Medical Data Base Controlled By Medical Knowledge Base

Mery Y. Chernyakhovskaya, Valeriya V. Gribova, Alexander S. Kleshchev

Expert System Department
Institute for Automation & Control Processes
Far Eastern Branch of the Russian Academy of the Science
5, Radio Street, Vladivostok, 690041, Russia,
Tel: 7 4232 314001, Fax: 7 4232 310452,
E–mail: chernyah@iacp.dvo.ru, gribova@iacp.dvo.ru, kleschev@iacp.dvo.ru
page: www.iacp.dvo.ru/es/

Abstract

World practice is evidence of that computer systems for an intellectual support of medical activities bound up with examination of patients, their diagnosis, therapy and so on are the most effective means for attainment of a high level of physician's qualification. Such systems must contain large knowledge bases consistent with the modern level of science and practice. To form large knowledge bases for such systems it is necessary to have a medical ontology model reflecting contemporary notions of medicine

This paper presents a description of an observation ontology, knowledge base for the physician of general tipe, architecture, functions and implementation of problem independent shell of the system for intellectual supporting patient examination and mathematical model of the dialog. The system can be used by the following specialist: therapeutist, surgeon, gynecologist, urologist, otolaryngologist, ophthalmologist, endocrinologist, neuropathologist and immunologist. The system supports a high level of examination of patients, delivers doctors from routine work upon filling in case records and also automatically forms a computer archives of case records. The archives can be used for any statistical data processing, for producing accounts and also for debugging of knowledge bases of expert systems. Besides that, the system can be used for rise of medical education level of students, doctors in internship, staff physicians and postgraduate students.

Keywords:

Knowledge base; Ontology; Database management system

Introduction

The amount of professional medical knowledge necessary for successful practical activity of physicians was always sizeable. However, if at the beginning of the century mastery of all indispensable information was a complicated as it is but realizable task then by the end of the century the situation has been considerably changed. According to the data presented in the International Journal of Medical Practice about 4 million articles on medical subject matter are published every year. To acquaint himself at least 1% of this enormous torrent of information a doctor would have to look through more than 100 (!) articles every day. At present, there is every reason to suppose that the lag of practical physicians from the level of the last achievements of medical science has assumed a character insuperable for efforts of individual doctors.

World practice is evidence of that computer systems for an intellectual support of medical activities bound up with examination of patients, their diagnosis, therapy and so on are the most effective means for attainment of a high level of physician's qualification. Such systems must contain large knowledge bases consistent with the modern level of science and practice. To form large knowledge bases for such systems it is necessary to have a medical ontology model reflecting contemporary notions of medicine.

Now a number of different systems are known such as Emycin [1], Opal [2], Kadstool [3], Alto [4], Protege-II [5] and others that have a broad spectrum of means for knowledge acquisition and domain model forming. These means are intended for using by knowledge engineers (maybe together with experts). They have a convenient interface allowing us to describe domain knowledge in a dialog manner. Knowledge describing in such a manner either is transformed into an inner representation, or is compiled into a rule base for subsequent using by an expert system. There are also medical information systems [6,7] permitting to get information of a section of medicine necessary for a physician. But all the mentioned systems do not have any means allowing a user (a doctor, a medical student and so on) not only to carry out a convenient survey of medical knowledge of a necessary section of medicine but also firstly, to form medical data bases using this knowledge, secondly, to use the knowledge base for development of expert systems, and thirdly, to use the medical data bases for debugging of the knowledge bases of expert systems in the domain.

In this paper a computer system for an intellectual support of examination of patients is described. The system can be used by the following specialist: therapeutist, surgeon, gynecologist, urologist, otolaryngologist, ophthalmologist, endocrinologist, neuropathologist and immunologist. The system supports a high level of examination of patients, delivers doctors from routine work upon filling in case records and also automatically forms a computer archives of case records. The archives can be used for any statistical data processing, for producing accounts and also for debugging of knowledge bases of expert systems. Besides that, the system can be used for rise of medical education level of students, doctors in internship, staff physicians and postgraduate students.

Requirements to the system for intellectual support of examination of patients

An analysis of the results of a prototype version of the system and also a discussion of the results with medical specialists has allowed us to formulate basic requirements to the system. They are the following.

- The knowledge base of the system must contain a description of different sections of medicine.
- During an examination of a patient a doctor using the system must have a possibility to follow the traditional structure of case record.
- A doctor using the system must have a possibility to put down any number of examinations and of results of medical consultations of other specialist into the case record.
- A doctor using the system must have a possibility to form case record archives and to look for any necessary information in the archives.
- The system must be easily modified according to development of medical knowledge or according to requirements of a customer.

A description of an observation ontology

The description of any observation (complaints, data of anamnesis or clinical examination and so on) forms on the grounds of a medical ontology model worked out in [8]. According to the model OBSERVATIONS can be ELEMENTARY having no inner structure or COMPOUND. If an observation has a complex structure (is compound one) then its components will be called PROPERTIES. Properties can have complex structures and be described by other properties and so on.

Every elementary observation or property in their turn can be described by THE SET OF POSSIBLE VALUES (SPV) that consists of VALUES. The values can be qualitative or numerical. In the first case a list of quantitative values is a SPV description. In the second case a SPV description is an interval of numerical values. Quantitative values can be mutually incompatible or mutually compatible. All the observations can be divided into symptoms, parameters, anatomical and physiological features, events and facts. Symptoms are observations that can have values only if the patient is ill. So their descriptions always have the property called "presence". Parameters are observations that can have values if the patient is either ill or healthy. Parameters are manifestations of vital functions. Therefore, their descriptions do not have the property of presence. Symptoms and parameters can be observed at different moments so their values depend on the time. Events also may happened with a patient at different moments so their values depend on the time too. The values of anatomical and physiological features and facts do not depend on the time.

A hierarchy of elements of the observation ontology

All the elements of the knowledge structure about observations can be divided into 4 classes. They are groups of observations, observations themselves, their properties and their values. OBSERVATIONS are such elements of the hierarchy that are traditionally determined as such concepts in medicine. Some examples are pain, vomit, body temperature, blood count and so on. A GROUP OF OBSERVATIONS is a set of observations conceptually connected one with another. Groups of observations are higher in the hierarchy than observations. Another group of observation or an observation can be lower in the hierarchy than a group of observations. For example, the group of observations of cardiovascular system consists of the following groups of observations: complaints, data of clinical examination, data of laboratorial and instrumental examination and so on. The complaints (the group of observations) consists of the following observations: pain, dyspnea, arterial pressure, palpitation and so on. PROPERTIES are in the hierarchy between observations and values. For example, the observation of pain has the following properties: localization, character, intensity and so on. An (compound) observation consists of properties if all of them are directly lower than the observation. A (compound) property consists of other properties if all of these other properties are directly lower than the (first) property. For example, the property of pain localization consists of the following properties: regions of the head, regions of the neck, regions of the chest, regions of the abdomen and so on. VALUES are the lowest elements of the hierarchy. If values are directly lower than an observation then the observation is elementary one. If values are lower than a property of an observation then the observation (and the property) is compound one.

The structure of the knowledge base description

The structure of the representation of different sections of medicine corresponds to traditions established in this domain. So the knowledge base for the physician of general type contains the descriptions of observations of systems according to the specific character of this section of medicine. The description of every system contains traditional groups of observations (see below). Except this peculiarities the structures of knowledge base descriptions for physicians of any specialization are traditional. In this paper a few fragments of the knowledge base for the physician of general type will be presented.

A description of the knowledge base for the physician of general tipe

The group of observation of the knowledge base "THERAPEUTIST" consists of the following groups: respiratory system, excretory system, cardoivascular system, digestive system, case history, life history, laboratorial and instrumental examinations.

The group of observations "respiratory system" consists of the following groups of observations: complaints, data of clinical examination. The group of observations "complaints" consists of the following observations: dyspnea, cough, sputum, hemoptysis, pain in chest, rigor, body temperature, sense of fever heat, sweating, flaccidity, and so on. Below some fragments of observation descriptions are presented.

The observation of cough consists of the following properties: character, rhythm and timbre, time of appearance, duration, aggravation. The property of cough character has the following values: dry, wet nonproductive, wet productive. The property of cough rhythm and timbre has the following values: tussiculation, short, careful, accompanied by a grimace, barking, husky, soundless. The property of time of cough appearance has the following values: morning, evening, night. The property of cough duration has the following values: permanent, periodical. The property of cough aggravation has the following values: on right side, on left side, during receiving food.

The observation of sputum consists of the following properties: amount, character, colour, smell, admixtures. The property of sputum amount has qualitative values. The property of sputum character has the following values: mucoid, serous, purulent, mucopurulent, sropurulent, sanguinolent in a state of veins, in a state of clots of blood, rusty. The property of sputum colour has the following values: colourless, albicansly mucoid, greenish, yellow, brown, black, pink. The property of sputum smell has the following values: absent, musty, putrefactive, fetid, unbearably foul. The property of sputum admixtures has the following values: food, blood, fractions of tumor.

The observation of hemoptysis consists of the following properties: presence, periodicity, pronounceability. The property of hemoptysis presence has the following values: absent, present. The property of hemoptysis periodicity has the following values: permanent, often repeating, occasional. The property of hemoptysis pronounceability has the following values: considerable, moderately pronounced, inconsiderable.

The group of observations "data of clinical examination" consists of the following groups of observations: medical examination, percussion, auscultation, and also the observation of palpation. The group of observation "medical examination" consists of the following observations: cyanosis, blush, chest. The group of observation "percussion" consists of the following observations: percussion sound, borders of the lungs, mobility of inferior borders of the lungs. The group of observation "auscultation" consists of the following observations: respiration, rale, crepitation on depth of breath, pleural murmur, pleuropericardial murmur, bronchophony.

The observation of cyanosis has the following values: diffusive of face, of lips. The observation of blush has the following values: of right cheek, of left cheek. The observation of chest consists of the following properties: type, lag of movement during breathing, deformity, type of respiration. The property of chest type has the following values: normosthenic, hypersthenic, asthenic, emphysematous, paralytic, rickety, funnel-shaped, navicular. The property of lag of chest movement during breathing has the following values: of the left side of chest, of the right side of chest.

The observation of percussion sound consists of the following properties: character, side, localization. The property of percussion sound character has the following values: clear pulmanory, dulled, dull, tympanitic resonance, bandbox. The property of percussion sound side has the following values: right, left. The property of percussion sound localization has the following values: apex, superior lobe of the lung, middle lobe of the lung, lower lobe of the lung.

The observation of respiration consists of the following properties: type, localization. The property of respiration type has the following values: vesicular, physiologic weakened, vesicular physiologic hyperpnea, vesicular pathologic weakened, , vesicular pathologic hyperpnea, harsh, interrupted, bronchial, bronchovesicular. The property of respiration localization has the following values: lingula of the left lung, superior lobe of the left lung, superior lobe of the right lung, lower lobe of the left lung, lower lobe of the right lung, middle lobe of the right lung. The observation of rale consists of the following properties: character, aggravation, weakening, localization. The property of rale character consists of the following properties: dry, moist. The property of dry rale character has the following values: sibilant almost inaudible, sibilant loud, droning almost inaudible, droning loud, humming almost inaudible, humming loud. The property of moist rale character has the following values: fine bubbling sonorous, fine bubbling unsonorous, medium bubbling sonorous, medium bubbling unsonorous, bubbling sonorous, bubbling unsonorous. The property of rale aggravation has the following values: absent, on depth of breath. The property of rale weakening has the following values: absent, on depth of breath, after coughing. The values of the property of rale localization are the same as the values of the property of respiration localization.

The observation of palpation (vocal fremitus) consists of the following properties: pronounceability, localization. The property of palpation pronounceability has the following values: aggavated, weakened, absent. The property of palpation localization consists of the following properties: from behind, lateral surface.

The group of observation of case history consists of the observations of date of the beginning of the disease, of character of the beginning of the disease, of the cause of the disease, and of the groups of the first manifestations of the disease and of development of the disease before seeing a doctor.

The values of the observation of date of the beginning of the disease are dates. The observation of character of the beginning of the disease has the following values: acute, gradual. The observation of the cause of the disease has the following values: psychic trauma, overwork, cooling, poisoning, infectious diseases, unsuitable diet, trauma, other causes.

The groups of observations of the first manifestations of the disease and of development of the disease before seeing a doctor are the same as the group of complaints.

The groups of observations of life history consists of the following observations: birthplace, the age of parents by birthday, number of pregnancy as a result of that the child was born, labor, development, living conditions, diseases in the past, undergone operations, undergone traumata, hereditary history, allergologic history, bad habits, insurance history.

The observation of birthplace consists of the following properties: city, village, and region. The values of these properties are represented verbally. The observation of the age of parents by birthday consists of the following properties: of mother, of father. Their values are numbers. The values of the observation of number of pregnancy as a result of that the child was born are integers. The observation of labor has the following values: partus malurus, premature labor, delayed labor. The observation of development consists of the following properties: mental, physical. Every of these properties has the following values: corresponding with the age, outstripping, lag. The observation of living conditions has the following values: satisfactory, good, unsatisfactory. The observation of diseases in the past consists of the following properties: bodily diseases, tuberculosis, acute hepatitis, venereal diseases. The values of the property of bodily diseases are represented verbally. The observation of allergologic history has the following values: not burdened, drug allergy, food allergy, nonspecific allergy, specific allergy, pollen allergy, epidermal allergy. The observation of bad habits consists of the following properties: smoking, smoking in the past, addicted to drink, drug addict. The property of smoking has the following values: does not smoke, sometimes, smokes less than 10 cigarettes a day, smokes more than 10 cigarettes a day. The property of addicted to drink has the following values: denies, 2 times a month, weekly, a few times a week, daily.

The groups of observations of laboratorial and instrumental examinations consists of the following groups of observations: general clinic examinations, biochemical examinations, functional methods of kidneys examination, ultrasonic methods of examination, roentgenologic methods of examination, endoscopic methods of examination, instrumental methods of examination.

Shell of the system

The shell of the system for intellectual supporting patient examination is a domain independent tool. The functions of the tool are knowledge base management and data base management using the knowledge base. The knowledge base is a hierarchy of domain concepts. Data bases (including medical ones) are formed and revised in an interactive mode. To make it an user should bypass the knowledge base in an arbitrary way and select necessary concepts and values. Since the dialog is a basis of the shell let's define a mathematical model of the dialog to generate data bases and to scan and to revise them using the requirements to the system from section 2.

A mathematical model of the dialog

A pair consisting of a generative model determining a set of calculus and a regulations of user operations will be called a model of the data base editor managed by a knowledge base. Any generative model [9] consists of a language for formal representation of calculus of this model and of a universal prescription of this model. This latter is determined by the structure of generative process state, by a way of forming the initial state, by a way of building a generative process using the rules of the calculus and the initial state, and also by a rule for stopping the generative process. Any calculus represented by the language of a generative model determines a set of generative processes. When in the course of performing the universal prescription of a generative model an ambiguity related to building the generative process occurs then the regulations of user operations determines his rights for resolving the ambiguity.

The language for formal representation of calculus rules of the generative model of the dialog for managing data bases using a knowledge base (the medical knowledge base is represented also by this language) will be defined as the tuple $<N, Ts, S>$, where $N = \{n_1,...,n_m\}$ is a set of nonterminal symbols, $Ts = T \cup F$ is a terminal dictionary consisting of a set of predetermined terminals $T=\{t_1,...,t_k\}$ and of a set of types of possible values F = {string, date, time, coordinate, [1, t]I (a unit of measurement), [1, t]R (a unit of measurement)}, $S \in N$ is the chosen symbol of the language (the axiom of the generative model). Every calculus of the generative model is represented by a set P of generative rules written by the language. Every rule has the form $\alpha \to \beta$, where $\alpha \in N$, and β has the form

$$\beta = \begin{cases} \tau, & \text{where } \tau \in F \\ \tau_1, \ldots, \tau_s, & \text{where } \tau_i \in N \cup T,\ s>1 \\ \tau_1 \mid \ldots \mid \tau_s, & \text{where } \tau_i \in N \cup T,\ s>1. \end{cases}$$

As this takes place, the following conditions must be met. For every $\alpha_1, \alpha_2 \in N$ if the rules $\alpha_1 \to \beta_1$, $\alpha_2 \to \beta_2$ belong to P then $\alpha_1 \neq \alpha_2$. For every $\alpha \in N$ if α is a part of the right side of a generative rule then in P there must be a rule of the form $\alpha \to \beta$. In the set P of rules there are not such rules that $\alpha \Rightarrow^* \tau$ and $\tau \Rightarrow^* \alpha$. Here, $\alpha \Rightarrow \tau$ denotes that in P there is either a rule of the form of $\alpha \to \tau_1|\ldots|\tau|\ldots|\tau_n$ or a rule of the form $\alpha \to \tau_1, \ldots, \tau, \ldots, \tau_n$; $\alpha \Rightarrow^* \tau$ takes place if $\alpha \Rightarrow \tau_1 \Rightarrow \ldots \Rightarrow \tau_n \Rightarrow \tau$.

A flowing state of the generative process is a pair consisting of a tree D_i (it will be called a tree of data) an active node γ of the tree. The root of the tree is a node labeled by the chosen symbol S of the language. Every nonterminal node is labeled by a nonterminal symbol from N. The nodes labeled by symbols from the set either N or T or by the symbols generated by elements of the set F are the leaves of the tree.

Let's describe the universal prescription of the generative model of the dialog. Every rule can have the following ways of performing: generation, scan, and revision [10]. What rule can be applied to a flowing state depends on the way of its performing.

If the way of performing a rule is generation then the rule $\alpha \to \beta$ is applicable one to a flowing state when α is the same as the label of the active node γ of the tree and the active node is a leaf of the tree. If the way of performing a rule is revising then the rule $\alpha \to \beta$ is applicable one to a flowing state when α is the same as the label of the active node γ of the tree and the active node is not a leaf of the tree. If the way of performing a rule is scan that what rule can be applied to a flowing state depends on the way of scanning. A jump from the active node to a node which is a direct descendant of the active node in the tree of data will be called scan ahead. A jump from the active node to the node which is the direct ancestor of the active node will be called scan backwards. If the way of performing a rule is scan ahead and the label of the active node γ is $\alpha \in N$ then the rule $\alpha \to \beta$ is applicable one to a flowing state when the following conditions are met: $\beta = \tau_1, \ldots, \tau_k$, in the tree of data there is a set of edges connecting the active node γ with admissible nodes $v_i, \ldots, v_j$ and among the nodes $v_i, \ldots, v_j$ there is at least one node v_n having the label $\tau_n \in N$; $\beta = \tau_1 |\ldots| \tau_k$, in the tree of data there is an edge connecting the active node γ with a node v having the label $\tau_i \in \{\tau_1, \ldots, \tau_k\}$ and $\tau_i \in N$. If the way of performing a rule is scan backwards, α is the label of the active node γ of the tree and γ is not the root of the tree of data then any rule $\alpha' \to \beta$ is applicable one when either $\beta = \alpha$ or $\beta = \tau_1, .., \alpha, .., \tau_k$ or $\beta = \tau_1 |..|\alpha|..| \tau_k$, and in the tree of data there is a node connecting the node γ' having the label α' with the active node γ having the label α (all such nodes γ' will be called admissible nodes for the flowing state).

The first state q_0 of a generative process is the tree consisting only of the root, i.e. the node labeled by the chosen symbol S. The active node γ in the tree is the root of the tree. The result of applying a rule to a flowing state q_{i+1} is the next flowing state q_i. q_{i+1} also depends on the way of performing the rule - generation, scan or revision.

Let's the way of performing an applicable rule is generation. If a rule $\alpha \to \beta$ where $\beta \in F$ is applied then the tree D_{i+1} of data of the state q_{i+1} is formed from the tree D_i of data of the flowing state q_i by adding the edge connecting the node labeled by the symbol α with a new node v to the tree D_i. The node v have the label τ that is a generation of the symbol β. After that, the node having the label α is a nonterminal node of the tree D_{i+1}, and v is the new active and terminal node having the label τ. Generation of the label τ from the symbol $\beta \in F$ consists in the following. If β = string then a finite sequence of symbols from an alphabet is generated, if β = date then a value of the type of date is generated and so on. For example, let's a rule has the form "the last name of patient $\to$ string" then the node having the label "the last name of patient" will be connected by an edge with a new node getting the label "Petrov" (for example).

If the rule $\alpha \to \beta$ is applied where $\beta = \tau_1, .., \tau_n$ and $\tau_k \in N \cup T$ for all $k = 1, \ldots, n$ then the tree of data D_{i+1} is formed by adding a set of edges connecting the node labeled by α with a nonempty set of nodes $v_f, .., v_j$ to the tree D_i. The nodes $v_f, .., v_j$ have the labels $\tau_f, .., \tau_j$ respectively and also $\{\tau_f, .., \tau_j\} \subseteq \{\tau_1, .., \tau_n\}$ where $1 \le f < j \le n$. After that, the node labeled by α is a nonterminal node of the tree D_{i+1} and $v_f, .., v_j$ are new terminal nodes of the tree labeled by $\tau_f, .., \tau_j$ respectively. The active node is the node labeled by v_j where $1 \le j \le k$.

If the rule $\alpha \to \beta$ is applied where $\beta = \tau_1|..|\tau_n$ and $\tau_k \in N \cup T$ for all $k = 1, \ldots, n$ then the tree of data D_{i+1} is formed by adding the edge connecting the node labeled by α with a new node labeled by τ_j to the tree D_i. The label τ_j is a label from the set of labels $\{\tau_1, .., \tau_n\}$ where $1 \le j \le k$. After that, the node labeled by α is a nonterminal node of the tree D_{i+1} and v is a new terminal node of the tree labeled by τ_j. The active node is the node labeled by τ_j.

When the way of performing a rule is scan then after the application of the rule to the flowing state the both trees of data of the flowing state and of the next one are the same. If the way of scanning is scan ahead and the rule $\alpha \to \beta$ is applied where $\beta = \tau_1, .., \tau_k$ then the active node of the next state is a node v_i labeled by such $\tau_i \in \{\tau_1, .., \tau_k\}$ that τ_i is a nonterminal symbol of the language. If the rule $\alpha \to \beta$ is applied where $\beta = \tau_1|..|\tau_k$ then the active node of the next state is the node v that is the direct descendant of the previous active node if the label τ of v is a nonterminal symbol of the language. In this case the visibility scope consists of the active node labeled by α and of all its direct descendants. If the way of scanning is scan backwards then the active node of the tree of data of the next state is the direct ancestor of the previous active node. In this case the visibility scope consists of the new and previous active nodes.

Revising is removal of a node of a tree of data. If the label of the active node γ of the flowing state is $\alpha \in N$ and the rule $\alpha \to \beta$ where $\beta = \tau$ or $\beta = \tau_1|..|\tau_k$ is applied then the node v that is the direct descendant of γ, the edge connecting γ with v and also the whole subtree having v as its root are removed. After removing the tree is the tree of data of the next state. The new active node is the same γ that now is a leaf of the tree. If the rule $\alpha \to \beta$ where $\beta = \tau_1,..,\tau_k$ is applied then a set of nodes $v_f,..,v_j$ that are direct descendants of γ and are labeled by $\{\tau_f,..,\tau_j\} \subseteq \{\tau_1,..,\tau_k\}$ respectively, the set of all the edges connecting γ with $v_f,..,v_j$, and the set of all the subtrees having $v_f,..,v_j$, as their roots are removed. After removing the tree is the tree of data of the next state. The new active node is the same γ that now is a leaf of the tree.

Building a generative process can be stopped only if it reaches a terminal state. A state in which every leaf of the tree of data is labeled either by a symbol of the set T or by a symbol that is a generation of a symbol of the set F will be called terminal one.

The regulations of user operations

During building a generative process the user's task is to resolve all ambiguities. The user has a possibility to select any applicable rule in every state of the generative process. Every time when in the course of building the generative process a rule is applied a user has a possibility to select a way of performing the rule (generation, scan or revision) according to his aim.

If he selects generation then he should resolve all the ambiguities taking place in this case. They are the following.

1. If the rule $\alpha \to \beta$ is applied where $\alpha \in N$, $\beta = \tau_1 |..|\tau_n$, and also $\tau_j \in N \cup T$ for all $j = 1,...,n$ then the value $\tau_i \in \{\tau_1,..,\tau_n\}$ of the premise of the rule is determined ambiguously where $1 \le i \le n$. In this case the user has a possibility to select any value of this premise within the confines above.

2. If the rule $\alpha \to \beta$ is applied where $\alpha \in N$, $\beta = \tau_1,..,\tau_n$, and also $\tau_j \in N \cup T$ for all $j = 1,...,n$ then the value $\{\tau_i,..,\tau_j\} \subseteq \{\tau_1,..,\tau_n\}$ of the premise of the rule is determined ambiguously where $i \neq j$, $1 \le i \le n$, $1 \le j \le n$. In this case the user has a possibility to select any value of this premise within the confines above. He also can select a new active node.

3. If the rule $\alpha \to \beta$ is applied where $\alpha \in N$, $\beta \in F$ then the user can select any possible value of the premise β.

If the user selects scan then he should resolve all the ambiguities taking place in this case. They are the following.

1. The way of scan is determined ambiguously. The user can select scan ahead or scan backwards according to his aims and the conditions of applicability of rules in the flowing state.

2. A new active node is determined ambiguously. The user can select a new active node from the set of all admissible nodes.

If the user selects revision then he should select a node for removal. In any cases he also should decide when he stop the generative process.

The architecture and functions of the shell

A scheme of connections among the components of the shell is presented by fig 1. As indicated in the scheme, the shell can be used for two activities. The first one is knowledge acquisition and generation of an applied system (for example, the system for intellectual support of examination of patients). The second one is using the applied system.

Domain knowledge is represented by a language worked out on the basis of the mathematical model of the dialog presented above. The knowledge base determines a script of the dialog for the applied system. The applied system is generated by the dialog builder using the knowledge base.

The applied system has two main function. The first one is revising its knowledge base. The second one is managing the data base using the knowledge base. This last function is performed by the dialog manager. The dialog manager is an implementation of the mathematical model of the dialog described above. The manager allow a user to bypass the tree of the dialog script, to select necessary data from menu or to input them by windows having their syntactical form. The manager also checks all input data.

A dialog script is a text. Such a form of the script is convenient for reading and for its modification by any text editor. A sentence of the knowledge representation language corresponds to every rule of the mathematical model of the dialog.

The dialog builder is intended for a transformation of the knowledge base into an inner representation and for generation of the applied system for managing bat a bases according to the knowledge base. Additional functions of the dialog builder are the following: revising the knowledge base of the applied system during using the applied system according to user's desires; revising some features of the applied system according to user's requirements; forming and revising the general illumination of the applied system; forming the key of the archive of the applied system; filing a dialog script into a the dialog script library.

The dialog builder transforms the knowledge base into a tree. The nodes of the tree corresponds to the language nonterminal symbols of the knowledge base. The set of nodes is extended by a set of housekeeping nodes that are necessary for forming data base. The tree (that will be called the script tree) is a control structure for inputting data. To form the tree its nodes are displayed to the screen in the form of a table. Every line of the table corresponds to node and contains some additional information. In the script tree either a menu or a window (it depends on the right side of a rule) corresponds to every domain concept. The name of the menu or of the window is the name of the node.

The archive of the applied system is initialized by the dialog manager automatically but the developers can determine a structure of the key for the archive, form windows to input the key and to output the key that has been formed according to requirements of users. The key of the archive is an unique record which a file of the archive is connected with.

The next step of generating an applied system is designing the nodes of the script tree according to user's requirements. The dialog builder has facilities for editing the nodes of the script tree. Editing a node is changing the color and the background of a selected string, frame, text or pointer, the size of a window or menu and their position on the screen. In the mode of the field window definition it is possible also changing attributes of a field, removing a field or introducing a new field.

When a complex system having a large knowledge base is developed it is convenient to divide a script into a few modules and file these modules into the dialog script library. For example, in the system for intellectual support of examination of patients medical knowledge is divided into knowledge of particular specialists. They are therapeutist, surgeon, gynecologist, urologist, otolaryngologist, ophthalmologist, endocrinologist, neuropathologist and immunologist. Every such a part of knowledge is formed and modified together with an expert of the appropriate specialty independent of the other parts. Then these parts can be integrated into a single system using the dialog script library. Modification of any script of the library is possible using the dialog builder. Call of any script of the library is possible using the dialog manager. The key of the archive determined by the dialog builder and the value of the key can be the same for different dialog scripts. For example, if medical knowledge is divided into parties corresponding to different particular specialists then the key that is the same for different scripts may be "the last name, the first name, the second name, the birthyear of the patient". In such a manner, the archive of scripts can be formed by the dialog builder.

The dialog manager is intended for forming and revising medical data bases by users of an applied system. It executes the following functions: selection of a script from the library; interpretation of the script (data input control); scanning input data and their revising; adding new values of the keys of the archive; filing input data into the archive using the given value of the key of the archive; extracting data from the archive; printing input data in a structural form.

The dialog manager begins its run displaying the general illumination on the screen. After an answer of the user (keystroke of "enter") the window of values of the key of the archive is displayed. The window contents the menu having the name of the key of the archive and values corresponding to it and also the window for inputting new values of the key of the archive. The user has a possibility to select a value from the set of the values of the key of the archive as well as to define a new value of the key of the archive. For example, if the key of the archive is "the last name" and the values of the key are "Petrov" and "Sidorov" then the user has a possibility either to select one of the two values of the key or to define a new value (for example, "Ivanov"). Its own data in the archive correspond to every value of the key. When the user selects a value of the key he extracts data from the archive corresponding to the value of the key. All data corresponding to a value of the key of the archive are displayed on the screen and then the user has a possibility to revise or to scan them and also to input new data.

If the user defines a new value of the key of the archive then the next stage is data input. Data input is their selecting from menus or inputting values by the windows that were determined by the dialog builder. When the user selects a node of the tree the image of a menu or a window corresponding to the node appears on the screen. A help window can be connected with every node of the tree. Every element of a menu corresponds to a descendant of the node that corresponds to the menu. If a terminal window has an input field then input of data using the keyboard is waited. The input value is checked according to the type of the field and the interval of values. If either the type or the value itself is wrong then a signal sounds and also the type and the interval of the field are displayed on the screen. If the check is successful then data are input from the next field. When data are input from the latest field the next node will be the ancestor of the flowing node. The input data can be scanned and revised during the input as well as its completing.

When the user selects scanning input data a structural menu containing values input by the user appears on the screen. The user has a possibility to scan all the values related to the flowing and lower node of the script tree. During scanning input values can be revised and removed. The input values can be scanned in an other order that is more convenient for reading without rescanning nodes for every value. But in this case there is no possibility to revise any values. Input data can be printed.

Implementation of the shell

The shell is implemented by a relational database management system. Database management system is chosen as the tool for implementation of the shell because database management systems have facilities for storage, search and retrieval of data as well as for implementing screen forms.

The dialog builder has the knowledge representation language processor. A dialog script described in this language is transformed by the language processor into an inner representation. In this representation the script is a table. When a user designs an applied system by the dialog builder some additional tables containing data about all the menus and windows are formed. All the tables are files. Dividing the knowledge base into several modules (scripts) makes more convenient using the applied system and easier searching, scanning and revising data. The archive formed by the dialog manager also is represented by two tables. They are the table of keys of the archive and the table of data. The interpretator of scripts as a part of the dialog manager interprets the table of data containing sequences of node numbers by the table of scripts to display input data on the screen.

A technology of developing the system for intellectual support of examination of patients by the shell

To develop the system for intellectual support of examination of patients by the shell it was necessary: to form a medical knowledge base; to represent the knowledge base in the knowledge representation language; to generate the applied system.

When the goals of developing an applied system are determined, there are all the participants of the project, and all necessary resources for its development are allotted, an analysis of the domain should be made to work out a mathematical model of the domain. In this case an expert and a knowledge engineer should identify all the concepts related to observations of the domain and hierarchy of these concepts. Such work was made when the system for intellectual support of examination of patients was developed and its results were represented as a text.

To represent this knowledge in the knowledge representation language using the text any text editor can be used. A knowledge engineer makes a list of domain observations, their properties, their groups and so on. Then a set of descriptors is added to every concept. The set contains a type of the rule corresponding to the description, the bounds of the intervals of all the quantitative properties and so on.

At last, the dialog builder generates the applied system. To call the dialog builder it is necessary to point to the name of a dialog script and the mode of operation (generation or revision of an applied system).

If the mode of operation is generation of an applied system then developers should determine values of tags for every node of the table. Every node of the table corresponds to either a domain concept, or its property, or a housekeeping node. The values of some tags of nodes are assigned by the dialog builder according to the dialog script (for example, the value of compatibility of a menu), the values of the others are assigned by developers (for example, for output of the number of the menu string of the flowing node). If the values of the tags assigned by the dialog builder do not meet the requirements for the applied system then developers have a possibility to change them. using the corresponding function button.

Then developers can change attributes related to every node in the table. They are the colour of the background, the colour of the frame, the size of the window or menu and their position on the screen. If the chosen node is a window then besides this information the fields of the window should be determined. Developers can connect a window of help with every node. The window contains necessary remarks or explanations for users of the applied system.

If the mode of operation is a revision of an applied system then developers revise necessary nodes of the script tree according to the user's requirements. It can be made in the manner described above.

Additional facilities of the shell

The shell can be used for development of an interface component of an expert system. The interface component gives a possibility to input data into the expert system. For implementation of this facility the dialog manager has a special function of transforming data input by the users into the form of a set of relations. Such a form is the most convenient data representation. The set of relation is the initial state of the expert system database.

Using the shell the interface component of the expert system "Consultant-2" [11] for medical diagnosis was implemented. The domain of the expert system is acute disease of abdominal cavity organs.

Conclusion

This paper has described an observation ontology, knowledge base for the physician of general tipe, architecture, functions, mathematical model of the dialog and implementation of problem independent shell of the system for intellectual supporting patient examination which can be used by therapeutist, surgeon, gynecologist, urologist, otolaryngologist, ophthalmologist, endocrinologist, neuropathologist and immunologist.

Acknowledgments

This work was carried out with financial support of the Russian Fund of Fundamental Investigations (grant 99–01–00634)

References

[1] van Melle, W.1979. A domain independent production rule system for consultation programs. In IJCAI-79: 923-925.

[2] Musen, M.A., Fagan, eds.1988. Use a domain model to drive an interactive knowledge editing tool. In Boose, J.H. & Gaines, B.R., editors, *Knowledge-Based Systems*, Volume 2: Knowledge Acquisition Tools for Expert Systems, 257-273, London. Academic Press.

[3] Albert, P., and Jacques, G. 1993. Putting Common KADS at work using Kads-Tool. In Kennistechnologie'93.

[4] Major, N., and Reichgelt, H. 1990. ALTO: an automated laddering tool. In Wielinga, B.J., Boose, J.H., Gaines, B.R., Schreiber, A., T.,& van Someren, M., editors, Current trends in knowledge acquisition, 222-236, Amsterdam, The Netherlands. IOS Press.

[5] Puerta, A.R., Egar, J., Tu, S. W., andMusen, M., A. 1992. A multiple-method shell for the automatic generation of knowledge based systems. IEEE *Transactions on Systems*, Man and Cybernetics, 22: 1361-1375.

[6] Shortliffe E.H. 1993. Clinical information systems in the era of manage care. Transactions of the American Clinical and Climatological Association: 203-215.

[7] S.B. Henry, L. Lenert, eds. 1993. Linking Process and Outcome with an Integrated Clinical Information System. 1993 Annual HIM SS Conference. San Diego, CA, 58-79.

[8] Chernyakhovskaya M. Yu. 1983. *Knowledge Representation for Diagnostic Expert Systems in Medicine Domain*. Vladivostok: FEBRAS.

[9] Uspensky V.A., Semenov A.L. 1987. *Theory of algorithms: basic discovery and applications*. M., Nayka.

[10] Gribova V.V., Kleschev A.S. 1997. Dialog models for management input data of expert systems. Technical Report, Vladivostok, Institute For Automation & Control Processes of FEBRAS .

[11] Chernyakhovskaya M. Yu. 1989. Estimation the expert system «Consultant-2» for medical diagnosis using an archival patient records of a few hospitals. Vladivostok, Institute For Automation & Control Processes of FEBRAS.