Using the expert systems in the operational management of production

IOAN CONSTANTIN DIMA, JANOS. GABRARA, VLADIMIR MODRAK, PACHURA PIOTR, CONSTANTA POPESCU,
Department Management-Marketing
University “Valahia” of Targoviste, ROMANIA
dima.ioan.constantin@yahoo.com, tantapop@yahoo.com

Abstract: In the conditions of computing the human society, the operational management activities can be improved by using artificial intelligence. The expert system is an intelligent system based on the symbolic representation of knowing, implemented on 2 hardware structure specific to the application, which processes a lot of knowledge for solving special issues about activities that are difficult to examine. The main modules of this system are of knowledge database of rules and of facts, the inference engine and the inference mechanism. The paper describes, in detail, an expert system for managing the technological costing process for metals at one metallurgical enterprise.

Key/Words: Expert systems, Artificial intelligence, Interface, Search engine, Implementing the expert system

1 Introduction

The efficient operation of the integral production systems is based on artificial reasoning, as condition for simulating the human reasoning and supplying the expert’s essential role in an actual domain. The expert system represents a major component of artificial intelligence, by means of which the knowledge of some experts in that domain are stored and, by means of a declarative approach, can be then dynamically exploited by an artificial mechanism, simulator of natural rationing triggered at the level of the human brain. The phrase expert system generally evokes the new techniques of management in various activity domains and which are currently assimilated through the notion “contents of a database management”.

2 Contents of research

The notion of “expert system” refers to software that uses all knowledge specific to the various activity domains during the process of elaborating a resolution for each problem that may intervene within a well defined domain. Therefore, the expert system aims at gathering the experience and rationing of a human expert for replacing it. This accumulation represents the formation foundation of the knowledge base, which is one of the basic components of these systems [1].

The expert system is an intelligent system based on the symbolic representation of knowing, implemented on a hardware structure specific to the application, which processes a lot of knowledge for solving special issues about activities that are difficult to examine.

Regardless of the definition of the expert system, there are four features which classify software as expert system. It works at the expert level of competence; it uses an inference mechanism for accomplishing deduction; the expertise performed is based on knowledge specially gained; programming such systems involves the presentation of the knowledge of some experts in the domain, this knowledge could be kept in the database for future use [3].

The architecture of any expert system must contain at least three main modules, which determine the so-called essential system, in figure no. 1, an expert system is represented, adapted to the necessities connected to the operational management.
The main modules of the expert system are [2]:

- **the knowledge database** – represents the component which stores the system’s permanent specialised knowledge, provided by the expert in that respective domain. The knowledge database is formed by the database of facts and the database of rules, integrating the knowledge related to solving a problem in a given domain, the knowledge consisting of facts (assembly of data related to the problem) and rules (laws, methods and elements of rationing applicable to facts) treated by means of an engine of inferences with the aim to accomplish an objective requested through the problem dealt with.

- **the knowledge purchase module** allows the formation of a system’s knowledge database. Usually, the knowledge specific to the domain is taken over from human experts by means of a specialist in the engineering of knowing. Another modality of getting knowledge specific to the domain is to automatically generate this knowledge using various methods of learning;

- **the database of rules and the database of facts** are elementary components of the knowledge database. The database of rules is formed by the manifold of relations representing the expert’s knowledge and fulfils the following meanings: it consists of the rules which the connections between facts are specified by; the rules use initial (known) facts for deducting new facts based on inductive and deductive processes (in the situation where the rules may provide the direction and control of the manner which other rules are applied, the concept of metarules is highlighted).

- **the database of facts** consists of all facts which the creator of the expert system has enclosed in it or which are deducted during the rationing (of the inference process). The database of facts consists of an assembly of structures of complex data charged with a certain meaning, generically called “facts”, related to a particular problem which a resolution is requested for through the system of a database of knowledge. The facts represent the assembly of the affirmative knowledge necessary to treat the domain dealt with;

- **the inference engine** is destined to exploiting the assembly of knowledge for solving the problems and it represents the actual processing element in the expert systems, which activates the corresponding knowledge in the knowledge database, starting from facts (problem input data), thusly forming rationing which lead to new fact called output facts. In conclusion, the inference engine is a software implementing algorithms of rationcinations of deductive (directed by facts), inductive (directed by aims) and mixed (featured by the course of an inductive rationcination, followed by a deductive rationcination) types, but which does not depend on the knowledge database;

- **the inference mechanism** is the executive part of the expert system that uses the codified knowledge from the database of knowledge for creating inferences and drawing conclusions. It is a software which allows: exploiting the know-how database, coordinating the rationcination for deducting new facts which would represent the new know-how database of the system, dynamically creating and explaining rationcination, adopting resolutions on rules that must be triggered and on order where they are applied, procuring new knowledge. In fact, the inference mechanism is the core of an expert system which performs deductions in the process of solving a problem, by covering and eventually changing and adjusting the elements from the know-how database.

Regardless of the rationcination mode used, the basic cycle of an engine of interferences carries four stages: the selection extracts from the database of rules and from the one of facts all information characteristic to the sub domain of solving the problem; the filter consists in comparing the premises, the rules previously selected to the facts featuring the issue to be solved, for determining the sub manifold of applicable rules and on must be chosen for being performed; the execution itself of a rule consists in adding one or more facts into the database. It is also possible that at this stage, external procedures would apply, respectively to have access to the database, to processors of tables or to questions asked to the user.

The occurrence of IT equipment and efficient software, which may be purchased for low prices, allow each manager of an industrial company to take into account the use of IT use, without incurring too big of a financial risk. In practice, there are several modalities of choosing and using an expert system. Thusly, a first choice must follow the possibility of the expert system to cover the time gap “t” that exists between the duration necessary to getting an answer and the duration...
necessary to forming knowledge at the time when it is questioned.

The starting point for such a step was to know the main features of the casting technology, as first modality of getting familiar with the process itself of performing the operations necessary to the cast items in the finished form, for costs as low as possible and in the conditions of minimising the risks of there occurring damages [3].

Thusly, casting represents the technological process of making parts for mining machines, by solidifying the molten material in the cavity of a mold.

Compared to other technologies, casting represents the following advantages: the possibility to obtain the parts with complex configuration, with a shape as close as possible to the finished one; the small amount of processing addition; the use of machines of simple and cheap construction, reusing the waste of materials by melting, which have resulted during the technological process or partial recovery of the value of scraps, lower unitary costs of the cast parts than of those made by other procedures etc. between the disadvantages associated to this technological process, the most frequent are: high probabilities of obtaining parts with defects; probabilities of lower resistance compared to the wrought parts; the use of some considerable amounts of materials for obtaining the parts etc [1].

Making the shapes represents the printing operation of the foundry pattern in the formation mixture. After extracting it, the pattern remains, where the melted metal is poured in order to obtain the parts. This is the main operation in the technological process of obtaining the cast parts and decisively conditions their quality parameters. Making the shapes can be performed manually (in frames, in soil or by patterns) or mechanised (by throwing, by pressing with a membrane, by shaking and pressing, by blasting etc.). At S.C. UTILAJUL S.A., the classical method is used, that of manual formation in frames, featured by low costs related to the documentation, shaping materials and tools under the conditions of low accuracy and quality of the cast surfaces [2].

Drying the shapes and cores represents the operation by means of which they are subject to heating in order to remove the water from the formation mixtures and to increase the bonding capacity of the binders. By drying, a considerable increase of the mechanical resistance of the formation mixture is performed, as well as the decrease of permeability and limitation of the gas amount which develop in the moulds during casting. At S.C. UTILAJUL S.A., the sand dryer on fluidising bed is used for drying the cores, as well as for drying the moulds.

Assembling the moulds for casting consists in assembling the semi-moulds and cores into a single system. The cores and mould cavity are supported by means of stamps, and when these are missing or are too long, it is done by means of wood supports. After assembling the moulds and elaborating the alloys, then follows the casting itself into moulds. The melted metal is taken from the casting pot and directed into the cavity of the mould, by means of the casting line.

Rapping is the operation by means of which the item is taken out from the shaping frame and most of the formation mixture is removed from the outside and the core mixture from the inner cavities of the items. Rapping the mould is done after the metal has solidified and cooled down at a temperature where no more defect dangers can occur, by cooling the item more rapidly in the air.

Cleaning is the operation of removing the formation mixture adherent to the cast items. Usually, cleaning is done in several stages: superficial cleaning done for removing the waste heads; preliminary cleaning consisting in removing as much as possible the formation mixture from the item; final cleaning which is done after the thermal treatments applied and consists in removing all materials which are still on the surface of

Fig. 2 The diagram of the technological process for obtaining the parts cast at S.C. UTILAJUL S.A.
the item (formation mixture remained or oxides formed during the thermal treatment). Final mechanised cleaning is done at S.C. UTILAJUL S.A. by means of the sandblast machines, which direct an abrasive jet over the items which are to be subject to this process [1].

Removing the waste heads represents the operation of cleaning away the crop ends and the casting network from item A. Removing burrs from the cast item implies the cleaning of the excess of material which appears on the plane of shape saturation. Thermal treatments are applied to the cast items, with the following aims: removing the internal strains occurred when casting; relieving the processing by chip removal; improving some mechanical properties when no other thermal treatments are performed anymore or when the structure is to be prepared for obtaining some better properties for future thermal treatments. The probability of there occurring defects in foundry is higher than in other activity sectors, reason for which the qualitative control must be run more intensely. Taking into account all of these, qualitative control in foundries is of two types: preliminary control applied on materials used and operations before casting the item, in order to eliminate the possibilities of defects to occur, respectively the final control, which is done on the cast item and must be performed in two stages. During the last stage, the item is monitored, starting from rapping and going through all other operations, in order to find out potential defects and for adopting as quickly as possible the adequate measures. During the second stage, the control of the items is done, after all operations for obtaining the gross cast item have finished, in order to retain those cast items which do not correspond to mechanical processing, respectively, to future use.

Taking into account all of the above, an expert system is elaborated below for managing the casting technological process for metals at S.C. UTILAJUL S.A. Expert system for managing the technological casting process for metals at S.C. UTILAJUL S.A. [3]

/*DATABASE OF RULES AND DATABASE OF FACTS*/

domains

data_list=data_type* data_type=is(symbol,symbol);
has(symbol,simbol);
is in(symbol,symbol);
is on(symbol,symbol);
is_by(symbol,symbol);
cost(symbol,symbol);
needs(symbol,symbol);
predicates
rule(integer,data_type,data_list,real)
question(integer,data_type)
clauses
rule( 1 ,has(x,bad_form),
[is on(x,sand_deposit)],l).
rule(2,has(x,good_form),
[is_by(x,models)],l).
rule(2,has(x,good_form),
[is_in(x,kernel)],l).
rule(3,has(x,unsatisfactory),
[has(x,bad_form),
has(x,reject)],l).
rule(3,has(x,unsatisfactory),
[has(x,bad?parameters)],l).
rule(4,has(x,nice),
[has(x, good_form),
has(x,mechanical_debate)], 1).
rule(5,has(x,expensive),
[cost(x,over_100)],l).
rule(6,has(x,inexpensive),
[cost(x,under_30)],l).
rule(7,has(x,bad_part),
[is(x,unsatisfactory),
needs(x,repair)],l).
rule(8,has(x,good_part),
[is(x,inexpensive),
has(x,processing)], 1).
rule(8,has(x,bad_part),
[is(x,nice),
has(x,cleaning)], 1).
rule(9,has(x,good_part),
[is(x,good_parameters),
has(x,heat_treatment)],l).
rule(10,has(x,good_part),
[is(x,expensive),
has(x,processing),
has(x,good_parameters)],l).

question(1 ,is on(x,bad_part)):-
write("Is",x," a bed part?").
question(2 ,is_by(x,sand_deposit)):-
write("Is "x," into the sand deposit?").
question(3 ,is_in(x,mixture)):-
write("Is "x," in mixture?").
question(4,has(x,good_part)):-
Write("Has "x," good part?").
question(5,cost(x,over_100)):-
Write("Does it cost ",x,", over 100?.")
question(6,cost(x,under_30)):-
Write("Does it cost ",x," under 30?.")
question(7,needs(x,repair)):-
Write("Needs "x," repairs?").
question(8 ,has(x,debate)):-
Write("Has "x," mechanical debate?").
question(9,has(x,heat_treatments)):-
Write("Has ",x," heat treatments?").
question(10,has(x,found_line)).
Write("Has ",x," found line?").
question(11,has(x,good_processing)).
Write("Has ",x," good for processing?").

/* INFERENCES ENGINE AND USER INTERFACE* 

domains 

intl=integer* 
file=dest include "motor.kwl" database 
fact(data_type) predicates 
proc user(char) 
startexp 
read_char(Char) 
interference(data_type,data_list,intl,real) 
getresponse(symbol) 
validresponse(symbol,real,symbol,symbol) 
get_first(data_list,data_type) 
delete(data_type,data_list) 
process(data_type) 
check ans(symbol) 
processwhyno(intl) 
display rule(integer) clearfacts addl(integer,intl) 
repeat 
convert(symbol) 
clauses 
start_exp:- 
makewindow(1,7,7,"Expert system", 1,1,20,20), repeat, 
write("i_init, a_add facts"), nl, write("q_quit, r_run"), nl, 
write("Enter your option: (i/a/q/r)"), read_char(char), 
procjaser(char). procuser('q'). proc_user('i'):write("Clear 
all facts ?"), read char(Resp), Resp='y', clear_facts,fail. 
proc_user('a'):write("Enter a fact:"), readln(S), 
convert(S,T), add_fact(T), !,fail. procuser('r'):-write("Your question",S, 
"has been proved with", Prob, "probability"). processrule(S,T):_ 
write("Using fact -".Fact), nl. fact(Fact), write("F 
act already exists"), nl. 
convert(S,T):-openwrite(dest,"convt.dat"), write(S), closefile(dest), 
openread(dest,"convt.dat"), readdevice(dest), 
createfile(dest), readdevice(keyboard). process(Query):- 
interference(Query, [Query], [] ,Prob), 
write("Your question",Query, 
"has been proved with", Prob, "probability"). process(Query):- 
write("Your question", Query, "cannot be proved"). interference(Query,Cond,Prob), 
check_fact(Rl,Query), delete(Query,Cond,NewCond), 
processrule(Cond,Rlist,Prob), delete(Query,Cond,NewCond), P1 = (P+Prob)/2, 
process rule(NewCond,Rlist,P1). 
process_rule([],_). process_rule(Cond,Rlist,P):-get_first(Cond,Q), 
interference(Q,Cond,Rlist,P). getresponse(R):- 
readln(Ask), check_ans(Ask,Rep, !), 
R=Rep. getresponse(_):-write("Try another answer, please"), 
write("Fact",Fact), nl. fact(Fact), write("F 
act already exists"), nl. 
convert(S,T):-openwrite(dest,"convt.dat"), write(S), closefile(dest), 
openread(dest,"convt.dat"), readdevice(dest), 
createfile(dest), readdevice(keyboard). process(Query):- 
interference(Query, [Query], [] ,Prob), 
write("Your question",Query, 
"has been proved with", Prob, "probability"). process(Query):- 
write("Your question", Query, "cannot be proved"). interference(Query,Cond,Prob), 
check_fact(Rl,Query), delete(Query,Cond,NewCond), 
processrule(Cond,Rlist,Prob), delete(Query,Cond,NewCond), P1 = (P+Prob)/2, 
process rule(NewCond,Rlist,P1). 
process_rule([],_). process_rule(Cond,Rlist,P):-get_first(Cond,Q), 
interference(Q,Cond,Rlist,P). getresponse(R):- 
readln(Ask), check_ans(Ask,Rep, !), 
R=Rep. getresponse(_):-write("Try another answer, please"), 
write("Fact",Fact), nl. fact(Fact), write("F 
act already exists"), nl. 
convert(S,T):-openwrite(dest,"convt.dat"), write(S), closefile(dest), 
openread(dest,"convt.dat"), readdevice(dest), 
createfile(dest), readdevice(keyboard). process(Query):- 
interference(Query, [Query], [] ,Prob), 
write("Your question",Query, 
"has been proved with", Prob, "probability"). process(Query):- 
write("Your question", Query, "cannot be proved"). interference(Query,Cond,Prob), 
check_fact(Rl,Query), delete(Query,Cond,NewCond), 
processrule(Cond,Rlist,Prob), delete(Query,Cond,NewCond), P1 = (P+Prob)/2, 
process rule(NewCond,Rlist,P1). 
process_rule([],_). process_rule(Cond,Rlist,P):-get_first(Cond,Q), 
interference(Q,Cond,Rlist,P). getresponse(R):- 
readln(Ask), check_ans(Ask,Rep, !), 
R=Rep. getresponse(_):-write("Try another answer, please"), 
write("Fact",Fact), nl. fact(Fact), write("F 
act already exists"), nl. 
convert(S,T):-openwrite(dest,"convt.dat"), write(S), closefile(dest), 
openread(dest,"convt.dat"), readdevice(dest), 
createfile(dest), readdevice(keyboard). process(Query):- 
interference(Query, [Query], [] ,Prob), 
write("Your question",Query, 
"has been proved with", Prob, "probability"). process(Query):- 
write("Your question", Query, "cannot be proved"). interference(Query,Cond,Prob), 
check_fact(Rl,Query), delete(Query,Cond,NewCond), 
processrule(Cond,Rlist,Prob), delete(Query,Cond,NewCond), P1 = (P+Prob)/2, 
process rule(NewCond,Rlist,P1). 
process_rule([],_). process_rule(Cond,Rlist,P):-get_first(Cond,Q), 
interference(Q,Cond,Rlist,P). getresponse(R):- 
readln(Ask), check_ans(Ask,Rep, !), 
R=Rep. getresponse(_):-write("Try another answer, please"), 
write("Fact",Fact), nl. fact(Fact), write("F 
act already exists"), nl. 
convert(S,T):-openwrite(dest,"convt.dat"), write(S), closefile(dest), 
openread(dest,"convt.dat"), readdevice(dest), 
createfile(dest), readdevice(keyboard). process(Query):- 
interference(Query, [Query], [] ,Prob), 
write("Your question",Query, 
"has been proved with", Prob, "probability"). process(Query):- 
write("Your question", Query, "cannot be proved"). interference(Query,Cond,Prob), 
check_fact(Rl,Query), delete(Query,Cond,NewCond),

3 Conclusion and Proposals

The expert system for improving the operational management related to the cast items is especially destined to assisting the exploitation and maintenance personnel when diagnosing the complex defects which may occur during the course of this process and for the efficiency of the technical measures that must be taken after the occurrence of those respective disfunctionalities.

This system has the advantage of exploiting the memory of the past events and the experience of more experts whom participated at founding the system's database of know how, without comprising them into an expert system, such knowledge cannot be transmitted to those who ensure the use of the system forward [2].

The complex mutations that occur in the current industrial context have determined companies to
revolutionise their strategies, to search permanent innovation for keeping and/or winning some new market segments. All these occur during a period of “explosion” of the IT systems, highlighting the need of integrating all company’s activities by means of the computer. As strategic option in the domain of operational management, computing the activity by using the expert systems represents a pragmatic modality of improvement, which must be explored thenceforth at S.C. UTILAJUL S.A., due to some incontestable advantages such as: performance – the expert systems do not lose their knowledge as time passes by, being capable of continuously working; the possibility of being multiplied – many copies of an expert system may be easily done, while creating new human experts represents a long and expensive process; efficiency – implies low costs compared to performing the expertises by the human specialists; consistency – similar actions are processed and treated in the same manner; objectivity – the system is not accessible as opposed to human experts whom can be subjective; documentation – an expert system may provide a permanent documentation of the decision process; high working speed etc.

References: