SANDIA REPORT

SAND96-3004

Unlimited Release

Printed December 1996

Liaison Based Assembly Design

Arlo Ames, Deepesh Kholwadwala, Randall H. Wilson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

SAND96-3004
Unlimited Release
Printed December 1996

Liaison Based Assembly Design

Arlo Ames
Advanced Engineering and Manufacturing
Software Development Department

Deepesh Kholwadwala
Advanced Engineering and Manufacturing
Software Development Department

Randall H. Wilson
Intelligent Systems Principles Department

Intelligent Systems and Robotics Center
Sandia National Laboratories
Albuquerque, NM 87185-1010

Abstract

Liaison Based Assembly Design extends the current information infrastructure to support design in terms of kinematic relationships between parts, or *liaisons*. These liaisons capture information regarding contact, degrees-of-freedom constraints and containment relationships between parts in an assembly. The project involved defining a useful collection of liaison representations, investigating their properties, and providing for maximum use of the data in downstream applications. We tested our ideas by implementing a prototype system involving extensions to Pro/Engineer [9] and the Archimedes assembly planner [7]. With an expanded product model, the design system is more able to capture design intent. When a product update is attempted, increased knowledge availability improves our ability to understand the effect of design changes. Manufacturing and analysis disciplines benefit from having liaison information available, so less time is wasted arguing over incomplete design specifications and our enterprise can be more completely integrated.

Acknowledgments

Special thanks goes to Steffen W. Parratt for help in developing basic liaison concepts in their early stage, Max G.Q. Chen for contributions to the liaison testbed software, Kenneth Eras for assistance in finding and understanding test assemblies, and Rob Bugos for being an eager customer.

Table Of Contents

Background.....	1
Previous work.....	1
Technical problem.....	2
Accomplishments.....	2
Liaison Representation.....	2
Liaison Promotion.....	4
Interfacing Liaisons to a Production CAD system.....	5
Using Liaisons in Novel Design.....	9
Functional Positional Relationships.....	10
Testing.....	10
Impact.....	11
Future directions.....	11
Migration to other CAD Platforms.....	11
Future Developments.....	11
Conclusion.....	12
References.....	13

Figures

1. Integration of liaison's GUI with existing menus.....	6
2. Symbols used to display liaisons graphically.....	7
3. Graphical display of press fit liaisons.....	8
4. A revolute (one degree of freedom) liaison.....	8
5. Two link mechanism with a revolute joint.....	9
6. Two link mechanism in early design process.....	9

Tables

1. Data requirements to specify kinematic liaisons.....	3
2. Data requirements to specify zero degree of freedom liaisons.....	4

Background

Feature-based modelers, such as Pro/Engineer [9], have taken the CAD market by storm. They provide a natural way of communicating design information to the computer, provide significant speed improvements, permit parametric variation, and archive the design. They provide approaches to creating and modifying part design that have moved solid modeling from a research curiosity to a practical design tool.

Modeling assemblies, however, is another matter. Current methods of designing mechanical assemblies are limited to defining relative positioning of the parts, with no regard for *why* the parts are in the relative positions or how to manufacture the relationships. In addition, current modelers require geometric relationships to specify 6 degrees of freedom, even if this means over- or under-specifying the part-part interactions (a rotational relationship could be simplified to require only that 2 axes be aligned). Once data leaves the CAD system for analysis or manufacturing planning, we spend time calling the designer on the phone to ask about this relationship information so that proper analyses and manufacturing planning can be performed.

Historically, we have used notes and symbols on drawings to communicate some of the part-part interaction information. It is possible with some solid modelers to attach information in the form of notes, but such notes would still be unintelligible to automatic analysis and planning applications. We have an opportunity to create more powerful representations for capturing part-part information and making it available in sensible form.

By providing standard ways to capture functional intent and use that functional intent to drive downstream activities, we believe we can provide the same kind of leverage to mechanical design that schematic modeling provides to electrical circuit design.

Previous Work

Feature-based modelers permit users to design component parts in terms of form features, such as holes, slots, fillets and chamfers. Form features are evaluated to produce a geometric model. Such an interface significantly improves the speed of the design process over modeling parts in terms of raw geometry and permits parametric variations of a part design. However, they fail to improve modeling of *assemblies*. They can document the presence of a hole, but do nothing to document what purpose the hole serves.

Commercial CAD systems support assembly modeling, but only at a rudimentary level. Assembly-level feature based modeling has been addressed by a number of efforts in the academic world (e.g. [1-6]). Similar to current commercial CAD systems, these efforts focused largely on part positioning, while ignoring the value of the features themselves. A number of previous efforts (e.g. tolerance analysis) have constructed application specific assembly models which are not sufficiently general. We build on these previous works by supporting improved semantics in both topological and geometric relationships between parts.

Technical Problem

The technical problem we address is defining the necessary representations and algorithms to permit assembly relationships to be modeled. These representations must be natural and appropriate for use in mechanical design, and should provide significant value to downstream applications.

A complete assembly modeling system will need to support a wide variety of part-part relationships. This project develops a first step towards complete assembly modeling, that is, modeling in terms of kinematic interactions between components of an assembly.

Liaison features can be viewed as a language for describing assemblies; liaisons are to assemblies what form features are to parts. Kinematic liaisons can represent degrees-of-freedom constraint (rotational and translational constraints) and contact information (point-point, point-edge, point-face, edge-edge, edge-face and face-face). Liaisons include non-geometric information to expand their representational capability for design, analysis, and manufacturing applications (e.g. weld parameters, thread specifications).

We approached the problem by defining a meaningful, reasonably powerful set of relationships between parts of an assembly and developing representations for these liaisons in a testbed (Pro/Engineer). Our initial set of liaisons concentrate on completely specified relationships between ideal parts. After developing the ability to represent liaisons, we worked to communicate liaison information to downstream applications, e.g. the Archimedes [7] assembly planner. To prove effectiveness, we tested our concepts on a "real" assembly, the pin-in-maze discriminator. Testing liaisons included both measuring the ability of the representation to capture the salient facts about the assembly and the suitability of this information for automating analysis and manufacturing processes. Finally, we investigated *functional part-part positional relationships*, that is, representations that would permit modeling the assembly purely in terms of liaisons, with the modeler inferring relative positions from the functional specification.

Accomplishments

Our work in liaison modeling made progress in the following areas: representation, liaison promotion, interface with a production CAD system, and use of liaisons very early in the design process.

Liaison Representation

Current solid modeling systems capture only relative positional relationship between parts, without representing why the parts have those positional relationships. In an attempt to remedy this deficiency, we developed data structures that extend Pro/Engineer to support the concept of liaisons, or the interactions between parts. We investigated the data requirements for specifying kinematic/dynamic relationships (using DADS [8], a product of CADSI), automatic assembly planning (with Archimedes, a Sandia-developed robotics assembly planner), and finite element analysis (as required for internally developed finite element analysis codes). We found a significant degree of commonality among the kinds of relationships required, and significant variation in the specifics of data requested.

Kinematic/dynamic analysis placed the fewest requirements, focusing on the types and locations of joints, without requiring knowledge of what local part geometry participates in the relationship (e.g. only requiring the location of an axis to specify a rotational joint). By contrast, finite element analysis requires detailed representation of which surfaces will be in contact for each joint, so that appropriate boundary conditions can be specified when meshes are generated. Assembly planning requirements are more manufacturing-driven, being focused on joint location and process (e.g. weld, glue, etc.). Additionally, assembly planning helped to define the notion of using fasteners.

The initial set of liaisons were defined to specify kinematic relationships. They include *revolute*, *translational*, *screw*, *cylindrical*, *planar*, *point-curve*, *point-surface*, *roll-curve*, *slide-curve*, *spherical* and *gear* joints. The data requirements for each kind of joint have been documented in the table below.

Table 1: Data requirements to specify kinematic liaisons.

<i>Liaison</i>	<i># of parts</i>	<i># of axes</i>	<i>unit direction</i>	<i># of points</i>	<i># of faces</i>	<i># of edges</i>	<i>supplement info</i>
Revolute	2	1	1	0	0	0	-
Translational	2	1	1	0	0	0	-
Spherical	2	0	0	1	0	0	-
Point-Curve	2	0	0	1	0	n	-
Point-Surface	2	0	0	1	n	0	-
Curve-Curve	2	0	0	0	0	n	-
Surface-Surface	2	0	0	0	n	0	-
Screw	2	1	1	0	0	0	-
Gear	2	2	2	0	0	0	ratio

The assembly planning application focuses on how liaisons are assembled, rather on how the resultant assembly works. The liaison definitions are extended to include process information, such as type of *joint* (*glue*, *weld*, *press-fit*, *snap-fit*, *fasten*) and process parameters (e.g. weld diameter and depth). For the current Archimedes assembly planner, the focus is largely on zero DOF joints, since they represent a rich domain of manufacturing information, while higher-degrees-of-freedom liaisons involve fixturing technology not yet available in Archimedes.

Table 2: Data requirements to specify zero degree of freedom liaisons.

<i>Liaison</i>	<i># of parts</i>	<i># of axes</i>	<i>unit direction</i>	<i># of points</i>	<i># of faces</i>	<i># of edges</i>	<i>supplement info</i>
Spot Weld/Glue	2	0	0	n	0	0	depth, diameter
Surface Weld/Glue	2	0	0	0	n	0	depth, diameter
Curve Spot Weld/Glue	2	0	0	n	0	n	depth, diameter
Curve Weld/Glue	2	0	0	0	0	n	depth, diameter
Fastener	2	1	1	0	0	0	-
Press Fit	2	1	1	0	0	0	-
Snap Fit	2	1	1	0	0	0	-

In discussions with a potential future user of liaisons, we learned of an additional requirement: the ability to impose limits on degrees of freedom. Rob Bugos, a designer of radar assemblies, asked that we develop the ability to place limits on rotational liaisons, to enable him to design assemblies that operate within a constrained envelope. This addition reflects a Sandia-specific need that we did not uncover in any of our studies of tools, indicating a need that is potentially not being met by Sandia's current tool set. Implementation entailed merely adding more attributes to the liaisons. This experience suggests the possibility that liaisons need to support the addition of arbitrary attribute data.

In order to support finite element analysis completely, we need to add the capability to support loads and boundary conditions. A direct approach to providing the needed capability would entail providing point, edge, face, subedge and subface- based geometries, with a simple attribute used to label nodesets, sidesets, etc. While such an approach is easy to implement, the data provided would be specific to one application.

Liaison Promotion

We identified requirements for "promoting" liaisons from one form of representation (e.g. purely joint-based) to another (e.g. with manufacturing information added). Under such a scheme, the set of relationships between parts doesn't change, but the amount of information attached to a given liaison grows to support increasingly more applications.

In a simple example, consider a liaison between a gear and a plate. A very simple liaison between these two parts would be a rotational joint, represented as an axis that relates the two parts. From such a relationship, we can perform kinematic analyses of the assembly. Later, we decide that this rotational joint relationship will be implemented by attaching a pin to the plate by pressing the pin into a hole and welding it in place, then press fitting a bearing into the gear and pressing the bearing onto the pin. From this new relationship, we can perform more sophisticated kinematic analysis, as

well as assembly planning (because we now have process information), and other analyses. The original liaison information, representing conceptual design intent, can still be present in the model, along with information that it is implemented in terms of other parts and liaisons.

Interfacing Liaisons to a Production CAD System

In order to test the feasibility of modeling in terms of liaisons, we developed extensions to Pro/Engineer, through Pro/Develop [10], to create and maintain liaison representations. Developing a liaison modeling capability involves defining data structures to represent the geometry and attributes of each liaison, ensuring that those data structures can survive operations such as file save and regeneration, providing an interface to create, modify and delete the data, providing means to display liaisons in the graphical context of assemblies, and developing technology for exporting liaison information to outside applications.

Development involved two attempts at representing liaisons. In the first attempt, we tried to make maximal use of Pro/Engineer data structures. We created each liaison as a datum point or curve, and attached user attributes to that datum. Development was difficult, as creation of the datum curves involved defining user defined features (UDFs), and this UDF functionality is notably some of the least effective Pro/Develop functionality. This original representation survived regeneration and file save trivially, but was not extendible to surface-based features or represent liaisons with multiple geometric references (such as gears), and was very difficult to maintain.

Our second attempt at liaison representation was less tightly coupled to Pro/Develop functionality. We defined C++ objects to represent liaisons. Each class contained attribute data and pointers to Pro/Engineer geometry. Because we used pointers rather than UDFs, we were able to represent arbitrarily complex collections of geometry, and reference every type of geometry.

The pointer-based representation involved more effort to maintain consistency with Pro/Engineer under modeling operations. The original representation was automatically saved whenever the geometry was saved; the new representation wasn't directly maintained by Pro/Engineer, so it required special consideration to make sure file saves and restores maintained the C++ objects. Feature modifying operations (suppression, deletion, parameter modification) had the potential for invalidating the C++ representation. These concerns necessitated that Pro/Develop notify functionality be used to track model changes, so that the liaison representations could be properly maintained.

Although the pointer-based representation entails significantly more programming effort, it provided more powerful representations that are much less dependent on Pro/Engineer. The programming effort to port liaisons to a new modeler is limited to defining mechanisms to reference geometry, and discovering how to track state changes in the modeler. We have proven this by implementing a capability for exporting liaisons that uses most of the liaison classes directly, with the only changes being in how geometry is referenced.

Our current representation is not complete. Pro/Engineer provides no mechanisms for directly controlling or changing the topology of models being created, so our liaison representations in Pro/E are limited to referring to complete edges and faces. The limitation has very little bearing on kinematic representations, but has profound consequence for finite element analysis, where it is necessary to have extremely precise control over where liaison information resides. Pro/FEM has

some capability for defining portions of surfaces, but that capability is not available to any other applications, and lacks robustness.

Building a natural liaison interface to the CAD system is necessary. Figure 1 shows the seamless integration of liaison's GUI (graphical user interface) with Pro/Engineer. Developing such an interface for other solid modeler should be very straightforward.

Figure 1: Integration of liaison's GUI with existing menus.

More important than seamlessly extending the interface is providing a capability for displaying liaisons in the geometric context of the assembly. Just as schematic symbols are used in electrical schematic design, we need to display schematic symbols that indicate at a glance the identity and position of liaisons. The problem of providing adequate symbology in the mechanical design context is more difficult than electrical schematics, however, because we are dealing with three dimensional solid objects, so symbols might overlap, be occluded, or be difficult to see due to sizing problems. We defined a collection of symbols that currently seem adequate, and the ability to rescale symbols at any time to provide a means to improve visualization. Further work in this area

will be necessary to provide a production capability. Figure 2 shows the symbols we defined to depict our kinematic liaisons. Figure 3 shows some of those symbols being displayed in a shaded solid assembly rendering. The gear is pressed into the plate to obtain a zero degree of freedom and the pin is pressed into the plate. The symbols indicate a press fit relationship and the direction insertion. Figure 4 illustrates a revolute liaison.

Graphical display requires functionality for drawing graphical elements (e.g. lines), and for maintaining graphics during viewing and modeling operations. Viewing transformations can be accommodated either by adding graphical elements to the modeler's graphical display list (enabling automatic redraw), or drawing the liaisons each time the view is transformed. Redrawing each time the view changed did not incur any significant performance penalty, and was easier than manipulating the display list. Any operations that modify shape require all of the liaisons to be redrawn.

Figure 2: Symbols used to display liaisons graphically.

Figure 3: Graphical display of press fit liaisons. At left, a gear, a plate, and a pin. At right, the parts in an assembly, with liaisons displayed.

Figure 4: A revolute (one degree of freedom) liaison.

Critical to our ability to use liaison information downstream is the ability to export the data. We developed two sets of functionality: a generic file writing/reading capability, and a customized capability for exporting liaisons to DADS (dynamic simulation). The generic capability exports all of the information related to each liaison, and provides the capability to restore that information. Custom interfaces are no more difficult to write, except that each custom interface requires additional code to be written. Both Archimedes and DADS are able to make use of liaison data. In the DADS case, we have not created a sufficiently large variety of liaisons to support a complete analysis, but were able to create a file that correctly represents the representations we support.

Using Liaisons in Novel Design

Current design approaches tend to focus on defining very detailed part geometry early, while delaying analysis as long as possible. We suggest here the possibility of defining liaisons very early in the design process, with very crude models of parts being present. Such an approach provides the possibility of validating performance very early, and of providing geometry to analysis without requiring simplification to be performed.

Consider the assembly in Figure 5, consisting of a bolt-nut, a pair of bearings, and two links which include features such as holes, chamfer, extrusion, fillets and slots. Waiting to perform analysis until all of the geometry is completely defined means that the analysis code has to consider all of the complex geometry, or simplification has to be performed.

Figure 5: Two link mechanism with a revolute joint.

If, instead, liaison information is added very early to an assembly representation, analysis might be provided with the model shown in Figure 6. For kinematic/dynamic analysis, such a model might be completely adequate; for other analyses, the geometry might be inadequate. In any case, the models of geometry and liaisons should grow together during design, to provide models of increasing fidelity to analysis. Growing incrementally can help guarantee that models are adequately simple, without requiring a posteriori simplification (which can be significantly more difficult). Modeling in this fashion (with liaison information available early) might encourage earlier consideration and validation of functionality, before committing too much effort to modeling geometric details.

Figure 6: Two link mechanism in early design process.

In order to use liaisons in this fashion, no new development is required. Skeleton representations of parts have to be created to provide something concrete to relate with liaisons, but these representations only have to be sufficiently correct to support the analysis being performed.

Functional Positional Relationships

Finally, we consider defining functional positional relationships between parts using liaisons. Pro/Engineer currently requires that positional relationships between parts be defined before liaisons can be applied. However, kinematic liaisons contain enough information to define the positional relationships between the parts. Defining part positions in terms of kinematic liaisons suggests that positioning would be truly functional -- that is, the positions of parts would be defined by the function they are expected to perform.

Two possibilities exist for implementing such a scheme: using Pro/Engineer's positioning relationships, or defining an algorithm that infers positioning information, thereby replacing the Pro/E functionality.

Pro/Engineer permits placing of parts relative to each other using a scheme that allows specifying of alignment and mating conditions on part geometry. When a sufficient number of constraints are specified, component placement is complete, and the part's position is completely specified. Such relationships survive regeneration, permitting automatic support of functional relationships. It may be possible to derive these relationships from kinematic liaisons, allowing either manual or automatic definition of a functional set of relationships to be used in part placement. The only currently implementable version of an automatic scheme involves user defined features, which are likely to make implementation difficult, if not impossible. It is possible to lead a user to manually specify the relationships in proper order, but such an approach would be rather unpalatable.

The alternative, an algorithm that derives relative position from kinematic liaison information directly, requires a kinematic solver to determine a set of part positions that simultaneously satisfy the kinematic relationships being specified. The solver would have to be executed at each regeneration to establish correct part placement.

The capability to derive positional relationships from kinematic liaisons has not been implemented, due to the magnitude of the tasks involved.

Testing

Liaison based modeling was tested by applying add-on liaisons to the pin-in-maze discriminator assembly. We were able to model all of the kinematic relationships, and the zero DOF manufacturing information is correct for Archimedes assembly planning. Specification of every relationship in the model is incomplete, due to lack of representations for intermittent contact and stepper-motor inputs. We have also successfully represented the limited rotational degrees of freedom present in the ITAG radar assembly. Liaison data from this assembly was intended for export to virtual reality (Eigen/MuSE) simulation. A demonstration was performed, but automatic data transfer was not completed.

Impact

This project is developing a fundamental manufacturing technology which can be shared with agencies throughout government and industry. It is an early effort in enabling function based design. The work affects any manufacturer of assembled components, and will additionally affect vendors of software used in designing such products. The increasingly small lot sizes we expect to build in the DOE will directly benefit from a more complete specification of the assembly, in terms of more complete archival of design requirements, better understanding of the effects of design changes, and ability to test assembly functionality. This project takes a proactive step in improving the capture and dissemination of design requirements throughout the agile manufacturing enterprise. We are aware of no other efforts to improve enterprise integration at the fundamental assembly specification level. We believe the central efforts in advanced manufacturing technologies will be of this kind.

Future directions

There are a variety of companies developing specialized add-on functionality to the larger CAD systems; such companies deal in products similar to what we are developing, and are much more likely to be interested in cooperative ventures and in marketing our results. As assembly functionality becomes important, liaison based design will become useful tool for CAD users and it will help close the gap between design and analysis. As with other system specific software, the difficulty with current liaison modeling software is that it only works with Pro/Engineer, while many of the larger end-user organizations (aerospace and automotive) currently tend to use other CAD systems. However, as Pro/Engineer becomes more prevalent in large end-user organizations, opportunities for fielding our liaison-based modeling software will increase.

Migration to Other CAD Platforms

Liaison based modeling requires a CAD system to serve as a basis. We have focused on Pro/Engineer to develop the prototype due to convenience in fielding a system to Sandia. If other CAD systems prove viable, liaison based modeling can be ported.

Future Developments

Our current work defines a set of kinematic liaisons which describe functionality in typical assemblies. Future work should focus on extensions such as functional positional relationships, finite element analysis, loads, and experimenting with more advanced liaisons.

Functional positional relationship: We experimented with the Pro/Develop capability to specify positional relationships algorithmically, rather than through a user interface, in preparation for developing software to permit liaisons to control the relative positioning of parts functionally. In other words, automatically defining part-part positional relationships based on the relationships implicit in liaisons and positioning parts in the assembly based on those relationships. This capability would enable users to create assemblies, not by constraining two parts in a positional relationship, but by specify the parts and their functional relationship. Hypothetically, selecting a pin and a part with a hole feature would automatically assemble after specifying the *Press fit* liaison. This capability would prove to be an efficient interface between the CAD system and the designer because less time is spent constraining the part position and more time is spent satisfying the design

intent. The functional positional relationships poses some semantic and numeric problems similar to problems found in developing variational geometry systems, so it may be computationally burdensome.

Finite element analysis: The finite element analysis applications requires that liaisons be associated with the geometric entities in contact, so that the properties of the mesh (mesh continuity between parts, slide planes, etc.) can be correctly modeled during preprocessing. To extend support to this application, we need a face-based liaisons, the ability to promote schematic liaisons to face-based liaisons, and the possibility of defining constraints in a finite element model to support schematic liaisons in analysis.

Load modeling: We have identified the need to model loads and constraints for supporting complete finite element and kinematic/dynamic simulations. Conceptually, loads and constraints should be implementable as modifiers to liaisons and a collection of liaison-like entities themselves. We have the additional consideration that an assembly may be subject to a number of different kinds of loads, therefore loads and constraints must be modeled in sets.

Advanced kinematic liaisons: Intermittent contact and constrained liaisons were identified to be useful for modeling gear-ratchet or cam-follower type of mechanisms. More study is required to determine whether and how such liaisons are supported in our target applications, and what kind of designer's interface would be required. Further work will be needed to understand the possibility of *multiple forms* of representation for any given liaison. For instance, a bolt used to hold two plates together can be thought of a rotational joint. Two bolts holding two plates together is a pair of rotational joints whose effects mutually constrain their rotational degrees of freedom, resulting in a zero DOF joint. Such situations can happen by design or accidentally. For now we will expect users to provide functional design intent, relying on simulation to deal with over- and under-constrained problems.

Conclusion

This report documents our investigation of extending the current solid modeling-based information infrastructure to support design in terms of kinematic relationships between parts, or liaisons. Liaisons capture information regarding contact, degrees-of-freedom constraints, and manufacturing process relationships between parts in an assembly. We tested this modeling paradigm by implementing a prototype system involving extensions to Pro/Engineer and applications such as automatic assembly planning and kinematic/dynamic analysis. Our work focused on identifying requirements of various applications, defining the interrelationships between those requirements, and defining representations and algorithms for supporting the required functionality in Pro/Engineer. Significant effort was expended to ensure that liaison representations would work seamlessly with a parametric regeneration capability and the limitations of Pro/Develop.

We were successful in modeling a variety of liaisons in practical assemblies, and in demonstrating the value of liaison information downstream. Liaisons represent important product information that has never before been captured in CAD representations.

References

1. K. Lee and D. C. Gossard, "A Hierarchical Data Structure for Representing Assemblies," Part 1. J. of CAD, V 17 (1), 1985.
2. K. Lee and G. Andrews, "Inference of the Positions of Components in an Assembly," Part 2, J. of CAD, V 17 (1), 1985.
3. D. Rocheleau and K. Lee, "System for Interactive Assembly Modelling," J. of CAD, V 19 (2), 1987.
4. M. A. Wesley, T. Lozano-Perez, et. al, "A Geometric Modeling System for Automated Mechanical Assembly," IBM J. Res. Develop., V 24 (1), 1980.
5. L. S. Homem de Mello and S. Lee, eds., "Computer-Aided Mechanical Assembly Planning," Kluwer Academic Publishers, Boston, 1991.
6. M. Henderson, Arizona State University, private communication.
7. S. G. Kaufman, R. H. Wilson, et. al, "LDRD Final Report: Automated Planning and Programming of Assembly of Fully 3D Mechanisms," Sandia Report, SAND96-0433 UC-705, 1996.
8. DADS Dynamic Analysis and Design System, CADSi Computer Aided Design Software, Inc. 1993.
9. Pro/ENGINEER Assembly Modeling, Parametric Technology Corporation. 1996.
10. Pro/DEVELOP, Parametric Technology Corporation. 1996.

Distribution

1	MS 0151	G. Yonas, 9000	
1	MS 1002	P. J. Eicker, 9600	
1	MS 1003	R. W. Harrigan, 9602	
1	MS 1003	R. D. Robinett, 9611	
1	MS 1008	D. R. Strip, 9621	
1	MS 1010	M. E. Olson, 9622	
1	MS 1004	R. D. Palmquist, 9651	
1	MS 1006	P. Garcia, 9671	
1	MS 1007	A. T. Jones, 9672	
10	MS 1010	D. K. Kholwadwala, 9622	
1	MS 1010	A. L. Ames, 9622	
1	MS 0501	R. M. Bugos, 2334	
1	MS 0329	K. Eras, 2643	
1	MS 1008	R. C. Brost, 9621	
1	MS 1008	R. H. Wilson, 9621	
1	MS 0188	LDRD Office, 4523	
1	MS 9018	Central Technical Files, 8523-2	
5	MS 0899	Technical Library, 4414	
2	MS 0619	Review & Approval Desk, 12630	For DOE / OSTI
3	MS 0161	Patent and Licensing Office, 11500	