skin,[color(yellow)]),
                 (urine,[color(dark)]),
                 (stool,[color(light_brown)])])).
```

Fig. 4. Medical history for patient 10001.

Only simple findings with their related attribute values are represented in this frame. The attributes have both *quantitative* and *qualitative* descriptions. For example, the attribute *duration* for the key *pain* shows how long the patient has been in pain, and might have the value "6 days".

3. *Clinical examination.* This frame consists of two slots: the knowledge related to the actual physical examination of the patient and the results of the cardiovascular tests routinely taken. As in the history frame of the patient, the simple findings are represented as binary key tuples.
4. *Tests.* Each test frame for a patient is indexed by the test name and the patient ID number. It consists of information about the date of the test and its result with respect to the site where it is taken. For tests such as *ultrasound*, the results are given as a collection of findings related to a site, its state and its condition, because those tests are used to determine the condition of different parts of the body. For blood or urine tests, their specific site is indicated along with a single result. Examples for patient 10001 are shown below.

```
test(10001,date(2,3,1987),sgot,[(blood,80)]).

test(10001,date(2,3,1987),alk_phosp,[(blood,120)]).
test(10001,date(2,3,1987),ultrasound,[(gallbladder,edema,present),
                                      (common_bile_duct,dilations,s)
                                      (gallbladder,gallstones,present),
                                      (pancreas,swelling,normal),
                                      (pancreas_head,dilatation,normal),
                                      (pancreas,state,normal)]).
```

3.3. General medical knowledge

Medical knowledge in DIJEST is represented independently from any particular patient. Examples of such knowledge are the general characteristics of jaundice, what the available tests measure along with their possible sites, the restricted anatomy of the human body that concerns the domain diseases, the domain specific qualitative representation of quantitative terms and the possibility distribution curves of blood test results. This knowledge is used by the MATCHER when creating the differential diagnosis. For example, ultrasound is used to determine the presence of gallstones in the gallbladder or the size of the bile ducts, or sgot is a blood test and its primary function is to detect liver injury. This information is represented declaratively in DIJEST and illustrated below with a few examples.

```
lab_test(ultrasound,[(gallbladder,gallstones,[present,absent]),
                     (gallbladder,edema,[present,absent]),
                     (pancreas_head,dilation,[normal,s,inc]),
                     (extrahepatic_ducts,dilatation,[normal,s,inc]),
                     (intrahepatic_ducts,dilatation,[normal,s,inc]),
                     (liver,hepatic_texture,[homogen,nothomogen]),
                     (pancreas,swelling,[head,diffuse,normal]),
                     (pancreas,state,[atrophic,indurated,cyst,normal])]).
lab_test(uribirilogen,[(urine,excretion_bile,[present,absent,decreased,increased])]).
```

4. PATIENT EVALUATION IN DIJEST

4.1. A Prolog-based MATCHER

As its name suggests, MATCHER compares a patient profile with the disease descriptors present in the system. It is a special interpreter written in Prolog which compares the frame structures, takes into account present, absent and unknown factors and establishes likelihood scores for the presence of a disease. The findings of a disease, namely its DD, is *matched* against a patient's profile in the three different contexts of history, clinical exam and laboratory test data. A likelihood score is

calculated for each context, and the overall likelihood score for the disease is computed as the average of the scores for the three contexts.

```
diagnose(Patient,History,Clinical,Tests,disease(Disease,DiseaseProb)) ←
    disease_desc(Disease,DH,DC,DT),
    eval_history(Disease,Patient,History,Clinical,Tests,DH,HistProb),
    eval_clinical(Disease,Patient,History,Clinical,Tests,DC,ClinicalProb),
    eval_tests(Disease,Patient,History,Clinical,Tests,DT,TestsProb),
    combine_prob(HistProb,ClinicalProb,TestsProb,DiseaseProb).

combine_prob(HP,CP,TP,FinalProb) ←
    FinalProb is (HP + CP + TP)/3.0.
```

Looking at the sample disease descriptor in Fig. 3, and a patient's medical history record from Fig. 4, it should be clear that the matching is *not* a direct unification of expected values of attributes for keys for a slot in a particular context. Nevertheless, the interpreter uses unification to determine key types to handle different type of keys. Details of the MATCHER will be covered in the following sections.

The findings of a patient are evaluated with respect to a list of domain diseases, called the *candidate disease list*. The candidate disease list in the current version of DIJEST is all the known diseases that are present in the knowledge base. Heuristic rules could be added as a front-end to generate a shorter list. For example, some sets of symptoms suggest very strongly viral hepatitis and nothing else. At the moment, all the candidate diseases are processed in a straightforward manner and for each DD on the candidate disease list, a likelihood score is calculated which represents the possibility that a patient has the disease.

4.2. Calculation of likelihood scores

A *confidence measure* (CM) is calculated separately for each slot in a context. The likelihood score for the context is a weighted sum of the CM for all the slots in the context. The weighting is affected by the number of *relevant* slots in a context. The relevance of a slot is disease-dependent. For example, the family background of the patient is not relevant for choledocholithiasis as shown in Fig. 3 and it is indicated by a *not_appl* value of the slot.

The CM for a slot is calculated from the CMs of all the elements in the slot. The confidence measure for an individual slot element represents how much the patient profile satisfies the requirements of that element of the disease descriptor. The calculation of individual CMs is tied to our use of contribution and absence factors to be described below.

Recall that an element is either a single key tuple or a disjunction of them. The CM calculation of a key tuple is determined by the key type, and the requirements satisfied by the patient profile which is related to the contribution and absence factors. The CM of a disjunction of key tuples is the largest CM of one of the disjuncts.

If the MATCHER can find the manifestations defined for a key tuple that are expected to be present in a patient with a particular disease, then this key tuple is *validated*. If only some of the findings are existent, then this key is *partially validated*. When the patient profile is known not to have those findings or there is evidence against the presence of the findings, then the key is *invalidated*. The MATCHER considers the keys to be *unknown* if it can not find the related attributes from the patient profile in the case of a direct key, or extract it in the case of extractable keys. Further details about the validation process are presented after the discussion of the use of contribution and absence factors.

4.3. The role of contribution and absence factors

Contribution and absence factors are the essence of the mechanism for reasoning under uncertainty in DIJEST. A *contribution factor* (CF) and an *absence factor* (AF) are defined for each key in every key tuple of a slot in the DD. The CF determines the degree of importance of the *presence* of the specific concept represented by the key name to the slot in which it occurs. It indicates the expectation that a patient has the specific disease when the information in his/her profile validates the requirements of this key. The contribution factor is defined as a real number between 0 and 1, inclusive. For example, the contribution factor of the direct key *pain* is 0.9 for

choledocholithiasis as shown in Fig. 3. It shows that the presence of pain as defined by its respective values is very important for choledocholithiasis.

The AF determines the importance of the absence of the concept in the patient profile. It effectively measures the likelihood of a patient to have or not to have a disease given the absence of the key. It is represented on a scale of $(-\infty, 1)$. The wide scale of absence factors is used to influence the importance of a specific key to the entire slot within which it is defined. For example, the absence factor of *pain* is -0.1 as shown in Fig. 3. CF values for a key are always greater than the AF values.

The analysis to determine whether the patient has the disease depends purely on CFs and AFs. Our scheme is similar to the scoring mechanism in PIP where the scores are given in the frames [1]. The CFs and AFs are actually the quantitative representation of the qualitative terms, such as "usually present", "confirming", "critical", "more likely", "less likely" and "contradicting", that were used by our medical experts. The terms have been distributed on two different scales by using CFs and AFs.

Each application has a base value, BV, which partitions the contribution factors into two sets, those above the BV and those below it. The BV is used as a point of reference for the distribution of contribution and absence factors of the keys. For DIJEST, a BV of 0.5 was used.

Two principles underly our choice of values for contribution and absence factors from their respective scales for specific keys:

- CF $\geqslant$ BV indicates that the key is important to establish that the patient has the disease under consideration.
- AF < 0 indicates that the absence of the key is important to contradict that the patient has the disease under consideration.

Confidence measure values are classified into four categories based on these two principles:

1. $CF > BV, AF \geqslant 0$. These keys are *confirming*. A confirming key in the patient profile contributes significantly to the likelihood score. Its validation will lead to a high score. However even if the key is not validated, the disease may still figure prominently in the final differential diagnosis.
2. $CF \geqslant BV, AF < 0$. These keys are *critical*. Critical keys have the most impact on determining the likelihood score. The validation of a critical key contributes to a high score. The invalidation of a critical key contributes negatively to the score by using the AF. If a critical key is unknown, a neutral position is taken.
3. $CF < BV, AF < 0$. These keys are *contradicting*. The validation of a contradicting key does not strongly confirm the existence of the disease. The invalidation of a contradicting key can lead to a very low likelihood score.
4. $CF < BV, AF \geqslant 0$. These keys are *minor*. Minor keys are used for fine tuning the differential diagnosis and will play a greater role in the future screening process.

This classification scheme approximately corresponds to the following use of *evoking strength* and *frequency* values in Internist's scoring mechanism.

- Critical keys: *evoking strength*, 4 *frequency* 4.
- Contradicting keys: *evoking strength*, 1 *frequency* 4.
- Confirming keys: *evoking strength*, 4 *frequency* 2.
- Minor keys: *evoking strength*, 2 *frequency* 1.

Each element in a slot list is evaluated according to the above classification. The MATCHER determines how well the patient profile fits the structure that is determined for this element. Using the state of the patient profile with respect to the attributes of each element in this slot list and using the CF and AF factors, the matcher determines the CM of this element. Repeating this iterative process, all CM values of the elements in a slot list are accumulated and normalized by the unique *normalization_factor* for the slot. The overall sum of the slots determines the score of a particular context and then the likelihood score of the disease.

The matching process for the slot values is illustrated below. In the code, *Context* refers to the current name of the context, *Hypothesis* refers to the name of the disease currently investigated

and *PatientSlot* has all the values that are currently known for the *Patient* for a particular slot, such as *symptoms*. The first clause illustrates that the slots which are not applicable are not skipped over, with the assumption that they are completely satisfied for probability calculations.

```
satisfy_slots(Context,Hypothesis,Patient,PatientSlot,not_appl,1.0).
satisfy_slots(Context,Hypothesis,Patient,PatientSlot,Slot,SlotProb) ←
  Slot = = not_appl,
  satisfy_slot(Context,Hypothesis,Patient,PatientSlot,Slot,SlotProb,0).

satisfy_slot(Context,Hypothesis,Patient,PatientSlot,[normalization_factor(NF)],
                                        SlotProb,AccProb) ←
  SlotProb is AccProb/NF. % normalize for a slot
satisfy_slot(Context,Hypothesis,Patient,PatientSlot,[Key|KeyList],SlotProb,AccProb) ←
  Key = = normalization_factor(NF),
  satisfy_key(Context,Hypothesis,Patient,PatientSlot,Key,CM),
  accumulate(AccProb,CM,AccProbNext),
  satisfy_slot(Context,Hypothesis,Patient,PatientSlot,KeyList,SlotProb,AccProbNext).
```

satisfy_key determines whether a key is a single key or a disjunction of keys.

The code for processing single keys is given below. The *find* predicate extracts the values for a particular key from the patient profile.

```
handle_single_key(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  is_direct_key(Context,Key,KeyValues),
  find(Key,PatientVals,PatientSlot),
  direct_key(Context,Hypothesis,Patient,PatientVals,Key,KeyValues,CF,AF,CM).
handle_single_key(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  not is_direct_key(Context, Key, KeyValues),
  extract_from(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM).
handle_single_key(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  base_value(BV),
  not_known(Context,Hypothesis,Patient,Key,KeyValues,BV,CF,AF,CM)

direct_key(Context,Hypothesis,Patient,PatientVals,Key,KeyValues,CF,AF,CM).
  check_whether_absent(PatientVals),
  absent_key(Hypothesis,Patient,PatientVals,Key,KeyValues,CF,AF,CM),
direct_key(Context,Hypothesis,Patient,PatientVals,Key,KeyValues,CF,AF,CM) ←
  check_whether_present(PatientVals),
  match_compare(Context,Hypothesis,Patient,PatientVals,Key,KeyValues,CF,AF,CM).
```

4.4 Calculating confidence measures for keys

This subsection describes how the individual CMs are calculated for individual keytuples. Both direct keys and extractable keys are treated in detail. Our description here is qualitative in nature. The exact formulae used can be found in Ref. [7].

The confidence measure of a direct key is calculated through an extended comparison of the values in the key attribute list of the DD with the patient values as shown below. The first stage is to calculate the *patient sum*, that is a score indicating how well the patient values match the attribute values. Patient sums are only calculated for keys which actually appear in the patient profile.

```
match_compare(Context,Hypothesis,Patient,PatientVals,Key,KeyVals,CF,AF,CM) ←
  compute_patient_sum(Context,Hypothesis,Patient,PatientVals,CF,AF,
                              Key,KeyVals,0,PatientSum,ContradictionFlag),
  member(threshold(Threshold),KeyVals),
  find_normalization(KeyVals,NormFactor),
  compute_key_prob(ContradictionFlag,PatientSum,NormFactor,Threshold,CF,AF,CM).
```

The MATCHER calculates patient sums as follows. First the terms used in the patient profile, which may be a mixture of qualitative and quantitative terms such as 6 days, are converted to the

domain dependent qualitative terms which are used in the DDs, for example short or medium. The terms are then compared with the actual terms in the DD and exact matches and contradictions are noted. The terms which exactly match are summed using weights which are given in the DD with respect to each attribute. This is illustrated with the code presented below.

```
compute_patient_sum(Context,Hypothesis,Patient,PatientVals,CF,AF,
        Key,[threshold(T)],TotalSum,TotalSum,no).
compute_patient_sum(Context,Hypothesis,Patient,PatientVals,CF,AF,
        Key,[Element|KeyVals],InterSum,NextSum,ContradictionFlag) ←
  Element\ = = threshold(T),
  match(PrevContradictionFlag,Context,Hypothesis,Patient,Key,PatientVals,Element,
        InterSum,TotalSum),
  check_contradiction(PrevContradictionFlag,Context,ContradictionFlag,Hypothesis,
        CF,AF,Key,TotalSum,NextSum).

match(no,Context,Hypothesis,Patient,Key,PatientVals,Element,InterSum,AccSum) ←
  match_single_val(Hypothesis,Element,PatientVals,AttrContr),
  AccSum is InterSum + AttrContr.
match(yes,Context,Hypothesis,Patient,Key,PatientVals,Element,InterSum,AccSum) ←
  is_a_contradiction(Hypothesis,Patient,Key,PatientVals),
  record_contradiction(Context,Hypothesis,Patient,Key).

match_single_val(Hypothesis,site(Site,Contr),AllValues,Contr) ←
  member(site(Patsite),AllValues),
  appropriate_site(Hypothesis,Patsite).
match_single_val(AnyConcept,Parameter,AllValues,Contr) ←
  %generalized matching
  Parameter = ..[Name,ParVal,Contr],
  FindVal = ..[Name,SomeVal],
  member(FindVal,AllValues),
  match_from_tables(AnyConcept,Name,ParVal,SomeVal).
% Sample facts
appropriate_site(choledocholithiasis,right_upper_quadrant).
appropriate_site(choledocholithiasis,epigestrium).
match_from_tables(_,duration,DAYS,short) ←
  number(DAYS), DAYS > =1, DAYS < 11.
match_from_tables(_,duration,DAYS,moderate) ←
  number(DAYS), DAYS > 10, DAYS < 36.
match_from_tables(_,duration,DAYS,long) ←
  number(DAYS), DAYS > 35.
```

Every direct key has a *threshold*, which is the minimum value of the patient sum considered to adequately match the key. The second stage of the MATCHER is to compare the patient sum with the threshold set for this key. On the basis of this comparison, the MATCHER concludes whether the patient profile satisfies the attribute values completely, partially, or contradicts them, and calculates the CM accordingly.

If the patient sum exceeds the threshold value, then we say that the direct key has been *validated*. The CM value is this case is the CF value. For example, the attribute values of the patient in Fig. 4 indicates a sum of 13 points. This is equal to the threshold value for this key, therefore the direct key *pain* is validated for this patient. The CM is then set to 0.9.

If the patient sum is less than the threshold, and no contradiction has been noted, the key has been *partially validated*. The confidence measure is a normalized fraction of the CF value. This is handled by *compute_key_prob*. More details are in Ref. [7].

If a contradiction has been noted, the value of the CM differs depending whether the absence factor of the key is positive or negative. If the AF is negative, it is returned as the CM. Otherwise the CM is the negative of the CF value. This is handled by *check_contradiction*.

We describe each of the four categories of extractable keys in turn, where the key appears in the patient profile:

(i) Special-purpose knowledge is used to handle the anatomical or physiological states that are indexed as a key, such as common bile duct obstruction as in Fig. 3 or swelling of the pancreas. Some sample facts are given below.

```
extract_from(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  anatomy(Key),
  anatomy_test(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM).

anatomy(Key) ← organ(Key).
anatomy(Key) ← system(Key,SystemComponents).
anatomy(Key) ← system(Sys,SystemComponents),
  part_of(Key,SystemComponents).

system(intrahepatic_ducts,[left_intrahepatic_duct,right_intrahepatic_duct]).
system(extrahepatic_ducts,[common_bile_duct,cystic_duct,pancreatic_duct]).

anatomy_test(Context,Hypothesis,Patient,PatientSlot,Organ,present,CF,AF,CF) ←
  organ(Organ),
  surgery(Patient,SurgeryList),
  not taken(SurgeryList,Organ).
anatomy_test(Context,Hypothesis,Patient,PatientSlot,TestContext,
        (Specification,FacttoDetermine),CF,AF,CM) ←
  test_illustrates(TestContext,Specification,ListofTests),
  prioritize(ListofTests,FinalTests),
  patient_satisfies(Hypothesis,Patient,PatientSlot,TestContext,
        Specification,FacttoDetermine,FinalTests,CF,AF,CM).
```

For each state, the set of relevant tests is determined along with their order of preference. The representation of anatomical knowledge in DIJEST has been designed to allow the MATCHER to find the necessary tests that would indicate the presence of the specified state. For example, the MATCHER finds that ultrasound and CT tests are indicative for understanding the condition of the common bile duct when checking choledocholithiasis [7].

After the necessary tests are found, it is determined whether the patient has taken the test. If he has not, the CM for this key is calculated using the CF and AF values, and varies depending in which of the four categories the CF and AF values lie. If the patient has taken the test, domain specific knowledge is used to determine whether the patient's test results satisfy the specified state. If so, the CM is set to the CF. Otherwise, the CM is equated to AF because a conflict exists between the expected condition of the patient and the patient profile. There is no possibility to partially validate these keys. For example, the results of the ultrasound for the patient in Fig. 4 are compared with the expected outcomes for the key *common bile duct*. For the test results [7], it is found that the common bile duct of the patient is very dilated. Therefore, CM is equated to 0.9. The ultrasound also shows there are gallstones in the gallbladder. CM for this key is set to 0.8.

Planning optimal order of tests, *prioritize*, is a complicated issue, and could be the domain of another expert system that would perform in parallel to DIJEST. Currently, the tests are checked in sequential order. Studies in decision analysis for developing clinical strategies similar to the one for the diagnosis of extrahepatic obstructive jaundice can be useful for the development of this module. Especially, the sensitivity, specificity, complications and the cost of the individual tests have been investigated to devise different adaptive strategies for tests taking, represented as decision trees in Ref. [8]. We have used availability as our criteria for ordering.

(ii) Keys referring to blood tests, such as amylase and bilirubin, are evaluated using possibility distribution curves which are graphs provided to us by our experts. First the MATCHER checks whether this is a key that requires curve fitting analysis by seeing whether a patient has taken the particular test. If not, the calculation of the CM is carried out by considering the four classes of CF and AF values as for the anatomical states. If the patient has the test, the patient value is checked by a disease specific possibility distribution curve, where each curve estimates the

likelihood that a patient with the particular test value has the disease being considered. The resulting possibility value is used along with the CF and AF to determine the CM of this key. For example, if the patient's test result shows a particular positive possibility, this value is used to normalize the CF specified for this key. Normalization is needed since the importance of this test result is specified with the CF, and how well the patient's result fits the expected value for the disease is determined by the curve. If the patient's test result contradicts the presence of the disease, invalidating the key, then the full AF value is used as the CM value. Curve fitting is actually not very suitable with Prolog if speed and accuracy is required. It should be implemented as an external procedure.

```
extract_from(Context,Hypothesis,Patient,PatientSlot,Key,KeyVals,CF,AF,CM) ←
  possibility_curve(Key),
  curve_fitting(Hypothesis,Patient,PatientSlot,Key,KeyVals,CF,AF,CM).

possibility_curve(Key) ← blood_test(Key).

curve_fitting(Hypothesis,Patient,PatientSlot,Key,KeyVals,CF,AF,CM) ←
  blood_test_analysis(Hypothesis,Patient,PatientSlot,Key,CF,AF,CM).

blood_test_analysis(Hypothesis,Patient,PatientSlot,BloodTest,CF,AF,CM) ←
  (get_patient_val(BloodTest,serum,PatientSlot,Result);
  get_patient_val(BloodTest,blood,PatientSlot,Result)),
  blood_test(BloodTest,Hypothesis,Result,Prob),
  calculate_CM(Prob,Hypothesis,Patient,BloodTest,CF,AF,CM).
```

(iii) Recall that compound keys refer to a collection of findings, for example prodrome. Their analysis requires the MATCHER to consider each finding in the collection similar to the consideration of each attribute of a direct key. Each finding for compound keys, though, has to be analyzed separately similar to an element of a slot. The collected result of all the findings determines the overall CM for this key.

```
extract_from(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  concept_table(Context,Key,KeyConcepts),
  satisfy_concept(Context,Hypothesis,Patient,PatientSlot,KeyConcepts,Prob),
  concept_prob(Prob,CF,AF,CM).
```

The sum of all the confidence measures of the findings that are related to this key is denoted CM_S. CM_S is tested with respect to an interval [0,Threshold) where the value of the threshold for compound keys is application-dependent. If CM_S lies within this interval, the presence of a finding can be neither validated nor invalidated, and is considered to be unknown. If the value is to the left of this region, the finding is invalidated and the overall CM is set to the AF. Otherwise, it is considered to be fully validated and the overall CM is set to the CF. This is handled by *concept_prob*.

(iv) The rule names that are used within key tuples are evaluated by activating each rule, for example for *liver tests* and *obstructive tests*. These rules, which represent for example a group of tests, need to be evaluated considering domain specific dependencies of the tests. Each rule is interpreted separately and the CM calculation varies for each. Default behavior if the patient has not taken the test is similar to the default behavior for anatomical states and blood tests. For example, the rule *obstructive tests* in Fig. 3 is activated for the patient in Fig. 4. The values of the tests of this patient is found to be sufficient for this rule. Therefore, the CM for this key is set to the CF value, which is 0.7.

```
extract_from(Context,Hypothesis,Patient,PatientSlot,Key,KeyValues,CF,AF,CM) ←
  call_proc([Key,Context,Hypothesis,Patient,PatientSlot,KeyValues,CF,AF,CM]).

/*CALL ANY PROCEDURE PASSED AS PARAMETER*/
call_proc([ProcName|List]) ← Proc =..[ProcName|List],Proc.
```

The MATCHER has a default behavior for evaluating keys which are not covered by the above discussion, for example a direct key in the DD which does not appear in the patient profile, or a compound key for which no information is known. The CMs of these keys are determined with

respect to the four categories of CF and AF values. The crucial categories of critical keys and contradicting keys are chosen so as not to contribute to the overall sum. The confidence measure CM is calculated as follows:

```
not_known(Context,Hypothesis,Patient,Key,KeyVals,BV,CF,AF,CM) ←
  % minor keys
  CF < BV
  AF > = 0,
  CM is(CF+AF)/2.
not_known(Context,Hypothesis,Patient,Key,KeyVals,BV,CF,AF,0) ←
  % critical keys
  CF > = BV,
  AF < 0,
  record_question([Context,Hypothesis,Patient,Key,KeyVals]).
not_known(Context,Hypothesis,Patient,Key,KeyVals,BV,CF,AF,0) ←
  % contradicting keys
  AF < 0,
  CF < BV
  record_possible_contra(Context,Hypothesis,Patient,Key).
not_known(Context,Hypothesis,Patient,Key,KeyVals,BV,CF,AF,AF) ←
  % Confirming keys
  AF > = 0
  CF > = BV,
  record_unknown(Context,Hypothesis,Patient,Key).
```

For example, the exact location of the obstruction can not be determined by ultrasound for the patient in Fig. 4 [7]. This key is a critical key. Therefore, the CM value is set to 0 by the default values as described above.

4.5. Overall likelihood score

The overall likelihood score of a slot L_{Slot_i} as mentioned earlier, is the sum of the CM for each key and normalized by the specific normalization factor of the slot. The normalization factor, NF, is defined as follows where n is the number of elements in a slot. We assume that not all absence factors are zero.

$$\text{NF} = \sum_{j=1}^{n} \begin{cases} \text{AF}_j, & \text{AF} \geqslant 0, \\ \text{CF}_j, & \text{AF} < 0. \end{cases}$$

The weighted sum, WS_i, can be defined as the best case where all the elements of the slot i is validated. Thus, $\text{WS}_i = \Sigma_{j=1}^{n} \text{CF}_j$. Therefore, $\text{NF}_i \leqslant \text{WS}_i$. With this relation, the normalization helps to increase the contribution of slot i to the likelihood of the overall context. The score might be greater than 1 with data that confirms all the expected values of a slot in a disease descriptor. The overall likelihood of a context is thus defined as

$$L_C = \frac{\sum_{i=1}^{\text{NV}} L_{\text{Slot}_i}}{\text{NV}}$$

where NV equals the number of valid slots in a context.

Let us illustrate this calculation by using the example disease in Fig. 3 and the patient in Fig. 4. If the *lab_tests* slot is considered, it is seen that the *normalization_factor*, 2.7 is calculated as described above. The calculation of CMs for each of the keys in this slot is illustrated in Section 4.3. Respectively, they are 0.7, 0.8, 0 and 0.9. The sum of these CMs is 2.4. Using these values, L_{Slot} is set to 0.89. Since there is only one slot in this context, L_{lab_tests} is equal to 0.89.

4.6. The patient analysis

When the MATCHER calculates the likelihood scores of a disease, special information related to the patient with respect to each disease is recorded along with the likelihood scores. This

information is used to produce an evaluation report about the status of a patient. It consists of the list of findings which are expected but not present in the patient data, which are contradictory to the evaluated disease, and the important concepts which have not been validated during the analysis of the MATCHER. The findings of the evaluation are divided into four categories, *questions*, *contradictions*, *possible contradictions* and *unknowns*. To record this information, again the four categories of CF and AF values are used. The code in the previous section is suitably adapted.

The evaluation report can be used to guide the subsequent stages of clinical diagnosis in the screening process shown in Fig. 2. For example, the missing necessary tests to check a specific condition that have not been performed are suggested by questions for a disease. Contradictions are the set of facts in the patient profile that contradict the existence of the disease. Possible contradictions are the unknown classes of information which might be critical. They can contradict the disease if their definite absence is proven. Unknowns is the category of data that can be used for confirmation but are unknown at the time of evaluation.

5. PERFORMANCE OF DIJEST

The development time for DIJEST was about nine months including our learning about aspects of jaundice, the diseases and the related anatomy and the physiology. The knowledge representation scheme and the uncertainty reasoning mechanism reflect our perception of medical concepts and clinical reasoning provided by our experts.

DIJEST has been tested with cases taken from medical text books and real patient records. For example, Table 1 shows a differential diagnosis produced by DIJEST for a patient with choledocholithiasis. The medical history of this patient is shown by Fig. 4. During testing, the evaluation of all the domain diseases were included. In clinical use, a threshold may be used to inhibit unlikely diseases.

The analysis shows that choledocolithiasis is given the highest likelihood score by DIJEST, even though it does not get the highest score in each context. The score for acute cholecystitis shows the way large absence factors can prevent a disease from being considered seriously as explaining the jaundice. The scores from the contexts of clinical examination and lab tests strongly suggest that cholecystitis could explain the jaundice, more so than choledocolithiasis, but the patient's history strongly contradicts the disease.

The evaluation report of this patient points out for example the lack of information about critical findings of hepatitis, such as the presence of a prodrome, or the exposure to the use of needles in the past. The evaluation report is not shown here.

Later on in the course of the disease, the same patient contracted pancreatitis, directly caused by the choledocholithiasis. We added new test results to the patient profile and re-ran DIJEST. The result of the second differential diagnosis is given in Table 2. The only changed scores are of those diseases related to the pancreas. Note especially that the likelihood score of pancreatitis has significantly increased.

Table 2 demonstrates the ability of DIJEST to cope with multiple diseases. Knowledge is still necessary, for example, to realize that hepatitis and choledocholithiasis do not in general co-exist, whereas choledocholithiasis may cause pancreatitis. Such reasoning, which would form part of the screening process, allows us to place more significance on the score for pancreatitis than for hepatitis even though it is actually marginally lower.

Table 1

Likelihood Scores for Patient 10001				
Disease	History	Clinical	Tests	Total Score
choledocholithiasis	1.00	0.84	0.89	0.91
viral hepatitis	−0.05	0.99	0.81	0.58
hepatitis	−0.75	0.99	1.00	0.41
acute cholecystitis	−1.50	1.17	1.08	0.25
pancreatitis	0.28	0.20	0.21	0.23
pancr. pseudo cyst	0.16	0.00	0.13	0.10
cirrhosis	0.69	0.90	−1.50	0.03
pancreatic cancer	−0.50	0.17	−0.87	−0.40

Table 2

Likelihood Scores for Patient 10001				
Disease	History	Clinical	Tests	Total Score
choledocholithiasis	1.00	0.84	0.89	0.91
viral hepatitis	−0.05	0.99	0.81	0.58
pancreatitis	0.28	0.20	1.06	0.51
hepatitis	−0.75	0.99	1.00	0.41
acute cholecystitis	−1.50	1.17	1.08	0.25
cirrhosis	0.69	0.90	−1.50	0.03
pancr. pseudo cyst	0.16	0.00	−0.30	−0.05
pancreatic cancer	−0.50	0.17	−1.23	−0.52

DIJEST was implemented by using Prolog constructs which are standard in almost all Prologs. It currently runs under Sicstus and Quintus Prologs. In terms of speed, producing a table such as above and the evaluation report takes only a few seconds on the average.

6. CONCLUSIONS

The features of DIJEST in its current state can be summarized as follows.

Medical knowledge is represented declaratively. The domain specific knowledge and domain specific reasoning is clearly distinguished from domain independent knowledge by the MATCHER by using different types of keys. The complex medical knowledge related to the diseases, the characteristics of different testing procedures and the basic anatomical and physiological structure of the body are all represented independently of patient information and illustrate characteristics of jaundice. Using Prolog enabled us to reach our objective, to have this separation and write a specialized interpreter very easily. The interpreter has also been generalized to handle domains other than DIJEST by customizing the general matching capabilities of the interpreter.

Representing the likelihood estimates by using two separate factors, contribution and absence factors, can distinguish between valid, invalid, unknown and absent data.

DIJEST presents very realistic likelihood estimates of the presence of the candidate diseases by evaluating the patient profiles, which may be incomplete. Of special importance is the calculation of likelihood scores of the individual contexts and their effect on the final diagnosis. DIJEST also emphasises significant factors in the evaluation of each disease. Contradictory findings and important data which may be required for further evaluation of the patient are noted.

DIJEST is very promising in the early detection of co-existing diseases in a patient and provides good likelihood estimates in the cases with multiple diseases.

The most difficult task in DIJEST is to obtain the contribution and absence factors for different keys. Especially, representing the experts' qualitative view of the subject by using those factors needs successive experiments and adjustment.

A weakpoint of DIJEST is its neglect of unexplained factors that are contained in the patient profile. The presence of a screening process for presenting the results of MATCHER in a user-oriented manner and for removing redundant information would enhance the performance of DIJEST. The consistency checking is also only partially complete.

At this stage, however, DIJEST is encouraging in its expressive power for medical knowledge and by providing useful likelihood estimates to indicate the presence of domain diseases. It has potential for detecting the co-existence of multiple diseases. It is unique in both its knowledge representation scheme and reasoning with uncertainty.

Acknowledgements—We would like to thank our medical expert, Professor David Ransohoff, for providing the medical knowledge embodied in DIJEST. We are grateful for his valuable time in attending the knowledge engineering sessions and acknowledge his influence in shaping DIJEST. Drs Arnold Shmerling and Lawrence Widman also supplied valuable medical insights. Dr Len Samuels commented on an earlier draft and provided the correspondence between contribution and absence factors of DIJEST and the scoring mechanism of Internist. We also thank Yuval Lirov for inviting us to submit this paper to the special isssue of *Computers & Mathematics with Applications*.

REFERENCES

1. S. Pauker, A. Gorry, J. Kassier and W. Schwartz, Towards the simulation of clinical cognition: taking a present illness by computer. *Am. J. Med.* **60**, 981–996 (1976).
2. B. G. Buchanan and E. Shortliffe, *Rule Based Expert Systems The MYCIN Experiments of the Stanford Programming Project*. Addison-Wesley, Reading, Mass. (1984).
3. R. Miller, H. Pople and J. D. Meyers, Internist I, An experimental computer based diagnostic consultant for general internal medicine. *New Engl. J. Med.* **307**, 468–476 (1982).
4. F. E. Mesarie, R. Miller and J. D. Meyers, Internist-I properties: representing common sense and good medical practice in a computerized medical knowledge base. *Computers Biomed. Res.* **18**, 458–479 (1985).
5. P. Szolovitz and S. Pauker, Categorical and probabilistic reasoning in medical diagnosis. *Artif. Intell.* **11**, 115–144 (1978).
6. *Harrison's Principles of Internal Medicine*, 11th edn. McGraw-Hill, New York (1987).
7. L. Ü. Yalçınalp, Uncertainty reasoning in a medical expert system: DIJEST M.S. Thesis, Department of Computer Engineering and Science, Case Western Reserve University, Cleveland, Ohio, (1987).
8. J. Richter, M. Silverstein and R. Shapiro, Suspected obstructive jaundice, A decision analysis of diagnostic strategies. *Ann. Internal Med.* **99**, 46–51 (1983).