J Med Syst
DOI 10.1007/s10916-009-9413-3

ORIGINAL PAPER

A Rule-Based Clinical Decision Model to Support Interpretation of Multiple Data in Health Examinations

Kuan-Liang Kuo · Chiou-Shann Fuh

Received: 6 September 2009 / Accepted: 29 November 2009

Abstract Health examinations can obtain relatively complete health information and thus are important for the personal and public health management. For clinicians, one of the most important works in the health examinations is to interpret the health examination results. Continuously interpreting numerous health examination results of healthcare receivers is tedious and error-prone. This paper proposes a clinical decision support system to assist solving above problems. In order to customize the clinical decision support system intuitively and flexibly, this paper also proposes the rule syntax to implement computer-interpretable logic for health examinations. It is our purpose in this paper to describe the methodology of the proposed clinical decision support system. The evaluation was performed by the implementation and execution of decision rules on health examination results and a survey on clinical decision support system users. It reveals the efficiency and user satisfaction of proposed clinical decision support system. Positive impact of clinical data interpretation is also noted.

Keywords Clinical decision support system · Knowledge representation · Ontology · Health examination · Rule-based

K.-L. Kuo
Family Medicine Department, Ren-Ai Branch,
Taipei City Hospital,
10F, No. 10, Sec. 4, Ren-Ai Road,
Taipei, Taiwan 106, Republic of China
e-mail: d93009@csie.ntu.edu.tw

C.-S. Fuh (✉)
Department of Computer Science and Information Engineering,
National Taiwan University,
No. 1, Sec. 4, Roosevelt Road,
Taipei, Taiwan 10617, Republic of China
e-mail: fuh052006@yahoo.com.tw

Introduction

Health examinations and the need of clinical decision support

Health examinations obtain relatively complete health information simultaneously and are important for the personal and public health management. A health examination not only provides clinicians with useful information for possibly early diagnosis of diseases but also supplies health examination receivers the recommendation for their self-health management. Both clinicians and health examination receivers benefit greatly by the comprehensive information from health examination results.

A detailed health examination package may compose of multiple examination items such as physical examinations, laboratory tests, radiological studies, endoscopies, and others. For clinicians, one of the most important works of health examinations is generating a comprehensive report after interpreting the health examination results. The results gathered from distributed sources of tests have to be simultaneously and entirely interpreted to get the whole picture of health conditions of every health examination receiver. However, the interpretation of multiple examination items is complex and error-prone for a busy clinician.

Furthermore, continuous interpretation of numerous health examination results of healthcare receivers is tedious and monotonous for clinicians. Clinicians have to spend much effort in screening plenty of health examination results to detect relatively few clinically significant findings. The reduction of the quality of health examination result interpretations is likely at the situation of human-fatigue. Under this circumstance, clinicians tend to simplify the procedures such as following their own experience rather than clinical guidelines to interpret clinical data and make decision.

Published online: 15 December 2009

All these unwanted conditions potentially threaten the most important issue of health examinations—the quality control of health examination reports. The attempt of obtaining a practical computer-assistance for health examinations impels the development of the clinical decision support system (CDSS) proposed in this paper.

The importance of ontology

In the context of knowledge sharing, ontology is defined as a specification of a conceptualization. Ontology is a description of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general [1].

It is now widely recognized that constructing a domain model, or ontology, is an important step in the development of a knowledge-based system. The advantages include the sharing and re-use of knowledge, and the better engineering of knowledge-based system with respect to acquisition, verification, and maintenance. The role of ontology is to capture domain knowledge and provide a commonly agreed upon understanding of a domain. With the help of ontology, the knowledge is not only human-readable but also machine-readable [2].

Review of previous works

Before presenting our CDSS, this paper briefly reviews some previous works with models for representing clinical knowledge and implementing clinical practice guidelines.

Clinical practice guidelines have been developed to improve health-care quality and to control medical costs, with the goal to improve reducing inappropriate variations in clinical practice [3]. Although the importance of guidelines is already widely recognized, health care organizations typically pay more attention to guideline development than to guideline implementation for routine use in clinical settings [4], evidently hoping that clinicians will simply familiarize themselves with written guidelines and then apply them appropriately during the care of patients [5]. Studies have shown that computer-based CDSSs can improve clinician performance and patient outcomes [6]. To seamlessly apply computer and guideline to clinical practice, guideline-based CDSSs have been developed for this purpose [7–13].

All of the reviewed models contain primitives that are used to represent specific clinical tasks. Decision and action are two major representation primitives. A decision is a selection from a set of alternatives based on some predefined criteria in a guideline, for example, selection of a test from a set of potentials. An action is a clinical task or intervention that is recommended in the process of guideline application, for example, a medication or a test [5].

Arden Syntax is a rule-based model that represents guidelines as propositional logics. Medical logic modules of Arden Syntax contain production rules that relate input conditions (e.g., clinical data) to a particular set of actions. It is incomplete as a structure for representation, especially multi-step practice guidelines [14, 15].

The GLIF model [16] is an object-oriented representation that consists of a set of classes for guideline entities, attributes for those classes, and data types for the attribute values [17]. The GLIF does not fully specify the representation of guidelines at the implementation level as it is focused mainly on the description of guideline's logical structure [18].

Asbru is a time-oriented, intention-based, skeletal-plan specification language that is used to represent clinical protocols [19]. Skeletal plans capture the essence of a procedure but leave room for execution-time flexibility in the achievement of particular intentions.

EON provides a suite of models and software components for creating guideline-based applications. It views the guideline model as the core of an extensible set of models, such as a model for performing temporal abstractions. EON uses a task-based approach to define decision-support services that can be implemented using alternative techniques [20, 21].

PROforma is an executable process modeling language that has been used to build and deploy a range of decision support systems, guidelines, and other clinical applications. It is proposed for representing clinical protocols and guidelines in a machine-executable format. It combines logic programming and object-oriented modeling. PROforma supports four tasks: decisions, enquiries, actions, and compound plans [22].

The PRODIGY project's aim is to produce the simplest, most readily comprehensible model necessary to represent this class of guidelines. Teams of clinicians have used Protégé's knowledge engineering environment to encode three complex chronic disease-management guidelines [23]. PRODIGY emphasizes a scenario-based approach, in which a guideline is organized as a collection of clinical contexts. Users select contexts from relevant clinical actions [24].

The guideline community is eager to facilitate authoring of well-structured, computer-encoded guidelines that can be delivered to the point of care, and integrated into applications used by providers in the course of delivering care [25]. We have reviewed many formats or systems for representing clinical knowledge and guidelines. There has been no standardized way to represent the clinical knowledge for every circumstance.

Present work

The preliminary studies of our CDSS already have been presented by posters in 2006 American Medical Informatics

Association Spring Congress [26, 27]. An original paper that introduces and evaluates a health examination system integrated with proposed CDSS has been accepted [28]. Moreover, this paper focuses on the methodological and technical details of proposed CDSS.

Decision support systems are well recognized as highly domain-specific, customized, and interdisciplinary. The intelligence of the clinical decision support system depends on its knowledge base and reasoning algorithm. The knowledge base needs continuous maintenance to keep it up-to-date. Thus a simple maintenance mechanism of knowledge base that can be directly operated by clinical workers will make the updating job much more efficient and make the knowledge base more precise. To enhance the intelligence of the CDSS is one of the most important goals of our research. Hence, this paper proposes novel rule syntax to implement computer-interpretable logic for health examinations.

System architecture and implementation

The health examination ontology

Each knowledge base is an extension of some domain ontology, where the ontology provides a roadmap for the class of the concepts that will comprise the knowledge base. In another words, ontology provides the framework for the domain knowledge base. With the help of ontology, the knowledge is not only human-readable but also machine-readable [2]. In Fig. 1, we illustrate the health examination ontology with the entities and the relationships among the entities. Figure 2 represents the hierarchy of the health examination ontology. Its root is the health examination entity. In the clinical entity subtree, every node denotes a data source of certain type of examination. In the non-clinical entity subtree, the ontology structure shows the activities related with the health examination reports.

CDSS architecture

The design of the proposed CDSS bases on propositional logics and blackboard control architecture as well as follows the philosophy of fitting the best compromise between ideal and practice. The blackboard control architecture seems to be a very efficient tool suitable for highly sophisticated exploration of more knowledge bases in parallel [29]. The blackboard control architecture comprises three components: (a) a central blackboard or a global database, (b) a set of independent knowledge sources, and (c) a scheduling mechanism [30].

The proposed CDSS is embedded as a component of a health examination system. Under the constraints of real world environments, each design concept has to adapt to current available resources, such as hospital information system, laboratory information system, server hardware, client hardware, network bandwidth, users, and system maintenance (Fig. 3).

The CDSS is composed of a set of functional and informational units. The functional unit is divided into the reasoning engine and the connection component. The informational unit comprises the data source and the knowledge base. The knowledge base consists of decision rules, diagnosis terms, and clinical recommendation contents (Fig. 4).

The reasoning engine takes health examination result data as its data source. After the execution of the decision rules on the data source, the reasoning engine generates the output objects with the structure:

Object {
Discode (D); // diagnosis code
Sugcode (S); // suggestion code
Seccode (S'); // section code
Group (G); // group code
Rank (R); // rank code
}

Five attributes are defined in the object. *Discode* is the code of a clinical disorder. *Sugcode* is a set of codes of clinical and lifestyle-modification recommendations associated with the *Discode*. *Seccode* is the code of recommended medical department for following up the clinical disorder mentioned above. Objects with the same *Group* codes are clinically related. *Rank* always couples with *Group*. Its value is the factor of priority in the group which the object belongs to.

The blackboard control structure acts as a global database for the initial reasoning output objects. Each event of reasoning monitors the state of the blackboard for conditions under which its output is applicable. If such a condition is found to exist, the output can be posted as an increment to the solution on the blackboard [30]. Each reasoning output object is aggregated into corresponding group by its *Group* attribute. *Rank* indicates the priority of each reasoning output object in the group. While an object is put into a certain group, its *Discode*, *Seccode*, and *Sugcode* are accumulated to their corresponding set in the group for further processing after the reasoning procedure completed.

The code-style outputs from the blackboard control structure are dispatched to the representation engine where they are converted to the final human readable text immediately through the mapping to corresponding diagnosis terms and recommendation contents (Fig. 5).

The Connection component of the functional unit controls the input and output of the CDSS (Fig. 4). The input of the CDSS via the connection component is the flow of the examination results from the data source. The output is the context from the reasoning engine after

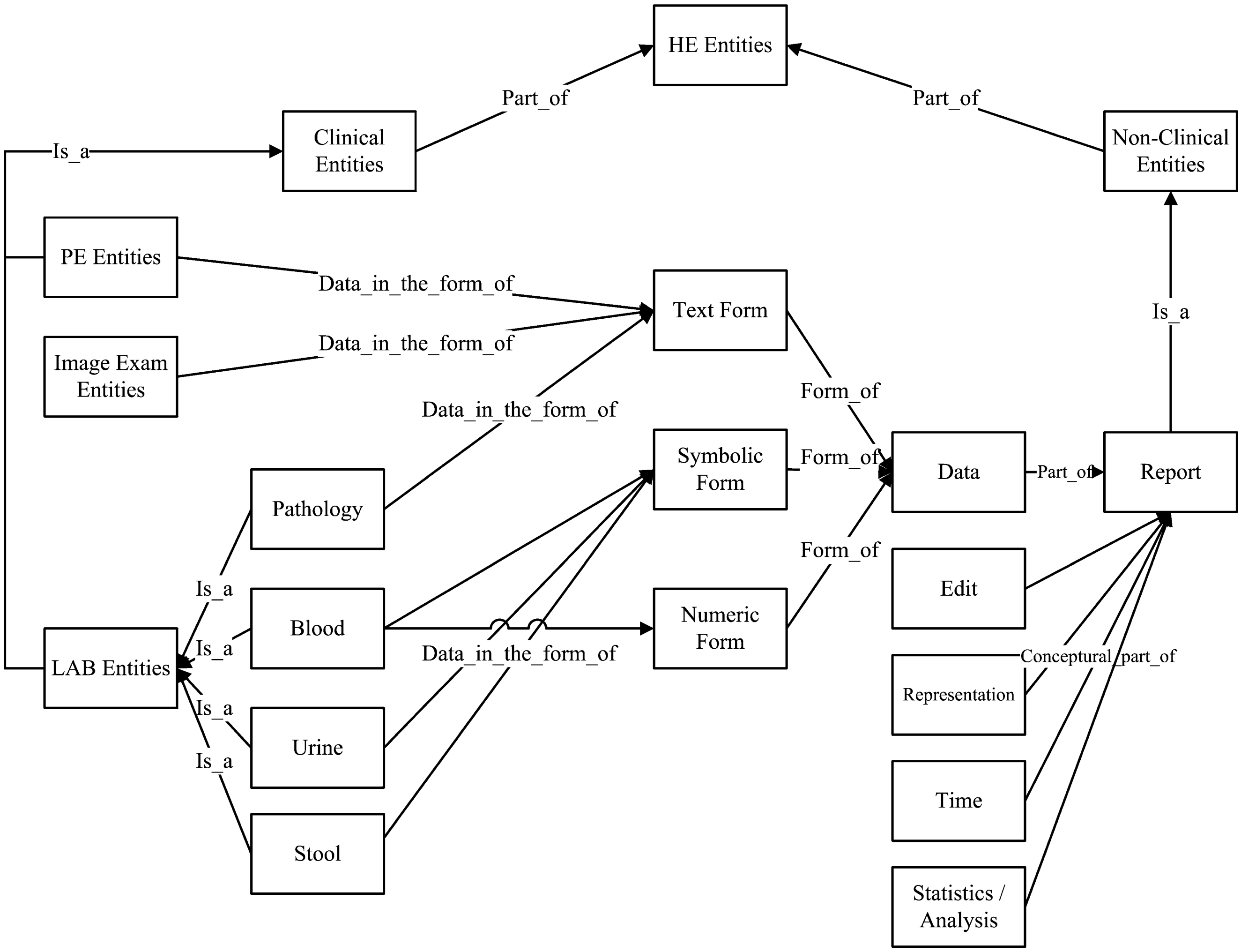


Fig. 1 Ontology for health examination (*HE* health examination, *LAB* laboratory)

automated reasoning. In the view point of software development, the connection component is isolated as a swappable module from the other components of CDSS to obtain the flexibility and reusability. Information systems of medical settings are mostly built on diverse platforms belonging to different levels of hardware, operating system, and software. The distributed environment makes the inter-system connectivity highly specified. Actually, we encountered the following circumstance: the hospital information system is a COBOL program running on a mainframe with UNIX operating system, the laboratory information system is a Visual Basic program running on an ×86 based server with Windows 2000 operating system, and the health examination system is a PHP (Personal Home Page) program running on an ×86 based server with Linux operating system. These three systems were built in different time points to solve different tasks and had their own software lifecycles. Especially, tremendous spending on developing these software systems means continuously using them is the only option. For solving the connection problem among different systems, the design concept of isolating the inter-system connectivity into the connection component makes the quarantine of "dirty codes" and acquires advantage of further system maintenance.

The data source of CDSS is the data flows consisting of clinical data of health examination receivers. Some data of health examinations gathered from bedside, for example, such as physical examination results by clinicians, are input by user key-in. Most of the health examination data come from the computer databases of Hospital Information System (HIS) and Laboratory Information System (LIS). The LIS supplies CDSS with laboratory data that are retrieved from laboratory machines (Fig. 6).

Rule syntax

The knowledge base is the intelligence-embedded component of the CDSS. Domain-specific knowledge is represented with decision rules, relation definitions among decision rules, diagnosis terms, and clinical recommendation context.

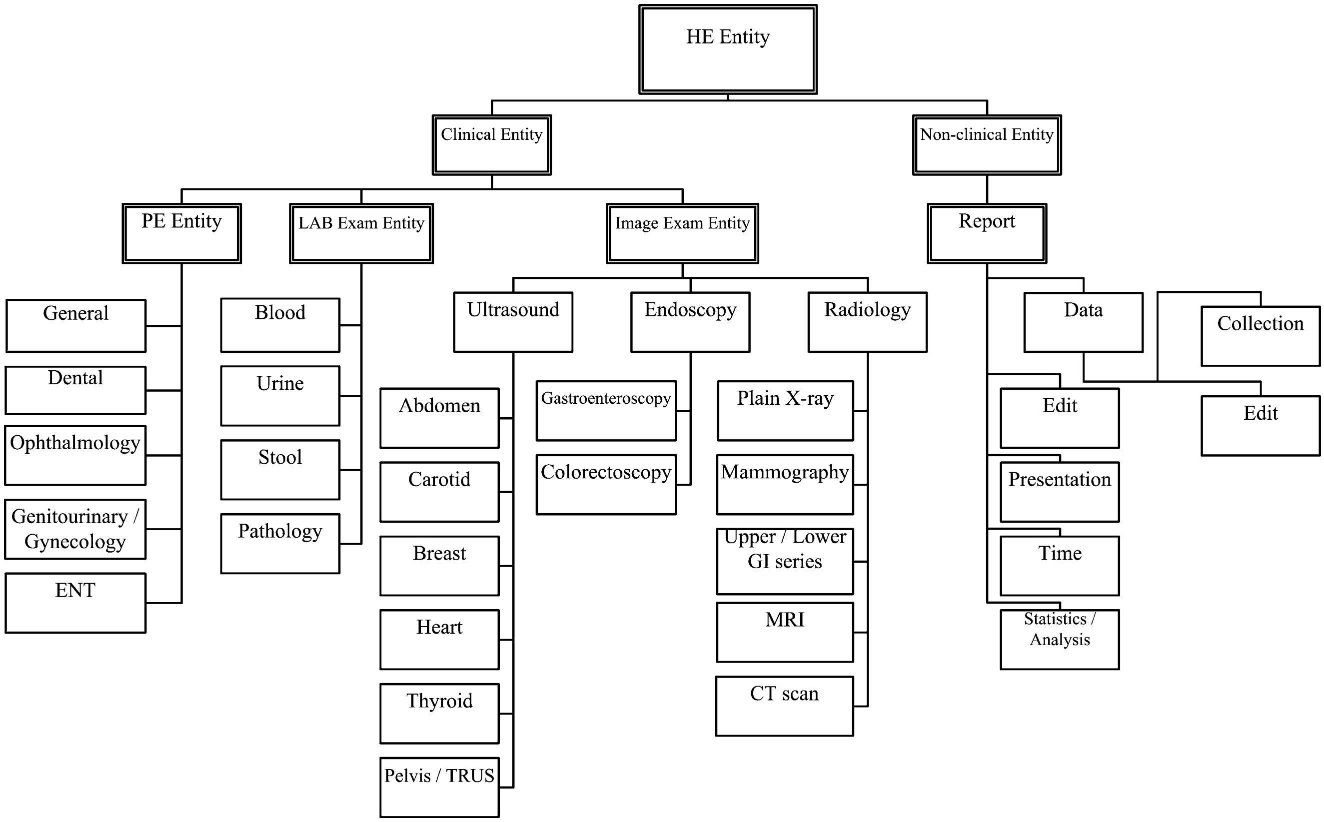


Fig. 2 The hierarchy of the health examination ontology (*HE* health examination, *LAB* laboratory)

This paper proposes a novel syntax for effectively and efficiently representing the decision rules used in the CDSS of the health examination system. To implement the philosophy of simplicity and clarity, in our design a rule is composed of three parts: a rule command, a rule name, and a sentence. Only four types of rule commands are introduced to facilitate the decision support system. Either *limitdef* or *rangedef* command declares atomic rules, which represent the constraints of variable values:

limitdef B1 VAR1 <logical operator> VALUE1
rangedef B2 VAR2 <value range>.

Commands *clausedef* and *ruledef* are used in declaring the composition of propositional logics:

clausedef B3 <disjunctions of literals>
ruledef B4 <conjunctions of literals>

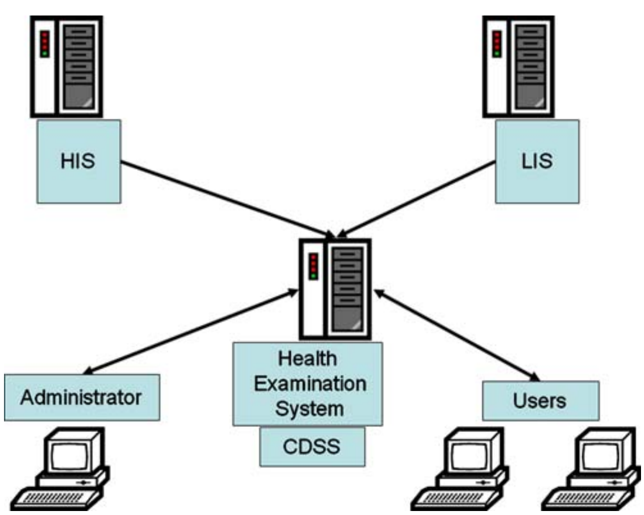


Fig. 3 Health examination system and the connections with related systems (*HIS* Hospital Information System, *LIS* Laboratory Information System, *CDSS* Clinical Decision Support System)

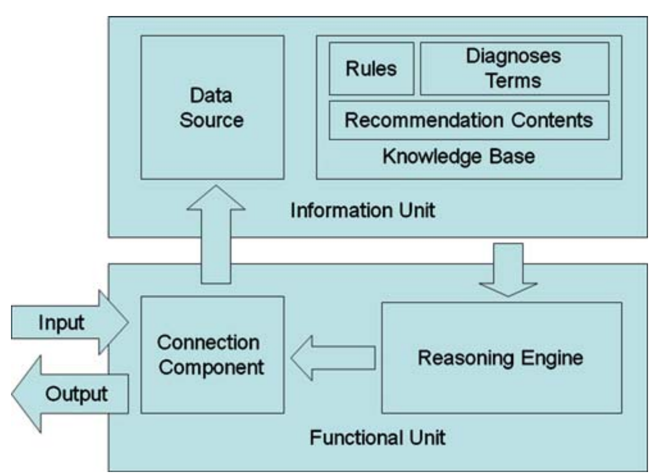


Fig. 4 The architecture of the clinical decision support system

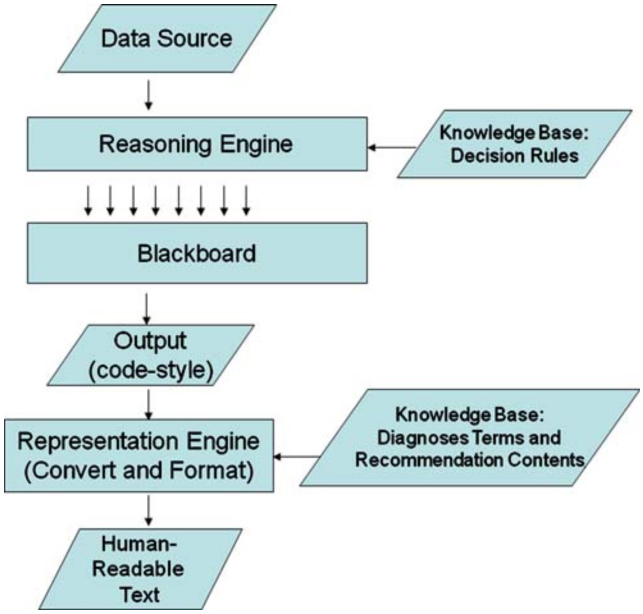


Fig. 5 The information flow of the clinical decision support system

where VAR1 and VAR2 are variables with the values of health-examination results. The rule name B1, B2, B3, and B4 are symbols with Boolean values which might be used as the literals in the following rules. Rule B4 can also trigger the final action if it is the end-rule defined in the knowledge base. The sentence in a rule declared by *limitdef* or *rangedef* contains elementary conditions. Sentence of command *clausedef* is purely a disjunction of literals; sentence of command *ruledef* is purely a conjunction of literals.

Follow the theorem that every sentence of propositional logic is logically equivalent to a conjunction of disjunctions of literals, that is, a sentence in conjunctive normal form [31]. It is trivial that combination of *clausedef* rules and *ruledef* rules can completely define propositional logics. Thus parentheses enclosure is undefined in the proposed syntax. This trade-off reduces computation and designation complexity.

System and languages used for implementation

The health examination system is a web-based system, which inherits the advantages of easy maintenance, familiar interface, and extensible access. Even though a web-based system lacks some flexibility in dealing with individual customization and control, the cost effectiveness of a web-based system still makes it the first choice. Client computers need only web browsers to connect the system and need minimal efforts on maintenance. PHP language [32] is found suitable to design a web-based application with wide technical support from the open source community. The health examination system is mainly developed with PHP language. The CDSS components are also written with PHP except some modules of the connection component are coded in Java [33]. The excellent cross-platform support for connectivity is the reason we choose Java.

For server construction, Linux operating system [34] is chosen with the advantages of robustness and high availability. Apache HTTP server [35] is enrolled since it is the best web server for Linux system. MySQL [36] is the choice of database server since its natively high compatibility with PHP.

Execution of the CDSS

The UML (Unified Modeling Language) activity flowchart of the proposed CDSS is illustrated in Fig. 7. The simulated execution of the proposed CDSS was initiated by implementing the decision rules of the decision tree for clinical decision-making. We demonstrate the implementation of the decision rules with Hepatitis B markers and anemia-related tests.

Hepatitis B markers

In Taiwan, Hepatitis B infection is an important medical and public health issues. Before early 1980, one in five persons who were infected by the hepatitis B virus (HBV) would become chronic carriers. Regular vaccination of hepatitis B at childhood was introduced in Taiwan thereafter. In health examinations, receivers would like to know their own conditions about hepatitis B. Those who have been infected by the HBV might have immunity to virus or became chronic carriers of hepatitis B. Some uninfected individuals will be recommended to receive vaccines for hepatitis B. In order to differentiate personal condition about hepatitis B and take indicated action, the clinician should consider many factors in advance. Surface antigen (HBsAg), surface antibody (Anti-HBs), and core antibody (Anti-HBc), are three major Hepatitis

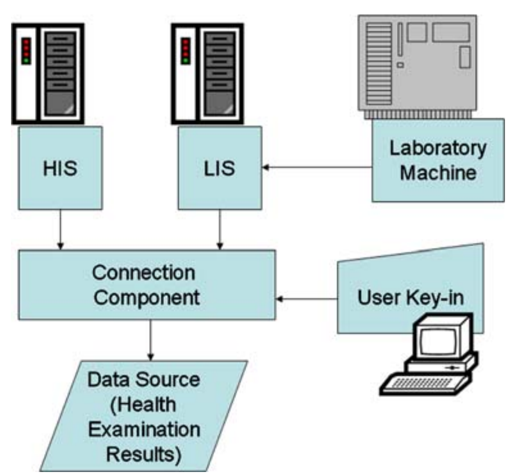


Fig. 6 The data flow of health examination results come from different data sources

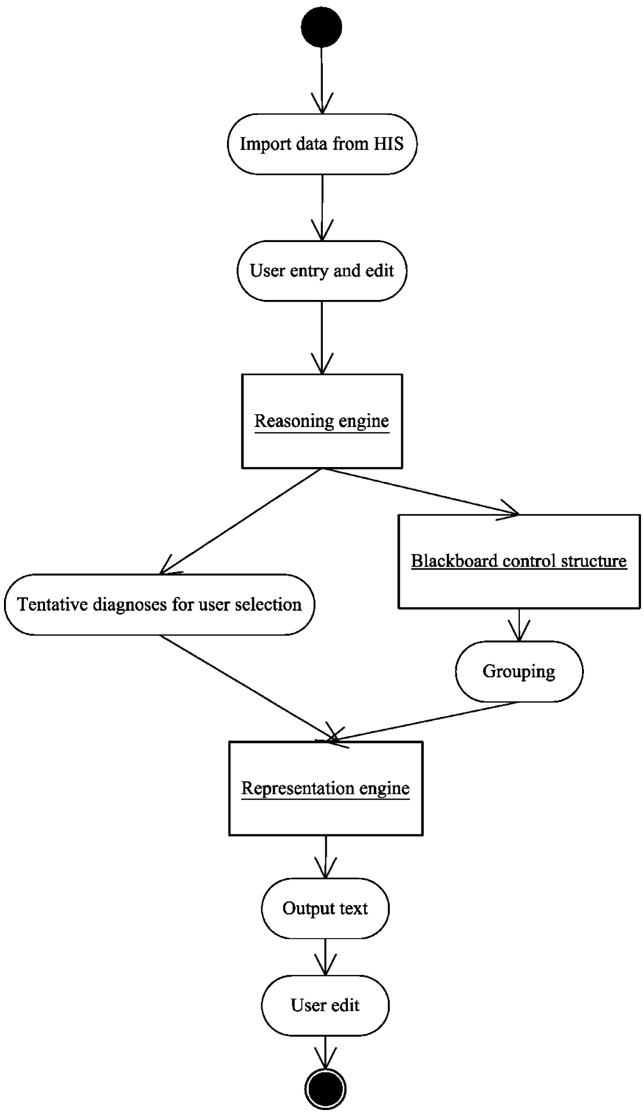


Fig. 7 The UML activity flowchart of the CDSS

B markers. Furthermore, for making a comprehensive interpretation, a clinician needs to know more information such as the values of liver enzymes and previous hepatitis B tests results of the patient.

The design of our decision rules is backtracking. In the scope of hepatitis B, we list four major possible conditions (Figs. 8, 9, 10, and 11) at first:

1. HBsAg is positive.
2. HBsAg is negative and Anti-HBs is positive.
3. HBsAg is negative; Anti-HBs is negative; and Anti-HBc is positive.
4. HBsAg is negative; Anti-HBs is negative; and Anti-HBc is negative or unknown.

In Condition 1, the diagnosis is chronic hepatitis B. Liver enzymes and alpha-fetoprotein should be taken into consideration to decide the severity of liver inflammation and the possibility of hepatic carcinoma. Furthermore, we can list four detailed conditions:

1.1 HBsAg is positive; liver enzymes are abnormal; and alpha-fetoprotein is abnormal.
1.2 HBsAg is positive; liver enzymes are abnormal; and alpha-fetoprotein is normal.
1.3 HBsAg is positive; liver enzymes are normal; and alpha-fetoprotein is abnormal.
1.4 HBsAg is positive; liver enzymes are normal; and alpha-fetoprotein is normal.

We write four rules to represent above four detailed conditions:

ruledef HBV_LFT_AFP	HBSAG_pos && LFT_ABN1 && AFP_high
ruledef HBV_LFT	HBSAG_pos && LFT_ABN1 && !AFP_high
ruledef HBV_AFP	HBSAG_pos && !LFT_ABN1 && AFP_high
ruledef HBSAG_pos_only	HBSAG_pos && !LFT_ABN1 && !AFP_high

HBV_LFT_AFP, HBV_LFT, HBV_AFP, and HBSAG_-pos_only are the names of these four rules. In each rule, the rule name is followed by one Boolean sentence which consists of one or several logic literals. Each literal should be defined before it was referred to. For example:

limitdef	HBSAG_pos	HBSAG==“+”
limitdef	AST_high	GOT>39
limitdef	ALT_high	GPT>42
clausedef	LFT_ABN1	AST_high || ALT_high
limitdef	AFP_high	AFP>13.4

The diagnoses need more consideration if HBsAg is negative. In Condition 2, the interpretation is that the case has hepatitis B immunity. Further information of Anti-HBc can reveal the source of immunity: Positive Anti-HBc means the immunity was from natural virus versus from vaccination with negative Anti-HBc. We can write three rules:

ruledef	HBV_npn	HBSAG_neg && HBSAB_pos && HBCAB_neg
ruledef	HBV_npp	HBSAG_neg && HBSAB_pos && HBCAB_pos
ruledef	HBV_np	HBSAG_neg && HBSAB_pos && HBCAB_nil

The literals are defined as:

limitdef	HBSAB_pos	HBSAB == “+”
limitdef	HBCAB_pos	HBCAB == “+”
limitdef	HBCAB_neg	HBCAB == “–”
ruledef	HBCAB_nil	!HBCAB_pos && !HBCAB_neg

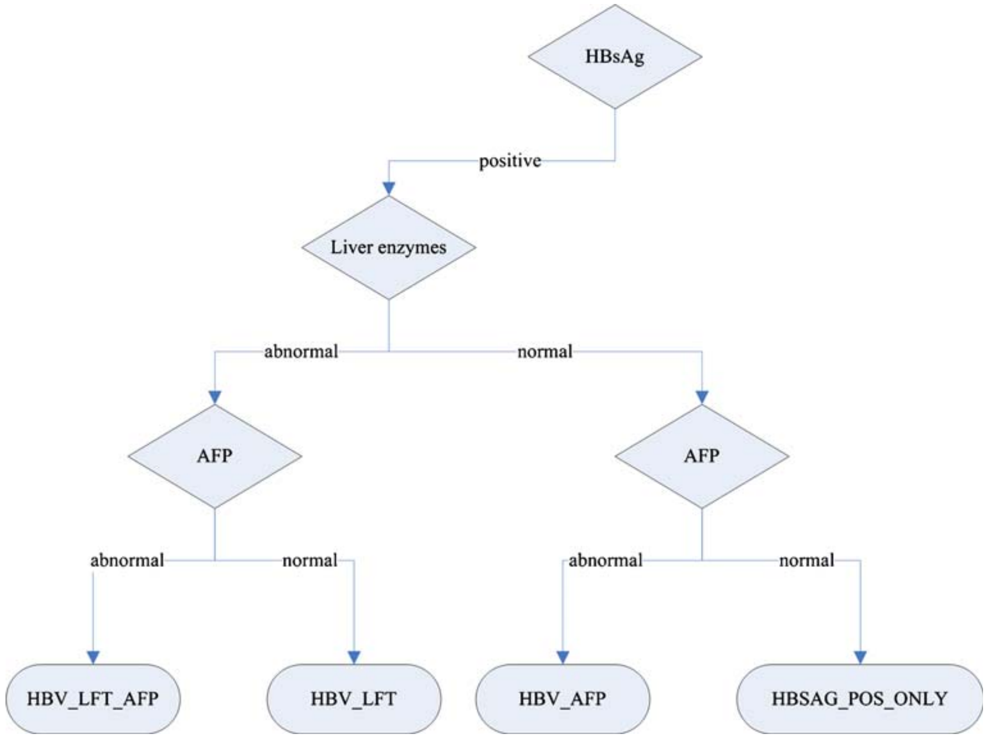


Fig. 8 Decision-making process of hepatitis B related tests. Condition 1

In Condition 3, we need additional information of previous tests of hepatitis B markers to differentiate diagnoses. If previous result of HBsAg is positive, diagnosis of chronic hepatitis B is more likely. If previous HBsAg is negative, we would like to know the result of previous Anti-HBs test. Positive previous Anti-HBs test confirms the immunity from natural virus. Negative previous Anti-HBs test makes the diagnosis still indefinite.

ruledef	HBV_nnp	HBSAG_neg && HBSAB_neg && HBCAB_pos
ruledef	HBV_nnp1	HBV_nnp && OLD_HBSAG_pos
ruledef	HBV_nnp2	HBV_nnp && OLD_HBSAG_neg && OLD_HBSAB_pos
ruledef	HBV_nnp3	HBV_nnp && OLD_HBSAG_neg && OLD_HBSAB_neg
ruledef	HBV_nnp4	HBV_nnp && OLD_HBSAG_unknown

Symbol OLD_HBSAG_pos means the HBsAg data of previous examination are positive. The decision support system will read it from database at runtime. This literal is defined as:

limitdef	OLD_HBSAG_pos	OLD_HBSAG == "+"

If HBsAg, Anti-HBs and Anti-HBc are all negative, we can confirm the status of never infected by hepatitis B virus. In Condition 4, it has two possible states that the case may not receive any vaccine or lose the immunity of vaccination. However, the antibody responses were obviously different between the young and elderly. In our consensus, hepatitis B vaccination is recommended only if

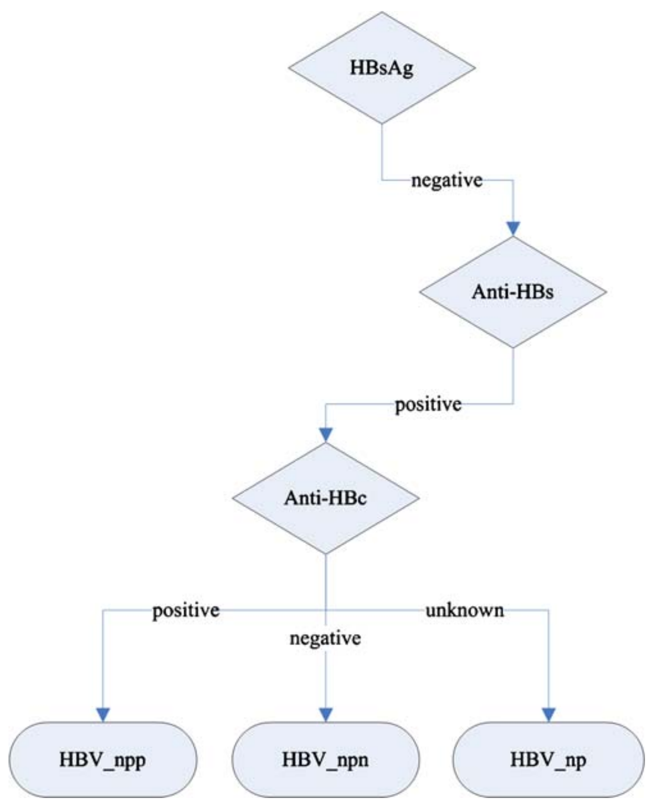


Fig. 9 Decision-making process of hepatitis B related tests. Condition 2

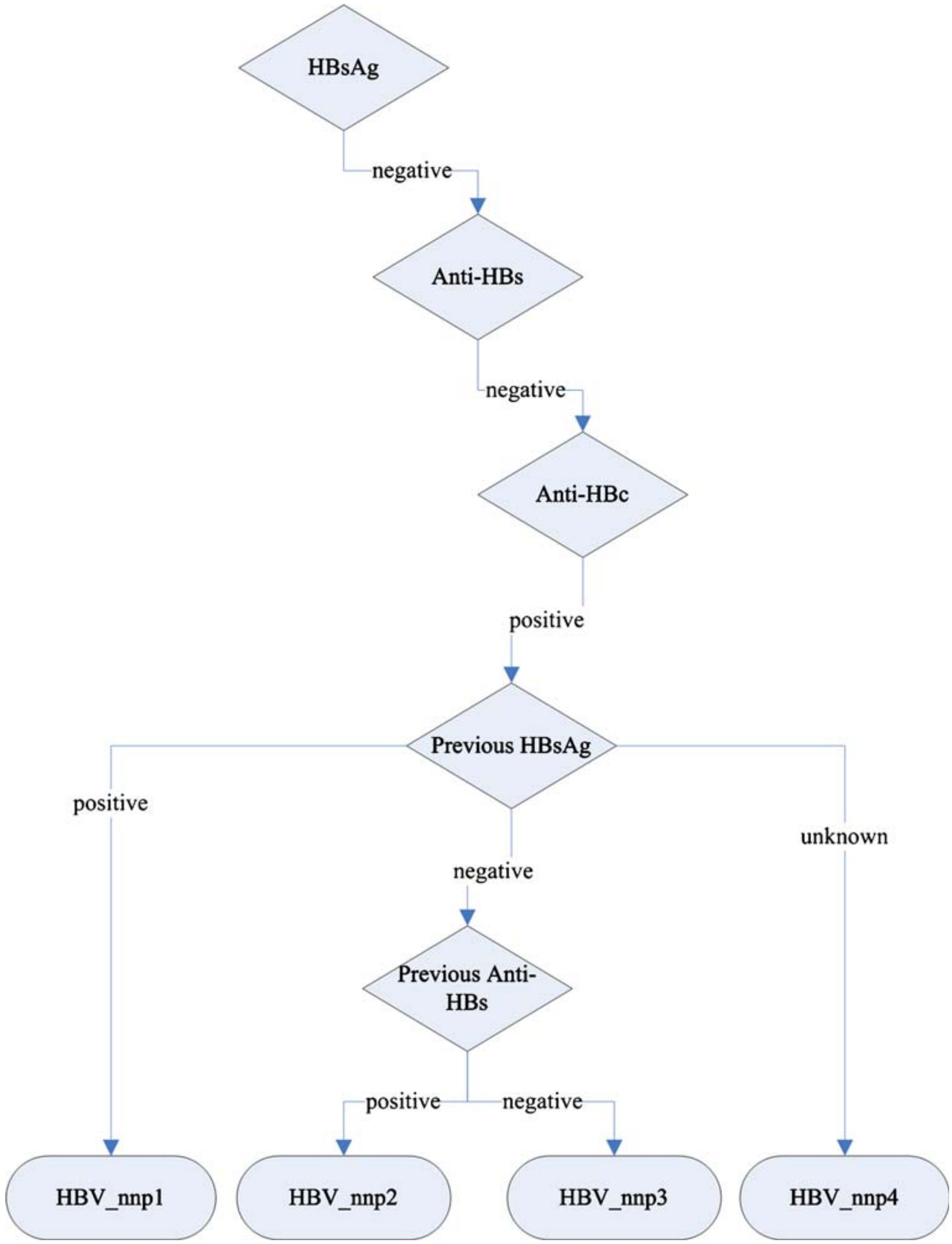


Fig. 10 Decision-making process of hepatitis B related tests. Condition 3

the patient's age is under 50 years old. We define these two endpoint rules:

ruledef	HBV_nnn	HBSAG_neg && HBSAB_neg && HBCAB_neg && !OLD_HBV_immunity && !AGE_50
ruledef	HBV_nnn2	HBSAG_neg && HBSAB_neg && HBCAB_neg && OLD_HBV_immunity

The related literal definitions are:

ruledef	OLD_HBV_immunity	OLD_HBSAG_neg && OLD_HBSAB_pos
limitdef	AGE_50	AGE >= 50

We have demonstrated that the complex process of hepatitis B interpretation can be represented by our decision rules clearly in reasonable steps. After the reasoning engine of our decision support system finishes executing the decision rules, the endpoint rules are given with Boolean values. If the Boolean value of an endpoint rule is true, the reasoning engine outputs an object which contains a set of codes. The blackboard will disassemble the output object and store the attributes allocated by the group. After the reasoning engine and the blackboard completed their jobs, the representation engine iteratively processes on the accumulative group data in the blackboard and synthesizes the human-readable text by mapping the code-style outputs to corresponding diagnosis terms and recommendation

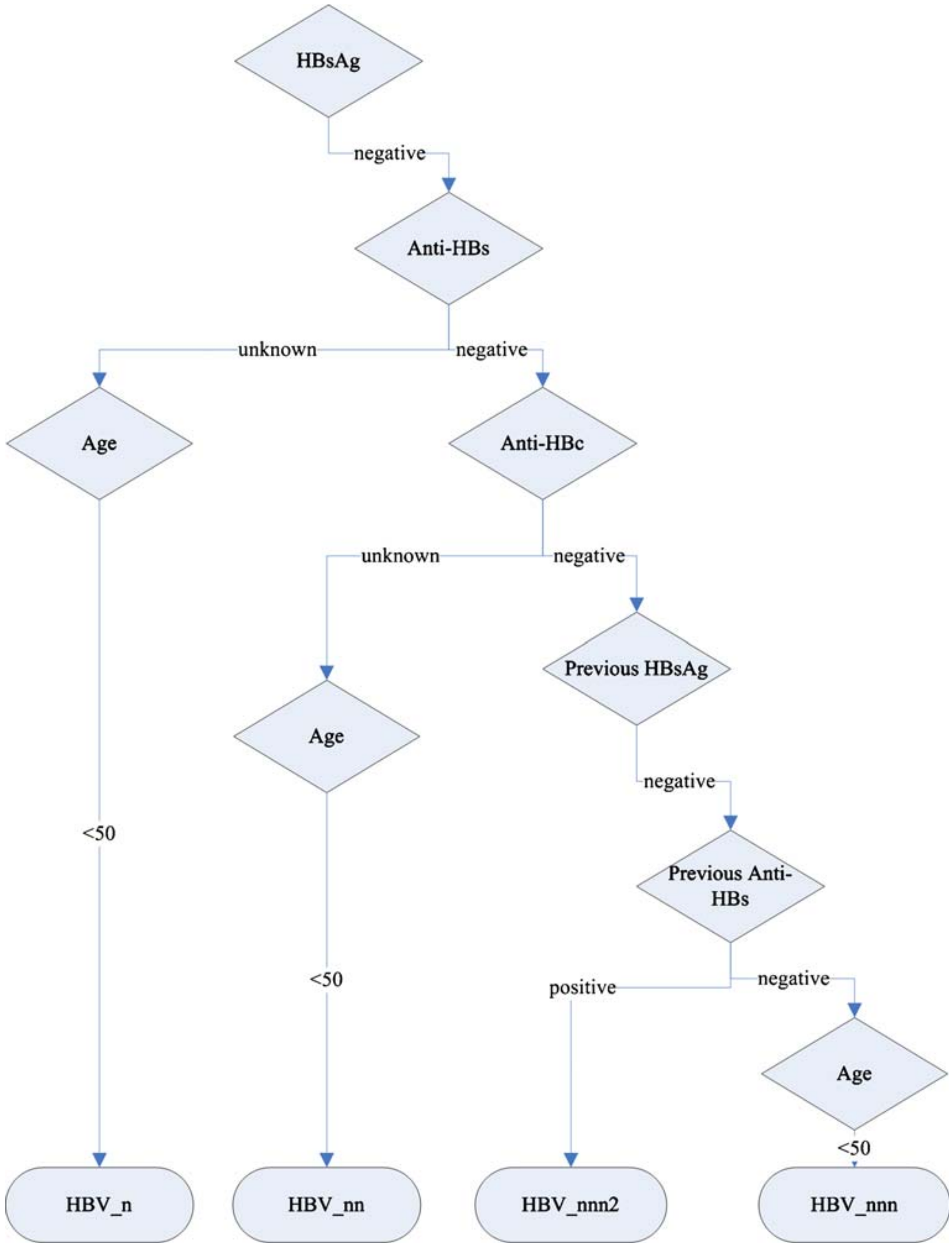


Fig. 11 Decision-making process of hepatitis B related tests. Condition 4

contents retrieved from the knowledge base. Figure 12 shows an example of how the blackboard accumulates data. In this example, T118 indicates the title string "HBsAg:−, Anti-HBs:−, Anti-HBc:+, but previous HBsAg:+". The list of codes (10, 21, 106, 707, and 708) maps to five sentences and forms the body of recommendation text. Symbols D7 and 1 map to the advice sentence of medical follow-up. Symbol G10R4 means "group 10 and rank 4" and defines the group this diagnosis belongs to and the weight. Symbol G10R5 means "group 10 and rank 5" and has higher priority than G10R4 thus the medical follow-up advice (1; D7) will be replaced with that of the one with higher priority (3; D1).

Anemia related tests

Anemia is a common disorder of the blood, specifically, RBC (Red Blood Cell). The symptom of anemia may be unobvious and non-specific such as fatigue, dizziness, poor exercise endurance, dyspnea on exertion, or even shortness of breath while resting. Thus many people suffered from anemia related blood problems are undetected until receiving laboratory tests. The causes of anemia could be excessive RBC loss or destruction as well as inadequate RBC production. The former includes gastrointestinal hemorrhage, trauma, and hemolysis. The later consists of iron deficiency, thalassemia, hereditary, substance induced, and so on.

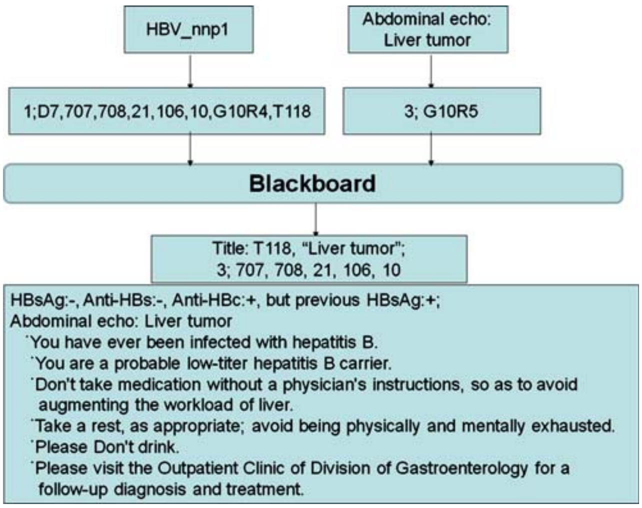


Fig. 12 The example result of the clinical decision support system. See details in the text

In a health examination, the anemia related tests such as CBC (Complete Blood Count), ferritin, and hemoglobin electrophoresis may be the routine or optional items. CBC contains RBC, MCV (Mean Corpuscular Volume), and hemoglobin. These are the most important information to diagnose anemia. Ferritin level represents the iron storage of human body. Hemoglobin electrophoresis can detect different types of hemoglobin.

We classify the hierarchical path of health examination interpretation for anemia related tests into four parts (Figs. 13, 14, 15, and 16):

1. Ferritin data is available and its level is low.
2. MCV level is low.
3. MCV level is normal.
4. MCV level is high.

In the first part of the hierarchical path (Fig. 13), if it coexists with low hemoglobin, the diagnosis is IDA (Iron Deficiency Anemia). Otherwise, if the level of hemoglobin is not low, the condition is insufficiency of body iron storage without anemia.

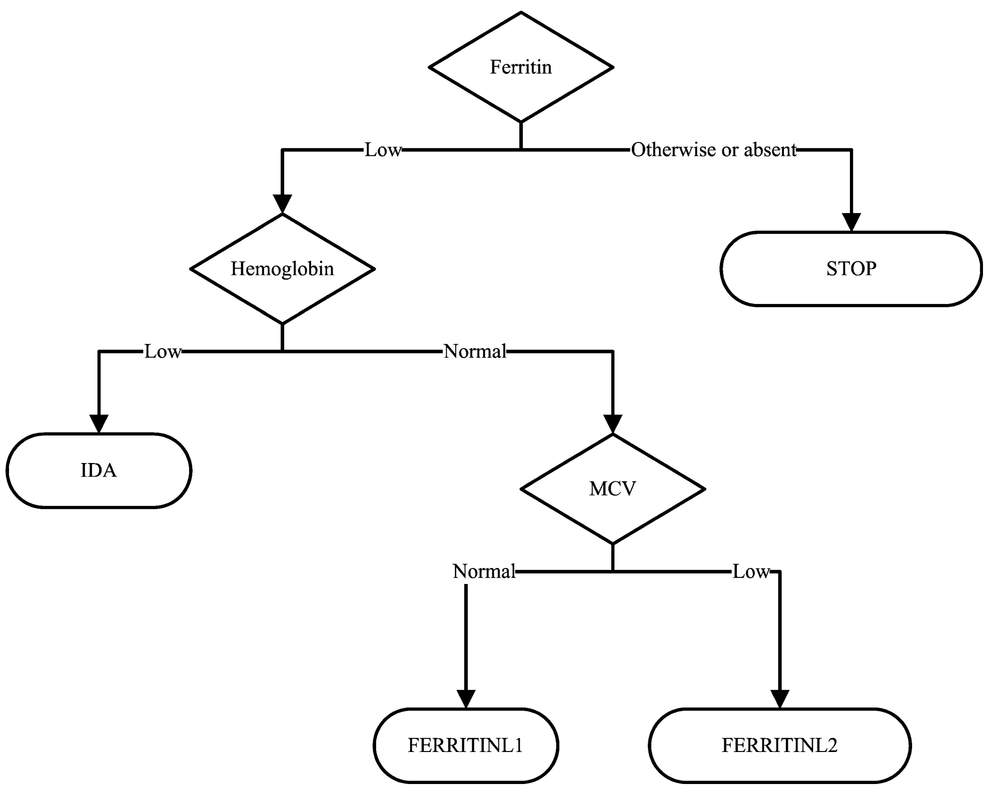
We write these rules to represent above conditions:

ruledef	IDA	HB_Low && FERRITIN_Low
ruledef	FERRITINL1	!HB_Low && FERRITIN_Low && !MCVL
ruledef	FERRITINL2	!HB_Low && FERRITIN_Low && MCVL

IDA, FERRITINL1, and FERRITINL2 are the names of these three rules. As mentioned above, each literal should be defined before it was referred to. For example:

rangedef	HB_Low	HB (0 11.3)
clausedef	FERRITIN_Low	FERRITIN_Low_Female || FERRITIN_Low_Male
ruledef	MCVL	!MCVABNL21 && MCV_Low

Fig. 13 Decision-making process of anemia related tests. Part 1



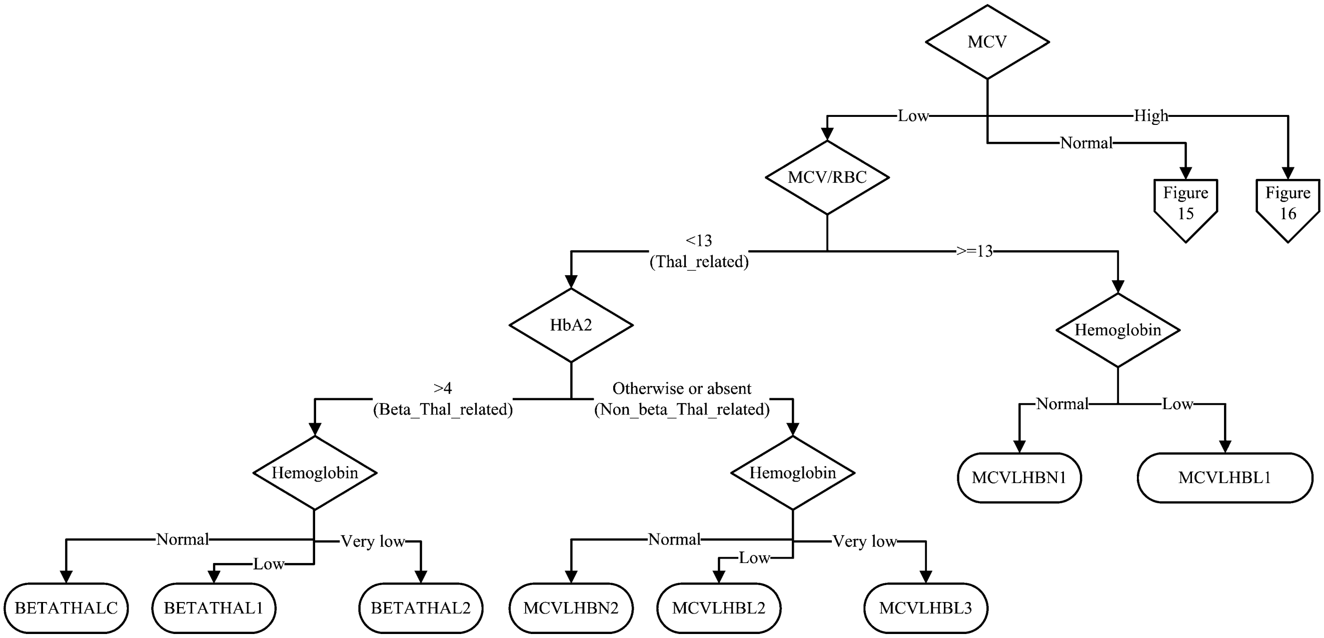


Fig. 14 Decision-making process of anemia related tests. Part 2

The second part of the hierarchical path (Fig. 14) starts with the condition of low MCV. We use the criteria MCV/RBC<13 as the cut-point of suspected thalassemia. If the data of HbA2 test is available, we can further differentiate beta-thalassemia with others. Otherwise, the diagnoses may be suspected thalassemia carrier or thalassemia according to the level of hemoglobin.

ruledef	BETATHALC	Beta_Thal_Related && HB_N
ruledef	BETATHAL1	Beta_Thal_Related && HB_Low
ruledef	BETATHAL2	Beta_Thal_Related && HB_severeLow
ruledef	MCVLHBN2	Non_Beta_Thal_Related && HB_N
ruledef	MCVLHBL2	Non_Beta_Thal_Related && HB_Low
ruledef	MCVLHBL3	Non_Beta_Thal_Related && HB_severeLow

Fig. 15 Decision-making process of anemia related tests. Part 3

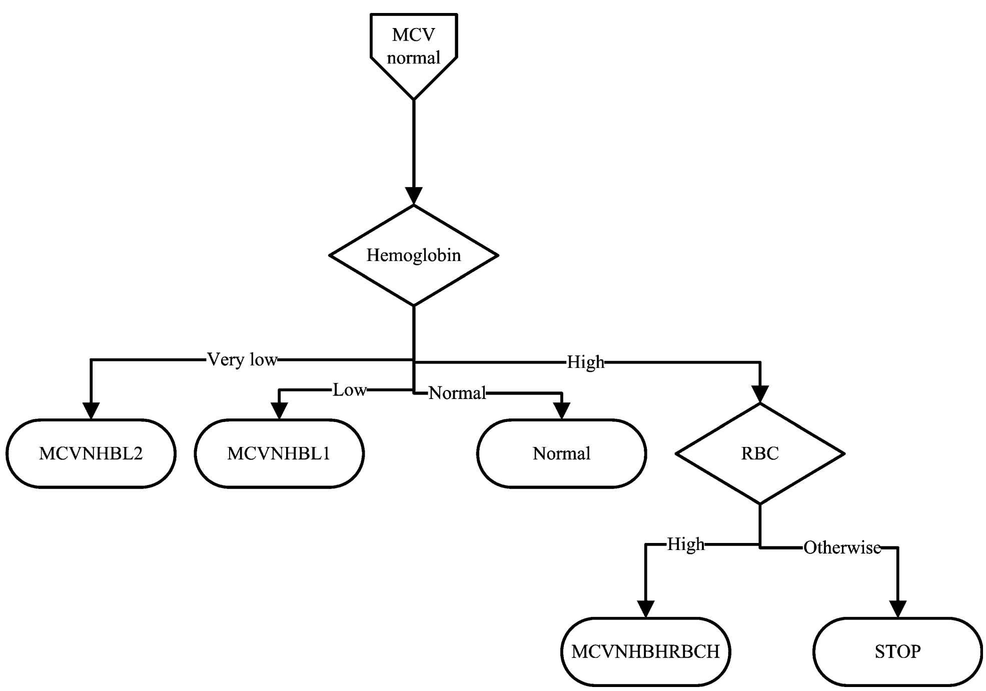
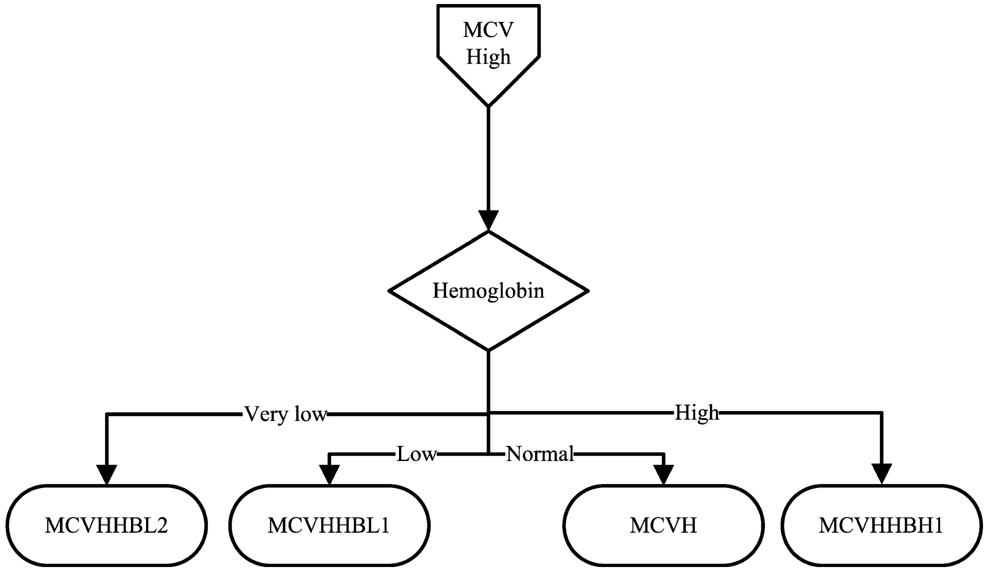


Fig. 16 Decision-making process of anemia related tests. Part 4



In the case of MCV/RBC>=13, if it comes with low level of hemoglobin, the diagnosis is microcytic anemia. If the level of hemoglobin is not low, only low MCV value is noted.

Ruledef	MCVLHBN1 MCVL && HB_N && ! MCV_DIV_RBC_Low && !FERRITIN_Low
ruledef	MCVLHBL1 MCVL && HB_Low && ! MCV_DIV_RBC_Low && !FERRITIN_Low

Some of the literals are defined as:

ruledef	Thal_Related	MCVL && MCV_DIV_RBC_Low
ruledef	Beta_Thal_Related	HBA2_High && THAL_Related
ruledef	Non_Beta_Thal_Related	!HBA2_High && THAL_Related

If the value of MCV is normal, the third part of the hierarchical path is processed (Fig. 15). The diagnosis of anemia depends on hemoglobin only if it is low. If the levels of hemoglobin and RBC are both high, polycythemia is suspected.

ruledef	MCVNHBL1	MCV_N && HB_mildLow && ! FERRITIN_Low
ruledef	MCVNHBL2	MCV_N && HB_severeLow && ! FERRITIN_Low
ruledef	MCVNHBHRBCH	!MCV_Low && HB_high && RBC_high

If the value of MCV is high, we refer to the fourth part of the hierarchical path (Fig. 16). Macrocytic anemia is noted when hemoglobin level is low. We defined these four endpoint rules:

ruledef	MCVHHBL1	MCV_high && HB_mildLow && ! FERRITIN_Low
ruledef	MCVHHBL2	MCV_high && HB_severeLow && ! FERRITIN_Low
ruledef	MCVH	MCV_high && HB_N && ! FERRITIN_Low
ruledef	MCVHHBH1	MCV_high && HB_high && ! FERRITIN_Low

Table 1 The results of the survey of user satisfaction on the clinical decision support system

Point	C1	C2	C3	C4	A1	A2	F1	F2	E1	E2	T1	T2	I1	Total	Total %
5	0	0	4	3	3	4	3	3	0	2	3	0	4	29	24.79
4	9	9	5	6	6	5	6	6	9	7	3	4	5	80	68.38
3	0	0	0	0	0	0	0	0	0	0	3	5	0	8	6.84
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00

The related literal definitions are:

limitdef	MCV_high	MCV>100
rangedef	HB_N	HB [11.3 16]
rangedef	HB_Low	HB (0 11.3)
rangedef	HB_mildLow	HB [10 11.3)
rangedef	HB_severeLow	HB (0 10)

We have demonstrated that the complex process of anemia related interpretation can also be represented by our decision rules clearly in reasonable steps. As mentioned in previous subsection, after the works of reasoning engine and the blackboard, the representation engine produces final human-readable text.

Testing and results

We tested the system with real health examination results. The health examination results consist of multi-domain clinical data. A quantified evaluation was performed simultaneously while daily system operating. Comparing the time for interpreting the data of a health examination case, automated inference time is less than one second per case in contrast to 2 to 5 min needed for physicians. The evaluation result demonstrates the efficiency of the CDSS in interpreting the health examination data. Our CDSS already performed well for more than 46,000 real health examination cases since 2005.

A survey was held among the users of HEALS. Responses were received from nine Family Medicine physicians. User satisfaction on our CDSS was evaluated using Doll and Torkzadeh's end-user computing satisfaction instrument [37]. There are 12 questions with a five-point scale. Overall, all of the physicians and assistants are satisfied with our CDSS. An additional question about the impact of clinical data interpretation ("Does the automated reasoning positively influence your interpretation on health examination results?") was used and got 100% rating above 4 points (Table 1).

Conclusions

In this paper, we propose a CDSS for the health examination information system. The most significant contribution is the decision rule syntax used by the reasoning engine of the CDSS. The requirement of the automated interpretation of health examination result data can be supported by this system. The health examination system has received high satisfaction and agreement from the staff in the Health Evaluation and Promotion Center. In reviewed studies, decision support systems can prevent errors [38] and improve patients' outcome. Our evaluation of decision support efficiency achieves significant result. The survey on user satisfaction and CDSS influence also reveals positive results.

Further developments of our CDSS will focus on the following scopes. A graphical tool for designing decision rules is helpful for users who have no programming experience. Analysis of the health examination database may retrieve useful feedback information for refining the knowledge base. Uncertainty of medical decision has not been considered in our clinical decision support system. Probability methodology and feature recognition classification [39] may be studied based on the statistical information of health examination database to create a new direction of future works.

Acknowledgement We thank the staff of the Health Evaluation and Promotion Center, Ren-Ai Branch of Taipei City Hospital for their assistance in this research.

References

1. Gruber, T. R., Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Human–Comput. Stud*. 43:907–928, 1995.
2. Huang, M. J., and Chen, M. Y., Integrated design of the intelligent web-based Chinese Medical Diagnostic System (CMDS)—systematic development for digestive health. *Expert Syst. Appl*. 32:658–673, 2007.
3. Field, M. J., and Lohr, K. N., *Guidelines for clinical practice: from development to use*. National Academy, Washington, 1992.
4. Audet, A. M., Greenfield, S., and Field, M., Medical practice guidelines: current activities and future directions. *Ann. Intern. Med*. 113:709–714, 1990.
5. Wang, D., Peleg, M., and Tu, S. W., Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *Int. J. Med Inf*. 68:59–70, 2002.
6. Johnston, M. E., Langton, K. B., and Haynes, R. B., Effects of computer-based clinical decision support systems on clinician performance and patient outcome: a critical appraisal of research. *Ann. Intern. Med*. 120:135–142, 1994.
7. Hunt, D. L., Haynes, R. B., and Hanna, S. E., Effects of computer-based clinical decision support systems on physician performance and patient outcomes a systematic review. *JAMA*. 280:1339–1346, 1998.
8. Lobach, D. F., Hammond, W. E., Development and evaluation of a Computer-Assisted Management Protocol (CAMP): improved compliance with care guidelines for diabetes mellitus. *Proceedings of Annual Symposium of Computer Application in Medical Care*. 787–791, 1994.
9. McDonald, C. J., and Overhage, J. M., Guidelines you can follow and can trust. An ideal and an example. *JAMA*. 271:872–873, 1994.
10. Osheroff, J. A., Teich, J. M., Middleton, B., *et al.*, A roadmap for national action on clinical decision support. *The Sixth Meeting of American Health Information Community*. 2006.
11. Teich, J. M., Osheroff, J. A., and Pifer, E. A., Clinical decision support in electronic prescribing: recommendations and an action

plan report of the Joint Clinical Decision Support Workgroup. *J. Am. Med. Inform. Assoc*. 12:365–376, 2005.
12. Tierney, W. M., Overhage, J. M., and Takesue, B. Y., Computerizing guidelines to improve care and patient outcomes: the example of heart failure. *J. Am. Med. Inform. Assoc*. 2:316–322, 1995.
13. Vissers, M. C., Hasman, A., and van der Linden, C. J., Impact of a protocol processing system (ProtoVIEW) on clinical behaviour of residents and treatment. *Int. J. Biomed. Comput*. 42:143–150, 1996.
14. Chu, S., Guideline representation formalism and electronic decision support systems: addressing the guideline—implementation gap. *Health Care and Informatics Review Online.* 2005.
15. de Clercq, P. A., Hasman, A., and Blom, J. A., The application of ontologies and problem-solving methods for the development of shareable guidelines. *Artif. Intell. Med*. 22:1–22, 2001.
16. Peleg, M., Tu, S., and Bury, J., Comparing computer-interpretable guideline models: a case-study approach. *J. Am. Med. Inform. Assoc*. 10:52–68, 2003.
17. Ohno-Machado, L., Gennari, J. H., and Murphy, S. N., The guideline interchange format: a model for representing guidelines. *J. Am. Med. Inform. Assoc*. 5:357–72, 1998.
18. Kolesa, P., Spidlen, J., and Zvarova, J., Obstacles to implementing an execution engine for clinical guidelines formalized in GLIF. *Stud. Health Technol. Inform*. 116:563–568, 2005.
19. Seyfang, A., Kosara, R., and Miksch, S., *Asbru's reference manual, version 7.3*. Vienna University of Technology, Institute of Software Technology, Vienna, 2002.
20. Tu, S. W., Musen, M. A., From guideline modeling to guideline execution: defining guideline-based decision-support services. *Proc. AMIA Symp.* 863–867, 2000.
21. Tu, S. W., Musen, M. A., Modeling data and knowledge in the EON guideline architecture. *Proc Medinfo*. 280–284, 2001.
22. Sutton, D., and Fox, J., The syntax and semantics of the PROforma guideline modeling language. *J. Am. Med. Inform. Assoc*. 10:433–443, 2003.
23. Purves, I., Sugden, B., Booth, N., et al., The PRODIGY project—the iterative development of the release one model. *AMIA Symposium*: *American Medical Informatics Association*. 359, 1999.
24. Peleg, M., Tu, S., and Bury, J., Comparing computer-interpretable guideline models: a case-study approach. *J. Am. Med. Inform. Assoc*. 10:52, 2003.
25. Elkin, P., Peleg, M., and Lacson, R., Toward standardization of electronic guideline representation. *MD Comput*. 17:39–44, 2000.
26. Fuh, C. S., Kuo, K. L., The decision support system used in HEALS (Health Examination Automatic Logic System). *American Medical Informatics Association Spring Congress Posters.* 2006.
27. Kuo, K. L., Fuh, C. S., HEALS: Health Examination Automatic Logic System. *American Medical Informatics Association Spring Congress Posters.* 2006.
28. Kuo, K.-L., Fuh, C.-S., A health examination system integrated with clinical decision support system. *J. Med. Syst.* (in press), 2009.
29. Lhotska, L., Marik, V., and Vlcek, T., Medical applications of enhanced rule-based expert systems. *Int. J. Med. Inform*. 63:61–75, 2001.
30. Mookerjee, V. S., and Chaturvedi, A. R., A blackboard control architecture for model selection and sequencing. *Eur. J. Inf. Syst*. 2:3–14, 1993.
31. Russell, S. J., Norvig, P., and Canny, J. F., *Artificial intelligence: a modern approach. 2nd ed*. Prentice Hall, Englewood Cliffs, 1995.
32. The PHP Group, PHP: Hypertext Preprocessor. [cited on Nov 5, 2006]. Available from: http://www.php.net/
33. Sun Microsystems, Inc., Free and Open Source Java. [cited on Nov 30, 2007]. Available from: http://www.sun.com/software/opensource/java/
34. Linux online, Inc., What is Linux? [cited on Nov 5, 2006]. Available from: http://www.linux.org/info/index.html
35. The Apache Software Foundation, The Apache HTTP Server Project. [cited on Nov 5, 2006]. Available from: http://httpd.apache.org/
36. MySQL_AB, About MySQL. [cited on Nov 30, 2007]. Available from: http://www.mysql.com
37. Doll, W., and Torkzadeh, G., The measurement of end-user computing satisfaction. *MIS Quart*. 12:259–74, 1988.
38. Balas, E. A., Information systems can prevent errors and improve quality. *J. Am. Med. Inform. Assoc*. 8:398–399, 2001.
39. Lin, Y. Y., Liu, T. L., Fuh, C. S., et al., Local ensemble kernel learning for object category recognition. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07*. 1–8 2007.