

Knowledge-based calculus: a logical-computational framework for engineering theories

I. A. Budyachevsky & V. A. Vittikh

*Samara Division of Mechanical Engineering Research Institute,
Russian Academy of Sciences, 61 Sadovaya st., Samara, 443020,
Russia*

e-mail: iab@sdmeri.samara.su

Abstract

To develop the engineering theories [1] it is proposed the concept of knowledge-based calculus (KBC), which is a deductive system that realizes object-oriented inference approach. A semantic information from initial setting of a problem while composing and solving of constraint satisfaction problems is widely used during such approach. An example that illustrates applying of KBC during kinematic analysis and synthesis of mechanisms is given.

1 Introduction

Possibilities of estimation of products' behaviour and features at the early stages of design are crucial to an engineer. For example, one of the standard tests of a new automobile prototype is analysis of the acceleration of a vehicle and its fuel consumption during speed-up [2]. The results of the test depend on many factors and processes.

Let us consider the speed-up process of an automobile. The propulsive force on the vehicle resulting in the speed is influenced by torque conversion between engine and wheels. These power losses in drive-train torque transmitting and also dynamical factors should be taken into account. Factors influencing losses may include forces that resist the straight-line motion of the vehicle: aerodynamical drag, mechanical drag, gravitational force, inertial force. Usually there are several models based on these factors and it is necessary to relate these models to each other to investigate a given problem. But it seems very reasonable to connect all factors and diverse models within a single framework. It is possible if a correspondent engineering theory of an automobile

[1] is developed. Such theory should be multi-disciplinary and thus provide a way to develop an integrated design environment. The theory itself consists of relationships between concepts from different domains or fields. The relationships can be fundamental laws, experiment-based relationships (empirical dependencies as curves and tables), functional dependencies (differential and algebraic equations, algebraic inequalities) and logical statements.

To develop a universal model of automobile's speed-up and fuel consumption it is necessary to describe an object in the language of engineering theory.

The terms used in such description comprise the information space of the model and the engineering theory provides a correspondent calculus which is systematic method of treating problems by a special system of notations. So the calculus is considered as a rule-based formal technique for processing of certain knowledge. It allows to compose a distinct description of a class of problems and to form an algorithm of solution for single subclasses of this class [3]. A calculus' axioms and inference rules thus form a deductive system which allows to determine new elements of the calculus through primary and obtained elements [4].

Existing differential calculus, for example, is widely used for description of objects. It is quite feasible within specialized fields and subject domains. However one and the same object is separately studied from many points of view: interacting forces, thermal conductivity, aerodynamics and others. There is a need of a new approach to solve the problems. A universal calculus defined over a given class of problems should be created for solving different problems instead of establishing diverse models for the specific problems. It is possible with the use of methods and tools of knowledge-based systems. This paper introduces the concept of a knowledge-based calculus as a logical-computational framework for engineering theories. The example of its application for solving of engineering problems is presented.

2 Knowledge-based calculus

Let's define the concept of applied computer's calculus (ACC) as a formalization of physical or engineering theory, so that to have a possibility to automatically generate models for the solution of a great number of computational problems in applied domain of this theory, using ACC as a support system for modeling and simulation.

From the other hand, a deductive system, also called a formal calculus, is defined as $D = \langle L, A, R \rangle$, where L is a language for representing the objects of the calculus, A is a set of some initial objects (axioms) and R is a finite set of inference rules for new objects deriving from the given ones [5].

Now in the sphere of AI there has been appeared an approach towards automatic solution of computational problems by means of constraint network's formulation. The constraints are relations over arithmetic terms. For general constraint networks the problem of solution finding is NP-complete, i.e.

solution algorithms have exponential worst-case behaviors. In certain cases, some solution algorithms can have acceptable performances. Networks of constraints are also called “constraint satisfaction problems” (CSPs). Any CSP can have variables with finite or infinite domains and a set of constraints.

Realization of physical or engineering theory on ACC basis must provide automatic CSP formulation and its solution according to a declarative setting of a modeling problem.

Unfortunately, in computer science the approach connected with physical or engineering theories’ realization has not received its name yet, though a whole range of new developments could be referred to it. Therefore we have to use ACC metaphor as a generalized name for this approach.

Our analysis of existing novelties showed the absence of adequate software tools or programming languages for engineering theories’ realization.

Constraint logic programming languages [6, 7] are aimed rather for a separate problems’ solution and do not provide the characteristics of modularity and integrity of knowledge referred to an engineering theory [8].

A whole range of existing systems, which use diverse methods of AI, may be referred to the class of ACC systems. Such systems could be completely based upon a formal system of logical inference - logic-based deductive system (LBDS) or must be approximated by a less restricted system that contains ready patterns of inference as expert knowledge - knowledge-based deductive system (KBDS) or a knowledge-based calculus (KBC).

The experience of LBDS using has shown that such foundation is not feasible for the realization of sufficiently rich computer engineering or physical theory. This approach enables to solve the problems only from a highly specialized applied domain. To some extent, as the examples of the systems that use LBDS approach could be a programming system PRIZ [9] that uses a constructive logic, and a logic-based support system for concurrent engineering SPARK [10] that uses order-sort logic. In these systems a declarative description of initial problem is translated into equivalent logic formulae. The solution of this problem is determined through interactive constraint satisfaction performed by deductive systems. LBDS approach does not allow to realize an essential computer engineering or physical theory because of a semantic information loss while translating into a formal logic language. Another problem of LBDS approach is a great number of axioms generated while translating. In the system PRIZ such translating generates hundreds or thousands of axioms for comparatively ordinary problems. E. Tyugu notes that the total number of axioms within large engineering knowledge bases while using LBDS were expected to be hundreds of millions [8].

The peculiarity of KBDS approach is a wide use of semantic information from initial setting of a problem while CSP formulating and solving. This is realized by inserting a great number of rules into KBDS considering this information.

The examples of the systems that could be referred to KBDS are (i) the system ANMEC [11,12] aimed for kinematic and dynamic analysis and synthesis

of planar and spatial mechanisms and constructing with the use of Constraint Oriented Knowledge Base [13]; (ii) the program “The Linkage Assistant” (TLA) [14] focused on kinematic simulation of mechanisms on the basis of Degrees of Freedom Analysis (DFA); (iii) a Constraint Based Design System that uses a geometric constraint solution technique called Operational Transformation Planning (OTP) [15]; (iv) a prototype system for variational geometry [16]. Each of these systems is rule-based and aimed for solving of problems from a certain subject domain.

All KBS systems find a solution to CSP by planning of the operations and then performing them. In this case to solve CSP it is rather used the information from a given subject domain than algebraic information about constraints. The operations are planned within the framework of semantic information and then the computational procedures are bound with them. It enables to find non-iterative, closed-form solutions to CSPs whenever possible and allows to formulate iterative problems with minimal number of unknown variables when closed-form solutions do not exist.

It enables ANMEC, in particular, to provide an export of kinematic or dynamic equations in an analytical form together with numeric computations and animation.

KBDS approach is seemed to be rather promising and requires further development to extend it upon different subject domains. The paper presents an attempt in this direction.

3 Object-oriented and constraint-based inference approach

For implementation of KBDS we use an approach, called Object-oriented and Constraint-based Inference (OCI), which is based on combining of methods of object-oriented and constraint-based, logic programming techniques.

The main ideas are (i) variables in clauses of inference rules are references to objects; the inference results in creation of new object; (ii) objects are considered in sense of object-oriented programming; (iii) constraints are connected only with objects; (iv) inference is possible only by satisfaction of constraints; (v) the result of inference is a model, which is considered as a consequence of operations that are connected with objects and solutions of their constraints.

Thus the resulting model is considered as a consequence of initial and newly created objects. Constraints of the objects that are satisfied in this consequential manner are thereby included into the model.

In OCI approach the constraints and inference rules belong to the description of objects' classes. Rules of inference are included into an object constructor, which determines conditions of its creation. The constructor can also contain no inference rules. An object of such kind could be created by user for task specifications. If object's constructor contains inference rules, the object is created during inference and can be available for user's manipulations (dynamic object), or not available (virtual object).

This approach is, in fact, a development of forming and using of Constrained-oriented Knowledge Base approach [13].

4 System architecture

OCI approach is a base of development system called Object-oriented and Constraint-based Modeling System (OCMS) (Figure 1). OCMS provides a realization of KBC of engineering or physical theory.

The result of automatic inference on the base of KBC is either a model, or a system of equations, which describes initial task. If the task has any analytical solution, then it can be obtained as one or several formulae. Results of the calculation model can be graphs, tables, animations.

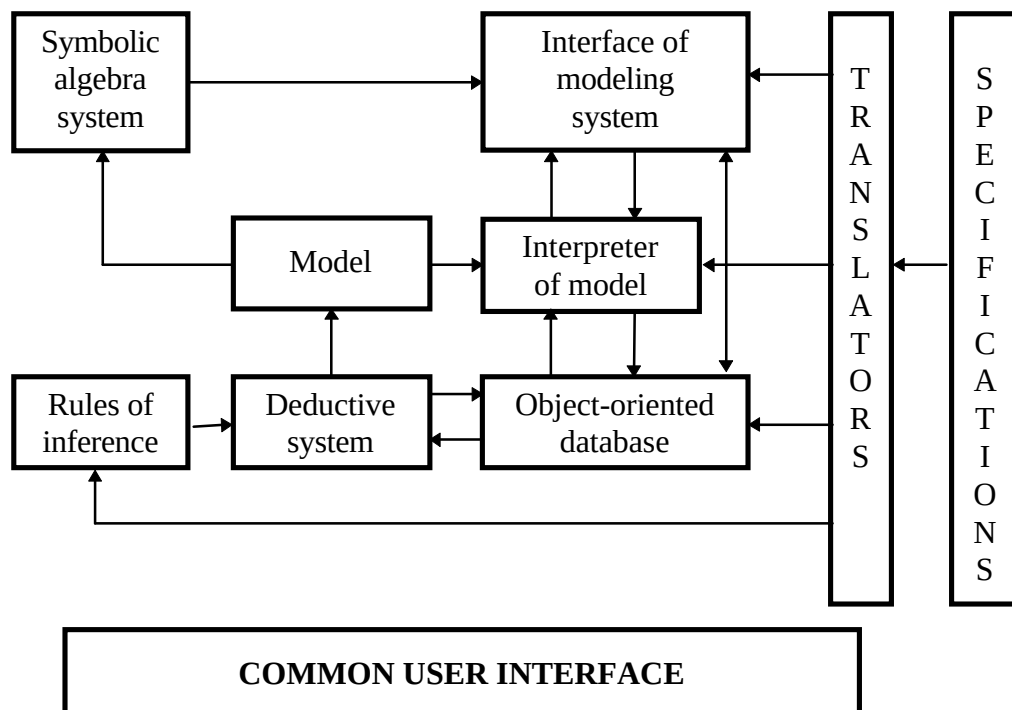


Figure 1: System architecture of object-oriented and constraint-based modeling system

Deductive system provides two levels of inference: object-oriented and constraint-based inference engine (OCIE) level and macro-planner (MP) level. OCIE produces inference on the base of rules and constraints, which are contained in definitions of objects' classes. MP manages the work of OIE without using of any constraints. MP prepares a task for output depending on class of the problem that is being solved. Examples of such tasks are, for example, analysis on some abstraction level, synthesis, optimization, animation. MP also checks the termination of input process and inspects if a generating model is underdetermined or overdetermined. If it is impossible to find out the

solution as a closed-form one then MP includes iterative processes into the model and orders OIE to provide planning of calculation parts of such processes. The activity of MP is also determined by a combination of correspondent rules.

To determine engineering or physical theory for OCMS the user should either prepare a set of specifications in declarative input language and create a class library (knowledge base), or use class libraries created beforehand. It is possible to work with several libraries of classes simultaneously.

5 An example of application

Constraint-Oriented Knowledge Base approach [13] made it possible to build knowledge-oriented system for kinematic and dynamic analysis in developed earlier system ANMEC. Nevertheless, composing and specification of knowledge-base required sufficient efforts because specification of objects and rules of knowledge base was carried out by different means. A proposed approach dramatically simplifies the problem of realization of computer kinematic theory and enables to enhance the system possibilities.

The Figure 2 presents a schematic diagram of eight-link mechanism, taken from the paper by Bahgat and Osman [17]. This research develops the closed-form equations for the coupler curve of an eight-link planar mechanism. In this article determination of coordinates of point J with input link O2AE is rendered to three parametric equations.

OCMS allows to automatically find the solution of the problem of kinematic analysis by composing only one closed-form equation.

Planning of solution is carried out in the following consequence:

1. MP determines a goal of solution -- to determine coordinates of point J. It is assumed that the coordinates of points O2, O5, length of all line segments except BC and EF are known. A value of angle of the line segment O2A (θ_2) with an axis X is also known.

2. OCIE plans the solution to find the coordinates of points A and E and informs MP that it is not possible to continue finding of a closed-form solution.

3. MP tries to plan an iteration process assuming that an angle of a line segment O5C with an axis X is known.

4. OCIE in this case plans the solution allowing to determine the coordinates of points B, D, F, G, H, C and lengths of line segments BC and EF.

5. MP defines that a length of a line segment DG is given and the coordinates of points D and G are determined by OCIE. MP sets that a length of a line segment DG is unknown and addresses to OCIE.

6. OCIE plans the solution to find the length of a line segment DG by the known coordinates of points D and G.

7. MP finishes the planning of iterative process and in fact inserts into the model the following equation:

$$G(\theta_2, \theta_5) - E(\theta_2, \theta_5) - DG.l = 0,$$

that is solved iteratively regarding the unknown variable θ_5 , when a variable θ_2 defining a current position of a mechanism is known.

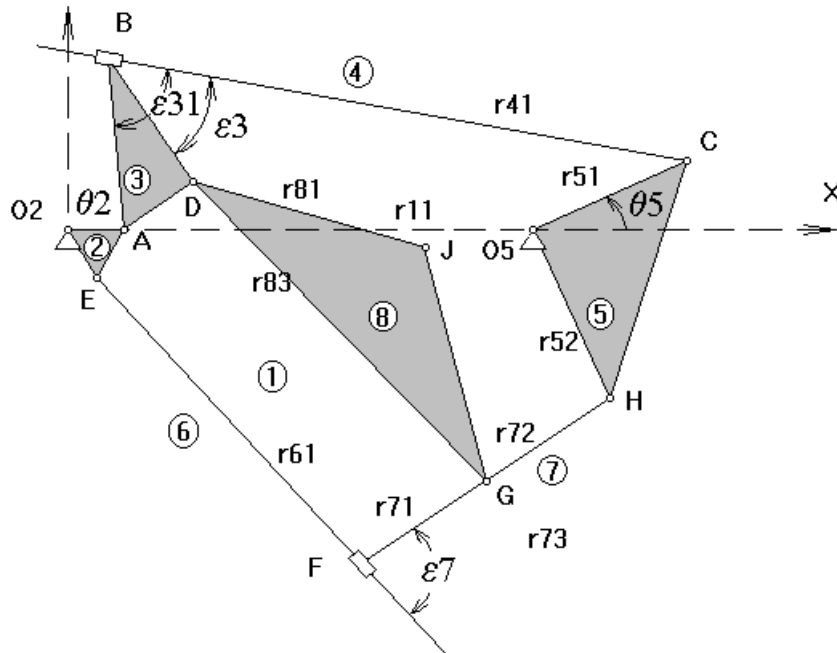


Figure 2: A schematic diagram of eight-link mechanism.

The functions $G(\theta_2, \theta_5)$ and $E(\theta_2, \theta_5)$ are generated by OCIE and inserted into the model as a closed-form solution.

Figure 3 presents a specification of objects' classes that are necessary to solve this problem.

Knowledge base	Planar Kinematics
-----------------------	-------------------

Types

real vector VEC2D [x,y]

Class point2D

Features hidden, fixed
VEC2D point2D

Class segment2D(a, b)

Ref point2d a,b
Features hidden, fixed, line
Substitution (A=a, B=b), (A=b, B=a)
real l, fi

Function

```
real uvec():=[cos(fi), sin(fi)]
```

Constraints

```
b=a + uvec() * l  
l:= | b - a |  
fi:= arctan(b.x-a.x, b.y-a.y)
```

Error

```
l < 0
```

Class onsegment2D(pnt, seg)

```
Ref point2D pnt
```

```
Ref segment2D seg
```

```
Features hidden, fixed, slider
```

```
real l1, l2
```

```
Virtual segment2D seg1(seg.a,pnt), seg2(pnt,seg.b)
```

```
l1 == seg1.l
```

```
l2 == seg2.l
```

```
seg.fi == seg1.fi
```

```
seg.fi == seg2.fi
```

Constraint

```
seg.l = l1 + l2
```

Class angle2D(seg1, seg2)

```
Ref segment2D seg1, seg2
```

```
Feature fixed
```

```
Substitution (SEG1= seg1, SEG2 = seg2),  
(SEG1= seg2, SEG2 = seg1)
```

```
real psi
```

Constraint

```
psi = seg2.fi - seg1.fi
```

Virtual class(2) dyad2D(pnt, seg1, seg2, mode)

```
Ref point2D pnt
```

```
Ref segment2D seg1, seg2
```

```
Branch mode
```

Constructor

```
segment2D(seg1), seg1.(B) = pnt,
```

```
segment2D(seg2), seg1≠seg2, seg2.(B) = pnt
```

External function

```
point2D dyad_plan(point2D, point2D, real, real, branch)
```

Constraint

```
pnt:= dyad_plan(seg1.(A), seg2.(A), seg1.l, seg2.l, mode)
```

Figure 3: An example of objects' classes specification.

6 Conclusion

Using of knowledge-based calculus (KBC) for composing of engineering theories of technical objects presents new possibilities for solving of multi-disciplinary engineering problems. Unlike the traditional approach, which considers an object separately, new approach combines points of view of different scientific disciplines. Means embedded into KBC for object-oriented inference allow to generate computer models of objects automatically. Such objects integrate in themselves knowledge from different domains (mechanics, electric engineering, thermodynamics, etc.) within the framework of a single conceptual system. The computer models can be important tools for research of objects at the early stages of design.

References

1. Vittikh, V.A. Engineering Theories as a Basis for Integrating Deep Engineering Knowledge, *Artificial Intelligence in Engineering*. (In printing)
2. Vittikh, V.A. Knowledge-Based Modeling Systems for Research of Engineering Objects, in AIENG 91 (ed), pp. 1011-1025, *Proceedings of the Sixth International Conference on Applications of Artificial Intelligence in Engineering*, Oxford, UK, 1991.
3. *A Brief Dictionary on Logic*, Prosvetchenie, Moscow, 1991.
4. *Encyclopedia on Mathematics*, Soviet Encyclopedia, Moscow, 1988
5. Maslov, S. *Theory of deductive systems and its application*. The MIT Press, 1987.
6. Fattah, Y.E. Constraint Logic Programming: Applications and Implications, *Artificial Intelligence in Engineering*, 1992, 7, 175-182.
7. Jaffar, J. & Michaylov, S. Methodology and Implementation of a CLP System, in LP/87 (ed. J.-L. Lassez), pp. 196-218, *Proceedings of the 4th Int. Conf. on Logic Programming*, Melbourne, 1987, The MIT Press, Cambridge, 1987.
8. Tyugu, E. Large engineering knowledge bases, *Artificial Intelligence in Engineering*, 1993, 8, 265-270.
9. Tyugu, E. Three new generation software environments, *Communication of the ACM*, 1991, 34, 46-59.

10. Greef, A.R., Fohn, S.M., Young, R.E. & O'Grady, P.J. Implementation of a logic-based support system for concurrent engineering, *Data & Knowledge Engineering*, 1995, 15, 31-61.
11. Budyatchevsky, I.A. & Vittikh, V.A. A knowledge-based system for interactive computer-aided kinematic analysis of planar mechanisms, in AIICSR/89 (ed. I. Plander), pp. 355-358, *Proceedings of the 5th Int. Conf. on Artificial Intelligence and Information-Control Systems of Robots-89*, Štrebské Pleso, Czechoslovakia, 1989, Elsevier Science Publishers B.V, Amsterdam, 1989.
12. Budyachevsky, I.A. & Vittikh, V.A. A knowledge-based system for automated generation of engineering models, in AIENG/94 (ed. G. Rzevski, R.A. Adey & D.W. Russel), pp. 229-236, *Proceedings of the 9th Int. Conf. on Applications of Artificial Intelligence in Engineering IX*, Pennsylvania, USA, 1994, Computational Mechanics Publications, Southampton, 1994.
13. Budyachevsky, I.A. & Vittikh, V.A. A Constraint-Oriented Knowledge Base for Modeling and Simulation in Concurrent Engineering, in CE/95 (ed. A.J. Paul & M. Sobolewski), pp. 13-19, *Proceedings of: Concurrent Engineering 1995 Conference*, McLean, Virginia, 1995, Concurrent Technologies Corporation, Johnstown, 1995.
14. Kramer, G.A. *Solving Geometric Constraint Systems: A Case Study in Kinematics*, The MIT Press, Cambridge, 1992.
15. Arbab, F. & Wang, B. A Constraint-Based Design System based on Operational Transformation Planning, in AIENG/89 (ed. J.S. Gero), pp. 403-426, *Proc. of 4th Int. Conf. on Applications of Artificial Intelligence in Engineering*, Cambridge, UK, 1989, Computational Mechanics Publications, Southampton, 1989.
16. Aldefeld B., Malberg H., Richter H. & Vass K. Rule-based variational geometry in computer-aided design, *Artificial Intelligence in Design*, ed. D.T. Pham, pp. 28-46, Springer-Verlag, Berlin and New York, 1991.
17. Bahgat, B.M. & Osman, M.O.M. The parametric coupler curve equations of an eight link planar mechanism containing revolute and prismatic joints, *Mechanism and Machine Theory*, 1992, 27, 201-211.