Deductive Inference in the Context of the Dialogue Process

Igor Chimir[1] and Waheeb Abu-Dawwas[2]
[1]Department of Information Technology, Odessa State Environmental University, Ukraine
[2]Department of Computer Science, University of Petra, Jordan

Abstract: *The paper is devoted to the investigation of a community between the dialogue process and the process of deductive inference. Attention is focused on the goal-oriented interactive process between two agents (active and reactive) involved into a step-by-step question-answering dialogue, which is considered as an activity directed to solving certain problem. A key element of the architecture of Dialogue Problem Solver is Dialogue Knowledge Base (DiKB). The paper demonstrates how an inference process of the classical rule-based systems can be emulated by navigation within DiKB. Some positive features of such emulation are discussed. Final section represents a formal theory of the DiKB structure from the inference process point of view. In conclusion the idea of a knowledge base agent with distributed architecture is discussed.*

Keywords: *Question-answering dialogue, logic of questions and answers, dialogue knowledge base, rule-based system.*

Received February 7, 2006; accepted April 18, 2006

1. Introduction

A theory presented in the article presupposes different interpretation of two concepts: the "interactive process" and the "dialogue process". We consider the interactive process as a generalization of a dialogue and define it as a step-by-step interaction of two agents, one of which is playing the role of an active agent, and the other plays the role of a reactive agent. The message which an active agent is sending to a reactive agent might have the sense of a command, an inquiry, or a question. In as much as in a given work the stress is made on the question-answering dialogue processes, we will consider a case when a message generated by an active agent has a logical structure of a question. A reactive agent carries out the work that is encoded in the question, and returns to an active agent the information about the results of the executed work, which has a logical structure of an answer. It is clear that there is a close logical connection between the pair:

1. Question of an active agent.
2. Answer of a reactive agent.

In the article, a conceptual basis of Logic of Questions and Answers is used (for example such concepts as question's subject and prerequisite), as well as the results of investigation of the structure of the question-answering pair mentioned in [1].

Understanding of an interactive process described above, is also correct for a dialogue process. However, in the case of a dialogue process, a considerable addition exists - logical connection between question-answering pairs themselves. Student work with an electronic searching system at the library is an example of an interactive process, and interrogation of a suspect by an investigator is an example of a dialogue process. The presence of close logical connection between question-answering pairs in the dialogue process gives it the nature of a problem solver, and the dialogue process itself can be considered as a meta-method of problem solving [2].

In [2], the dialogue process is considered from the point of view of declarative-procedural dichotomy of knowledge of an active agent, and conception of a Dialogue Knowledge Base (DiKB) is offered. The DiKB stores separately declarative and procedural knowledge necessary and sufficient for problem solving by the question-answering dialogue process. It is noted that declarative knowledge of an active agent which is necessary and sufficient for solving a certain problem is represented by a number of questions, while procedural knowledge is represented by the relationship between a number of questions, a number of dialogue steps, and a number of answer queues that are expected at every step of the dialogue process.

Let declarative knowledge of an active agent (in the form of encoded descriptions of questions needed for the question-answering dialogue in a given domain) be stored in the *memory of questions*, QueMem. Despite the fact that during the dialogue process a certain question may appear in many parts of the dialogue, QueMem keeps only one copy of each question. We consider QueMem as a memory with direct access to its elements; and, hence, we need an address to get access to the concrete question.

Let procedural knowledge of an active agent be stored in the structure called a *dialogue access method*, DiAM. The dialogue access method keeps a sort of knowledge such as "which question should be next?" and, therefore, is able to transform the current answer of a reactive agent into the QueMem address.

A *dialogue knowledge base*, DiKB, is defined as a composition of the *memory of questions*, QueMem, and the *dialogue access method*, DiAM. One of the advantages of such a structure of the DiKB is that it excludes multiple storing of encoded descriptions of questions. Storing of declarative knowledge of an active agent requires much more computer memory resources than storing of its procedural knowledge because declarative knowledge of an active agent (represented by question's subject elements) might have not only symbolic, but also non-symbolic representation in the form of graphical and sound files.

DiAM operates only with references to active agent's questions and reactive agent's answers; therefore, it does not require substantial computer memory resources. An active agent does not "compute" the subsequent question but searches it out in QueMem using DiAM as a method for achieving the goal. Therefore, we can also consider DiAM as a *certain problem-solving method* of an active agent which the agent uses for achieving the goal. A question-answering dialogue is a discrete process with a step as its structural and dynamic element. In Figure 1 the structure of DiKB in the form of UML class diagram is shown. DiAM is represented by a fragment of the diagram, which includes three classes: Step, AnticipatoryQueue (class of queues of expected answers), and Answer. Reflexive association with the indication of navigation direction models the fact that class Step must include the field that stores a reference on objects of the same class. Multiplicity of roles point out that, any step of the scenario can have several preceding and several subsequent steps. Relationship of composition type models the structure of a separate step. It is seen that one step consists of one question (class QueMem) and one queue of expected set of answers (class AnticipatoryQueue). The class AnticipatoryQueue, in its turn, consists of sequenced collection of answers (class Answer). Let's note that the class Answer represents a number of all non-repeated answers, necessary and sufficient to achieve the goal of the dialogue.

In [2], a Dialogue Problem Solving Process concept is introduced, and the analogy between the dialogue process and the process of searching solution in state space is demonstrated. The given work continues to develop the idea of a Dialogue Problem Solving Process and considers DiKB as a basic component of a Dialogue Problem Solver, and DiAM as a structure, which stores procedural knowledge of an expert.

Classical system designated to store procedural knowledge of an expert is a Rule-Based System, which represents a subclass of knowledge based agents, solving problems by means of making formal deductive conclusions. The work of a Rule-Based System generates not only the deductive, but also the dialogue process. In the following parts of the paper, theoretical research of deduction is made in the context of the dialogue process. The aim is to demonstrate that both processes are, in fact, different manifestations of a more global process of problem solving.

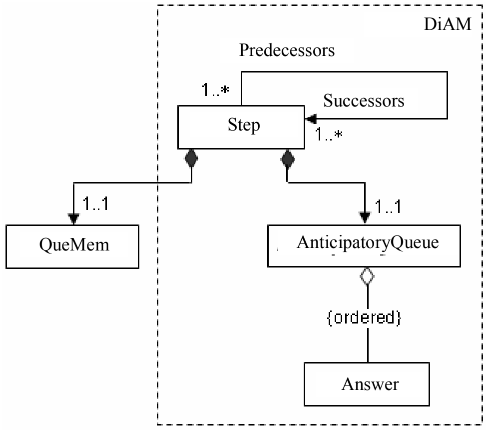


Figure1. The structure of a dialogue knowledge base.

2. Dialogue and Deduction in Rule-Based Systems

Dialogue with a user is an integral part of functioning of production systems after referred to as a Rule-Based System (RBS). In the dialogue process, RBS inquires a user about premises needed to continue the process of inference. From this point of view, RBS can be considered as a question-answering dialogue system. Classification of RBS in relation to the dialogue process depends on whether the explanatory component is used during its operation. In the case when the explanatory component is not used, RBS plays the role of an active agent (generating question), and its user plays the role of a reactive agent (answering the question), and the dialogue itself belongs to the category of dialogues with fixed roles distribution. Subjects and prerequisites of questions are formed in the way that users' answers contain the required facts-premises missing in the knowledge base. Here the subject and prerequisite of the question are interpreted from the viewpoint, offered in [2]. In the case when explanatory component is used, the dialogue belongs to the category of dialogues with free roles distribution, and, during the inference, RBS plays the role of a reactive agent. In order to proof these statements, it is sufficient to study protocols of interaction between RBS and the user. In [3] a fragment of the dialogue between a user and expert system MYCIN is shown. Analysis of this fragment easily allows us to come to the conclusion that, if one removes all WHY inquiries with subsequent explanations, then the remaining part is an example of the dialogue with fixed roles distribution, in which MYCIN plays a role of an active agent of the dialogue.

Dialogues with MYCIN, as well as with the other RBS, are supported by a special unit. The form of data kept in this unit differs from the form of facts and rules kept in the knowledge base. Thus, for RBS, the dialogue is only a form of communication (or interface) with a user. As shown in [2], the dialogue, in relation to DiAM, is simultaneously a form of knowledge representation and a form of communication with the opposite agent.

We can consider RBS as a system, which imitates the ability of human's intellect to make conclusion by means of building deductive chains. Knowledge needed to organize the deductive process is kept in the form of facts and production rules connecting them. The fact, from the point of view of the deductive process, is either a premise or a conclusion, or a premise-and-conclusion, if it is placed inside the deductive chain.

Types of problems solved by RBS are defined by two ways of building deductive chains, called forward and backward chaining [5]. The goal of forward chaining can be defined, as a transition from the initial set of facts-premises to the resulting fact-conclusion. During the forward chaining, each of the facts from the initial set of premises is connecting by a deductive chain with a fact-conclusion. The goal of a backward chaining is a transition from the initial fact-conclusion to the resulting set of facts-premises. During the backward chaining, a fact-conclusion is connected by deductive chains with each of the facts from the set of premises. Thus, we can speak about two types of connections between facts:

1. Connections between the facts that are set by production rules. Each rule connects a small (usually from 2 to 4) number of facts. Aggregate of production rules reflects knowledge of an expert (or a group of experts) regarding the target domain.
2. Connections between the facts determined in the inference process during deductive chains' creation. Deductive chains interconnect a big number (tens and hundreds) of facts. A principle that controls the creation of the deductive chains is called a deductive inference.

Connections of the first type exist in the knowledge base in explicit form and are represented by production rules' descriptions, and connections of the second type exist in implicit form. They are explicitly formed by an inference machine in the process of deductive inference. However, in the publications related to RBS we may find various ways of explicit representations of connections of the second type. As a rule, these are the following diagrammatical ways of representation of deductive inference chains: Inference Net [7], AND-OR graph [6] and Decision Tree [3, 6].

In the basis of a specific way of explicit representation of deductive chains there is, as a rule, the way of graphical representation of production rules. In symbolic form we may write down production rules in the following form:

If antecedent-facts (1)
Then consequent-fact

In (1), antecedent-facts mean facts-premises and consequent-fact means conclusion. Two graphical symbols, which are used to depict Inference Net (a) and AND-OR graph (b), are shown in Figure 2.

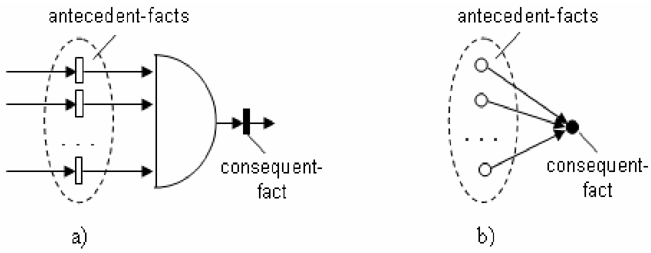


Figure 2. Graphical symbols for production rules, a) in Inference Net, b) in AND-OR graph.

In [3], authors use a special type of diagrams (called Decision Tree) for graphical representation of the relatedness of facts in the knowledge base. This kind of graphical representation of relatedness of facts differs from Inference Net and AND-OR graph just examined. In Decision Tree graphical symbols of production rules are not represented explicitly. Decision Tree is a network, in the form of bichromatic oriented graph. Nodes of this graph correspond to questions (ovals) or conclusions (rectangles). Branches are provided with legends, which contain answers to questions. A simple analysis allows us to generate a set of productions from the Decision Tree by passing through all possible paths from the top to one of the conclusions.

Thus, in RBS all facts are implicitly connected in the network by chains of possible conclusions. These implicit connections are set explicitly by the inference machine during the process of RBS functioning. Following this line of speculations, we can talk about some fixed number M, which is equal to maximum number of possible conclusions for a given knowledge base. From this point of view, "new" facts obtained during the inference are, in fact, in the knowledge base, and the process of inference provides only an access to them.

3. Deductive links in Question-Answering Dialogue

Forming explicit connections among facts in the inference process can occur within forward or backward chaining. In both cases, the facts are accumulated in the working memory of RBS. Forward chaining differs from backward chaining in the way these facts are interpreted. After the process of inference has been completed, we can segregate facts accumulated in the working memory, into two classes:

1. Intermediate facts.
2. Terminate facts.

In the process of forward chaining, the accumulated facts are interpreted as Antecedent Facts (AFact), and the final goal of forward chaining is to receive the terminate Consequent Fact (CFact).

In the process of backward chaining, the accumulated facts are interpreted as CFact, and the final goal of backward chaining is to get the terminate list of AFact. Diagram in the form of Inference Net shown in Figure 3, illustrates an understanding of the inference process given herein.

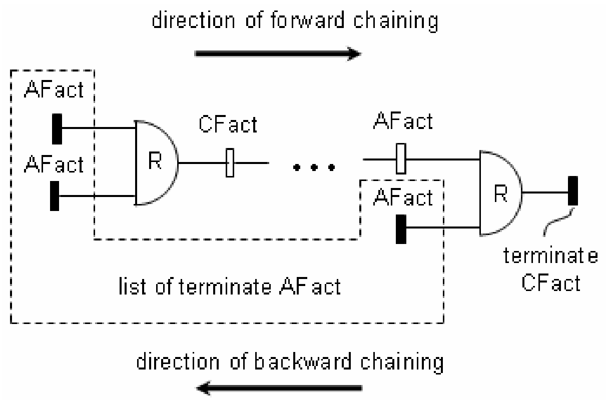


Figure 3. Inference net, which illustrates the concepts of intermediate and terminate facts.

Any intermediate fact is an antecedent-fact and a consequent-fact simultaneously. Namely, a consequent fact in relation to the previous rule, and an antecedent fact in relation to the subsequent rule. Intermediate facts are inside the inference tree, and are needed only to get terminate facts ultimately. Terminate facts are an aggregate of target AFact or CFact. They are placed "at the edges" of the inference tree and determine desired task solution. It is clear that the number of intermediate facts determinates the duration of the inference process; therefore it is desirable that in each concrete inference the number of intermediate facts be minimal.

After completion of building the inference tree, the inference itself can be presented without intermediate facts in the form of mapping of the list of terminate antecedent-facts into terminate consequent-fact (in the case of forward chaining), or in the form of mapping of the terminate consequent-fact into the terminate list of antecedent facts (in the case of backward chaining).

$\{AFact_i\}$, $i = 1, \ldots, n \rightarrow CFact$, *forward chaining* (2)
$CFact \rightarrow \{AFact_i\}$, $i = 1, \ldots, n$, *backward chaining* (3)

where:

$\{AFact_i\}$: A list of terminate antecedent-facts.
CFact: Terminate consequent-fact.

Thus, after a certain inference has once occurred, the inference tree can be "convolved" into an equivalent production rule (4), which includes only terminate facts.

$$\text{If } (AFact_1 \,\&\, AFact_2 \,\&\, .. \,\&\, AFact_n) \quad \text{Then } CFact \tag{4}$$

Rule (4) can be placed into the knowledge base. In this case, when it is necessity to make exactly the same inference, the system can use rule (4) instead of repeatedly building the inference tree. We would like to note that the above mentioned process of convolving of inference tree into the rule (4) was used in the unified cognitive Soar model where it was called "chunking" [4]

Rule (4) allows us to complete the inference process in *n* steps, if there is a method, which can produce just one fact at each step. The goal of the following part of this paragraph is to demonstrate how a question-answering dialogue can work in such a method.

As was noted earlier, that for a fixed knowledge base the number of inferences is bounded above, and thus there are no more than M possible inferences that can be modeled with no more than M rules (4). Let us represent conjunctions in the left part of (4) as a chain of *n* sequentially connected switching elements, each of which corresponds to AFact. For this purpose, we will use a Petri-model of DiAM [2]. In Figure 4, rule (4) is depicted as a fragment of DiAM network. Direction of arrows coincides with the direction of forward chaining. In Figure 4, answers of reactive agent correspond to facts; therefore they are represented by transitions. A question of an active agent, from the point of view of logical inference, plays a secondary role. It provides a reactive agent of the dialogue (by its subject) with an expanded set of related and alternative facts. For instance: place of rest is mountains; place of rest is the beach, place of rest is the casino. While answering the question, a reactive agent chooses one fact from the expanded set of alternative facts, and in such a way forms next conjunct in the antecedent (4).

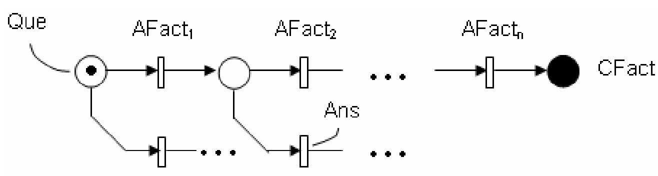


Figure 4. Fragment of DiAM network, which corresponds to rule (4)

It is natural to suppose that, if a fragment of DiAM network depicted in Figure 4, is equivalent to one inference tree, then the whole DiAM network is equivalent to all M inferences that are possible for a given knowledge base.

Movement of token along the network models a step-by-step inference process and the inference itself is completed within *n* steps, where $n + 1$ is a number of terminate facts in the inference tree. At each step, exactly one fact is added to the antecedent (4). The type of inference is determined by interpretation of nodes. If a node, which corresponds to CFact completes the inference chain, then a forward chaining takes place. If the inference chain begins from the node that corresponds to Cfact then a backward chaining takes place.

With the aim of getting a formalized description of deductive connections in a question-answering dialogue, let us consider a simple example of the transformation of RBS knowledge base into the DiAM network. Let us have the following set of facts:

a intention is rest; (5)
b road is bumpy;
c to use a Jeep;
d place of rest is mountains;
e place of rest is the beach;
f speed is needed.

Let the rules, which connect facts (5), are as follows:

1 If a & b Then c; (6)
2 If d Then b;
3 If a Then f;
4 If e Then b.

We can consider rules (6) as certain inference schemes. For example, rule 1 allows to conclude c if we have premises a and b. Rule 3 is incorporated into the knowledge base to illustrate a potential conflict character of the set of rules, which were randomly chosen. If the working memory contains facts a and b at the same time, then both inferences: c (rule 1), and f (rule 3) are possible. In the table in Figure 5, the list of facts (5) is put into correspondence to: The list of questions (second column) and graphical representation of nodes in DiAM network (third column).

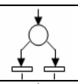
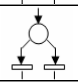
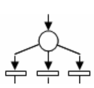
Facts of the Knowledge Base Given in (5)	Questions, Answers of which Generate Corresponding Facts	Graphical Representation of DiAM Node
a) Intention is rest	What are you intending to do: To have a rest or to work?	
b) Road is bumpy	What kind of road is expected: Bumpy or flat?	
d) Place of rest is the mountains e) Place of the rest is the beach	Where are you going to take a rest: in the mountains, on the beach or somewhere else?	

Figure 5. Example of transformation of a knowledge base' facts into DiAM nodes.

A fact is present in the question in the form of one of the elements of its subject. During transformation of facts into corresponding DiAM nodes, facts d and e were transformed into one question as shown in Figure 5. Despite the fact that these facts are presented as separate statements in the knowledge base, from the point of view of logic of questions-answering relations, they represent elements of the subject of one question.

Transformation of facts shown in the table in Figure 5 allows us to create a DiAM network, which is equivalent to the inference trees of the knowledge base, described in (5) and (6). Such a network is shown in Figure 6.

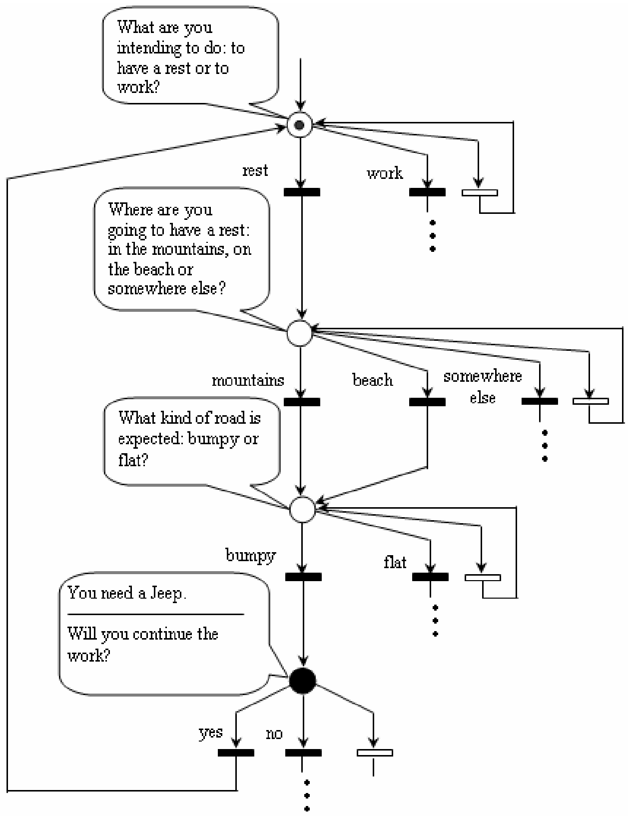


Figure 6. DiAM network equivalent to inference trees that are possible for the knowledge base, described by facts (5) and rules (6).

DiAM network in Figure 6 corresponds to a forward chaining as the inference is completed when the token reaches target CFact position. In Figure 6, this position is black.

Target positions have their own interpretation, which is distinguished from regular interpretation of positions and transitions: transitions correspond to facts, and positions correspond to questions. *Target position*, in case of forward chaining, contains *Consequent-Fact* (CFact) and *next question*. Network in Figure 6 presupposes that, after receiving a conclusion and a positive answer to the question "Will you continue the work?", the system returns to initial position.

By comparing the ways of knowledge representation by means of DiAM network with the form accepted in RBS, we can mention the following:

DiAM network has one input, which marks the beginning of the dialogue process, and an inference tree having several inputs. The presence of a number of inputs of inference tree complicates and even precludes combining an inference with the dialogue process in the framework of classical RBS.

In the DiAM network the inference process reduces to the navigation process controlled by a reactive agent. Thus, inference machine of RBS is transformed into a simpler one, navigation machine.

An attractive characteristic of DiAM is a principal absence of conflict situations in the process of navigation. This is a result of natural restriction that exists in the question-answering dialogue represented in the DiAM network. According to this restriction, during the question-answering dialogue, an active agent can generate only one question as a response to the reactive agent's answer.

In the DiAM network there emerges the possibility of uniting several facts in one node, in the case when they are elements of the subject of the same question. In the example examined, two initially different facts: “place of rest is the mountains” and “place of rest is the beach”, are elements of the subject of one question “Where are you going to have a rest: in the mountains, on the beach or somewhere else?”

In the process of inference machine's operation, quantity of data stored in the working memory increases according to development of the inference process, and must be regulated, considering in view of a limitation of computer resources. In the dialogue knowledge base, created on DiAM base, at every step of the dialogue it is necessary to store data, which are relevant only to one DiAM node irrespective of the length of the deductive chain.

4. Question-Answering Relations in a Logical Inference Context

In previous sections it is shown that the dialogue and deductive processes are, per se, the manifestation of dualism of some more general process, which has step-by-step and interactive nature. In the present section we offer a model of the question-answering dialogue considered from the point of view of logical inference.

In the question-answering dialogue process, according to Logic of Questions and Answers [1], an active agent, in order to receive the next fact, presents to a reactive agent a message, which has logical organization of a question:

$$Que \overset{def}{=} ?\, \rho\, \sigma \qquad (7)$$

where:
ρ: Prerequisite of the question.
σ: Subject of the question.

In [2] it is shown that the subject of the question is either an extended list of properties and characteristics that are assigned to an object-thing:

$$\sigma = x,\ \{P_\alpha (x)\},\ \alpha = 1, \ldots, m \qquad (8)$$

or an object-property that is intrinsic to an extended list of things.

$$\sigma = P\ (x),\ \{x_\alpha\},\ \alpha = 1, \ldots, m \qquad (9)$$

The subject of the question σ, as it was noted earlier, stores relative and, often, alternative facts.

For example, a set of facts:

place of rest is the mountains;
place of rest is the forest;
place of rest is the casino

can be presented in the form (9) in the following way:

$$\sigma = P\ (x),\ \{x_\alpha\},\ \alpha = 1, \ldots, 3 = \{Fact_1, Fact_2, Fact_3\} \qquad (10)$$
$$Fact_1 = P\ (x)\ /x = x_1$$
$$Fact_2 = P\ (x)\ /x = x_2$$
$$Fact_3 = P\ (x)\ /x = x_3$$

where:
P (x): Is a one-place predicate: “Thing x HAS A PROPERTY TO BE A PLACE OF REST”.
x1: Mountains.
x2: Forest.
x3: Casino.

As only one conjunct is needed at each step to form an antecedent in (4) (this means that it is necessary to get exactly one fact from subject σ), the answer must represent either one object of a thing type, or one object of a property type:

$$Ans/Fact = x,\ \{P\ (x)\} \qquad (11)$$
$$Ans/Fact = P\ (x),\ \{x\} \qquad (12)$$

Formulas (11) and (12) can be described verbally in the following way:

“answer/fact is an object-thing x, which has property P (x) ”, or

“answer/fact is an object-property P(x), which is intrinsic to thing x”.

Therefore, in the case of forward chaining, an answer is one of subject's facts that can be interpreted as AFact (or intermediated fact), if the inference process is not completed, or as CFact, if the inference process is completed.

Prerequisite ρ is needed to form an answer from the point of view of the cardinal number of the answer set. However, as in the given case, an answer represents a single fact, the semantic of a prerequisite is to inform a reactive agent that the question's subject contains alternative list of facts.

Thus, a single cycle of question-answer exchange between active and reactive agents of the dialogue can be used to get the next fact in the process of logical inference. A question and a fact-answer relative to it, can be represented in the following way:

$$Que = \rho,\ x,\ \{P_\alpha (x)\};\ \ \alpha = 1, \ldots, m \qquad (13)$$
$$Ans/Fact = x,\ \{P\ (x)\}$$

or

$$Que = \rho,\ P\ (x),\ \{x_\alpha\};\ \ \alpha = 1,\ \dots,\ m \quad (14)$$
$$Ans/Fact = P\ (x),\ \{x\}$$

where:

ρ: Prerequisite, which determinates the alternativeness of elements of sets

$$\{P_\alpha(x)\} \text{ and } \{x_\alpha\};$$

P(x): One-place predicate: "x HAS PROPERTY P(x) ";

m: Cardinal number of question's subject.

A node of DiAM network correspondes to one question-answering cycle. A chain, which consists of consequently connected nodes, models inference (in the sense of rule (4)), and the whole DiAM can be considered as a model of the knowledge base, which stores procedural knowledge (which has changed over to the category of automatic) in the form of the network of "all possible inferences".

A formal description of DiAM, as a composite of a dialogue knowledge base, can be presented using a representation of a question-answering dialogue in the form of Petri network.

Let:

- FACT: Be a complete set of facts-answers or their identifiers, that cover a certain domain, and are needed and sufficient to form a dialogue knowledge base, and
- QUE: Be a complete set of questions and their identifiers in the form (13) or (14).

Let an aggregate of transitions, which are incidental to an i-position of DiAM inference network, and which are connected with this position by outgoing branches, correspond to the following set of facts:

$$F^i \cup NF^i$$

where:

F^i: Is an i-step set of expected and recognizable alternative facts-answers (which belong to a question's subject of a given step).

NF^i: Is an i-step set of non-recognizable facts-answers.

Then an i-step of inference in the DiAM inference network is modeled by a pair:

$$Que_i,\ (F^i\ U\ NF^i) \quad (15)$$

In the graphical interpretation of the inference network DiAM a question Que_i corresponds to a Petri net position and (F^i U NF^i) is an aggregate of transitions that are incidental to it. A condition of a transition launching in the DiAM inference network is a coincidence of the answer received from a reactive agent, with one of facts from (15).

DiAM Petri-model, considered as a knowledge base in the form of the network of "all possible inferences", can be represented by the following quadruple:

$$DiAM = (QUE,\ FACT,\ NextQue,\ NextFact) \quad (16)$$

where:

$$NextFact:\ QUE \rightarrow FACT \quad (17)$$

Is the function of the subsequent and expected facts-answers, which sets the mapping of a complete set of questions (or their identifiers) into a complete set of expected facts.

$$NextQue:\ FACT \rightarrow QUE \quad (18)$$

Is the function of the subsequent questions (or their identifiers) that sets the mapping of a complete set of expected facts into a complete set of questions. A specific feature of a question-answering dialogue (the active agent generates and transmits to the reactive agent only one question at one time) puts over the NextQue function the following limitation. For each fact-answer from the complete set of expected facts-answers, the function of the subsequent questions determines the single question from the complete set of questions.

Hence, an i-position of DiAM network of inferences can be described in the following way.

$$NextFact(Que_i) = (F^i\ U\ NF^i) \quad (19)$$

$$NextQue(Fact_j) = Que_i \quad (20)$$

Figure 7 is a graphical presentation of a step of Petri-model of DiAM inference network and illustrates the concepts introduced.

5. Conclusions

The comparison of deductive connections in the RBS, and in the dialogue system based on the dialogue knowledge base, allows us to offer a conception of a knowledge base agent (KB-agent) with distributed architecture.

Organization of classical RBS obeys a certain standard *de facto*, according to which an architecture of such a system includes the following basic components:

1. Global base of facts.
2. A set of production rules or base of rules.
3. Control system, or rules' interpreter.

The evolution of RBS architecture is directed to improving each of the listed components, separately taken. In other words, the architecture of the most RBS is based on conception of concentration of basic functions of the system in separate blocks. Such "concentrated" architecture is universal and allows designers to improve separate components independently. However, it generates some problems, for example, a problem of developing the effective strategy for rules' interpreter, which takes into account a potential conflict character of a set of rules (inference mode with return, etc.).

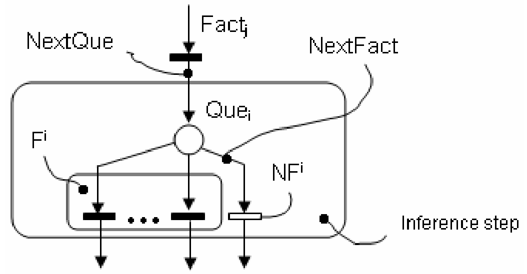


Figure 7. Graphical presentation of main concepts of Petri-model of DiAM inference network.

Distributed architecture is such an organization of the system when all its functions are distributed among a number of single-type and structurally similar elements, each of which contains a fragment of: Global base of facts, base of rules, control system, and interface with a user. The process of functioning of distributed KB-agent is a step-by-step interpretation of the above-mentioned single-type elements controlled by the dialogue process with a user. Thus, for such KB-agent the dialogue with a user is an essential part of the problem solving process. Stemming from the results obtained in the given work, the basis of organization of distributed KB-agent can be a dialogue knowledge base, represented by DiAM and memory of questions (QueMem).

The aim of this paper is to demonstrate a DiKB conception in comparison with classical RBS but the theory has also several practical implementations.

In one of our projects we designed a DiKB-oriented software called Dialogue Problem Solver and applied this software to solve such a problem as foreign languages teaching. In this project the Dialogue Problem Solver plays the role of a skillful tutor and the DiKB a repository of declarative knowledge of the language itself along with procedural knowledge of the tutor.

We are going to describe the Dialogue Problem Solver in our further publications.

References

[1] Belnap N. D. and Steel T. B., *The Logic of Questions and Answers*, New Haven and London, Yale University Press, 1976.

[2] Chimir I. and Abu-Dawwas W., "Modeling of a Procedural Knowledge by a Dialogue Knowledge Base," *The International Arab Journal of Information Technology*, vol. 1, no. 1, pp. 128-134, 2004.

[3] Levin R. I., Drang D. E. Edelson B., *A Comprehensive Guide to AI and Expert Systems: Turbo Pascal Edition*, McGraw-Hill, 1988.

[4] Newell A., *Unified Theories of Cognition*, Cambridge, Harvard University Press, Massachusetts, 1994.

[5] Paul H. and Davis K., *Expert Systems: Artificial Intelligence in Business*, John Willey, 1985.

[6] Russell S. and Norvig P., *Artificial Intelligence: A Modern Approach*, Second Edition, New Jersey, Pearson Education Inc, 2003.

[7] Winston P. H., *Artificial Intelligence*, Addison-Wesley, 1984.

Igor Chimir is a professor of computer science at Odessa State Environmental University, Ukraine. He has been involved in cognitive science research for more than 15 years and has particular interests in the subject-independent theory of computer and human dialogue, intelligent tutoring systems, and in object-oriented modeling of cognitive processes. He has published more than 90 technical papers and is the author of several textbooks. His current activities focus on investigating the applicability of modern software engineering approaches to the area of cognitive psychology.

Waheeb Abu-Dawwas is an assistant professor at the Department of Computer Science, University of Petra, Jordan. He obtained his PhD from the Ukrainian National Academy of Science, Ukraine, in 1999. His research interests include dialogue process modeling, and the development of personalized computer educational systems. He has published many original contributions in the field of dialogue-oriented tutoring systems, personalized tutoring systems, and in cognitive systems.