INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

ANNACON: ANNOTATION WITH CONSTRAINTS TO SUPPORT DESIGN

Lian Ding[1], Jason Matthews[1], Glen Mullineux[1]
(1) University of Bath, UK

ABSTRACT
The use of constraint-based and optimization techniques have proven valuable throughout the design process. Environments for such techniques are becoming available as stand alone and add-on packages for CAE environments. Most preliminary design layouts are performed on CAD environments; such environments also provide the designer with the mechanism to enrich such models with additional information, namely, annotation. This paper presents an approach that allows the designer to annotate the model with the design constraints, including its goals, relationships and bounds. These can then be used as the design evolves. To model solutions, that firstly, monitor and satisfy the constraints, and then present the opportunity to find an optimal solution. The context of this research is mechanism synthesis and machine system improvement. Therefore the constraints mainly dictate the geometric, topological and kinematics properties of a given design. The approach is demonstrated with the synthesis of a transfer mechanism in a production line.

Keywords: Design constraints, machine system design, product models, annotation

1. INTRODUCTION
The design process can be considered to be constraint oriented [1]. It involves the identification, negotiation and resolution of an evolving set of constraints. The nature of engineering design is such that a problem can be rarely considered as meeting only a single objective. In reality, the designer is trying to meet a whole range of goals, extending from basic functional requirements through to more vaguely defined parameters of style and form. In contemporary machine system design and development constraints not only come from the correct configuration of the system to produce the required function and satisfy company specifications, design rules and international standards, but there has to be a consideration of the interactions of a mechatronic environment: wiring, electronic circuitry and performance bounds induced by control software. It is not surprising, as the design evolves, that the designer misses or overlooks some of these constraints [2]. Missed constraints at the design stage can be expensive and time consuming to put right in the down stream activities. Nowadays most of the design activities are conducted in CAD environments, constraint-based modelling is also employed to aid a variety of the activities throughout this process (cf. figure 1). So, with the above point in mind, it would be advantageous to:

- Attach constraints at a ‘high level’ to the machine design representation as it is being produced.
- Retain the ability to automate the constraint monitoring process, so that all applicable constraints in a machine’s design are checked, without the designer having manually to initiate a search for constraint violations.
- Investigate the effects of design changes to configuration and assembly of mechanism designs and find failure modes states.
- Retain design constraint knowledge for collaborating or follow-on users.

The work presented in this paper is directed at the types of machine system design that takes place in small to medium sized enterprises. Very often such companies have a base range of products, which they offer. However the resources are such that there is not the expertise or time available to perform any in-depth analysis [3], For SMEs the modelling environment discussed in this paper has proved effective as a stand alone tool for such companies [4].

2. BACKGROUND: OVERVIEW

This section gives an overview of two aspects of design, related to the work presented in this paper. These aspects are: constraints in the design process and, the application of annotation to aid designer through the design process.

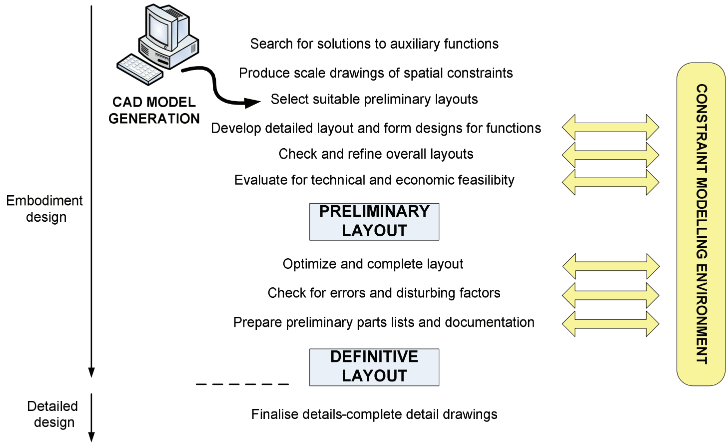

Figure 1. Later design stages (adapted from[5])

2.1 Constraints in the design process

The importance of constraints has previously been discussed in design activities discussed in Suh [5], Ullman [6] and Pahl and Beitz [7], and is the basis of the approach presented by Gross [1]. Such research has shown that the central element of the design process is the recognition, formulation, and satisfaction of constraints which are constantly added, removed and modified in an iterative fashion. Constraints can represent design rules, relations, conventions, and natural laws to be maintained and adhered to throughout the design activity [1, 2, 5]. Some constraints and objectives are given at the outset of a design but many more are adopted along the way partly as a result of a greater understanding of the design task being obtained. Varying the constraints and the objectives is part of the design process. The knowledge of the constraints that bound the limits of performance allows the designer to develop strategies to look firstly to define what the current performance of the design is. It gives the ability to search for potential optimal solutions to the design performance [3]. As the design progresses the understanding of the design space improves and further constraints become apparent. If some or all of the constraints are in conflict there is no feasible solution that simultaneously satisfies all the constraints. Figure 2 depicts regions of individual constraint satisfaction as circles. In Figure 2(a) these sets overlap and a fully feasible solution exists. However, in Figure 2(b), the constraints are in conflict and the intersection of the sets is empty. In this case, the experience of the designer is required to correctly identify the constraints that can be relaxed or removed in order to obtain a set of feasible solutions without jeopardising the quality of the final product.

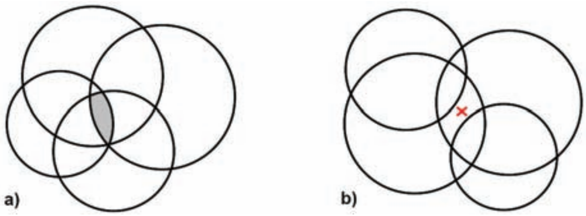

Figure 2 Constraints presented as sets

Constraint based approaches has shown merit throughout the design activity. A designer most commonly encounters constraint in the construction of design representations and models. These are generally geometric constraints, of two forms: numerical constraints, such as distance and angle, which gives numerical information and symbolic constraints, such as coincidence and parallelism. Discussion on their application and effectiveness can be found in the works of Hoffmann [8], Mullineux [9] and Anderl and Mendgen [10]. Work by Thornton [11], Johnston[12], Matthews *et al* [3] has provided engineers with problem solving and analysis tools at the embodiment stage and the early detailed stages of the design process. In general at these stages, the constraints are isolated from the designer within the CAD/CAE software. Such software systems offer the designer the efficient ability to construct 2D and 3D models and respective assemblies, but they do not always allow all the information a designer might require to be extracted or processed. Previous similar research to that presented in this paper has been undertaken by Bowen et al [13] who developed a constraint-based language capable of representing a design as a collection of variables. Constraints are then specified between these variables. Constraint monitoring is then employed to evaluate the design. The approach was limited due to the fact that all variables must be defined at the outset. An alternative approach has been proposed by the AKN group, who produced a stand-alone software package, Co-Editor [2]. This ontology-based environment allowed the designer to produce a design then edit and check constraints on configurations. Although limited in functionality, the approach shows merit for design tasks and for wider application within a commercial environment.

Current applications of constraints-based modelling aim to resolve conflicts in a proposed or existing design. It is common to apply such constraints at various levels: hard and soft constraints [3, 14]. The hard constraints relate to the function of the part or system, whereas the soft constraints relate to the performance of the system. It is generally the soft constraints that the designer has to resolve once the CAD model has been produced. When such constraints are employed there is a need to process/ handle the information they contain. To this end there are three types of handling approaches based on numerical techniques used for design and modelling: *Constraint monitoring:* used to check if the proposed design solutions provided do not violate any specified constraints; *Constraint satisfaction:* finds feasible solutions to constraints (without considering an optimal solution); and, *Constraint optimization:* aims to find the best solutions from alternatives in order to achieve the objectives, subject to constraints; this requires some measure of performance of a solution.

2.2 Application of annotation in design

Core to this approach is the ability to 'attach' the constraints to elements/ features of the designed mechanism. One well established method of attaching additional information to a CAD representation is annotation. Annotation can be simply understood as an act of adding extra information. Forms of annotation have long been a part of engineering practice [e.g. 15, 16]. Annotation is used to aid communication both:

- *Direct*, such as engineers annotating a depiction of a design during a face-to-face discussion of the product
- *Indirect*, such as an engineer sending an annotated depiction of the product to a colleague.

With the rapid development of digital techniques, many CAD systems have already introduced 'annotation' support. For examples, AutoCAD 2008 offers automatically scaled annotation capability;[17]; SolidEdges V19 supports the ASME Y14.41 standards for 3D annotation [18]; Pro/ENGINEER provides annotations of dimensions, tolerances, surface finish, etc.[19]; NX5 allows users to create annotation manufacturing information [20]; and CATIA V5R13 supports creation annotation like tolerance information [21]. However, such 'annotation' functions provided are normally limited to a single type (e.g. free-text, URI or numerical format), they cannot satisfy the requirements of annotation schemes supporting context-specific information and multiple viewpoints. Furthermore, the annotation strategy normally adopted is the 'inline' annotation' method, in which the content of the annotation is actually associated with product models, and therefore it is only reusable for invariant topologies. More importantly, it is difficult to place multiple independent sets of annotation in the same document as the syntax of the tokens used may easily interfere with one another [22]. Thus, an extension to existing 'annotation' functions is really needed. Ding *et al* [23] have explored the possibility of using '*stand-off*' annotation methods on the CAD models, which the means that the annotation is stored in a separate external document utilizing a system of references or pointers to indicate the element (such as a face, or a feature within a CAD model) to which the annotation refers. The 'stand-off' annotation has many advantages. It is a good mechanism to support multiple viewpoints. The information pertaining to one viewpoint can be put in a separate annotation file; and multiple independent annotation files can be safely applied to the same CAD model. Thus, it allows context-specific information to be manipulated into different viewpoints, freely tailored to a reduced version for reasons of security/IP and the requirements of different target users, and re-organized for various purposes and applications. It allows support of knowledge sharing throughout a products lifecycle [24, 25, 26]. The same annotation file can be applied to any copy of the model in any format, so that the annotation information can be independent from the model format and shared by various users. It is important that all users can contribute their experiences and knowledge by separate annotation rather than by editing the CAD models directly.

2.3 Background summary

Prior research has shown that modelling a design problem using constraints is useful, but:

- During the initial construction of representations, the designers often miss important constraints [2]
- Generally the user is isolated from the respective constraints, limited ability to employ the constraint handling techniques [3].

Meanwhile, 'stand-off' annotation has shown the capabilities of insertion of multiple context-specific information and being independent of model format (e.g. CAD model). The question is whether 'stand-off' annotation is able to overcome the issues identified for previous constraint based design applications. In other words, how should annotation be used effectively to drive constraint based design? The 'stand-off' annotation method offers many advantages. Firstly, the same annotation file can be applied to any copy of the model in any format; this allows the annotation information to be shared by a wide variety of users. Secondly, multiple independent annotation files can be safely applied to the same CAD model. Thus, it allows context-specific information to be stored in a number of separate files and passed around only as needed. Thirdly, annotations can be edited, circulated, and processed independently of the model, and the CAD model remains unchanged.

3. INITIAL IMPLEMENTATION: ANNACON

To resolve the problem identified in section 1, an approach using constraints and standoff annotation has been created. This section introduces the initial implementation of the approach. The overall approach is shown in figure 3.

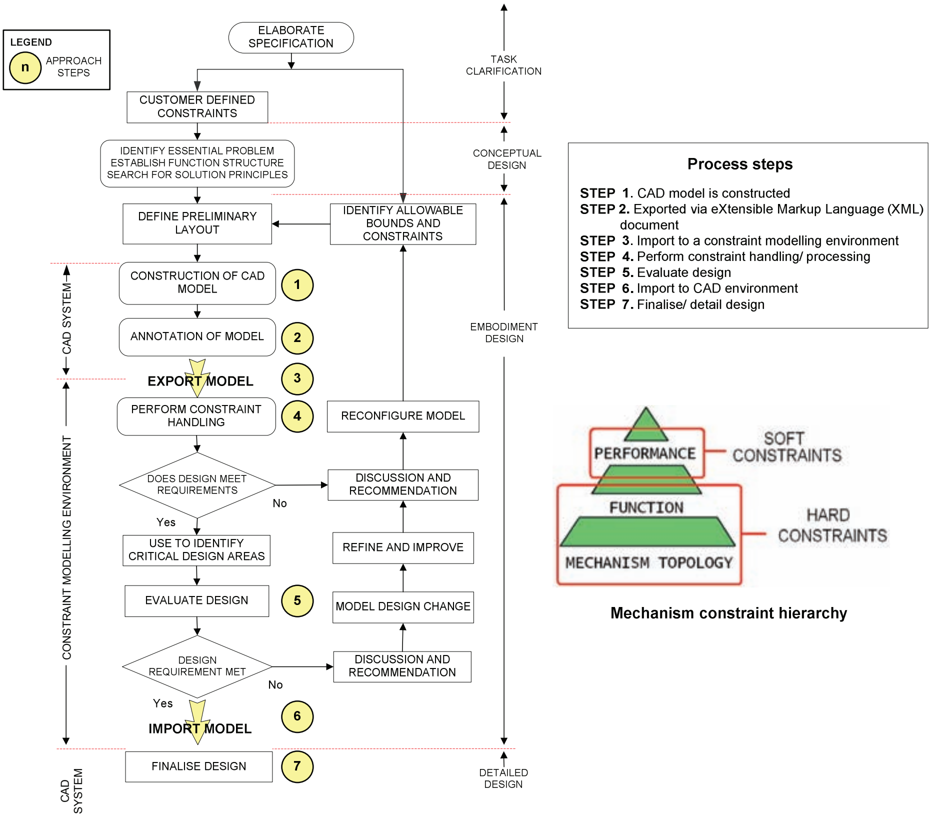

Figure 3. Implementation approach

3.1 Implementation of a prototype system

To demonstrate the proposed approach, a prototype system has been implemented within a commercially available design software package: Unigraphics NX3. Visual C++ and Microsoft XML Core Services (MSXML) are used to write/read the XML-based annotation files. The NX3 Open C API is employed to build a reference system between the annotation files to the specific entities of the CAD model that the annotations refer to. In addition, the UIStyler dialogue in NX3 has been utilized to develop a friendly user interfaces.

3.2 Constraint modeller

The constraint modeller [9] was created to help understand the design process. The software is designed to handle evolving design structures, shifting design goals, and changing design constraints. The constraint modeller is an interactive tool allowing users to easily adjust design parameters, and modify, add or remove design constraints. It allows the user to quickly generate feasible solutions and the corresponding set of design parameters. It provides an interface where the influence of the design constraints can be ascertained and the effect on the resulting design can be studied and visualized. The interface language of the constraint modeller is based on an underlying C-like syntax. This language is used to declare and assign values to the design parameters, to create geometric objects such as points, lines and curves, and to define and solve design constraints. The language includes much of the common functionality of C such as array structures, mathematical operations and functions. In addition, other bespoke functions required to model engineering systems are included. These include numerical differentiation algorithms for kinematic analysis of mechanisms, curve fitting procedures for defining cam profiles, and various visualization tools. The language also supports user defined functions which can be invoked as required. The body of the function itself can consist of any combination of other user-supplied commands and commands from the interface language. Functions can take any number of input variables and can return any number of output values. An important inbuilt function is the rule command. Each time the rule command is invoked a "constraint rule" is defined which is associated to a mathematical expression written in terms of some or all of the design parameters. For example if `vel` is a design parameter representing the velocity of a point in a mechanism then the following statement defines the constraint rule specifying that the velocity of the point should be equal to 10.0.

```
(rule( vel - 10.0 );)
```

Any number of `rule` statements can be used to build up the constraint set of a given problem. During resolution the constraint expression for each rule command is evaluated and the sum of the squares is calculated. If this is zero then the constraint set is already satisfied, if not the constraint resolution process commences. The constraint software uses numerical optimization methods [27] to minimize the objective function formed from the sum of the squares of the constraint expressions.. The sum of the squares of the constraint expressions are used to generate the corresponding objective function. Mathematically, this problem is written in equation 1

$$\text{minimize } f(\mathbf{x}) = \Sigma f_i(\mathbf{x})^2 \quad (1)$$

where $\mathbf{x}$ is the vector of the n design parameters and $f_i(\mathbf{x})$ corresponds to the i-th constraint rule. For practical problems the "best compromise" solution may be unacceptable because the corresponding design violates one or more essential constraints or physical laws. In this case additional weighting terms can be added to high priority constraint expressions and the resulting objective function is then defined equation 2

$$f(\mathbf{x}) = \Sigma\, [W_i f_i(\mathbf{x})\,]^2 \quad (2)$$

where W_i is the weighting term corresponding to the i-th constraint rule. Large relative weighting terms act as penalty factors against the violation of the corresponding constraint rule and help to ensure these more important constraints are satisfied. All algorithms for optimization problems require at least one set of design parameters to use as a starting point denoted by $\mathbf{x}_0$. From $\mathbf{x}_0$ a sequence is generated $\mathbf{x}_k$ that terminates when a solution has been found to the required accuracy or when no further progress can be made. Methods differ by how they move from one iterate to the next with a lower value of the objective function. The simplest numerical algorithms for the solution of optimization problems are direct-search methods [9].

3.3 What needs to be annotated?

For this approach to be successful, there are three particular aspects of information that the constraint modelling environment requires:

- *Transformation of the entities:* The constraint modelling environment adopts simple wireframe entities, the basic geometric of which are points, lines and circular arcs. Thus, the transformation of the 3D solid entities in a CAD model to the wireframe entities is needed. It is different from the transformation to other lightweight representations [28], which not only depends on the geometry and topology, but also depends on the functions of the entities in the system.
- *Constraints:* Constraints can be broadly classified into two types: hard and soft. Generally, the hard constraints are concerned with assembly, which ensure that the various parts of a system connect together correctly [14].
- *Definition of variables:* The variables refer to two key aspects:
    - The variables for optimization process refer to the variables that are needed for building constraints and execute optimization process.
    - The variables for optimization of a product represent the variables that define geometry and topology of the product, and can be updated after optimization. Such variables must link with the parameters that appear in the CAD model.

3.4 XML schema for the annotation document

A technique/ approach is required to pass data between the constraint modeling environment and the CAD environment. Building on previous work by the authors [23, 24] XML is employed. XML is a descendant of SGML (the Standard Generalized Markup Language, ISO 8879) and became an ISO standard in 1998. XML offers many benefits, including: XML is both computer interpretable and human readable. XML tag names are normally transferred from the meaning of the data and therefore they are readable; XML allows users to define their own tags based on the specific needs of a document, and therefore it is extensible; and XML lies in the separation of the information and its presentation so that the content of the XML file can be transformed for different viewers, devices and applications.

The structure of the XML schema aims to record the information presented in section 3.3, and have the capability of subsequent structuring and manipulation of the information into a computer-processable form. In addition, the structure of the schema must be open so as to be easily expanded to other application domains. The general structure of the XML schema is designed to be flat, which consists of two main sections: *header section* and *main body*. The *header section* records the metadata for the annotation file and the corresponding references to the CAD model, including the path and the file name of the CAD model (*model_file*), the specific element that the annotation links to (*geometry_reference_list*), the people who insert the annotation (*editor_list*), the dates of creation and the latest modification (*date*); what particular viewpoints the annotation belongs to (*viewpoints*), e.g. constraints optimization, etc. The *main body* describes the detailed annotation information. The structure of the *main body* could be a special structure, which is open for users to define, as the number of different possible structures of information required for various applications throughout a product lifecycle is effectively infinite. For the constraint-based design, the structure of the *main body* is specified as: constructing information of the product (*assembly*), defining information of the motion added on the product (*motion*), and optimising information for the product (*optimization*).

- The *assembly* further includes three aspects: *assembly_name, model_space, parameter_definition* and *geometry*. The *assembly_name* is the name of the assembly, which means a component can own multiple annotation files to support different assemblies the component is involved, and only the annotation files for the particular assembly are used. The *model_space* defines the local geometry that is specially adopted by the constraint-based language [9]. It is described by the *model_name* and the *model_command*. The *parameter_definition* records the parameters that actually define the geometry and topology of a product, and the variables needed for the constraints. According to different types of the parameters, it is divided into *int_definition*, *real_definition* and *entity_definition*, each of which further consists of name and a value. Here, the value for the parameter represents the initial value before the optimization, and it is able to be updated during the optimization and then finally used to update the CAD model. The *geometry* stores the information that transforms the entities in the CAD model to the wireframe entities defined in the constraint-based environment.
- The *motion* generally gives two aspects of the information: the motion information that is added to the product, and what output needs to be measured. For example, a circle-driving motion, it includes which model space it belongs to, what kind of motion, definition of each motion step, and the output entity.
- The *optimization* records constraints in the product. As discussed earlier, it includes the hard constraints and the soft constraints. According to different kinds of constraints, it can be further divided, such as the connection link, the assembly link, and the speed requirement.

3.5 Generation of the annotation files

An XML-based annotation file can be generated by collecting the corresponding information and written based on the XML schema. The information that is required for the optimization can be gained by three major ways:

- *Initialisation of the template annotation file:* Each component model can have one or more template annotation files, each of which includes some standard information required for the constraint-based environment. These template annotation files can be initialized once the component is added to an assembly. For example, a triangle-shape component is normally transformed into three lines in the constraint-based environment. Such information can be stored in an annotation file with the CAD model of the component, and used to generate a new annotation file by initialising the parameters giving the appropriate co-ordinates, lengths and model spaces when the component is used by a new assembly.

- *Capturing directly from the CAD model:* A lot of information needed for the constraint-based environment that can be extracted from the CAD model. For example, a mate-assembly usually refers to a hard constraint – connection link.
- *Input of information manually:* It is always possible there is extra information required for the constraint-based environment but that cannot be obtained from the CAD model. To support users on easily inputting such information, various interfaces have been developed, such as the interface to input hard constraints like connection link, the interface to input soft constraints and the interfaces to input transformation information.

4 INITIAL CASE STUDY

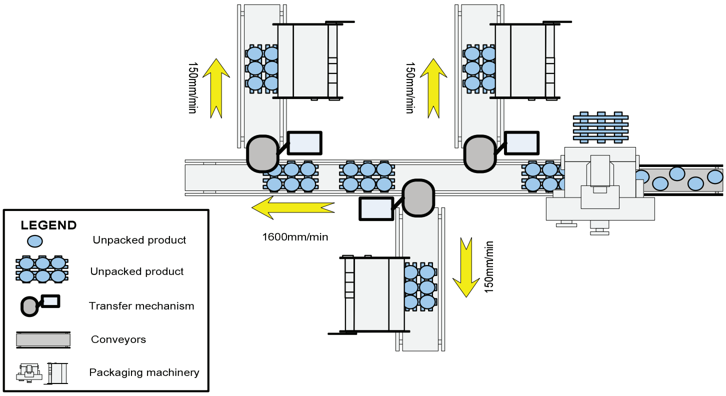

Figure 4. Machinery layout

The case study is a transfer mechanism used to pick-and-place produce from one conveyor to another conveyor. Given new package and speed requirements (as shown in Figure 4), the design needs to be modified and optimized: the speed at the pick-up position must be 1600mm/sec in the x direction, and the speed at the drop-off position must be 150mm/sec in the y direction.

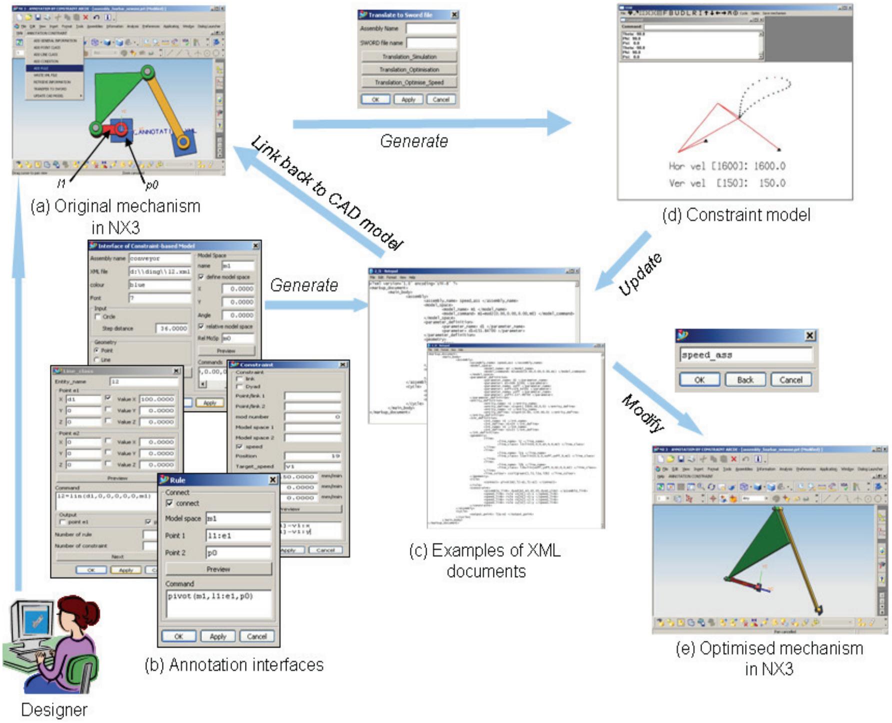

Figure 5 Case study

Figure 5(a) shows the original mechanism of the machine. Users annotate the requirements through various interfaces within the CAD environment, which are shown in Figure 5(b). All the new requirements are stored into separate XML-based annotation documents, which are linked back to the CAD model. In this case, there are five XML-based annotation files, which are linked with the base and the moving links. Figure 5(c) shows the annotation files. Based on these annotation files, a developed module within the CAD system is used to generate a constraint model, which can be used to optimize the design. The XML files form a record of the constraints as a design evolves. Figure 5(d) shows the constraint model. After optimising the constraint model, the annotation files are modified. Finally, the modified annotation files are used to update the original mechanism. Figure 5(e) shows the optimized mechanism in the CAD environment.

5 CONCLUSION AND FUTURE WORK

As design has change to become increasing knowledge-intensive covering broad areas, the constraints include not only from traditional design issues, like structural mechanics, control systems, material properties, but also from the requirements at later stages of a product lifecycle, such as manufacturing process, inspection, service feedback and various human and business related issues. It is easily for designers, especially for new designers, miss important constraints, which could pay a high penalty cost later. Both academics and software industry have shown their interests on these issues and some supporting tools have been developed, such as Coeditor and annotation tools in CAD systems. However, the major results are still limited to geometrical constraints, numerical calculations and small parts of manufacturing rules. This paper proposed a new strategy using the strengths of stand-off annotation combined with constraint modeller to allow the association of product data throughout the lifecycle with geometric form of the product. A prototype implementation has been developed and a case study has been provided to demonstrate the proposed method. Although the initial case study has been aimed at the machine design, the process and outcome would be beneficial to design activities that span across different domains and various resource rich companies. Comparing to other method, the proposed AnnaCon has the following advantages:

1. The stand-off annotation method records the constraints in separately documents so that it allows users to retrieve/reuse the constraints outside of CAD systems. On the other hand, the knowledge and new constraints required from other users (e.g. collaborative partners, or users at later stages of a product lifecycle) can also be feedback to the CAD models directly via separate annotation documents.
2. The constraints are individually recorded in XML-based format. XML is not only computer interpretable, but more important it can be assembled, tailored and transformed in different ways (e.g. through XSL/XSLT), for example, combining constraints pertaining to one viewpoint or one optimization process and tailoring subsets of the constraints for different users according to different level of security. Although the demonstrator shows the capability on how the annotated constraints to be used in the constraint modeller, the method can be easily extended to support any stand-alone software.
3. The templates of standardized constraint annotations can be expanded by new applications and requirements, and therefore further support various design tools. In addition, the templates of standardized constraint annotations can not only help designers to escape respective work, but also minimize design errors and speed the design process.

As design evolves, some constraints are added and others are removed. Thus, the future stream of research is to investigate the rationale for these changes and looking to see whether knowledge management techniques like data mining can be used to reorganize annotations so as to aid future designs of similar machine and avoid making the same 'wrong' design choices. Also to investigate how effective and 'rich' constraints could be for a design rationale indicator. In addition, to avoid loss of stand-off annotations when a CAD model is changed, further work is need to explore how annotations should cope with changes to underlying model

ACKNOWLEDGMENTS

The ideas reported in this paper have emerged from work supported by a number of grants funded by Engineering and Physical Sciences Research Council (EPSRC), Department of Trade and Industry and Department for Environment Food and Rural Affairs (DEFRA), involving a number of industrial collaborators. In particular, current research is being undertaken as part of the EPSRC Innovative Manufacturing Research Centre at the University of Bath (reference GR/R67507/01). The authors gratefully express their thanks for the advice and support of all concerned.

REFERENCES

[1] Gross, M.D., Ervin, S.M., Anderson, J.A., and Fleisher, A. Constraints: Knowledge representation in design. *Design Studies* 9, pp133–143, 1988.
[2] Ajit S., Sleeman, D., Fowler D.W. and Knott D., ConEditor: Tool to Input and Maintain Constraints. In Proceedings of EKAW 2004 Conference (Whittlebury Hall, UK), Springer, Heidelberg, pp466-468, 2004.
[3] Matthews, J., Singh, B., Mullineux, G., Medland, A. J. A constraint-based approach to investigate the process flexibility of food processing equipment. *Computers and Industrial Engineering*. 2006. 51(4), pp809-820.
[4] Neale, G., Mullineux, G., Matthews, J and Medland, A. J. Case study: Constraint-based improvement of an over-wrapping machine. *Proceedings of IMechE part B*., 2009, 223(2), 207-216.
[5] Suh, N. P (1990) *The Principles of Design*, Oxford University Press. ISBN 0-19-504345-6.
[6] Ullman, D.G (1992) *The mechanical design process*, Mc-Graw-hill, USA, ISBN 0-07-065739-4.
[7] Pahl, G. and Beitz, W., Engineering Design A Systematic Approach, Second edition, Ken Wallace (Ed.), Springer-Verlag (London), 1996.
[8] Hoffmann, C Constraint-based CAD *J. Comput. Inf. Sci. Eng*., 5(3), pp. 182–197, 2005.
[9] Mullineux, G. Constraint resolution using optimization techniques. Computers and Graphics, 25(3):pp483–492, 2001
[10] Anderl, R and Mendgen, R Modelling with constraints: theoretical foundation and application. *Computer-Aided Design*, 28(3):pp155–168, 1996.
[11] Thornton, A. C. The use of constraint-based design knowledge to improve the search for feasible designs. Engineering Applications of Artificial Intelligence. 1996, 9(4) pp393-402.

[12] Johnson, A.L and Thornton, A. C. Towards real CAD, *Design studies*, 1991 12(4).pp232-236.
[13] Bowen, J. and Bahler, D, Supporting cooperation between multiple perspectives in a constraint-based approach to concurrent engineering, *Design and Manufacturing*, 1, pp89-105, 1991.
[14] Dechter, R (2003) Constraint Processing, Morgan Kaufmann Publishers. USA.
[15] Catalano C., Giannini F., Monti M. and Ucelli G., A framework for the automatic annotation of car aesthetics, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 21, 73-90, 2007.
[16] Hisarciklilar, O and Boujut, J-F An annotation-based approach to support design communication. *Procs International Conference on Engineering Design (ICED'07),* August 2007, Paris, France.
[17] Fane, B., Autodesk AutoCAD 2008 (Cadalyst Labs Review) – New annotations are the name of the game, *Cadalyst,* April, 2007, http://manufacturing.cadalyst.com/manufacturing/article/articleDetail.jsp?id=417300&pageID=2 (accessible on 24/04/09)
[18] Ferris, S., NX CAM Express is latest tool in Velocity Series for midsize companies, *Cadalyst,* Jun 2006, http://manufacturing.cadalyst.com/manufacturing/article/articleDetail.jsp?id=333200
[19] Rowe, J., Cadalyst Labs Review: PTC Pro/ENGINEER Wildfire 3 Workflow efficiency and productivity take center stage, *Cadalyst,* Sep 2006, http://management.cadalyst.com/cadman/article/articleDetail.jsp?id=370276
[20] Rowe, J., Cadalyst Labs Review: NX5, Part2 – Enhancements in software architecture enable major improvements in assemblies, visualization, and drawings, *Cadalyst,* Dec 2007, http://manufacturing.cadalyst.com/manufacturing/article/articleDetail.jsp?id=475124
[21] CATIA - 3D Functional Tolerancing and Annotations, 2(FTA), CATIA V5R19, http://www.catia.cz/fileadmin/Pictures_Menu/Catia/Catia_V5/Mechanical_Domain/pdf/FTA.pdf
[22] Tennison, J., Creole Q. Validating overlapping markup. In: XTech 2007: The Ubiquitous Web. Paris. IDEAlliance: Alexandria, VA. Available from: http://2007.xtech.org/public/schedule/paper/46. (accessed 13/12/08).
[23] Ding, L., Davies, D. and McMahon, C.A., The integration of Lightweight Representation and Annotation for Collaborative Design Representation, *Research in Engineering Design* , 2009, 19(4) 223-227.
[24] Ding, L., Davies, D. and McMahon, C. A, Sharing Information throughout a Product Lifecycle via Markup of Product Model, *Proceedings of the ASME 2008 International Design Engineering Technical Conference, Computers and Information in Engineering Conference (IDETC/CIE 2008)*, August 3-6, 2008, Brooklyn, New York, USA
[25] Patel, M., Ball, A. and Ding, L., Strategies for the Curation of CAD Engineering Models, 4th International Digital Curation Conference "Radical Sharing: Transforming Science?", 1-3 December, 2008, Edinburgh, Scotland, U.K., The International Journal of Digital Curation.
[26] Ding, L., Matthews, J., McMahon, C. A and G, Mullineux, G. An Extended Product Model for Constraint-based Redesign Applications, *Procs International Conference on Engineering Design (ICED'07)*, August 2007, Paris, France.
[27] Fletcher, R. (2000). *Practical methods of Optimization*, 2nd ed., Wiley London. UK.
[28] Ball, A., Ding, L., Patel, M., An approach to accessing product data across system and software revisions, *Advanced Engineering Informatics*, 22(2), 2008, pp22-34.

Contact: Lian Ding
University of Bath
Department of Mechanical Engineering
Bath, BA2 7AY
UK
Tel: Int +44 1225 386138
Fax: Int +44 1225 826928
Email: ld218@bath.ac.uk

Dr. Lian Ding is a researcher at the Innovative Design and Manufacturing Research Centre in Bath. She received her bachelor degree (1993) from Anhui Institute of Technology and master's degree

(1997) from Southeast University in China. She became a lecturer at Southeast University after graduation, teaching a variety of courses in mechanical engineering and conducting research in mechanical engineering. She then undertook doctoral research at the University of Luton, gaining a PhD in 2003. Her current research interest is in the area of computer integrated manufacturing systems. She has published one book and over 40 refereed journal and conference papers.