The Open Assembly Design Environment Project: An Architecture for Design Agent Interoperability

Kevin Lyons
National Institute of Standards and Technology
100 Bureau Dr, Stop 8260
Gaithersburg, MD 20899–8260
Kevin.lyons@nist.gov

Steven Shooter
Bucknell University
Lewisburg, PA 17837
shooter@bucknell.edu
(NIST Guest Researcher)

Walid Keirouz
Lebanese American University
475 Riverside Dr., #1846
New York, NY 10115
walid@lau.edu.lb
(NIST Guest Researcher)

Peter Hart
National Institute of Standards and Technology
100 Bureau Dr, Stop 8262
Gaithersburg, MD 20899–8262
hart@nist.gov

Abstract

The Open Assembly Design Environment (OpenADE) project is an initiative at the National Institute of Standards and Technology (NIST) to provide an integrated and augmented CAD environment for assembly design. The goals of the project are: (1) to identify representations and issues for the next generation of assembly-related standards and (2) assist designers with assembly considerations throughout the phases of a product's design—from conception to final process plan development. OpenADE's open architecture provides standard interfaces that allow it to link to commercial and non-commercial design tools: parametric design systems, virtual reality environments, assembly analysis tools, and assembly process planners. The OpenADE project has explored issues relating to knowledge representations, virtual reality, assembly-level tolerances, constraint-based specifications, and assembly process management. This article describes the OpenADE architecture and the components that have already been implemented. It also describes plans for extending OpenADE's assembly knowledge representations and handling of geometric and kinematic constraints.

Introduction

The emergence of high performance desktop computing has opened new avenues for improving the product realization process. Computationally intensive design tools and analysis models that were previously considered too unwieldy and complex to be practical are now being reexamined as viable candidates. Furthermore, network-based communication allows a geographically distributed design team to share tools and data to coordinate a design effort. However, in order for the coordination effort to succeed, the design team and the tools used by the team must use the same semantics to describe a design. The shared semantics derive from the set of design problems of interest, namely assembly-oriented design. Furthermore, the design information must reside in a shared workspace accessible to design team members and the design tools they use.

Research into the product realization process has resulted in the development of various process models described as Design-For-X (DFX). Among these DFX models are Design-For-Manufacturing, Design-For-Robustness, and Design-For-Assembly (DFA). Some DFX models have been implemented into tools that analyze existing designs for possible improvements within the Design-For-X framework (Boothroyd, *et al.*, 1994). The impact of these tools has been substantial. However, this impact has been limited to the final stages of the design process. As such, these tools must be extended to assist designers in the early stages of design.

The Open Assembly Design Environment (OpenADE) project seeks to capitalize on advances in communication and computational technologies to expand the DFA process model to encompass the complete product realization process. The overall goals of the project are:

- Develop an assembly-oriented representation that supports collaborative design and uses standardized interfaces to integrate assembly-oriented tools with existing and emerging CAD applications.
- Identify and address issues that could accelerate the use of assembly-oriented tools in early stages of design.
- Identify assembly representations and integration issues that aid in standards development by the design community.
- Illustrate a solution to the identified issues by developing a demonstration virtual assembly tool that

augments existing CAD systems with a virtual reality subsystem.

This article summarizes the results achieved within the OpenADE project, provides a current status of the OpenADE system itself, and describes proposed extensions. The article is organized by presenting related research, the OpenADE architecture, the assembly design process manager, assembly-level tolerancing, assembly mating constraints and component kinematics, immersive CAD, and future extensions.

Related Research

OpenADE's architecture attempts to unify representations for designed artifacts, assemblies, and geometric and kinematic constraints. It also attempts to address standards-related issues. This section briefly surveys some of the related research in each of these areas.

Design Representations

Three projects, SHARED (Gorti and Sriram, 1996; Gorti *et al.*, 1998), Design Repository (Szykman, 1998), and the Integrated Product Development Environment—IPDE (Liang *et al.*, 1997; Qureshi *et al.*, 1997) have similar goals and scope to OpenADE. SHARED is an object-oriented framework for conceptual design whose representation goals are very similar to those of –OpenADE. It can represent a process model, a designed artifact, and its design history. SHARED models a top-down design process using *goal*, *specification*, *plan* and *decision* objects. It has a multifaceted representation of artifacts. At the functional level, SHARED represents the form, function and behavior of an artifact. It explicitly models inter-object relations using *relation* and *constraint* objects. This allows the framework to support reasoning about function-to-form mapping. The goals of SHARED and OpenADE are similar concerning conceptual design. However, SHARED did not address issues related to standards-based data exchanges, whereas this is a major concern of OpenADE.

The "Design Repository" project focuses on the functional decomposition of existing designs and uses a representation that is similar to SHARED. It has an ontology for modeling functional systems and defines semantics for reasoning about them. It also models an assembly as part-whole hierarchy, but does not represent the mating constraints between an assembly's components. As will be discussed later, one of the main extensions of the OpenADE architecture is to support the "Design Repository" project.

The IPDE project differs from OpenADE in that it deals with design information on a large scale and focuses on detailed design. It demonstrates how a variety of CAD tools, including a product data management system, can be integrated using STEP-based tools. IPDE uses an extension to STEP to keep track of design versions within the context of product data management. However, it does not explicitly represent functional systems and cannot trace the evolution of individual components.

Additional systems relate to parts of OpenADE or do not have a scope similar to that of OpenADE. For example, the Design-For -Tolerance architecture outlined by Narahari *et al.* (1998) does not specify a given representation for design. However, it requires tolerance-related data to be represented in all stages of the design of an assembly. It also requires the tracking of tolerancing decisions throughout the design stages. These requirements are exactly those that OpenADE is trying to address.

The "Issue-Base Information System" (IBIS) project (Conklin *et al.*, 198x) developed representations for reasoning about design rationales. IBIS modeled rationale informally as consisting of a set of related "issues," "proposals," "arguments," and "decisions." An IBIS-like representation can be easily integrated into a larger design representation to keep track of design rationales.

Assembly Representations

Many systems have proposed representations of assemblies. Of particular interest is the one used by HyperGEM (Pabon *et al.*, 1992). HyperGEM used a knowledge-based representation to integrate defined features recursively as collections of features, geometric elements, and constraints (algebraic and geometric). Furthermore, a feature can exports components (attributes, parameters, geometric elements, or dimensions of geometric elements) to its parent feature so these components can be used in constraints that apply within the parent feature. HyperGEM uses constraint solution algorithms to maintain the internal and external consistency of features. HyperGEM did not have an explicit representation of assemblies beyond that of features. However, its flexible representation of features allowed it to decompose part-whole hierarchies and mating constraints.

Some systems have used graph-based representations that can be easily used for assembly analysis. Whitney (1996) proposed using "liaison diagrams" and "key characteristics" to represent assemblies because they make tolerance analysis easier. Other systems have combined graphs with geometric constraints and degrees of freedom for assembly planning and reasoning about the possible motions within an assembly

OpenADE Architecture

A skeleton view of the OpenADE architecture is shown in Figure 1. The architecture has three major components: an *integrated design database*, *agents*—independently developed and autonomous applications—acting on the database, and *data translators* that allow agents to exchange data with the database and with each other.

1. The integrated design database implements the shared design workspace. This design database acts as a gateway for accessing all design related information. The database may be centralized to store all of the design information or it may be distributed such that design information is stored in multiple locations. The distributed locations include site of the applications that generated the information

2. The integrated design database is viewed as containing all the data relating to a design. This information includes data used or generated in early stages of the design such as customer requirements and system specifications. It also includes data generated during detailed design such as detailed part geometry, assembly planning sequences, and the results of various analysis programs. Additionally, the database includes a process model to control the design process, design histories to track changes in a design, and rationale to justify decisions and changes. All data is organized using an *integrated schema* that encodes the semantic model for the domain under consideration—assembly design.

3. Agents are programs that access the integrated design database to retrieve design data or to store generated information. Agents vary in nature. They can be traditional applications that act in specific domains (*e.g.*, FEA programs) or general-purpose CAD systems that provide functionality during detailed design (*e.g.*, part geometry and bill of materials). They can also be novel applications such as a process manager for controlling a design process, a design browser for navigating in a design repository—reviewing past designs and their rationales, or a knowledge-based tool that assists a designer in going from a system specification to functional and behavioral models.

   Each agent is concerned only with its view of the integrated schema. The integrated database makes sure that the requested data is available and that data to be stored is consistent with existing data.

4. Data Translators coordinate the exchange of data among the agents and the database. These translators are based on existing standards whenever possible (*e.g.*, ISO FDIS 10303-47). When such standards are not available, the translators will be formalized to provide input into the standards development effort.

Figure 1. Architecture of OpenADE

In the current version of OpenADE, the shared workspace is implemented as a *distributed* database and agent interaction with the database goes through a *database system manager*. This manager provides mechanisms to store and retrieve data from potentially different locations, query data, define relationships between data, maintain metafiles, maintain version control, and performing other pertinent services. When an agent requires information from the database, the system manager queries the database for the needed information. If the information is not available, the manager invokes an agent that can provide or compute the needed information. The information is then retrieved from the agent, stored in the database, and passed back to the agent that made the initial request.

The database system manager decides which agent to invoke by consulting a table of data exchange translators. One translator is written per agent and is registered with the manager. The translator specifies the data an agent and the OpenADE database can provide to each other, the form of this data, and routines for translating the data between the two forms.

Current standardized representations of assembly information do not contain all the data needed in assembly-related design applications. The missing data includes assembly level tolerances, assembly mating constraints, and component kinematics information. As such, the current version of OpenADE uses proprietary formats for storing this data.

Assembly Design Process Manager

The Design Process Management System (Kim and Szykman, 1997) is an agent that allows the designer to manage the overall design process and explore the assembly design space by explicitly representing design stages, constraints, and rationale. The agent uses a shared design space to support collaborative design. It also allows a designer to access the design data from a user-defined perspective. This approach differs from the approach of traditional design management systems where all aspects of the design cycle are tightly controlled.

Acting as an independent agent, the Design Process Management System integrates a geometric modeling system, a design process manager, a Design-For-Assembly tool, and an optimization tool. Figure 2 shows an overview of the DPM architecture. The central component is the Design Process Manager (DPM) which allows the user to specify a generic design process model and then design within that model in a structured manner. DPM provides a graphical view of design evolution and alternatives using a tree-like representation where each node corresponds to a design step and each level corresponds to a particular assembly design stage. DPM is integrated with several other design and analysis tools as shown in Figure 2.

Title:
dpm-arch.eps
Creator:
fig2dev Version 3.2 Patchlevel 0-beta3
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 2. Design Process Management Architecture

As a part of OpenADE, the DPM provides its capabilities at two conceptual layers: a Conceptual Assembly Modeling Framework and a Design Refinement Tool (Kim and Szykman, 1997). The Conceptual Assembly Modeling Framework (CAMF) allows a designer to create and maintain evolving assembly designs through top-down and bottom-up assembly modeling. It incorporates assembly-related tools such as analysis programs, design case bases, and a materials library. CAMF does not prescribe the design approach, but allows a designer to move between various representation levels. These levels represent different design views, design stages, and levels of abstraction. CAMF allows a designer to explore a design space. However, the designer is responsible for keeping track of alternative designs.

The Design Refinement Tool (DRT) builds on CAMF's capabilities and allows a designer to: (1) rapidly generate design alternatives, (2) augment partial designs by selecting values for design variables, (3) refine existing designs, and (4) generate a feasible design starting with an infeasible design. DRT can generate design alternatives subject to design constraints on certain attribute types. Refining the alternative designs is driven by an optimization algorithm based on Simulated Annealing (Kirkpatrick *et al.*, 1983).

Assembly-Level Tolerancing

Tolerancing is a critical issue in the design of electromechanical assemblies (Sudarsan *et al.*, 1997; Narahari *et al.*, 1998). Design-For-Tolerancing (DFT) is an approach to consider tolerancing issues as an integral part of a design process. Existing approaches to design tolerancing, such as Motorola's Six Sigma program (Harry and Stuart, 1988) and Taguchi Methods (Kacker, 1988), have had a substantial impact on industry. However, these approaches typically require detailed knowledge of the geometry of the assemblies and are therefore only applicable during detailed design. As such, these systems encourage design iteration and result in added development time. By contrast, the assembly structure and the associated tolerance information evolve continuously throughout the design process. Therefore, significant gains can be achieved by effectively using the produced information, as it becomes available instead of during detailed design only.

The DFT effort within OpenADE has two goals: (1) advance tolerancing decisions to the earliest possible stages of design, and (2) address the appropriate use of available methods and best practices for tolerance synthesis and analysis at successive stages of design. Achieving these goals requires an effective representation of tolerance information.

Research on an effective representation that supports tolerancing considerations throughout the design process led to the architecture presented in Narahari *et al.*(1998). This multi-level approach integrates the design process, the assembly models for tolerancing, and the tolerancing methods, and best practices. The model includes a four-level approach to design tolerancing affecting all design stages. The four tolerance-related levels (TR level) are as follows:

<u>TR Level 1: Assembly Layout and Configuration</u>

Decisions in TR Level 1 include rough allocation of space, number of subassemblies, grouping of components into subassemblies, and rough layout of the assembly. To affect these high-level tolerance decisions, aggregate-level manufacturing process capability data is required and is often available. This level also includes the use of simple statistical assumptions and probabilistic calculations.

TR Level 2: Location Logic and Assembly Features

This level advances from the availability of assembly response function (approximate), tolerance requirements at part and subassembly interfaces, and relevant process capability data. This level includes liaison diagrams, feature determination, and location logic in the data flow chain.

TR Level 3: Assembly Planning and Sequencing

Level 3 considers that the following are known: the detailed assembly response function, detailed process capability data, skeletal geometry of the assembly, assembly features, and specifications of parametric or geometric tolerances of individual parts and features. The decisions focus on the selection of the detailed assembly sequence that achieves a superior set of tolerance specifications.

TR Level 4: Detailed Tolerance Analysis and Synthesis.

Level 4 builds on a known assembly sequence; geometric data on parts and features; detailed part-level tolerance requirements; and a complete assembly-response function. Most available tolerancing studies and tools support this level of design.

The OpenADE Architecture of Figure 1 shows a Tolerance Tool acting as an agent and a tolerance model within the design database. The Tolerance Tool is an automated *Design for Tolerance Environment* under development for exploration of enhancements to the current standard ISO 1030-203. This agent will support the tolerance model within the design database. The tolerance model within the design database will be available to be shared with other agents.

Assembly Mating Constraints and Component Kinematics

Commercial CAD systems interpret assembly modeling to allow a designer to easily position components (location and orientation) with respect to each other within an assembly (Lyons *et al.*, 1997). These systems use geometry-based mating constraints (mate/against, align, insert/fit, orient, *etc*.) to relate components to each other. Furthermore, the kinematics modules within these CAD systems typically require the designer to specify different types of joints (revolute, prismatic, ball, *etc*.) between the assembly's components. As a result, conflict may arise between the surface mating constraints and the kinematic joint definitions because there is no direct link between the two representations.

OpenADE implements an algorithm that determines Component Degrees Of Freedom (DOFs) from the assembly surface mating constraints (Rajan *et al.*, 1997a; Rajan *et al*., 1997b). This algorithm is used to propagate and verify the kinematic design intent as the design progresses through each design stage and across different tools.

The algorithm identifies the set of unconstrained degrees of freedom by considering the various types of mating constraints. It enumerates parts that can separate to determine the component degrees of freedom. OpenADE uses an icon-based representation to display the component degrees of freedom to the designer. This visual feedback helps the designer identify under- and over-constrained components. Furthermore, this approach avoids redundant and potentially conflicting specification of kinematic joint constraints because it automatically computes this information from the assembly mating constraints.

A prototype implementation incorporating basic mating constraints ("*against*" and "*fits*") has been developed. The application interfaces with SDRC's IDEAS system using the IDEAS Open Architecture Open Link Application Programming Interface. Figure 3 shows a four-part assembly taken from a circular saw: "*rear-cage*", "*arm*", "*pin*", and "*arm-spring-cage*". OpenADE updates the component degrees of freedom for each part and its associated icons as the assembly process progresses. As indicated in the figure, "*rear-cage*" and "*pin*" have six degrees of freedom. By contrast, "*arm*" and "*arm-spring-cage*" have been assembled and only have rotational DOFs. The current prototype is a proof of concept that is limited to interpreting "*against*" and "*fits*" constraints associated with planar and cylindrical contacts. The algorithm itself can be enhanced to include other types of surface mating constraints.

The algorithm and the prototype implementation does not act as an independent agent. The algorithm is intended as part of the database to be invoked by the designer to display additional information within the CAD tool.

The assembly representation used in this work is closely related to the kinematics representation given in ISO 10303-105 with additional enhancements to include rigid attachments and excluding detailed component and mating geometry. Mating direction sets, joint attributes, and a relational structure embedded within the hierarchical assembly structure are explicitly represented to facilitate assembly analysis and planning. These representations allow the exchange of design intent as well as assembly constraint information between modeling, analysis and planning systems (Lyons *et al.*, 1997; Rajan *et al.*, 1997).

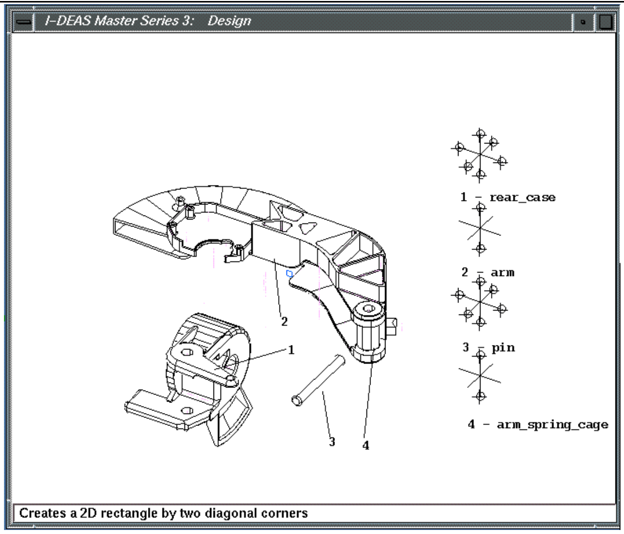

Figure 3. Components & Associated Icons Showing C-DOFs

Immersive CAD

The Virtual Assembly Design Environment—VADE (Chandrama, 1997; Connacher, 1997) acts as an immersive CAD agent for OpenADE. Washington State University is developing VADE under a grant from NIST to investigate the use of virtual reality for assembly design. VADE combines advanced CAD/CAM software with the latest in virtual reality technology to produce an environment that allows engineers to virtually assemble a series of components. This produces an augmented CAD system that provides the user with a fully immersive three-dimensional environment for assembly design and analysis.

VADE presents the user with a simulated environment to perform an assembly, as shown in Figure 4. The user has in front of him/her all the parts of an assembly. These parts are ordered according to the assembly sequence used in the CAD system. VADE stores the location and orientation of the final parts in the assembly. It also stores the constraints used to assemble the system, and the mass and material properties of the components. The first part to be assembled, known as the base part, is attached to a virtual left hand.

The user assembles a component by grasping it and "*properly*" positioning it. VADE monitors the component's position through the proper satisfaction of the constraints that were used in the assembly procedures of the CAD system. Once a component satisfies all its assembly constraints, the user releases the component and it becomes part of the assembly. The user can then grasp another component and assemble it, or remove components already assembled. This procedure is repeated until the assembly is complete.

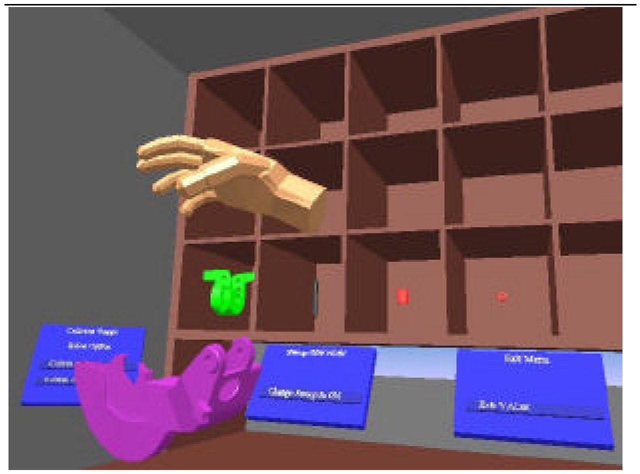

Figure 4. VADE Simulated Environment and Components

VADE can visualize the CAD system's assembly constraints in the virtual environment. This capability assists the user in properly placing the component. VADE can also capture the movement of the part being assembled as it moves through space. Once captured, this volume in space, known as the swept volume, can be visualized and analyzed for space requirements.

By assembling the parts in a virtual environment, the user can detect problems unforeseen when using traditional CAD applications only. These problems include a blockage in an assembly path, a sequencing of parts that was not predetermined, accessibility problems, and component interference.

VADE retrieves several pieces of information from OpenADE's design database. Some of this information is independent of the assembly process and is used to display the assembly's components: faceted representations, colors, normals, and other material properties. VADE also uses information generated during the creation of the CAD assembly: the placement of all of the parts, the constraints used in placing each component, the mass and material properties of each component, and the sequencing of the parts for the assembly. While VADE is linked to OpenADE, some of this data, such as sequencing information, could be retrieved from sources other than the CAD system.

VADE itself generates data that must be transferred and stored in OpenADE's database: swept volumes, interference information, and design modifications recommended by the user. The latter include sequencing information and geometric changes.

Future Extensions

OpenADE is currently being extended to provide a comprehensive infrastructure that will support information exchange among a large number of design agents operating throughout the stages of assembly design. These extensions are described below.

Assembly Knowledge Representations

Current design tools require sophisticated users who understand the intricacies of a particular design domain in order to obtain the benefits promised by the use of design tools. However, this requirement is not realistic as the design of complex systems is interdisciplinary in nature and designers do not possess adequate background knowledge across the many domains needed to produce effective designs. As a result, the potential benefits of design tools remain unrealized and the opportunity to produce superior designs is missed. For design tools to benefit the broadest audience of designers operating in an integrated design environment, the tools must account for inexperienced designers who do not possess the background and expertise in the application's particular domain(s). These users require assistance in the capabilities of the software and in the background of the domain itself. This is especially true in the area of design for assembly (DFA),

The extensions to OpenADE will build on the current version to (1) explore techniques to support the designer throughout all stages of the design process, (2) support information exchange across a broad range of design agents, and (3) broaden the audience of designers who will use DFA practices.

The approach we will follow focuses on the following aspects:

- Identify domain-specific knowledge and methods needed to develop representations that can be applied across multiple domains.
- Ensure that these representations are open and allow the sharing of key information among disparate assembly applications.
- Minimize the additional load placed on the designers by the new functionality. Designers may find the additional burden of using these tools acceptable only if the benefits derived from using these tools is quite substantial.

Constraint Representations

Our approach is to build a framework—domain-specific representations and techniques—that will provide applications with assembly-related functionality.

- The new representation of assemblies will be able to handle all the information generated and used in OpenADE and will support the evolution of an artifact's description as a design progresses. It will also be compatible with representations used in companion projects at NIST such as the *Design Repositories* project (Szykman, 1998).
- Constraints play a critical role in encoding various design relationships (geometric, functional, *etc.*). The new representation will formalize the representation and processing of constraints, and will allow constraints to evolve as an artifact's design evolves.

Kinematic Relations

The new representations will be used to further develop OpenADE's capabilities in kinematic relations. The kinematics-related work will build on the work by Rajan *et al.* for verifying and propagating kinematic constraints. It will explore methods to evolve the kinematic joint definitions during conceptual design into full-fledged geometric specifications during detailed design.

Assembly-Level Tolerancing

The design for tolerance framework will be expanded to include the early stages of product development and will include considerations of tolerance synthesis and tolerance analysis. The assembly model for tolerance will be closely coupled to the design process and will represent the assembly and tolerance information at multiple levels of abstraction. Other attributes identified for the model are the effective capture of design intent and the embedding of different views. The comprehensive model will support the mapping of functions to tolerances with a formal procedure rather than ad-hoc experience-based methodologies.

Conclusion

The current version of OpenADE has demonstrated information exchange between a small number of design agents providing different functions in an integrated assembly design environment: process management, tolerancing, mating constraints and component kinematics, and virtual reality environment. However, these tools have operated in the latter stages of a design.

OpenADE is currently being extended to provide a comprehensive infrastructure that will support information exchange between a large number of design agents operating throughout the stages of assembly design. The extensions include assembly knowledge representations, constraint representations, kinematic relations, and system level tolerancing. The representations developed will act as prototypes for new standards in assembly design.

Acknowledgements

The authors acknowledge the following researchers for their contributions to the OpenADE project: S. Angster, M.R. Duffey, G.J. Kim, Y. Narahari, V. Rajan, S. Sreerangam, R. Sriram, R. Sudarsan, and S. Szykman.

The OpenADE project is supported by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA).

References

OpenADE

Connacher, H., S. Jayaram and K. Lyons (1995), "Virtual Assembly Design Environment," *Proceedings of the 15th ASME International Computers in Engineering Conference*, Boston, MA, September.

Connacher, H., S. Jayaram and K. Lyons (1996), "Integration of Virtual Assembly with CAD," *Symposium on Virtual Reality in Manufacturing Research and Education*, Chicago, October.

Connacher, H. S. Jayaram and K. Lyons (1997), "Virtual Assembly Using Virtual Reality Techniques, accepted for publication in Computer Aided Design.

Kim, G. J. and S. Szykman (1997?), "Combining Interactive Exploration and Optimization for Assembly Design," ASME Journal of Mechanical Design,

Lyons, K., S. Angster, P. Hart, Y. Narahari, V. Rajan, and R. Sudarsan (1998), "Open Assembly Design Environment," Technical Report, National Institute of Standards and Technology.

Lyons, K.W., V.N. Rajan and R. Sreerangam (1997), "Representations and Methodologies for Assembly Modeling," Technical Report, National Institute of Standards and Technology, August.

Narahari, Y., R. Sudarsan, K.W. Lyons, M.R. Duffy and R.D. Sriram (1998), "Design for Tolerance of Electro-Mechanical Assemblies: An Integrated Approach," Technical Report, National Institute of Standards and Technology, September.

Rajan, V.N., K.W. Lyons and R. Sreerangam (1997), "Generation of Component Degrees of Freedom from Assembly Surface Mating Constraints," *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Sacramento, California, September, DETC97/DTM-3894.

Rajan, V.N., K.W. Lyons and R. Sreerangam (1997), "Assembly Representations for Capturing Mating Constraints and Component Kinematics," *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, Marina del Ray, California, August pp. 206-211.

Sudarsan, R., K. Lyons, and M. Pratt (Editors) (1997), Assembly Level Tolerancing: Standards and Implementation Issues, Workshop Handout, National Institute of Standards and Technology.

Supporting References

Boothroyd, G., P. Dewhurst, and W. Knight (1994), Product Design for Manufacture and Assembly, Marcel Dekker, Inc., New York.

Chandrana, H. (1997), *Assembly Path Planning Using Virtual Reality Techniques*, M.S. Thesis, Washington State University, Pullman, WA, May.

(Conklin *et al.*, 198x)

Connacher, H. I. (1996), *A Virtual Assembly Design Environment*, M.S. Thesis, Washington State University, Pullman, WA, May.

Gorti and Sriram, (1996), "From Symbol to Form: A Framework for Conceptual Design," Computer-Aided Design, Vol 28, No 11, pp. 853-870.

Gorti, S.R., A. Gupta, G.J. Kim, R.D. Sriram, and A. Wong (1998), "An Object-Oriented Representation for Product and Design Processes," *Computer-Aided Design*, Vol. 30, No. 7, pp. 489-501.

Harry, M.J. and R. Stewart (1988), "Six Sigma Mechanical Design Tolerancing," Technical Report, Government Electronics Group, Motorola, Inc., Scottsdale, Arizona.

ISO 10303-47, *Product Data Representation and Exchange, Part 47: Integrated Generic Resources: Tolerances*, ISO TC184/SC4.

ISO 10303-105, *Product Data Representation and Exchange, Part 105: Application Resource: Kinematics*, ISO TC184/SC4, 1994.

ISO 10303-203, *Product Data Representation and Exchange, Part 203: Application Protocol: Configuration Controlled Design*, ISO TC184/SC4, 1994.

Kacker, R.N. (1988), **Handbook of Statistical Methods for Engineers**, McGraw Hill, pp. 19.1-19.27.

Kirkpatrick, S., C. Gelatt, and M. Vecchi (1983), "Optimization by Simulated Annealing," *Science*, **220**(4598):671-679.

Liang, J., J. Shah, S. Urgan, E. Harter, and T. Bluhm (1997), "STEP Pilot Project: Consolidate Data Schema for Airframe Engineering", *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Sacramento, California, September.

Qureshi, S., J. Shah, S. Urban, E. Harter, C. Parazzdi, and T. Bluhm (1997), "Integrated Model to Support Archival of Design History in Databases," *Proceedings of the 1997*

ASME Design Engineering Technical Conferences, Sacramento, California, September.

Szykman, S., J.W. Racz, C. Bochenek, and R.D. Sriram (1998), "Web-based System for Design Artifact Modeling," *Design Studies*, (accepted for publication)

Whitney, D. E. (1996), "The Potential for Assembly Modeling in Product Development and Manufacturing," http://web.mit.edu/ctpid/www/whitney/papers.html.