Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

A knowledge-based system for automated generation of engineering models

I.A. Budyachevsky & V.A. Vittikh

Samara Division of Mechanical Engineering Research Institute, Russian Academy of Sciences, Russia

Abstracts

In this paper a knowledge-based approach towards the development of computer models of engineering objects is being considered. There is also a discussion of constraint-oriented software tools as possible means of realization of the stated approach. Constraint-oriented knowledge base is proposed. The paper describes COTGEM software system for development of applied modeling systems.

1 Introduction

Computer Modeling is on the increase in solving different tasks and becoming a routine tool in engineering. Therefore, the requirements to quality and time of computer-aided development of objects' models are constantly improving.

Satisfaction of these requirements is connected with solving of the following problems:
- knowledge extraction from the expert specialists;
- automatic programming of computer models;
- development of a friendly user's interface.

Moreover, a bottleneck in the process of computer-aided development of a model is a work with the experts. It is necessary to find qualified specialists and manage the extraction of essential knowledge from them and then skillfully use the obtained knowledge in object model design. The key question is a knowledge system's completeness degree since it is possible that some important model's factors are not considered at all. The formulation of a primitive system of notions and relations with the help of experts is usually a labour-consuming procedure. In other words, it is hard to consider as effective the computer-aided development of objects' models approach that assumes to enlist experts each time when the new modeling task arises.

Another approach is preferable. Experts carry out the conceptual analysis of domain to make possible solution of wide range of modeling tasks within the scope of this domain. An appropriate knowledge base is created. It is necessary then to develop software tools (or to use existing ones) for composing a model-program. The model-program is automatically generated using knowledge base according to the initial object description in some input language.

In this case there is no need to enlist experts for solving new modeling task within work domain. The first results that have been achieved [2,7] demonstrate the potential advantages of such automation of computer-aided development of objects' models using knowledge base.

2 Approach to computer-aided development of objects models with the use of knowledge base

The scheme on Figure 1 illustrates the proposed approach. As a result of conceptual analysis of domain a set of primitive notions and relations is formed by the experts. Knowledge engineers use the notions and relations for development of knowledge base. It is they who create programs for computer model design using knowledge base. The user interface should provide the dialogue facilities in a domain language that is close to the natural one.

The user's work with this automatic modeling system is carried out in the following way. A user describes engineering object in the input language within modeling system. The program-designer of modeling system generates a computer model of the object using existing knowledge base. The generated model corresponds to object description. The user could explore the model and flexibly change its parameters and structure.

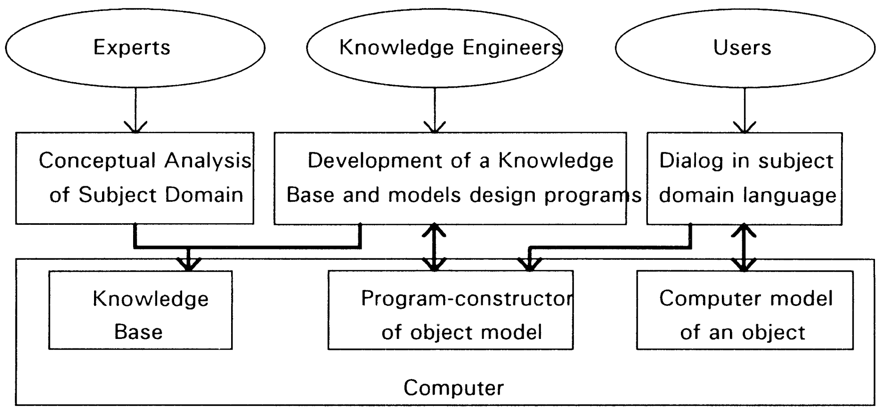

Figure 1: The scheme illustrates a knowledge-based approach to automated generation of computer models.

Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Inside the limits of the knowledge base possibilities the user can solve a variety of modeling tasks without enlisting to the experts. If necessary, the variety of tasks could be expanded by inserting of some additions into the knowledge base. But this is the work of experts. When the work is completed, the user acts with the system independently.

Automation of objects models development using knowledge base has another important advantage: working with such system doesn't require a user's high qualification.

What programming methods and tools could be used to realize that approach?

3 Constraint-oriented software tools

That sort of methods ought to be looked for in the sphere of software tools to provide Concurrent Engineering (CE) [5]. It is caused by the fact that in CE modeling role is dramatically increased unlike traditional computer-aided design applications. One of the key methodologies, which forms the basis of programming modeling systems development for CE is constraint-oriented programming in which constraint-oriented programming language plays a major role. However, no one of the existent languages of that type is adequate for modeling in CE because of orientation to their highly specialized domain [1].

It could be explained by the fact that the designers of constraint-oriented languages give too much attention to the possibility of fully automatic problem solving of constraint satisfaction. The appropriate software usually has a built-in constraint satisfier or solver that is not programmable by user. In other words, the implementation of constraint-oriented languages for modeling in CE is available but not effectual.

It is necessary to extend a domain. To ensure a high degree of knowledge integration it is suitable to use a rule-oriented knowledge base. In this case, constraints ought to be directly inserted into knowledge base rules unlike constraint-oriented languages [3], in which the constraints are included either in the program's text, or in the language objects. Let us call that kind of knowledge bases as Constraint-Oriented Knowledge Base (COKB).

The principles of the Constraint-Oriented Tools for Generation of Engineering Models (COTGEM) are considered below.

Constraint-Oriented Knowledge Base. Knowledge Base System (KBS) is a central part of the COTGEM system. It uses the Rule-Based Representational Scheme and Horn-clause logic. KBS operates with the objects that belong to the object-oriented data base of COTGEM. The Constraint-Oriented Knowledge Base of KBS is a collection of rules, predicates, functions and meta-rules. The COKB description is written in Knowledge Specification Language (KSL). The example of this description for two notions -- a point and a straight line segment -- is defined as follows:

```
KBASE     geometry
TYPES
   vec      = struct { real x, y, z };
   seg_type = enum { spatial, planar }
OPERATORS
   vec = vec + vec;   vec = vec - vec;
   vec = vec * real;   vec = vec / real
CLASSES
   point       { vec p }
   segment     { seg_type state; real  l; vec uvec; point pnt1, pnt2 }
EXTERNAL FUNCTIONS
   vec planarDyad( point, vec, vec, real, real )
FUNCTIONS
   vec  planarAngleToVec( real fi ) { return { cos( fi ), sin( fi ), 0 } }
CLAUSES
   seg_points( segment S, point A, point B ) :-
     segment( S ), ( S.pnt1 == A, S.pnt2 == B or S.pnt1 == B, S.pnt2 == A ).
RULES
   segment_l ( segment S ) :-
     segment( S )   =>
     S.l = sqrt( ( S.pnt2->p.x - S.pnt1->p.x )^2 +
        ( S.pnt2->p.y - S.pnt1->p.y )^2 + ( S.pnt2->p.z - S.pnt1->p.z )^2 ).
   segment_p ( segment S ) :-
      segment( S )  => S.pnt2->p  =  S.pnt1->p + S.vec->u * S.l.
   planar_dyad_p ( segment S1, segment S2, point A, point B, point C ) :-
      seg_points( S1, A, B),   S1.state == planar,
      seg_points( S2, B, C),   S2.state == planar,
      S1 \= S2    =>   B.p := planarDyad( B, A.p, C.p, S1.l, S2.l ).
```

The section RULES contains the description of COKB rules. A condition part provides matching of the rule with the current data base state. A symbol '=>' is followed by conclusion part that includes a list of constraints. A constraint can be of explicit form (assignment) or implicit form (equation). There is an exterior similarity between rules in COKB of COTGEM and rules of Constraint Logical Programming Language CLP(R) [4], but the rules of CLP(R) interact with each other. In contrast, the rules of COKB are independent from each other. Rules of COKB never use arguments. As a result, the bindings of the rules' variables are realized through data base.

Constraint solver. The constraint solver is intended for solving of constraint satisfaction problems. In COTGEM, the most of constraint solver functions are executed by the inference engine of KBS on the one hand and the KSL translator from the other hand.

Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Planner. In knowledge base one or several planners are determined. Each application domain needs its own planner. Planner creates a model as a program for model interpretator. Actually, a planner is a meta-rule of knowledge base. On the same data base the various models can be generated by different planners. For example, for dynamics and kinematics of mechanisms one planner forms the model for the computation of initial mechanism position and the other prepares the model for the different types of analysis and synthesis.

Generation of models. COTGEM provides the output in the form of answer model for the current data base state. A user specifies a model that is to be generated by defining the data base values that are known, unknown or neutral. Planner uses the known variables to calculate the unknown ones and involves neutral variables into account if necessary, consider them as unknown ones. A model is always represented in the form of a program for the model interpreter of COTGEM. Additionally, the model can be presented to the user in a symbolic form or it can be output in the FORTRAN format or in MathCAD or Mathematica [8] input format. Particularly, the system Mathematica can execute symbolic transformations of a generated model.

4 Example Applications

The system ANMEC [2] was created with the help of COTGEM. The system is intended for dynamic analysis and synthesis of planar and spatial mechanisms and robots. It also carries out displacement, kinematic and force analysis. ANMEC finds a non-iterative, closed-form solution in many practical cases or combines closed-form and iterative numerical solution. ANMEC finds an efficient solution reaching on a PC the power of workstation that used conventional linkage analysis packages. Consider two examples of the ANMEC usage.

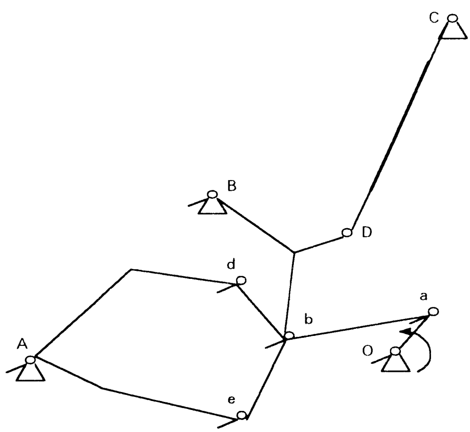

Figure 2: Seven-Link Planar Mechanism Benchmark Problem.

Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

The Figure 2 shows a seven-link planar mechanism with one driven link. The driving torque and all the geometrical and inertia parameters of all bodies, spring (CD) properties are listed in [6]. This linkage is used in [6] as the benchmark problem. ANMEC finds a close-form solution for kinematic task then creates and solves one differential equation for dynamic task. It takes 3.2 sec. for planning and optimizing solution and 12.4 sec. for 300 simulation steps to the ANMEC system running on IBM PC 486DX (33 MHz). The solution time of the same task of the most widely used software package ADAMS [6] that is intended for multibody system analysis running on Vaxstation 2000 is 380 sec. ADAMS uses iterative numerical techniques for solving 182 equations that it generates for this task.

In the ANMEC system a user could define his own relations between arbitrary objects and their variables. This possibility is illustrated by the second example. Figure 3 depicts a diagram of the task from a textbook on optics. A beam that is emitted from the point A is sequentially mapped from two mirrors P1P and P2P. The angle between them is fixed and its value is $\boldsymbol{a}$. It is required to prove that the angle between the emitted beam and another beam that arrived into point B is constant and depends neither from rotation of beam around point A, nor from the mirrors turn around point P.

It is possible to solve this task with ANMEC by defining relation in which the angle of beam incidence equals to the angle of its reflection. The process should be modeled in the following way. At first the beam is rotating around point A then it stops and two mirrors begin to rotate around the point P.

The value of angle between emitting and final beams is being stored step by step for its graph is to plotted. The animation of this process can also be seen on the display.

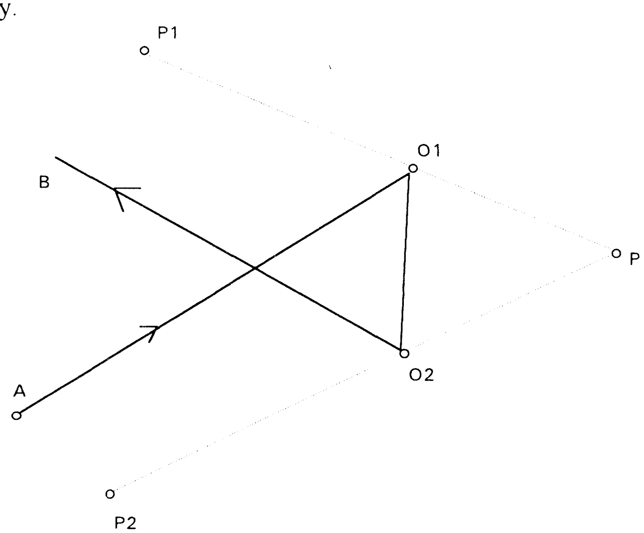

Figure 3: The scheme of optical task.

Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

As a result of evaluations, a constant value of the angle between emitting and final beams equals to doubled value of the angle between mirrors. Then the generated model has been exported in the format of system Mathematica [8]. The system Mathematica gives the result -- ***2a*** for this input file. Thereby, we have got a numeric solution and an analytical proof of the given task. In this case, the modeling is rather qualitative than quantitative because the diagram was simply depicted on the display and there were not any numeric values. The user needs about 10 minutes to prepare and solve this task by means of ANMEC and Mathematica together. It took 1.3 sec. of CPU time for planning solution in ANMEC. Optical task model in the format of Mathematica is given below:

```
<<trigonom.m
psixA=0
psixP=0
dpsirxA=If[t<15,Pi/180*t,Pi/180*15]
dpsirxP=If[t<15,0,Pi/180*(t-15)]
dpsi1xAAO1=Pi/180*35
dpsi1xPP2P=Pi/180*25
dfiP1PP2P=a
psirxA=psixA+dpsirxA
psirxP=psixP+dpsirxP
uyAO1=Cos[psirxA+dpsi1xAAO1]
uzAO1=Sin[psirxA+dpsi1xAAO1]
uyP2P=Cos[psirxP+dpsi1xPP2P]
uzP2P=Sin[psirxP+dpsi1xPP2P]
uyP1P=uyP2P*Cos[dfiP1PP2P] + uzP2P*Sin[dfiP1PP2P]
uzP1P=uzP2P*Cos[dfiP1PP2P] - uyP2P*Sin[dfiP1PP2P]
dfiP1PAO1=ArcTan[uyAO1,uzAO1]-ArcTan[uyP1P,uzP1P]
dfiP1PO1O2=-dfiP1PAO1
uyO1O2=uyP1P*Cos[dfiP1PO1O2] - uzP1P*Sin[dfiP1PO1O2]
uzO1O2=uzP1P*Cos[dfiP1PO1O2] + uyP1P*Sin[dfiP1PO1O2]
dfiP2PO1O2=ArcTan[uyO1O2,uzO1O2]-ArcTan[uyP2P,uzP2P]
dfiP2PO2B=-dfiP2PO1O2
uyO2B=uyP2P*Cos[dfiP2PO2B] - uzP2P*Sin[dfiP2PO2B]
uzO2B=uzP2P*Cos[dfiP2PO2B] + uyP2P*Sin[dfiP2PO2B]
dfiAO1O2B=ArcTan[uyO2B,uzO2B]-ArcTan[uyAO1,uzAO1]
Simplify[TrigFactor[dfiAO1O2B]]
```

The ANMEC system allows to accumulate in COKB a great number of effective and robust methods for analysis and synthesis of mechanisms. A particular mechanism could be decomposed to proper parts by ANMEC. These methods are applied to the parts automatically. It is the difference of ANMEC from the traditional software packages that use only one method for a whole range of tasks or require a user's prompt to choose the proper method of the solution.

5 Conclusion

COTGEM is a tool that supports a knowledge-based approach for generating of engineering models. On its foundation, it is possible to create a variety of applied modeling systems. Such systems are flexible and could provide the possibility of solving a great number of tasks of a domain. The development of such applied systems assumes to carry out conceptual analysis of the domain and creation of Constraint-Oriented Knowledge Base. As a result, a user gets an effective and friendly computer modeling environment, within which he could solve his own tasks, using the language of his domain.

References

1. Bowen, J. & O'Grady, P., Constraint Networks for Life-Cycle Engineering, *Proceedings of the Fifth International Conference on Applications of Artificial Intelligence in Engineering V*, pp. 281-296, 1990.

2. Budyatchevsky, I.A & Vittikh V.A., 1989. A Knowledge-Based System for Interactive Computer-Aided Kinematic Analysis of Planar Mechanisms, *Artificial Intelligence and Information-Control Systems of Robots-89*. (ed I.Plander). North-Holland: Elsevier Science Publishers B.V, pp. 355-358.

3. Fattah, Y.E., Constraint Logic Programming: Applications and Implications, *Artificial Intelligence in Engineering*, 1992, Vol. 7, No 3, 175-182.

4. Jaffar, J. & Michaylov, S., Methodology and Implementation of a CLP System in Logic Programming, *Proceedings of the Fourth International Conference*, Cambridge, Massachusetts, The MIT Press, pp. 196-218, 1987.

5. O'Grady, P. and Young, R., Issues in Concurrent Engineering Systems, *Journal of Design and Manufacturing*, Vol. 1, No 1, pp 27-34, 1991.

6. Schiehlen, W. (Ed.), *Multibody Systems Handbook*, Springer-Verlag, Berlin, 1990.

7. Vittikh, V.A, Knowledge-Based Modelling Systems for Research of Engineering Objects, *Proceedings of the Sixth International Conference on Applications of Artificial Intelligence in Engineering VI*, pp 1011-1025, 1991.

8. Wolfram, S., *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley, 1988.