ARCHIVES
of
FOUNDRY ENGINEERING

Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences

ISSN (1897-3310)
Volume 8
Issue 3/2008

75 – 80

14/3

Shell expert system for technological knowledge acquisition and access

A. Macioł[a,]*, A. Stawowy[a], R. Wrona[b]

[a] Faculty of Management, AGH University of Science and Technology, Gramatyka 10, 30-067 Krakow, Poland

[b] Faculty of Foundry Engineering, AGH University of Science and Technology, Reymonta 23, 30-059 Krakow, Poland

* Corresponding author. E-mail address: amaciol@zarz.agh.edu.pl

Received 22.04.2008; accepted in revised form 29.04.2008

Abstract

Our studies on codification of knowledge about castings and techniques of their fabrication have led us to discover that it is possible to define the technological knowledge in open and flexible format. As a result the shell expert system for technological knowledge acquisition and access is proposed in this paper. The solution designed by us differs from the similar tools in that the whole knowledge has been written down in the form of relational databases. At the same time, the knowledge has been structured in the form of parameters in such a way that its restructuring and further development is possible without going into the structure of relations and by operating only in the extensional part of data model. Using new available sources, user may develop the knowledge base by himself.

Theoretical assumptions and preliminary results of proposed system are also given. Our original idea is to design database structure in such way that any source of knowledge can be classified into many categories. Due to that idea as well as user-friendly interface the users can find any information in the way which is related to their needs and tooting.

Keywords: Application of information technology to the foundry industry; Product development; Expert systems; Structured query language; Rule-based reasoning

1. Introduction

The knowledge which is necessary for technological problems solving (e.g. technology selection, process design) has various sources and nature. This is partly – specific for human-beings – *tacit knowledge* which is hard to discover and to formalize. Although it is difficult task, it is possible to describe the technological knowledge. So why a lot of information systems supporting technology design process is developed. The proposed systems commonly use relational database for data storing and rule-based expert system for providing access to these data. In those systems the structure of database is adjusted to specific designing tasks and all rules are a priori defined. It means that developed systems are tailored to solve a strictly defined group of problems and the user can not change the structure of collected knowledge.

Our studies on codification of knowledge about castings and techniques of their fabrication have led us to discover that it is possible to define the technological knowledge in open and flexible format. Now we can propose shell expert system for technological knowledge acquisition and access. In authors opinion, full automatic design of casting technology will not be possible in foreseeable future. Literature studies confirm that computer systems are able to aid the design process. These systems can support preliminary decision making or finally process planning when the decision of client order realization was made. In the first case not precision of solution but fastness of decision making process plays a key role.

The solution designed by us differs from the similar tools in this that the whole knowledge has been written down in the form of relational databases. At the same time, the knowledge has been structured in the form of parameters in such a way that its restructuring and further development is possible without going into the structure of relations and by operating only in the extensional part of data model. Using new available sources, the user may develop the knowledge base by himself. Proposed system collects the knowledge of verbal form (technical standards, research paper or another sources) and formal form (declarations and procedures). In the first case relational database designed by us performs as some guide through the sources recorded on local server or on wide area network. Our original idea is to design database structure in such way that any source of knowledge can be classified into many categories. Due to that and users friendly interface the user can find any information in the way which follows his/her needs and tooting.

We have discovered that the knowledge necessary to technological problem solving may always be presented as set of statements with various number of strings which depict facets in reasoning rules. As the statements are translated to form enabling their evaluation in universal procedure it is possible to record the knowledge in relations which are irrespective of the knowledge meaning and the number of facets.

2. Related works

In the field of automatic casting process selection and related problems, the literature is not extensive. Yue et al. [1] have discussed the CAD/CAE/CAM integrated system to aid die casting dies of sedan parts. Their expert system is built to design the technological scheme of die casting process: the parameters such as the injection pressure, plunger speed, gate velocity and filling time are selected and checked in accordance with the type of alloy, the wall thickness and structure of the casting. Brown et al. [2] have described a web-enabled repository system for supporting distributed automotive component development. The repository is using EPISTLE generic entity modelling principles. Component design and analysis representational models are referenced as resources through document meta-data, with access via the virtual repository. Ravi and Akarte [3] have proposed a virtual foundry environment for preliminary process planning of cast elements. The plans are automatically generated by case based reasoning using the nearest neighbour algorithm. The database contains the castings descriptions created using Casting Data Markup Language. More recently, Maciol et al. [4] have presented the model of decision problem and rule-based expert system for the selection of casting process. The shell system was used to prototype the concept.

One of the examples of AI techniques application in castings technological problems is defects modelling. A team of the AGH researchers have elaborated information-diagnostic systems called INFOCAST and OntoGRator, formalized in the form of semi-symbols [5, 6].

The similar researches are conducted in various area of technology e.g. forming. Katayama et al. [7] have presented system based on fuzzy rules for cold forming process designing. Glynn et al. [8] have proposed system for turbine blades forging, based on similarity detection. Next category of expert systems are suitable for aiding of particular processes designing. Im et al.[9] have showed system for designing of ball-stud forging. The system chooses the best sequence of operations, basing on starting and final shape of detail. In this case, FEM is also used for inference result verification.

In summary, computer aided systems for technological process design presented in literature have a priori defined, fixed structure. As a result the user is not able to develop the knowledge without the reformulating the knowledge base (and he needs system's designer support).

The aim of our work was to develop an open system that guarantees easy knowledge changing and extending.

3. Problem formulation

Engineering professionals, who design plans of casting process, need propositional (scientific) knowledge related to the domain area where they want to generate required "non-existent-yet" useful products and "future objects". But to do so they also need, as a necessary condition, non-propositional knowledge, including what Polanyi [10] identified as tacit or personal knowledge. Tacit knowledge is implicit, and is related to the outcome of individual skill, practice and experience. Tacit knowledge cannot be easily made explicit or represented formally. Visual representations like pictures, diagrams, and descriptions, help to expose tacit knowledge, but it largely is embedded in experience as personal knowledge and it results from individual practice. Tacit knowledge cannot usually be transmitted verbally, through oral or written form. It is subjective, personal knowledge. It is usually not mediated by reasoning or logic; it is immediate knowledge. Tacit knowledge is usually learned by working side by side with an expert. Perrin [11] affirms that operational knowledge usually "remains tacit because it cannot be articulated fast enough, and because it is impossible to articulate all that is necessary to a successful performance and also because exhaustive attention to details produces an incoherent message".

Opinions mentioned above confirm that creating automatic, artificial "engineering" system is impossible. However, it is possible to support technological problems with computer tools as CAD/CAM, CAE, CAPP, etc.

The aim of our research is to build artificial system, which extensively supports the decisions making about whether to take production of part ordered by customer. System should answer the following questions:

- is it possible to manufacture the certain part in defined terms of production ,
- which method of casting to choose,
- what will be the approximate cost of fulfil order.

Problem of choosing technology without full analysis of design task is often solved by expert systems. Knowledge of those systems contains rules pointing out qualities, restrictions and faults of technological processes in connection with products attributes, which have to be manufactured. Building this kind of systems does not require to collect complete knowledge about process plans. In case of casting process such approach is not suitable. Economical effectiveness of casting process is strongly related to details, which evaluation is possible after confirmation of manufacturing process.

Our solution assumes the possibility of examine simplified procedure of design process based on costs evaluation. We assume that information about customers order will be inputted manually but we consider the possibility of automatic information inputs by using tools that interpret CAD and XML documents. System – after rough analysis of problem – suggests user the technological solutions possible to use.

The user chooses some solution and realizes the process of simplified design. The final result of such session is presentation with manufacturing process assumptions and evaluated costs. User can repeat this procedure many times for different available technological solutions. Because working objective of the system is evaluating of customer's order cost, reasoning process is based on model of variable casting costs. Costs of charge, costs of mould production, energy and labour costs have a main influence on castings' cost. The costs of production quality depending on number of faults which can appear in process are also important. It is necessary to remember that the casting is often only one phase of full production process. The more precise is shape of final product, the lower are costs linked with mechanical working. So cost evaluation needs solution of following particular tasks: charge qualifying, choose of mould technology, confirmation of size of raw casting with allowance, define type of heat treatment and finishing method. In relation to mould technology it is necessary to solve particular problems such as the process and materials used to chills, models and cores preparing.

4. Knowledge formulation method and inference engine

The experts' knowledge has various characteristics. We can distinguish the following forms: procedures, mathematical formulas, data tables, nomographs, schemas and figures. This is why we have problem with knowledge modelling. The rule-based modelling is a common solution. In our opinion, the possibility of combining the relational database and SQL with the rule-based reasoning system enables to construct the complete knowledge ensuring the data integrity and integration. The solution is based on the IwQ (Inference with Queries) method developed by Macioł [12].

The ability of IwQ method to knowledge representation was proved in our previous researches [13]. IwQ joins the concept of frame-based knowledge representation, replaced partially with relational database model, with inference possibilities given by procedural languages (Transact-SQL in used now) and is near to rule-based languages idea. Thanks to them it is possible to handle relational databases tools (SQL) to store and edit knowledge rules and also to solve the basic selection problems in the rule-based system. IwQ model of the rule-based system use an extended form of the rules including both control statement and dynamic operations. Generic form of a rule can be presented as follows:

$$
\begin{aligned}
&rule(i):(A_1\ r\ d_1)\wedge(A_2\ r\ d_2)\wedge\ldots\wedge(A_n\ r\ d_n)\\
&\rightarrow\\
&\quad set(B_1=b_1),B_2=b_2,\ldots,B_b=b_b)\\
&\quad H_1=h_1,H_2=h_2,\ldots,H_h=h_h\\
&\quad next(j)\\
&else\\
&\quad set(C_1=c_1,C_2=c_2,\ldots,C_c=c_c)\\
&\quad G_1=g_1,G_2=g_2,\ldots,G_g=g_g\\
&\quad else(k)
\end{aligned}
$$

where $(A_1\ r\ d_1) \wedge (A_2\ r\ d_2) \wedge \ldots \wedge (A_n\ r\ d_n)$ is the regular precondition formula, $B_1 = b_1, B_2 = b_2, \ldots, B_b = b_b$ is the specification of the facts to be changed in knowledge base after successful execution of the rule, $H_1 = h_1, H_2 = h_2, \ldots, H_h = h_h$ is the specification of conclusions forming a direct output of the rule (e.g. decisions or queries to be displayed on the terminal or control actions to be executed) in case when it is successfully executed, $C_1 = c_1, C_2 = c_2, \ldots, C_c = c_c$ is the specification of the facts to be changed in knowledge base in case of failure, $G_1 = g_1, G_2 = g_2, \ldots, G_g = g_g$ is the specification of conclusions in case of failure and ***next(j), else(k)*** are the specifications of control; the next(j) part specifies which rule should be examined immediately after successful execution of rule i and else(k) part specifies which rule should be tried in case of failure.

In our model we use one object class to store all the necessary data. This object is called facet. The facet can have a value taken from a relational structure and can be used as object attribute value or variable value. The main part of knowledge is recorded in relational database tables which simplifies the reasoning process and the data maintenance. In case of new information inserting only the data tables updating is necessary.

As information describing manufacturing processes is variable, it is not possible to formulate in advance the sets of all features meeting all potential requirements. Thus we decided to move the main part of knowledge from data model to data tables. Owing to transfer of information (concerning knowledge structure) to extensional part, introducing complex relations is possible without modifying scheme of relation. Such elasticity of knowledge notation causes trouble connected with its interpretation. IwQ method allows to utilize data selection mechanism in relations and joins. Even so it is impossible to build a SQL query without knowing *a priori* number of requirements which have to be met by analysed query.

Data model consists of 5 relations: Properties, ProductionProblems-Properties, ConditionPropertyValues, Conditions, and Dictionary. The user is able to define any manufacturing problem and to insert its definition into the tables. All information necessary for the inference process is written down in the form of relational databases. The Entity Relationship Diagram of relations depicting the technology information is presented in Figure 1.

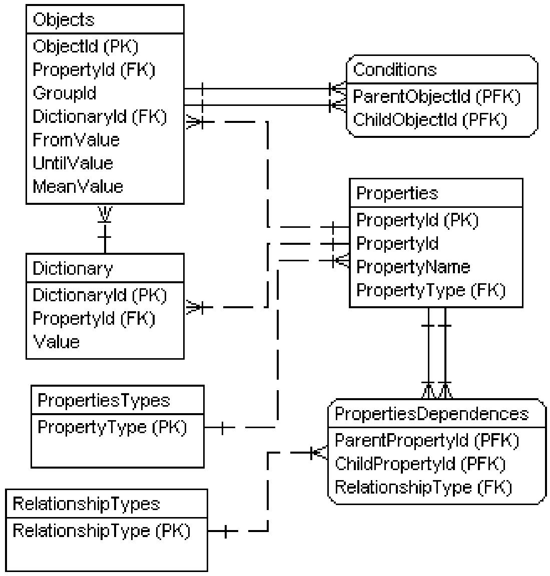

Fig. 1. Entity Relationship Diagram of knowledge model

Concepts defined by the user are stored in Properties relation. Relationships between concepts are many-to-many type and are stored in PropertiesDependences relation. Thanks to it the user is able to link any child concept with many parent concepts. For example, concept *temperature* may be linked with charge, tool, mould etc. If a record in Properties relation depicts the *charge temperature* then two records linking this concept with concepts *temperature* and *charge* will appear in PropertiesDependences relation. The relationships between concepts may have various nature. The types of relations are stored in RelationshipsTypes relation. The order which creates the structure of ontology is hierarchical by nature and may be many-to-many type. Another type of relation depicts the membership of one concept to another concept. It is aggregate relationship which describes physical relation *is-part-of* and the type of relation between abstract concepts. We can use this kind of relation to enterprise description (the machines are parts of the plant) or design process description (if the sequence of the phases is not indicated). If the sequence of some processes is indicated we have to use the precedence relationship which means that parent concept precedes child concept. This relationship is used to designing or manufacturing process description.

Every concept in Properties relation is additionally described by concept type from PropertiesTypes dictionary (value, dictionary value, reference to text file, reference to procedure, etc.).

Instances of concepts are stored in Objects relation; the value of concept may be:

- symbol from area defined in Dictionary relation (e.g. jolt moulding machine, turn-table moulding machine),
- value (single) or set of values (FromValue, MeanValue, UntilValue),
- reference to outer resource (document, web site, picture, CAD file, etc.).

The relationships between object are also recorded. These relationships describe the rules which link instances of concepts (equivalent relationship). As single object may be both premise and conclusion, the possibility of storing many-to-many relationship between objects has to be incorporated. The records in Conditions relation perform this task.

In ***Objects*** relation may occur more than one record representing conditions of particular connected conclusion, which is combined in conjunction. It concerns table type information as well as inference rules. Instance of this issue may be simple table representing the quantity of Mg added to iron during spheroidization process in relation to basic temperature and sulphurous content (see Table 1).

Table 1.
Exemplary table type information

S content in basic cast iron [%]	Mg addition [%] and temperature of cast iron [C]			
	1350	1400	1450	1500
0.02	0.07	0.12	0.15	0,20
0.03	0.08	0.13	0.16	0,22
0.04	0.12	0.14	0.19	0,24
…	…	…	…	…

Equivalent records in relations ***Objects*** and ***Conditions*** are depicted in Tables 2 and 3.

Table 2.
Objects relation contents

ObjectId	GroupId	PropertyId	…	MeanValue
1	1	1		0.02
2	1	2		1350
3	2	1		0.03
4	2	2		1350
5	3	1		0.04
6	3	2		1350
..				
17	9	1		0.02
18	9	2		1400
19	10	1		0.03
20	10	2		1400
…	…	…		…
33	1	3		0.07
34	2	3		0.08

Table 3.
Conditions relation contents

ParentObjectId	ChildObjectId
33	1
33	2
34	3
34	4
…	…

On account of fact that set of conditions is neither written in relations scheme nor know, a priori selection can be only done by exclusion of tuples which do not satisfy the search criteria. Well known logical thesis is used:

$$\neg(a_1 \wedge a_2 \wedge ... \wedge a_n) \Leftrightarrow \neg a_1 \vee \neg a_2 \vee ... \neg a_n$$

which may be, in this case, interpreted that if any premise from conjunction of premises is contradicted, it results in exclusion of particular inference. Therefore it is possible to exclude these tuples from the temporary table (a copy of ***Objects*** table), which are equivalent to inferences linked with successively tested premises. Consequently only the tuple matching the confirmed conclusion remains.

Suppose we search for appropriate Mg quantity when S content in basic cast iron is equal to 0.03 and temperature is equal to 1350 C. Process of seeking adequate values is carried by inference system which operates according to instructions of IwQ method. Owing to relation stored in ***PropertiesDependences*** relationship, between concept ‘spheroidization’ and concept ‘the quantity of Mg addition’, system identifies necessity of finding suitable value. From the relationship ***Conditions*** system reads that for determining that parameter, knowledge concerning S content in basic cast iron and temperature is obligatory. Inference mechanism searches for adequate information in relations which describe design task. If fails to find information, it may take an attempt to find values which would allow to determine them. Such search is conducted continuously until it is found, that there are no more premises which would allow to update the value of wanted parameter. That time system asks user for the most elementary data. After determination of the input data, system searches for output value. Let us assume that established property name is stored in facet ***@CurrentProblem*** and values of S content in basic cast iron and temperature are written successively into facet ***@A***.

Resulting from join:

```
SELECT
ObjectId, GroupId, PropertyName, MeanValue, ParentId
INTO @Objects
FROM Conditions
INNER JOIN Objects AS Objects_1
ON Conditions.ParentId = Object_1.ObjectId
INNER JOIN Objects
INNER JOIN Properties
ON Objects.PropertyId = Properties.PropertyId
ON Conditions.ChildId = Objects.ObjectId
INNER JOIN Properties AS Properties_1
ON Objects_1.PropertyId = Properties_1.PropertyId
WHERE (Properties_1.PropertyName = @CurrentProblem)
```

indispensable temporary table is created. Table contest allows to determine names of output parameters which are read in succession from query:

```
SELECT DISTINCT PropertyName FROM @Object
```

and stored into facet (e.g. ***@PName***). Afterwards system has to exclude from temporary ***@Objects*** table all the tuples, which are linked with groups of premises in such a manner, that for concept name ‘S content’ value in column ***MeanValue*** differs from ***@A*** and for concept’s name ‘Basic temperature of cast iron’ ***MeanValue*** differs from ***@A*** (after substitution of strength value there). In order to do that ,operation of tuples exclusion is carried out twice:

```
DELETE FROM @Object
WHERE PropertyName = @PName AND MeanValue <> @A
```

Consequently in table ***@Object*** remain only tuples corresponding to the value of The quantity of Mg addition equal to 0,08.

It is sufficient to read value of any tuple from the ***ParentId*** column, and basing on it find adequate value. In the same manner are depicted all the rules stored in presented relations.

Important part in technical design process are calculations. IwQ method provides straightforward computation mechanisms performing on the facet values. More complex operations must be carried out by independent executable modules. In relationship ***ProceduresParameters*** are stored references to concepts, which constitute parameters of executed by particular object procedure. Inference engine searches for values of the object, which is an instance of properties described as procedure. It identifies, by type, which procedure must be called (with name defined by ***MeanValue*** column) and passes required parameters. If their values were not determined, system searches for them in the same manner as shown above.

5. Example of system use

The test application was built to provide some investigation of proposed method.

The main function of system consists in designing the simplified technology of casting. The process is driven through the set of standard forms which input the values for inference engine. The designing process is illustrated in treeview form. The user is able to analyze the decision taken at each phase of process in relation to inputted data. The exemplar guide for pressure die casting of Al-Mg alloy is presented in Figure 2. The final output is technological document containing product model, forming method, material data, the variable costs of each operation, and anticipated cost of mechanical working.

It can be observed that the same result may be achieved in different ways. The place of particular object in the presented treeview is related to its parent relationship in knowledge base. At the same time, all child objects of given parent are presented. This is not redundancy: due to the relationships involved each object is stored only in one place. The exemplar object is WWW document, which URL address is stored in knowledge base. The types of other objects are marked with the proper icons.

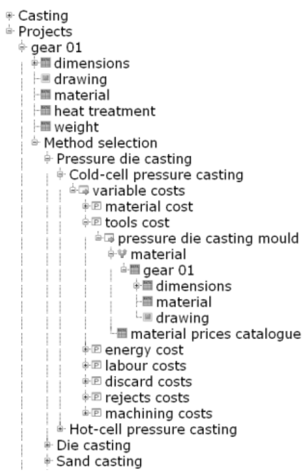

Fig. 2. An exemplar guide for pressure die casting of Al-Mg alloy

The interface of inference engine working in information query mode is presented in Figure 3.

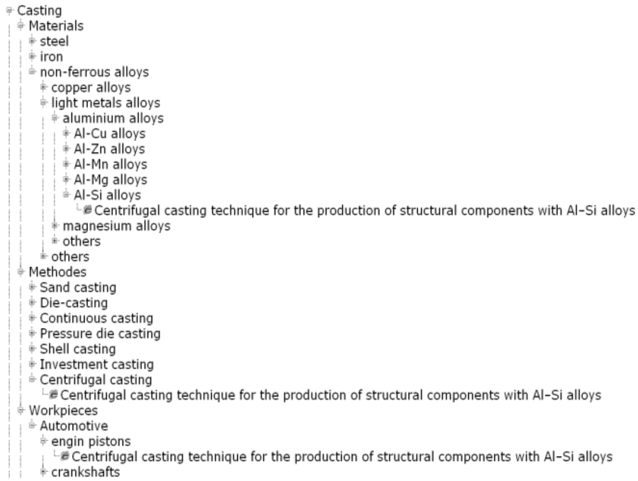

Fig. 3. Interface of the inference system utilized in information search mode

6. Conclusions and further works

The presented system is tested now. Preliminary results confirm that the idea of knowledge modelling fulfils the authors expectations. Flexibility of knowledge and open structure of system guarantee the possibility of knowledge extension.

Some problems have though appeared. The way of knowledge storing in relations is very flexible but data editing is difficult and information query may be longer.

We are dealing with the first problem trying to build highly automated interface which would translate rules and tables into set of relations. The problem of query time is connected with the knowledge size. We are testing now the partition of main files into the virtual tables which is solution taken from blackboard expert systems.

Simultaneously we work to develop an ASP.NET interface which will enable the presented prototype to work as web based casting process generation system.

Acknowledgements

This work is supported by the MN from the resources assigned for scientific researches in years 2006 – 2009.

References

[1] S. Yue, G. Wang, F. Yin, Y. Wang, J. Yang, Application of an integrated CAD/CAE/CAM system for die casting dies, Journal of Materials Processing Technology, No 139 (2003) 465–468.

[2] D. Brown, D. Leal, C. McMahon, R. Crossland, J. Devluki, A Web-enabled virtual repository for supporting distributed automotive component development, Advanced Engineering Informatics, No. 18 (2004) 173–190.

[3] B. Ravi, M. Akarte, Web-based Collaborative Engineering of Cast Products, 30th Computers and Industrial Engineering Conference, 2002, Greece.

[4] A. Macioł, A. Stawowy, R. Wrona, Casting process selection using rule based reasoning, Archives of Foundry vol. 6, No. 19 (2006) 189–194 (in Polish).

[5] S. Kluska-Nawarecka, J. Tybulczuk, M. Kisiel-Dorohinicki, H. Połcik, E. Nawarecki, Distributed information system for foundry industry with application of multi-agent technology, Archives of Foundry, vol. 6 No. 18 (2006) 45–52 (in Polish).

[6] B. Mrzygłód, S. Kluska-Nawarecka, Collecting and formalization of knowledge about surface faults of metal products, Archives of Foundry, vol. 5 No. 17 (2005) 175-182 (in Polish).

[7] T. Katayama, M. Akamatsu, Y. Tanaka, Construction of PC-based expert system for cold forging process design, Journal of Materials Processing Technology, vol. 155–156 (2004) 1583–1589.

[8] G. Glynn, G. Lyons, J. Monaghan, Forging sequence design using an expert system, Journal of Materials Processing Technology, vol. 55 (1995) 95-102.

[9] C.S. Im, S.R. Suh, M.C. Lee, J.H. Kim, M.S. Joun, Computer aided process design in cold-former forging using a forging simulator and a commercial CAD software, Journal of Materials Processing Technology, vol. 95 (1999) 155-163.

[10] M. Polanyi, The tacit dimension, Doubleday Anchor, New York, 1967.

[11] J. Perrin, The inseparability of technology and work organization, History and Technology, vol. 7 No. 1 (1990), 1-13.

[12] A. Macioł, An application of rule-based tool in attributive logic for business, Expert Systems with Applications, vol. 34 (2008) 1825–1836.

[13] A. Macioł, A. Stawowy, R. Wrona, A hybrid system for acquisition of and access to the data from foundry technical standards, Archives of Foundry Engineering, vol. 7 No. 3 (2007) 213-216.