NISTIR 6111

Proceedings of the 1997 Knowledge-based Systems Interoperability Workshop

3 & 4 November 1997
Gaithersburg, Maryland

Robert H. Allen, Ram D. Sriram (editors)

U.S. Department Of Commerce
Technology Administration
National Institute of Standards and Technology
Manufacturing Engineering Laboratory
Manufacturing Systems Integration Division
Engineering Design Technologies Group
Gaithersburg, MD 20899

NIST

NISTIR 6111

Proceedings of the 1997
Knowledge-based Systems Interoperability Workshop

3 & 4 November 1997
Gaithersburg, Maryland

Robert H. Allen, Ram D. Sriram (editors)

U.S. DEPARTMENT OF COMMERCE
William M. Daley, *Secretary*
Technology Administration
Gary Bachula, *Acting Under Secretary for Technology*
National Institute of Standards and Technology
Raymond Kammer, *Director*

January 1998

Disclaimer

These proceedings are a summary of the NIST-sponsored workshop on Knowledge-based Systems Interoperability: Standards and Implementation Issues, which was held on 3 and 4 November 1997. Because participants included software users and representatives from commercial vendors, certain products are identified in this report to present specific views and to facilitate understanding of concepts and implementations. The National Institute of Standards and Technology does not judge, recommend or endorse these products. The opinions expressed in this report are those of the workshop participants and not necessarily those of NIST or its employees.

Table of Contents

Introduction

On 3 and 4 November 1997, the National Institute of Standards and Technology (NIST) sponsored[1] an information-gathering workshop that focused on "Knowledge-based Systems Interoperability." Held in Gaithersburg, MD on the NIST campus, and in response to the growing need for integrating knowledge in distributed computing environments, the workshop addressed the general issue of interoperability among knowledge-based systems[2] especially in engineering design and manufacture. The workshop, which had more than thirty participants, included seven presentations from developers, vendors and users, group discussions on knowledge-based system interoperability -- its present capabilities and some of its main drawbacks -- and a general session to target specific research and development, and end-user needs. This report documents the workshop background and goals, its participants and agenda, the speakers' abstracts and slides, and provides a summary of the workshop results.

Workshop Background & Goals

The purpose of this workshop was to bring together knowledge-based system (KBS) developers, vendors and users from different engineering disciplines to discuss matters of common interest concerning software interoperability. Functional interoperability is fundamental to the success of complex engineering processes such as collaborative design. Although much effort has been put forth in standardizing geometric product data exchange with the development of the international STandard for the Exchange of Product model data, STEP, ISO 10303 [ISO94], such standards do not yet address the exchange of parametric data such as design rationale, functional specification and design intent. To achieve functional interoperability, computer-aided engineering (CAE) applications in general, and KBS in particular, need to be implemented in such a way that the exchange of data and knowledge can occur without loss of information, tolerance or robustness. How to bring about this interoperation is precisely the reason for this workshop.

The workshop mission was to provide an open forum for KBS vendors, engineers and manufacturers, to discuss the state-of-the-art, identify gaps in current technology, and to begin proposing solutions to close those gaps.

Specific workshop goals include the following:

- to provide an overview of the state-of-the-art in KBS interoperability issues in industry, government and academia,
- to present industry case studies on current practices in KBS interoperability,
- to draw roadmaps that will aid in research and development in KBS interoperability, especially in collaborative engineering projects, and
- to identify interoperability standards and technology issues.

The workshop was organized as a series of presentations from speakers representing KBS developers, KBS researchers, and engineers who use KB and CAE systems in their design and manufacturing activities (two developers, three researchers and two engineers). NIST personnel provided additional input on the state of comparable standards and government activity. Following the morning of presentations, workshop organizers split the participants into two subgroups. Each subgroup brainstormed on one of these two themes:

I- State of the Art on KBS Interoperability

---

[1] Specifically, this workshop was sponsored by the Engineering Design Technologies (EDT) Group, a part of the Manufacturing Systems Integration Division (MSID), under the auspices of the Defense Advanced Research Projects Agency's (DARPA's) Rapid Design Exploration and Optimization (RaDEO) program.

[2] A KB system, also known as an expert system, is software that has some knowledge or expertise about a specific, narrow domain, and is implemented such that the KB and the control architecture are separated. KB systems have capabilities that often include inferential processing (as opposed to algorithmic processing), explaining rationale to users and generating non-unique results [Mah87].

II- Barriers and Requirements for KBS Interoperability

The subgroups reconvened to discuss the issues raised, and report on each subgroup's findings to the entire workshop. The second day was used for a general discussion, refinement of our findings, and for the group to agree on a list of action items to be taken.

Workshop Results

Of the more than a dozen issues identified by the groups (and listed below), two main themes emerged:

1. Interoperability among KB and CAE systems is a major bottleneck today.

2. Current standards do not address many of the interoperability issues associated with KBS.

Within these main themes, five concepts emerged as priority issues. These are:

Characterization	There is strong need to characterize - perhaps even standardize - the capabilities, behavior and underlying philosophy of KB systems.
Usability	Engineers and manufacturers who use KB and CAx systems must not be unduly burdened with interoperability issues.
Vocabulary	For design and manufacturing applications, a core set of primitives (such as artifact, design plan, goal, form, function and behavior) need to be understood and represented in a standardized way so that meaningful exchange of such knowledge can be achieved.
Collaboration	The commercial, academic and governmental communities must collaborate to address the interoperability issues in a most meaningful way.
Cost	The cost of KB systems and their interoperability must be manageable for midsize companies.

Participants also identified 14 issues as being important in KBS interoperability. These are listed below:

1. Knowledge representation (KR) is the critical element for interoperability because if different KR schemes need to interact, there must be some commonality among representations. One possible solution is to link different KR schemes by using the Knowledge Interchange Format, KIF [Gen92], with a formal explicit specification of a conceptualization, often referred to as a frame ontology [McG93].

2. Mediation is important for interoperability because it places context on a specific knowledge base, otherwise known as semantic heterogeneity.

3. Problem solving cooperation is necessary to limit the amount of knowledge sharing in specific interoperable transactions.

4. Knowledge base validation is important for interoperability because of the consistency issue associated with individual KBs, and the ramifications for downstream propagation of possible misinformation.

5. Negotiation is an important attribute in interoperable KB systems because of the nature of most engineering design and manufacture activities.

6. Knowledge base comprehension is important for global context. To efficiently interoperate, KB systems require agents that describe the knowledge a specific KB contains, thereby streamlining search.

7. Knowledge capture is clearly achievable for specific domains, yet this activity remains a bottleneck.

8. Knowledge history, or meta-knowledge, is important to trace the reason for a particular conclusion or action.

9. Knowledge types must be varied for interoperability to be effective. Many types of objects should be recognizable - business objects, design objects, management objects and manufacturing objects.

10. KIF was developed as an interchange format and may prove very useful as a building block in representing knowledge across different KR schemes.

11. Design rationale is one level of knowledge that must be made interoperable.

12. Common Object Broker Request Architecture, or CORBA [OMG96], compliance is important for communication across different platforms and applications implemented in different languages.

13. Java™ [Cam96] compliance may be important for distributing knowledge across networks.

14. Problem solving method libraries are important so that meta-knowledge can be used to locate appropriate knowledge sources.

Action Items

The workshop concluded with a set of five action items that participants agreed to address. These are:

1. Begin surveying KBS developers and characterizing existing tools.

2. Develop sample practical problem involving multiple KB and CAx systems.

3. Define a taxonomy of domain entities, or primitives, that lend themselves specifically to interoperability in design and manufacture.

4. Explore the similarities and differences between KIF and the STEP data modeling language, EXPRESS, and its extensions.

5. Draft position paper on KBS interoperability discussing goals, challenges, strategies and areas of application.

References

[Cam96] Campione, M. and K. Walrath, "The Java Tutorial: Object-Oriented Programming for the Internet (Java Series)," Addison-Wesley, 1996.

[Gen92] Genesereth, M. and R. Fikes, "Knowledge Interchange Format, Version 3.0 Reference Manual," Computer Science Department, Stanford University, Tech Report Logic 92-1.

[ISO94] "STEP Part 1: Overview and Fundamental Principles," ISO TC184/SC4, November 1994.

[Mah87] Maher, M.L. and R.H. Allen, "Expert System Components," In: *Expert Systems for Civil Engineers: Technology and Application* (M.L. Maher, editor)," ASCE, New York, 1987.

[McG93] McGuire, JG, et al., "SHADE: Technology for Knowledge-Based Collaborative Engineering, In: *Concurrent Engineering: Research and Applications*, Volume 1, Number 3, September 1993.

[OMG96] "The Common Object Request Broker: Architecture and Specification, Volume 1," Object Management Group, July 1996.

Workshop Agenda

Monday, 3 November 1997, Shops Building (304), Conference Room

8:00 - 8:30 Registration and continental breakfast

8:30 Welcome to NIST
Richard Jackson, Director
Manufacturing Engineering Laboratory (NIST)

8:45 KBS Interoperability in Design: A NIST Perspective
Ram Sriram, Leader
Engineering Design Technologies Group (NIST)

9:00 Overview and Workshop Goals
Robert Allen, IPA Researcher
Engineering Design Technologies Group (NIST)

9:10 Knowledge-based Design Automation and Optimization Systems in a Production Environment -
Siu Tong,
Engenious Software

9:35 The ICAD System: A Generative KB Technology
Prasanna Katragadda,
Concentra Corporation

10:00 Intelligent systems using KB Engineering
Adel Chemaly
TechnoSoft, Inc.

10:25 Rule-Based Interoperability of Heterogeneous Systems
Stanley Su
University of Florida

10:50 BREAK

11:15 Configurator Synchronization
Bruce Ambler
Lucent Technologies

11:40 OKBC: A Programming Foundation for KB Interoperability
Vinay Chaudhri
SRI International

12:05 Knowledge Source Awareness models for Interoperable KB Systems
Ramana Reddy
West Virginia University

12:30 LUNCH

1:30 Breakout Group Organization

1:45 Breakout Working Groups
State of the art in KBS interoperability
Barriers and requirements for KBS interoperability

3:15 BREAK

3:45 Joint panel discussions of BG session summaries

4:45 Software Demonstration
Engenious Software

7:00 Social Hour and Banquet, Gaithersburg Hilton

Tuesday, 4 November 1997, Bldg. 304 - Shops Conference Room

8:00 Continental breakfast

8:45 Summary of Day One Results
Robert Allen

9:00 Group Discussion

10:30 Break

10:45 Summary Discussion/ Action Items Identified/Adjourn

Appendix A

Participant List

Robert Allen
NIST
Building 220, Room 127
Gaithersburg, MD 20899
301 975-3818
robert.allen@nist.gov

Bruce Ambler
Lucent Technologies
101 Crawford Corner, 4J205
Homdel, NJ 07733
(732) 949-5315
bambler@lucent.com

Richard Bolton
NIIIP
1055 Washington Blvd.
Stamford, CT 06901
(203) 978-2946
rbolton@vnet.ibm.com

Scott Chase
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-6535
scott.chase@nist.gov

Vinay Chaudhri
SRI International, MS: EI225
333 Ravenswood Avenue
Menlo Park, CA 94025
(650) 859-3368
vinay@ai.sri.com

Adel Chemaly
TechnoSoft, Inc.
4434 Carver Woods Drive
Cincinnati, OH 45242
(513) 985-9877
adel@technosoft.com

Ismail Deif
Concentra Corporation
61 Londonderry Road
Grafton, MA 01519
(781) 229-4698
ish@concentra.com

Rohinton Irani
General Electric Corporation
1 River Road, Box 8
Schenectady, NY 12301
(518) 387-5635
irani@ crd.ge.com

Kevin Jurrens
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-5486
kevin.jurrens@nist.gov

Prasanna Katragadda
Concentra Corportation
21 North Avenue
Burlington, MA 01803
(781) 229-4617
prasanna@concentra.com

Michael Kerstetter
The Boeing Company
P.O. Box 3707, MS 2T-20
Seattle WA 98124
(206) 544-5535
michael.s.kerstetter@boeing.com

Hans Killicote
ICES/CMU
Hamburg Hall 1201
Pittsburgh, PA 15213
(412) 268-4097
kiliccote@cmu.edu

Bill Lange
InfoTech Enterprises, Inc.
4900 Seminary Road
Alexandria, VA 22311
(703) 671-1900
blange@infotech.org

John Mitchiner
Sandia National Labs
PO Box 5800, MS: 0722
Albuquerque, NM 87185-0722
(505) 844-7825
jlmitch@sandia.gov

Edward Mytych
Boeing Rotorcraft
216 Elton Circle
West Chester, PA
(610) 591-7846
edward.j.mytych@boeing.com

Walter Nederbragt
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-4467
walter.nederbragt@nist.gov

Kishore Pochiraju
Mechanical Engineering Dept.
Stevens Inst. of Technology
Hoboken, NJ 07030
(201) 216-8053
kishore@dmi.stevens-tech.edu

Y. V. Ramana Reddy
Computer Science Dept.
West Virginia University
Morgantown, WV 26506
(304) 293-7226
rar@cerc.wvu.edu

Mark Schwabacher
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-4574
mark.schwabacher @nist.gov

Ram Sriram
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-3507
ram.sriram @nist.gov

Stanley Su
DBSystems R&D, 470 CSE
University of Florida
Gainesville, FL 32611-6125
(352) 392-2693
su@cise.ufl.edu

Rachuri Sudarasan
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-4264
rachuri.sudarasan@nist.gov

Simon Szykman
NIST
Building 220, Room 127
Gaithersburg, MD 20899
(301) 975-4466
simon.szyman@nist.gov

Siu S Tong
Engenious Software
1800 Perimeter Park Drive
Morrisville, NC 27560
(919) 462-7903
tong@engenious.com

Appendix B

Abstracts

Issues in Deploying Knowledge Based Design Automation and Optimization Systems in a Production Environment

Siu S Tong
Engineous Software, Inc.

This presentation describes the successes and difficulties in deploying knowledge based design automation and optimization systems in a production environment. Twelve years of developing and implementing KBSs at GE, and three years experience vending software, form the basis of these observations and conclusions.

To meet the industrial challenge of drastically reducing the product development cycle and cost while maintaining product quality, a new software system was developed at GE Corporate R&D in the early 1980s. It combined traditional mathematical based design optimization algorithms and modern knowledge based system (KBS) approaches to automate, integrate, and optimize engineering design. The software, Engineous, was successfully deployed in eight of 13 GE businesses. In the past three years, a redesigned, commercial version of Engineous, called iSIGHT, was developed and tested in large corporations in five major industries — aerospace, defense, power & utilities, automobile, and industrial manufacturing.

The hybrid knowledge based system and mathematical approach has proven to be useful and efficient in solving complex problems such as the design of aircraft engine turbines, power generation equipment, satellites, transformers, utilities planning, and electrical devices. On average, this technology reduces the design cycle time and manufacturing costs by an order of magnitude, saving tens of millions of dollars. However, there are many challenges in large-scale deployment of this technology to commercial users. The most difficult one is enabling end-users, not knowledge based system developers, to create and maintain the knowledge base. The existing KB systems are too complex for most engineers to learn, and developing complex, practical KBS application often takes too much time and effort. Also, there are many CAD, CAE, and other productivity tools (e.g., spreadsheet) in use in most design environments and substantial development efforts are often needed to link these tools together.

This presentation will highlight some of these challenges, discuss the successes and failures in working around these problems, and suggest future development/improvement of KBS that could significantly increase its use in a practical design and manufacturing environments.

The ICAD System - A Knowledge Based Generative Technology

Prasanna Katragadda
Concentra, Inc.

The ICAD System is a Knowledge Based Engineering software solution used by world class manufacturers in aerospace, automotive and industrial equipment manufacturing, such as Boeing, British Aerospace, Pratt and Whitney, GM, Ford, Jaguar, Lotus, and others, to automate system-level design, product design, tooling and product configuration. The ICAD System uses generative technology to capture and apply generic product design knowledge - both geometric and nongeometric - which includes product structures, development processes and manufacturability rules. Companies that use ICAD greatly trim cycle time, reduce downstream costs, and provide a flexible environment in which to process engineering change orders. Ultimately, ICAD System users shrink a good portion of the design or configuration process, allowing it to be completed in significantly less time than nonusers.

Recently, the ICAD "vision" has grown beyond the individual engineering effort. Through its KBO (Knowledge Based Organization) initiative, The ICAD System is attempting to examine, understand and define such aspects of an organization's "knowledge" as how it is represented, stored, examined, used, exchanged, updated and refined.

Recognizing that in today's business and engineering environment, knowledge without means of interchange is not very useful, our presentation also includes anticipated interoperability issues, such as representation and access methods for knowledge, and the role of international standards in facilitating these tasks.

Rule-based Interoperability of Heterogeneous Systems in NIIIP

Stanley Y. W. Su
Database Systems Research and Development Center
University of Florida

Heterogeneous information systems such as agent systems, knowledge-based systems, database application systems and CAx systems generally have different data and knowledge representations and run on different operating systems and dissimilar computing platforms. To make these heterogeneous systems interoperable as an integrated information system on a local or wide-area network, one popular approach is to encapsulate the functionalities and data of these systems as objects. By doing so, they can be uniformly represented and processed in the integrated information system. This approach is taken by the Object Management Group (OMG), which introduced CORBA and ORB to provide the architecture and communication infrastructure for the interoperation of distributed objects through method activations. In the NIIIP project, distributed objects are modeled in terms of 1) their structural properties and constraints using the international standard modeling language EXPRESS, 2) their methods using OMG's IDL, and 3) their knowledge rules using an event-condition-action-alternative-action (ECAA) rule language developed at the University of Florida. The ECAA rules capture enterprise business rules, policies, security and integrity constraints, and other rules of interoperation associated with distributed objects. An object-oriented knowledge base management system (KBMS) is used to provide the following:

1) GUIs for modeling, editing, browsing, and graphically querying the conceptual model of an enterprise,

2) An object-oriented query language OQL for accessing and manipulating metadata and shared data, and

3) An event and rule server to provide both build-time and run-time event and rule services.

ECAA rules are pre-compiled into rule code which are incorporated into program bindings generated by an IDL compiler for distributed objects, thus achieving "rule-based interoperability" over an ORB. They can also be stored in the KBMS and triggered at run-time when the enterprise knowledge base is accessed and manipulated.

Configurator Synchronization

Bruce Ambler
Lucent Technologies

Lucent Technologies sells complex telecommunications equipment, where much of the equipment configuration is custom designed for each sale. Engineers configure this equipment with the aid of two knowledge-based systems: the first is a sales configurator the second is a factory configurator. The sales configurator is operated by sales people and configures the product to a level that it can be priced and contracted. The second cofigurator is executed when the order gets to the factory and configures the components to the level that the equipment can be built.

Because changes in product design require the configurators to be changed, there is a need for interoperability between a product information system and the configurators. The configurators must be kept in synch with the product and each other since the output of the sales configurator is the input to the factory configurator. The interoperability requirements include an event notification service and a data exchange mechanism. The nature of the data exchange depends on the nature of the knowledge based system. Rule based systems require different information than constraint resolution systems.

OKBC: A Programmatic Foundation for Knowledge Base Interoperability

Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, James P. Rice
SRI International and Stanford University

Open Knowledge Base Connectivity (OKBC) is an application programming interface for accessing knowledge bases stored in knowledge representation systems (KRSs). OKBC is being developed under the sponsorship of DARPA's High Performance Knowledge Base program (HPKB), where it is being used as an initial protocol for the integration of various technology components.

OKBC is a successor of Generic Frame Protocol (GFP) which was primarily aimed at systems that can be viewed as frame representation systems and was jointly developed by Artificial Intelligence Center of SRI International and Knowledge Systems Laboratory of Stanford University.

OKBC provides a uniform model of KRSs based on a common conceptualization of classes, individuals, slots, facets, and inheritance. OKBC is defined in a programming language independent fashion, and has existing implementations in Common Lisp, Java, and C. The protocol transparently supports networked as well as direct access to KRSs and knowledge bases.

OKBC consists of a set of operations that provide a generic interface to underlying KRSs. This interface isolates an application from many of the idiosyncrasies of a specific KRS and enables the development of tools, such as those currently being developed at SRI and Stanford.

Knowledge Source Awareness Models For Interoperable Knowledge Based Systems

R. Reddy
Concurrent Engineering Research center
West Virginia University

Knowledge Based Systems, by definition, depend on one or more sources of knowledge for their operation. In a stand-alone knowledge based system, these knowledge sources are usually "attached" to the inference engine – the heart of the knowledge-based system. With the emergence of the World Wide Web (W3) as a seamless global information infrastructure, it is now possible to construct problem solutions based on a collection of cooperating knowledge based systems. In such an endeavor, each component may depend partly on the knowledge sources associated with one or more knowledge based systems in the group. This can only be possible if these component systems can inter-operate, insofar as they can exploit each other's knowledge sources. Let us take a simple example of a case where two members of a team, each using an "expert office assistant" program wish to manage scheduling and communications. Each system depends on its own knowledge source – say an address book. Unless each system knows about the existence of an address book and deal with converting each other's formats to their own representation, they can never cooperate – because they can not inter-operate. To overcome this problem, the following characteristics are needed:

1. A classification system for various types of knowledge,

2. A means for transforming one representation into another (perhaps using an intermediate canonical representation), and

3. A meta-model, which may be used by each knowledge-based system, to discover the needed source from the domains of the co-operating systems.

This talk provides some plausible scenarios for dealing with the above imperatives.

Appendix C - Presentation Slides

Issues in Deploying Knowledge Based Design Automation and Optimization Systems in a Production Environment

Siu S. Tong
Engenious Software, Inc.

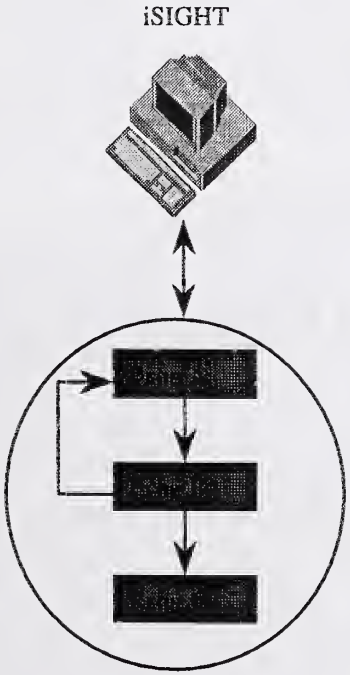
Issues in Deploying Knowledge Based Design Automation and Optimization Systems in a Production Environment

Siu S. Tong
Engineous Software, Inc.

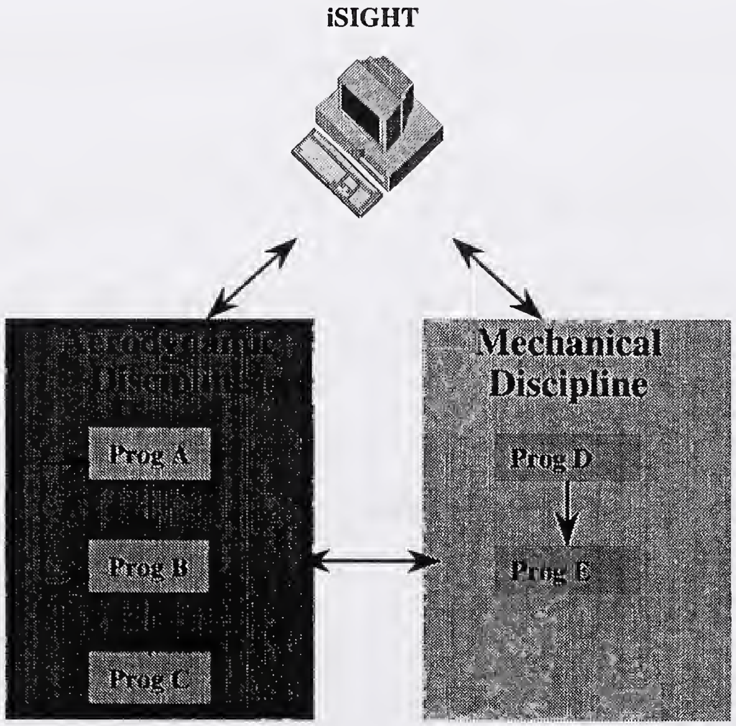
The iSIGHT CAO Environment

Computer-Aided Optimization (CAO) Framework

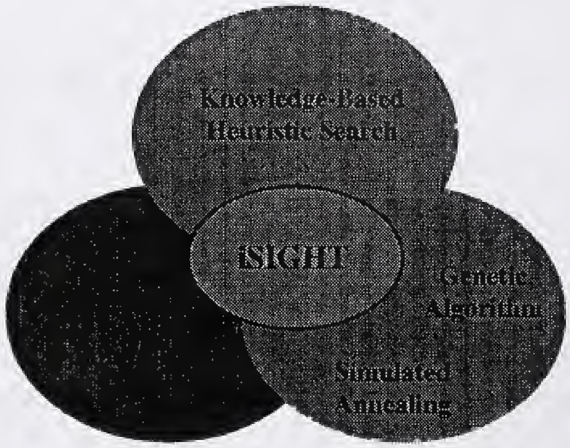
Design Automation



Design Integration



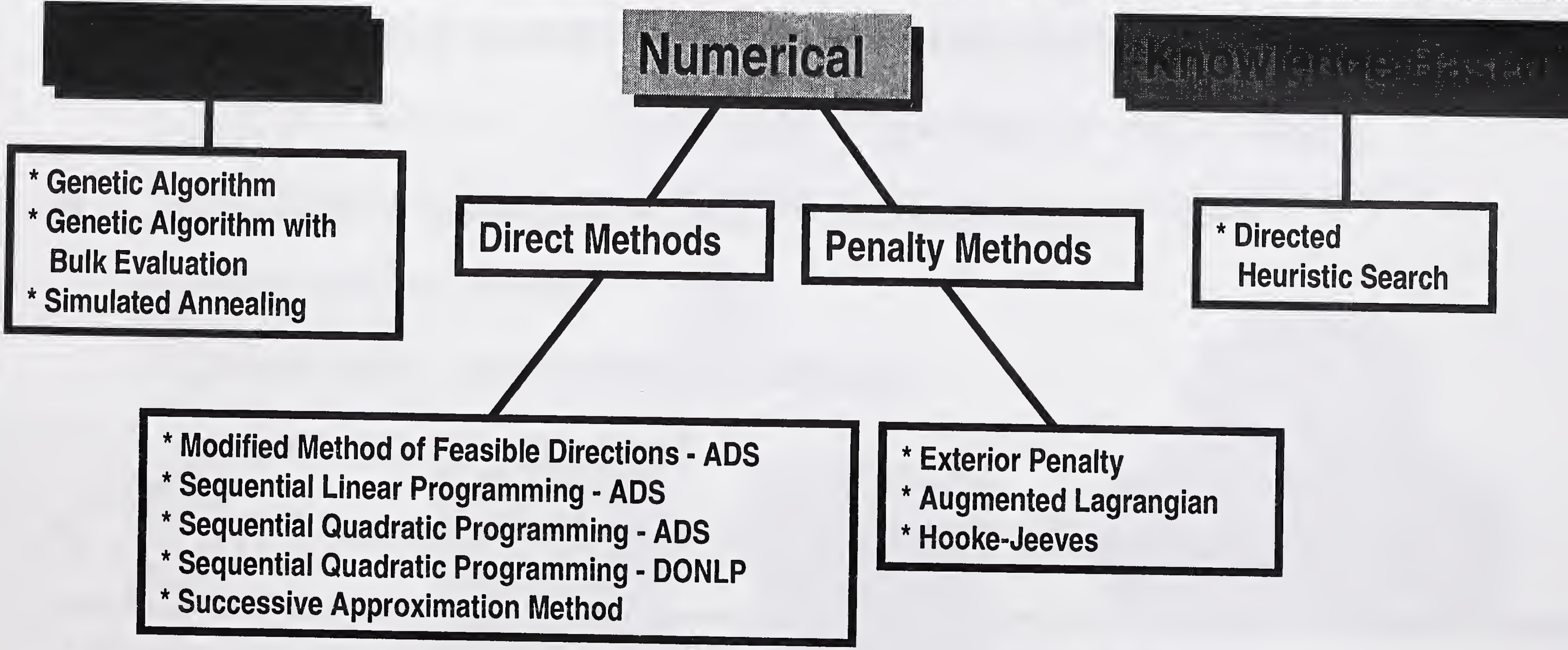
Design Optimization



Optimization Algorithms in iSIGHT

- iSIGHT provides a suite of optimization algorithms
  - Numerical hill climbing
  - Exploratory semi-random search
  - Knowledge based
- Optimization algorithms can be combined in a *plan*
  - Apply a series of optimization algorithms with a plan
  - Add loops & branches for more sophisticated control

iSIGHT Optimization Algorithm Suite



[illegible]

- * Genetic Algorithm
- * Genetic Algorithm with Bulk Evaluation
- * Simulated Annealing

Numerical

Direct Methods

- * Modified Method of Feasible Directions - ADS
- * Sequential Linear Programming - ADS
- * Sequential Quadratic Programming - ADS
- * Sequential Quadratic Programming - DONLP
- * Successive Approximation Method

Penalty Methods

- * Exterior Penalty
- * Augmented Lagrangian
- * Hooke-Jeeves

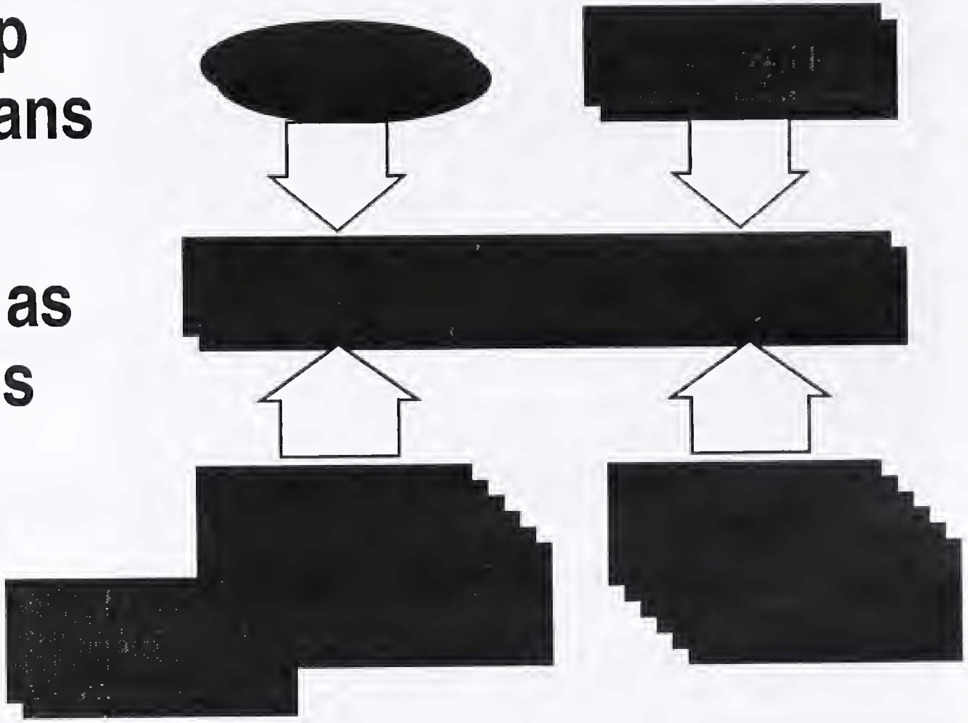
Knowledge-Based

- * Directed Heuristic Search

Optimization algorithms can be combined in series, conditionally, in loops, ...

Knowledge in iSIGHT

- Users can add rules to iSIGHT's expert system to automate set-up and execution of optimization plans
- The optimization plan can use Directed Heuristic Search (DHS) as one of its optimization algorithms

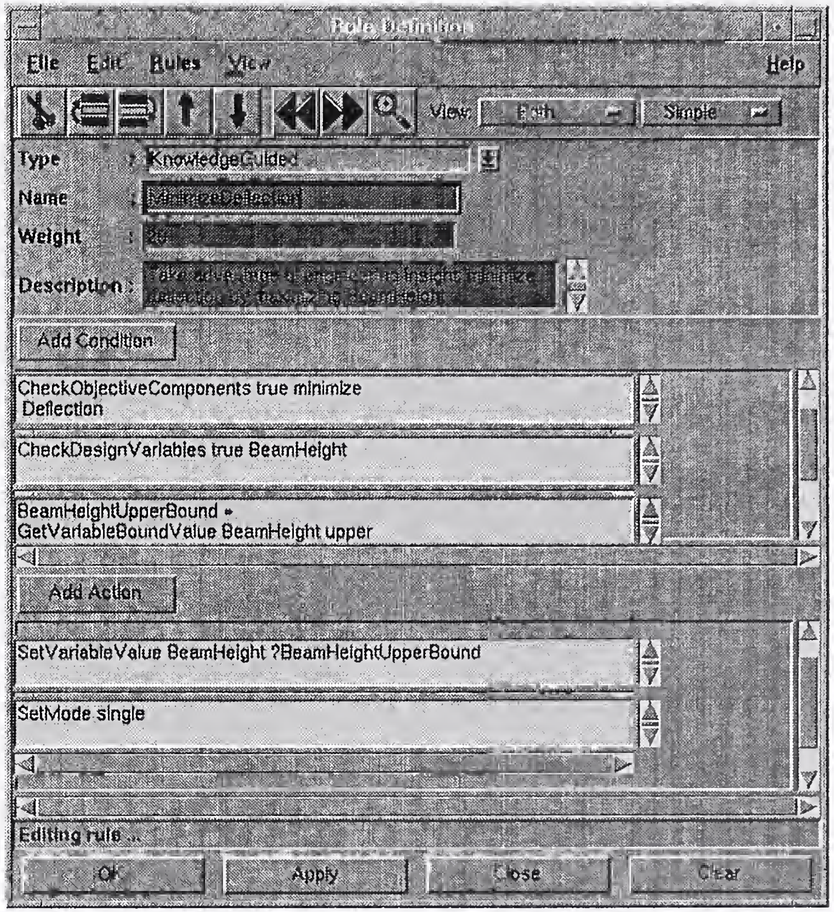


Why Add Knowledge Based Techniques to Mathematical Optimization?

- Find solutions more efficiently than general purpose mathematical techniques
  - Utilize knowledge of engineering physics
- Automate design tool use
  - Add expert knowledge on how to use iSIGHT, how to choose optimization algorithms, how to ...
- Quickly generate a design that is “good enough” before a time consuming search for the best design
  - Deliver a heuristically-derived design in a time crunch, or use the heuristic design to jump start rigorous optimization

iSIGHT Expert System

- Built-in expert system can run iSIGHT using the same commands available to interactive users
  - Choose design variables & objectives, assign values & constraints, specify optimization plans, run programs, ...
- Rule engine based on CLIPS
- Rules can be entered via point & click user interface



Type : KnowledgeGuided

Name : MinimizeDeflection

Weight : 20

Description : [illegible]

CheckObjectiveComponents true minimize Deflection

CheckDesignVariables true BeamHeight

BeamHeightUpperBound = GetVariableBoundValue BeamHeight upper

SetVariableValue BeamHeight ?BeamHeightUpperBound

SetMode single

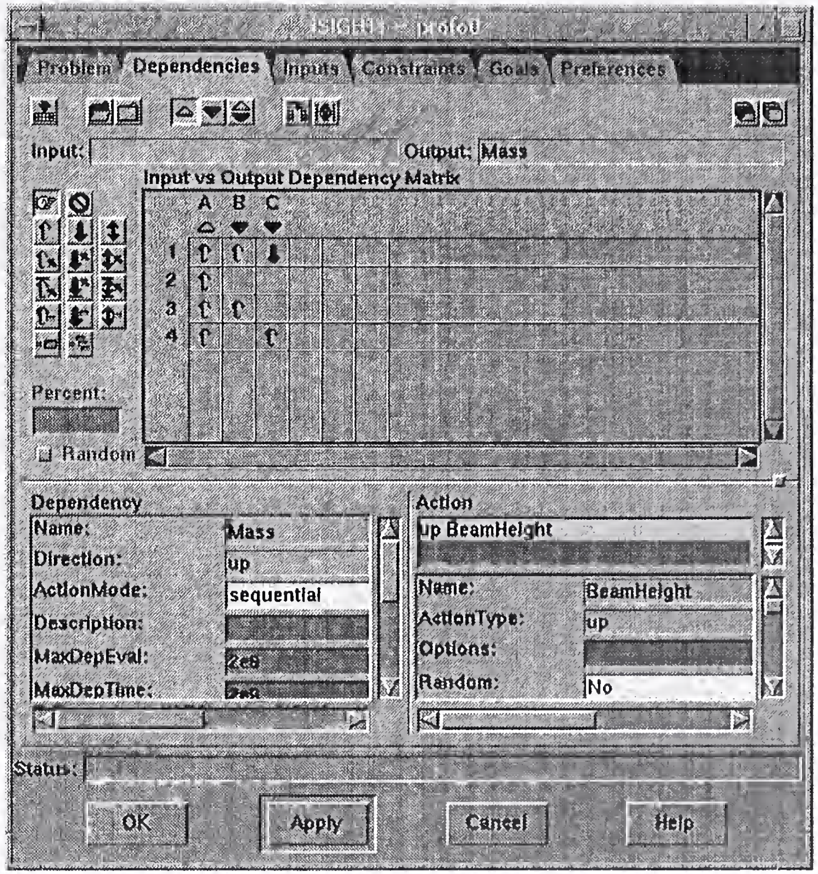
Editing rule ...

Example Application: Preliminary Design of a Multi-Stage Turbine

- Problem: Efficiently determine values for more than 50 preliminary design variables such that turbine efficiency is maximized and weight is minimized
- Experienced turbine designers develop the flowpath geometry first, then solve for other variables that have more localized effects
- Expert system splits the design problem into two parts:
  - If starting with an initial design then
    Use only the flowpath parameters as design variables
  - If the flowpath shape has been optimized then
    Add the remaining design variables to the problem

iSIGHT Directed Heuristic Search

- Knowledge based optimization algorithm invented by Engineous Software
- User describes qualitative relations between parameters
  - "Increasing A increases X"
  - "I don't know how B affects Y"
- User directs search through design domain
  - "Change A before changing B"



Input:

Output: Mass

Dependency

Name: Mass

Direction: up

ActionMode: sequential

Description:

MaxDepEval: 2e8

MaxDepTime:

Action

up BeamHeight

Name: BeamHeight

ActionType: up

Options:

Random: No

Example Application: Jet Engine Installation

- Problem: A major aerospace manufacturer needed to optimize equipment installed with a jet engine
- While using iSIGHT with mathematical optimization, the engineer observed 3 design variables always increased
- The engineer replaced mathematical optimization with DHS and added his new knowledge
- DHS found a better solution in half the time
- Manufacturer estimates improved solution saves several thousand pounds per aircraft

Uses of iSIGHT's Knowledge Based Tools

- Capture knowledge as the user explores the design space
  - Not an easy process with massive amount of data
- Automate design tool use
  - Learning curve for the knowledge base system itself
- Share knowledge with others
  - What's in it?
- Capture existing design knowledge
  - Some early successes

Issues

- Knowledge capture & reuse needs
  - Easy-to-use tools for capturing human knowledge
  - Easy-to-use tools for documenting, searching for, and helping users understand stored knowledge
  - Automated learning/derivation of qualitative relationships to drive heuristic problem
- Lacks standardized access to knowledge bases and knowledge transfer between systems
  - Hardwired CLIPS to iSIGHT
  - Hardwired interface to other rule systems

Appendix D - Presentation Slides

The ICAD System: A Generative Knowledge-based Technology

Prasanna Katragadda
Concentra Corporation

The ICAD System
&
I.C.E.
Product Overviews

Prasanna Katragadda
Director of Operations, The ICAD System

CONCENTRA

Let's define some terms . . .

Knowledge-Based Engineering (KBE)

A methodology for capturing the knowledge about a process and applying it to solve engineering problems.

Dataquest:

"KBE Tools - tools used to capture design intent and build standard practices for controlling and modifying design and manufactring activities."

CONCENTRA

Let’s define some terms . . .

- **Generative Technology**

  Concentra’s implementation of KBE which enables companies to capture and automate the way engineering practices are performed generating new designs directly from functional requirements in minutes not months.

- **The ICAD System**

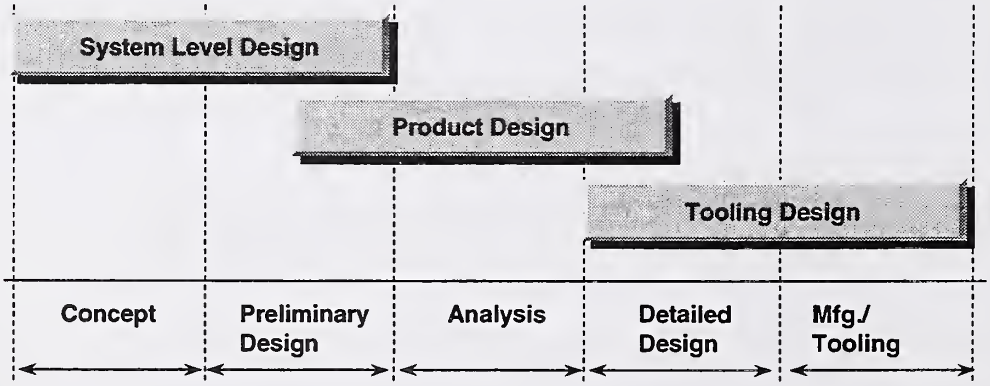
  The name of Concentra’s engineering product which incorporates Generative Technology and is used to create a Generative Model.

- **Generative Model**

  The actual application created when applying Generative Technology with The ICAD System to model an engineering process.

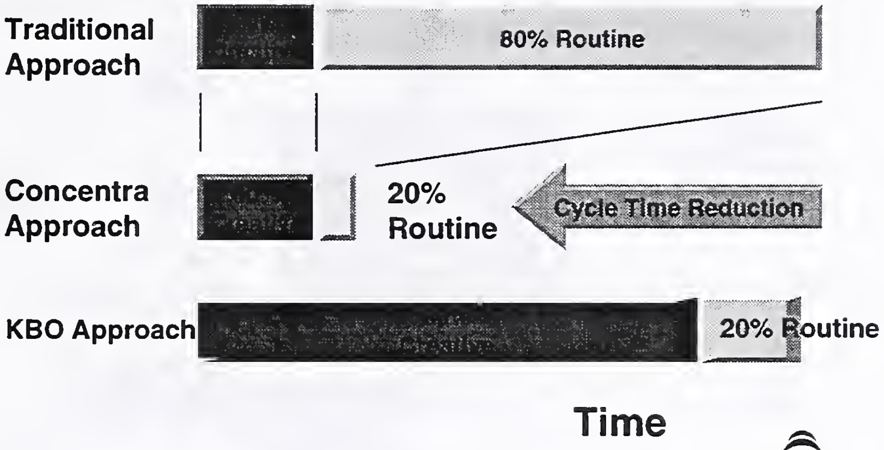
CONCENTRA

What should I automate ?



CONCENTRA

Automating Routine Engineering Cuts Cycle Time



Traditional Approach

80% Routine

Concentra Approach

20% Routine

Cycle Time Reduction

KBO Approach

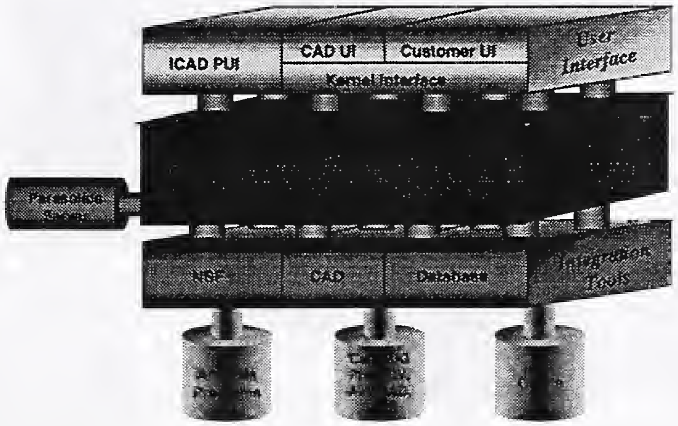
20% Routine

Time

CONCENTRA

Concentra's Generative Technology provides the underlying foundation for application development ...

The ICAD System



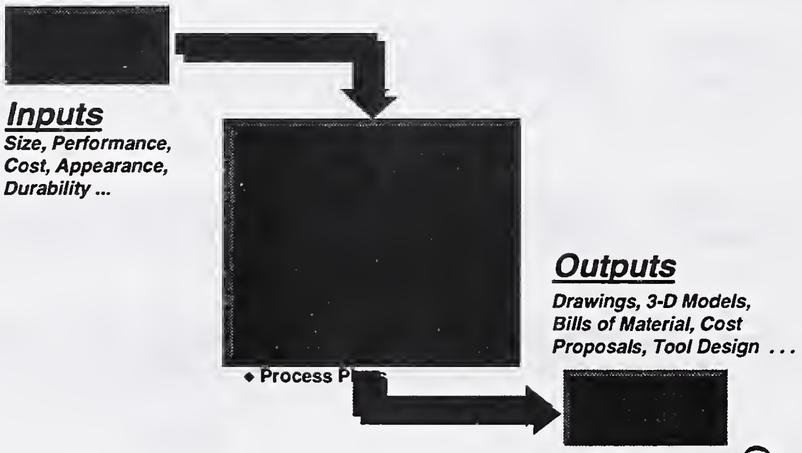
- Enterprise Fit
- Customizable Layout

- Object Oriented
- Scalable Structure

- Open Architecture
- Standards Compliant

CONCENTRA

Customers Generate Designs from Functional Requirements . . .



Inputs

Size, Performance, Cost, Appearance, Durability ...

Outputs

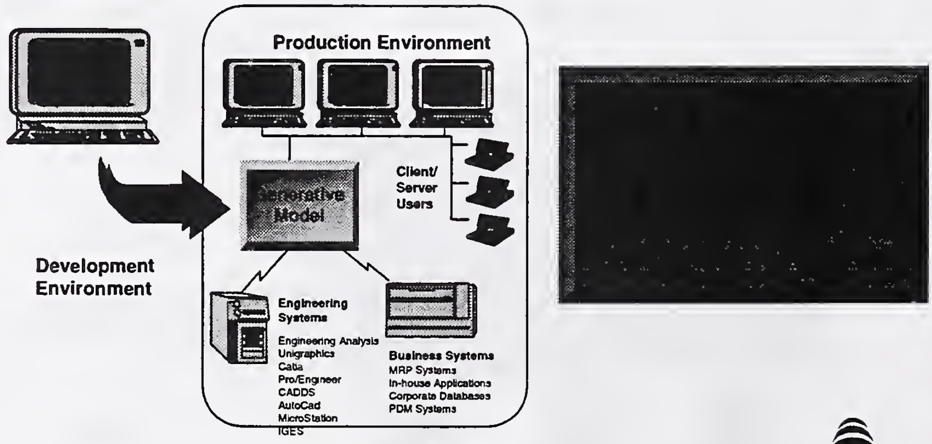
Drawings, 3-D Models, Bills of Material, Cost Proposals, Tool Design . . .

◆ Process Pl

CONCENTRA

System Configuration

Building a Generative Model ...



Production Environment

Generative Model

Client/ Server Users

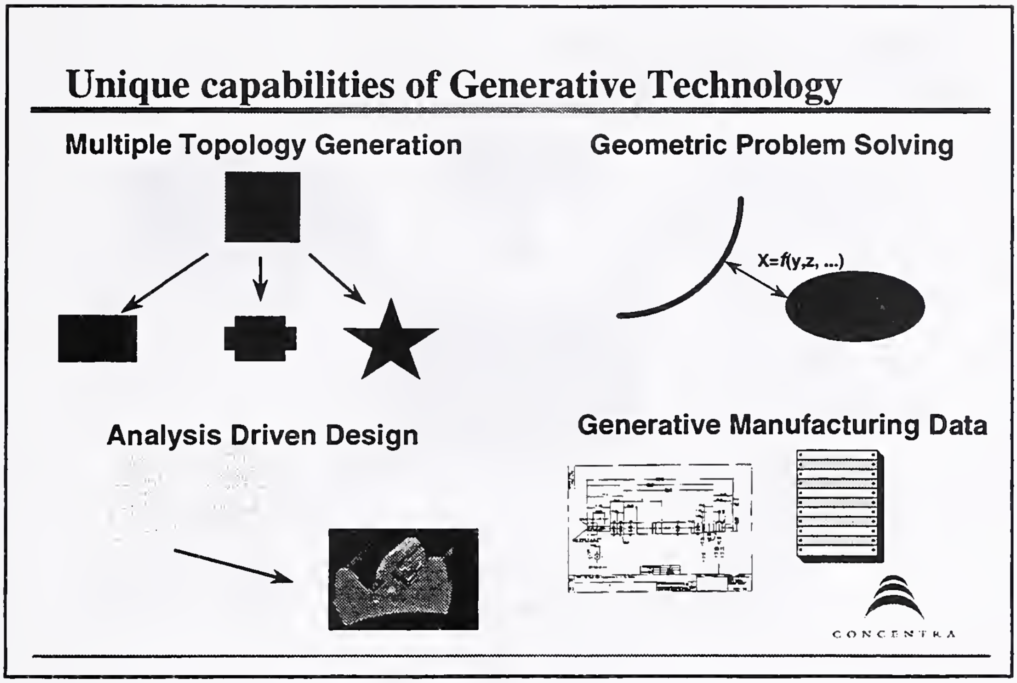
Development Environment

Engineering Systems
- Engineering Analysis
- Unigraphics
- Catia
- Pro/Engineer
- CADDS
- AutoCad
- MicroStation
- IGES

Business Systems
- MRP Systems
- In-house Applications
- Corporate Databases
- PDM Systems

... and using a Generative Model

CONCENTRA



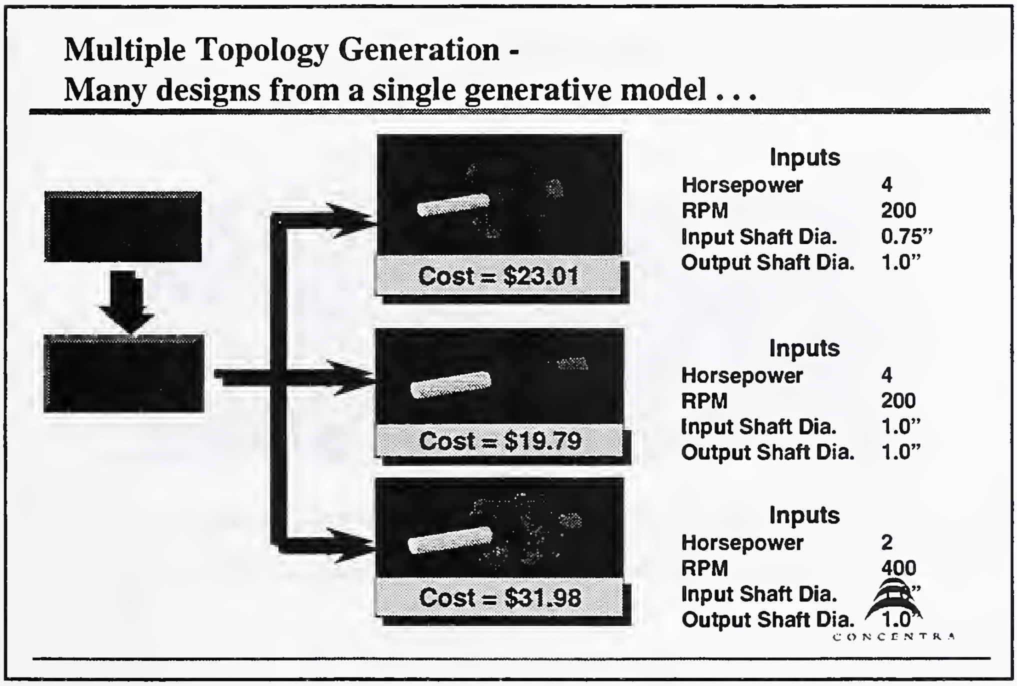
Unique capabilities of Generative Technology

Multiple Topology Generation

Geometric Problem Solving

Analysis Driven Design

Generative Manufacturing Data



Multiple Topology Generation -
Many designs from a single generative model . . .

Cost = $23.01

Inputs	
Horsepower	4
RPM	200
Input Shaft Dia.	0.75”
Output Shaft Dia.	1.0”

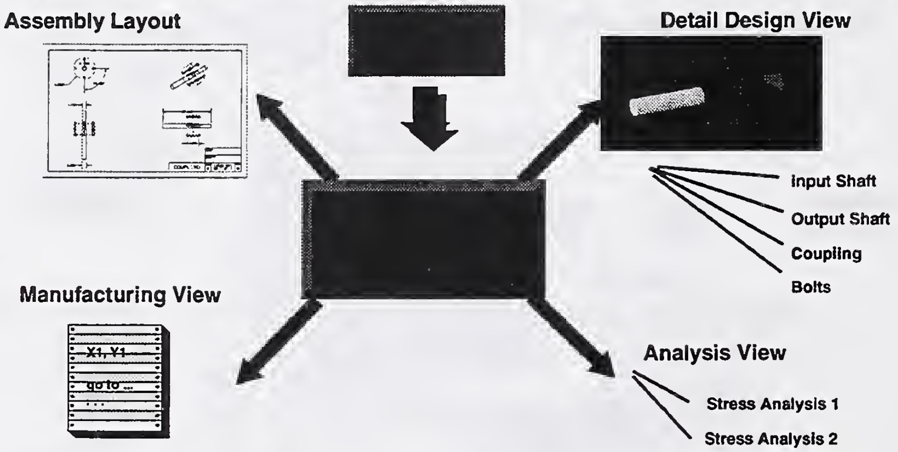
Cost = $19.79

Inputs	
Horsepower	4
RPM	200
Input Shaft Dia.	1.0”
Output Shaft Dia.	1.0”

Cost = $31.98

Inputs	
Horsepower	2
RPM	400
Input Shaft Dia.	[illegible]”
Output Shaft Dia.	1.0”

Multiple Topology Generation - Many views associated with a single design . . .



Assembly Layout

Detail Design View

- Input Shaft
- Output Shaft
- Coupling
- Bolts

Manufacturing View

Analysis View

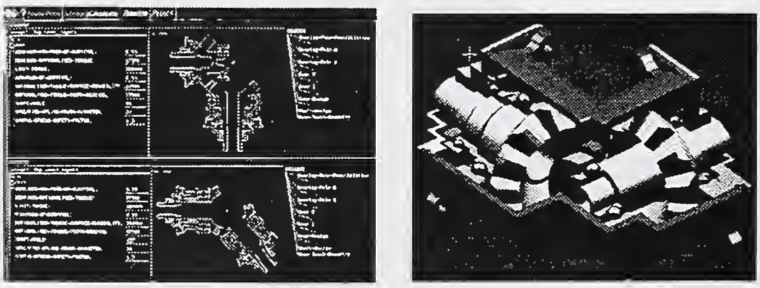
- Stress Analysis 1
- Stress Analysis 2

Full Associativity between all views of the design

CONCENTRA

Driving Product Design With Analysis . . .

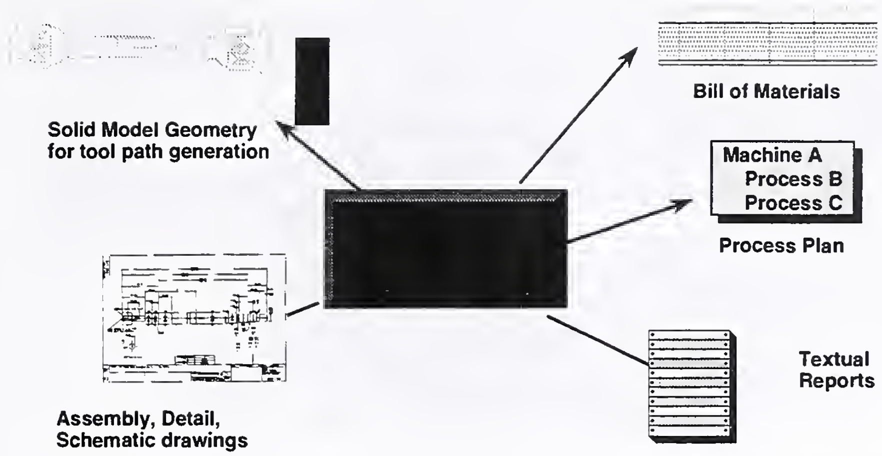
Transmission Gearboxes



- Generative Model drives a series of stress analysis programs.
- Design time reduced from 3 weeks to less than one day.

CONCENTRA

Generating Data for the Manufacturing Process . . .



Bill of Materials

Solid Model Geometry
for tool path generation

Machine A
Process B
Process C

Process Plan

Textual
Reports

Assembly, Detail,
Schematic drawings

Unique capabilities of Generative Technology

✔ ***Multiple Topology Generation*** **allows a single generative model to associate and drive multiple design, assembly, analysis, and manufacturing views.**

✔ ***Geometric Problem Solving*** **provides the ability to solve the difficult geometric design problems insuring proper form, fit, and function.**

✔ ***Analysis Driven Design*** **provides the tools to integrate and drive the model with analysis programs.**

✔ ***Generative Manufacturing Data*** **means feeding downstream processes with the required information necessary to make the part.**

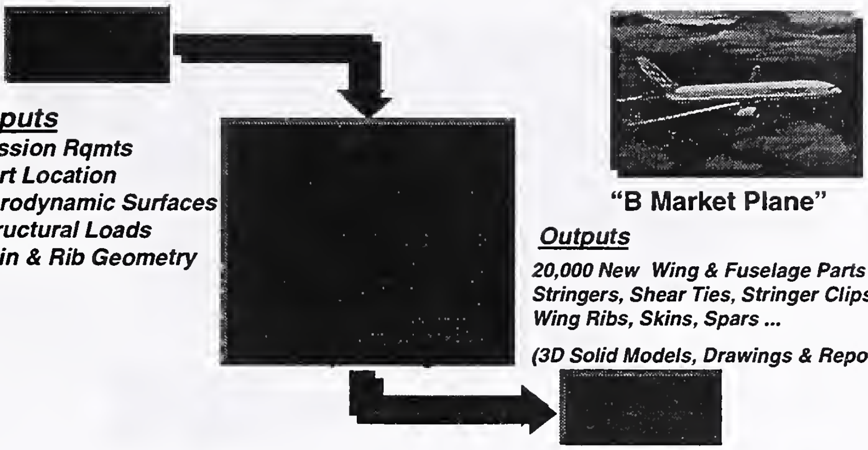
Benefits

- Reducing time to market
  - ➢ through process automation
- Improving product quality
  - ➢ through integration of functional requirements (IPD)
- Reducing costs
  - ➢ through minimizing design changes
- Facilitating technical memory retention
  - ➢ through design practice and process capture

Concentra in Aerospace - Making Concepts Fly

- Leaders in aircraft manufacturing realizing dramatic design cycle time reductions.
  - ➢ Boeing
  - ➢ Aerospatiale
  - ➢ British Aerospace

Boeing 777 - New Longer Range Plane



Inputs

- *Mission Rqmts*
- *Part Location*
- *Aerodynamic Surfaces*
- *Structural Loads*
- *Skin & Rib Geometry*

"B Market Plane"

Outputs

20,000 New Wing & Fuselage Parts
Stringers, Shear Ties, Stringer Clips
Wing Ribs, Skins, Spars ...

(3D Solid Models, Drawings & Reports)

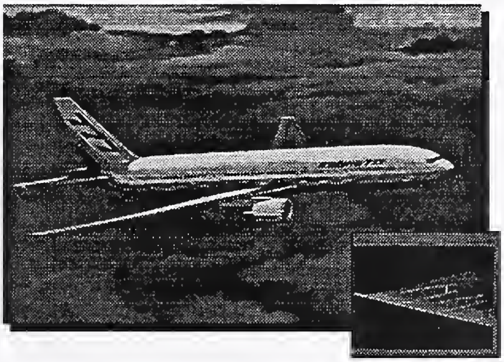
Cycle Time Reduction Achieved

Results:

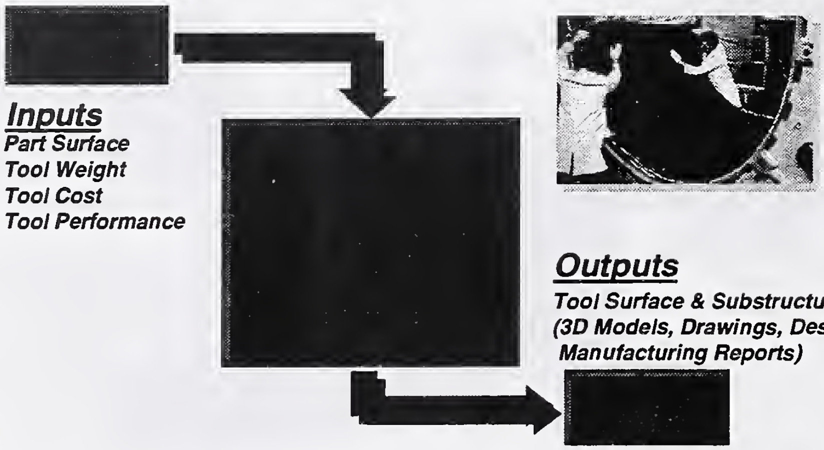
◆ $20 benefit for each $1 invested

◆ One million engineering hours

◆ Total savings over $40 million



Composite Tooling - *Short Brothers lay-up tools*



Inputs

Part Surface
Tool Weight
Tool Cost
Tool Performance

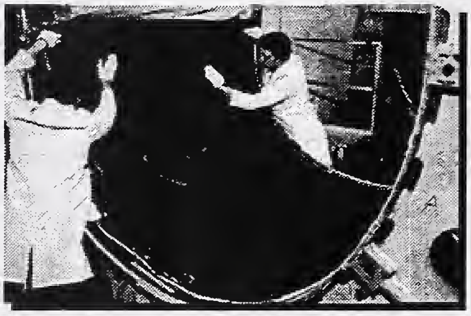
Outputs

Tool Surface & Substructure layout (3D Models, Drawings, Design & Manufacturing Reports)

Cycle Time Reduction Achieved

Results:

- **Tooling Design Cycle time Reduced 93%**
- **9 Month Payback**
- **ROI = 280%**
- **ECO's dramatically reduces**



Concentra in Automotive

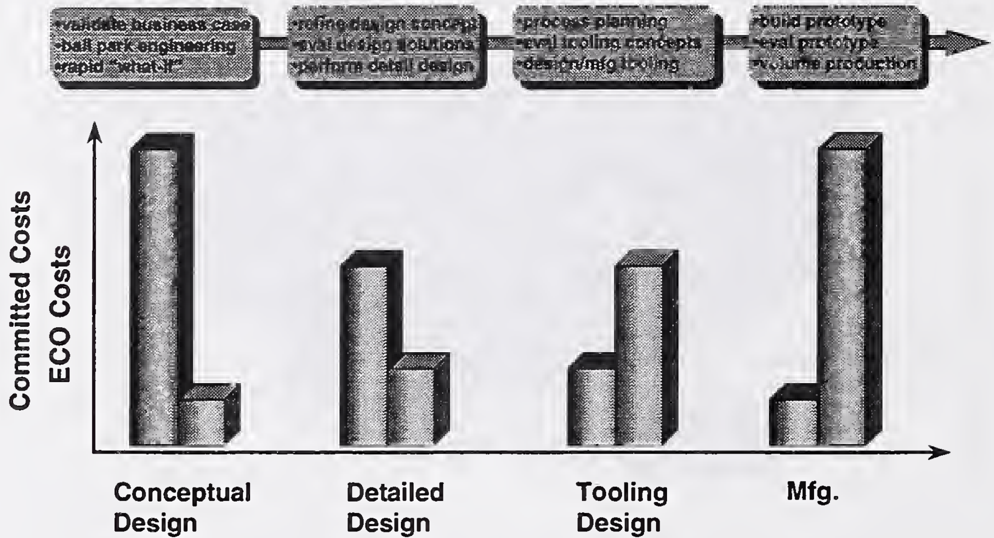
I.C.E.

Integrated Car Engineer™

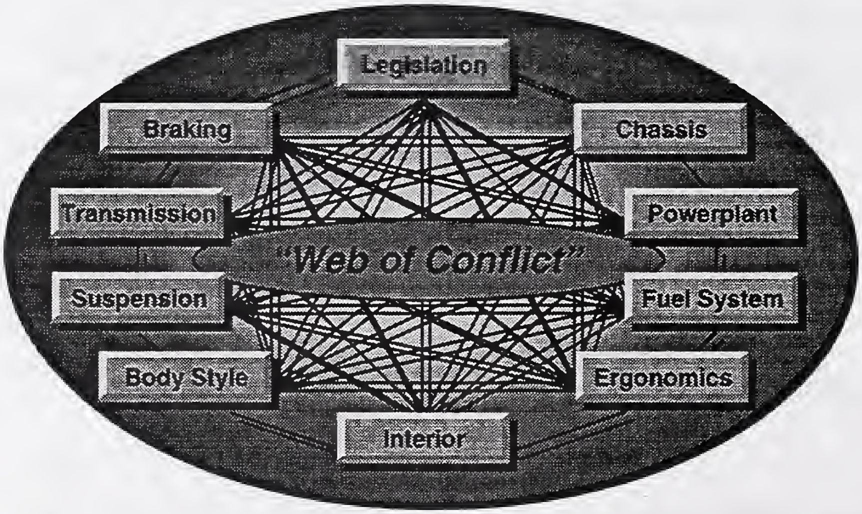
Concentra in Automotive - From Zero to Concept in Record Time!

- I.C.E. is an ICAD Application
- Designed for Automotive Manufacturers & Suppliers
- Automates the Conceptual Design Process
  - A Very Manual Process Today
  - CAD/CAM Tools Not Sufficient This Early in the Design
- Feasibility Studies Now Much Quicker and Easier
  - Knowledge Base of your Best Engineers
  - Relevant Legislation and Design Codes
  - Manufacturing Rules
- Months to Minutes is REALLY working!

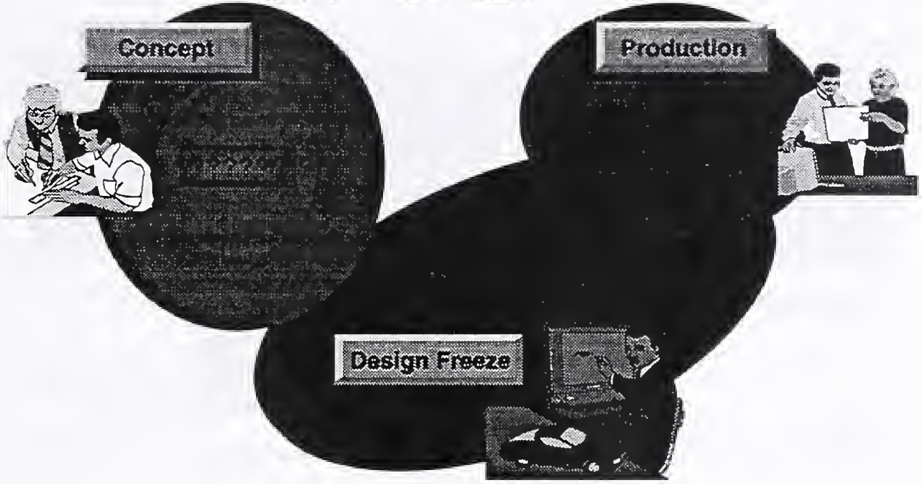
The vehicle development process commits 80% of cost during the conceptual design phase



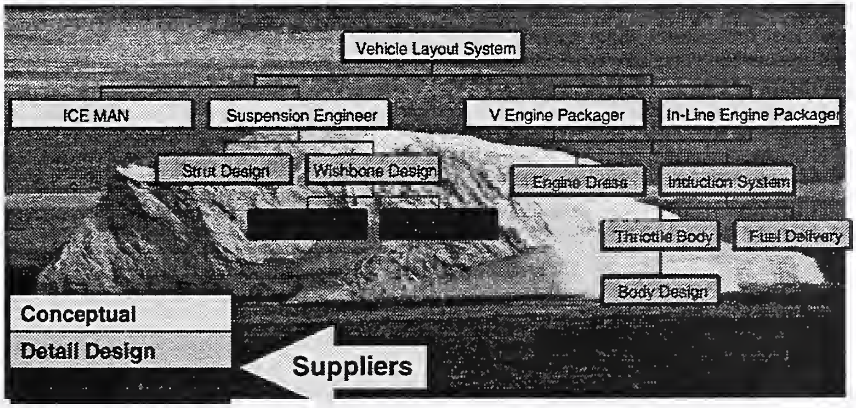
To meet these challenges car makers must balance many different disciplines



Tools available today



The I.C.E. Berg



Vehicle Layout System

ICE MAN

Suspension Engineer

Strut Design

Wishbone Design

V Engine Packager

In-Line Engine Packager

Engine Dress

Induction System

Throttle Body

Fuel Delivery

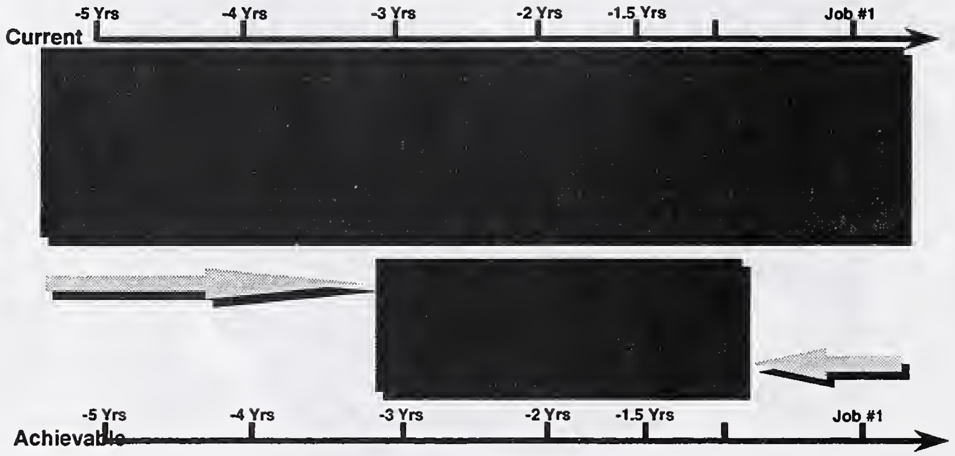
Body Design

Conceptual

Detail Design

Suppliers

A cleaner upfront vehicle design will further compress vehicle development schedules ...



Benefits of I.C.E.

- Frees the creative spirit!
- Drives the process from the highest level, the Concept
- Enables delayed decisions (Ref Toyota Second Paradox)
- Multiple iterations improves quality
- Skills and best practices 'baked into the process' (Right first time, consistency, retention of 'design intellect' etc)
- Collapsed timescales

...easily the fastest route from Concept to Design Freeze Point

Delivering Engineering Knowledge to the Enterprise

Positioning Engineering Companies for the 21st Century

Current Realities in Manufacturing Companies

- Competitive edge is driven by process dominance rather than technology prominence
- Customer dictated markets require flexible engineering and manufacturing capabilities
- New product development projects typically leave poorly understood, tangled and inefficient engineering and manufacturing processes
- IT technologies have been only marginally effective at improving product development process
- Downsizing has become the dominant management approach for controlling expenses

The Downsizing Paradox

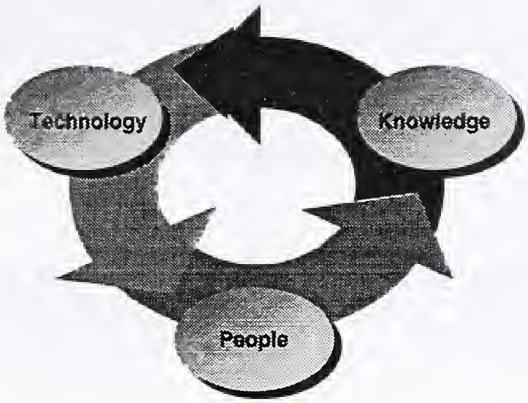
- Knowledge is the key asset for the corporation
- Today, a companies knowledge assets are embodied in its employees
- Employees no longer expect to spend more than a few years at one company
- Knowledge assets are extremely volatile

Characteristics of Successful Product Development

- Maximizes reuse of company knowledge
- Maximizes use of “Best-Practices”
- Takes full advantage of automation
- Allows flexibility and customization
- Provides a manageable legacy of artifacts and technologies

Product Development Strategies

- A winning strategy must link together three key elements



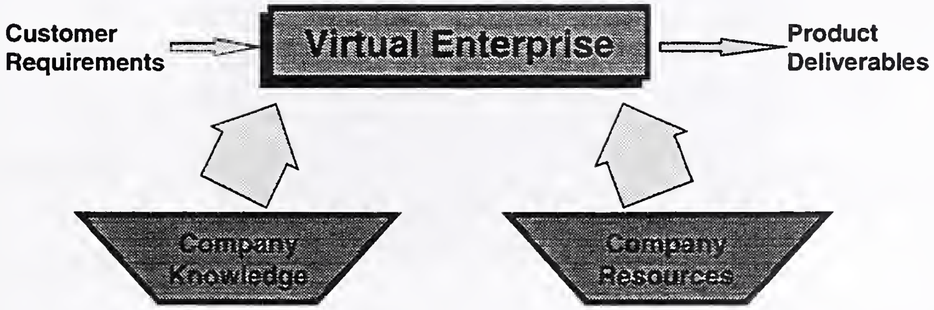
Technology

Knowledge

People

Virtual Enterprises for Product Development

- A Virtual Enterprise is a collection of company resources and knowledge which are orchestrated together to produce a given product offering



Customer Requirements

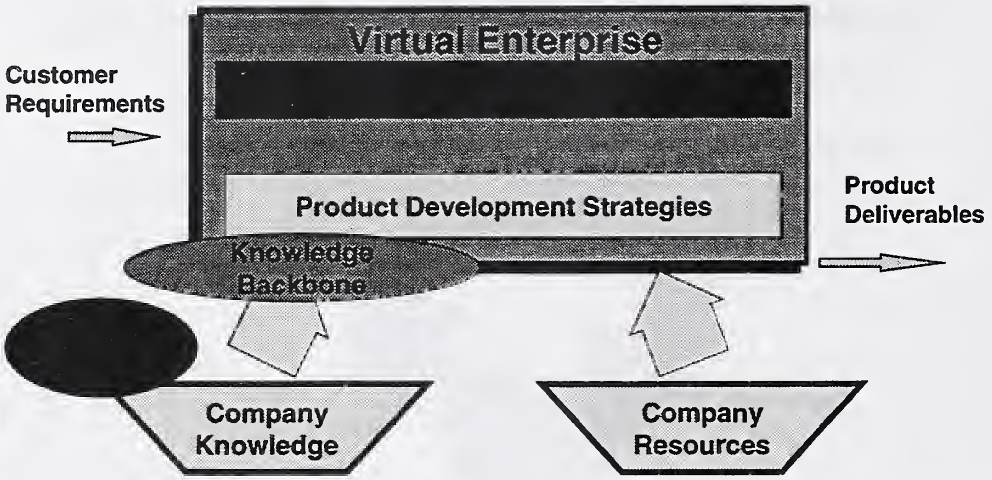
Virtual Enterprise

Product Deliverables

Company Knowledge

Company Resources

A Modular Product Development Strategy



Customer Requirements

Virtual Enterprise

Product Development Strategies

Knowledge Backbone

Product Deliverables

Company Knowledge

Company Resources

Knowledge Packets

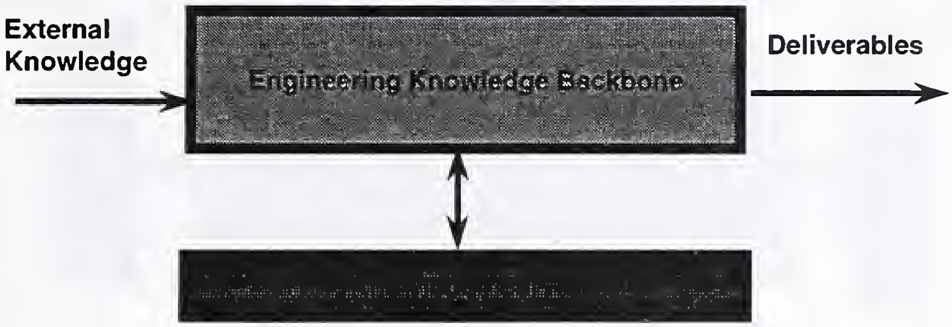
- A fundamental package of stable company knowledge

Inputs

Outputs

Engineering Knowledge Backbone

- A flexible framework for knowledge integration and artifact generation



External Knowledge

Engineering Knowledge Backbone

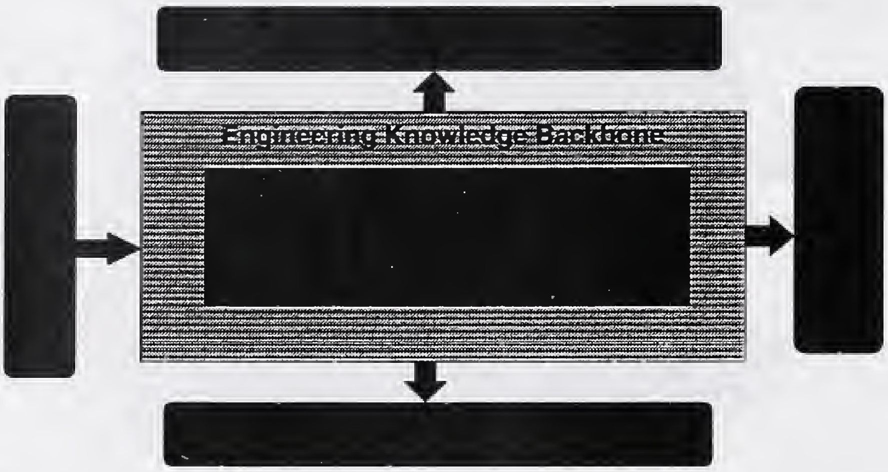
Deliverables

Engineering Knowledge as a Competitve Weapon

- The legacy of the past 50 years has left engineering companies with core competancies based on engineering knowledge
- Product development strategies hinge on being able to deliver the right knowledge to the key product engineers when they need it
- Capturing Engineering Knowledge requires a language built around Engineering Objects
- Sharing Engineering Knowledge requires an architecture which supports Knowledge delivery across the organization

Concentra's Role in the Virtual Enterprise

- The new KBO architecture



Concentra's Future Role in the Virtual Enterprise

- Knowledge Packets are stored in the Knowledge Repository
- Knowledge Server exposes Knowledge Packets as Components (Corba Objects)
- External engineering knowledge contained in other systems can be integrated by exposing them as Components

CONCENTRA

Knowledge Interoperability

(A discussion)

Ish Deif
Principal Engineer, The ICAD System

No one has all the knowledge

Even if we developed a virtual Isaac Newton, he'd only discover Newtonian mechanics...

We'd still need to develop a virtual Einstein to discover relativity!

Knowledge Interoperability

First, data sharing became important

Next, information sharing became crucial

Now, Knowledge sharing is essential

Guiding Principle:
The whole is greater than the sum of its parts

Knowledge Interoperability: What

- Domain of knowledge to capture and represent
  - Engineering knowledge
    - creation of geometric entities
    - code checks
  - Manufacturing knowledge
    - manufacturing processes
    - routings and tooling
  - Business knowledge
    - efficiencies of scale
    - labor and manufacturing costs
  - Other types of knowledge

Knowledge Interoperability: How

- Means of knowledge representation
  - Language(s) rich enough to capture targeted knowledge domains
  - Knowledgebases (Databases holding knowledge)

- Means of knowledge interchange and access
  - File transfer mechanisms
  - Knowledgebase access
  - Embedded systems
  - Distributed objects
  - Intelligent agents
  - Knowledge packets

Knowledge Interoperability: Who

- Mechanisms for recognition of knowledge
  - Automatic browsing mechanisms
  - Standard representation
  - Standard interfaces

Knowledge Interoperability: Issues

- Maintaining existing interfaces
  - Browsing and smart clients can help
  - Intelligent agents
    - can use translation facilities to talk to client in client's own language

- Extending knowledge
  - How to extend existing knowledge domain
  - How to add knowledge for new applications
  - How to integrate old and new knowledge

Role of Knowledge Standards

- Define knowledge
  - What constitutes “knowledge” in a particular domain
  - Knowledge as based on application
    (Knowledge of how to design is different from knowledge of how to manufacture, how to use, etc.)
  - What capabilities/interfaces to support

- Define means of access to knowledge

- Universal language

Knowledge Interoperability: Present and Future

- Present
  - CORBA and COM
  - SDAI

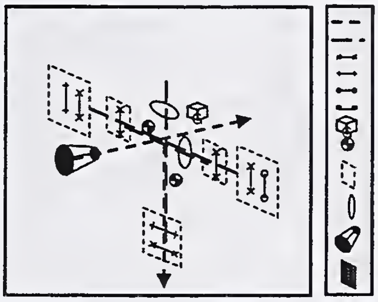
- Future
  - Declarative knowledge
    - Inferred procedures
  - Knowledge marts
    - “Lease your expertise”
  - Cooperative approaches to solving problems
    - Virtual engineers using knowledge packets
  - High-level problem descriptions
    - Natural language

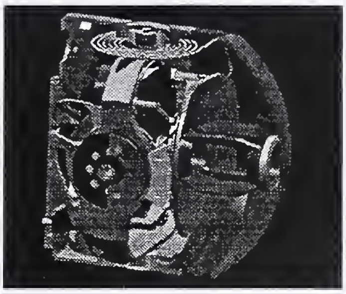
Appendix E - Presentation Slides

Intelligent Systems Using Knowledge-based Engineering

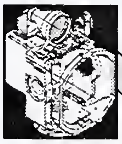
Adel Chemaly
Technosoft, Inc.

Intelligent Systems
using Knowledge Based Engineering





TECHNOSOFT INC.



LANTIRN

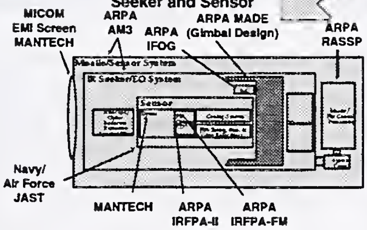
Javelin

TADS/PNVS

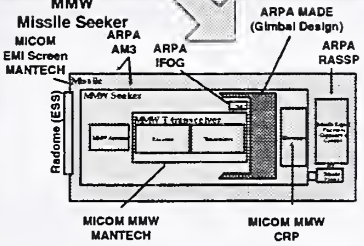
The RaDEO IGD Program Interactive Gimbal Design (IGD)

The design, analysis, and manufacturing of gimbals and integrated optical systems is the very foundation of the entire product line at Lockheed Martin Electronics and Missiles, ranging from fire control systems to missile systems. The RaDEO-IGD System, offering an integrated Gimbal Design Process, directly impacts each of E&M's Programs and complements other DARPA and DoD investments.

EO/IR
Seeker and Sensor

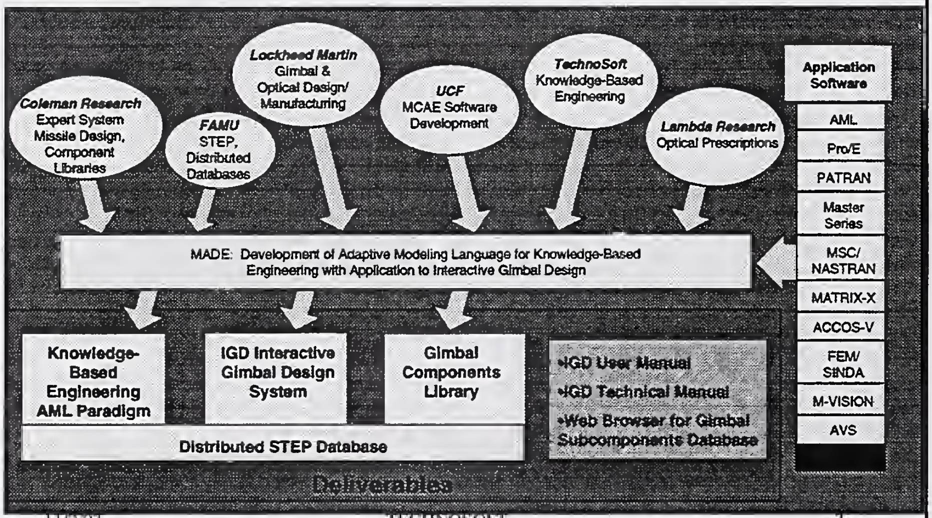


MMW
Missile Seeker



RaDEO-IGD Program

Deliverables



Coleman Research
Expert System
Missile Design,
Component
Libraries

FAMU
STEP,
Distributed
Databases

Lockheed Martin
Gimbal &
Optical Design/
Manufacturing

UCF
MCAE Software
Development

TechnoSoft
Knowledge-Based
Engineering

Lambda Research
Optical Prescriptions

Application Software
- AML
- Pro/E
- PATRAN
- Master Series
- MSC/ NASTRAN
- MATRIX-X
- ACCOS-V
- FEM/ SINDA
- M-VISION
- AVS

MADE: Development of Adaptive Modeling Language for Knowledge-Based Engineering with Application to Interactive Gimbal Design

Knowledge-Based Engineering AML Paradigm

IGD Interactive Gimbal Design System

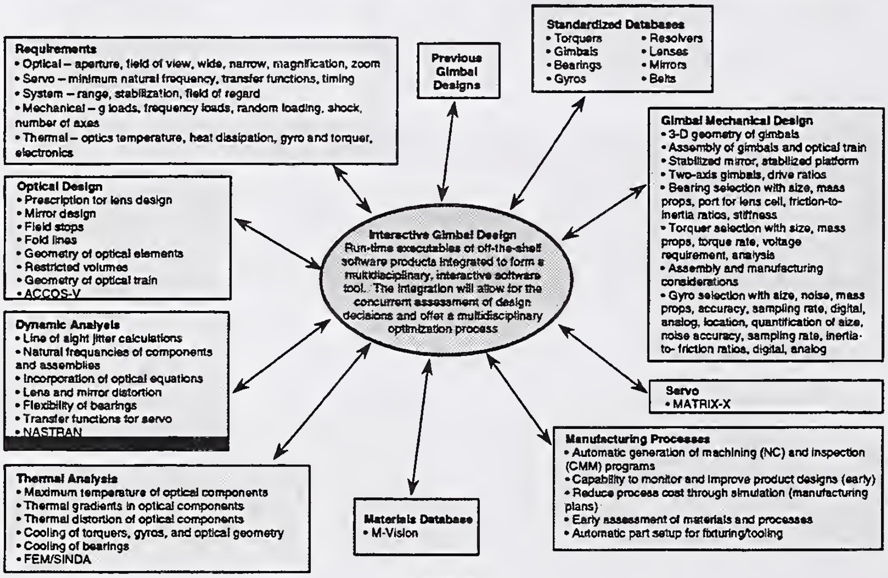
Gimbal Components Library

Distributed STEP Database

- •IGD User Manual
- •IGD Technical Manual
- •Web Browser for Gimbal Subcomponents Database

Deliverables

Functional Capability of the IGD System



Requirements
- Optical – aperture, field of view, wide, narrow, magnification, zoom
- Servo – minimum natural frequency, transfer functions, timing
- System – range, stabilization, field of regard
- Mechanical – g loads, frequency loads, random loading, shock, number of axes
- Thermal – optics temperature, heat dissipation, gyro and torquer, electronics

Previous Gimbal Designs

Standardized Databases
- Torquers
- Gimbals
- Bearings
- Gyros
- Resolvers
- Lenses
- Mirrors
- Belts

Gimbal Mechanical Design
- 3-D geometry of gimbals
- Assembly of gimbals and optical train
- Stabilized mirror, stabilized platform
- Two-axis gimbals, drive ratios
- Bearing selection with size, mass props, port for lens cell, friction-to-inertia ratios, stiffness
- Torquer selection with size, mass props, torque rate, voltage requirement, analysis
- Assembly and manufacturing considerations
- Gyro selection with size, noise, mass props, accuracy, sampling rate, digital, analog, location, quantification of size, noise accuracy, sampling rate, inertia-to- friction ratios, digital, analog

Optical Design
- Prescription for lens design
- Mirror design
- Field stops
- Fold lines
- Geometry of optical elements
- Restricted volumes
- Geometry of optical train
- ACCOS-V

Interactive Gimbal Design
Run-time executables of off-the-shelf software products integrated to form a multidisciplinary, interactive software tool. The integration will allow for the concurrent assessment of design decisions and offer a multidisciplinary optimization process

Dynamic Analysis
- Line of sight jitter calculations
- Natural frequancies of components and assemblies
- Incorporation of optical equations
- Lens and mirror distortion
- Flexibility of bearings
- Transfer functions for servo
- NASTRAN

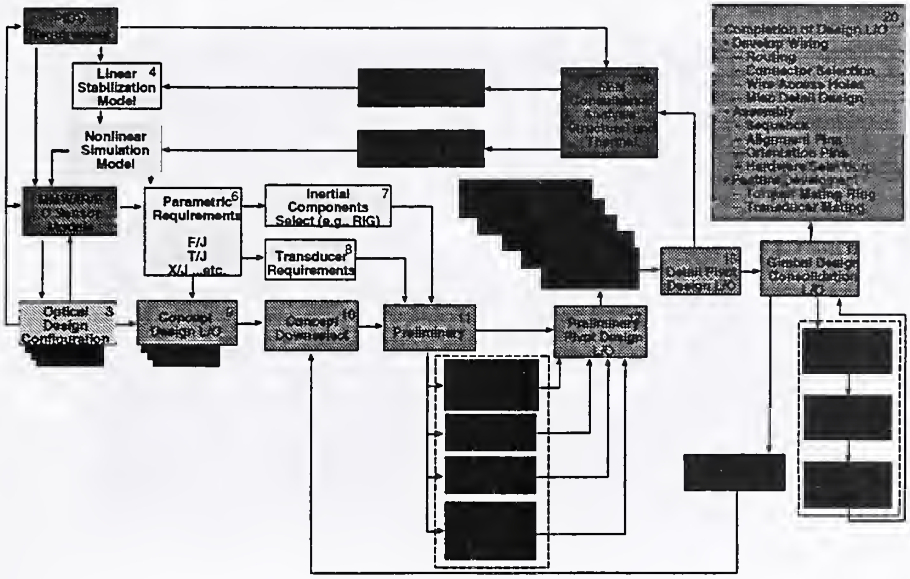
Servo
- MATRIX-X

Thermal Analysis
- Maximum temperature of optical components
- Thermal gradients in optical components
- Thermal distortion of optical components
- Cooling of torquers, gyros, and optical geometry
- Cooling of bearings
- FEM/SINDA

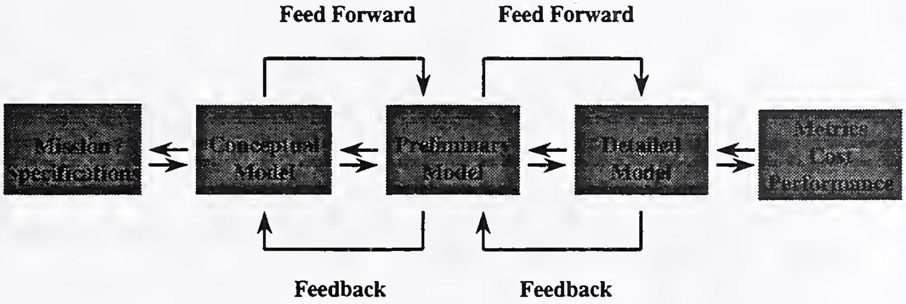
Materials Database
- M-Vision

Manufacturing Processes
- Automatic generation of machining (NC) and inspection (CMM) programs
- Capability to monitor and improve product designs (early)
- Reduce process cost through simulation (manufacturing plans)
- Early assessment of materials and processes
- Automatic part setup for fixturing/tooling

The Gimbal Design Process



Product Design Process


Feed Forward

Feed Forward

Mission / Specifications

Conceptual Model

Preliminary Model

Detailed Model

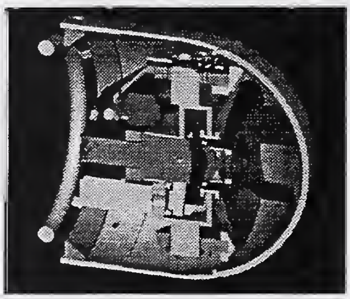
Metrics Cost Performance

Feedback

Feedback

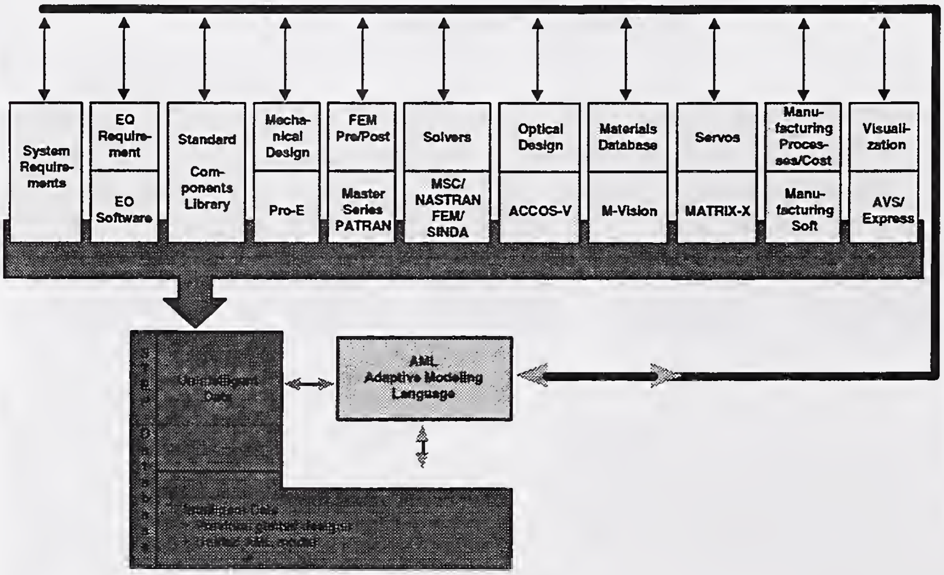
Gimbal Design Statistics of a Missile System

	Optics	Mechanical	FE Analysis	Servo	Total
Design 1	1,002[1] hrs	4,955 hrs	1,214 hrs	1,363 hrs	8,534 hrs
Design 2	1,285 hrs	2,356 hrs	3,384 hrs	0 hrs	7,025 hrs



[1] All data in this table has been multiplied by the same constant factor in order to protect proprietary information.

Basic Structure of the IGD System



System Requirements

EQ Requirement
EQ Software

Standard Components Library

Mechanical Design
Pro-E

FEM Pre/Post
Master Series PATRAN

Solvers
MSC/NASTRAN FEM/SINDA

Optical Design
ACCOS-V

Materials Database
M-Vision

Servos
MATRIX-X

Manufacturing Processes/Cost
Manufacturing Soft

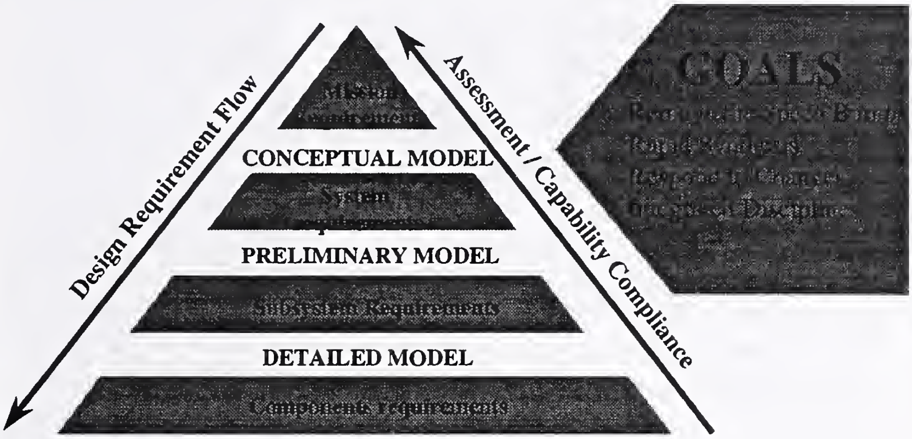
Visualization
AVS/Express

Unintelligent Data

AML Adaptive Modeling Language

Intelligent Data

Basics of the IGD System Part Models

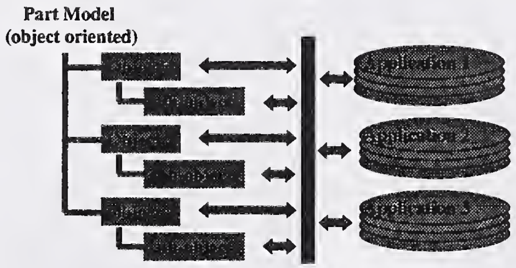


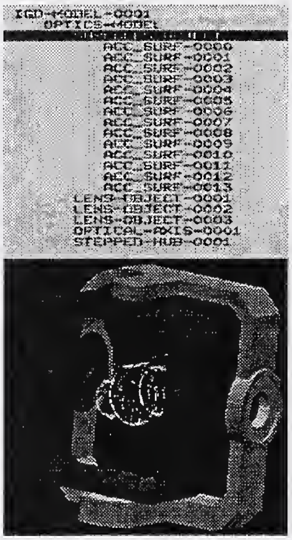
OBJECTIVES

Organize the vital engineering knowledge and expertise to integrate and automate the entire gimbal engineering cycle from conceptual design to production

Development of an Interactive Gimbal Design (IGD) System based on AML that will capture the gimbal design process followed at Lockheed Martin. The IGD structure will allow for a creative design environment and capture the knowledge of that creativity.

IGD's Part Model Representation

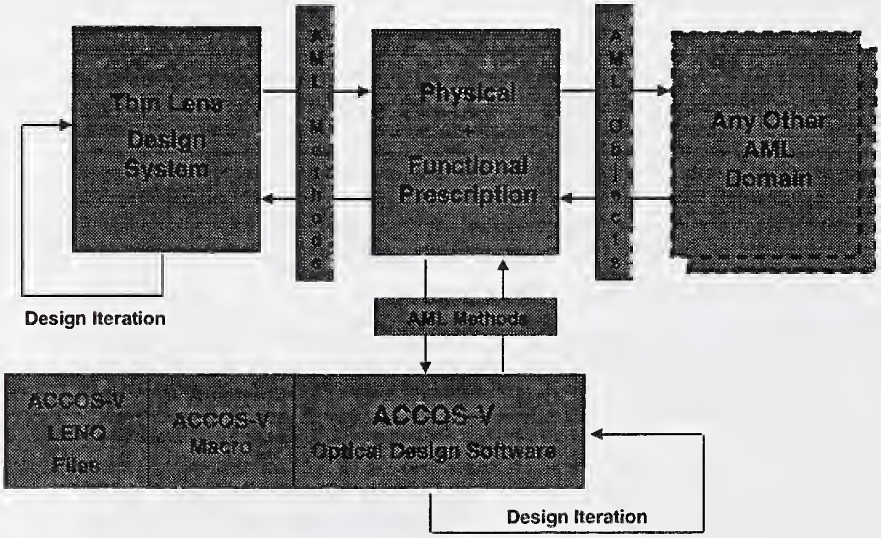




```
IGD-MODEL-0001
  OPTICS-MODEL
        ACC_SURF-0000
        ACC_SURF-0001
        ACC_SURF-0002
        ACC_SURF-0003
        ACC_SURF-0004
        ACC_SURF-0005
        ACC_SURF-0006
        ACC_SURF-0007
        ACC_SURF-0008
        ACC_SURF-0009
        ACC_SURF-0010
        ACC_SURF-0011
        ACC_SURF-0012
        ACC_SURF-0013
    LENS-OBJECT-0001
    LENS-OBJECT-0002
    LENS-OBJECT-0003
    OPTICAL-AXIS-0001
    STEPPED-HUB-0001
```

The design strategy is captured within a part model, represented by a hierarchy of objects, that reflects the design intent and the strategy of various engineering processes.

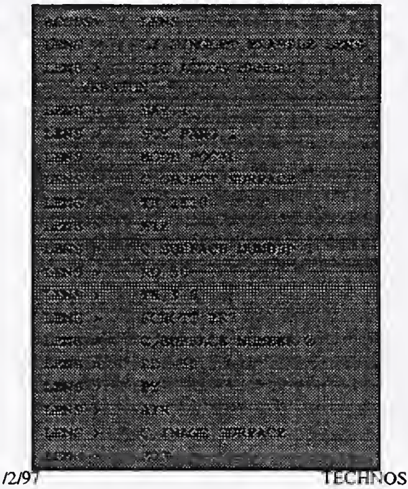
Knowledge Based System for Optical Design

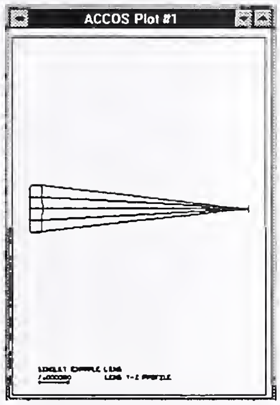


Optical System Objectives

- Dynamic link between ACCOS-V and AML IGD
- Read ACCOS-V "LENO" files (legacy data)
  - Converts ACCOS LENO to AML Optics Analysis Model
  - Allows use of legacy data and analysis strategy
- Writes ACCOS-V "LENO" files
  - Translates AML data to valid ACCOS-V format data
- Performs parameter lookups
  - Requests data from ACCOS-V for solve data
- All ACCOS-V analysis is available without reinventing analysis in AML

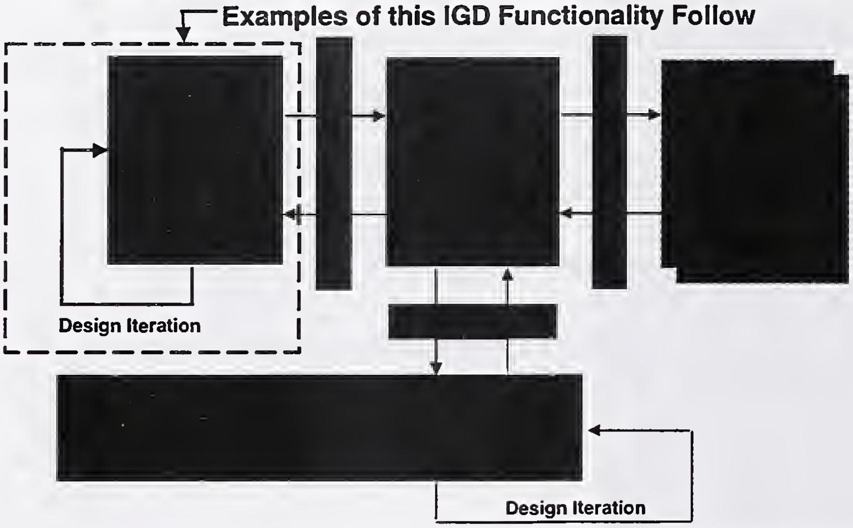
Example Lens Prescription: LENO File





Optical Design

Thin Lens Design



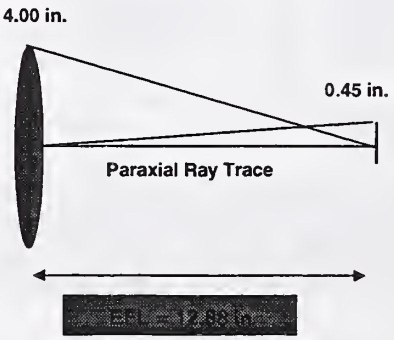
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]

Optical Design

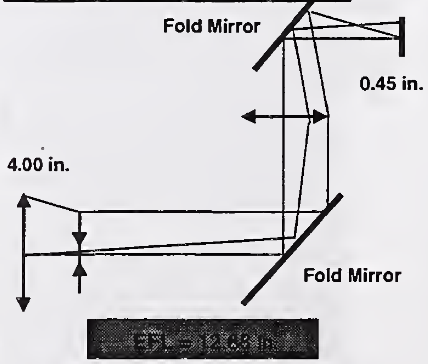
Thin Lens Design Example

[redacted]

Simple Solution:

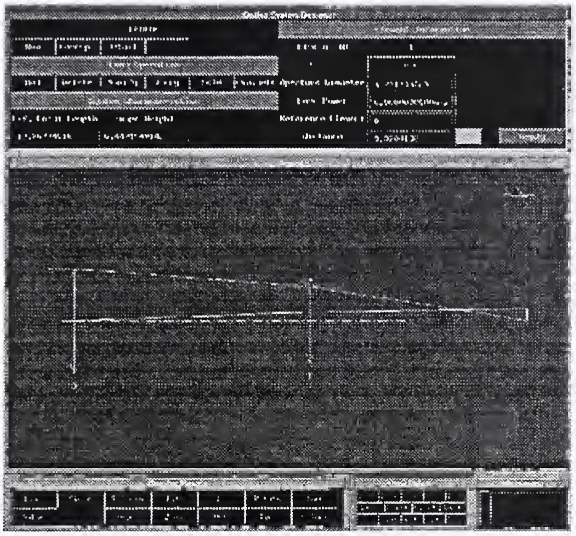


Realistic Package Solution:



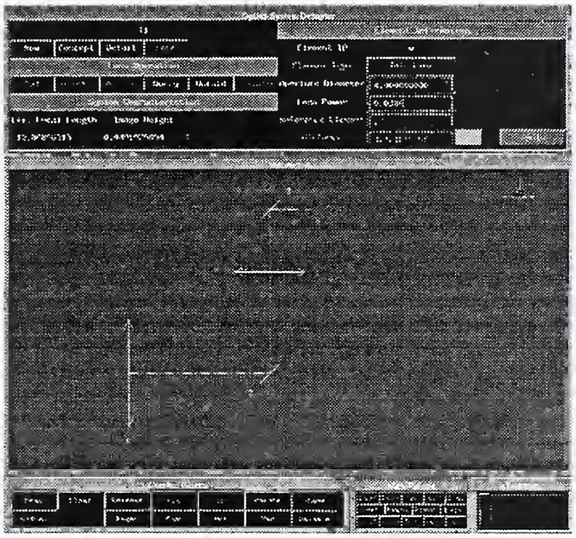
Optical Design

Thin Lens Example:



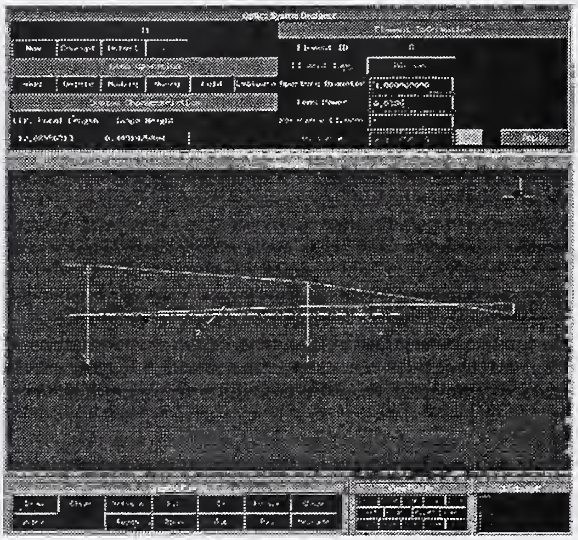
Optical Design

Thin Lens Folded:



Optical Design

Thin Lens, Suppressed Fold:

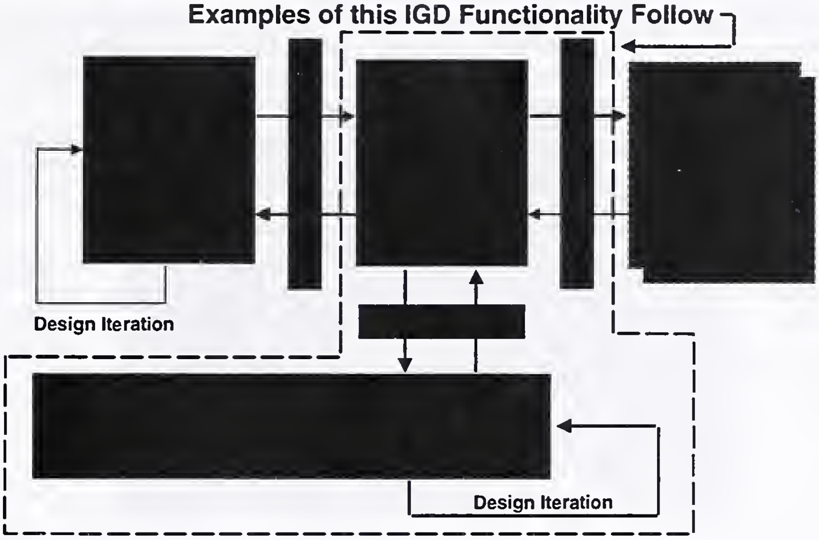


Optical Design

Thin Lens Design: AML - Optics Object Tree

Optical Design

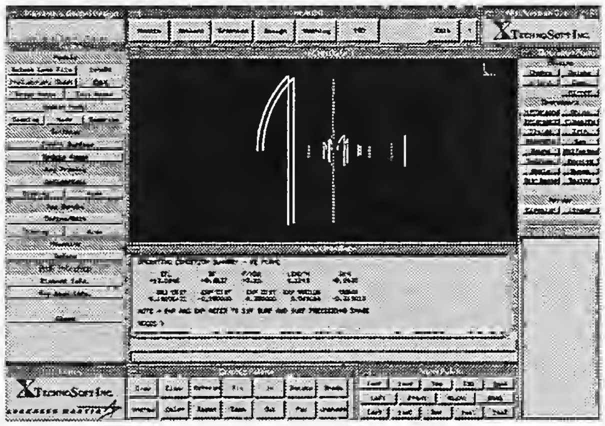
Detailed Lens Design



Optical Design

AML ACCOS-V Optical Design Scenario:

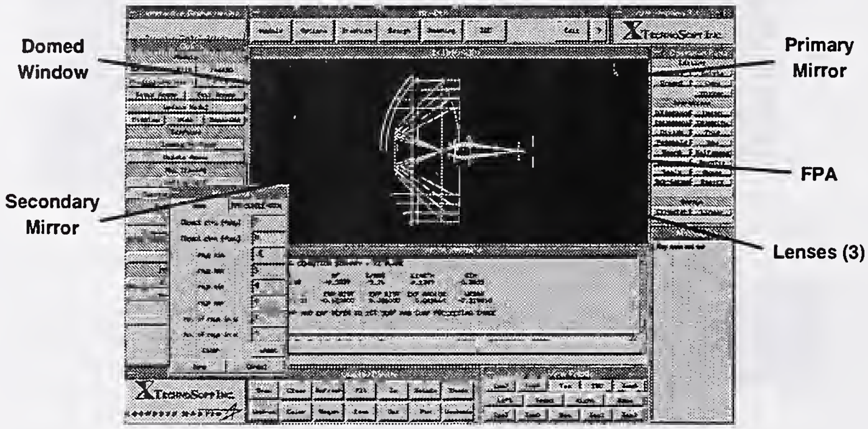
AML illustrates an ACCOS-V LENO (legacy file being read in and the resulting graphical display confirming this. Note GUI for AML optics, ACCOS-V, and various functional capability.



Optical Design

AML ACCOS-V Optical Design Scenario:

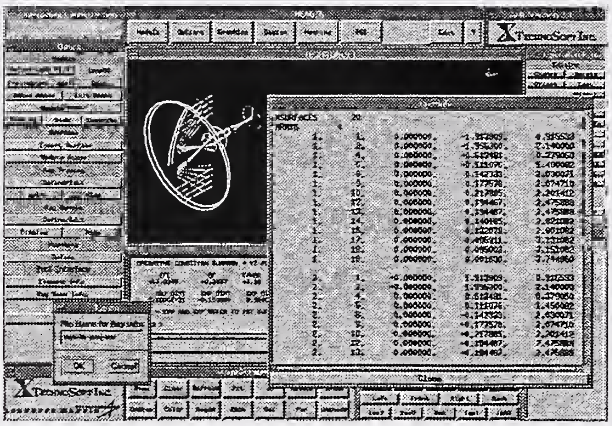
An ACCOS-V command has been issued in AML that evaluates the optical prescription in ACCOS-V and then displays the results in AML. A ray bundle has also been traced through all surfaces automatically in ACCOS-V and displayed in AML.



Optical Design

AML ACCOS-V Optical Design Scenario:

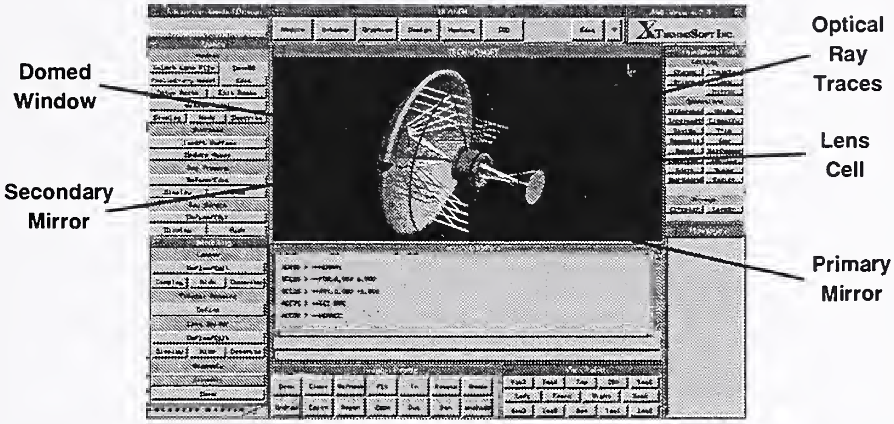
AML provides detailed information about the ray bundle. This data, along with the prescription data, is used to transfer the optical design directly into Pro/E.



Optical Design

AML ACCOS-V Optical Design Scenario:

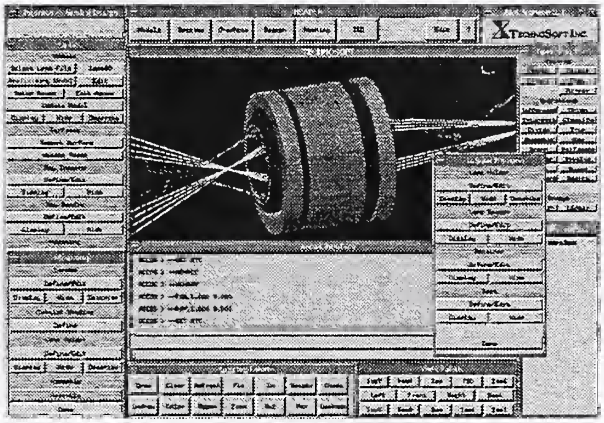
The AML ACCOS-V optical system illustrates a 3-D, dynamically rotatable view of the design. A lens cell has also been generated by the system.



Optical Design

AML ACCOS-V Optical Design Scenario:

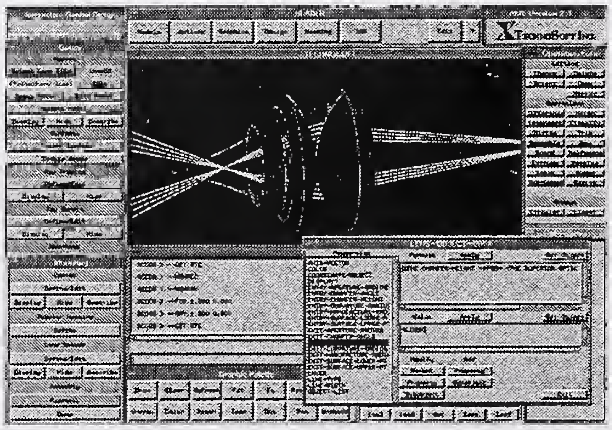
Close-up of the lens area illustrates the lens cell. Note the functionality offered in AML allows for a lens spacer, retainer, and seats. At any time, additional ray traces can be served from ACCOS-V to check for vignetting.



Optical Design

AML ACCOS-V Optical Design Scenario:

With the lens cell removed, a close-up illustrates the three lens. Adjustments to manufacturing features can be made at any time.



Optical Design

Qualitative Metric: Assessment of Detailed Optical Design

Six senior optical designers were presented an overview and demo of the AML ACCOS-V detailed optical design capability (as illustrated here). The following table summarizes their projected impact of the AML ACCOS-V detailed optical design sub-system of the IGD. A fully integrated IGD System of the multi-disciplinary gimbal design process was not assessed by this group.

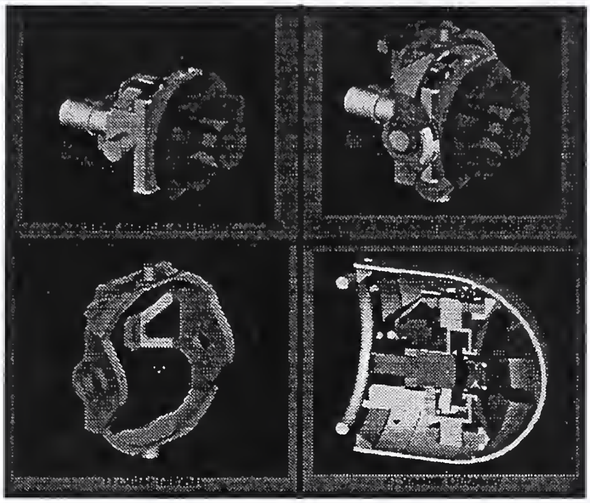
[illegible] Optical Designer	Years of Experience
Max Amon	33
Al Lyon	20
Gary Wiese	22
George Bradley	25
Ed Poppeliers	27
Robert Murphy	19

Projected Impact for Detailed Optical Design[1]	%
% Improvement in Design Time	20%
% Improvement in Design Capability	25%
% Improvement in Analysis Time	50%
% of Analysis that can be done more rigorously	50%

[1] These projections are for detailed optical design, not conceptual design.

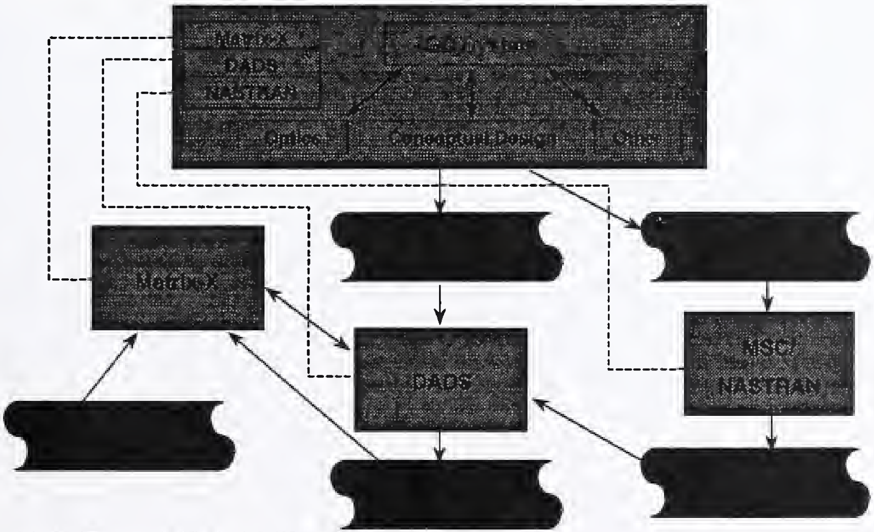
Mechanical Design

Gimbal Test Bed System: Used for Proof-of-Concept Modeling



Mechanical Design

Flow of data from IGD showing DADS ⟷ Matrix-X Simulation



----- = Interactive control of application from IGD

IMPACT

Applicability of the RaDEO-IGD KBS :

- The RaDEO-IGD system is primarily a preliminary design tool with ties to manufacturing. 80% of a missile system costs are committed in the first 20% of the program. The IGD system is a “direct hit” at this early-on design phase. The IGD system integrates final design software tools thus allowing for an efficient transfer of effort from preliminary design to detailed design.

- The Seeker/Gimbal component of a missile system contribute to 60-70% of the cost of a missile system. The RaDEO-IGD System is a “direct hit” to attack this high cost area.

- While the DARPA RaDEO program in general and the IGD system in particular does not focus directly on cost (10 fold increase in design space), AML class-objects have been developed to predict manufacturing costs. This capability is integral to AML and can be readily applied to the missile systems.

- The RaDEO-IGD System and associated gimbal sub-components database is directly applicable to missile systems.

Appendix F - Presentation Slides

Rule-Based Interoperability of Heterogeneous Systems

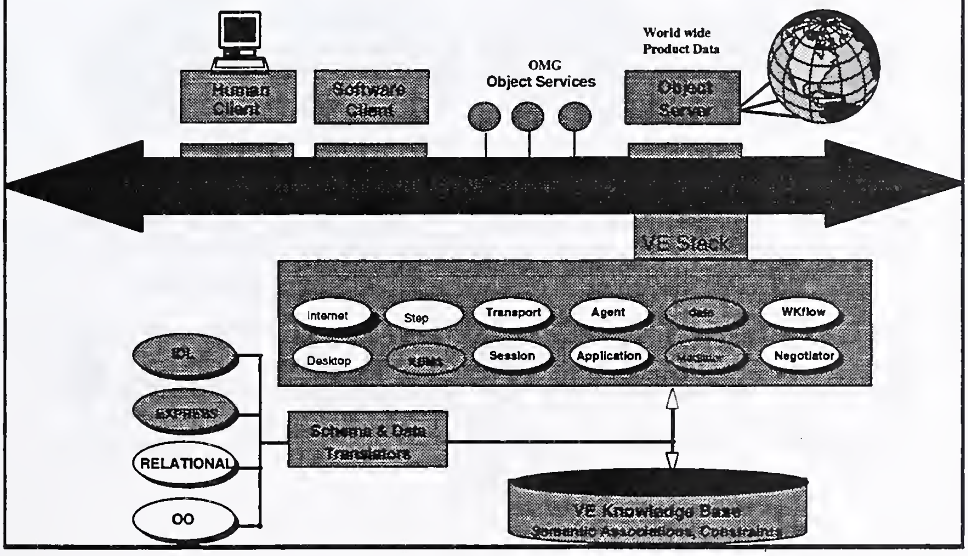
Stanley Su
University of Florida

Rule-based Interoperability of Heterogeneous Systems in the NIIIP Project

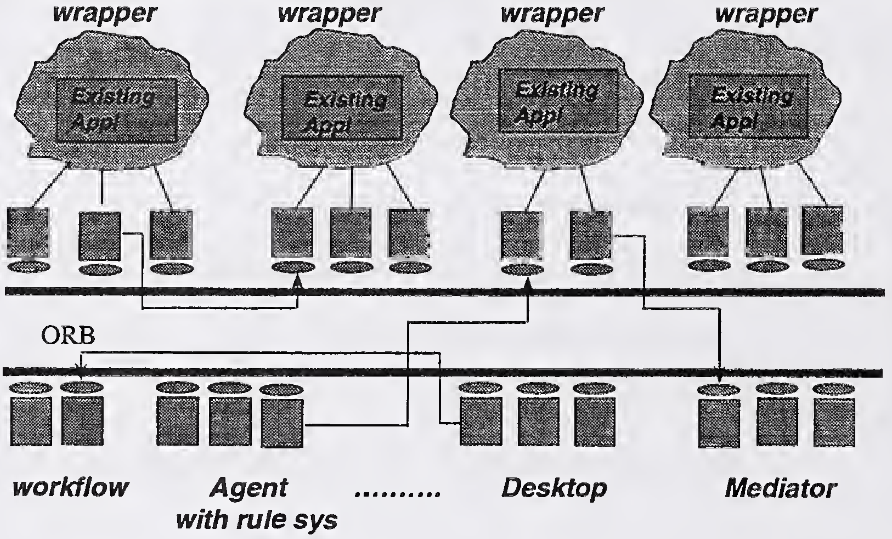
Outline

- *NIIIP Reference Architecture*
- *Inter-operation of Distributed Objects*
- *Event-Condition-Action-Alternative-Action Rules*
- *Events and Event Model in CORBA*
- *Rules Triggered by Events*
- *Event Server and Rule Server implementation Approaches*

NIIIP Reference Architecture



Inter-operation of Distributed Objects by Method Invocations



wrapper

wrapper

wrapper

wrapper

Existing Appl

Existing Appl

Existing Appl

Existing Appl

ORB

workflow

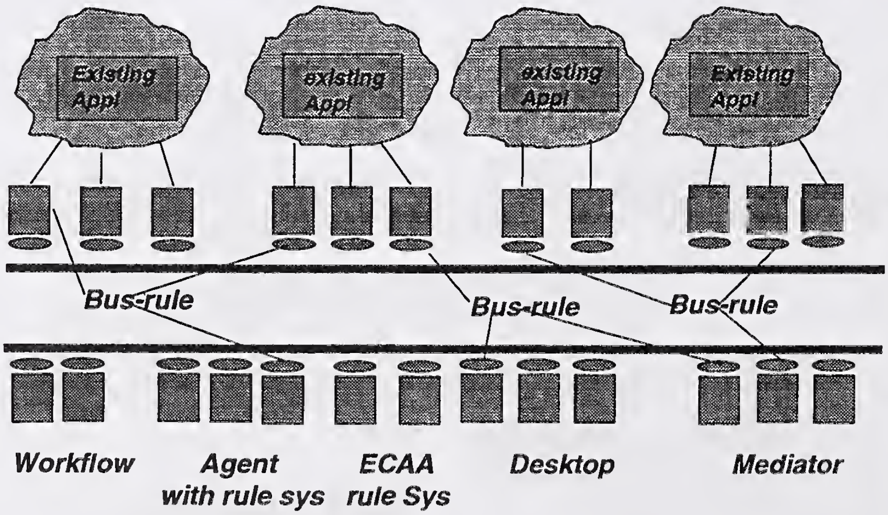
Agent with rule sys

....

Desktop

Mediator

Inter-operation of Distributed Objects Defined in ECAA Rules



Existing Appl

existing Appl

existing Appl

Existing Appl

Bus-rule

Bus-rule

Bus-rule

Workflow

Agent with rule sys

ECAA rule Sys

Desktop

Mediator

Examples of ECAA Rules for Data and Data Access Constraints

- Security rule:
  Event: Before retrieving data from a file
  Condition: If the user's access privilege is not sufficient
  Action: Reject the access request and initiate actions to report and handle the security break
- Requirement rule:
  Event: After an order has been placed
  Condition: If QuantityOrdered < OrderedProduct.StockLevel
  AltAction: Abort transaction and notify.
- Computation rule:
  Event: After an order has been placed
  Action: OrderedProduct.StockLevel = OrderedProduct.StockLevel - QuantityOrdered
- Cascaded Update/Delete Rule
  Event: Update part or supplier's identification number
  Action: Update the corresponding number in the Part-Supplier instances

---

Examples of ECAA Rules for Controlling Distributed Objects

- Event MES foreman determines that machine is down
  Condition: If machine type is X or Y and an Agent for repair is available
  Action: Call the agent
  AltAction: MES ships a work location status reference object to MS/FS & CA

- Event: The engineering change status has been changed to "initiated"
  Condition: If it is a part used in a government-supported project
  Action: Initiate a "government-EC" workflow
  AltAction: Initiate the regular EC workflow

- Event: MS/FS received work location status reference object from MES
  Action: MS/FS issues a re-plan; MS/FS triggers a manufacturing order to ERP;

- Event: Before MEX issues a re-plan
  Condition: A planning assistance agent is available
  Action: Call the agent to generate data for re-plan

Some Properties of an ECAA Rule

Rule <rule_id>
Triggered <coupling mode> <method_call >| <post_event>
Priority <integer>
Condition <guarded_expression>
Action <operations>
Otherwise <operations>

Extensions:

- ***An event can trigger a structure of rules***
- ***A rule or rule group can be dynamically activated or deactivated***
- ***Static rules are pre-compiled and dynamic rules are interpreted***
- ***Events can be posted synchronously or asynchronously***

Object Model

Object Specification in EXPRESS

```
DEFINE ENTITY entity_id
  SUPERTYPE OF (supertype_expression)
  SUBTYPE OF (subtype list)
  attr_id: [OPTIONAL] base_type
  .
  .
  DERIVE
  .
  INVERSE
  .
  UNIQUE
  .
  WHERE rule_label: rule expression
  .
END_DEFINE;
```

Method Specification in IDL

```
METHODS:
 EXCEPTION exception_id (var : type;..)
 .
 .
METHOD [ONEWAY] method_id
  ([IN | OUT | INOUT] para_id: para_type; ...)
  [RAISES (exception_id, ...)]
END_METHODS;
```

Rule Specification in UF's Rule Language

```
RULES:
RULE rule_id
 [TRIGGERED triggered_time trigger_operation, ...]
 [PRIORITY integer]
 [CONDITION guarded_expression]
 [ACTION statement_list]
 [OTHERWISE statement_list]
END_RULES;
```

Why Rules?

- Implementation of a method should perform only its intended function.
- Semantics of the pre-condition and post-condition should not be embedded in the implementation code of the method.
- Instead, they should be explicitly specified as ECAA rules to augment the interface definition of the class.
- Change in business rules affect only the ECAA rules, not the method implementation.
- High-level declarative rules are easier to understand than program code.

What can we do with these ECAA rules?

- ***Generate in-line code for some existing applications (e.g., embedding rule code into implemented Java program in the CIIMPLEX project)***
- ***Generate program bindings for linking to some existing application systems or agents (e.g., CORBA-bindings for rule-based interoperability demonstrated in the NIIIP project)***
- ***Enforced by a centralized or distributed rule processing system at run-time***

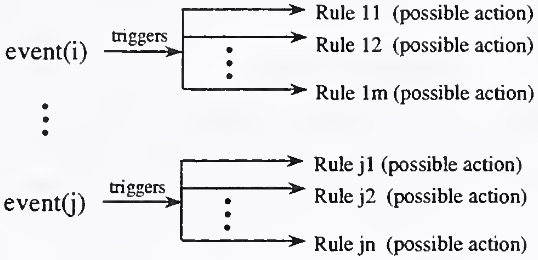
Triggering Events

Meaningful Triggering Events in CORBA:

- Method invocation
- Explicitly posted COSS events

Event Model

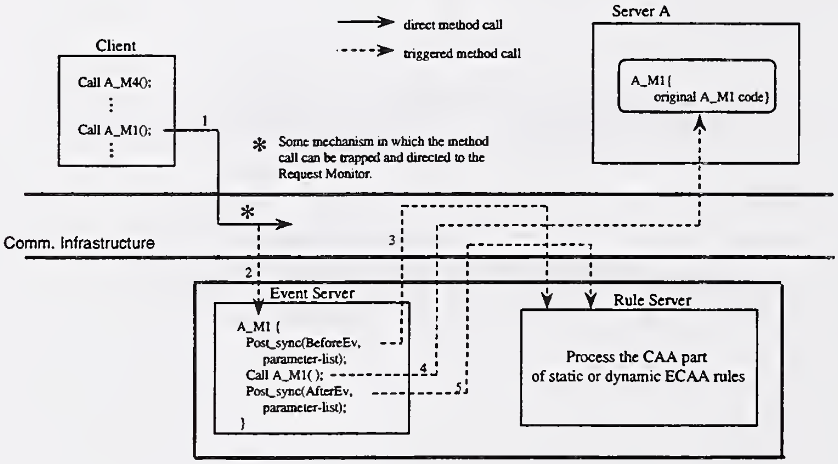
ECAA Rules Triggered by Events



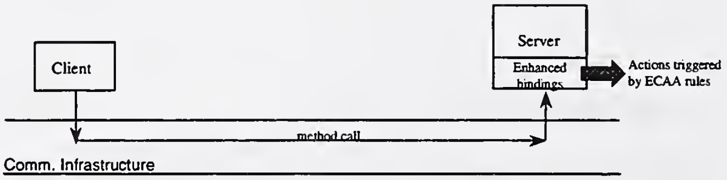
Implementation Approaches

- Centralized event and rule servers
- Distribute event monitoring and rule code to distributed objects by enhanced binding
- Distributed event and rule servers in each computing system

Centralized Event Sever and Rule Server



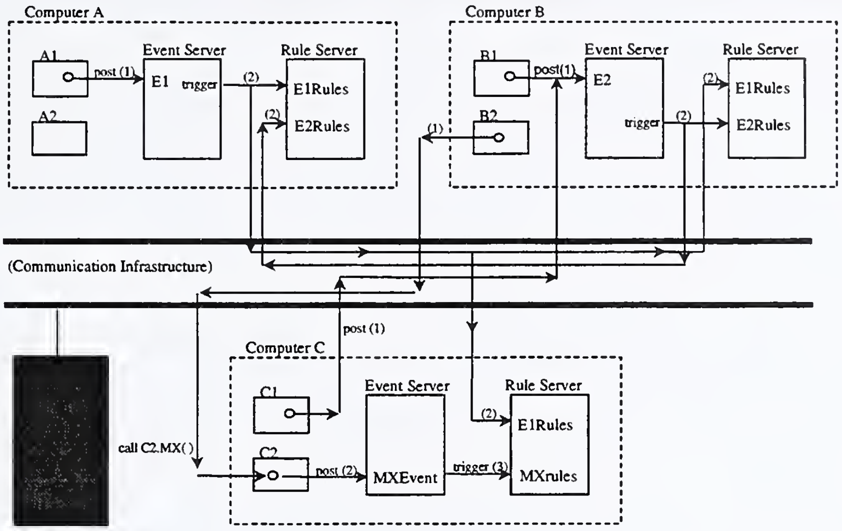
Enhanced Bindings for Distributed Objects



- Enhanced bindings for compiled and distributed event monitoring and rule processing.
- Enhanced bindings =

  "normal" IDL binding + generated code for ***event monitoring***
  and ***rule processing***

Distributed Event & Rule Servers In Computing Systems



Conclusion

Value-added Information Infrastructure for Integrating Heterogeneous Systems

- Rule-based Interoperability: Extension of OMG/ORB
- Semantics-rich Object Model: Extension of STEP/EXPRESS and OMG/IDL
- Active Integrated Information System

Appendix G - Presentation Slides

Configurator Synchronization

Bruce Ambler
Lucent Technologies

Interoperability with Knowledge Based Systems

- Product Configurators
  - Purpose - to design telephone switching equipment to meet customer requirements.
  - Requirements examples - traffic, features
  - Output - part numbers, quantities, location assignments.

Two Stages of Configuration

- Sales - Creates a design down to the level of specificity which allows a product to be priced, contracted, and ordered.
- Factory - Uses the output of the Sales Configurator and further reduces the solution to parts that can be manufactured/assembled

Configurator Technology

- Rules - If Then Else, Ratios, Exclusions, etc.
- Constraints - Balances resources required and resources offered by components.

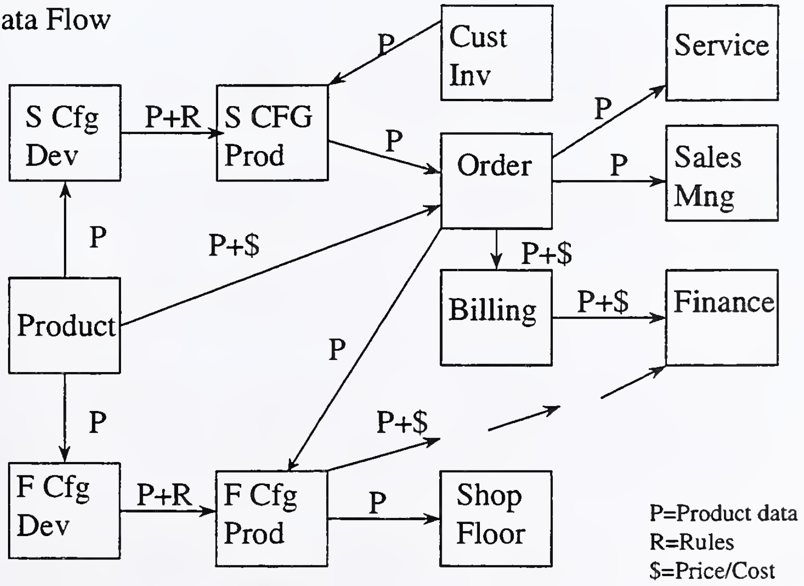
Interoperability - Input

- Input to Configurators from Other Systems
  - Sales Configurator
    - Product/Part
    - Customer Inventory
  - Factory Configurator
    - Product/Part
    - Sales Configurator Output

Interoperability - Output

- Sales Configurator
  - Order then Factory Configurator
  - Services
  - Sales Management
  - Financial - Revenue
- Factory Configurator
  - Shop floor
  - Financial - Cost accounting

Configurator
Data Flow



Coordination Problems

- Sales Configurator outputs product codes not recognized by order.
- Sales Configurator outputs product codes not recognized by Factory Configurator.
- Factory Configurator outputs product codes not recognized on shop floor.
- Financial can’t match Sales-Revenue view of product with Factory-Cost view

Interoperability Issues

- Knowledge Based systems use logic (rules, constraints) to select or produce data.
- Data needs to be coordinated across systems that share it.
- With Knowledge Based systems sharing from a common data server is not a solution.
- Rules and data need to be managed together

Appendix H - Presentation Slides

OKBC: A Programming Foundation for Knowledge-based Interoperability

Vinay Chaudhri
SRI International

AI Center

OKBC: A Programmatic Foundation for Knowledge Base Interoperability

Vinay K. Chaudhri
Adam Farquhar
Richard Fikes
Peter D. Karp
James P. Rice

Artificial Intelligence Center
SRI International

Knowledge Systems Laboratory
Stanford University

AI Center

Motivation

- DARPA's High Performance Knowledge Bases Program has three knowledge servers and about 20 technology developers.
  - Loom (Information Sciences Institute)
  - Ontolingua (Knowledge Systems Laboratory, Stanford)
  - CYC (Cycorp)
- Some technology developers are required to work with multiple knowledge servers.
- Some technology developers will be required to change affiliation at the end of first year.

AI Center

Approach

- Three knowledge servers will support OKBC as their API
  - Supporting OKBC for a system means defining mappings from OKBC to the native API of that system
- Technology developers will write their applications using OKBC
  - KB browsers and editors
  - Theorem provers
  - Knowledge acquisition tools

AI Center

Outline

- Design Approach
- Overview of OKBC
- Some Design Difficulties
- OKBC Compliance
- Implementation
- Future Plans

AI Center

Design Approach

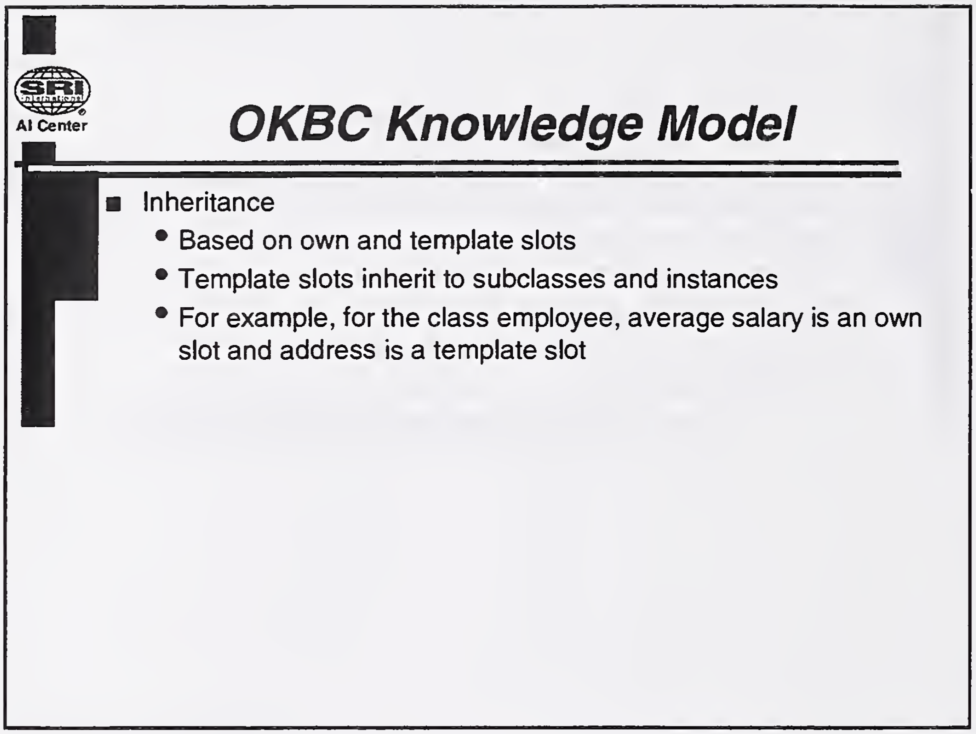
- Analyze a few knowledge representation systems (KRSs), and identify a common denominator
  - classes, individuals, ...
  - add, remove, replace, ...
- Represent the differences using behaviors
  - Some KRSs support frame names and others do not

OKBC = Knowledge Model + Operations + Behaviors

AI Center

OKBC Knowledge Model

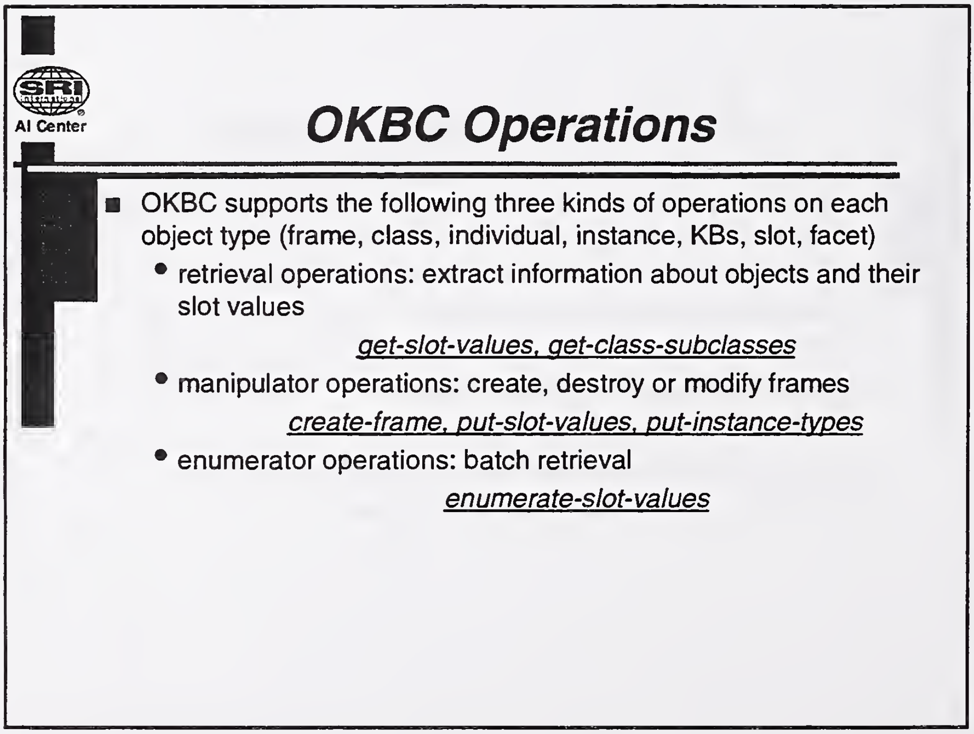
- Frames. Represent entities in the world (concrete or abstract). Are organized into a taxonomy of classes and individuals
  - Person, Organization
- Slots. Represent binary relationships
  - Age, Name
- Facets. Represent properties of slots
  - Value Type of Age is Integer
- Subclass, Instance-of, Slot-of and Facet-of relationships
- Knowledge Base. Is a collection of frames



AI Center

OKBC Knowledge Model

- Inheritance
  - Based on own and template slots
  - Template slots inherit to subclasses and instances
  - For example, for the class employee, average salary is an own slot and address is a template slot



AI Center

OKBC Operations

- OKBC supports the following three kinds of operations on each object type (frame, class, individual, instance, KBs, slot, facet)
  - retrieval operations: extract information about objects and their slot values

    get-slot-values, get-class-subclasses
  - manipulator operations: create, destroy or modify frames

    create-frame, put-slot-values, put-instance-types
  - enumerator operations: batch retrieval

    enumerate-slot-values

OKBC Operations

- Tell/ask interface
  - OKBC defines a restricted assertion language
    - `Classes, slots and facets are relation symbols`
    - `subclass-of, instance-of, slot-of, facet-of`
    - `class, individual`
  - But, tell can accept any sentences that are tellable
  - Provides unlimited extensibility

OKBC Behaviors

- Behaviors allow KRSs to advertise how they are different
  - For example, support for frame names
  - Applications can query the value of the behavior and take appropriate actions
- Behaviors can also be used to control the KRS
  - For example, constraint checking
  - Applications can indicate the necessary functionality to the KRS

AI Center

Example use with LOOM

```
(defconcept  Healthy-Parent :is (:and Parent (:at-least 3 child)
                                             (:at-most 2 son)
                                             (:exactly 1 pet)))
```

```
(create-class Healthy-Parent Parent)
(create-slot child Healthy-Parent)
(create-slot son Healthy-Parent)
(create-slot pet Healthy-Parent)
(put-facet-value Healthy-Parent child :minimum-cardinality 3)
(put-facet-value Healthy-Parent son :maximum-cardinality 2)
(put-facet-value Healthy-Parent pet :cardinality 1)
```

Loom contexts are represented using OKBC KBs.

AI Center

OKBC Design Issues

- Assumptions about entities that are represented as frames
  - Semantics of frame operations
- Inference mechanisms
  - Control of inference
- Data types
  - Symbols, packages
- Deletion
  - Is not always thorough

AI Center

OKBC Design Issues
All Entities are not Frames

- Should get-kb-classes return classes that are not frames? e.g., (set-of 1 2 3)
  - Unreasonable to require that all classes be frames
  - Classes that are not frames are inherently different from the ones that are frames
  - It should be possible to pass the results of one OKBC operation to another

Decision:

get-kb-classes returns only those classes that are frames

AI Center

OKBC Design Issues
All Entity Types are not Frames

- Not all KRSs represent slots as frames
- If we execute get-kb-frames on such KRSs, we will get different results

Solution:

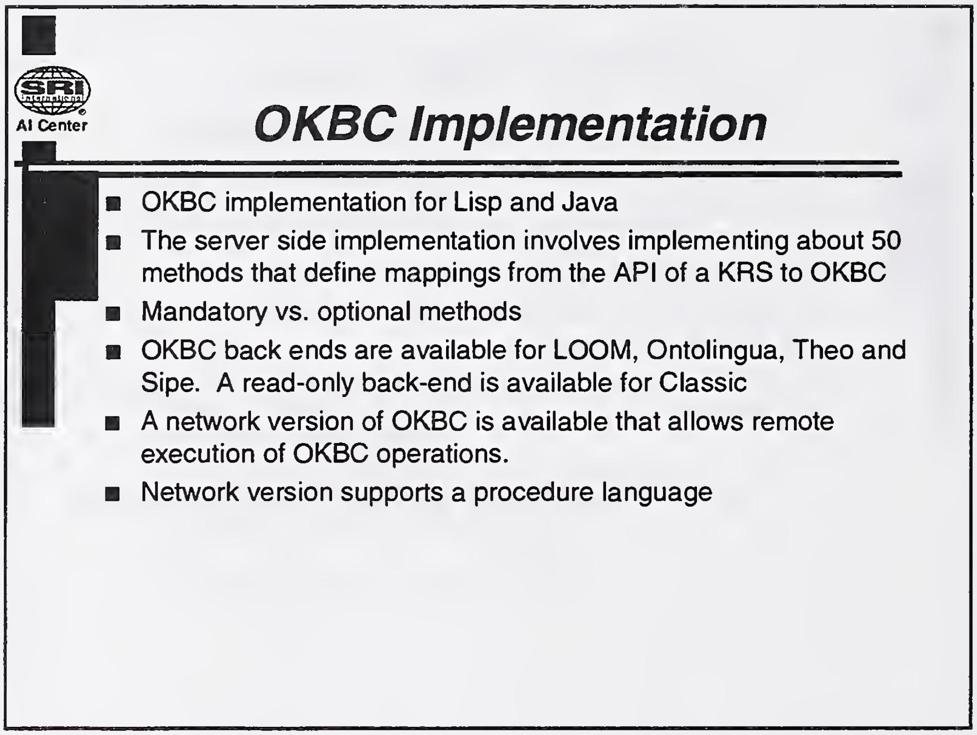
- Introduce a behavior called :are-frames with possible values of :class, :individual, :slot, and :facet

Desired Behavior

- For two KRSs with the same value of :are-frames behavior, get-kb-frames is guaranteed to return the same result

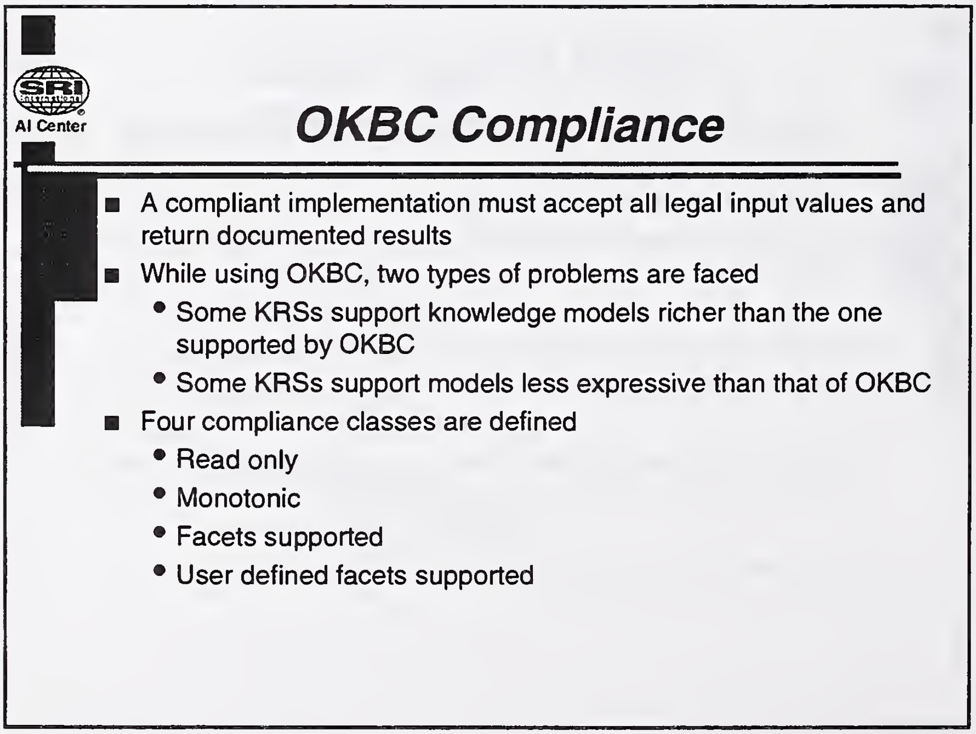
Achievable Behavior

- Applications are made aware of obvious differences



OKBC Implementation

- OKBC implementation for Lisp and Java
- The server side implementation involves implementing about 50 methods that define mappings from the API of a KRS to OKBC
- Mandatory vs. optional methods
- OKBC back ends are available for LOOM, Ontolingua, Theo and Sipe. A read-only back-end is available for Classic
- A network version of OKBC is available that allows remote execution of OKBC operations.
- Network version supports a procedure language



OKBC Compliance

- A compliant implementation must accept all legal input values and return documented results
- While using OKBC, two types of problems are faced
  - Some KRSs support knowledge models richer than the one supported by OKBC
  - Some KRSs support models less expressive than that of OKBC
- Four compliance classes are defined
  - Read only
  - Monotonic
  - Facets supported
  - User defined facets supported

AI Center

Current Work

- OKBC/CORBA implementation
- Expand knowledge model
  - Assertions
  - Justification/Explanation
  - Contexts
  - Probabilities

AI Center

References

- http://www.ai.sri.com/~okbc

Appendix I - Presentation Slides

Knowledge Source Awareness Models for Interoperable Knowledge-based Systems

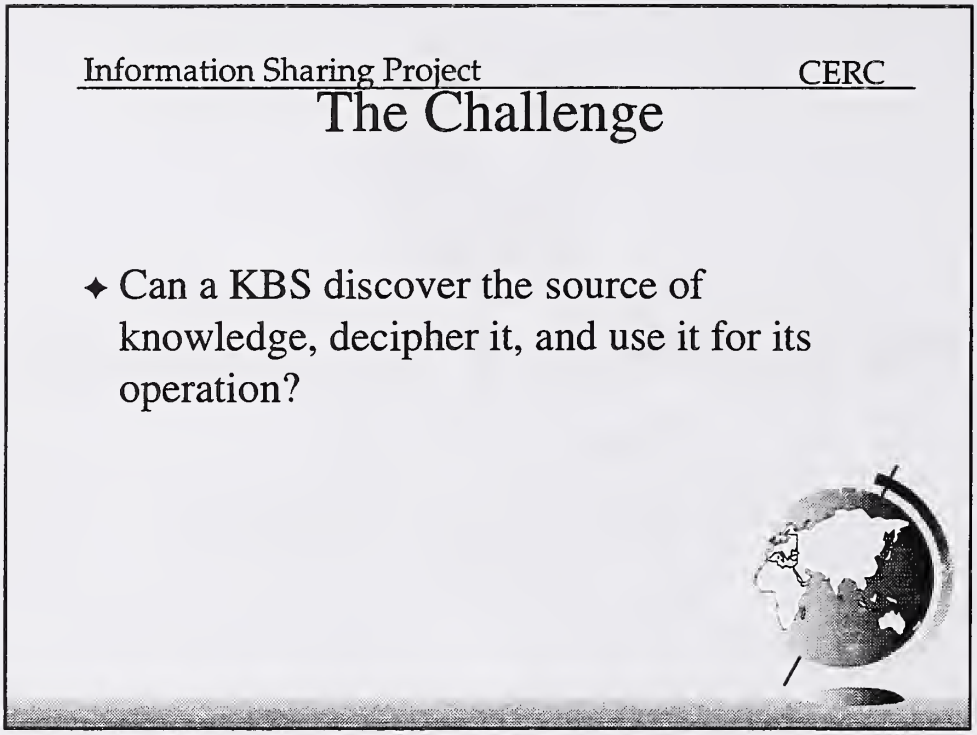
Ramana Reddy
West Virginia University

Knowledge/Source Awareness Models for Inter-Operable KBS's

Ramana Reddy
Concurrent Engg. Research Center

Premises

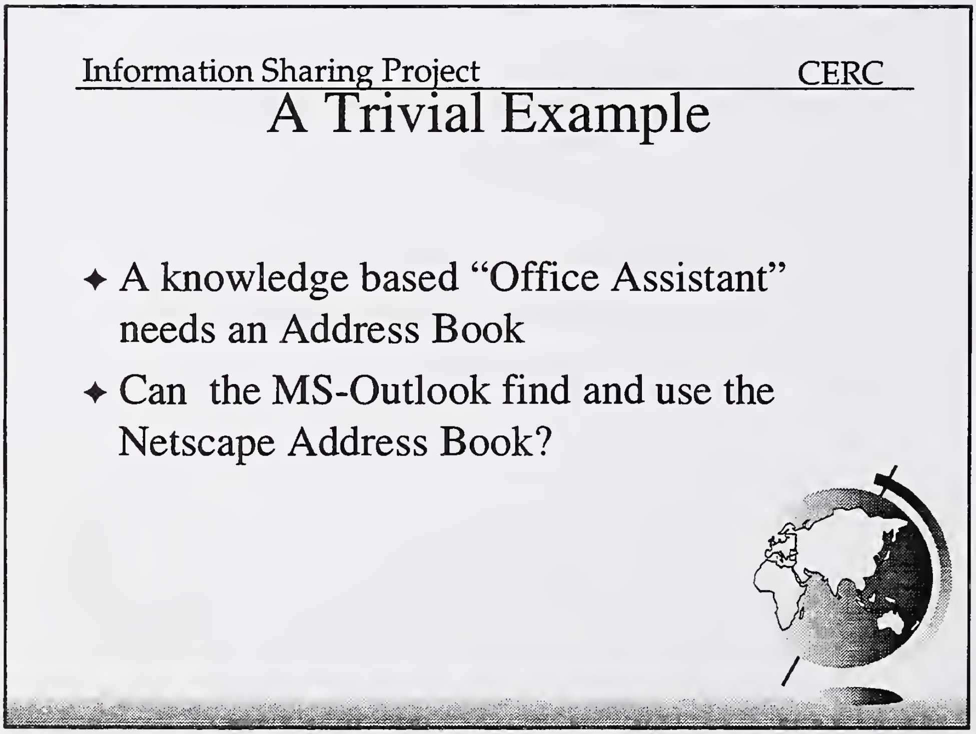
- KBS's depend on knowledge sources
- Knowledge exists in differing formats
- Knowledge is distributed



Information Sharing Project CERC

The Challenge

- Can a KBS discover the source of knowledge, decipher it, and use it for its operation?



Information Sharing Project CERC

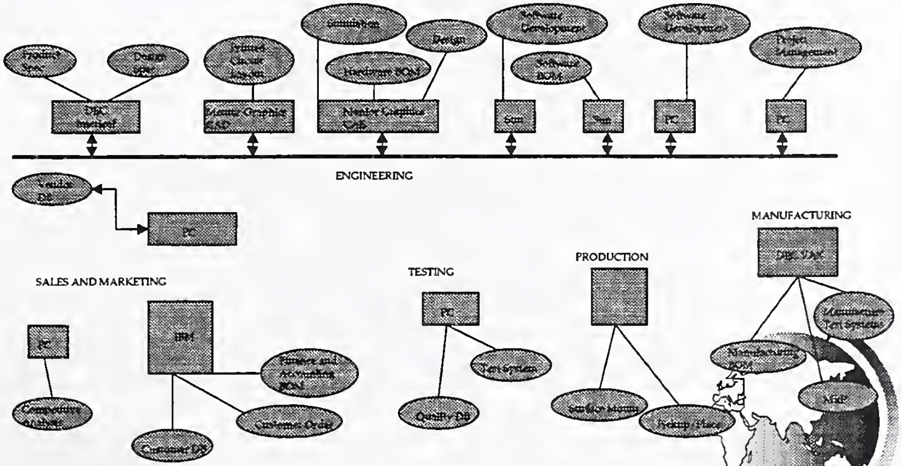
A Trivial Example

- A knowledge based “Office Assistant” needs an Address Book
- Can the MS-Outlook find and use the Netscape Address Book?

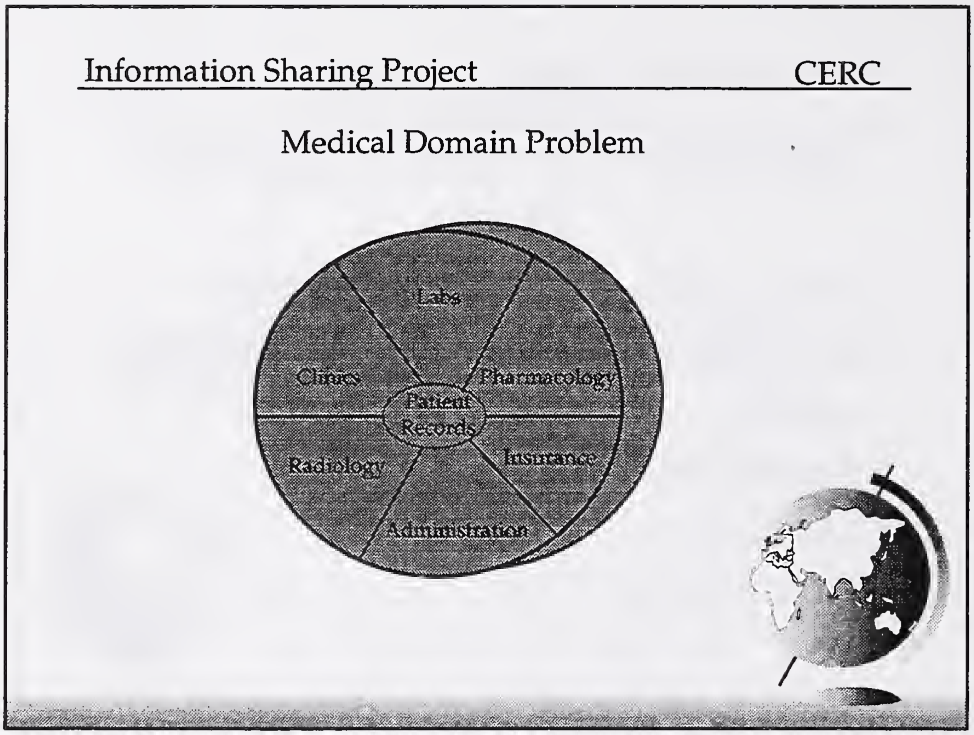
Imperatives for Inter-Operable KBS's

- ✦ Reasoning models for KS discovery
- ✦ A taxonomy for domain-specific knowledge sources
- ✦ Transformers (n+m problem)
- ✦ Learning
- ✦ Pliant systems

Goals of Information Sharing

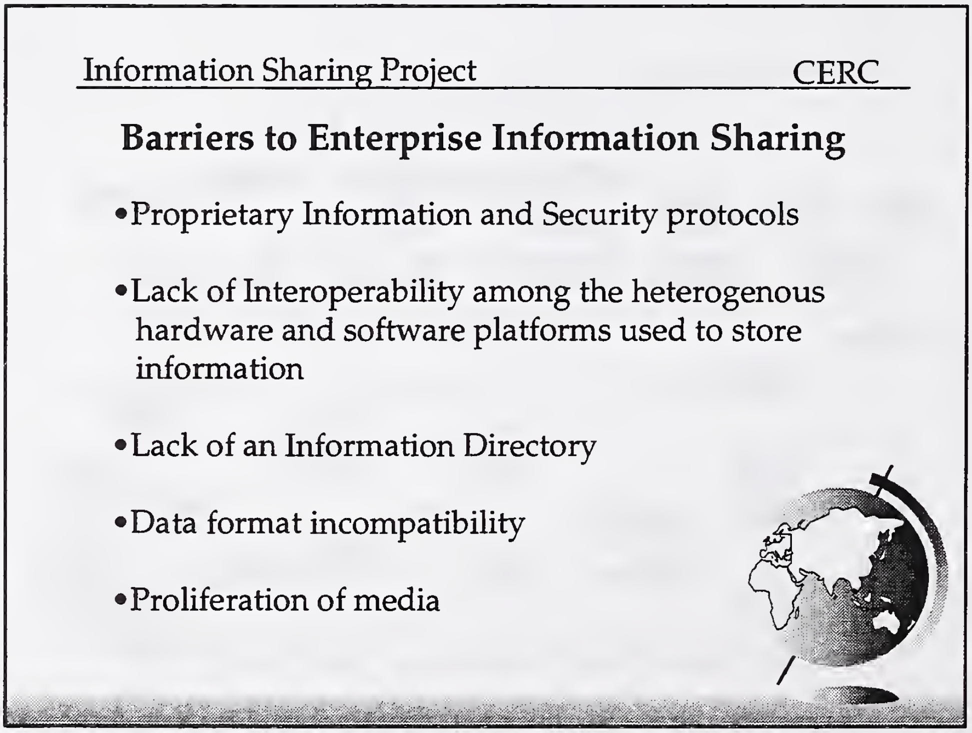


Provide easy, uniform, secure and transparent ways to share geographically distributed, heterogeneous information in support of organizational processes and policies.



Information Sharing Project CERC

Medical Domain Problem

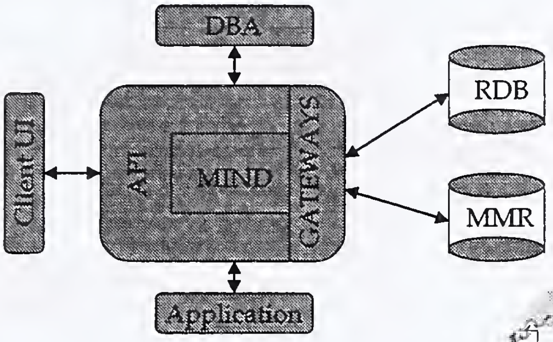


Information Sharing Project CERC

Barriers to Enterprise Information Sharing

- Proprietary Information and Security protocols
- Lack of Interoperability among the heterogenous hardware and software platforms used to store information
- Lack of an Information Directory
- Data format incompatibility
- Proliferation of media

CERC ISS Approach



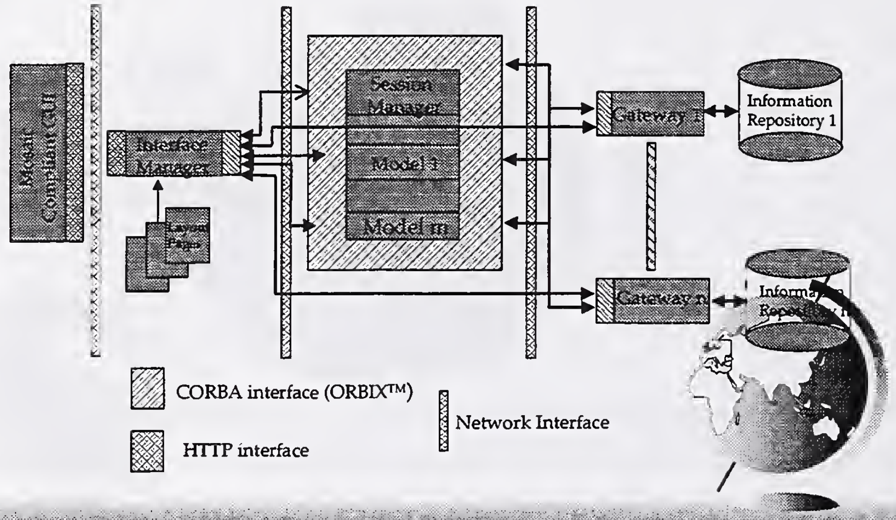
An enterprise model provides an integrated view of information in remote data repositories.

ISS (Version 3.0) Features

- Adoption of "HTTP" protocol for client-end interoperability.
- Adoption of "CORBA" specifications for server-end interoperability using Orbix™.
- Gateways to Commercial relational database (Oracle™).
- Adoption of "Kerberos" for authentication.
- Model-based Wide-area access and update capabilities to structured and unstructured information.
- Federated access control mechanisms (Information provider decides who can access information).
- Adoption of Hyper-text document metaphor (Mosaic) to support ease-of-use.

Information Sharing Project CERC

ISS Version 3.0 Modules



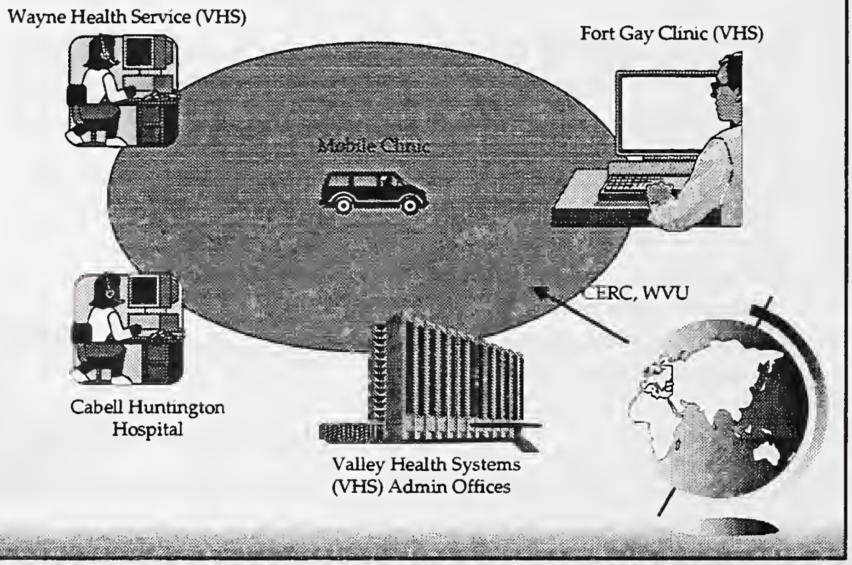
▨ CORBA interface (ORBIX™)

▩ HTTP interface

▨ Network Interface

Information Sharing Project CERC

ARTEMIS Project


Wayne Health Service (VHS)

Fort Gay Clinic (VHS)

Mobile Clinic

CERC, WVU

Cabell Huntington
Hospital

Valley Health Systems
(VHS) Admin Offices