ASME Engineering Information Mgt. Symposium (EIMS'97)
at DETC'97, Sep 14-17, 1997, Sacramento.

Impact of Data Modeling and Database Implementation Methods on the Optimization of Conceptual Aircraft Design

Neil S. Hall
Research Engineer
Atlanta Electronic Commerce Resource Center
Georgia Institute of Technology
Atlanta, Georgia

Robert E. Fulton
Professor
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia

ABSTRACT

Advances in the aircraft technologies have resulted in an increase in the amount of data required to define a design during the conceptual stages. A conceptual design dictates a close multidisciplinary effort requiring large amounts of data exchange. In order to optimize the design process, it is crucial that a top-down data management design structure be in place in the early phases of the design. This structure will provide consistency in data format and allow ease of data exchange between the various disciplines involved in the design process. In the conceptual design phase, consideration must be given to the changing structure of the of the database as the product design evolves. Current database design approaches are typically limited to the detailed design phase where the data organization is fixed.

The goal of the research was to develop a database design approach to support the conceptual design of complex engineering products where the database organization is evolving. The research investigates the relative merits of a relational and object-oriented approach to database design for a multidisciplinary aircraft design effort. On a conceptual level, complete database design methodologies have been developed that include all disciplinary data required in the conceptual design phase. The relational and object-oriented design methodologies were applied directly to the stability and control section of the design. This research documents these proposed approaches and recommends possible database design strategies.

NOMENCLATURE

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CE	Concurrent Engineering
EXPRESS	An object-flavored information model specification language
FAR 25	Federal Aviation Regulation 25
HPC	High Performance Computing
HSCT	High Speed Civilian Transport
ICAM	Integrated Computer Aided Manufacturing Program
IDEF	ICAM Definition
IDEF0	ICAM definition used to produce a function model that is a structured representation of activities or functions and the relationships between those activities within a system.
IDEF1X	ICAM definition used to produce a data model that represents the information within the environment or system. IDEF1X is a design method for automated systems implementation of relational databases.
IGES	Initial Graphic Exchange Specification
IPPD	Integrated Product and Process Development
MDO	Multidisciplinary Design Optimization
MDT	Multidisciplinary Design Technology
OODBMS	Object-Oriented Database Management System
PDES	Product Data Exchange Using STEP
RDBMS	Relational Database Management System
RFP	Request for Proposal
SQL	Structured Query Language
STEP	Standard for the Exchange of Product Model Data
1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
BCNF	Boyce-Codd Normal Form
DKNF	Domain Key Normal Form

INTRODUCTION

New aerospace designs will incorporate new concepts as a result of advances made in the scientific and engineering technologies. These new concepts will afford the aircraft designer with an interesting and somewhat envious dilemma. The aircraft designer will have unprecedented flexibility in design concepts. However, this new flexibility will often be paralleled in ever increasing design complexity. Aircraft such as the High Speed Civil Transport (HSCT) will provide a design environment which will require the efficient use of new technologies in an arena which has historically proven to have stringent performance and cost goals which must be met in order to result in a successful design. The complexity of the HSCT design will dictate a close multidisciplinary effort requiring large amounts of data exchange. Moreover, with the enormous development costs associated with such a design, corporate teaming is essential. It is critical to the success of the HSCT and future aircraft design that a new approach be taken toward the management and exchange of information. A top-down data management design structure should be developed and implemented in the early stages in order to optimize the design process.

THE DESIGN PROCESS

It is common in the design process for the aircraft designer/configurator to begin with a set of aircraft specifications defined by the customer. A study is made of various configurations which have the qualities which satisfy these specifications. As the designer/configurator nears completion of the design iteration, the design is chosen which first satisfies the major constraints which define the aircraft geometry such as overall span for airport gate access, cruising speed, passenger load, cargo capacity, etc.. Reliance must then be placed on the expertise of other disciplines in order to determine whether or not the configuration meets performance and cost goals. The exchange of data in this stage of the design could often be characterized as a "specific need" exchange. In order to calculate aircraft lift and drag, the aerodynamicist might request planform and cross-sectional geometric data. However, the structural engineer might want geometric data that defines crucial stress and load points such as the geometry that defines door and landing gear locations. The terminology of "specific need" is chosen because the designer/configurator typically provides each discipline with only that data which is required in performing the specific task of that discipline. A very common problem with this method of data exchange is data consistency. It is not uncommon to find that during the conceptual design phase a particular discipline's updated calculations have not been effectively communicated with other disciplines involved in the design effort. This breakdown in the data exchange process results in inconsistent predictions among the various disciplines and valuable design time is lost in the process of redefining a common basis for evaluation. Other problems with this approach are redundancy and the lack of a standard data format. It is quite common to find that the data exchanged between disciplines and supplied by the designer/configurator are often duplicated in a slightly different format for the various discipline's use. Moreover, each discipline is typically concerned with “its data requirements” only, and not much thought or concern is given as to how the data will be used by another discipline. Figure 1 shows the data management problem that currently exists in aircraft conceptual design. The figure is somewhat comical in the way in which it portrays each discipline involved in the conceptual design process. However, this representation is not far from reality.

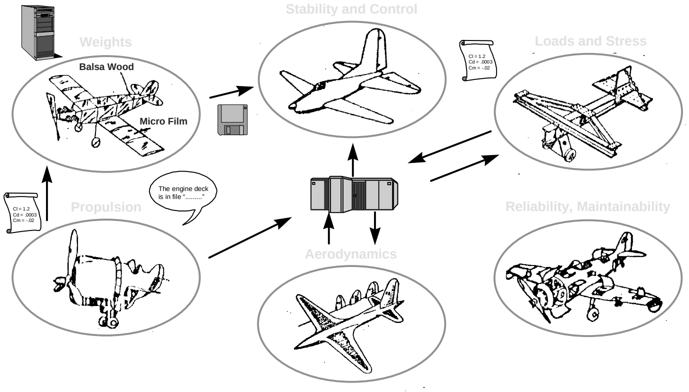


Figure 1. The data management problem.

Due to the complexity in design and the use of advanced technologies, the HSCT will require a multidisciplinary effort. Multidisciplinary Design Optimization (MDO), or Multidisciplinary Design Technology (MDT), will take advantage of the evolving High Performance Computing (HPC) environment and will be a critical component in the design of the HSCT. The concept of Integrated Product and Process Development (IPPD)/Concurrent Engineering (CE) as a means of improving the product development process is now becoming more critical. In order to ensure design success, it is crucial that a top-down data management design structure be in place in the early phases of the design. This structure will provide consistency in data format and allow ease of data exchange between the various disciplines involved in the design process.

DATA FLOW

Figure 2 diagrams a data flow structure that is logically centralized around a shared database and will serve as the model for use in development of proposed approaches and possible database design strategies for aircraft conceptual design data. This logically centralized database could also exist as a distributed database.

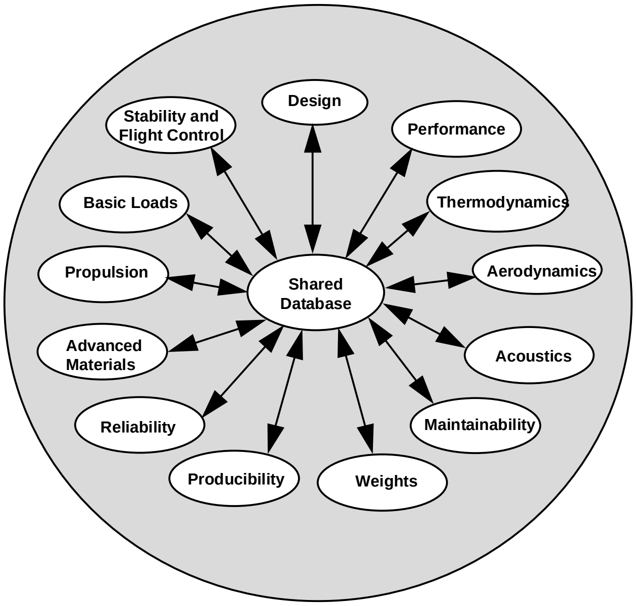


Figure 2. Data flow for conceptual design.

An important point to note from figure 2 is the inclusion of the disciplines of maintainability, reliability, and producibility. Traditionally, these disciplines have not been represented in the earlier phases of aircraft design (i.e. the conceptual stage). However, there has been an increased realization that while MDO presently addresses the integration of the traditional aerospace disciplines such as aerodynamics, propulsion, structures, and controls earlier in the design process, Concurrent Engineering (CE), which is concerned with the earlier integration of product life cycle phases such as manufacturing and support should be addressed in order to optimized the aircraft design process[2].

PROCESS MODELING

The distinction between conceptual design and preliminary design is sometimes fuzzy. However, for the purposes of this paper a distinction will be made in order to provide a better understanding of how the process model for the design of a HSCT was developed. For discussion, the term conceptual design refers to the development of *global* concepts. Global is used here to represent macro or "big picture" concepts. The conceptual design phase of aircraft is the process in which the outer moldlines of the aircraft are created with minimal internal systems and refinements. Preliminary design refers to the development of *specific* concepts. Specific is used to represent micro concepts, which are the concepts for the individual parts and systems leading toward final design. The beginning of the preliminary design phase includes the basic testing of "Will everything work? Will everything fit together? Will everything work together?". During the preliminary design phase, conceptual parts are properly placed within the moldlines of the aircraft. It is in this phase that the conceptual design is validated from more detailed perspective. These parts are further developed and refined in the final design phase. It is in this phase that detailed drawings are produced for the manufacturing of the aircraft systems and parts. The overall process for the HSCT design is represented by the IDEF0 model presented in figure 3.

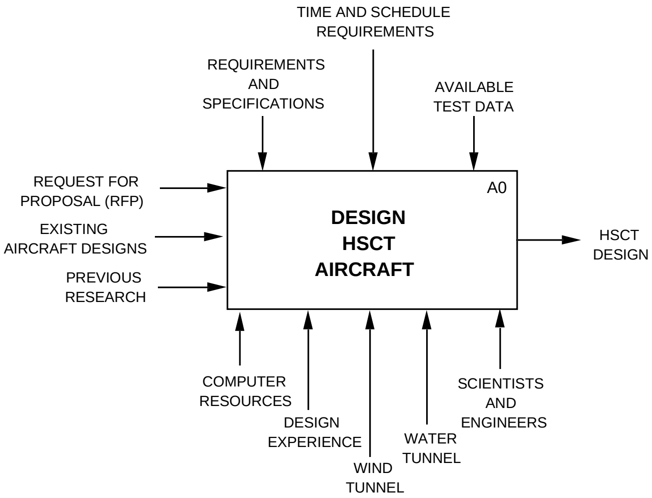


Figure 3. IDEF0 Diagram - Level 0.

This zero-level view shows that the design of the HSCT is limited by design requirements and specifications, time and schedule, and available test data. The design study is usually initiated by a request for proposal or (RFP). Figure 4 presents the level-one IDEF0 diagram which shows the process flow required in developing a HCST design up to the preliminary design phase. Each node of figure 4 was further broken down into blocks A1 - A5 for this research.

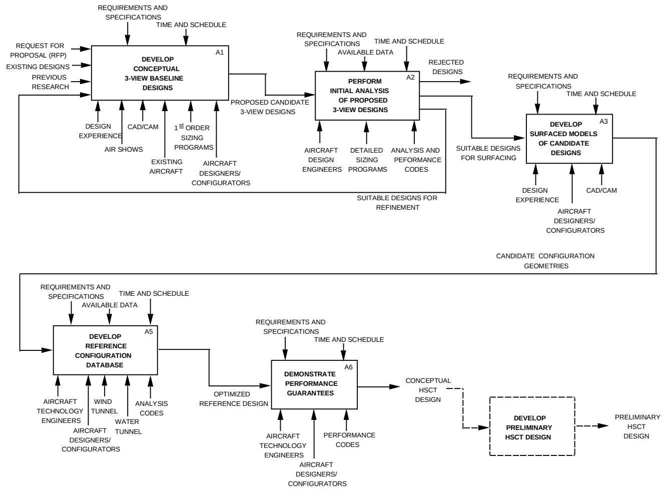


Figure 4. IDEF0 Diagram - Level 1.

RELATIONAL DATABASE DESIGN APPROACH

Data Relationships Modeling

The HSCT relational design data model includes the database schema and a data dictionary. The specific categories for the database design are as follows:

1. Aerodynamics
2. Aircraft Components
3. Cost
4. Materials
5. Performance
6. Stability and Control
7. Weights

The function model identifies a common process in order to ascertain what the data requirements are for the conceptual design process. Figure 5 shows examples of the types of data that are required during the aircraft conceptual design phase.

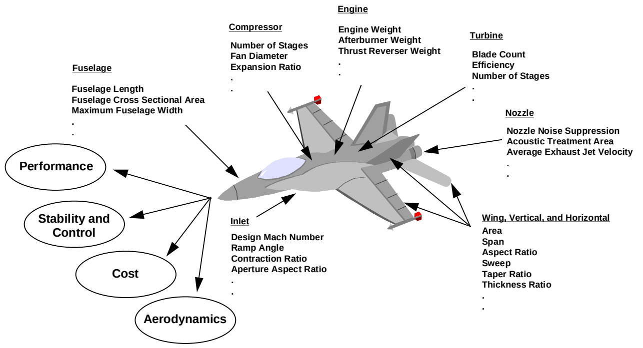


Figure 5. Example data required for aircraft conceptual design.

Database Schema

A complete database schema for the relational database design was developed. Examples for the engine and inlet are shown below.

ENGINE (COMPONENT_ID, ENG_LEN_TOT,
ENG_MAIN_FRAME, ENG_MAXWT_LEN,
ENG_REAR_FRAME, FAN_CASE_DIA,
FAN_FACE_LEN, MAX_NOZZLE_HEIGHT,
MAX_NOZZLE_WIDTH, NOZZLE_HT,
NOZZLE_INTERN_WIDTH, WT_ENGINES,
WT_PER_ENG, WT_ENG_INSTALL,
WT_START_SYS, WT_AFTERBURN,
WT_THRUST_REV, KPG_TOREN, KB_TOREN,
KEC_GD)

INLET (COMPONENT_ID, APER_AR_INLET,
AREA_RATIO_INLET, BLEED_AREA_INLET,
BYPASS_AREA_INLET, CAP_AREA_INLET,
CONTR_RATIO_INLET, CORR_AIRFLOW_INLET,
CORR_ECS_AIR_INLET, DESIGN_M_INLET,
FACE_RECOV_INLET, IN_LIP_ANG_INLET,
LEAK_AREA_INLET, PRESS_RECOV_INLET,
RAMP_ANG_FIN_INLET, RAMP_ANG_INIT_INLET,
SPILL_AREA_INLET, SUBSONIC_DIF_LD,
THROAT_M_INLET, WT_RAMP,
WT_SPIKE, WT_AIRINDUCT_SYS, KD_GD, KM_GD,
KD_TOREN, KR_GD, KS_GD)

Data Dictionary

A comprehensive data dictionary for the relational database design was developed and a total of 461 variables were defined. The data dictionary section describing the inlet is shown as an example.

INLET

DESIGN_M_INLET	Design Mach number
RAMP_ANG_INIT_INLET	Initial ramp angle (degs)
RAMP_ANG_FIN_INLET	Final ramp angle (degs)
IN_LIP_ANG_INLET	Internal lip angle (degs)
CONTR_RATIO_INLET	Contraction ratio
THROAT_M_INLET	Throat Mach number
APER_AR_INLET	Aperture aspect ratio (BL/WL)
CAP_AREA_INLET	Capture area inlet (sq ft)
PRESS_RECOV_INLET	Main inlet average pressure recovery
FACE_RECOV_INLET	Engine face recovery
CORR_AIRFLOW_INLET	Corrected engine airflow (lbm/sec)
CORR_ECS_AIR_INLET	Corrected environmental control system airflow (lbm/sec)
BLEED_AREA_INLET	Bleed (% capture area)
SPILL_AREA_INLET	Spillage (% capture area)
LEAK_AREA_INLET	Leakage (% capture area)

BYPASS_AREA_INLET	Bypass (% capture area)
SUBSONIC_DIF_LD	Subsonic diffuser L/D
AREA_RATIO_INLET	Area ratio (throat:face)
WT_RAMP	Weight of a variable geometry ramp (lbs)
WT_SPIKE	Weight of an inlet spike (lbs)
WT_AIRINDUCT_SYS	Weight of air induction system (lbs) (includes inlet ducts, ramps, spikes, and associated controls)

Air Induction Weight Estimation

KD_GD	Air induction factor for buried engine installation for a commercial transport (General Dynamics Method) = 1.33 for ducts with flat cross sections = 1.00 for ducts with curved cross sections
KM_GD	Air induction factor for buried engine installation for a commercial transport (General Dynamics Method) = 1.0 for M_D below 1.4 = 1.5 for M_D above 1.4
KD_TOREN	Air induction factor for buried engine installations (Torenbeek Method) = 1.00 for ducts with curved cross sections = 1.33 for ducts with flat cross sections
KR_GD	Ramp factor (General Dynamics Method) = 1.0 for M_D below 3.0 = $(M_D + 2)/5$ for M_D above 3.0
KS_GD	Inlet spike constant (General Dynamics Method) = 12.53 for half round fixed spikes = 15.65 for full round translating spikes = 51.80 for translating and expanding spikes

Normal Forms in Relational Design

In order to avoid data redundancy in the relation design, relational tables are further normalized beyond the first normal form (1NF). The first normal form is defined as a relation that has atomic or single-valued attributes, i.e. only one value for a given row and column in a relational table. This normalization alleviates many problems that typically arise during updates when data redundancy exists. C. J. Date describes a good relational design principle as “one fact in one place”.[21] Numerous normal forms have been defined by relational database experts. The first three normal forms (1NF, 2NF, 3NF) were defined by Codd in reference 22. The motivation behind Codd’s definitions was that 2NF was “more desirable” than 1NF, and 3NF in turn was more desirable than 2NF. That is, the database designer should generally aim for a design involving relations in 3NF, not relations that are merely 2NF or 1NF.[21] However, Codd’s original definition of 3NF turned out to suffer from certain inadequacies. These inadequacies led to the revision of Codd’s original 3NF definition and the creation of a stronger definition known as the Boyce-Codd normal form (BCNF). Table 1 shows the ascending series of normal forms.

A relation is in this normal form	*... if it is in all more basic normal forms and obeys these constraints:*
First normal form (1NF)	It has atomic (single-valued) attributes.
Second normal form (2NF)	All of its nonkey attributes are functionally dependent on all of its keys.
Third normal form (3NF)	It is free of transitive dependencies.
Boyce-Codd normal form (BCNF)	Every one of its determinants is a candidate key.
Fourth normal form (4NF)	It is free of multivalued dependencies.
Domain/key normal form (DKNF)	All logical restrictions on its contents are logical consequences of its key and its attributes’ domains.

Table 1. Normal form definitions.

Date expands on the definition of 3NF by stating that a relation is in third normal form if and only if the nonkey attributes (if any) are: (a) mutually independent, and (b) irreducibly dependent on the primary key.[21] Relations in first or second normal form have anomalies concerning modifications and those in third normal form do not. Therefore, third normal form was chosen as a minimum normalization for this research.

Logical Database Design (IDEF1X)

Figure 6 shows the IDEF1X model for the aircraft components. An aircraft configuration is made up of components. For this application those components are the: engine, fuselage, gear, inlet, nozzle, canard, horizontal, vertical, and wing.

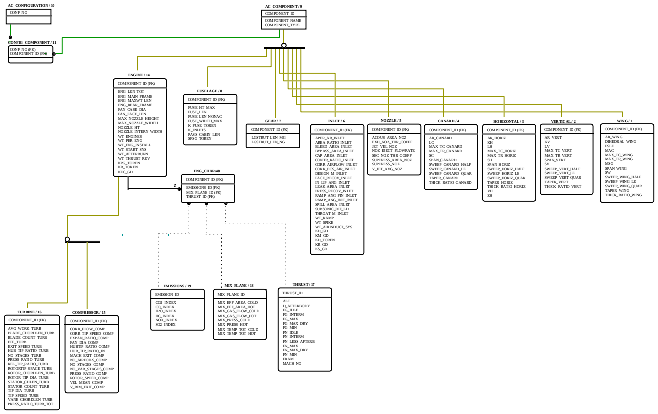


Figure 6. IDEF1X diagram of aircraft components.

Figure 7 shows the IDEF1X model describing an aircraft member along with member material, and load and stress characteristics.

Aircraft stability and control, aerodynamic, performance, cost, and weight data have been modeled as a function of the aircraft configuration. This relationship between aircraft configuration and this calculated data is parent to child. For example, an aircraft configuration has a given weight. This weight can be made up of many different fixed equipment combinations, fuel systems, etc.. IDEF1X models for the weight, stability and control, cost, performance, and aerodynamic data respectively were developed for this research.

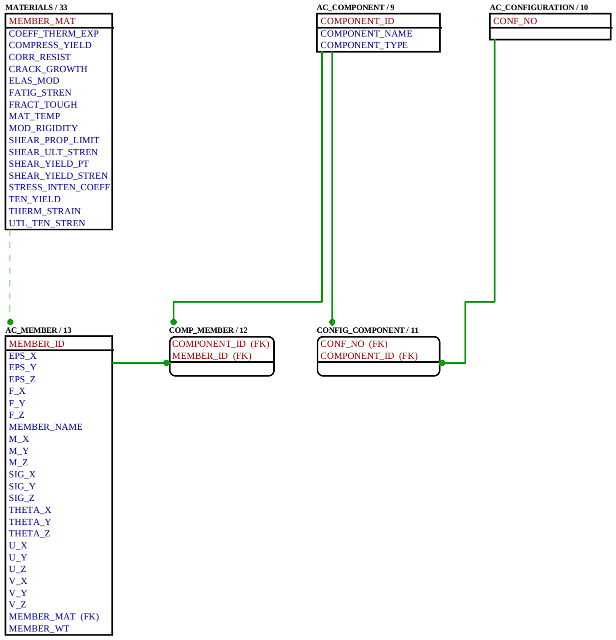


Figure 7. IDEF1X diagram of aircraft component members and the respective material, load, and stress.

Implementation of Database

Implementation of the database model can be on any of the available relational database management systems such as ORACLE, SYBASE, or INGRES.

OBJECT-ORIENTED DATABASE DESIGN APPROACH

Data Relationships Modeling

The design treats an aircraft configuration as a object which is composed of other component objects. The objects making up an aircraft configuration are a: wing, horizontal, vertical, canard, fuselage, engine, nozzle, inlet, and gear. An engine is made up of a compressor and a turbine. Each of the aircraft component's objects are made up of member objects which have load, stress, and material characteristics. The typically disciplinary calculations of aerodynamics, cost, weights, performance, and stability and control are treated as objects of an aircraft configuration. This seems a little unnatural, however, these calculations have been traditionally grouped by discipline and it is probably a good guess that they will continue to be associated in this manner for some time to come.

Database Schema

A complete lexical EXPRESS model was developed for the Object-Oriented design. An excerpt from that lexical model is shown below for example.

```
ENTITY ac_configuration;
   conf_no                    : identification_no;
   consists_of                 : SET [1:?] OF ac_component;
   has_specific               : costs;
   demonstrates               : performance;
   has_defined                : weights;
   characterized_by           : stability_and_control;
   has_inherent               : aerodynamics;
UNIQUE
   un_conf_no                 : conf_no;
END_ENTITY;

ENTITY ac_component;
   component_id               : identification_no;
   component_name             : name;
   component_type             : name;
   made_up_of                 : SET [1:?] OF ac_member;
UNIQUE
   un_component_id            : component_id;
END_ENTITY;
```

Data Dictionary

The HSCT object-oriented design data model utilizes the same data dictionary as the relational design.

Logical Database Design (EXPRESS)

Figure 8 shows the EXPRESS model for the aircraft components. Different from the relational design, the aircraft configuration object (ac_configuration) has attributes that extend beyond simple data types. The disciplinary calculations of costs, weights, aerodynamics, performance, and stability and control are considered attributes of aircraft configuration. Another important point is that a uniqueness constraint exists for the simple data type of identification_no. This identification number is inherited by the disciplinary calculation objects as well as the aircraft components (ac_component).

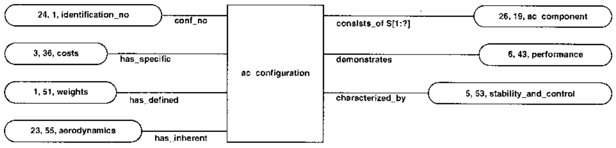


Figure 8. EXPRESS model -aircraft configuration.

Figure 9 shows that an aircraft component can be an: engine, fuselage, gear, inlet, nozzle, canard, horizontal, vertical, and wing, with each object being a subtype of aircraft component (ac_component) and therefore the heavier black lines.

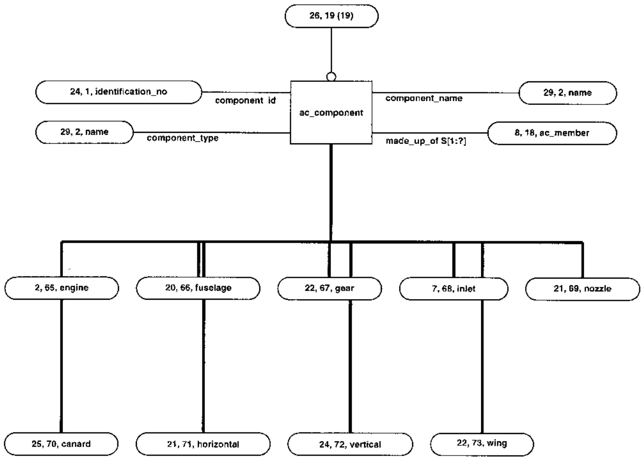


Figure 9. EXPRESS model - aircraft components.

An aircraft component is made up of aircraft members. Figure 10 shows that an aircraft member (ac_member) is made from materials and therefore the material object is considered an attribute of aircraft member (ac_member). The figure also shows that an aircraft member has load and stress characteristics which are considered part of the member object.

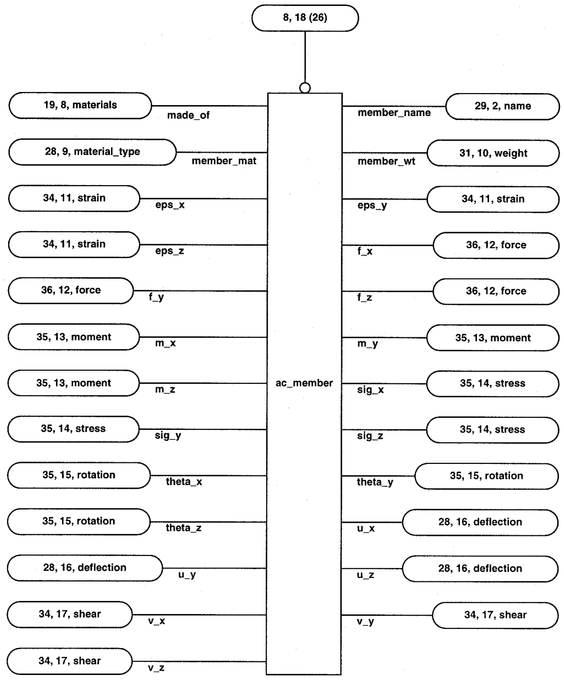


Figure 10. EXPRESS model - aircraft member.

Aircraft component objects that make up an aircraft configuration such as the wing, horizontal, fuselage, etc., and the aircraft disciplinary calculation objects: cost, aerodynamics, weights, performance, and stability and control were also modeled in EXPRESS for this research.

Implementation of Database

Implementation of the database model can be on any of the available object-oriented database management systems or object-oriented database programming languages such as Objectivity/DB, ONTOS, ObjectStore, VERSANT, and GemStone.

DATABASE PERFORMANCE METRICS

Benchmarks

Little work has been done on performance in the field of object data management, despite its importance to most applications.[15,16] Moreover, there seems to be even less research in the area of performance comparisons between relational and object-oriented DBMSs. One difficulty lies in the understanding of what constitutes performance? In his book, R. G. G. Cattell discusses two kinds of DBMS performance issues, model-based and architecture-based.

> *Model-based:* In some cases, performance is limited by the data model, regardless of how good the implementation. For example, relational-model implementations have an impedance mismatch between programming and query language, forcing an application to represent a list (such as the chapters of a book) as a table, and to copy the data wholesale from the table to a list in the programming language at runtime in order to manipulate the elements efficiently.
>
> *Architecture-based:* ... the implementation of specific ODMS features can have major performance implications. In some cases, the implementation choices for two particular features, such as concurrency control and remote databases, can interact favorably or very badly for overall speed. Thus, it is important to consider the overall view.[14]

Application speed is still considered one of the most important performance metrics when comparing DBMSs. Cattell and Skeen developed the OO1 (Object Operations, Version 1) benchmark to address some of these performance issues. The OO1 benchmark is intended as a generic measure of ODMS performance. It was designed to approximate database needs of CAD, CASE, and similar applications. A simple database of parts is used with a many-to-many connection relationship between the parts. Three kinds of operations were performed on the parts and connections: lookup, traversal, and insert.[17] Figure 11 shows the results of Cattell and Skeen’s work using the OO1 benchmark for a cold start and with database caching. The benchmark operations were run on a database of approximately 4 megabytes, with 20,000 parts and 60,000 connections. Cattell and Skeen showed that with database caching, the OODMBS was 30 times faster overall than the RDBMS. The OO1 benchmark was run on the object-oriented database programming languages Objectivity/DB, ONTOS, ObjectStore, VERSNAT, and GemStone, as well as the relational database products SYBASE and INGRES at a later time. Cattell and Skeen’s findings were again consistent with the results shown in figure 11. They attribute the differences in relational and object-oriented DBMS performance to be to architecture-based rather than model-based.

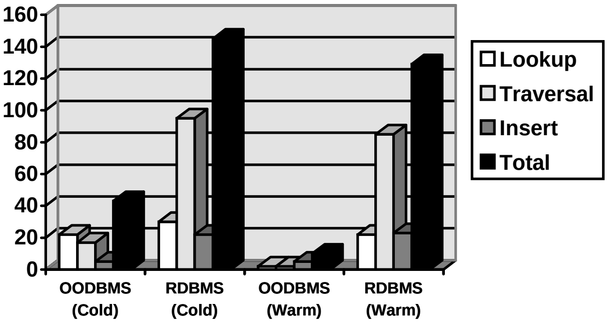


Figure 11. OO1 benchmark comparison of traditional relational DBMS against object-oriented database programming language in seconds.

Performance Comparisons of ORACLE and ROSE/C++ Database Models

In order to evaluate the performance of the conceptual IDEF1X and EXPRESS designs, the stability and control portion of the database design was populated with data. The relational representation of the stability and control data requires seven (7) tables in order to adhere to strict third normal form. The object representation of the same data modeled stability and control data as an attribute of aircraft configuration. An experiment was set up to conduct table lookups for all 47 variables representing the stability and control data. The number of variable lookups was then doubled and tripled, and the system CPU time in seconds required to carry out the task was recorded. The ORACLE and C++ models were ran on Sun SPARC stations 2000 and 20 respectively, with Solaris 2.4 operating systems. A better comparison would have been to run each model on the same machine. However, due to limitations in the availability of software for given machines, this was the only available option. Figure 12 shows the results of the experiment. The question immediately arises as to how to separate out the two performance issues, model-based and architecture-based. From the work done by Cattell and Skeen it was determined that it is not possible to compare the performance of different implementations through abstract analysis except in some simple cases.[17] A better comparison might be to emulate the relational tables in an object environment (and on the same machine), then using C++ as the query language. This would possibly eliminate the architecture-based performance issue, but the model-based issue would still remain. The focus of this research is evaluating database design methods and how these methods are impacted by an evolving database design. The conclusion is that for this domain (i.e. aircraft conceptual design), the measure of performance that is deemed most important is how the methodology performs in the environment in which it was designed. It would be quite unnatural to model objects like relational tables in order to provide a more neutral ground from which to evaluate performance. The bottom line is that the aircraft designer is interested in how easy it is to introduce changes to a data schema and implement those changes.

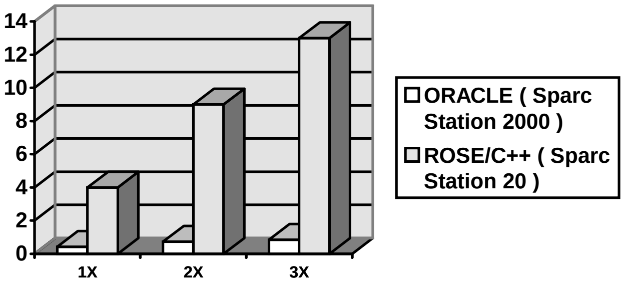


Figure 12. Comparison of ORACLE against object-oriented database programming language (C++) for table lookups in seconds.

DESIGN COMPARISONS

Evaluation of IDEF1X and EXPRESS (General)

Eastman and Fereshetian have developed an excellent set of design product-modeling criteria. Table 2 shows this criteria for the IDEF1X and EXPRESS data models. Eastman and Fereshetian found that IDEF1X lacks adequate support for object-oriented concepts and does not address operator semantics provided by abstract data types of methods. EXPRESS provides strong capabilities for defining the structures often developed in object-oriented databases. Both IDEF1X and EXPRESS fail to reflect the dynamic and evolutionary nature of design, because of the varied sequence of applications and the possibly dynamic definition of the database schema as the design proceeds. The IDEF1X and EXPRESS models appear to address more the manufacturing end of the product development process, where the issues of change are less important.[19] The domain for this work was derived from the structure and function of walls.

Evaluation of IDEF1X and EXPRESS (Specific)

Database designs have been generated using both the IDEF1X and EXPRESS data models. Through this work certain benefits/detriments of the IDEF1X/Relational and

Concept	Design Need	IDEF1X	EXPRESS
Full abstract data types	Needed for object semantics	Missing operators	Yes, with operating constraints
Multiple specialization’s	Important for abstraction	Yes, supports partial orders	Yes, supports partial orders
Composite objects	Important for abstraction	Supported	Supported
Relations within compositions	Important for abstraction	Not supported	Supported with precedence on relations
Relations on object structure	Needed for semantics	Supported	Supported
Relation between variables	Needed for semantic definition	Not supported	Supported
Variant relations	Needed for schema evolution	Not supported	Not supported
Variant relations defined operationally	Needed to define state of integrity	Not supported	Partial support
Integrity management of external applications	Needed for applications management	Not supported	Partial support
Management of partial integrity	Needed for iterative design	Missing - assumes total integrity	Missing - assumes total integrity
Supports schema evolution	Needed to support design evolution	Basically static, some structure evolution	Basically static, some structure evolution
Cont. refinement versus class instances	Needed for design refinement	Class instance	Class instance

Table 2. Evaluation of information models according to design product-modeling criteria.

EXPRESS/C++ design and implementation have been found that supplement those documented in reference 19. Table 3 details these findings.

Attribute	IDEF1X/Relational Implementation	EXPRESS/Object-Oriented Implementation
Data redundancy	No data redundancy	Data redundancy
Real world representation	Data structure is unnatural. Does not replicate how data is actually collected and kept during the design phase.	Data is structured in a way that better replicates how data is actually collected and kept during the aircraft conceptual design phase.
Design tools	Limited design tools	Limited design tools
Speed		Has been demonstrated to be faster than conventional RDMSs when model-based and architecture-based issues are eliminated.
Ease of schema changes	Can be difficult to modify schema, but not impossible	A little more flexible to modify schema than IDEF1X, but still requires some work
Programming language interface	Application language and query language typically different	Can have a common application and query language
Manipulation of objects	Requires multiple queries and can be somewhat difficult	Object orientation makes it easy
Inheritance	Does not support inheritance	Supports inheritance
Many-to-many relationships	Many-to-many relationships require the introduction of a relationship table in the relational model	Many-to-many relationships can be represented directly in the object-oriented model

Table 3. Evaluation of information models for conceptual aircraft design.

In nearly all applications, it is important to be able to modify a schema with minimum impact on exiting applications. This can be even more important in design applications, because the user as well as the application programmer may modify the schema (for example, to define new types of design components or design constraints). Current DBMSs do not provide good facilities to migrate data to new schemas.[14] Typically, the application language and query language are different in RDBMSs requiring pre-compilers. If the C++ programming language is chosen, then the application and query language can be common. The OODBMSs allows for an object type to have all of the attributes of an existing object whereas RDBMSs do not support inheritance. Binary many-to-many relationships can be represented directly in the object-oriented model through two list-valued attributes, but they demand the introduction of a relationship table in the relational model (if the database is to be in first normal form).[14]

CONCLUSIONS

New aircraft designs have become increasingly advanced and complex. Advances made in the scientific and engineering technologies have resulted in nontraditional aircraft designs using high technology materials. Multidisciplinary Design Optimization (MDO) will take advantage of an evolving high speed computing environment and will be a critical component in the design of the HSCT. A major emphasis is also being placed on using concepts such as Integrated Product and Process Development (IPPD) and Concurrent Engineering (CE) as a means of improving the product development process.

The multidisciplinary design effort of the HSCT will require large amounts of data exchange. The advancements made in computing technology will further this enormity of data. It is critical that a data management system be in place very early in the design process, preferably before the process begins. The design of a data management system should command the same level of priority as that given to other disciplines involved in the process. Moreover, customers have been independently developing data management structures for use internally in order to streamline processes and costs. In today's environment, the customer wants to be directly involved in the design process. This has certainly been proven with the design of the Boeing 777. In order to be responsive to customer requirements, a data management system must be in place.

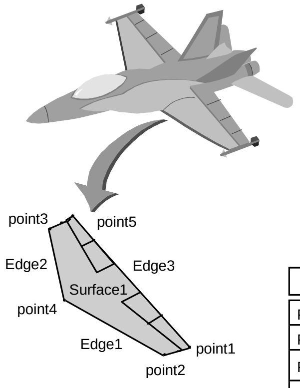
This research has focused on the impact of data modeling and database implementation methods in order to gain a better understanding of how efficient data management can optimize the aircraft design process. This research has included the development of a formal process model for the conceptual aircraft design sequence. The author has been involved in numerous conceptual studies over the last ten years with two major airframe companies. Although each company was very active in the conceptual design process, there seemed to be a lack of process formality. Part of this research has been to identify a common process in order to ascertain what the data requirements are for the process.

Two database design approaches have been taken. An IDEF1X approach with a relational implementation and an EXPRESS approach with a C++ programming language implementation. In the development of these database designs it became apparent that current database design approaches are typically limited to the detailed design phase where the data organization is fixed. A major problem is the development of a database design approach to support the conceptual design of complex engineering products where the database organization is evolving.

The popularity of the relational data model is partly due to its simplicity. It is easy to understand because data is structured in tables, a concept familiar to almost everyone. The maturity level of the RDBMs also makes it quite attractive. It provides a very powerful query language and very little programming is required for implementation. However, the relational model is best suited for the data retrieval and manipulation of business application requirements and not engineering applications. In modeling the data required for conceptual design, if third normal form is strictly enforced, the organization of the data is very unnatural. Unnatural here means that the data structure is very unrepresentative of how that data exists in the physical world.

The object-oriented representation of conceptual design data does a better job at providing a more realistic or natural data structure than the relational approach. Cattell states that the context of object data management are in the three areas of software engineering, mechanical and electrical engineering, and documents.[14] Design tools such as CAD and CAE have database systems embedded inside them which are not typically accessible by the user. The problem arises when the user is faced with a variety of application all with incompatible data representations.

Consider the simplistic wing example shown in figure 13. In order to describe a single surface with relational tables would require a minimum of the three tables shown. An edge is described by two points and five edges describe the single surface denoted as *Surface 1*.



Edges

EdgeID	Length	Attribute
Edge1	100.	. . .
Edge2	100.	. . .
Edge3	350.	. . .
. . .	. . .	. . .
. . .	. . .	. . .

Points

PointID	X	Y	Z
Point1	132.0	75.0	100.0
Point2	125.0	75.0	100.0
Point3	125.0	-75.0	100.0
. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .

StartEnd

EdgeID	PointID
Edge1	Point2
Edge1	Point4
Edge3	Point1
. . .	. . .
. . .	. . .

Figure 13. Relational database representation of a wing surface.

A major drawback of a relational design becomes quickly apparent in this example, that is the problem of segmentation. Kemper states that one of the most severe drawbacks of the relational model is the need to decompose logically coherent application objects over several base relations.[20] Due to the segmentation introduced in the relational design, in order to perform a simple rotation of the surface requires a query consisting of multiple joins. The query serves to reconstruct the object for rotation. The object-oriented environment provides a means for modeling the structure of the object as well as its behavior. It is this function that makes the object-oriented approach to data modeling very appealing for engineering applications.

RECOMMENDATIONS

Conceptually, both the IDEF1X/ORACLE and EXPRESS/Object-oriented programming language approaches fall short of providing the user with the ease of schema changes. However, less work is required when using the EXPRESS/Object-oriented programming language methodology when making schema changes. This is a very desirable feature in the domain of conceptual aircraft design. It is important for the user to be able to modify a schema with minimum impact on existing applications.[14] Of the two approaches researched, the EXPRESS/Object-oriented programming language offers the best solution at the present time. The reasons for this choice are: (1) more commonality with the physical world, (2) commonality in application and query language, (3) the increased activity and support for STEP protocols, which are developed using EXPRESS, (4) relative ease in the manipulation of objects, (5) OODBMSs currently have the best chance for providing a solution where applications and CAD models share a common database, and (6) OODBMSs are still in the infancy stage when compared to RDBMs. The power of the RDBMS is partly derived from the RDBMS's maturity level. Where RDBMS capabilities are becoming more saturated over time, the OODBMS capabilities are just beginning to be exploited, and this will continue to increase over time.

Figure 14 is a recommended database design strategy. The ultimate goal would be to develop an application protocol within STEP. With the numerous applications that have been developed already, as well as those currently in development, there does not exist a project for developing an application protocol for core conceptual aircraft design. The author realizes that the conceptual design phase is highly dependent upon the design in question, however, a baseline standardization would definitely serve as a design optimization tool. This would be extremely helpful in the situation where there are a variety of different companies working on the same design and the need for sharing data exists. The steps proposed for database design are:

1. Development of a process model
2. Development of a data dictionary
3. Development of a data schema
4. Creation of populated objects
5. Object-oriented programming language for data manipulation
6. Development of a STEP application protocol

It is also suggested that the data schema should serve as a template to help optimize the design process early in the data management development stages. One of the impediments to successful database design implementation is overcoming the existing cultural barriers within a company. Using the data schema as a template prior to any database implementation would at least eliminate some of the data redundancy problems that are so common today. Cultural barriers are built up over time. Overcoming some of the data management problems will take time as well. This first small step could serve as the first increment toward breaking down some these cultural barriers.

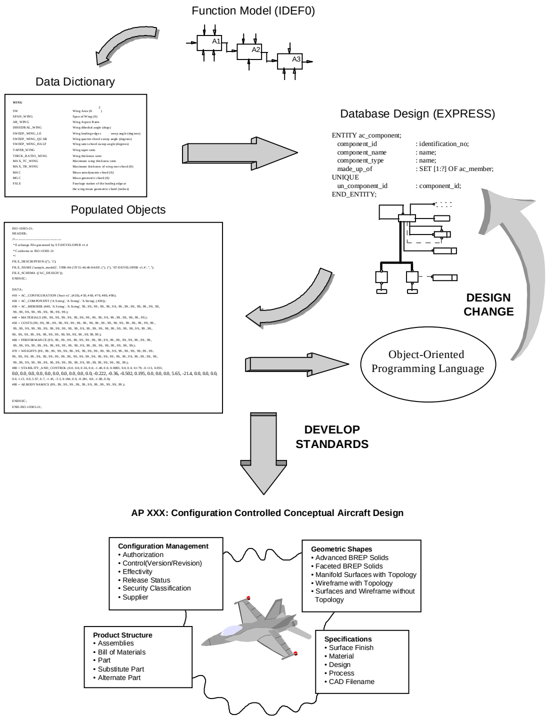


Figure 14. Recommended database design strategy and standards.

ACKNOWLEDGMENTS

The author would like to express his sincere appreciation to Dr. Robert E. Fulton for providing me with the unique opportunity to conduct research in the area of information management while supporting a family. His encouragement and confidence will always be valued.

I would like to thank James J. Duhig, Jr. of the Atlanta Electronic Commerce Resource Center. Jim’s wealth of industry experience provided me with immeasurable insight into the practical implications of information management.

The author would like to express his appreciation to Johnny Stamper, Aaron Harcrow, Paul Cole, Bill Hood, and Charlie Novak of Lockheed Aeronautical Systems Company for their creative and technical input.

And lastly, the author would like to thank Gintas Jazbutis and M. C. Ramesh for answering any question that I could come up with, and a special thanks to Diego Tamburini for his assistance with STEP tools.

REFERENCES

1. Fulton, R.E. and Chadha, B., Engineering Data Management, Course Notes, ME 6754, Georgia Tech Publishing, 1994.

2. Schrage, D.P., et. al., "New Approaches to HSCT Multidisciplinary Design and Optimization", Georgia Tech Proposal in Response to NASA Research Announcement 93-LaRC-1, 29 July 1993.

3. Kroo, L., and Takai, M., "A Quasi-Procedural, Knowledge-Based System for Aircraft Design", AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, Atlanta, GA, Sept. 1988.

4. Sobieski, J., "Multidisciplinary Optimization for Engineering Systems: Achievements and Potential", NASA TM 101566, March 1989.

5. Ashcom, J.R. and Bsharah, F., "Engineering Data Requirements Using an Information Model".

6. Schenck, D.A. and Wilson, P.R., Information Modeling: The EXPRESS Way, Oxford University Press, 1994.

7. Peak, R.S., "An Overview of STEP", Presentation to GT CALS Research Group, 3 December 1991.

8. Roskam, Jan, Airplane Design, Part I: Preliminary Sizing of Airplanes, Roskam Aviation and Engineering Corporation, 1985.

9. Roskam, Jan, Airplane Design, Part V: Component Weight Estimation, Roskam Aviation and Engineering Corporation, 1985.

10. Nicolai, Leland M., Fundamentals of Aircraft Design, METS, Inc., 1975, Revised 1984.

11. Roskam, Jan, "Understanding Cost", Roskam Aviation and Engineering Corporation, 1986.

12. Lan, Chuan-Tau Edward and Roskam, Jan, Airplane Aerodynamics and Performance, Roskam Aviation and Engineering Corporation, 1980.

13. Roskam, Jan, Airplane Flight Dynamics and Automatic Flight Controls, Part I, Roskam Aviation and Engineering Corporation, 1979.

14. Cattell, R.G.G., Object Data Management: Object-Oriented and Extended Relational Database Systems, Addison-Wesley Publishing Company, 1993.

15. Maier, “Making Database Systems Fast Enough for CAD Applications”, Technical Report, Oregon Graduate Center, Beaverton, Oregon, 1988.

16. Hardwick, R., “Why ROSE is Fast: Five Optimizations in the Design of an Experimental Database System for CAD Applications”, in Proceedings of the ACM SIGMOD Conference, San Francisco, California, 1987.

17. Cattell, R.G.G., Skeen, J., “Object Operations Benchmark”, ACM Transactions on Database Systems (in press). 1991-92. Derivative paper in [Gray 1991].

18. Gray, J. N., The Benchmark Handbook for Database and Transaction Processing Systems, Morgan-Kaufmann, 1991.

19. Eastman, Charles M., Fereshetian, Nirva, “Information models for use in product design: a comparison”, Computer_Aided Design Volume 26, Number 7, July 1994.

20. Kemper, Alfons Heinrich, Object-oriented database management: applications in engineering and computer science, Prentice-Hall, Inc., 1994.

21. Date, C. J. , An Introduction to Database Systems, Addison-Wesley Publishing Company, 1995.

22. Codd, E. F., “Further Normalization of the Data Base Relational Model.” In data Base Systems, Courant Computer Science Symposia Series 6, Prentice-Hall, 1972.