Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 15, 2015, No. 1

INTELLIGENT INTERFACE FOR DECISION SUPPORT SYSTEM IN METALLURGICAL DOMAIN

KRZYSZTOF REGULSKI[1]*, GABRIEL ROJEK[1], DOROTA WILK-KOŁODZIEJCZYK[1], STANISŁAWA KLUSKA-NAWARECKA[2], EDWARD NAWARECKI[1], GRZEGORZ DOBROWOLSKI[1]

[1] *AGH University of Science and Technology, al. A. Mickiewicza 30, Krakow, Poland*
[2] *Foundry Research Institute, Zakopianska 73, Krakow, Poland*
**Corresponding author: regulski@agh.edu.pl*

Abstract

Extensive decision support system composed of several modules providing different functionalities for the user can be a challenge in designing of a usable, plain and consistent user interface. A characteristic of advisory systems in the domain of metallurgy is their complex architecture designed to deal with multiple types of tasks in many areas. The objective of this paper is to present the possibility of creating an intelligent interface providing the user with complete freedom to act without precise knowledge of the multiple functionality of the system. The paper presents the architecture of developed advisory system in the domain of metallurgy enriched with the description of mechanisms, which enable obtaining an intelligent user interface. The intelligent user interface should guide the user at navigation through all system modules in order to meet all needs that are the reason for this system usage. In designing of such interface, the Case-Based Reasoning methodology is proposed to be used. Case-Based Reasoning ensures a pattern for building systems that are able to learn on the basis of human actions associated with human decisions and use such knowledge to resolve the currently appearing problems. Each user query directed to the system, and each of his decisions about the choice of functionalities of the system modules supplies the system with the crucial information on proper use of the whole decision support unit related with the needs of a user. Skillful use of this knowledge can provide convenient and efficient performance of intelligent interface.

Key words: intelligent interface, metallurgy, decision support system, Case-Based Reasoning, information technology

1. INTRODUCTION

For a few years, Foundry Research Institute in Krakow has been leading an international project entitled: "Developing solutions for the conceptualization and sharing of knowledge components about foundry technologies in the context of innovation and improvement of production processes" (No. 820 / N-Czech Republic / 2010/0), within the framework of which system modules have been developed constituting information and functional equivalent of the components of knowledge about the foundry technology. These modules make up the platform for knowledge sharing, which provides the user with a constant supply of current information and knowledge in the field of casting technology, derived not only from the literature, but also from other sources, such as expert works, databases or algorithmically acquired knowledge from online sources.

The information system can largely automate the process of decision-making supplying to the user knowledge of the issues, which are technological problems after a previous dialogue with the user to establish the facts. It can be anticipated that the problems the user will want to solve may involve a variety of issues and provide very different tasks for the system. What emerges is the goal that one has to put in front of the system, and in particular in front of its interface - design knowledge base in such

ISSN 1641-8581

a way that it is interactive, thus enabling the user to get through dialogue with the system only the knowledge that is necessary to solve the problem and the system to anticipate on the basis of this dialogue what kind of knowledge the user has in mind.

A key element of the system from the point of view of the user becomes the interface which enables using diverse knowledge stored in databases that make up the local knowledge base. Within the framework of the discussed system, an integrated knowledge base of this type will hold: 1) literature and scientific publications, 2) materials (articles, data) obtained from the Internet, 3) various types of production and experimental data gathered by the users, 4) the accumulated knowledge in the form of rules and ontological models, 5) data on the characteristics of the materials, including dilatograms.

So rich a variety of document types will require diverse approaches and many record formalisms, as well as a preprocessing of user queries. Imagine a user who plans to use the system. He begins by entering a single phrase, a search query. It is expected that the system will return the output documents related to this query. They are supposed to be publications, photographs, data, reports and rules related to the topic. The task of the system in such a situation is not only going through a variety of integrated resources looking for the searched term, but also finding topics related to this term, i.e. "similar" topics. For the user, this process has to be transparent. As an output the user may obtain a list of suggested documents divided into sections comprising publications / photos / data / rules.

Without an intelligent interface, the user is forced to use the system as a set of modules, in which each has a separate query window and an individual way of interaction with the user - sometimes it is enough to give the search query, sometimes the user has to select a different mode of action (give domain list, select the set of descriptors, or click on a properly formulated query to the database).

This what happens inside such a system becomes crucial, especially during the integration of different sources of knowledge and semantic integration, i.e. the search for documents thematically "similar" to the searched query.

An example of the flow of events in a typical case of using the system with intelligent interface could look like this:

1. User opens the system interface.
2. User enters the subject.
3. System (interface) analyzes the subject
   a) searching for similar queries in the history of use,
   b) searching for data on the Internet and in the documents and databases in natural language,
   c) screening databases of materials if the user has mentioned the name of the material,
   d) searching data on treatment if the name of the technology has been mentioned,
   e) retrieving statistical data about the searched resources,
   f) searching alternative sources of knowledge (source materials, rule-based expert knowledge, database of publications, database of innovation, designs and patents, database of dilatograms, etc.).
4. The results are saved in XML files.
5. The results are displayed in a proper form.

The system not equipped with an interface agent would not have had step 3 in exchange for what in step 2 it could have: "the user selects an appropriate module of the system and enters his question in an appropriate form". Our goal is to design an effectively operating system which, based on information provided by the user (query, selection of module, automatically collected history of interaction with the system), will carry out the task of knowledge management, i.e. the location, acquisition and sharing. Such an interface should be fully compatible with the modules directly cooperating and should be as much as possible susceptible to subsequent modification or development. It will constitute in itself a learning system based on user behavior.

2. METALLURGICAL-DOMAIN DECISION SUPPORT SYSTEM

Implemented in the Foundry Research Institute in Cracow, the advisory system has a modular architecture, which consists of the following modules already described in most of other publications:

- CastSemaWiki - a platform for sharing and storing of documents based on the wiki mechanism, enriched with module for semantic annotation and semantic search based on the domain ontology for cast iron (Haratym et al., 2013).
- Module for the integration of material data for cast iron along with the application which allows collecting the experimental data on cast iron treatment and changes in its properties under the influence of this treatment (figure 1) (Kluska-Nawarecka et al., 2013).

– System for information retrieval from the Web (SWATLocalCrawler) - an automatic system for searching the web data resources to extract information about the foundry products. The detectors operate in conjunction with SWATLocalCrawler and allow searching for predefined patterns in the content of Web pages in domains that are selected for searching (Opaliński et al., 2011).
– InnoCast Database - a relational database with a web application on innovation, patents, publications and research projects issued in recent years for problems related to forging, rolling and casting.
– Dedicated ETL module - system for integration of data from metallurgical databases (e.g. SINTE: database of publications in the field of casting, NORCAST: database of standards, and InnoCast database) consistent with the ETL methodology - Extract, Transform and Load (Wilk-Kołodziejczyk et al., 2014).
– CastExpert + - the existing and developed at the Foundry Research Institute expert system in the field of diagnosis of casting defects enhanced with image simulations in the field of iron casting and multiscale modeling (Nawarecki et al., 2012).
– Domain ontology for cast iron supplemented by a set of rules in SWRL regarding changes in cast iron under the influence of treatment (Kluska-Nawarecka et al., 2013).
– Base of dilatograms together with an application that allows the digitization of graphs and searching.
– RoughCast – attribute casting defects classification system implemented according to the theory of rough sets (Dobrowolski et al., 2011).
– OntoGrator - system of knowledge integration based on the domain ontology (Regulski & Kluska-Nawarecka, 2012).
– The semantic integration system for unstructured text documents executed by algorithms of latent semantic indexing (Gurda et al., 2013).

Each of these modules has a different functionality, characteristics, architecture and method of operation. User interaction with these modules has different requirements - simple queries are addressed to relational databases, some modules perform operations instead of returning the results to the screen,

Fig. 1. *Sample screen from Module for the integration of material data for cast iron (in Polish).*

etc. The user may use the individual modules in different situations - some of them will be used in the search for documents containing information, some will be used to find specific material or process parameters, others will serve visualization or monitoring of the Internet or rule-based inference. Development of an intelligent interface is meant to improve navigation in the complex structure of the system functionality.

3. INTELLIGENT INTERFACE

The concept of an intelligent interface is not a new concept. IUI - Intelligent User Interface - is a group of issues well-known in computer science for years. Interest in such systems is due to the still developed database systems whose interfaces are inflexible and ill-suited to user different requirements. Many also are the reasons why the computer application interfaces can be called intelligent:
– interfaces that have knowledge about the functionality of the application,
– interfaces that have knowledge of user preferences,
– self-adaptive interfaces, i.e. the interfaces that on the basis of interaction with the user can adapt themselves to his needs,
– interfaces processing natural language and using semantic analysis.

Creating smart interfaces goes hand in hand with the development of artificial intelligence techniques and uses its achievements. Computer systems are becoming increasingly better adapted to the needs of users, often having more intelligent interface than an application that lies beneath it. At a pace relatively most slow is developing the first of the above mentioned groups of intelligent interfaces, which might be due to the fact that it emerged as the first one and therefore could be treated as a problem already solved. Everyone has come across the infamous MS Office Assistant, which further testifies to the fact that in this field there is still much to be done.

3.1. CBR methodology

The proposed solution for the described interface is to use case-based reasoning to assist the use of the system. User after each use of the system leaves his mark in the form of a given query, selection of individual modules of the system or indications what type of knowledge is needed. Any such interaction with the system is saved as one of the cases. Using the accumulated knowledge about how to interact with the system (cases base), the interface may next time suggest to the user the keywords, potentially the best modules for his problem, or the type of data that are worth to search on a basis of revised former cases that fits the best to the user's query.

The main paradigm of CBR methodology is inference regarding the current problem by using knowledge of the similar problems solved in the past (Aamodt & Plaza, 1994). The main component of the CBR system is a case-base, which contains a collection of problems and solutions (cases) that occurred in the past. When a new problem emerges, CBR system searches the database for similar cases, and then the solution of the case searched is adapted to the current situation. CBR methodology also allows for learning - at any time instant, when the solution of the new case takes place, the knowledge related to this experience is attached to the case-base. Thanks to this, relevant knowledge will be available in subsequent uses of the CBR system to solve the following problems. The CBR methodology can be compared to the process of "scooping" and using experience.

Despite many differences in the construction of various systems using CBR methodology, a common element is the algorithm called CBR cycle. CBR cycle consists of four phases (Rojek et al., 2014):

1. searches for the most similar case or cases,
2. the re-use of information and knowledge contained in the case most similar to solution of the current problem,
3. revision of the proposed solution,
4. preservation of information and knowledge regarding current case together with its solution for use in the future.

CBR cycle starts when there is a new problem to solve - a new user or a new query. Then the four phases shown above are executed sequentially. In the first phase, there is search for the case or group of cases which are most similar to the current problem. Each case stored in the database corresponds to one interaction with the system. The case consists of the keywords specified by the user, possibly indicating modules, which he wants to use and the type (types) of the information sought. Search among the collected cases will be done using a similarity measure that best suits the attributes compared. For example, for the vector of keywords, a cosine distance relative to the vectors of keywords from the previous interactions can be used. In many known applications of CBR methodology, a similarity measure uses a Euclidean or Hamming distance (Kusiak & Rojek, 2012).

In the phase of re-use, the case searched in the previous phase is used to provide the user with hints on the use of the system. This phase can be very simple when the solution related to the searched case is returned as the solution of the current case. For example, it is corresponding to the query result that was obtained in the past results. In many cases, however, an adaptation of the solution is necessary, which may occur by transforming the previous query or using other modules associated with the new keywords.

During revision, the proposed solution is evaluated, and in the case of a negative evaluation, introduction of corrections to the solution is also possible. Evaluation takes place by matching result to the user's needs and is the subjective opinion of the user. The use of an expert is also anticipated. He could serve as a "model user" and provide information on the assessment of individual results, or the necessary amendments. This first phase of the implementation of the system anticipates creating the preliminary case-base consisted of cases created by this "model user". The phase of platform testing provides dozens of examples of the use cases of the system. That is why the case-base is never empty, even at "first use".

The phase of storing information and knowledge is the phase in which learning of the system occurs. This process typically involves a simple addition of the current event (after the revision phase) to the case-base. With this, the knowledge of the problem, which has just been solved, can be used in the future functioning of the system (figure 2).

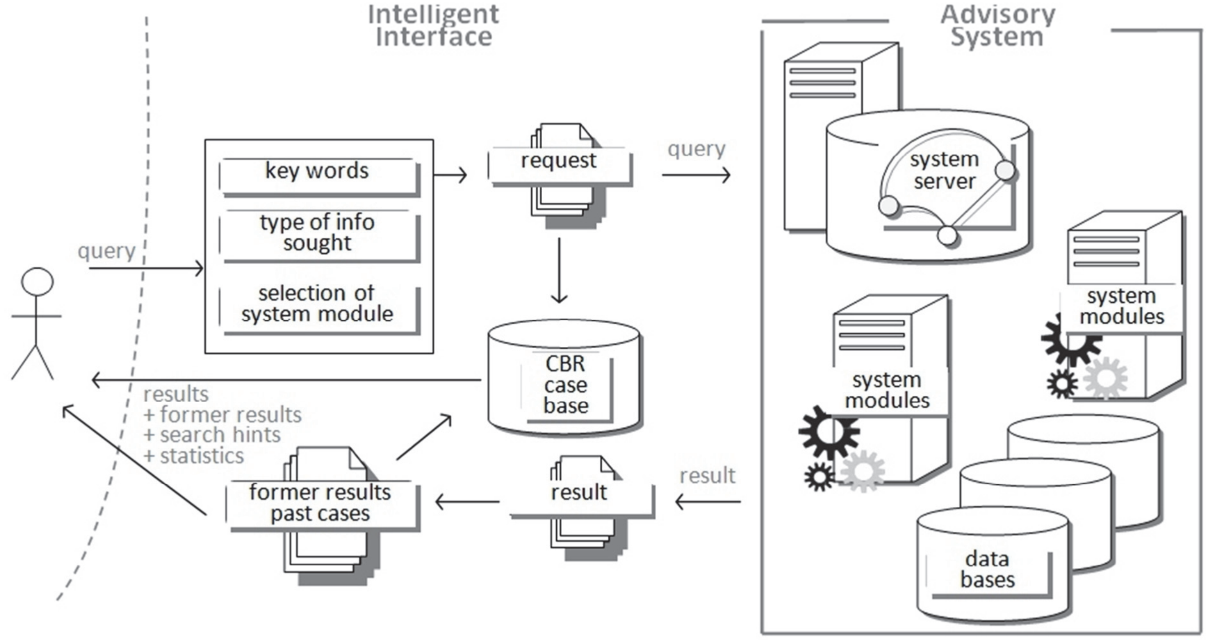

Fig. 2. Schematic flow of information between the user, intelligent interface and a system.

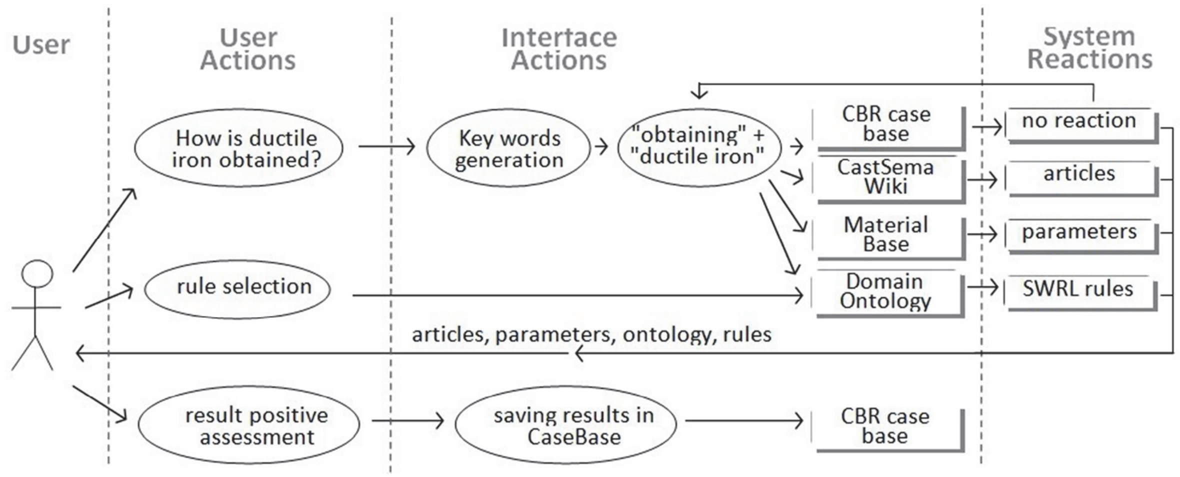

Fig. 3. Diagram of the interface actions referred to the example shown.

Information flow and saving "past cases" to a base forming the above illustrative embodiment shown in figure 3. This diagram clarifies the general concept of the intelligent interface shown in figure 2.

For example, user can ask a question expressed in natural language: "How is ductile iron obtained?". Agent of the interface by analyzing the sentence chooses keywords: "obtaining" + "ductile iron". This query devoid of other attributes, i.e. selected modules and the type of knowledge, is directed to the advisory system. Individual modules return a variety of results, e.g. CastSemaWiki will return an article on ductile iron, in which the user will find an answer to the question posed; the database of material characteristics will indicate the parameters of ductile iron, but in order to get information on how to obtain this material, the user will have to find in a domain ontology a suitable SWRL rule indicating that the grey cast iron subjected to modifications and appropriate heat treatment changes into ductile iron. One can also search the base of publications with the specified keywords, or check whether in the field of ductile iron in recent years some research works have been performed.

The interface stores the actions performed by the user thereby complementing the case of interaction with the type of information and system modules used. When re-call of this query occurs, the system may propose further proven steps of search.

4. SUMMARY

This paper describes a proposal to apply the techniques of inference drawn from the studies of artificial intelligence to support user interface design in large systems of decision support in the field of metallurgical technology and metal processing. Complex systems can be a valuable tool for the user who is an employee of industry, but the level of complexity in their use can be a psychological barrier to their widespread application. Designing of intelligent interfaces that are guides during operation of the system may decide about the success of the operation of information science tools. The proposed solution uses the CBR methodology in the process of building a database of cases of user interaction with the system, in order to give further support and advice in the course of working with the advisory system. User can easily take advantage of the proposed functionalities when his demand for information is not fully specified, or has problems with proper expression of the query. In the literature on information science, the concept of intelligent interface is well-known. Innovative is the use of methodologies in the field of episodic inference to control contact with the advisory system in the field of metal processing. The advantage of the use of a CBR methodology comparing to the machine learning methods (classification methods, Bayesian Networks, etc.) commonly used in IUI ist that the case-base need not necessarily be very complex, especially at the beginning - each "use-case" is represented by one case in base, not like in others methods - by group of cases.

ACKNOWLEDGEMENTS

The work was financed within the framework of the international project No. 820/N-Czechy/2010/0 of 30 November (part: modules of knowledge and databases). Financial support of the National Centre for Research and Development (LIDER/028/593/L-4/12/NCBR/2013) is also gratefully acknowledged (in the part of intelligent interface engine model).

REFERENCES

Aamodt, A., Plaza, E., 1994, Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *AI Commun*, 7, 39-59.

Dobrowolski, G., Kluska-Nawarecka, S., Wilk-Kolodziejczyk, D., Regulski, K., 2011, Rough Sets Applied to the RoughCast System for Steel Castings, Intelligent Information and Database Systems, *Proceeding of ACIIDS 2011 Conference*, eds, Nguyen, N.T., Kim, Ch-G., Janiak, A., 6592, 52-61.

Gurda, M., Lesniak, G., Krzyzak, M., Regulski, K., Kluska-Nawarecka, S., 2013, System of Semantic Integration of Non-structuralized Documents in Natural Language in the Domain of Metallurgy, *Arch Metall Mater*, 58, 927-930.

Haratym, A., Regulski, K., Kluska-Nawarecka, S., Nawarecki, E., Dobrowolski, G., 2013, The Platform for Semantic Integration and Sharing Technological Knowledge on Metal Processing and Casting, *Computer Methods in Materials Science*, 13, 304–312.

Kluska-Nawarecka, S., Pirowski, Z., Jančiková, Z., Vrozina, M., David, J., Regulski, K., Wilk-Kołodziejczyk, D., 2013, A User-inspired Knowledge System for the Needs of Metal Processing Industry, *Computer Methods in Materials Science*, 13, 295–303.

Kusiak, J., Rojek, G., 2012, Case-based Reasoning Approach to Control of Industrial Processes, *Computer Methods in Materials Science*, 12, 250–258.

Nawarecki, E., Kluska-Nawarecka, S., Regulski, K., 2012, *Multi-aspect Character of the Man – Computer Relationship in a Diagnostic-advisory System*, Human – Computer Systems Interaction: Backgrounds and Applications 2, eds, Hippe, Z.S., Kulikowski, J.L., Mroczek, T., Berlin, Springer-Verlag, 85-102.

Opaliński, A., Turek. W., Kiesiel-Dorohinicki, M., 2011, Extensible Web Crawler – Towards Multimedia Material Analysis, *Proceedings of 4th International Conference on Multimedia Communications, Services and Security MCSS 2011,* eds, Dziech, A., Czyżewski, A., Springer-Verlag, Communications in Computer and Information Science, Kraków, 183–190.

Regulski, K., Kluska-Nawarecka, S., 2012, *Knowledge Integration Computer Tools and Algorithms in the Improvement of the Production Processes of Cast-steel Castings*, Artificial Intelligence in the Knowledge and Information Systems, Foundry Research Institute, Kraków.

Rojek, G., Regulski, K., Dobrowolski, G., 2014, Computer System of Diagnosis of Casting Defects using Case-based Reasoning Methodology, *Hutnik Wiadomości Hutnicze*, 81, 72–75 (in Polish).

Wilk-Kołodziejczyk, D., Dajda, J., Macura, M., Regulski, K., Kluska-Nawarecka, S., 2014, Computer-assisted Integration of Knowledge in the Context of Identification of the Causes of Defects in Castings, *Arch Metall Mater*, 59, 744–746.

INTELIGENTNY INTERFEJS DLA SYSTEMU WSPOMAGANIA DECYZJI W OBSZARZE METALURGII

Streszczenie

Rozległy system wspomagania decyzji złożony z kilku modułów zapewniających różne funkcjonalności użytkownikowi stanowi wyzwanie w projektowaniu użytecznego, prostego i spójnego interfejsu. Specyfiką systemów doradczych w zakresie metalurgii jest stopień skomplikowania ich architektury zaprojektowanej celem wspierania różnorodnych typów zadań w wielu obszarach. Celem tego artykułu jest prezentacja możliwości stworzenia inteligentnego interfejsu zapewniającego użytkownikowi pełną swobodę działania nawet w sytuacji, gdy nie zna on dokładnie wszystkich funkcjonalności systemu. Artykuł przedstawia architekturę tworzonego systemu wspomagania decyzji w zakresie metalurgii wzbogaconą o opis mechanizmu pozwalającego skonstruować inteligentny interfejs. Interfejs taki powinien wspomagać użytkownika w nawigacji po modułach systemu tak, aby użycie systemu było najefektywniejsze. Do tworzenia takiego mechanizmu zaproponowano metodykę wnioskowania epizodycznego (Case-Based Reasoning, CBR). CBR stanowi wzorzec tworzenia systemów uczących się na podstawie przeszłych zachowań użytkownika związanych z wykorzystaniem systemu w rozwiązywaniu problemów. Wiedza o poprzednich zachowaniach i wyborach użytkownika służy do wspomagania bieżących problemów. Każde zapytanie użytkownika skierowane do systemu oraz każdy jego wybór użycia poszczególnych funkcjonalności modułów systemu wzbogaca system o cenną wiedzę: właściwe wykorzystanie całego systemu doradczego związane z potrzebami użytkownika. Umiejętne wykorzystanie tej wiedzy zapewni wygodną i efektywną pracę inteligentnego interfejsu.

Received: October 1, 2014
Received in a revised form: November 21, 2014
Accepted: November 24, 2014