Conceptual Graphs in CAD

O. W. Salomons, F. van Slooten, G. W. F. de Koning, F. J. A. M. van Houten (2), H. J. J. Kals (1),
Laboratory of Production and Design Engineering, University of Twente, The Netherlands

Received on January 5, 1994

Summary.

This paper elaborates on the use of conceptual graphs in a prototype of a computer based support system for re–design. Re–design support involves the modelling of assemblies and components. The requirements of the components to be modelled are a compromise between the functioning of the assembly and the manufacturability of the individual components. Conceptual graphs provide for an elegant way of representing both functioning and manufacturing aspects. In the prototype system, conceptual graphs are used for representing and defining assemblies, components and features as well as the relations between these entities. Constraints, such as kinematic, tolerance and manufacturing constraints are also represented using conceptual graphs.

Key Words: CAD, CAPP, features.

1. Introduction.

In mechanical engineering design a high percentage of all the design tasks can be considered as re–design tasks. In this context, re–design is considered as design with a priori knowledge on the functions and the physical principles to be used in the design process. Re–design is often expensive and error prone. Re–design seems attractive for computer support in order to reduce the time of re–design and to improve quality. Proper computer support tools for re–design are not yet available. In a re–design support tool, it should be possible to model assemblies and components, taking care of both functioning and manufacturability. Functioning is related to the assembly that is being modelled. Manufacturability is related to single components, focusing on the features of the components. This paper reports on the use of conceptual graphs as an elegant form of representing both functioning and manufacturing aspects. The conceptual graphs are employed in a re–design support tool called FROOM. Section 1.1 elaborates on conceptual graphs. Section 1.2 details on the FROOM prototype. Section 1.3 provides an overview of the remainder of this paper.

1.1 Conceptual graphs.

Conceptual graphs, or conceptual structures, have been proposed for use in natural language processing and for representing mental models by Sowa [18]. However, the number of applications of conceptual graphs in other domains is increasing. One of these domains is that of (re)design support in CAD, which is also the focus of this paper.

Conceptual graphs are graphs with two different kinds of nodes: concepts and conceptual relations. Conceptual graphs are therefore called bipartite. Sowa defines a conceptual graph as follows [18]. A conceptual graph is a finite connected bipartite graph. The two kinds of nodes of the bipartite graph are concepts and conceptual relations. Every conceptual relation has one or more arcs, each of which must be linked to some concept. If a relation has n–arcs it is said to be n–adic and its arcs are labelled 1,2..n. The term monadic is synonymous with 1–adic, dyadic with 2–adic and tri–adic with 3–adic. A single concept may itself form a conceptual graph, but every arc of every conceptual relation must be linked to some concept.

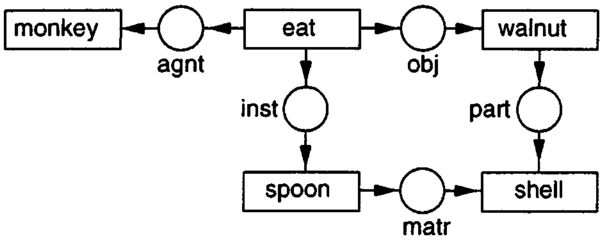


Fig. 1 An example of a conceptual graph, redrawn from [18].

The concepts in a conceptual graph are usually denoted by rectangles and the conceptual relations are usually denoted by circles. Figure 1 shows a simple example of a conceptual graph, taken from [18]. The graph in figure 1 may be read as "a monkey eating a walnut with a spoon made out of the walnut's shell". The concepts are monkey, eat, walnut, spoon and shell while the conceptual relations belong to a pre–defined set of relations [18]. The directions of the arcs in fig.1 depend on the relation definitions, as defined by Sowa [18].

1.2 FROOM.

FROOM is a prototype of a re–design support system, currently under development. The development of FROOM has been motivated by the following facts. First, in mechanical engineering design a high percentage of all the design tasks can be considered as re–design tasks. For these tasks proper computer support tools are not yet available. Second, CAD tools that integrate well with automated process planning (CAPP) systems are not yet available. The CAPP system that is of particular interest to us, is the PART system [7]. FROOM is an acronym for Feature and Relation based Object Oriented Modelling. FROOM is a feature based system, allowing the modelling of both components and assemblies. Features in the FROOM context can be design form features, manufacturing form features and even abstract features. For a review of feature–based design, refer to [12]. FROOM employs conceptual graphs for knowledge representation, which is the focus of this paper. The conceptual graphs allow re–design support to be provided and facilitate the link with CAPP. The latter subject will not be addressed in this paper, however. Figure 2 shows Froom's system architecture.

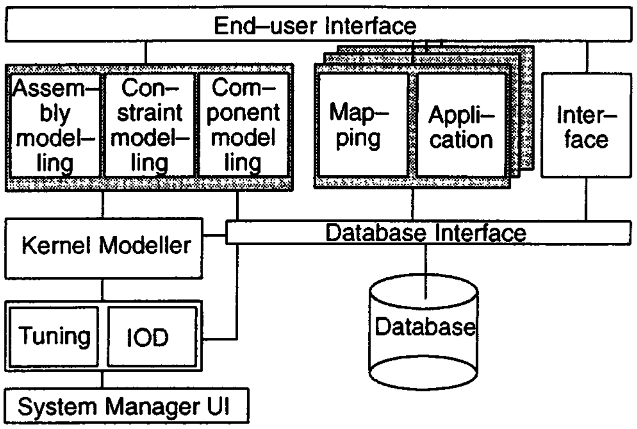


Fig. 3 System architecture of FROOM; IOD means interactive object definition.

As can be seen from figure 2, two different kinds of users are determined: end–users and system manager users. End–users are the designers who design assemblies, components etc.; they perform the actual design tasks. System managers customize the system to a certain application domain, company and user (group). They define the features to be used, the catalogues from which selections can be made etc. For these two different user groups, separate user interfaces are available.

In the modelling module of FROOM, components, assemblies and constraints can be modelled. This module has access to the kernel modeller, which offers the basic geometry processing functionality. In FROOM, a commercially available kernel modeller is used for this. The application module includes the possible mappings to applications and the applications themselves. An example of an application might be manufacturability evaluation. For more details on FROOM refer to [13] and [14].

1.3 Organization.

The organization of this paper is as follows. Section 2 briefly summarizes related literature. Section 3 provides the main differences of the conceptual graph approach as implemented in FROOM, when compared to other approaches. Section 4 elaborates on the use of conceptual graphs in FROOM for assembly representation and definition. Section 5 details on the use of conceptual graphs for representing and defining components and features. Section 6 summarizes constraint representation via conceptual graphs. Finally, in section 7, conclusions and recommendations are presented.

2. Review of relevant literature.

Schmekel reported on the representation of product data in a computer system for the conceptual design of mechanical products, based on the use of conceptual graphs [16]. The work by Schmekel is based on the general principles of design by Suh [19] and the systematic principles of design by Pahl and Beitz [11]. Closely related to the work by Schmekel is the work by Andersson [1], [2]. Andersson reports on a design language based on engineering terminology called CANDLE – also used by Schmekel – that is based on the conceptual graphs as described by Sowa [18]. The concept nodes in CANDLE represent entities, attributes, states and events while the relation nodes show how the concepts are interconnected. Schmekel classifies relations into part, port, connection, equality, attribute and constraint relations [16]. There are some deviations as compared to Sowa [18]: the graphical symbols are somewhat different and entities and relations have so-called "roles". Roles represent e.g. the procedures or rules that are utilized for the synthesis and analysis of various solutions during the design work. Object oriented classifications of both concepts and conceptual relations are provided. Constraints are represented in CANDLE by attaching them to the conceptual relations.

Billo et al. reported on the use of conceptual graphs for feature representation and definition [3], [4]. Feature representation is defined here as representing both the individual (form) features and the relations between the individual features that constitute a component. Billo et al. would like to bring some more rigour into feature representation by using conceptual graphs. By doing this, feature recognition and feature mapping or conversion would be made easier. A simple example of feature transformation to Group Technology coding is given. The feature representation used by Billo et al. – in case of prismatic features – consists of face concepts and adjacency relations between them. The face concepts are part of the general feature concept. However, it is not fully clear whether other relations between faces are also possible such as for example perpendicular, concave, convex, parallel etc. Other, more complex features are defined using conceptual graphs for sweeps and ruled surfaces.

Hashim et al. use conceptual graphs as a means for functional reasoning in generating design variants [6]. Ullman [22] has a similar notion of objects and relations as Schmekel [16], Andersson [1], Hashim [6] and as described in this paper. However, Ullman does not report on the use of conceptual graphs for representing objects and relations [22]. Taleb–Bendiab et al. use Entity–Relationship diagrams for the semantic modelling of engineering designs [20]. Entity relationship diagrams are very similar to the conceptual graphs as described in this paper.

3. Conceptual graph approach in FROOM.

In FROOM, the objects that are of concern during the (re)design process are considered as concepts in a conceptual graph. Concepts can be assemblies, components, features and even faces and other low level geometric entities. Assemblies are built up of components which are built up of features which in turn are built up of faces. These objects can be very abstract or very concrete.

Relations in FROOM can exist between any combination of the different types of objects, e.g. assembly–assembly, assembly –feature, component–feature or feature–feature etc. The relations often indicate functioning or manufacturing constraints. The relations can be very abstract or very concrete. FROOM allows an initial set of objects and relations, defined by the system manager, to be refined by the end–user. As in the work by Schmekel [16] and Andersson [1], taxonomies of both objects and relations are present. Fig. 3 shows the combined taxonomy that is currently partly implemented.

The most important deviations of the conceptual graphs in FROOM from the conceptual graphs as defined by Sowa [18] are the following:

– Only assemblies, components, features, faces and other geometric objects can be concepts that are shown in the graph. Attributes for instance, are part of the object oriented descriptions of concepts and relations and are usually not shown in the conceptual graph. That is, unless the constraint network of parameter constraints is shown (section 6).

– The relations used in FROOM are monadic, dyadic or tri–adic. Dyadic relations are most commonly used. Monadic relations are mainly used for defining subgraphs which can represent alternatives. From these, the end–user can select the alternative which will finally be instantiated (joined) in the overall graph (see also fig.5). As a consequence, if a join on a relation is to be performed there should be two monadic relations in the subgraph. If a join on a concept is to be performed, no monadic relations should be part of the subgraph [8]. Monadic relations are also used in tolerance representation. This also holds for tri–adic relations.

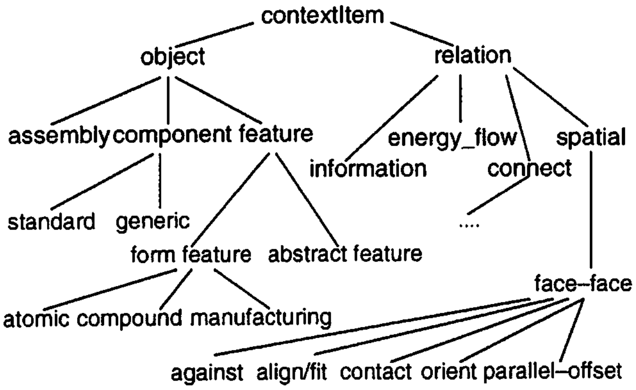


Fig. 3 Taxonomy of objects and relations in FROOM.

– Directed arcs – as shown in fig.1 – are only used when necessary. Often relations just denote a general connection relation like "component A is connected to component B". Usually, "component B is connected to component A" holds equally well, so that the arcs could have been directed in both directions. In these cases, arcs are omitted.

– In the graphical notation of concepts and relations, sometimes other symbols are used than the small rectangle or circle. Symbolic icons are sometimes used instead of the circles or blocks; see for instance figures 4, 5 and 7.

4. Conceptual graphs for assembly representation and definition.

Assemblies consist of components and/or sub assemblies. Therefore, the conceptual graph representing the assembly consists of the objects that make up the assembly and the connection relations between these objects. Figure 4 illustrates this. Figures 4 – 5 show the current end–user interface of FROOM. The upper left window is the geometry window in which actions similar to those in current CAD systems can be performed. In the upper right window, the context window, a conceptual graph is shown. In the following, conceptual graphs will also be referred to as context structures. The graph reflects the actual internal representation of all objects and relations. The concepts in the graph correspond to the geometry objects in the geometry window. There is a bi–directional association between the context window and the geometry window.

The two main windows allow a mixed top–down and bottom–up design style. Bottom–up support is given in the geometry window similar to existing CAD systems. In the geometry window, component modelling using features is supported as well as assembly modelling by relating previously defined components to one another. Sketching could offer additional functionality in the geometry window. Currently, sketching is not implemented however. Top–down support is given from the context window: if an object or relation has the color green, then something predefined can be found on a lower level which can be detailed further. This is illustrated in figures 4 and 5. Figure 4 shows a relation between a shaft and a gearwheel that has been predefined by the system manager (the connect–1 relation). This relation can be refined by the end–user to become a key connection; see figure 5. Because of the association between the geometry and context windows, a mixed top–down and bottom–up design style can be adopted.

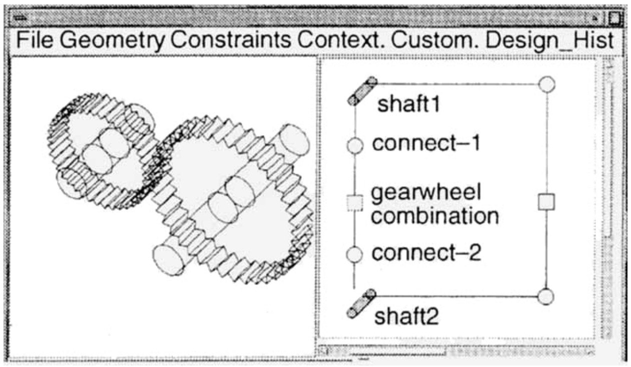


Fig. 4 The two main windows of the FROOM end–user interface.

An example of bottom–up support that has been implemented in FROOM is relating (positioning) existing components relative to one another. This is achieved by so–called low level relations or mating relations. At present, the following low level relations have been implemented in FROOM: against, align/fit, contact, orient and parallel offset. The implementation of these relations is based on a slight modification of the work presented by Liu and Nnaji [10]. This approach allows for automatically determining the kinematic degrees of freedom of each component and for automatically determining the functional surfaces which is important for tolerancing [5].

The way in which a system manager can (pre)define an assembly is shown in fig. 6. This environment is the so–called context editor. The context editor allows the system manager user to define a conceptual graph (context structure), representing the assembly. Geometry can be added to the concepts in the predefined graph. After completing the graph, the validity of the graph can be verified automatically. The verification of the graph is based on rules defined by Sowa in [18]. The verification procedure is described in [8].

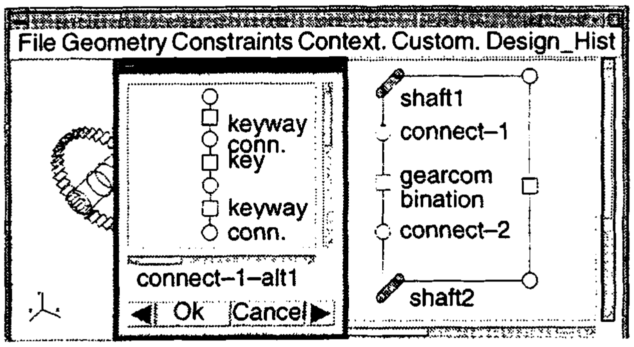


Fig.5 FROOM end–user interface, alternative selection.

5. Conceptual graphs for component and feature representation and definition.

As in the work by Billo et al. [3], [4], feature representation in FROOM is based on the use of conceptual graphs. However, more relations can be defined between the individual faces constituting a feature other than just an adjacency relationship. At present, the following relations are also supported: adjacent (convex, concave, tangential), perpendicular, parallel and co–axial. In fact, these relations can be viewed at as feature internal constraints. Fig.7 gives an example of how features are currently interactively defined by the system manager. This feature definition environment is analogous to the context editor for predefining assemblies. Features can be defined by editing a conceptual graph representing the feature. A difference with the context editor, however, is that the user instantaneously also gets sample geometry in the geometry window. This is considered to be very important as users usually do have geometry in their mind of which the graph is an abstraction. In the future, the feature definition user interface will have to be extended to enable direct geometric input, e.g. by means of sketching, followed by automatic conceptual graph generation.

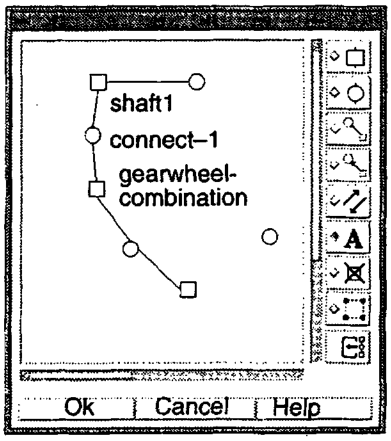
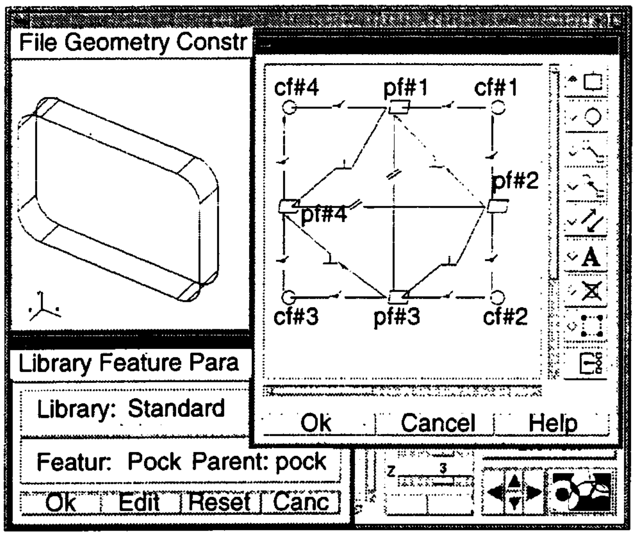


Fig. 6 Assembly definition in FROOM.

The types of available concepts and relations are different when compared with the context editor for assemblies. Presently available concepts are "planar face" and "cylindrical face". Future extensions will have to be made to the available concepts. For instance, the definition of free–shaped features (sweeps, ruled surfaces etc.) and abstract features will have to be accounted for. The feature definition capabilities of FROOM are elaborated in [15].

Component modelling or definition by the end–user, using predefined features, has not yet been implemented as this functionality is already present in most commercially available CAD systems.



Library: Standard

Featur: Pock Parent: pock

Fig. 7 Feature definition in FROOM.

6. Conceptual graphs and constraint representation.

Sowa describes the use of actors in dataflow graphs as an extension of conceptual graphs for representing sequencing problems [18]. This concept could be used for representing constraints. Billo et al. use these actors in dataflow graphs [3], [4]. However, they only use them for defining relatively simple parameter dependency constraints. Schmekel [16] and Andersson [1], [2] also represent parameter constraints in conceptual graphs. However, in their approach the constraints are attached to the conceptual relations. Schmekel distinguishes integrity, quantitative, qualitative and finite constraints [16].

In FROOM, the main types of constraints that are considered are: nominal geometry constraints, variational geometry constraints (tolerances) and parameter constraints. All these constraints can be represented using conceptual graphs. The nominal geometry constraints are comparable to Schmekel's integrity constraints, the tolerance constraints to Schmekel's qualitative constraints and the parameter constraints to Schmekel's quantitative and finite constraints.

The relations that can be defined between faces of features in the feature definition module in FROOM can be regarded as feature–internal nominal geometry constraints. Feature external constraints are usually defined between faces that belong to a feature and those that do not. These constraints are also related to nominal geometry. The assembly relations that can be defined between different faces of different components like against, fit etc., can be viewed as (kinematic) nominal geometry constraints. These geometry constraints are represented and defined as relations in the conceptual graphs. Solving the nominal geometric constraints is based on work by Kramer [9] and solving variational geometric constraints is based on work by Clément et al. [5].

Parameter constraints in FROOM presently are simple linear, uni–directional parameter dependency constraints. This, however, does not involve the use of conceptual graphs. In the future, a more advanced parameter constraint satisfaction mechanism is needed, for instance based on [17] and [21], in which the conceptual graphs can be of use for representing and showing the constraint network. In this case, the variables or parameters are the concepts while the constraints can be represented as the conceptual relations.

7. Conclusions and recommendations.

The focus of this paper is on the application of conceptual graphs in a re–design support tool, called FROOM. Conceptual graphs have been applied for representing and defining assemblies, components, features, low level geometric objects and constraints. Conceptual graphs are used as a means of supporting a mixed top–down and bottom–up design style. Conceptual graphs can represent both functional and manufacturing related information. Therefore, they offer opportunities for supporting (re)design and for improving the link with CAPP. The use of conceptual graphs for constraint representation is an issue of further research.

Acknowledgements.

This research is supported by the Technology Foundation.

References.

[1] Andersson, K., 1993, A design language based on engineering terminology, PhD thesis, R.I.T., Stockholm.

[2] Andersson, K., 1993, CANDLE – A new tool for conceptual design, proc. of ICED'93, The Hague, Vol.1, 45–52.

[3] Billo, R., Henderson, M., Rucker, R., 1989, Applying conceptual graph inferencing to feature–based engineering analysis, Comp. in Industry, vol. 13, 195– 214.

[4] Billo, R.E., 1989, An object–oriented modeling methodology utilizing conceptual graphs for form feature definition, PhD thesis, Arizona State University.

[5] Clément, A., Rivière, A., 1993, Tolerancing versus nominal modelling in next generation CAD/CAM system, CIRP sem. Comp. Aided Tolerancing, Cachan, 97–113.

[6] Hashim, F.M., Juster, N.P., de Pennington, A., 1993, Generating design variants based on functional reasoning, proc. of ICED '93, The Hague, Vol. 1, 60–67.

[7] Houten, F.J.A.M van, 1991, PART, a computer aided process planning system, PhD thesis, Univ. of Twente.

[8] Koning, G.W.F de, 1993, System manager user interface funktionaliteit in FROOM, in dutch, internal report, PT 457, University of Twente, Enschede.

[9] Kramer, G.A., 1992, Solving geometric constraint systems, a case study in kinematics, The MIT Press, Cambridge.

[10] Liu, H–C., Nnaji, B.O., 1991, Design with Spatial Relations, J. of Manufact. Systems, Vol. 10, No. 6, 449–463.

[11] Pahl, G., Beitz, W., 1984, Engineering Design, A systematic approach, Springer Verlag.

[12] Salomons, O.W., Houten, F.J.A.M. van, Kals, H.J.J., 1993, Review of research in feature–based design, J. of Manufacturing Systems, Vol.12, No.2, 113–132.

[13] Salomons, O.W., Kappert, J.H., Slooten, F. van, Houten, F.J.A.M. van, Kals, H.J.J., 1993, Computer support in the (re)design of mechanical products, a new approach in feature based design, focusing on the link with CAPP, in: Knowledge Based Hybrid Systems, Mezgár I., Bertók P. (eds.), IFIP Trans. B–11, North Holland, 91–103.

[14] Salomons, O.W., Slooten, F. van, Houten, F.J.A.M. van, Kals, H.J.J., 1993, A computer support tool for re–design, a prototype system resulting from applying a methodic design approach, proc. of ICED'93, The Hague, Vol.3, 1559–1570.

[15] Salomons, O.W., Jonker, H.G., Slooten, F. van, Houten, F.J.A.M. van, Kals H.J.J., 1994, Interactive feature definition, acc. for publ. at IFIP Conf. on Feature Modelling & Recognition in Adv. CAD/CAM systems, Valenciennes.

[16] Schmekel, H., 1992, A system for conceptual design based on general and systematic principles of design PhD thesis, Royal Institute of Technology, Stockholm.

[17] Serrano, D., 1987, Constraint management in conceptual design, PhD thesis, Cambridge, MIT.

[18] Sowa, J.F., 1984, Conceptual Structures, information processing in mind and machine, Addison–Wesley.

[19] Suh, N.P., 1990, The principles of design, Oxford University Press, New York.

[20] Taleb–Bendiab, A., Oh, V.,Sommerville, I., French, M., 1993, Semantic modelling of Engineering Designs, Journal of Engineering Design, Vol. 4, No.1, 41–54.

[21] Thornton, A.C., Johnson A., 1993, Constraint specification and satisfaction in embodiment design, proceedings of ICED'93, Vol.3, The Hague, 1319–1326.

[22] Ullman, D.G., 1993, A new view on function modeling, proceedings of ICED '93, The Hague, Vol.1, 21–29.