Computer Assisted Mechanics and Engineering Sciences, 14: 753–765, 2007.

Integration of a case-based reasoning and an ontological knowledge base in the system of intelligent support of finite element analysis

Peter Wriggers
Leibniz University Hannover, Institute of Mechanics and Computational Mechanics
Appelstr. 9A, 30167 Hannover, Germany

Marina Siplivaya
Volgograd State University of Architectural and Civil Engineering, IT Department
Akademitcheskaya 1, 400074, Volgograd, Russia

Irina Zhukova, Alexander Kapysh, Anton Kultsov
Volgograd State Technical University, Department of Software Development
Lenina av. 28, 400131, Volgograd, Russia

(Received in the final form September 18, 2007)

The process of engineering analysis, especially its preprocessing stage, comprises some knowledge-based tasks which influence the quality of the results greatly, require considerable level of expertise from an engineer; the support for these tasks by the contemporary CAE systems is limited. Analysis of the knowledge and reasoning involved in solving these tasks shows that the appropriate support for them by an automated system can be implemented using case-based reasoning (CBR) technology and ontological knowledge representation model. In this paper the knowledge-based system for intelligent support of the preprocessing stage of engineering analysis in the contact mechanics domain is presented which employs the CBR mechanism. The knowledge representation model is formally represented by the OWL DL ontology. Case representation model, case retrieval and adaptation algorithms for this model and the automated system are described.

Keywords: artificial intelligence, case-based reasoning, ontologies, finite element analysis

1. Introduction

Modern engineering analysis employs sophisticated and evolving computational methods. While these methods are efficient for a variety of real-world problems, their application can be really complicated for an engineer. The process of engineering analysis comprises non-algorithmic knowledge-based tasks, which significantly affect accuracy and reliability of the obtained results. Contemporary engineering analysis tools effectively implement computational methods, but provide little or no support for the non-algorithmic tasks which are carried out by engineers based on their experience. These factors increase time estimations and costs of engineering analysis and the need for highly-qualified specialists, and also restrict the transfer of engineering analysis-related research achievements into industry. Thus an important problem is to provide support for non-algorithmic knowledge-based tasks of the engineering analysis process which can be implemented using the appropriate artificial intelligence (AI) technologies.

The study of the process of solving engineering problems was performed and the concept of intelligent support of finite element analysis in contact mechanics was formulated [19].

By this concept case-based reasoning (CBR) was selected as the appropriate AI technology for the intelligent support of engineering analysis. The study of the existing approaches to CBR [1, 3, 5, 7, etc.], including case representation models, retrieval and adaptation algorithms, shows that they do not fully satisfy the domain requirements. Therefore tasks for the development of appropriate knowledge representation models and reasoning algorithms to support the CBR processes within the framework of the intelligent support of engineering analysis were formulated [19].

2. State-of-the-art

Some of the tasks within the FEM analysis process are of algorithmic nature — meshing (assuming that all the parameters of the mesh are specified), the FEM computation itself, and visualization of results. Other tasks are non-algorithmic and imply a knowledge-based approach: problem classification, making decisions about computation model parameters (geometry simplifications, finite elements type and size, material model etc) and numeric algorithm type and parameters, evaluation and interpretation of the numeric results.

Algorithmic tasks are effectively automated by analysis subsystems of modern CAD systems and finite element analysis systems (ABAQUS, I-DEAS, ANSYS, MSC.NASTRAN-based systems etc) and stand-alone tools for specific tasks (meshing tools etc) or domains. While AI technologies are widely applied in the fields of conceptual design, diagnostic and fault detection, relatively little work has been done to automate and/or support non-algorithmic tasks of the engineering analysis which imply heuristic classification and decision making. In particular, the following research can be mentioned: GENIUS expert system for controlling the iterative FEM computation process [18]; REAP methodology which aims at FE model construction on the basis of modular reuse of fragments of existing models [9]; IPA – "the designer's personal assistant" on the base of decomposition and coordination of multicriteria decision problems, evolved through cooperation with designers in field of engineering [15]; BPA – a best practice advice system to support automotive engineering analysis processes [13]. Contemporary AI technologies can be used to represent knowledge and model reasoning processes employed in solving non-algorithmic tasks of FEM-based engineering analysis. A system for intelligent support of FEM-based engineering analysis can be designed on the basis of these technologies. The current state of the art for intelligent support of FEM-based engineering analysis in the field of contact mechanics has several aspects open to further investigation:

- development of an ontology-based knowledge base in the field of contact mechanics comprising knowledge about mechanisms, formal contact problems and solution techniques;
- development of advanced CBR algorithms considering the features of domain;
- integration of case-based reasoning and ontological knowledge base.

In this paper we present an automated system for intelligent support of the preprocessing stage of FEM-based engineering analysis in the field of contact mechanics using ontology for knowledge representation and case-based reasoning.

3. Concept of the integration of case-based reasoning and ontology

The processes involved in CBR can be represented by a schematic cycle. Aamodt and Plaza (1994) have described CBR typically as a cyclical process comprising the four REs: retrieve the most similar case(s); reuse the case(s) to attempt to solve the problem; revise the proposed solution if necessary, and retain the new solution as a part of a new case. A new problem is matched against cases in the case base and one or more similar cases are retrieved. A solution suggested by the matching cases is then reused and tested for success. Unless the retrieved case is a close match the solution will probably have to be revised producing a new case that can be retained.

CBR approach is aimed for avoidance of first order knowledge use for problem solving, so support of CBR with ontology could be interpreted as disloyalty to grounding principles of CBR. But it is not completely correct and ontologies are used in various ways to make accurate and useful CBR possible. Generally case base does not comprise all knowledge required for correct solution of a new problem, but ontology allows to formalize domain knowledge and to reuse it for problem solving. Also ontology allows achieving semantic interoperability between case descriptions from different sources. Ontology could define framework for case description which allows simplifying case retrieving stage of CBR process. And finally it is recognized that general domain knowledge is necessary at the stage of case adaptation.

Employment of ontology coupled with a CBR process is possible in few ways:

- ontology describes general concepts and rules of domain;
- ontology describes general concepts and rules of domain and case representation model;
- ontology describes general concepts and rules of domain, case representation model and contains case base.

Consequently we decided to use a third model from the listed above since this model allows using ontology for supporting all stages of CBR process:

- ontology defines case representation model and sets of parameters for case description with their possible values. Ontology contains information about possible aims and features of analysis for all particular types of concerned mechanisms. Ontology defines foundation for case description, and it could be used as knowledge base for subsystem of automated case formulation from textual and graphical descriptions of problem and its solution;
- if the user has defined a case index incompletely, ontology can be applied at the stage of case retrieval for redefinition of a case. Also the algorithm of similarity measure calculation depends heavily on the case representation model, which is defined by ontology;

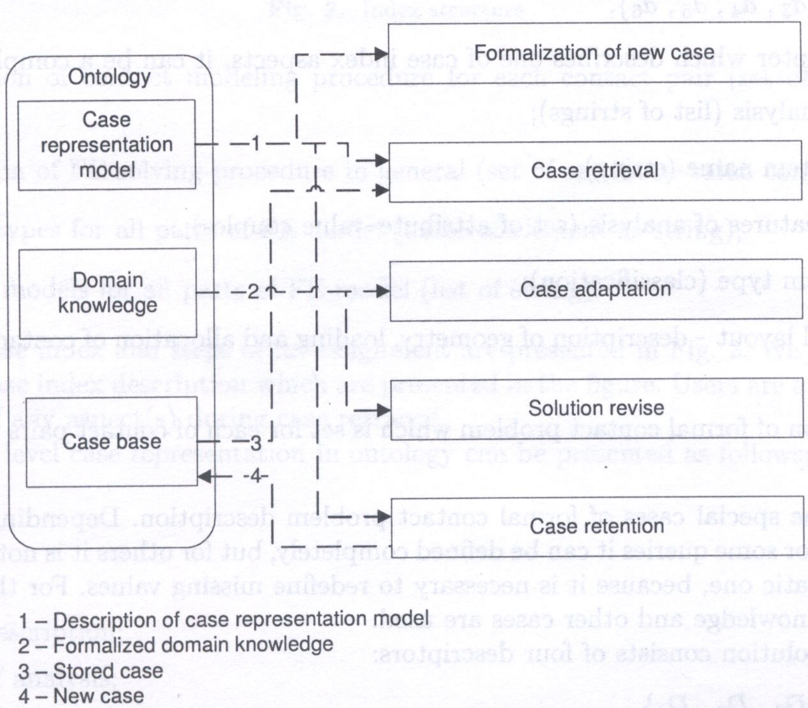

Fig. 1. Schema of the integration of CBR and ontology

- ontology is used at the stage of case adaptation to make a decision about alternative values of parameters if adaptation algorithm proposed few equivalent choices, for example, few possible values for one attribute;
- ontology contains rules which can be used at the stage of revising. These rules define infeasible combinations of parameter values for different aspects of case description.

The general scheme of the integration of CBR and ontology is shown in Fig. 1. Ontology as case representation model describes case as set of individuals and relations. Ontology as domain knowledge base describes domain knowledge as DL rules, using procedures of knowledge inferring. Ontology as case base stores cases as a part of domain knowledge.

4. Case representation model

There are many models of case representation, in close to engineering analysis domains such models are used as textual representation [4] and structural representation [2]. There are hash table (attribute–value couples), object-oriented and graph models among structural. Structural objected-oriented model was selected as case representation model. This approach is useful at the concerned domain, because besides case it is necessary to have additional domain knowledge to obtain satisfactory results. Domain knowledge model also allows controlling quality of new cases which should be added to the case base, accordingly, it allows decreasing maintain costs substantially. The main disadvantage of this approach is a necessity of intense effort for development of domain knowledge model.

Case description consists of two main parts,

$$\text{Case} = \{P, S(P)\}, \tag{1}$$

where P – problem description (case index), $S(P)$ – problem solution description.

Problem description consists of six descriptors which reflect the main aspects of case problem definition:

$$P = \{d_1, d_2, d_3, d_4, d_5, d_6\}, \tag{2}$$

where d_i – descriptor which describes one of case index aspects, it can be a complex structure:

d_1 – aim of analysis (list of strings);

d_2 – CAE system name (string);

d_3 – special features of analysis (set of attribute–value couples);

d_4 – mechanism type (classification);

d_5 – structural layout – description of geometry, loading and allocation of contact pairs (semantic network);

d_6 – description of formal contact problem which is set for each of contact pairs (set of attribute–value couples).

There are some special cases of formal contact problem description. Depending on competence level of the user, for some queries it can be defined completely, but for others it is not possible. Second case is a problematic one, because it is necessary to redefine missing values. For this procedure the general domain knowledge and other cases are used.

Definition of solution consists of four descriptors:

$$S(P) = \{D_1, D_2, D_3, D_4\}, \tag{3}$$

where a descriptor D_i describes one of case solution aspects. It can have a complex structure:

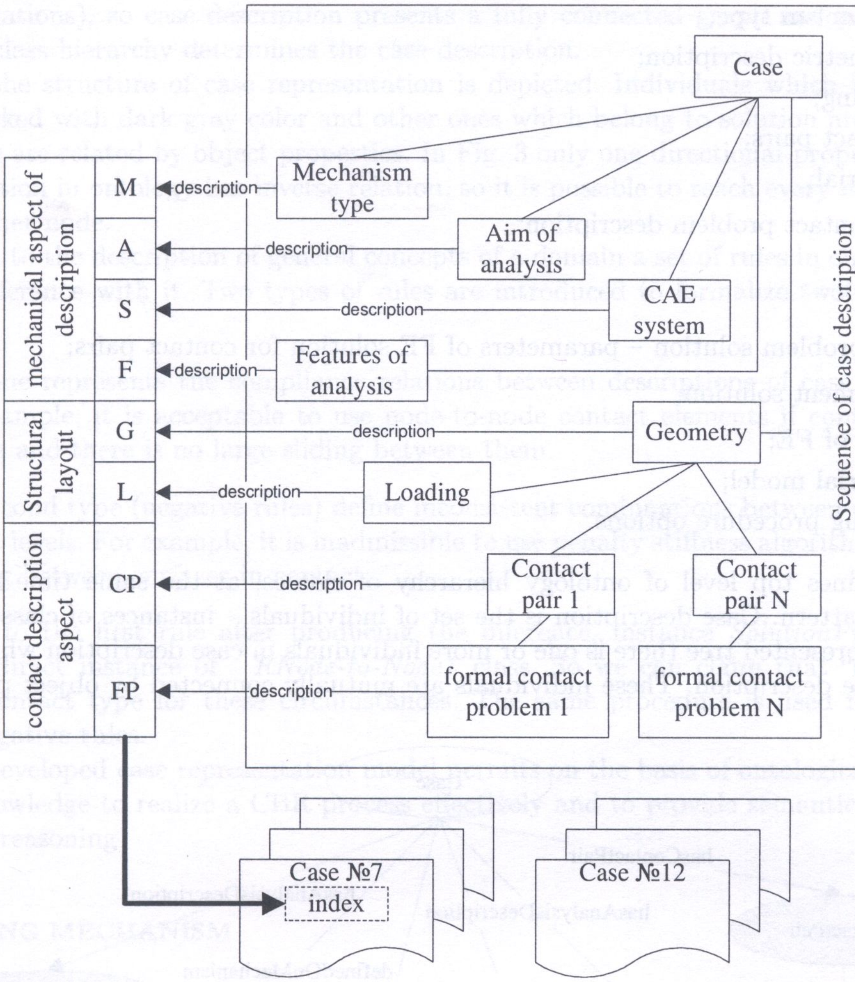

Fig. 2. Index structure

D_1 – description of contact modeling procedure for each contact pair (set of attribute–value couples);

D_2 – description of FE solving procedure in general (set of attribute–value couples);

D_3 – element types for all parts of FE model (name of element as string);

D_4 – material models for all parts of FE model (list of string).

Structure of case index and steps of its assignment are presented in Fig. 2. We picked out three main aspects of case index description which are presented in the figure. Users are able to emphasize the importance of any aspect(s) during case retrieval.

So, on the top level case representation in ontology can be presented as follows:

- Case name;
- Case index;
  - Analysis description;
    - Aim of analysis;
    - Special properties;
  - Physical object description;

    * Mechanism type;
    * Geometric description;
    * Loading;
    * Contact pairs;
    * Material;
  – Formal contact problem description;

• Case solution;
  – Contact problem solution – parameters of FE solution for contact pairs;
  – Finite element solution;
    * Type of FE;
    * Material model;
    * Solving procedure options.

This tree defines top level of ontology hierarchy of classes, at the same time it defines case representation pattern. Case description is the set of individuals – instances of classes in ontology. For each leaf of presented tree there is one or more individuals in case description which defines one of aspects of case description. These individuals are mutually connected by object properties (two

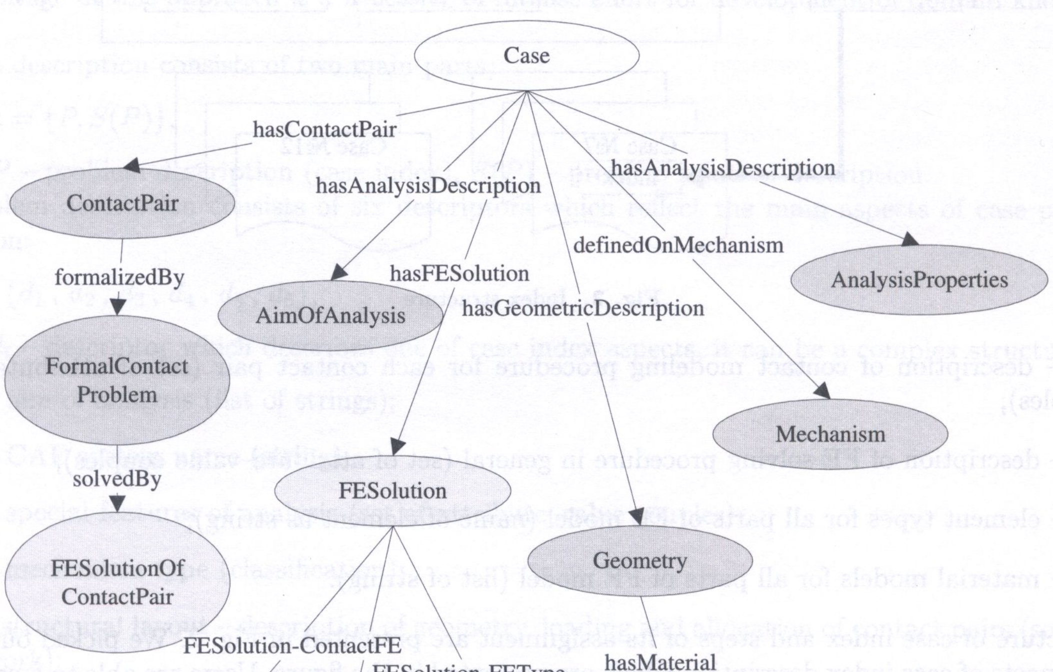

Fig. 3. General case structure

directional relations), so case description presents a fully connected graph and positioning of the nodes at the class hierarchy determines the case description.

In Fig. 3 the structure of case representation is depicted. Individuals which belong to a case index are marked with dark gray color and other ones which belong to solution are gray-colored.

Individuals are related by object properties. In Fig. 3 only one-directional properties are shown, but every relation in ontology has inverse relation, so it is possible to reach every node of the graph from every other node.

In addition to the description of general concepts of a domain a set of rules in ontology is defined to produce inference with it. Two types of rules are introduced to formalize two types of domain knowledge:

- The first one represents the compliance relations between descriptions of case at different levels. For example, it is acceptable to use node-to-node contact elements if contact surfaces are conforming and there is no large sliding between them.
- Rules of second type (negative rules) define inconsistent combinations between case descriptions at different levels. For example, it is inadmissible to use penalty stiffness algorithm if large sliding is occurring between surfaces in contact.

According to the first rule after producing the inference, instance *SolutionProperties* will be defined as a direct instance of *_RNode-to-Node1* class. So we can claim that "node-to-node" is appropriate contact type for these circumstances. The same procedure is used for classes which implement negative rules.

Thus the developed case representation model permits on the basis of ontological representation of domain knowledge to realize a CBR process effectively and to provide semantic interoperability of case-based reasoning.

5. Reasoning Mechanism

The following algorithms were developed or modified to operate on the proposed case representation model in the framework of the case-based reasoning mechanisms implementation:

1. Case retrieval algorithms:
   - 1.1. General algorithm of similarity measure calculation;
   - 1.2. Class-based similarity (CBS) algorithm for calculation of property similarity;
   - 1.3. Slot-based similarity (SBS) algorithm of maximal matching calculation
2. Case adaptation algorithms:
   - 2.1. Adaptation of contact pair solutions;
   - 2.2. Adaptation of general solution
3. Case revising and retaining algorithms.

5.1. Case retrieval algorithms

For case retrieval the CBS and SBS algorithms based on ontology representation of a case were developed [19]. These algorithms are used for similarity measure calculation for all aspects of case description. CBS algorithm calculates similarity measure between two individuals of ontology on the basis of their positioning in concept hierarchy of ontology. SBS algorithm calculates similarity measure on the basis of comparison of case description subgraphs formed by individuals and roles in ontology.

Calculation of similarity between different individuals is carried out during calculation of total similarity measure between cases. Both, CBS and SBS algorithms are used for calculation of similarity between individuals,

$$S(i_1, i_2) = \frac{k_1 * CBS(i_1, i_2) + k_2 * SBS(i_1, i_2)}{k_1 + k_2}, \tag{4}$$

where k_1, k_2 – weighting coefficients; $\mathrm{CBS}(i_1, i_2)$ – CBS-similarity between individuals i_1 and i_2; $\mathrm{SBS}(i_1, i_2)$ – SBS-similarity between individuals i_1 and i_2.

In the framework of the proposed ontological case representation model, one individual may belong to the several classes of attribute values. The basic CBS algorithm [19] being applied to similarity calculation of individuals, which belong to several classes of attribute values with different depth in class hierarchy, gives incorrect value of similarity. Hence a modified CBS algorithm for calculation of similarity between values of attributes for all individuals was developed. It is based on definition of attribute vectors for each attribute individually and calculation of CBS-similarity as a weighted combination of these vectors and permits to obtain correct values of similarity for all individuals.

The schema of modified CBS algorithm of attribute vectors composition is presented in Fig. 4.

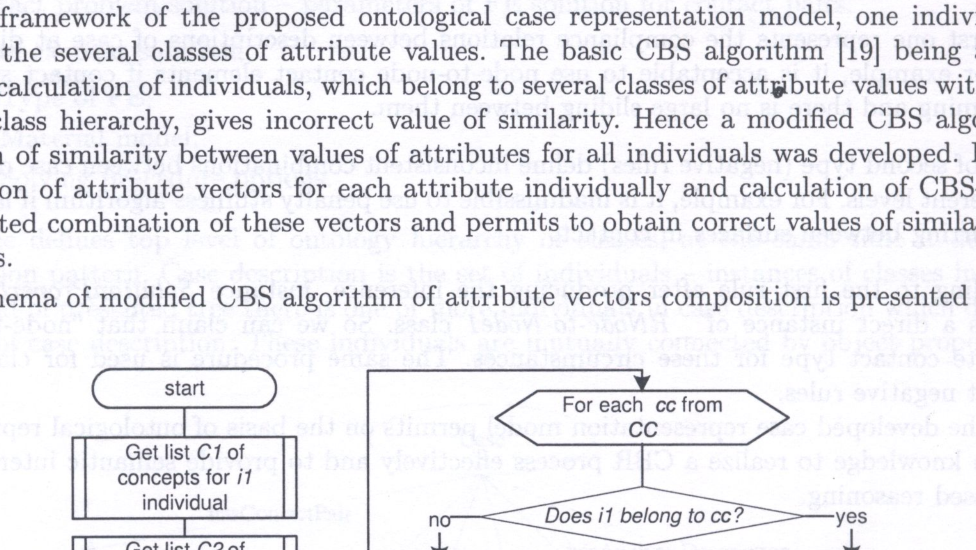

Fig. 4. Algorithm of attribute vectors composition

Case base contains two different types of cases–cases of contact problem solving and cases of mechanism analysis. It allows to perform a more accurate adaptation of a query case. Hence, the basic SBS algorithm [19] was modified to take account of difference in description of these two types of cases. The main idea of modified SBS algorithm is to find not general matching between components of cases but most appropriate mapping of case components from case base into components of query case. The main modification of algorithm is calculation of average similarity not at all components of query case and case from case base but only at matched components. This approach permits to match cases with different number of physical parts or contact pairs.

Hence the modified algorithm differs from the original [19] only by stage of M-list formulation. The schema of modified stage of M-list formulation is shown in Fig. 5.

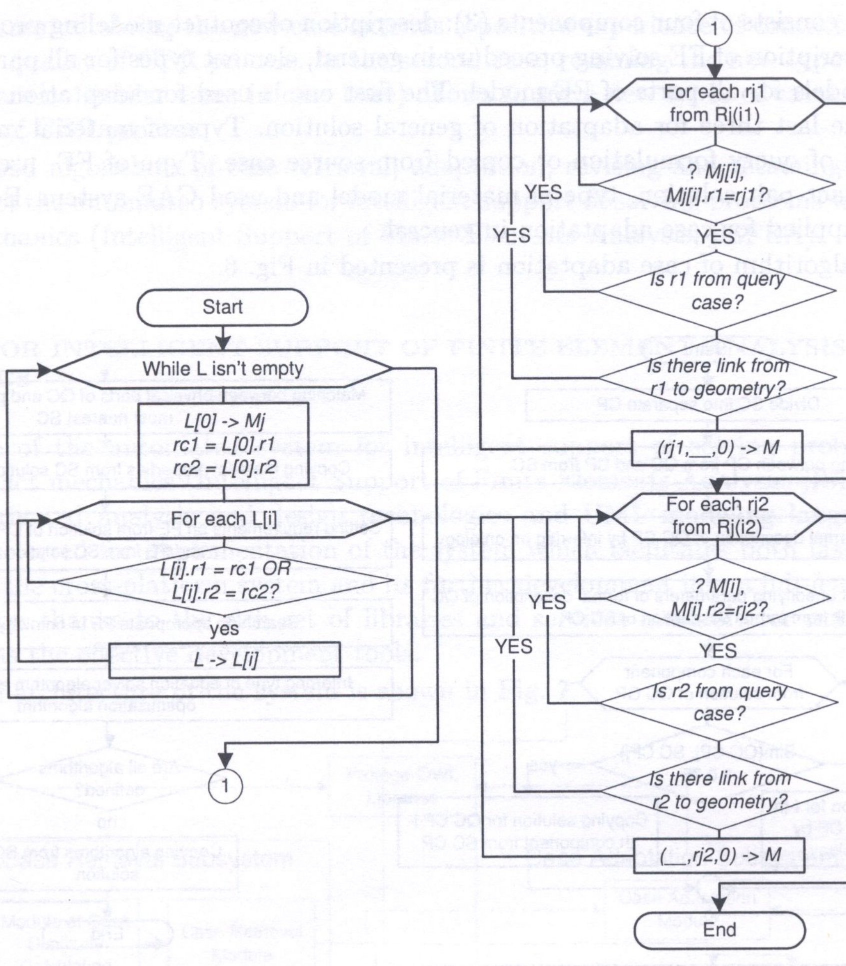

Fig. 5. Schema of M-list forming stage of modified SBS-algorithm

5.2. Case adaptation algorithms

For adaptation of solution to query case we used generative adaptation [12], it allows to use formalized domain knowledge about dependencies between case solution and case description. Case adaptation includes adaptation of contact pair solutions and adaptation of general solution.

General schema of adaptation of contact pair solution is described by following algorithm:

1. initial matching of contact pairs for query case and source cases;
2. redefinition of contact pairs formalization;
   (a) inferring of values of formal properties;
   (b) reuse of formal properties from source cases;
3. adaptation of solution for contact pairs;
   (a) copy of values of solution properties with more than threshold value similarity;
   (b) generation of solution property values;
   (c) copy of solution properties values without use of threshold value.

Case solution consists of four components (3): description of contact modeling procedure for each contact pair, description of FE solving procedure in general, element types for all parts of FE model and material models for all parts of FE model. The first one is used for adaptation of contact pair solutions and the last three for adaptation of general solution. Type of material model is defined by user at stage of query formulation or copied from source case. Type of FE, used for meshing, depends on contact pair solution, type of material model and used CAE system. Both adaptation procedures are applied for case adaptation in general.

The general algorithm of case adaptation is presented in Fig. 6.

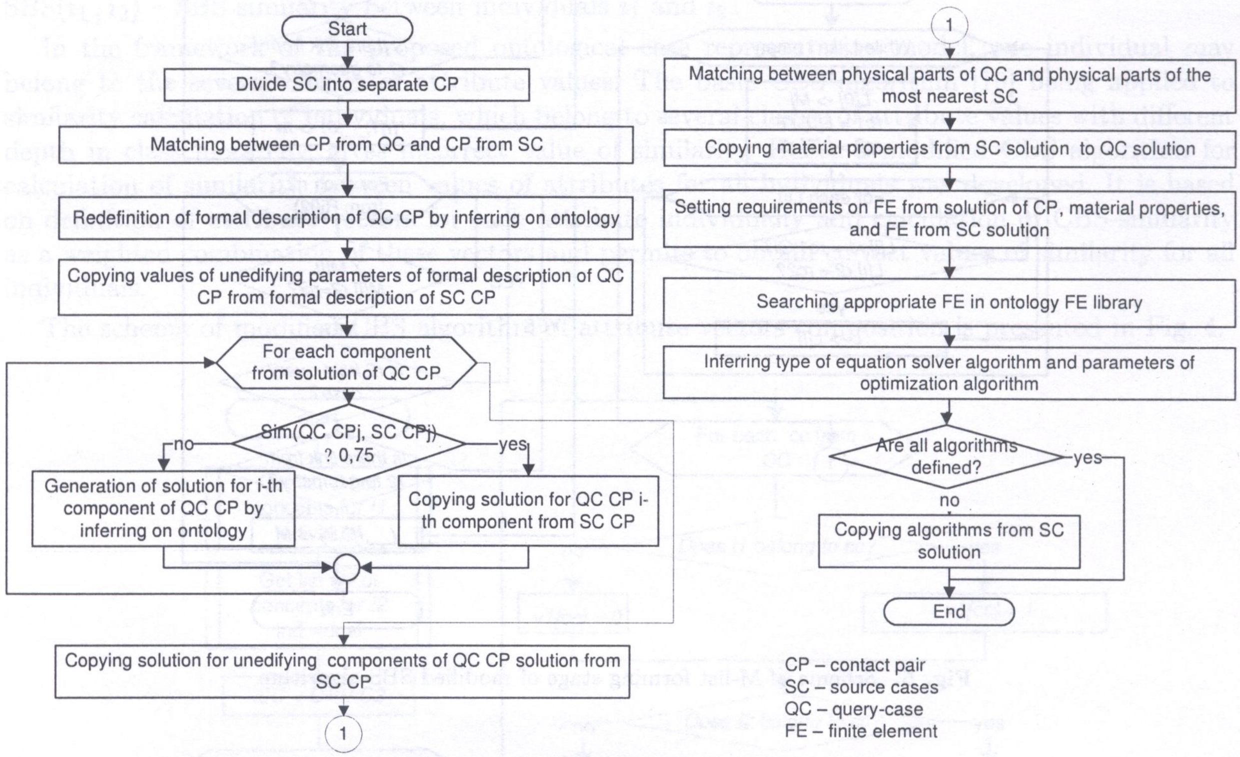

Fig. 6. General algorithm of case adaptation

5.3. Case revising and retaining

Consistency and completeness of case solution and need of new case retaining are revised at the last stages of CBR process.

Consistency of case solution is revised using formalized description logic (DL) rules in ontology. Testing the new case solution is performed by implementation of standard procedures of knowledge inferring with ontology – "classify taxonomy" and "compute inferred types". If an individual of a case solution is determined as instance of one of forbidden classes then case solution is treated as inconsistent. After inconsistency detection it's possible to give an explanation of the probable cause of inconsistency and its description.

Decision about new case retaining should be made by the user, but system can give an advice about possibility of retaining the new case. A new solved case can be saved into case base, if:

- solution of new case is consistent, or;
- the nearest case from case base has similarity measure less than some threshold.

In the first case, retaining the new case extends a positive experience of contact problem solving and refines the quality of CBR process. In the second case, retaining the new type of case (it's new because there are no similar cases in case base) allows to solve new types of cases and significantly refines quality of CBR process.

The developed algorithms of case retrieval, adaptation, revising and retaining were realized in the prototype of the automated system for intelligent support of solving problems with FEM related to contact mechanics (Intelligent Support of Finite Elements Analysis, jISFEA).

6. System for intelligent support of finite element analysis in contact mechanics

The prototype of the automated system for intelligent support of solving problems with FEM related to contact mechanics (Intelligent Support of Finite Elements Analysis, jISFEA) is designed using object-oriented analysis and design technologies and UML modeling language. The Java2 technology is selected for implementation of the system which facilitates both fast and convenient prototyping of the cross-platform system and its further development into a full-featured enterprise-scale application thanks to the rich set of libraries and services available for many programming tasks, as well as the effective development tools.

The general architecture of the system is shown in Fig. 7.

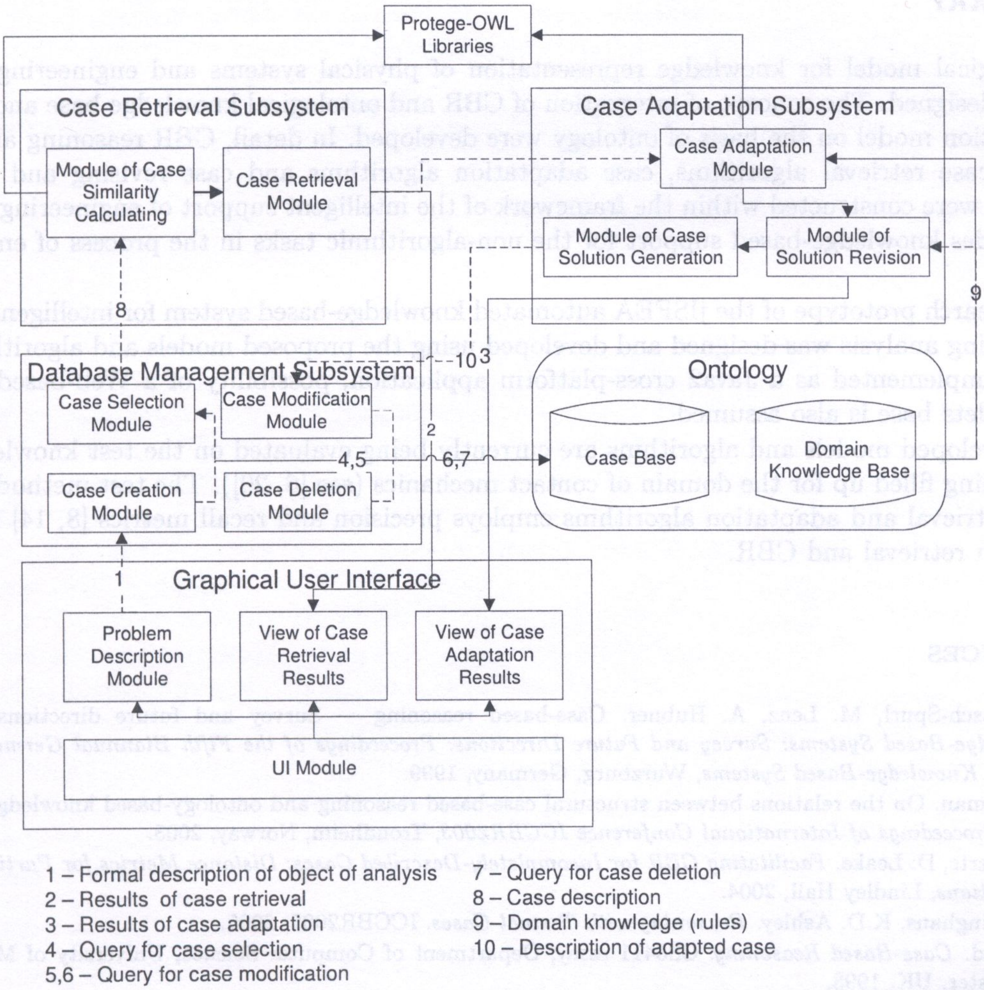

Fig. 7. The general architecture of the jISFEA system

The 3-tier architecture 'Data–Logic–Interface' is used while designing the jISFEA system. It is regarded as the most appropriate for development of extensible and scalable applications. Groups of specialized classes ('Data/knowledge base access module' and 'User interface support module') are introduced to serve as interfaces between the architecture levels; such an architecture facilitate mutual independence of implementations of these levels which is useful both for research-oriented prototyping and for further development into a full-featured application.

In particular, two variants of implementation of data storage and user interface are assumed.

The interface level is assumed to be implemented as Web interface using the applets and window interface with Swing components. The former variant allows potentially to provide broad access to the knowledge base of the system for engineering and research communities (to use and extend the engineering cases collection) able to operate without installation on any 'thin client'. At the same time the latter variant can be simpler and faster implemented because of the efficient support by modern development environments and therefore is chosen for the current implementation of the jISFEA prototype.

The data storage is implemented on the basis of XML documents collection in OWL format using the Protégé OWL libraries for management. This variant provides opportunities for efficient use of standard reasoning mechanisms of description logic (DL) for optimal implementation of the developed CBR and related algorithms.

The logic layer is comprised by the developed CBR algorithms presented above.

7. Summary

An ontological model for knowledge representation of physical systems and engineering analysis cases was designed. The concept of integration of CBR and ontological knowledge base and the case representation model on the basis of ontology were developed. In detail, CBR reasoning algorithms including case retrieval algorithms, case adaptation algorithms and case revising and retaining algorithms were constructed within the framework of the intelligent support of engineering analysis. This provides knowledge-based support for the non-algorithmic tasks in the process of engineering analysis.

The research prototype of the jISFEA automated knowledge-based system for intelligent support of engineering analysis was designed and developed using the proposed models and algorithms. The system is implemented as a Java2 cross-platform application, possibility of a Web-based solution on the applets base is also assumed.

The developed models and algorithms are currently being evaluated on the test knowledge base which is being filled up for the domain of contact mechanics (see [6, 20]). The test methodology for the case retrieval and adaptation algorithms employs precision and recall metrics [8, 14] known in information retrieval and CBR.

References

[1] B. Bartsch-Spurl, M. Lenz, A. Hubner. Case-based reasoning — Survey and future directions. *XPS-99: Knowledge-Based Systems: Survey and Future Directions: Proceedings of the Fifth Biannual German Conference on Knowledge-Based Systems*, Wurzburg, Germany, 1999.

[2] R. Bergman. On the relations between structural case-based reasoning and ontology-based knowledge management. *Proceedings of International Conference ICCBR2003*, Trondheim, Norway, 2003.

[3] S. Bogaerts, D. Leake. *Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions*, Lindley Hall, 2004.

[4] S. Bruninghaus, K.D. Ashley. *Reasoning with Textual Cases.* ICCBR2005, 2005.

[5] A. Broad. *Case-Based Reasoning.* CS3411 essay, Department of Computer Science, University of Manchester, Manchester, UK, 1998.

[6] Q. Feng, N.K. Prinja. *NAFEMS Benchmark Tests for Finite Element Modelling of Contact, Gapping and Sliding.* NNC Ltd, Manchester, UK, 2002.

[7] M. Gomez-Albarran, P. Gonzales-Calero, B. Diaz-Agudo, C. Fernandez-Conde. *Modeling the CBR Life Cycle Using Description Logics*, 1999.

[8] P. Gonzales-Calero, B. Diaz-Agudo, M. Gomez-Albarran. *Applying DLs for Retrieval in Case-Based Reasoning*, 2000.

[9] Ch.H. Hsiung. *A Reusable Engineering Analysis Preprocessing Methodology to Support Design-Analysis Integration*, 2002.

[10] A. Konter. *Advanced Finite Element Contact Benchmarks: A FENET report.* NAFEMS Ltd., Manchester, UK, 2005.

[11] A. Konter. *How To Undertake a Contact and Friction Analysis.* NAFEMS Ltd., Manchester, UK, 2000.

[12] P.K. McKinley, S.M. Sadjadi, E.P. Kasten, B.H.C. Cheng. *A Taxonomy of Compositional Adaptation.* Springer, Madrid, 2002.

[13] C.A. McMahon, Y. Liu, R. Crossland, D. Brown, D. Leal, J. Devlukia. A best practice advice system to support automotive engineering analysis processes. *Engineering with Computers*, **19**: 271–283, 2004.

[14] S. Petrovic, G. Kendall, Y. Yang. A tabu search approach for graph-structured case retrieval. *Starting Artificial Intelligence Researches Symposium*, Lyon, France, 2002.

[15] J. Pokojski. *IPA: Concepts and Applications in Engineering.* Springer, Berlin-Heidelberg, Germany, 2004.

[16] P. Wriggers. *Computational Contact Mechanics.* Springer, Berlin–Heidelberg, Germany, 2004.

[17] P. Wriggers. *Finite Element Algorithms for Contact Problems.* IM TH Darmstadt, Darmstadt, Germany, 1995.

[18] P. Wriggers, N. Tarnow. *Interactive Control of Nonlinear Finite Element Calculations by an Expert System.* IBNM, University of Hannover, Hannover, Germany, 1989.

[19] P. Wriggers, M. Siplivaya, I. Joukova, R. Slivin. Intelligent support of engineering analysis using ontology and case-based reasoning. *Engineering Application of Artificial Intelligence*, **20**: 709–720, 2007.

[20] G. Zavarise, P. Wriggers, U. Nackenhorst. *A Report of Leonardo Project NUFRIC.* TCN, Bergamo, Italy, 2006