3-1997

Net-Centric Design and Analysis of Information Systems

Timothy R. Schmoyer

NET-CENTRIC
DESIGN AND ANALYSIS
OF
INFORMATION SYSTEMS

THESIS

Timothy R. Schmoyer
Captain, USA

AFIT/GE/ENG/97M-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/97M-03

NET-CENTRIC DESIGN AND ANALYSIS OF INFORMATION SYSTEMS

THESIS

Timothy R. Schmoyer
Captain, USA

AFIT/GE/ENG/97M-03

Approved for public release; distribution unlimited

AFIT/GE/ENG/97M-03

NET-CENTRIC

DESIGN AND ANALYSIS

OF

INFORMATION SYSTEMS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Timothy R. Schmoyer, B.S.E.E.

Captain, USA

March 1997

Acknowledgements

There are many people who have contributed their time, support, and expertise to this research. I thank everyone I talked to at the Pentagon for helping me to understand the issues and terms that are involved in such ambitious projects as Force XXI, the Army Enterprise Strategy, battlefield digitization, situational awareness and situational age.

For their patience, ability to always find time to help, and willingness to explain more than once, I would like to thank Colonel Deale and Major Renner of the Office of the Director of Information Systems for Command, Control, Communications and Computers. Without their expertise and sponsorship, this would never have been possible.

For his interest and time spent discussing with me Joint communications and perishability of information, I thank Lieutenant Colonel Spalviero of the Joint Chiefs of Staff J6.

For his friendship, guidance, and willingness to provide resources, I am once again indebted to Mr. Joe Landa of Trident Systems, Inc. He not only allowed me to overcome the learning curve at the expense of his leisure time, he opened his home to me while I was at the Pentagon. Thanks again, Joe!

Special thanks to the instructors in the Electrical Engineering and Computer Science Department at the Air Force Institute of Technology. To my thesis advisor, Captain Rick Raines, thanks for your guidance, direction and uncanny ability to lift up and push at the same time. To Captain Mike Talbert, I never would have believed it possible to find excitement in developing database technology, but you made it happen, thanks for all your input and patience.

Finally, and most importantly, thank you Rachel for always being there. Your encouragement, support, and willingness to endure the hazards of my chosen profession make everything possible.

Timothy R. Schmoyer

Table of Contents

Page

List of Figures

List of Tables

List of Symbols

Symbol Page

List of Abbreviations

Abbreviation	Page

AFIT/GE/ENG/97M-03

Abstract

This thesis presents an unique methodology merging state of the art Internet and distributed database technology to support distributed simulations with programming language and platform independence. Standardized models of Command, Control, Communications, Computers and Intelligence (C4I) systems using Integrated Definition (IDEF) models and executable simulation objects are placed in database repositories which can be accessed and implemented over a distributed simulation network using the Common Object Request Broker Architecture (CORBA). The CORBA distributed simulation network accesses heterogeneous distributed databases and performs distributed processes, and supports portability and reuse of simulation objects and interoperability across operating systems and programming languages.

NET-CENTRIC
DESIGN AND ANALYSIS
OF
INFORMATION SYSTEMS

I. Introduction

1.1 Background

My experiences as a communications officer in the U.S. Army have led me to realize the usefulness of modeling and simulation in the planning and analysis of communications networks, the design of communications systems, and the importance of including communications in larger wargaming simulations. As the military becomes more "info-centric", or reliant on the timely exchange of information, communication capability and reliability become increasingly important in our mission planning and assessments. Especially as the U.S. military engages in world-wide, multi-national operations, communication interoperability and reliability are integral factors to the decisions of our military and civilian leaders. Distributed modeling and simulation provide the means to bring together heterogeneous communications systems, and conduct trade-off analysis and optimization within the battlefield scenario, to ensure the highest possible communications interoperability and reliability.

1.2 Motivating Factors

1.2.1 Simulation Disparities. There are two types of simulation disparities which must be overcome in future battlefield simulations:

1. Simulations which can not interact with each other(31).

2. Simulation objects which represent the same real world system, but do not behave similarly in different simulations.

These disparities result because of "stovepipe" simulation applications and the creation of simulation objects without consideration for their reuse or portability (Figure 1.1). Both disparities can be overcome by using an *open* architecture distributed simulation network. Simulations can interact with each other through interfaces which translate requests from one programming language to another. The architecture also directs these requests to repositories where approved simulation objects are stored (Figure 1.2). These simulation objects are based on standardized models to ensure consistent behavior, and enforcing the use of approved objects ensures simulation consistency.

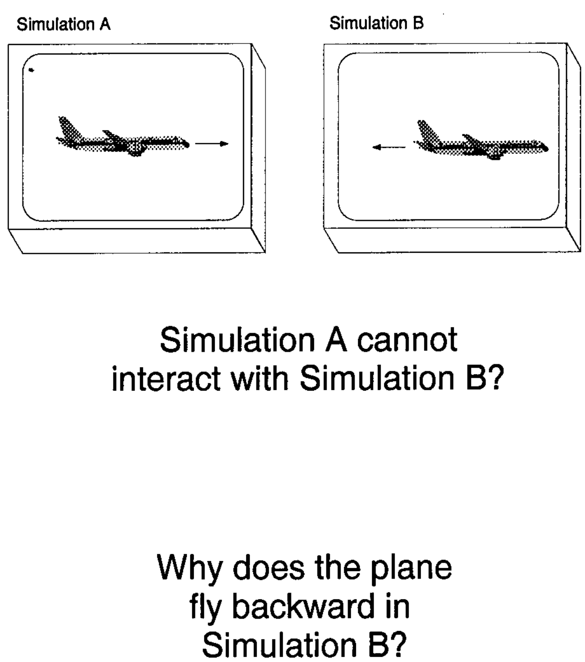


Figure 1.1 Disparate Simulations

1.2.2 Communications in C4I Systems. Situational awareness on the modern battlefield relies on information. The acquisition, analysis, exchange, presentation and use of that information is supported by C4I systems. Communications, within the context of C4I, is the key factor or element that links all the other elements to form a cohesive system. Advances in communica-

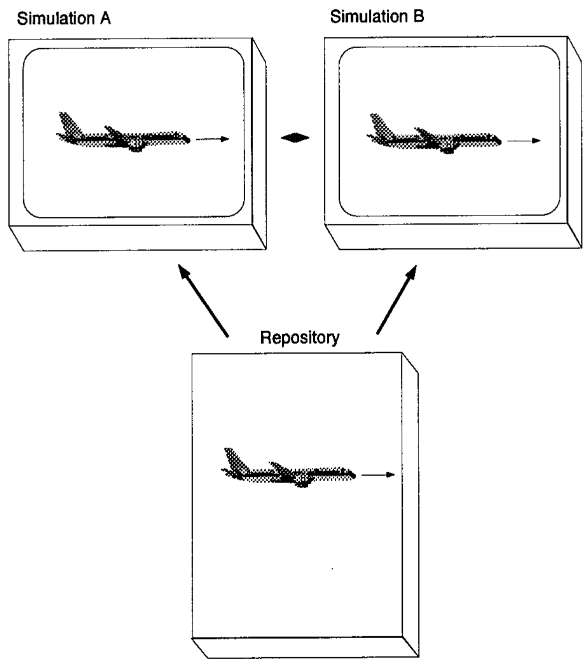


Figure 1.2 Interactive, Consistent Simulations

tions technologies and methods together with inherent changing architectures make consideration of system reliability and availability of services key elements in the decision making process. Unless communications is included as an integral element of wargaming, modeling, and simulations, decisions by our most senior civilian and military leaders will be flawed(24).

The importance of information in the 21st century, demands a fully integrated (horizontally and vertically), robust, stable battle command infrastructure. To support the battle command infrastructure, complex management and communications systems must provide the flexibility and robustness to meet the needs of the highest and lowest echelons of command.

In order to meet the growing demand for information, the information infrastructure must move away from "stove pipe" systems(17) to seamless, interoperable, and worldwide systems. This requirement is reflected in the U.S. Army's Force XXI program. The goal of this program is to create a digital battlefield which can be used to develop tactics, doctrine, training, and evaluate future systems to enhance lethality, survivability, tempo, sustainability, deployability, joint/combined linkages, and versatility(5). To achieve the goal of Force XXI, communications, the backbone of the 21st century force, must be realistically modeled in the digital battlefield.

The digital battlefield provides a virtual world in which highly complex, distributed simulation can be used to test and analyze current and future systems. However, for the digital battlefield to be useful, the virtual world must realistically represent actual battlefield characteristics and conditions faced by the military today and in the future. The digital battlefield must represent how the military organizes and communicates as a function of the mission, environment, terrain and time.

In the dynamic environment of the battlefield, where mobile communications are being extended deep into the enemy's area, the need for timely information over line-of-sight radio, satellite and cable from the farthest forward warfighter to the highest decision maker is increasing. In the 21st century force, communications dependence and its reliability become more important as the rear support area may be within the continental United States, while the mission forces are conducting operations around the world (Figure 1.3).

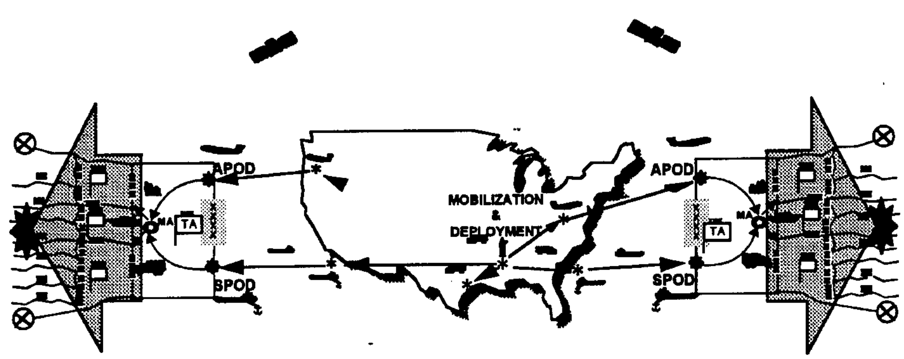


Figure 1.3 Expanding and Dynamic Battlefield (12)

In addition, a communications network is never static; the network is mobile, changing topology constantly. The environment is also changing. Weather and ionospheric changes affect radio paths, and external forces affect cable (more than once a wire was cut by the treads of an armored vehicle during Desert Shield/Desert Storm). These temporally varying conditions must also be realistic in the digital battlefield - reflecting that communications are especially susceptible to these dynamic changes in operations and the environment.

An example scenario is shown in Figure 1.4 to illustrate the dynamics involved in modern scenarios. An enemy tank (A) is observed by an airborne sensor system (B). Both the tank and aircraft are mobile. The sensor's ability to observe the tank is affected by environmental conditions (C) in the path. The information from the sensor is sent to a U.S. Army command organization (D) which identifies the tank, via satellite (E) to a helicopter (F) as a target. A simulation would have to update the scenario based on the mobility of the tank, aircraft, and changes in the environmental conditions which may interfere with the sensor's "visibility" and the communications systems. These obstacles, to name only a few, must be overcome to achieve true models of the actual world in the digital battlefield.

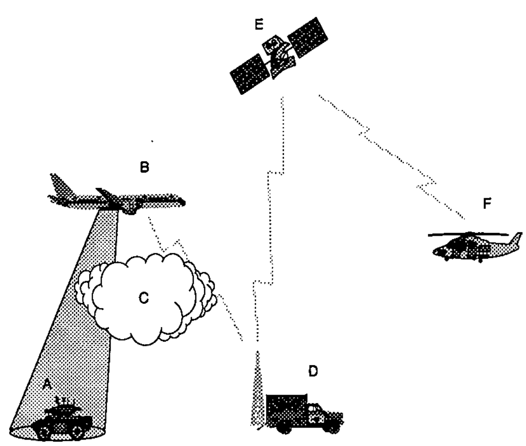


Figure 1.4 Precision Strike Scenario

1.3 Problem Statement and Scope

1.3.1 Review of the Problem. C4I simulations must realistically represent the battlefield conditions the military faces today and will face tomorrow. These simulations must use common objects to ensure consistent behavior and results. These simulations must also be able to interact with each other, regardless of application programming language or operating system inhomogeneities. Finally, communications, which is often overlooked in C4I simulations(24), must be included as the backbone of any information system.

1.3.2 Scope of the Problem. In order to develop an open distributed simulation network using standardized models and simulation objects stored in repositories, three steps must be completed:

1. Model a system described by a given scenario.
2. Develop simulation object(s) from the model.
3. Develop a network architecture to store and access objects using heterogeneous databases, over distinct geographical locations, and return results regardless of programming language and operating platform diversities.

For the purposes of this study, a narrowly defined scenario is chosen to be modeled. The scenario consists of receiving a signal at an OE-254()/GRC Antenna Group and measuring the link margin at the receiver to determine if the message was received. Recall from Figure 1.4 that environmental conditions may affect the strength of the signal actually received by the antenna. This is a representative scenario of all radiowave communication systems. It also uses a metric to determine if the information was successfully received, and can be included as part of a larger scenario.

The contribution of this thesis then, is a methodology to support standardization of the modeling and simulation (M&S) object development process, which combines emerging database and Internet technologies to present a realistic, executable, distributed simulation architecture. The methodology (as shown in Figure 1.5) proposes a process to describe any given scenario by its specific architectures, from which persistent data and executable simulation objects can be

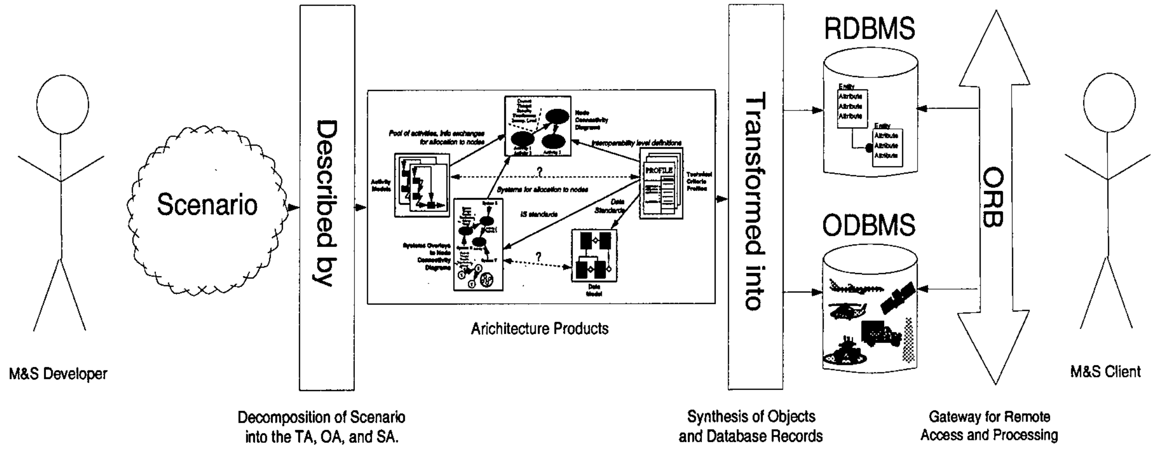


Figure 1.5 From the Real-World Scenario to the Distributed Simulation Architecture

derived for storage in databases. The information and objects in these databases can then be accessed over the Internet using an Object Request Broker. By implementing this development process and distributed architecture, simulations will meet the portability, interoperability and reuse requirements described in the DoD Modeling and Simulation Master Plan (DoD 5000.59-P) and the Defense Modeling and Simulation Office (DMSO) High Level Architecture (HLA).

1.4 General Approach

1.4.1 Develop the Model. The antenna example is modeled by describing its technical characteristics, activities, and the internal and external relationships of the system. The information required to describe the antenna's technical characteristics is modeled in an Entity-Relationship (ER) diagram according to the IDEF1X design method. An example of an ER model is shown in Figure 1.6.

The ER model captures non-transitory data requirements and relationships describing the system. The functions of the system must then be described in an activity model according to the IDEF0 design method. An example of an activity model for a mobile radiowave propagation path is shown in Figure 1.7. The examples are simplistic to give the reader without a background in

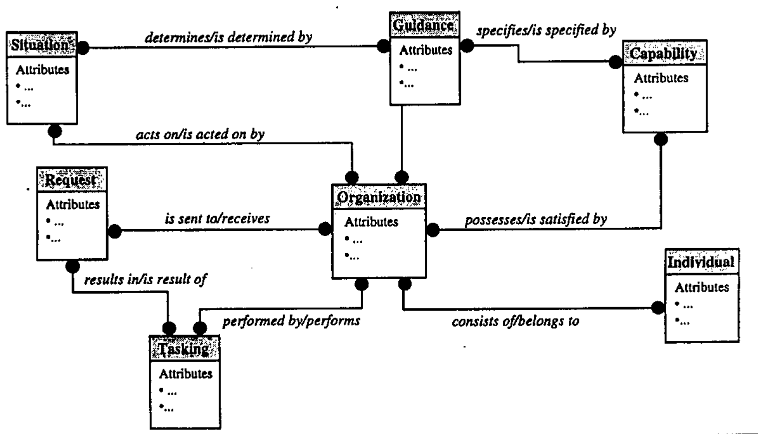


Figure 1.6 Example of an IDEF1x Entity Relationship Model

IDEF modeling an idea of what type of information is captured in each model. IDEF models are further described in Chapter 2.

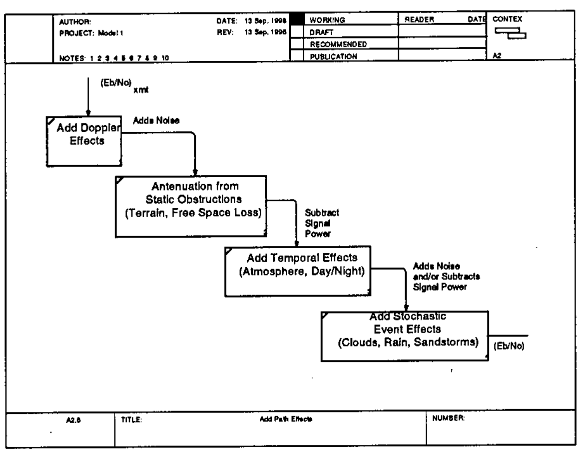


Figure 1.7 Example of an IDEF0 Activity Model

Once the antenna's technical characteristics have been described in the IDEF1X model, they are mapped to the schema of a relational database. Each entity (table) in the database is populated by instances of particular antenna configurations. For example, the OE-254()/GRC Antenna Group is one instance of an antenna, and the antenna entities' attributes are populated using data from MIL-A-49204A, ANTENNA GROUP OE-254()/GRC and Training Circular 24-24, Signal Data

References: Communications-Electronics Equipment. Many antenna configurations can be represented in a relational DBMS in this way, each identified by a unique set of attribute values.

1.4.2 Develop Simulation Object. An executable antenna object is created based upon the IDEF0 activity model. The object is written in an object-oriented language such as ADA, C++, or JAVA. A data sublanguage such as SQL is used to access the data stored in the relational database. The compiled object and any associated objects such as images or sounds are then stored in an object database. This method is chosen to take advantage of the well established relational model for data handling, and to maintain independence between the data and the object. Therefore, the object does not need to be recompiled for minor changes in data values.

An additional advantage to using an object-oriented programming language is the ability to embed objects within objects. For example, a super object could be created which would perform the functions of the airborne sensor and have embedded links to other objects. The links would call other objects which would, for example:

- list the technical references (TA),
- open a graphic of the operational architecture (OA),
- open a graphic of the system architecture (SA) and,
- retrieve model validation documentation.

Each of these embedded objects could also contain other embedded objects or links. This concept of a super object is illustrated in Figure 1.8.

1.4.3 Develop Simulation Network Architecture. These objects would be available to the M&S community over the Internet by an implementation of the Object Management Group's CORBA. A database connectivity bridge to the database(s) where the objects and data are stored would also be required. The complexity and modularity of C4I systems makes object-oriented

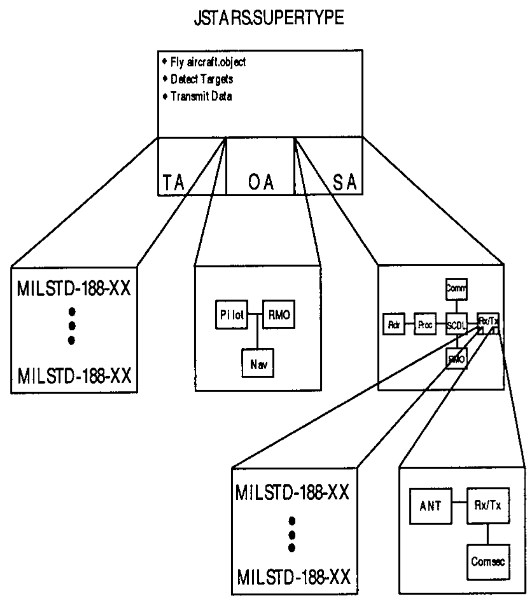


Figure 1.8 Linking and Embedding Objects

modeling and database technology an obvious approach. This approach takes advantage of the distributed nature of the Internet; the interactive, multimedia, and security capabilities of object-oriented databases; and the data handling and security capabilities of relational databases.

1.5 Thesis Organization

This chapter identifies the problem and its scope. Chapter 2 highlights published results which are synthesized to accomplish the goals of this thesis. Chapter 3 elaborates facets of the methodology and describes the process of employing the methodology to solve the problem. Chapter 4 reports the results and importance of the outcomes achieved by this research. Chapter 5 summarizes the problem and the results of this effort and makes recommendations for further research.

II. Background

2.1 Introduction

The first chapter presents a brief explanation and context for the problems faced by developers of C4I systems simulations. These problems are described as disparate simulations which cannot interact with each other, simulation objects which are not standardized among simulations, and simulations which do not account for the fragilities of the communications backbone. The recommended methodology to solve these problems is illustrated by developing an OE-254()/GRC Antenna model and executable simulation object which can be accessed over the Internet using distributed databases and a CORBA implementation. The purpose of this methodology is to ensure standardized, consistent, simulations which can interact with each other; regardless of differences in underlying programming languages or operating systems. This chapter presents background material essential to understanding the methodology described in Chapter Three and the analysis of the results in Chapter Four.

A clear approach to developing standardized models is to describe systems in a standardized way. A system can be described by its technical characteristics and constraints, by its role and function in a larger activity, and by its construction and components. Each of these views provides a unique description of the system. The Defense Information Systems Agency (DISA) and the Office of the Director of Information Systems for Command, Control, Communications and Computers (ODISC4) have defined three architectures which describe C4I systems: the technical, operational, and system architecture. These three architectures together completely define the system and form the basis for the system model. ODISC4 has adopted the IDEF model method to capture and communicate the operational architecture of the system. The IDEF models serve as tools to develop executable simulation objects and map relational and object-oriented databases. The Internet provides the infrastructure to access and process the simulation object and system definition over a distributed network. The following topics are covered in this chapter: 1) The technical, operational, and system architectures, 2) IDEF modeling, 3) Integrating the three architectures into an executable simulation, 4) Distributed databases and the Internet, and 5) The three-tiered network architecture.

2.2 The Technical, Operational, and Systems Architecture

Figure 2.1 illustrates the relationship between the three architectures. Primary inputs to the systems architecture are the operational architecture (defining the process and task flows, logical connectivities and performance bounds to be supported) and the technical architecture (defining the protocols and standards mandated for use to ensure interoperability and seamless connectivity between these component systems). The technical architecture provides the "building code" to the systems architecture(29).

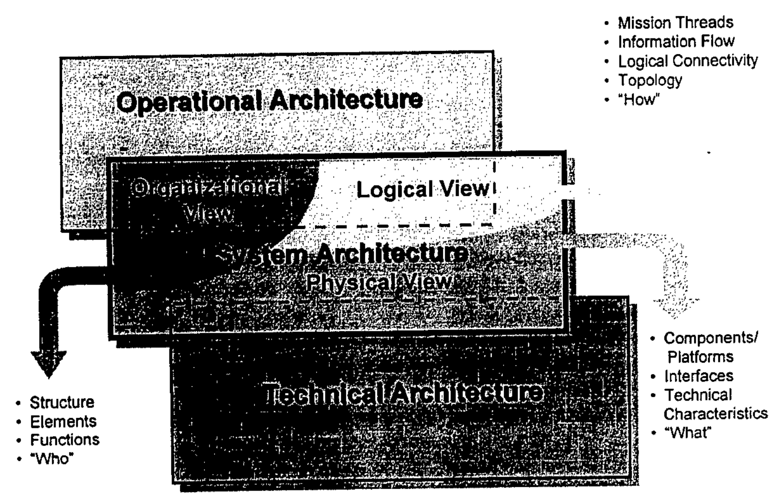


Figure 2.1 Architecture Interfaces (29)

Figure 2.2 illustrates the development of the systems architecture from the operational architecture within the constraints (i.e., "under the umbrella") of the technical architecture. The operational architecture is developed in concert with the US Army Training and Doctrine Command (TRADOC) schools and centers and consequently incorporates current doctrine. The operational architecture describes the information exchange requirements, force structure, movement

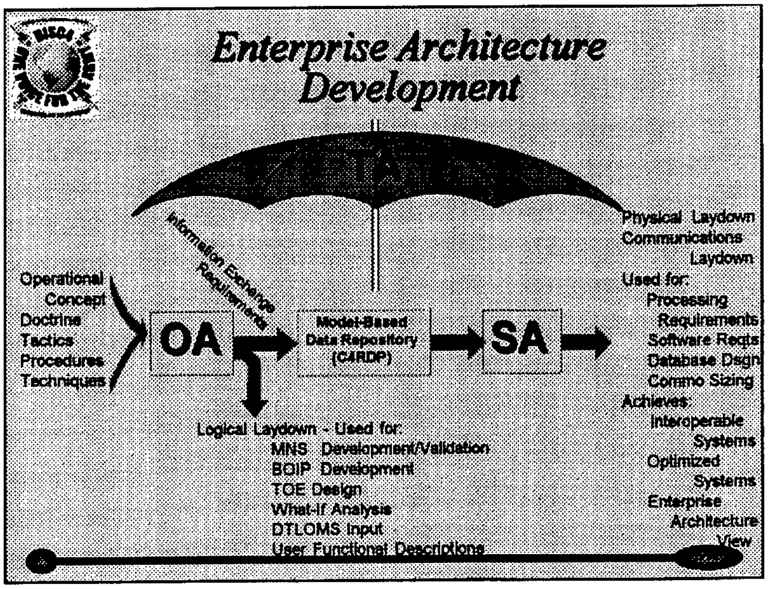


Figure 2.2 Systems Architecture Development (29)

plans, and logical connectivity among organizations and personnel. These are necessary inputs to the system architecture. This information captures connectivity requirements, informational content of exchanges, frequency, speed, perishability, impact of failure and security requirements. The model-based data repository (C4RDP, now known as the C2 Core Data Model (C2CDM)) serves two purposes: 1) it provides standardization of model entities and 2) it takes the product of the operational architecture as an input, and transforms it to the format of the representation needed as input to the system architecture. The system architecture then overlays system solutions to the information requirements to produce the physical and communications structure. In short, the operational architecture represents the information exchange perspective, and the resultant system architecture captures how the hardware and software support those needs.

2.2.1 Operational Architecture. The operational architecture describes who needs to exchange information, what information needs to be exchanged, and how that information will be used. This architecture describes the functions, processes and data required to meet the needs of the warfighter. The operational architecture is captured and communicated using the IDEF modeling methods described in Federal Information Processing Standards (FIPS) 183 and 184.

The IDEF0 method is used to specify function models, which are "what do I do" models. The IDEF0 models let the modeler portray a view of the process using inputs (I), controls (C), outputs (O), and mechanisms (M) (referred to as ICOMs). *Inputs* to an activity consist of resources consumed or transformed by a process and state outputs from other activities. *Controls* are the standards, policies, and guidelines that constrain and guide the process. *Outputs* are the products created by the transformations of the inputs by the process. *Mechanisms* are the agents (people, manual tools, automated tools, etc.) that accomplish the actions delineated within the process. Once these blocks are defined, their relationship to the process and the "flow" of the input through the controls and mechanisms to its final output must be defined(15, 26). An example of an activity model for a radio path is illustrated in Figure 2.3.

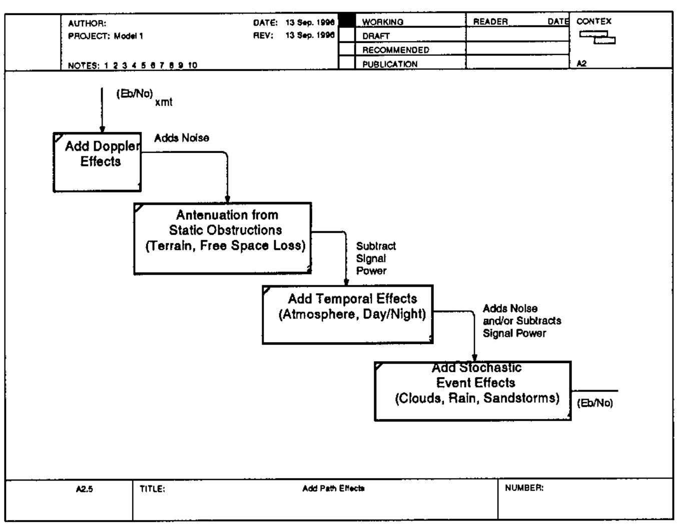


Figure 2.3 Example Activity Model (IDEF0)

The IDEF1X method is based on an ER model(25) and used for data modeling. It captures the logical view of the activity's data. It is a method for logical database (library) design once the information requirements are known. The focus is on the actual data entities, their attributes, and their relationships to other entities in the information system to be developed. This description of the system supports the development of the operational component and system modules once the primitive ICOMs have been defined(15, 25, 21, 8). An example of an information entity relationship model is illustrated in Figure 2.4.

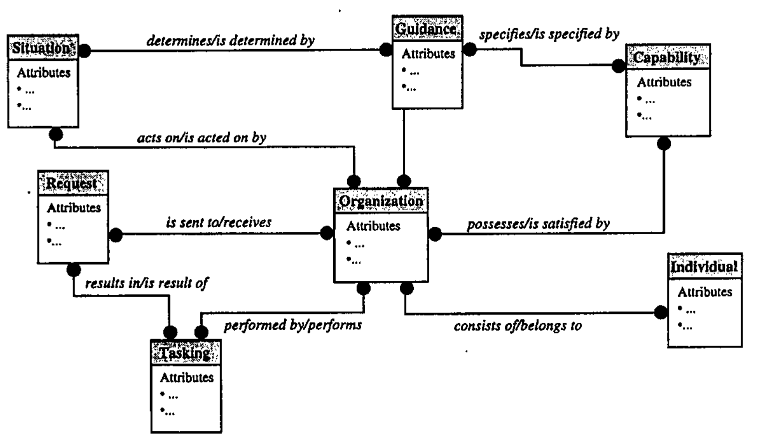


Figure 2.4 Information Model

Therefore, we can take a scenario, or an operational concept, and break it into its functional (IDEF0) and physical architectures (IDEF1X). An operational architecture is obtained by mapping a functional architecture onto a physical one, *for a given operational concept* (Figure 2.5). The operational architecture shows the allocation of resources to functions, the flow of data, and the order in which functions are performed(3).

The operational architecture is then used to(12, 5):

- Conceptualize operational requirements
- Identify unsatisfied requirements
- Prioritize requirements
- Plan evolution of systems
- Develop data standards

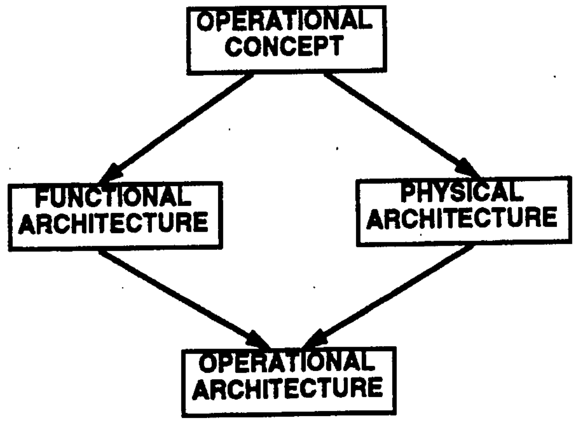


Figure 2.5 Operational Architecture Development (29)

- Measure resource utilization
- Identify interoperability requirements
- Drive the technical architecture

2.2.2 System Architecture. The system architecture shows the specific hardware needed and network topology to provide the connectivity required in the operational architecture. Both architectures are very closely linked. The system architecture is a description of the physical connectivity of an information system, which includes: identification of all equipment (radios, switches, terminals, computers, inter-networking devices, and local area nets (LANs)), their physical deployment, the specification of such parameters as the bandwidth required on each circuit, and the description of the technical characteristics and interconnection of all parts of an information system(5). The system architecture is made up of the top-level diagram in the network model or simulation and each successive component-level diagram in a top-down system of systems model.

The system architecture permits trade-off analysis to optimize overall network and system performance within the performance bounds defined in the operational architecture(5). This top-level view of the information system indicates to the mission and network planner when the system is meeting performance requirements or when new requirements or unsatisfied requirements must be addressed. The requirements in the operational architecture drive this feedback to the planner. However, the system diagram must also identify the impact on the operational architecture as the system degrades as well as the component(s) in the technical architecture which are restricting the system performance. This architectural relationship is illustrated in Figure 2.6 (5, 29).

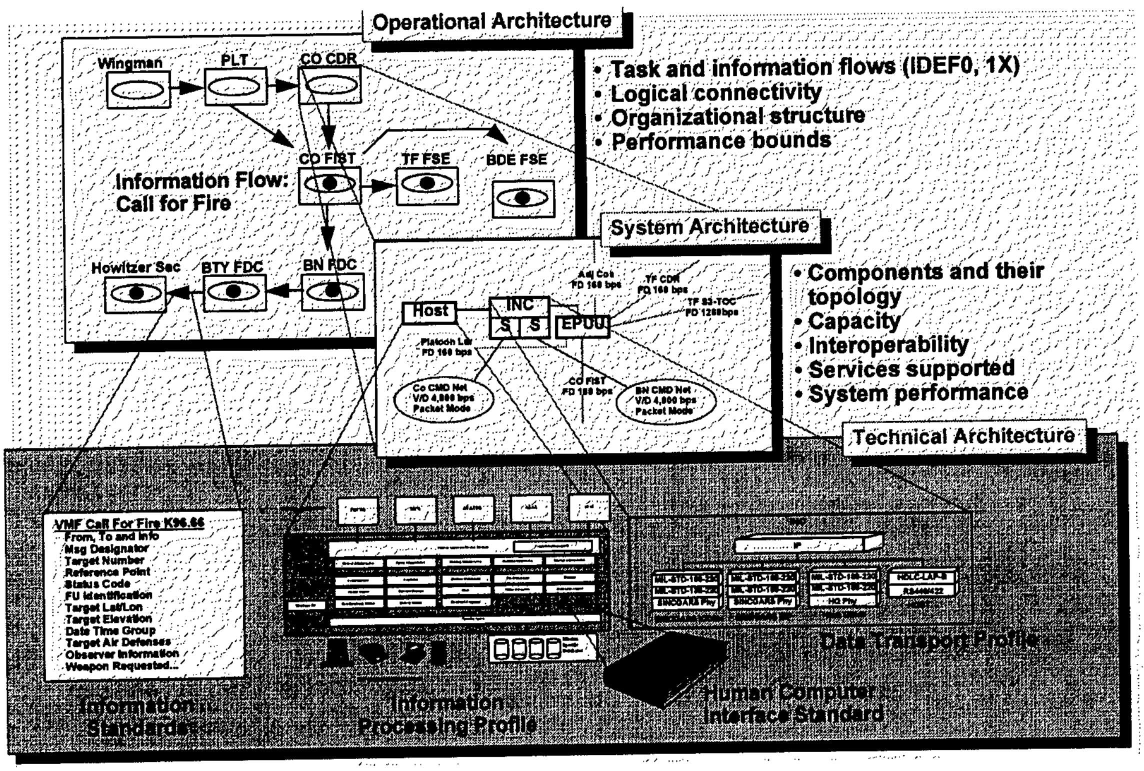


Figure 2.6 Architecture Relationships (29)

2.2.3 Technical Architecture. The technical architecture is defined as a minimal set of rules governing the arrangement, interaction, and interdependence of the parts of an information system. The technical architecture does not say what to build (that is defined by the planner in the operational architecture), nor does it say how to build (that is defined by the planner in the system architecture), but it does say that when you build you must adhere to the set of rules/protocols/standards that it specifies. The standards in the technical architecture establish the framework for achieving interoperability and commonality among component hardware/software and seamless communications connectivity between battlefield elements(5).

The technical architecture provides the "building code" for the systems architecture to meet the requirements in the operational architecture. It addresses the technical aspects of how to integrate the command, control, communications, computers and intelligence (C4I) systems to achieve the operational requirements set forth by the operational architecture. The technical architecture establishes the spectrum of available and approved system implementation options. The technical architecture therefore consists of standards which can be referenced (i.e., COE, C2CDM, use of ADA, HCI standards, MILSTD-188-220 and VMF)(5) and standardized components which meet these technical standards. Upon completion, the technical architecture identifies a frame of standards and includes top-level system specifications, architecture wiring diagrams for technical interface specifications, identification of transactions between equipment and the protocols used in this interface, and supporting performance and modeling results(12).

The Army technical architecture consists of four elements: 1) the information processing profile, 2) the information transport profile, 3) information standards, and 4) human-computer interface (HCI)(35). The information processing profile details standards consistent with commercial and Department of Defense (DoD) Technical Architecture Framework for Information Management (TAFIM)(10). It provides guidance for developing technical information system architectures at all organizational levels of DoD environments that share functions and data. The information transport profile defines the communications support required to provide seamless, reliable, and timely data exchange between users, regardless of the specific communications system(s) used to access the network infrastructure. A communication system conforming to the Open System Interconnection (OSI) Reference Model is an example of this type of information system. Information standards, such as standardized data elements and message format standards, provide the basis for sharing information across gateways using common interfaces. These standards are recorded in

DOD 8320.1 series guidance (Standard Data Elements for Automated Information Systems (AIS) Design and Development), Military Standards (MIL-STD), and commercial Internet standards. Most notable examples for military systems standardization are the Protocol Data Unit (PDU) and the LINK16/Variable Message Format (VMF) (replacement to the United Service Message Text Format (USMTF)). Finally, standardizing the HCI to attain a user interface that presents a common appearance and behavior across joint service systems requires a common set of HCI standards and a common approach to implement them. Volume 8 of the DOD TAFIM contains the DOD HCI Style Guide. This style guide sets the basic standards for graphical HCIs.

The technical architecture describes the major aspects of(12):

- Functions which must be performed to meet required operational capabilities
- Information (e.g., databases, data elements and data definitions, images, sound, text, etc.)
- Applications (e.g., E-Mail, maps, targeting systems, data collection systems, control systems, mission planning systems, office tools)
- Technology (e.g., hardware, systems software, standards, protocols, and communications)

The antenna described in the previous section has predetermined technical architecture constraints and requirements. Once these standards are captured and represented in the IDEF models of the operational architecture, the information can be stored in a model library or repository. If the technical architecture does not allow the system to meet the operational requirements, the operational requirements must be changed. For example, a network planner may be using the antenna in a communications network. However, the antenna may not have the gain required (as set by the technical architecture) to meet link margin requirements across a specific path in the network. A common change to the system architecture by the network planner to maintain satisfactory performance in the network is in the network topology. For example, a platform may move from communicating by one means (e.g., LOS) to another (TACSAT, SATCOM) or the size of the network may be altered by adding or dropping nodes to increase reliability. If the network cannot be modified to maintain communications while meeting the requirements of the operational architecture, the operational architecture must be examined to measure the effect of a loss of communications. The result may be a different tactic or a new requirement. Finally, an operational requirement may be met through an improvement in the technical architecture which removes a constraint that prevented the performance of the system from meeting the operational requirement. The interrelationships of the architecture products discussed in this section are illustrated in Figures 2.7 and 2.8.

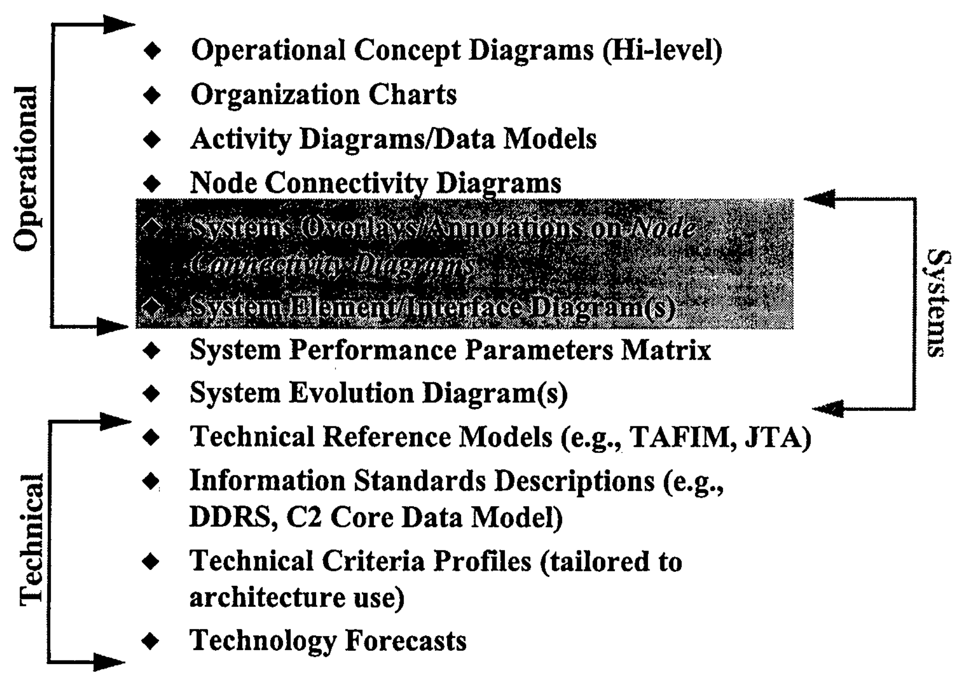


Figure 2.7 Architecture Products (29)

2.3 Integrating the Three Architectures into Executable Simulation Objects

The requirements for the distributed simulation architecture are similar to the requirements described in the Information Mesh project by Sollins of MIT and Dyke of Trilogy Development Group for a General Information Architecture. The General Information Architecture lays out a common, extremely general substrate that sits between network-based systems such as distributed

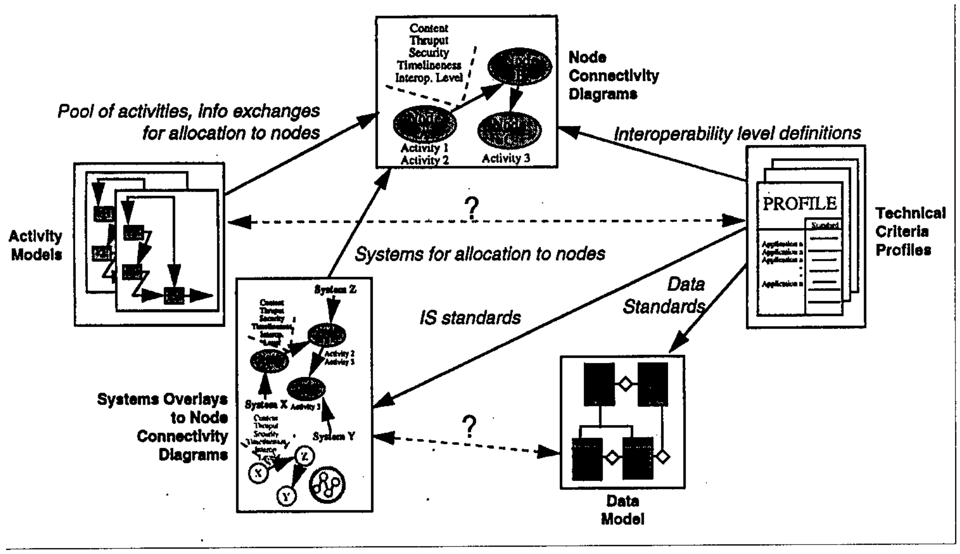


Figure 2.8 Architecture Product Interrelationships (29)

programming or database environments and network transport protocols such as TCP or other bit stream protocols. The intention of the information mesh is to provide sharing and exchange of potentially persistent information independent of the underlying platform operating systems, programming languages, transport protocols and related infrastructure. Although the methodology uses the Object Management Group's (OMG) CORBA, the requirements of the General Information Architecture are identical to the requirements for the distributed simulation network and are listed here(33):

- *Multiplicity.* One must be able to express relationships between two or more objects.
- *Link Typing.* It is important to be able to express the nature of a link or relationship extensively and with unlimited scope, in order to support evolution in the representation of knowledge.
- *Linking into the structure of objects.* In order to provide generality and flexibility in linking objects, it is necessary to be able to link components, aspects, views, or parts of objects to each other.

- *Linking to links.* There are situations in which the ideas being linked may be themselves links. We must be able to relate to links.

- *Composition of objects.* Relationships can be divided into those that are intrinsic or integral to an object and those that are not. If a relationship is intrinsic to an object, that composite object is unable to behave in the intended manner without its components. In contrast, other extrinsic relationships do not have a direct bearing on whether or not the related objects behave as expected. It is important to be able to express this distinction and support composite objects.

- *Support existing models of linking.* Existing systems define the minimum set of characteristics we must be able to support.

The links and composition of objects are the relationships and ICOMs in the IDEF models discussed in previous sections. These links and relationships can then be encoded into objects through the methods and data sublanguage of an object-oriented programming language. The objects can then be stored in an object-oriented database and the data values in a relational database. The advantage to this independence is being able to alter data values without having to recompile objects. The disadvantage is they must be managed jointly, so if the antenna object was deleted from the object database, the antenna records or tables would also need to be deleted from the relational database. This weakness is resolved by using a multi-database management system (MDBMS). An MDBMS would manage relational **and** object-oriented databases, or a hybrid database which combines the two. A great degree of research is currently being conducted to perform this service, and some proprietary systems have been produced to varying degrees of success(36). However, for the purposes of this research, the databases must be carefully managed by the Database Administrator to ensure they are kept current.

Having the data and objects needed for a simulation stored in databases is only useful if they can be called and executed by any user, even when non-collocated with the system hosting the objects. Relational database management systems (RDBMS) are commonly accessed remotely through a client/server architecture today (see Figure 2.9)(2). These applications are often proprietary and operating system specific. However, the recent introduction of open database connectivity (ODBC) applications such as SQL*Net have allowed clients to connect to different vendors' SQL databases over networks, independent of the client's operating system (O/S) platform.

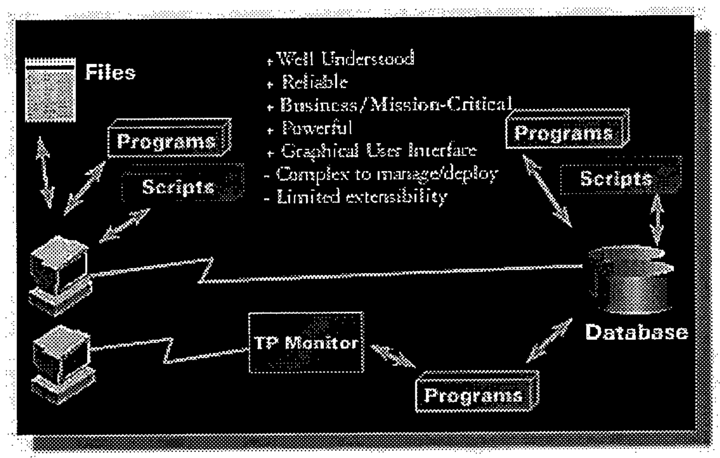


Figure 2.9 Today's RDBMS Client/Server (2)

Object-oriented database management systems (ODBMS) are also accessible in the same manner as relational databases(20, 19) (see Figure 2.10). The object-oriented database provides the capability to store compiled object-oriented programming files and multimedia data types that relational databases cannot. By combining the use of object-oriented databases and relational databases, a multidimensional database is created that is transparent to the client. An example of this multidimensional database system is shown in Figure 2.11.

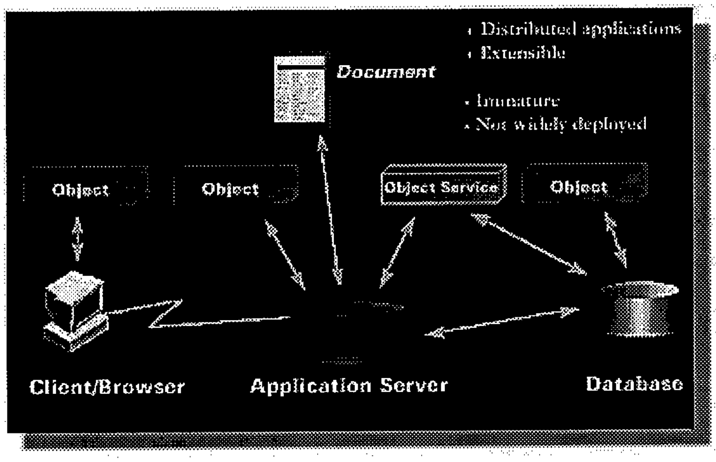


Figure 2.10 Today's Object DBMS Client/Server (2)

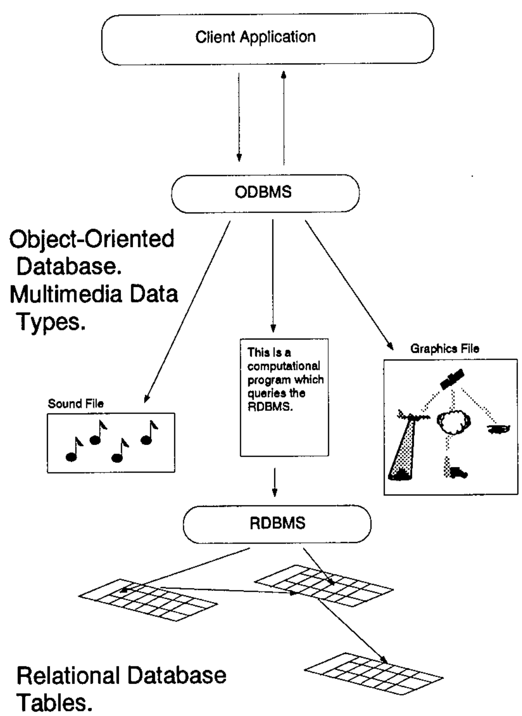


Figure 2.11 Multidimensional Database Example

The desire for the ability of a single client application to access more than one database motivates the creation of distributed databases. A properly implemented distributed database system has the following characteristic(11):

> ...full support for distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases, managed by a variety of different DBMSs, running on a variety of different machines, supported by a variety of different operating systems, and connected by a variety of different communication networks–where the term *transparently* means that the application operates from a logical point of view as if the data were all managed by a single DBMS running on a single machine.

In a distributed database system, the client may access a remote object database and retrieve a program which calls data from another remotely located relational database. The program may be processed at the client's machine, the local host, or on a remote host, meaning not only is the information distributed, but the processing is also distributed. This sharing of information and processing has the advantage of using resources efficiently, if managed properly. An example of a distributed database system is shown in Figure 2.12

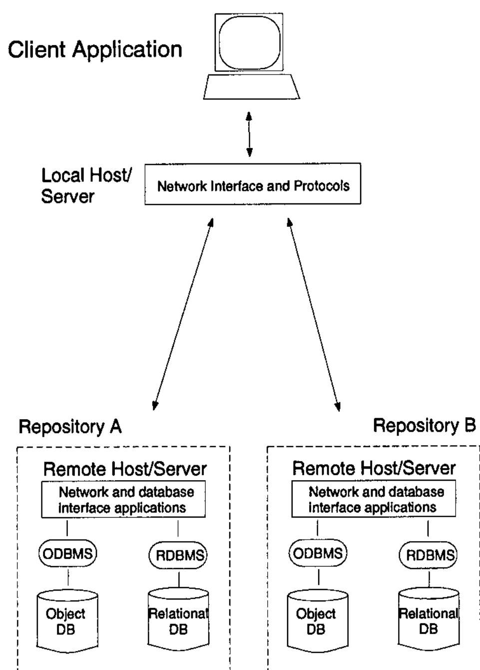


Figure 2.12 Distributed Database Example

2.5 *The Three-tiered Network Architecture*

The client/server terminology of the past becomes vague when discussing distributed systems over the Internet. The client in one transaction could be the server for a client in the next transaction. Therefore, the requester for any transaction is considered the client and the applications making the request and providing the user interface are considered the front-end or client side. The applications which retrieve the information requested by the client are considered to be the back-end. An example of a back-end application is the database management system (DBMS). The software that acts as an intermediary between the client's application program and the DBMS is called *middleware*(7). An example of such middleware is the ODBC. This grouping of the software with some examples shown is illustrated in Figure 2.13

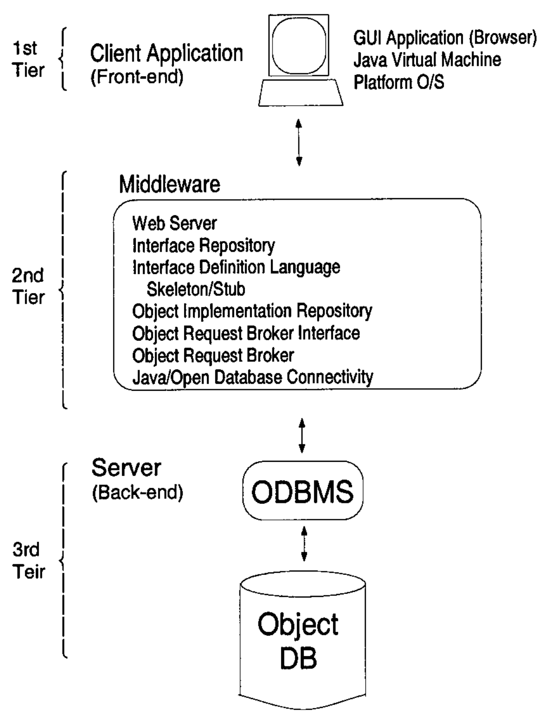


Figure 2.13 Three-tiered Architecture Example

2.6 Summary

This chapter has provided an overview of the operational, technical and system architectures used to model C4I systems. Additionally this chapter describes and motivates the terminology and structure of the Internet architecture which allows interoperability and portability of objects and data, via IDEF modeling, relational and ODBMS, as well as the OMG CORBA standard. The next chapter describes the antenna model and object, creating relational database tables to store the persistent data identified in the antenna IDEF1X model and storing the object-oriented antenna program in an object database. The structure of each tier is then developed to allow the antenna object to be accessed and implemented as part of a client's simulation application.

III. Methodology

3.1 Introduction

The first chapter presents a brief historical background of the problems faced by simulators of C4I systems, and states the problem to be solved. The second chapter presents background material essential to understanding the solution presented in this chapter and the analysis of the architecture in Chapter Four. This chapter presents the methodology used to develop the models, executable objects and distributed architecture for distributed simulations.

3.2 Developing the Operational Architecture

The ER model and activity model are developed jointly to describe the operational architecture. Although each can be developed independently, the process of developing them jointly serves to check that each is complete. Later, when the antenna model is used to develop relational database tables, and object-oriented code, the advantage to developing these IDEF models jointly will become obvious.

In this thesis, the illustrative scenario is defined as the reception of a message using an OE-254()/GRC Antenna Group and compatible receiver. The modeler should identify the agents in this scenario as an antenna, receiver and radiowave propagation path. The receiver object which will require information from the antenna and a path object will pass information to the antenna (see Figure 3.1).

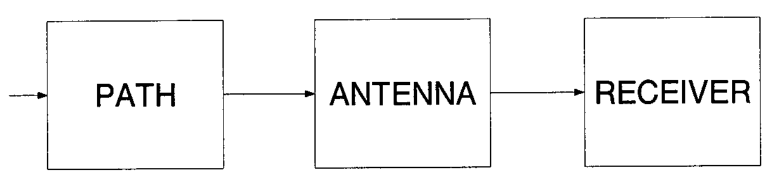


Figure 3.1 Information Flow Diagram

Recall from Figure 1.4, the path is not assumed to be degradation free. To determine if the message was received in a simulated transmission, the link margin is calculated given a required bit energy per noise power spectral density. The link margin is calculated using the Equation 3.1 (see appendix C):

$$M = \left(\frac{E_b}{N_o}\right)_{recvd} - \left(\frac{E_b}{N_o}\right)_{reqd} \tag{3.1}$$

where M is the link margin, E_b is the bit energy, N_o is the noise power spectral density, and all quantities are in dB.

3.2.1 Developing the Entity Relationship Model. IDEF1X is used to produce a graphical information model which represents the structure and semantics of data relevant to an enterprise or system. This model can then be used to develop relational databases which contain the information used by simulation objects, and the embedded SQL code for the host simulation language. Each block in the IDEF1X model represents an entity. An entity represents an item of interest to the modeler from a "relational" perspective. An entity is described by a set of attributes. An attribute is a property or characteristic that is common to some or all of the instances of an entity. An attribute draws legal values from a natural or prescribed domain of values, consistent with the system and scenario being modeled. A domain is a named set of data values all of the same data type, upon which the actual value for an attribute instance is drawn. Every attribute must be defined on exactly one underlying domain. Multiple attributes may be based on the same underlying domain.

Often, entities in the real world are related to other entities. For example, one entity may be the component of another entity. A relationship is an association between two entities or between instances of the same entity. It is important to identify the relationships between entities. The relationships are part of the definition of the system and help to properly manage and access information about the system which may be stored in many entities. IDEF1X models can become quite large as entities and relationships are identified, as shown in the JDBE ER antenna model in Appendix A, but a simple model for this scenario is shown in Figure 3.2 (25).

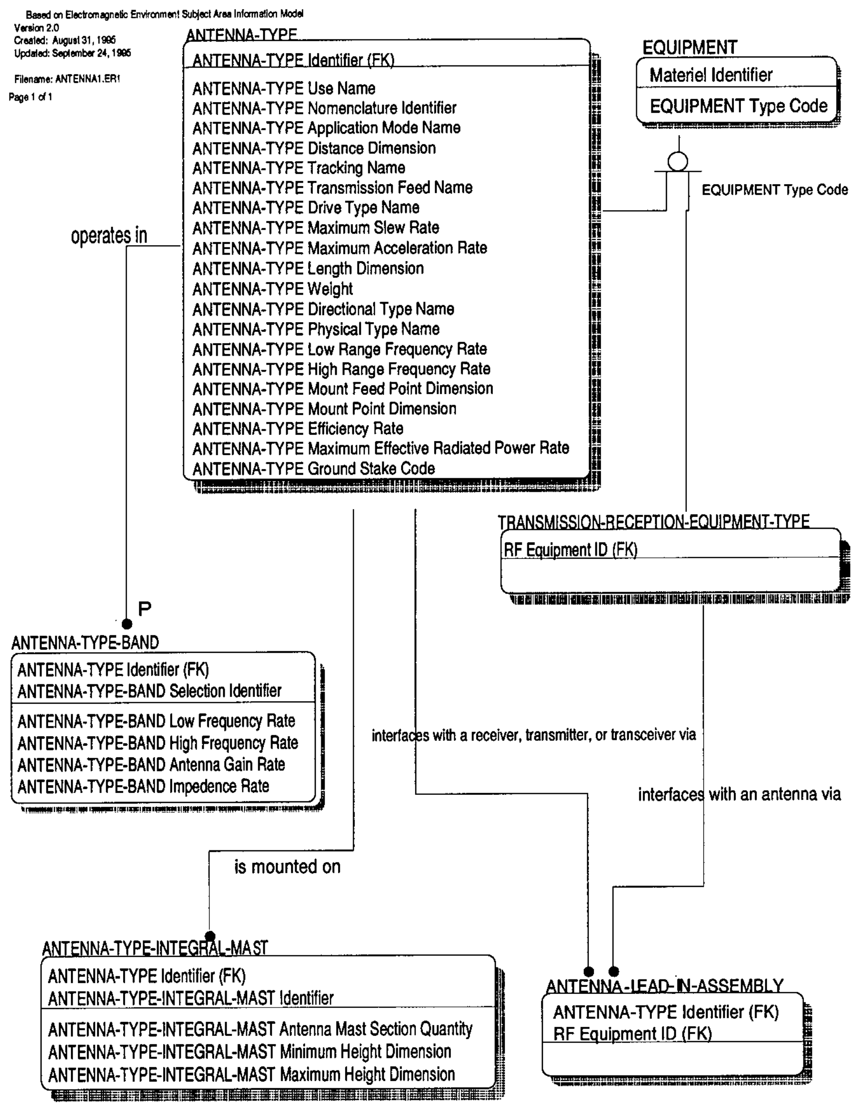


Figure 3.2 IDEF1X Antenna ER Diagram

The IDEF1X diagram supports defining relational database elements which are populated with instances and values representing the real world system being modeled. For example, the

ANTENNA_TYPE table for the OE-254()/GRC is shown in Figure 3.3. All the attribute values represent the persistent data of the antenna system of interest for the level of abstraction required by the scenario (all lengths and distances are in meters, and all weights are in pounds).

ANTENNA-TYPE

Material_ID	5985-01-063-1574
ant_use_name	transmit and receive
Ant_Nomenclature	OE-254()/GRC
Ant_Application	linear
Ant_Distance	13.70
Ant_Tracking	manual
Ant_Transmission	coax
Ant_Drive_Type	
ant_max_slew_rate	
ant_max_accelerati	
ant_length	0.16
Ant_Weight	45
ant_direction_type	OMNIDIRECTIONAL
Ant_Physical_Type	CONICAL-SPIRAL
Ant_Low_Freq	30
Ant_High_Freq	88
Ant_Mount_Feed	10.30
Ant_Mount_Point	5.15
Ant_Efficiency	.75
Ant_Max_ERP	350
Ant_Ground_Stake	Y

Figure 3.3 ANTENNA_TYPE Table for the OE-254()/GRC

Each table captures the persistent data from the technical specifications and operational architecture products (refer to Figure 2.7) relating to the OE-254()/GRC antenna and relevant to the scenario provided. However, the relational database is not an executable system and does not capture the process, function, or activity of the antenna. For this purpose, the IDEF0 activity model is developed.

3.2.2 Developing the Activity Model. The IDEF0 model begins with only one block identifying the complete activity; this is the top-level diagram (Figure 3.4) (26).

The single function represented on the top-level context diagram is decomposed into its major sub-functions as shown in Figure 3.5.

These diagrams describe the basic functions of the antenna required to calculate the link margin based on the scenario provided.

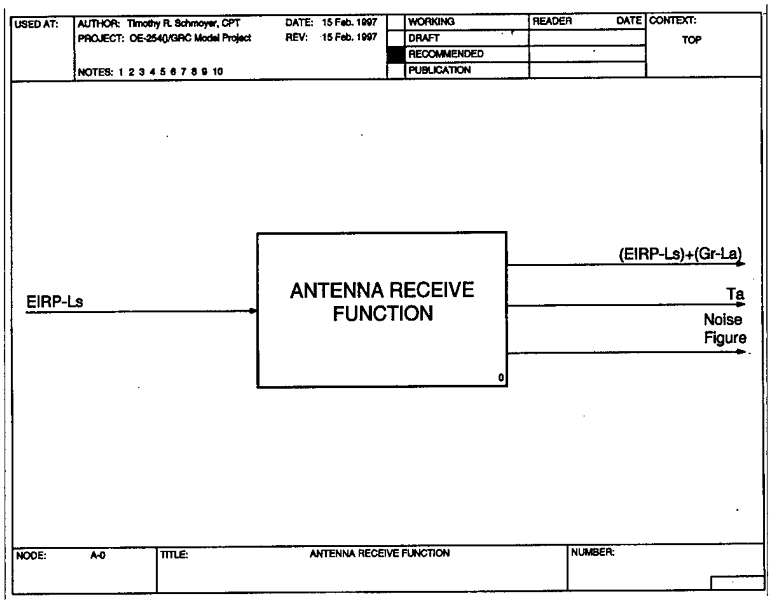


Figure 3.4 IDEF0 OE-254/GRC Top-level (A-0) Diagram

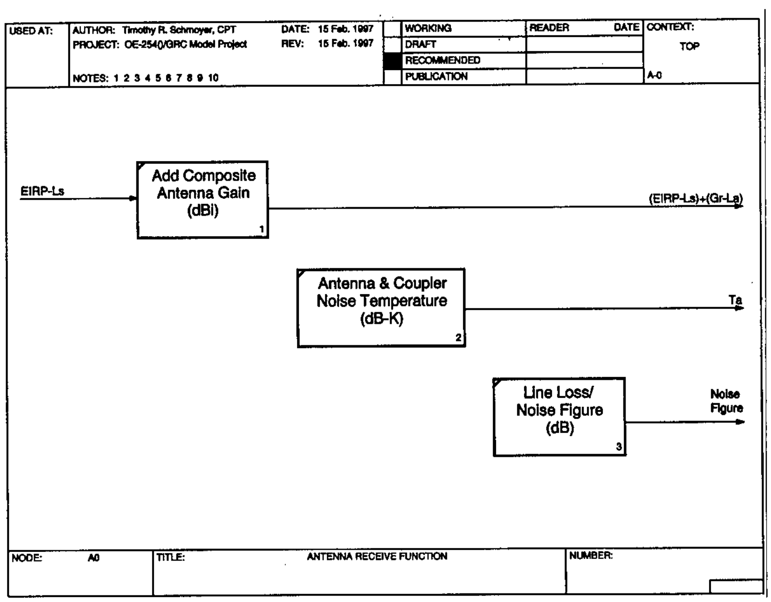


Figure 3.5 IDEF0 OE-254/GRC (A-0) Child Diagram

3.3 *Integrating the Technical Architecture*

The activity model presented in the last section describes the basic functions of the antenna; however, the *constraints* of the technical architecture have not been included. For example, the frequency range of the antenna is specified to be between 30 and 88 MHz by the technical architecture. To achieve operational realism, it would be useful to verify that the signal received across the path has a center frequency in the specified range. A similar constraint could be added to verify that the receiver uses the same frequency range as the antenna and that it is a type of receiver which can use the OE-254()/GRC antenna system. These constraints are included in the activity model as f and Receiver-Type, as shown in Figures 3.6 and 3.7 (see Appendix B for IDEF0 model documentation).

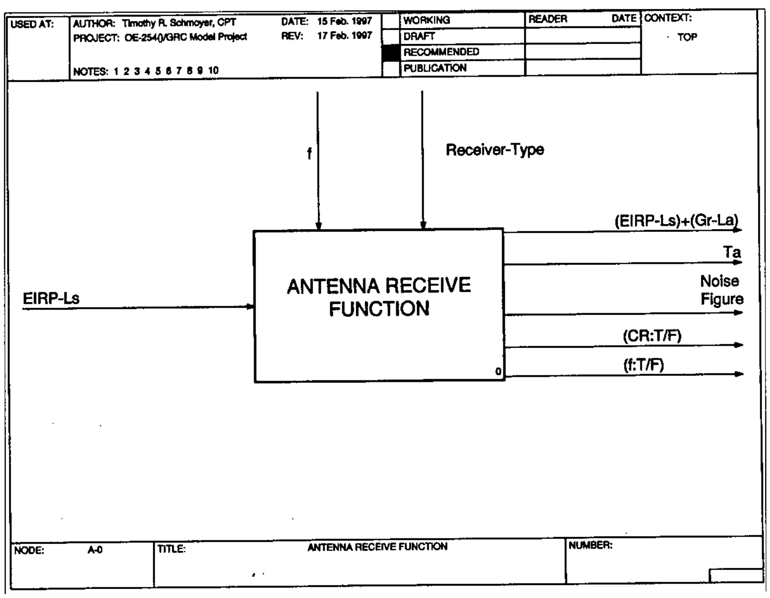


Figure 3.6 IDEF0 OE-254/GRC (A-0) Diagram with Constraint

These constraints ensure that incompatible radio systems with different frequency ranges are not connected across the path in a model, or if they are, the simulation does not allow the radios to communicate.

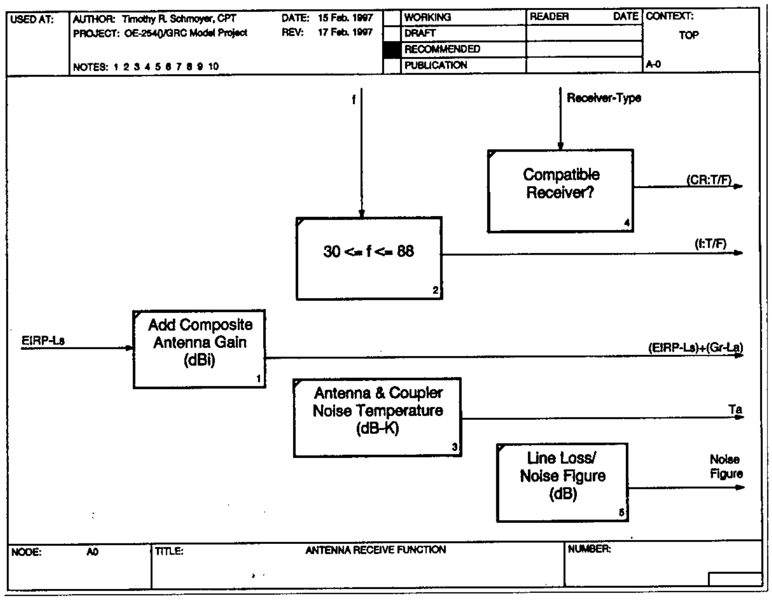


Figure 3.7 IDEF0 OE-254/GRC (A-0) Child Diagram with Constraint

3.4 Developing the System Architecture

The system architecture is designed to meet the requirements and functions of the operational architecture within the constraints of the technical architecture. An example graphical representation of the OE-254()/GRC system architecture is shown in Figure 3.8 (Graphic modified from TC24-24). The input parameters from the path, specified by the operational architecture ($EIRP - L_s$, f, and Receiver Type), are included on the graph by the antenna section. The components of the antenna, the antenna's frequency range, loss characteristics and cable length are specified by the technical architecture. The system architecture is the combined view dictating the height required to receive the signal (within the height specified by the technical architecture), the number of mast sections required, the distance between the antenna and receiver based on site conditions, etc.

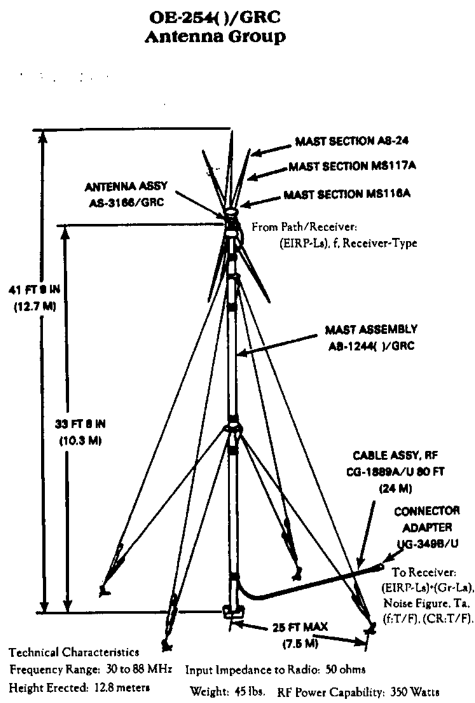


Figure 3.8 System Architecture of OE-254/GRC Antenna System

3.5 Coding the OE254_Path Object/Applet

The OE-254()/GRC antenna model is then coded as an executable object using an object-oriented programming language. The executable applet can be referenced from a client's GUI application and implemented at run-time as a complete simulation, or as part of a larger simulation. The object-oriented language chosen is not important, since the Common Object Request Broker Architecture discussed later will allow objects written in different languages to communicate fluidly in a distributed simulation environment. The salient point from this discussion is the use of embedded SQL and OQL to allow the object-oriented program to access persistent objects or data and perform the computational logic over a distributed network. For example, the technical architecture constraints are enforced by using the data in relational and object-oriented databases. Therefore, the OE254_Path antenna object would use the data in these databases to verify that the parameters it receives from the path and receiver objects are in fact valid values.

Assume an antenna class already exists and has been defined. The first step is to create and initialize the OE254_Path object(18, 37, 11):

```
Antenna OE254_Path = new Antenna();
```

Now, to get the frequency range of the antenna from the ANTENNA-TYPE table in the Relational database:

```
/* connect to the database system using the database connectivity method, location,
/* id, password and name of the database.
database.connect("ODBC", "FTGORDON", "myname", "mypwd", "ANTENNA")
if (!database.connected()) {
write ("Unable to connect to the database.")}
    ⋮
/* SQL to get frequency range
select ANTENNA.Ant_Low_Freq
into :LowFreq
from ANTENNA_TYPE
where ANTENNA.Ant_Nomenclature = "OE-254()/GRC"
    ⋮
select ANTENNA.Ant_High_Freq
into :HighFreq
from ANTENNA
where ANTENNA.Ant_Nomenclature = "OE-254()/GRC"
/* end SQL
    ⋮
if (PathFreq < LowFreq OR > HighFreq) {
write ("Received frequency is not within operating range of the antenna."}
```

Also we want to check an array named *receivers* in an object-oriented database to verify that the receiver is compatible with the antenna(9):

```
/* SQL to get NSN of the receiver
select ANTENNA.Material_ID
into :OE254ID
from ANTENNA_TYPE
```

```
where ANTENNA.Ant_Nomenclature = "OE-254()/GRC"
select RF_EQUIPMENT.Material_ID
into :Receiver
from ANTENNA_LEAD_IN_ASSY
where ANTENNA.Material_ID = OE254ID /*NSN of OE-254
/*end SQL
    ⋮
database.connect("ODBC", "FTGORDON", "myname", "mypwd", "COMPATIBIL-
ITY")
if (!database.connected()) {
write ("Unable to connect to the database.")}
    ⋮
/* OQL to check array
if (!exists Receiver in COMPATIBILITY.receivers : COMPATIBILITY.antenna = OE254ID)
{ write("Receiver not compatible with OE-254()/GRC antenna group.") }
/* end OQL
```

The embedded query languages serve two purposes. Once the program has been written, changes to the persistent data do not require changes to the program and subsequent recompiling and linking. The reliance on the database improves data management and use of the current standardized values.

3.6 Storing the OE254_Path Object in the Object Database

The antenna object is stored in the object database with a unique object identity (OID). Object identity is the property of an object that distinguishes it from all other objects(19), even similar or duplicate objects. Each object in the database must have a unique identity for the ODBMS to recognize which object is being requested. This is especially important when many duplicate objects are being used in a scenario. For example, in a communication network, several OE-254()/GRC antenna systems may be employed. The ODBMS must be able to distinguish each duplicate object as a unique instance of an OE-254()/GRC antenna system.

Objects may be shared by other objects. For example, a graphic file with a universal antenna symbol may be shared by all antenna objects for display in a client's GUI simulation application. In some directory systems (e.g., DOS), to share a file among directories, the file must be copied to each directory. To avoid the redundancy that results and the requirement for the user to maintain consistency by making changes to each copy every time an update is made, the identity method must allow linking between objects.

Several methods are appropriate to ensure a unique OID. One method is to use the name of the object's creator, birth place of the file, and a file name. A global repository identifies each object this way, and the birthplace repository tracks the location of the file in case it is ported to another location. Type-State-Identity Trichotomy (19) is also used to uniquely identify objects. Using this method, the antenna object is identified by its class, instance and state. For example, the OE-254()/GRC antenna object is stored as OE254_PATH and the OID "path" notation is 24thIDCmdNet/NodeE1/OpsSINCGARS/ANTENNA/RECEIVE/OE254_PATH (see Figure 3.9). If the antenna object is copied, or ported to another location, it is still uniquely identified by changing the name of the net, node, or communication system.

3.7 Using a CORBA ORB to Request and Implement Objects

The OE254_PATH object is made available to distributed simulation clients using the CORBA shown in Figure 3.10 (27). The following sections explain the function of each component, from requesting the object in the client's simulation application through returning results to the simulation.

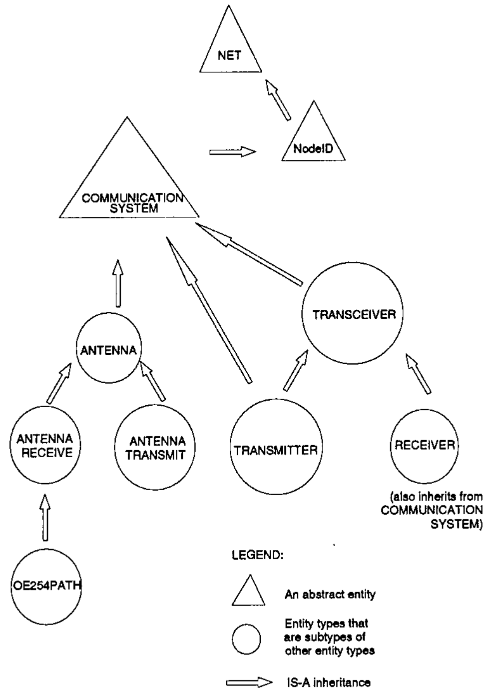


Figure 3.9 Semantic Model for a Communication Net.

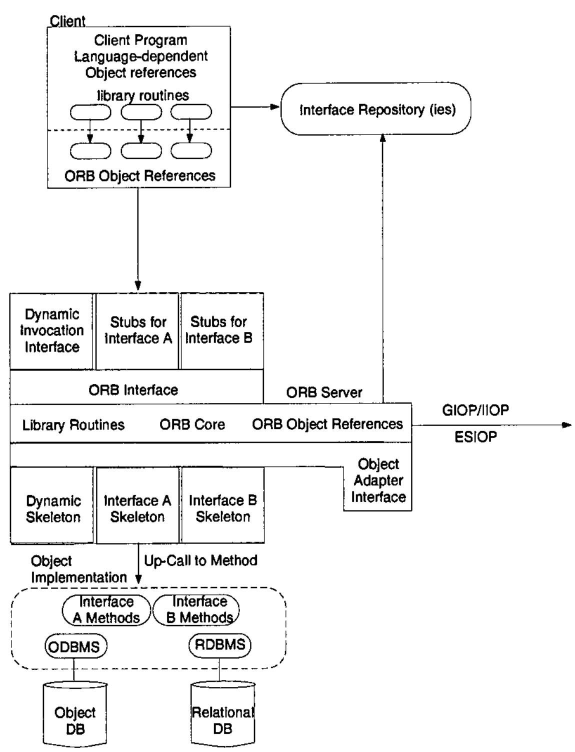


Figure 3.10 The Common Object Request Broker Architecture for a Distributed Simulation

3.7.1 Requesting the Ant_Path Object from the Simulation. The simulation executing on a client requests an object using an object reference. An object reference is a token that may be invoked or passed as a parameter to an invocation on a different object. Invocation of an object involves specifying the object to be invoked, the operation to be performed, and the parameters to be given to the operation or returned from it. The client knows only the logical structure of the object according to its interface, and experiences the behavior of the object only through the object's invocation. Clients access object-type-specific stubs as library routines in their program (see Figure 3.11). The client program thus sees routines callable in the normal way in its programming language. This language-specific data type to refer to objects is often an opaque (hidden) pointer. The client then passes that object reference to the stub routines to initiate an invocation(27).

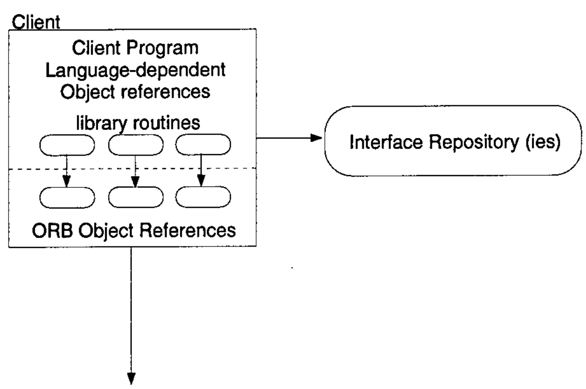


Figure 3.11 The CORBA Client Architecture

Clients generally see objects and ORB interfaces through the perspective of a GUI and language mapping. Clients have no knowledge of the implementation of the object, which object adapter is used by the implementation, or which ORB is used to access it. The client may only see drop down menus which allow a selection of resident object references (and stubs) to be chosen. In simulation and modeling software this is very common. Only after the modeler has developed the system model is the simulation code generated, compiled, and run (see Figures 3.12 and 3.13).

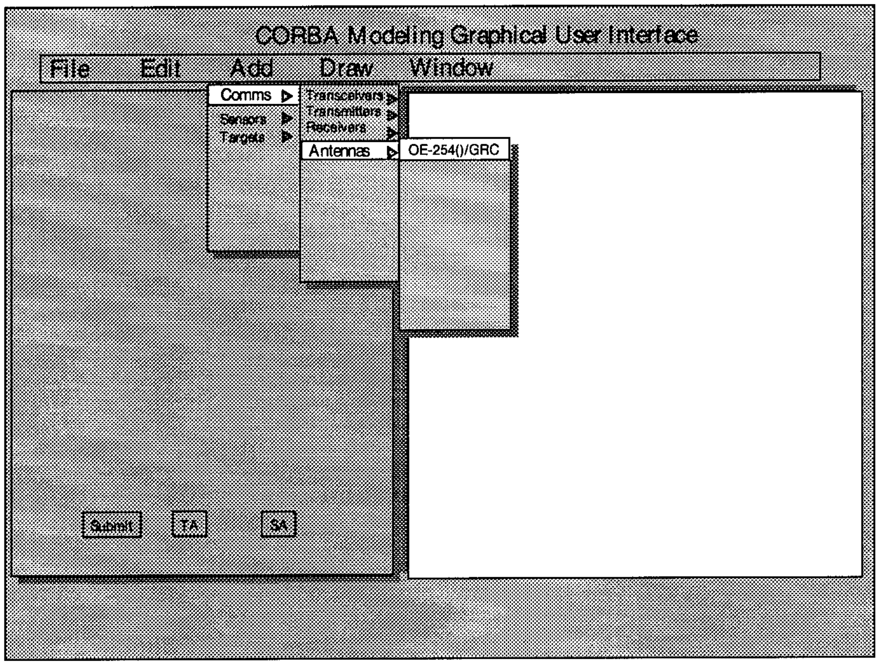


Figure 3.12 Example of a Client GUI to Select Objects for Simulation

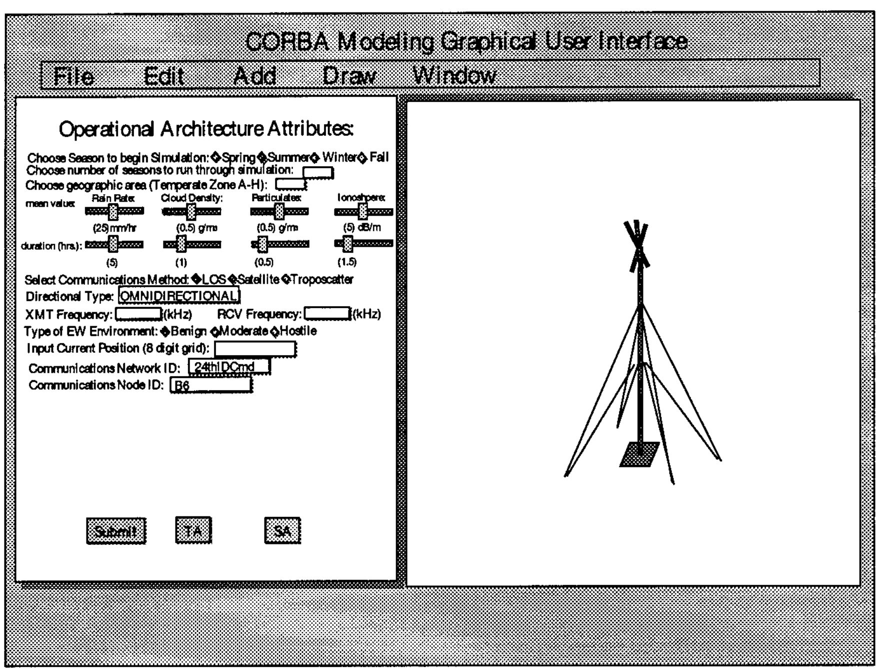


Figure 3.13 Filling in the Parameters for the Operations to be Performed

3.7.2 Processing the Request. The Object Request Broker (ORB) receives the request from the client through the stubs or the Dynamic Invocation Interface (DII) (Figure 3.14). The stubs have access to the object reference representation and interact with the ORB to perform the invocations. The stubs make calls on the rest of the ORB using interfaces that are private to, and presumably optimized for, the particular ORB Core. The ORB core provides the basic services of control, representation of objects and communication of requests. The operations of the ORB, which serve as an interface between the client's request and implementation of the requested object are:

- Operations that are the same for all ORB implementations
- Operations that are specific to particular types of objects
- Operations that are specific to particular styles of object implementation

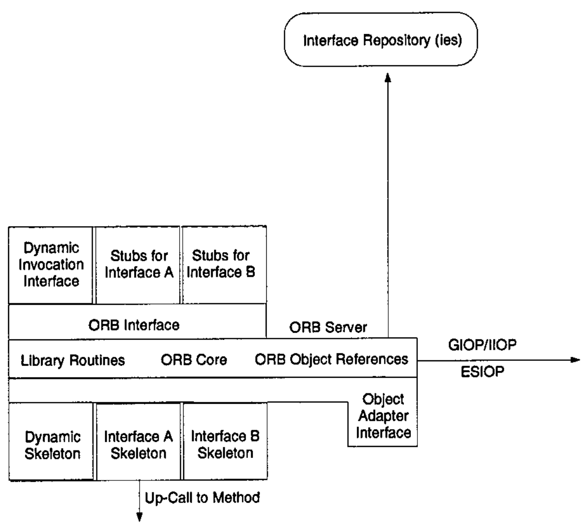


Figure 3.14 The CORBA ORB Architecture

An interface is also available that allows the dynamic construction of object invocations, that is, rather than calling a stub routine that is specific to a particular operation on a particular object, a client may obtain the information required for an object reference from an Interface Repository. The Interface Repository is a service that provides persistent objects that represent the Interface

Definition Language (IDL) information in a form available at runtime. In addition to its role in the functioning of the ORB, the Interface Repository is a common place to store additional information associated with interfaces to ORB objects. For example, libraries of stubs or skeletons and routines that can format or browse particular kinds of objects might be associated with the Interface Repository. This provides the client with the ability to search for a simulation object and include its object reference in the simulation model(27).

The operations that are the same for all ORB implementations are sent through the ORB Interface. These operations include converting the object reference to a string for storage, duplicating and releasing copies of object references to clients and implementations, equivalence checking and object reference identity(27).

There are a wide variety of ORB implementations possible within the CORBA. The implementations which best support distributed simulations are the Server-based ORB and the Library-based ORB. The Server-based ORB centralizes the management of the ORB and all clients and implementations can communicate with one or more servers whose job it is to route requests from clients to implementations. In some cases, where the objects are "light-weight" with regards to processing and storage, and their implementations can be shared, the implementation might actually be in a library. In this case, the stubs could be the actual methods. For example, the client could request the OE254 simulation, which would request information from the relational database and return results to the client. This is a Server-based ORB. Or the client could run the simulation as a resident program, which invokes objects to access the relational database(s). This assumes that the client's privileges allow access to the data for the objects and that the implementation trusts the client not to damage the data in the relational database(27).

The ORB also provides interoperability to other ORBs through the General Inter-ORB Protocol (GIOP) and Environment-Specific Inter-ORB Protocols (ESIOP). The Internet Inter-ORB Protocol (IIOP) specifies how GIOP messages are exchanged using TCP/IP connections. The DCE

ESIOP specifies how CORBA ORBs communicate with The Open Group's Distributed Computing Environment (DCE).

3.7.3 Accessing the object/data. The ORB determines if the object reference is implemented through another ORB or not. If the object is implemented locally, the object adapter performs the management functions for the object implementation. The Basic Object Adapter (BOA) supports convenient interfaces for generating object references, registering implementations that consist of one or more programs, activating implementations, and authenticating requests. It also provides a limited amount of persistent storage for objects that can be used for connecting to a larger or more general storage facility, for storing access control information, or other purposes.

The Object-Oriented Database Adapter performs the task of accessing the ODBMS where the OE254_Path object is stored and passing any control or authentication information required by the DBMS. During the execution of the antenna object, the object adapter is responsible for generating object references and passing them to the ORB if other objects are requested by the antenna object. The object adapter also informs the ORB when execution is complete, so the ORB can pass control back to the client with the results(27).

For a particular language mapping, and depending on the object adapter, there will be an interface to the methods that implement each type of object. There is also a Dynamic Skeleton Interface (DSI) which is the server side analogue to the client side DII. The DSI is a way to deliver requests from an ORB to an object implementation that does not have compile-time knowledge of the type of the object it is requesting.

3.7.4 Implementing the Object. The OE254_PATH object is implemented as an *executable module* or by a file server integrated with the DMBS(19) (Figure 3.15). The implementation is transparent to the client and distributed processing is supported easily in large simulations through the distributed database storage of simulation objects. Control is maintained by the ORB(s) and

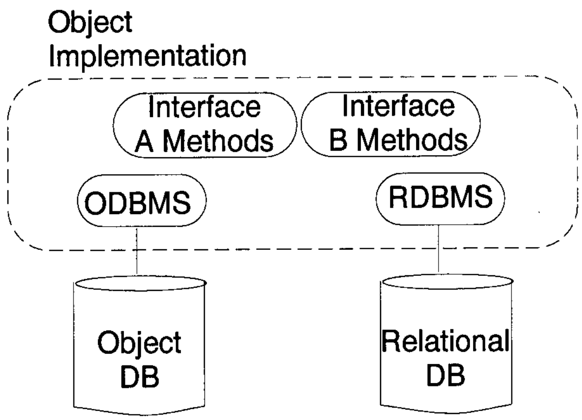


Figure 3.15 The CORBA Object Implementation Architecture

DBMS until all requested objects have been executed. When execution is complete, the results are returned to the object adapter and any locks are removed.

3.7.5 Returning Results to the Simulation. Results are returned to the client by reversing the process by which the request was made. The results are provided to the ORB by the object adapter. The stubs provide the language mapping to provide the results to the client. Regardless of what language the object was implemented in or what operating system the client is using, appropriate implementations of the CORBA architecture allow distributed processing and heterogeneous distributed database access over possibly several operating systems in the process.

3.8 Summary

Objective 1-1 of the DoD Modeling & Simulation Master Plan states(28):

- Establish a common high-level simulation architecture to facilitate the interoperability of all types of models and simulations among themselves and with C4I systems, as well as to facilitate the reuse of M&S components.
  - Simulations developed for particular DoD Components or Functional Areas must conform to the High Level Architecture.
  - Further definition and detailed implementation of specific simulation system architectures remain the responsibility of the developing Component.

The CORBA distributed simulation architecture described in this chapter, fully supports the objective stated above. The mapping of programming languages to OMG IDL facilitates interoperability of all types of models and simulations. Additionally, the standardized development of models using the IDEF method provides a "paper trail" process to approve the further development of simulation objects for use in the distributed network. All objects of the same type would be required to provide similar in/out parameters, or the interface between specific simulations would be clearly defined and provided as a stub, skeleton and/or object adapter. Simulation objects are reusable by referencing the object during simulation development and their code can be copied or ported for local use by using a unique OID naming system.

Each Service Component can maintain a federated interface and implementation repository for the simulation objects they maintain. These repositories can be further federated to organizations responsible for specific C4I systems. For example, within the U.S. Army, Fort Gordon would maintain communications repositories and Fort Huachuca would maintain the repository for intelligence systems.

IV. Analysis

4.1 Introduction

The first chapter presents a brief historical background of the problems faced by simulators of C4I systems, and the exact problem to be solved is stated. The second chapter presents background material essential to understanding the approach Chapter Three presents. This chapter presents an analysis of the methodology and determines which goals were achieved and which were not.

4.2 Background

In 1991, the Joint Analysis Directorate of the Joint Chiefs of Staff cataloged over 700 simulations, war games, exercises, and models in general use throughout the U.S Department of Defense, as well as in Australia, Canada, England, and Germany. There was almost no ability for these simulations to interact and no evidence was found that communications availability and fragility was included in any of the simulations(24).

Only in recent years, have computer scientists begun to write interfaces for disparate simulations. Pratt and Johnson's work(31) in 1993 to interface the Janus Combat Model and DIS is an example of a simulation specific interface. However, there is no flexibility for handling objects written in varying programming languages or to take advantage of distributed processing across the network. The World Modeler application they developed only provided translation and control for Janus and DIS. The CORBA distributed simulation would use client stubs and server skeletons to perform language translation to and from OMG IDL, as well as providing network management and transaction control.

In 1995, the U.S. Army Communications-Electronics Command introduced the Integrated Terrain-Environment-Multipath Model to add communications realism to simulations using radiowave communications systems in their models. A distributed simulation architecture would make this program available for inclusion within other client simulation applications.

4.3 Summary

As stated in the first chapter, the problem to be solved by this thesis is to develop a methodology using a standardized modeling process which integrates the technical, operational, and situational architectures, and results in a distributed simulation architecture by which current systems can be evaluated and future systems analyzed. The proposed methodology is illustrated in Figure 4.1, and is described below.

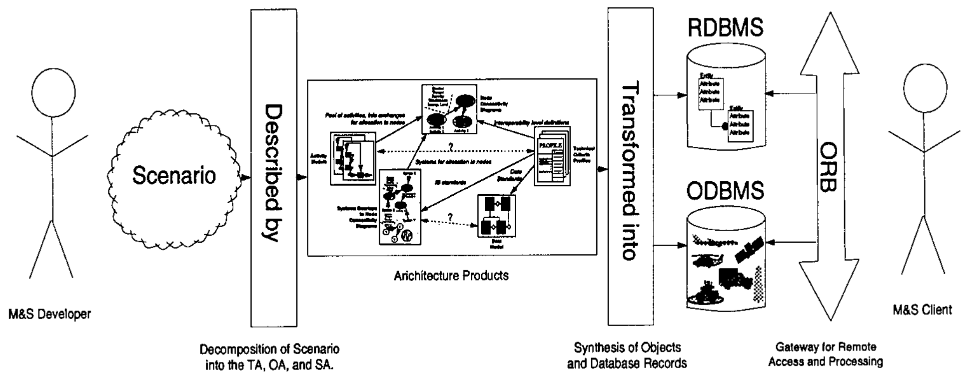


Figure 4.1 Developing the Model, Simulation Object,and Distributed Simulation Network

An antenna, the OE-254()/GRC, was modeled using IDEF modeling methods to describe the operational architecture of the antenna given a common scenario. By including the technical architecture constraints in the IDEF models, these architectures were integrated to describe the system architecture. This standardized process provides the foundation for developing the object-oriented code for an executable simulation object. The development of the object-oriented code could be further mapped using the IDEF4 object-oriented design method not discussed in this thesis. A brief overview of IDEF4 by Knowledge Based Systems, Inc. is provided in Appendix D.

The persistent data describing the OE-254()/GRC antenna from the IDEF1X model is then stored in a relational database. Embedded SQL in the executable simulation object is used to apply the technical constraints. Other objects could be embedded with the antenna object, or the antenna object could be embedded in a larger communication system. Embedding objects only requires the

CORBA object reference as a pointer within the host program to invoke and implement the object. The storage of the simulation object using an unique OID in an ODBMS was then discussed.

Finally, an approach to achieve a fully distributed simulation architecture using the CORBA was conceptually provided by stepping through each of the component blocks. The ability to access heterogeneous distributed databases and perform distributed processes was discussed as part of the capabilities, as well as support for portability and reuse of simulations and interoperability across operating systems and programming languages.

4.4 Review of the Results

The value of this research is the feasibility of implementing the methodology described in Chapter 3, and the integration of current modeling techniques with emerging distributed object technology. The results of this research are described below:

- The modeling standardization process proposed in the research uses the Reverse Engineering for Data Integration and Sharing (REDIS) Methodology currently incorporated in the DMSO Modeling and Simulation Resource Repository (MSRR) standards. REDIS provides a solution to the lack of common or standard data definitions in the M&S community.
- The IDEF0 modeling method is used to describe the behavior of the simulation object, its functions, methods and constraints. The IDEF0 model clearly defines the input parameters and output parameters required by the operational architecture to simulate the given real-world scenario. The IDEF0 and IDEF1X (ER) models provide the historical documentation necessary to validate the model of the system to be simulated. The advantage is once the model has been validated, it can be reused by others in the M&S community without the time and expense of re-validating the model.
- The REDIS ER product can then be used to define a relational database to store and manage data for the real world objects to be simulated. The relational model provides a strong

foundation for the definition and management of the persistent data, which is not available in current object models. This moves the information from the bookshelf into a standardized format which can be easily accessed and reused. By incorporating the data in the simulation object programming language through embedded SQL, the data comes alive in the simulation. The data is no longer just reference material, but is put to use validating input parameters, providing output parameters to other objects, and enforcing technical constraints. The integration of relational and object-oriented modeling allows us to take advantage of the strengths in each model while avoiding some of the weaknesses.

- The development of modular object-oriented simulation objects serves two purposes. Objects can more realistically represent real-world systems. Inheritance, embedding or linking objects through their methods, OQL ,and ORB object references (pointers), allow a hierarchical and multidimensional representation of the real-world scenario. Additionally, modular objects can be created at the fidelity required by a simulation and reused as components by other simulations.

- By placing the simulation objects in an object-oriented database and using a CORBA compliant ORB, the distributed simulation network allows simulation clients to find and reference simulation objects through the Interface Repository. The ORB provides the means to manage network connections and processing resources for the distributed processing of multiple objects in a simulation, or to allow the client to port simulation objects to a local processor. Simulation objects can then be reused in many simulation applications. In the distributed processing implementation, the client references the objects used by the simulation without regard to the programming language the object or application uses or the operating platform capabilities of the client. The client simply references the simulation object(s) using the information provided by the interface repository and receives the results from the ORB.

- Additionally, the combination of emerging database and Internet technologies into a unique infrastructure for distributed simulation is realistic. The CORBA is an industry standard developed by the OMG consortium. It is supported by the Object Management Database Group (OMDG) consortium and many applications using CORBA compliant ORBs are currently being demonstrated, or are in use today(36). Also, many database software companies are moving to make their DBMSs CORBA and Internet accessible(39). The framework recommended in the methodology is achievable using commercially developed products such as; LogicWorks ERWin and BPWin; KBSI AI0WIN, SmartER, and ProSim™ Oracle 7 Relational Database; Object Design's ObjectStore; and DEC's ObjectBroker, HP's ORB Plus, or IONA's Orbix. This is not a complete listing of all the products available which could be used to implement the methodology, nor is it an endorsement of any product. However, it is included to support the feasibility of implementing the proposed distributed simulation framework.

4.5 Comparison with Other Distributed Simulation Architectures

4.5.1 Common Operational Modeling, Planning, and Simulation Strategy (COMPASS). COMPASS Technology is a middleware application which interfaces existing M&S programs by formatting data into Distributed Interactive Simulation (DIS) Protocol Data Units (PDU)(1). DIS protocols can be exchanged with any other program that implements the DIS protocol interface standard over a network.

At the moment, DIS is working well. However, DIS captures entity interactions in a flat (non-hierarchical manner). Since DIS PDU's cannot inherit from each other, interactions among entities must be individually developed. This means that every entity must have a set of PDU's developed for every other entity with which it must interact. DIS also suffers from a lack of abstraction. Every entity in DIS exists independently and equally. This means that DIS fails to exploit hierarchical structures and commonalities which exist among varying real-world entities(30).

The proposed methodology exploits the object-oriented model and uses distributed object processing, database management and the CORBA to build the distributed simulation network. Object-oriented design allows the structured characterization of entities based on the hierarchical relationships of real-world systems, and inheritance from higher (base) classes to lower (sub) classes derived from the base classes. The object-oriented model also allows the encapsulation of data, and linking (or embedding) of objects, to represent complex behaviors, instead of crafting an entirely new DIS PDU. In fact, DIS itself could be encapsulated in an object using this paradigm.

COMPASS is successful in providing interactivity between current C4I systems using the DIS PDU and the Internet. However, the proposed methodology provides:

- the scalability of object-oriented design not found in DIS PDU based simulation systems,
- scalar data definition and management using the relational model,
- data independence by integrating the two,
- and interoperability with platform and programming language independence.

4.5.2 Joint Modeling and Simulation System (J-MASS). Although the methodology presented was developed independent of any knowledge of J-MASS, the simulation architectures are strikingly similar. The major efforts of J-MASS have been(23):

- Development of a common digital simulation architecture
- Definition of standard interfaces
- Application of commercial standards (e.g., POSIX, MOTIF, X-Windows, CORBA)
- Visual Computer Assisted Support Environment (CASE) tool environment
- Application of modeling and simulation to the acquisition lifecycle

The J-MASS is currently a software application that runs on SUN and SGI machines. It uses an ODBMS for object storage and does not use relational databases. The architecture proposed in the methodology integrates relational and object-oriented databases to strengthen the underlying model and simulation repositories. Also, J-MASS does not use a commercially standard CORBA, but has developed a unique CORBA-like architecture called the Simulation Support Environment (SSE) Interconnect Backplane (IBP). Since J-MASS states individual organizations would solicit solutions from the commercial marketplace for unique requirements and capabilities, these products would have to be created or existing products made "J-MASS compliant". The added J-MASS compliance means added cost to the client. Finally, J-MASS does not specify a process for developing DoD standardized models of real-world systems, as the proposed methodology does with the technical, operational, and system architectures, and IDEF modeling methods. J-MASS does require objects representing those real-world systems to be structured compliant with the J-MASS Software Structural Model (SSM). The SSM enforces software structure and interface standards for all levels of object decomposition to ensure any object in the J-MASS system can be syntactically "connected" with any other objects in the system with guaranteed success.

Since J-MASS is current technology developed by the U.S. Air Force, the ODISC4 should use this system as a basis for developing a CORBA compliant distributed simulation architecture, enforcing the standardize model development process, and employing the multi-database architecture proposed in the methodology.

4.6 Goals Achieved

The goals achieved by the proposed methodology are:

- a process for standardization of C4I data models,
- integration of the technical and operational architectures into the simulation,
- historical records of data collection and process description, and

- distributed database access and processing using CORBA.

CORBA supports heterogeneous distributed systems incorporating object-oriented databases, relational databases, operating systems independence, and programming language independence. As a result, requirements for portability, re-usability and interoperability described in the DMSO HLA can be achieved using this architecture.

4.7 Goals Not Achieved

The original intent of this research was to implement a distributed simulation network using a CORBA compliant ORB and the modeling standardization, object development, and database techniques described by the methodology. Additionally, the simulation to be run over the distributed network was to include time-dependent weather characteristics found in a desert environment. The research conducted to include weather in a C4I simulation utilizing radiowave communications is included in Appendix C. This goal proved too ambitious given the time and resources required to complete such a goal. However, the need for a clear methodology to construct such a network motivated the research and publication of this work.

V. Recommended Continuing Research

5.1 Introduction

The first chapter presents a brief overview of the problems faced by simulators of C4I systems, and the problem addressed by this research was stated. The second chapter presents background material essential to understanding the process and infrastructure described in Chapter 3. Chapter 4 presents a brief summary of the methodology and discussion of its availability and applicability. This chapter reviews the problem and the answers provided by this thesis and makes recommendations for further research.

5.2 Review

The development of a multi-database federated repository system, using CORBA to manage the distributed simulation network combines the power of state-of-the-art technologies with time tested simulation applications and data management techniques. The explosive growth of the Internet and the enabling technologies that have grown with it, provide the simulation community with an opportunity to achieve greater interoperability, re-usability and portability. The framework provided by this research sets the foundation for taking greater advantage of this technology.

5.3 Recommended Research

The recommended research is directed at the steps necessary to implement the distributed simulation network described in this research. These are listed as areas of recommended research.

1. Expand the model standardization process by combining IDEF2 (Simulation Model Design) and IDEF4 (Object-oriented Design) methods with the IDEF0 and IDEF1X methods described in Chapter 2.

2. Develop a client GUI Modeling and Simulation application using CORBA object references.

3. Create and populate an Object-Oriented database and Relational database and use the CORBA gateway architecture as a multi-database management system(20).

4. Integrate a file server with the multi-database system to perform distributed processing.

5. Create a simulation of simulations invoking objects over multiple ORBs.

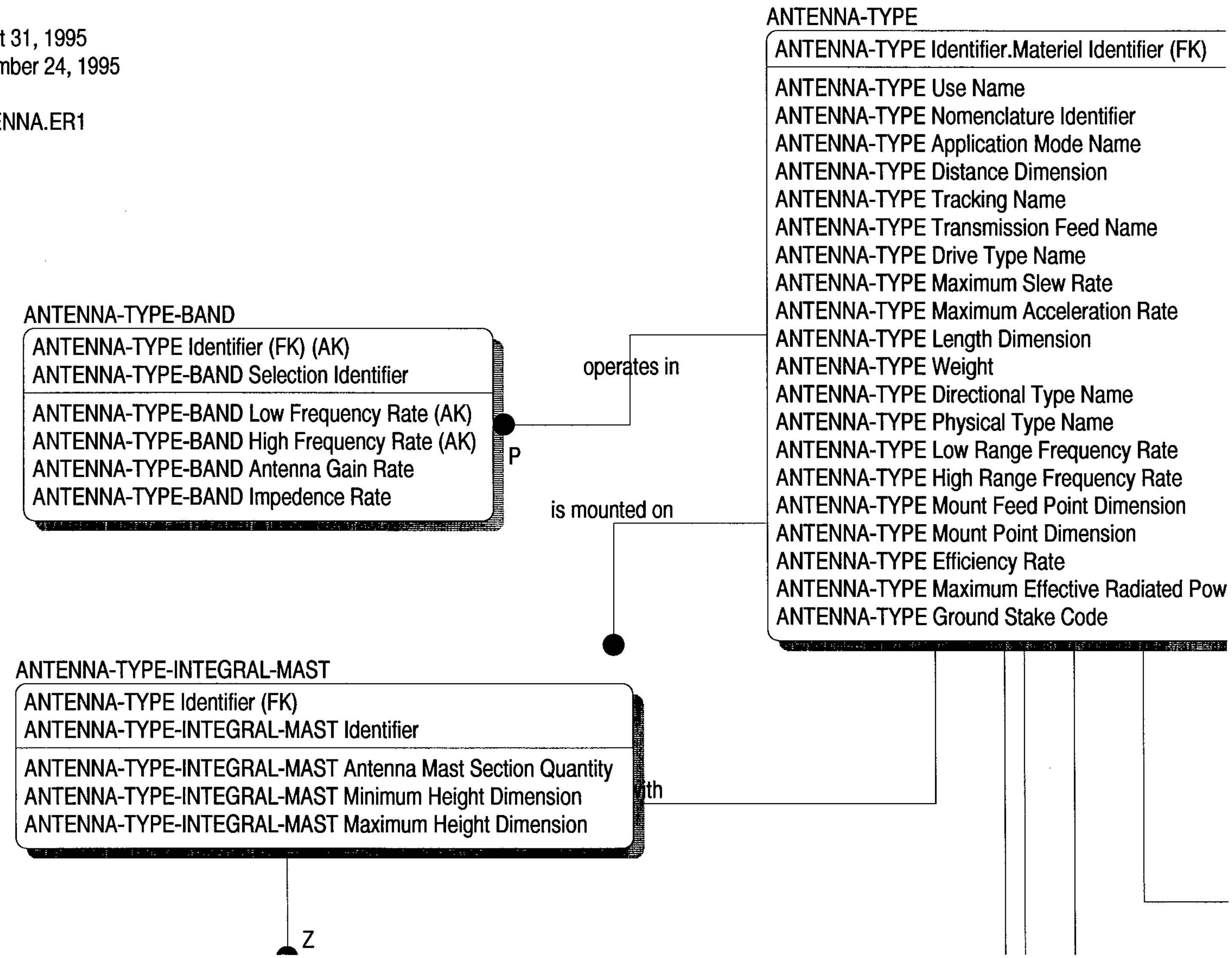
6. Develop a geographic mapping of the distributed simulation network to optimize accessibility and object processing. The distributed capabilities of the recommended architecture require management of resources to optimize the network. Component models and simulations can be developed and stored at facilities which have responsibility for technical and doctrinal development of the systems. Not discussed in this thesis is the structure of a federated model and simulation repository for the Department of the Army. For example, Fort Gordon would administer the CORBA ORB server, relational databases, object-oriented databases and federated interface repository for U.S. Army communications systems. The Global Interface Repository and approval authority for including models and simulations in the federated simulation repository would be controlled by a Defense Department agency.

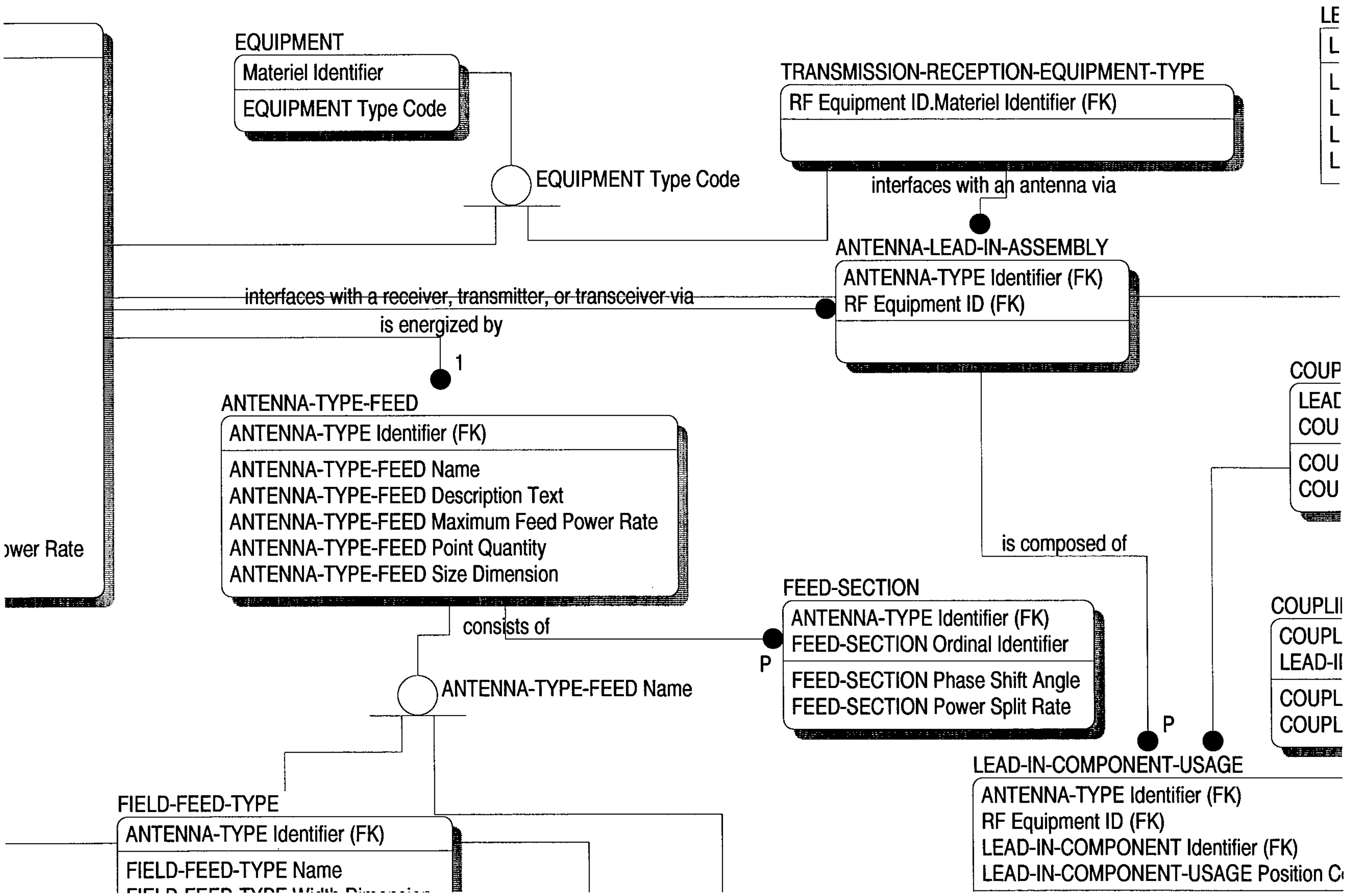
Appendix A. IDEF1X Antenna Model

The following diagrams show the entity relationship model of an antenna, as reverse engineered by the Joint Database Element, Fort Huachuca, and schema for the entities required in this thesis.

Electromagnetic Environment Subject Area Information Model
Version 2.0
Created: August 31, 1995
Updated: September 24, 1995

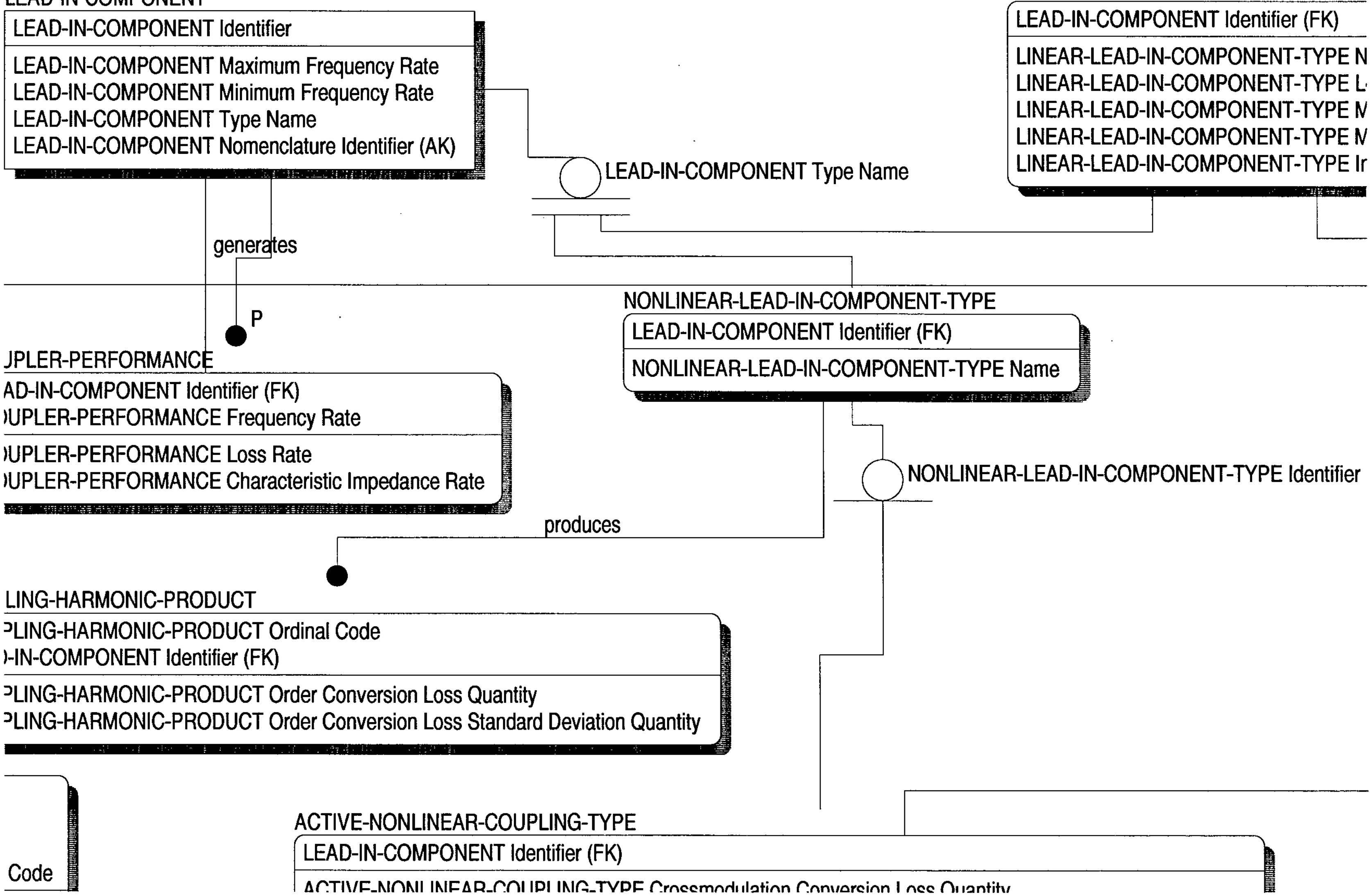
Filename: ANTENNA.ER1
Page 1 of 13

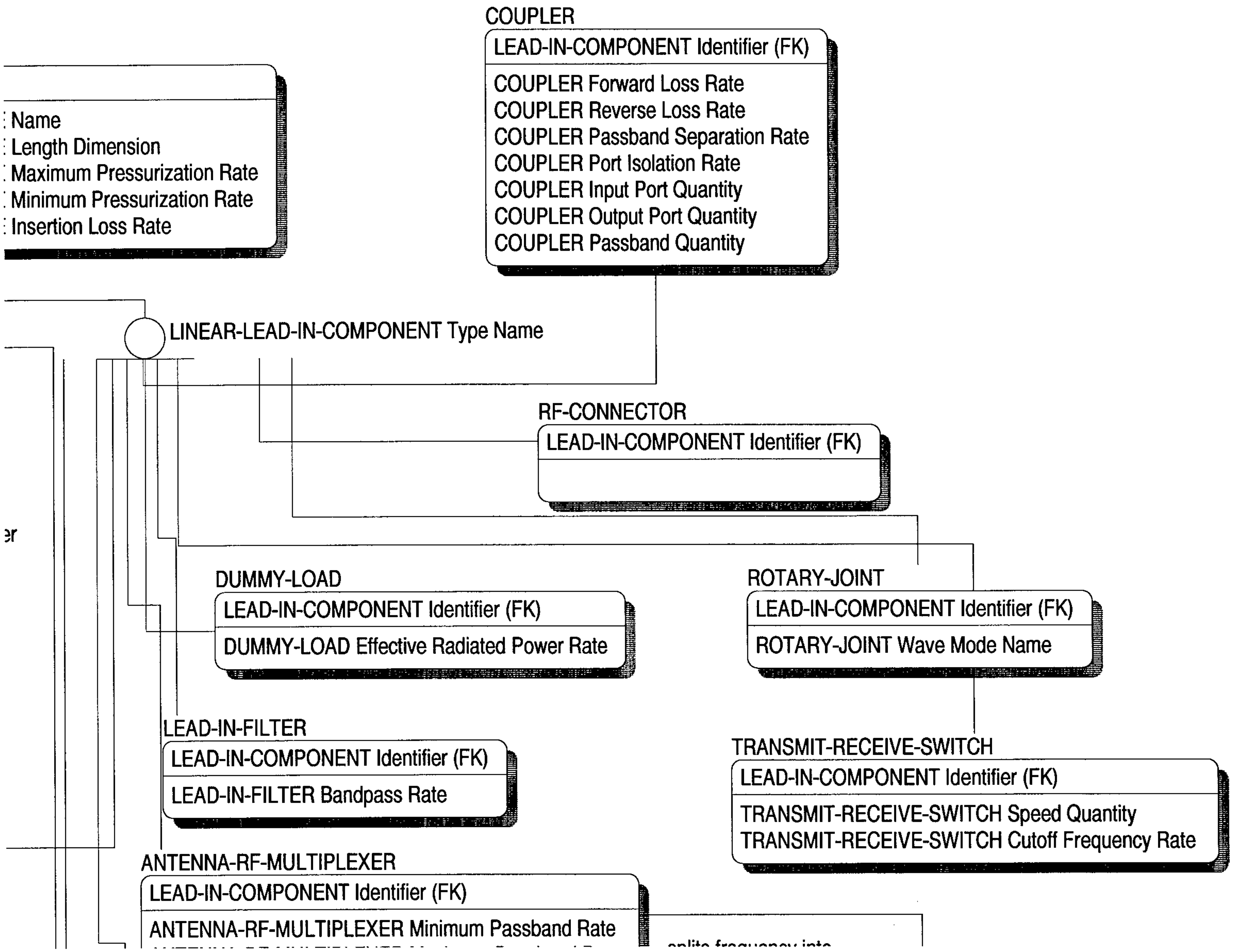



EQUIPMENT
Materiel Identifier
EQUIPMENT Type Code
EQUIPMENT Type Code
TRANSMISSION-RECEPTION-EQUIPMENT-TYPE
RF Equipment ID.Materiel Identifier (FK)
interfaces with an antenna via
ANTENNA-LEAD-IN-ASSEMBLY
ANTENNA-TYPE Identifier (FK)
RF Equipment ID (FK)
interfaces with a receiver, transmitter, or transceiver via
is energized by
1
ANTENNA-TYPE-FEED
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-FEED Name
ANTENNA-TYPE-FEED Description Text
ANTENNA-TYPE-FEED Maximum Feed Power Rate
ANTENNA-TYPE-FEED Point Quantity
ANTENNA-TYPE-FEED Size Dimension
ower Rate
consists of
ANTENNA-TYPE-FEED Name
FEED-SECTION
ANTENNA-TYPE Identifier (FK)
FEED-SECTION Ordinal Identifier
FEED-SECTION Phase Shift Angle
FEED-SECTION Power Split Rate
P
is composed of
P
LEAD-IN-COMPONENT-USAGE
ANTENNA-TYPE Identifier (FK)
RF Equipment ID (FK)
LEAD-IN-COMPONENT Identifier (FK)
LEAD-IN-COMPONENT-USAGE Position C
FIELD-FEED-TYPE
ANTENNA-TYPE Identifier (FK)
FIELD-FEED-TYPE Name

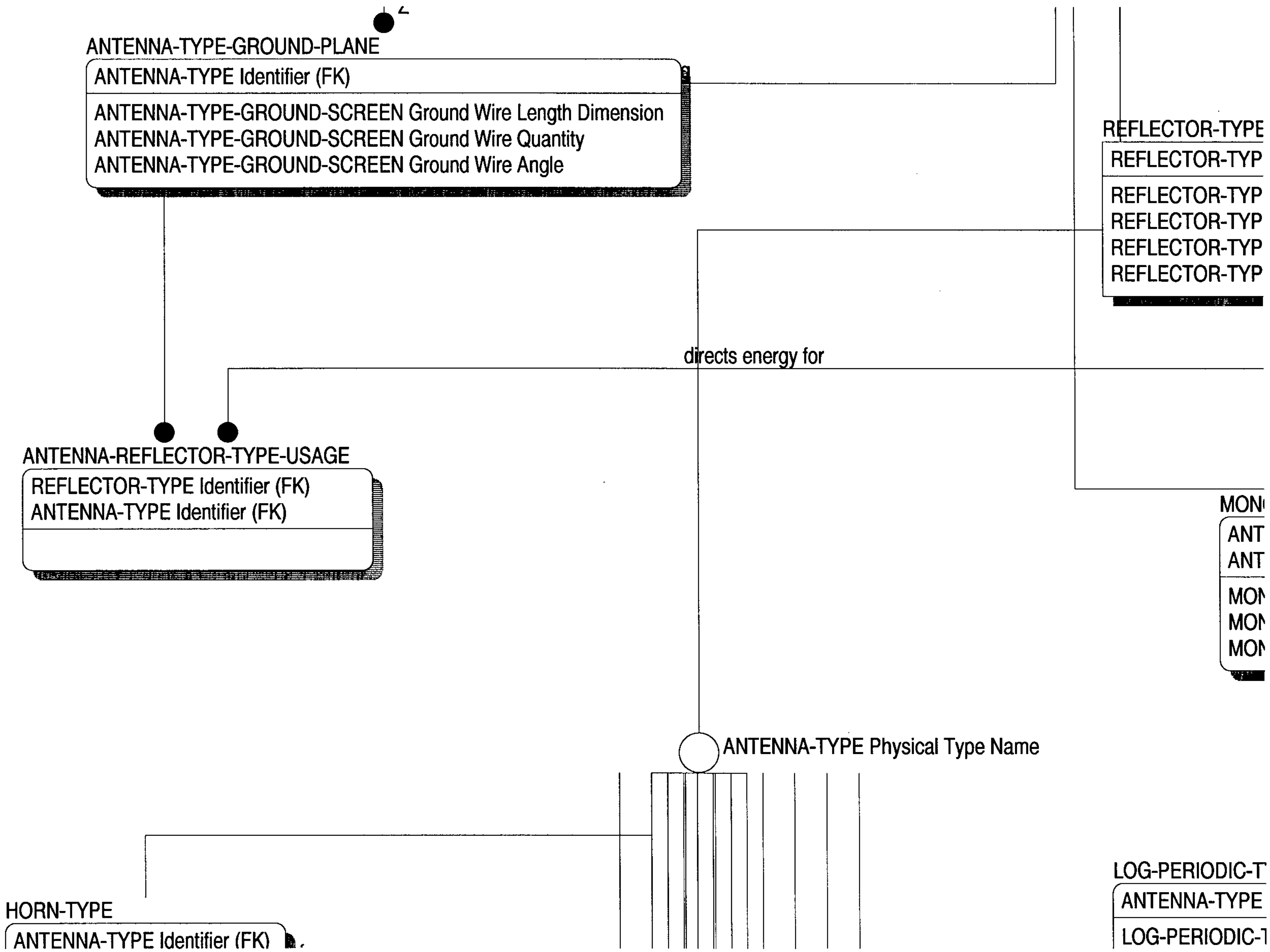
DDDS Page 3

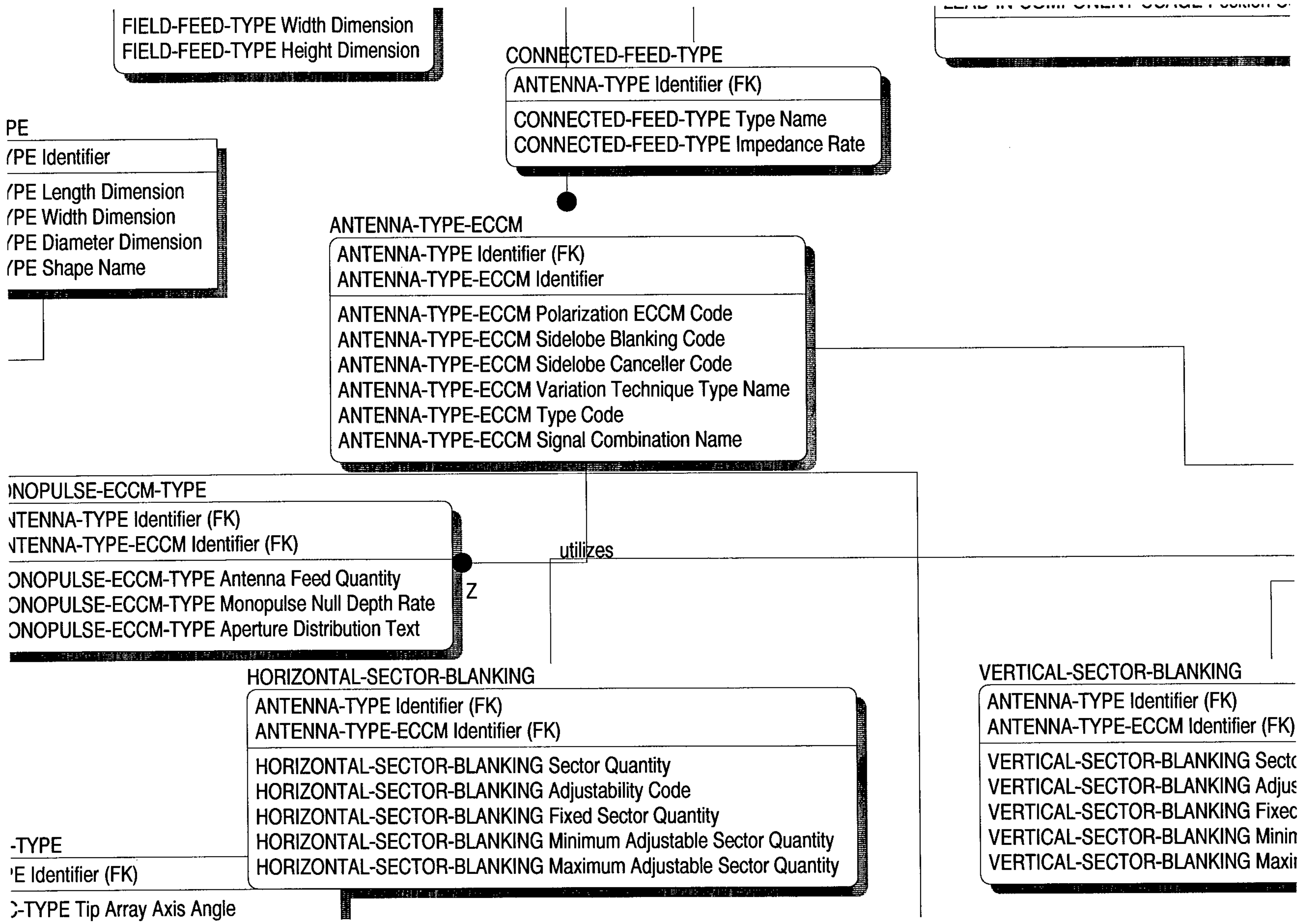
DDDS -Page 2

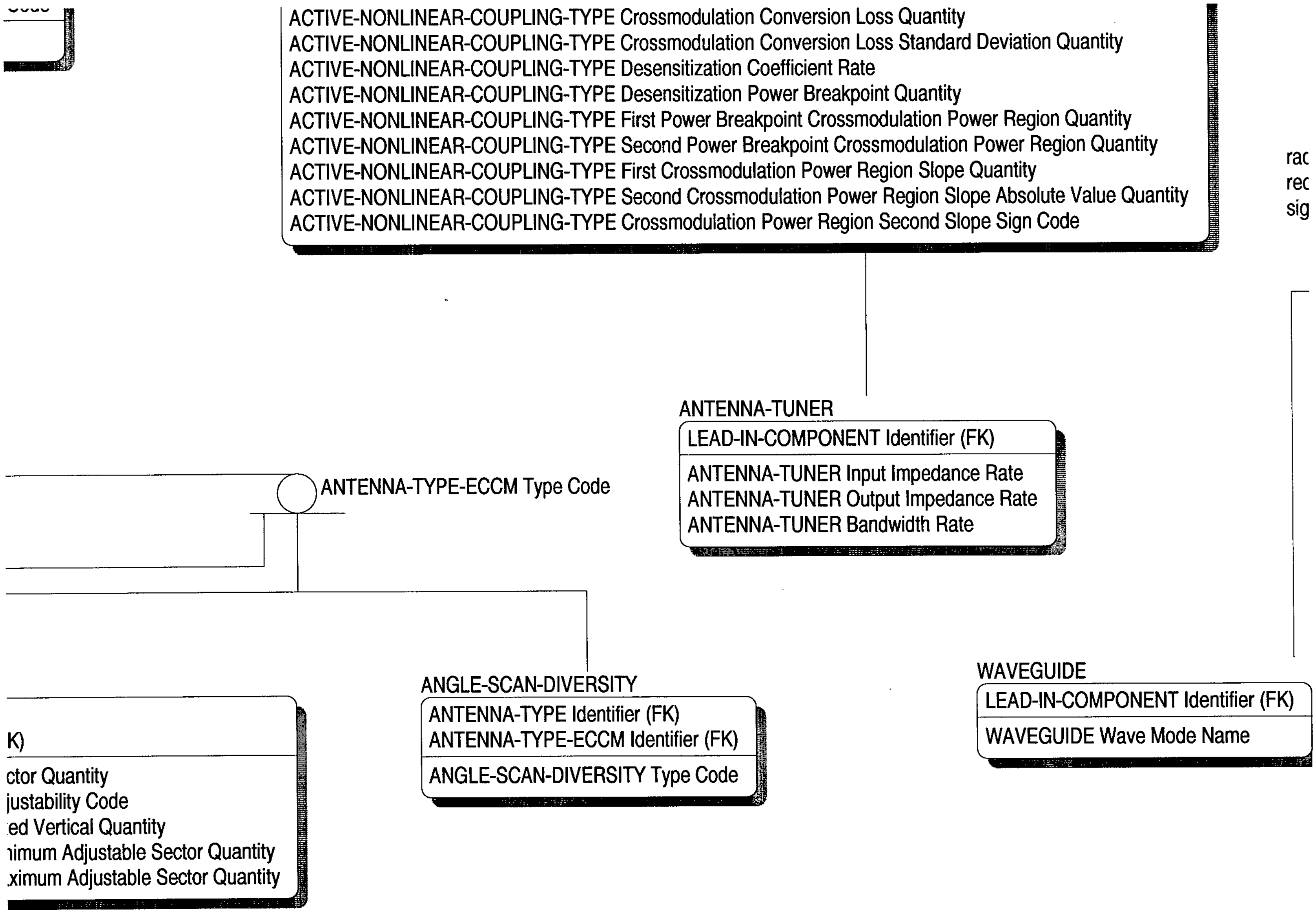


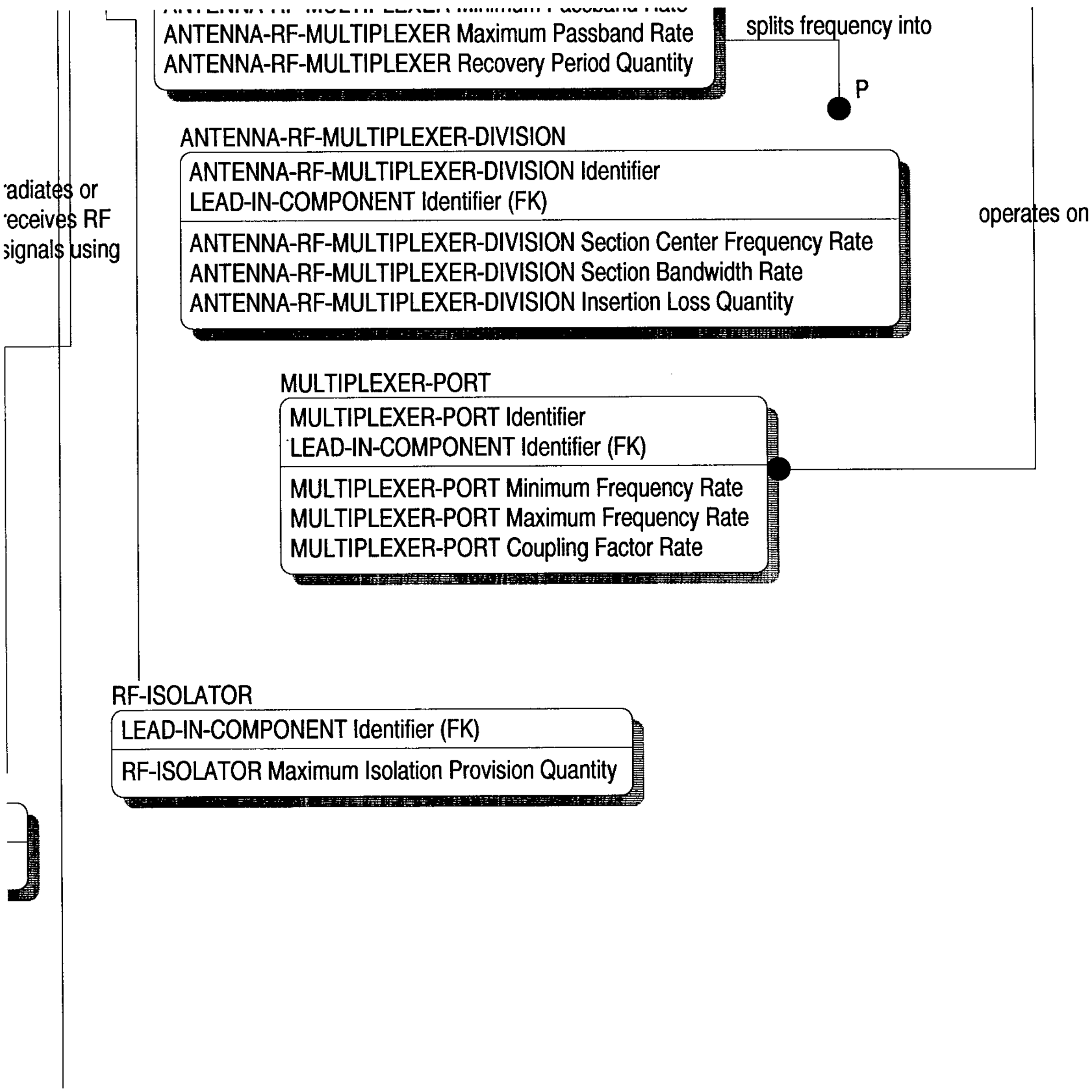


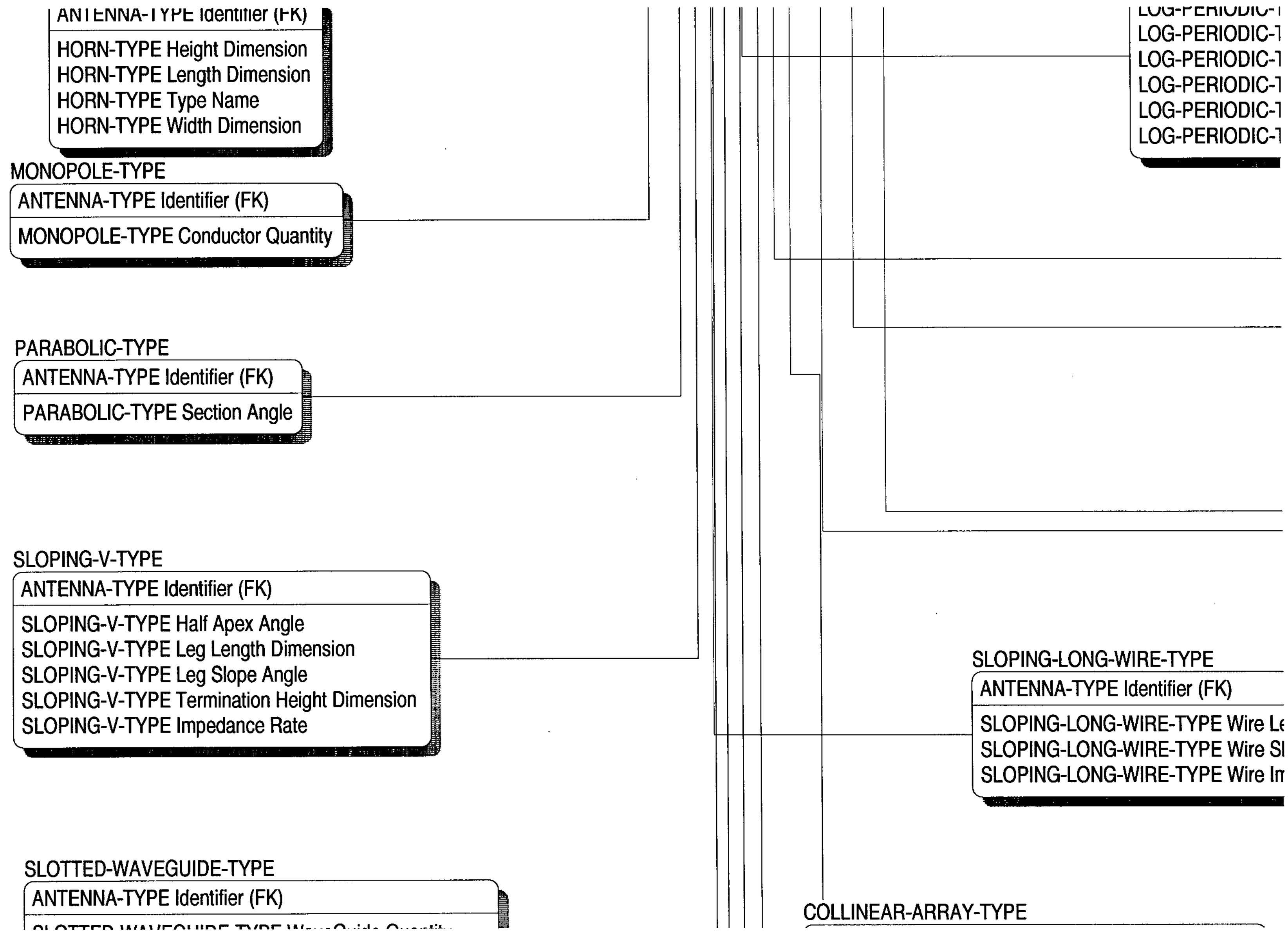
Name
Length Dimension
Maximum Pressurization Rate
Minimum Pressurization Rate
Insertion Loss Rate
er
COUPLER
LEAD-IN-COMPONENT Identifier (FK)
COUPLER Forward Loss Rate
COUPLER Reverse Loss Rate
COUPLER Passband Separation Rate
COUPLER Port Isolation Rate
COUPLER Input Port Quantity
COUPLER Output Port Quantity
COUPLER Passband Quantity
LINEAR-LEAD-IN-COMPONENT Type Name
RF-CONNECTOR
LEAD-IN-COMPONENT Identifier (FK)
DUMMY-LOAD
LEAD-IN-COMPONENT Identifier (FK)
DUMMY-LOAD Effective Radiated Power Rate
ROTARY-JOINT
LEAD-IN-COMPONENT Identifier (FK)
ROTARY-JOINT Wave Mode Name
LEAD-IN-FILTER
LEAD-IN-COMPONENT Identifier (FK)
LEAD-IN-FILTER Bandpass Rate
TRANSMIT-RECEIVE-SWITCH
LEAD-IN-COMPONENT Identifier (FK)
TRANSMIT-RECEIVE-SWITCH Speed Quantity
TRANSMIT-RECEIVE-SWITCH Cutoff Frequency Rate
ANTENNA-RF-MULTIPLEXER
LEAD-IN-COMPONENT Identifier (FK)
ANTENNA-RF-MULTIPLEXER Minimum Passband Rate

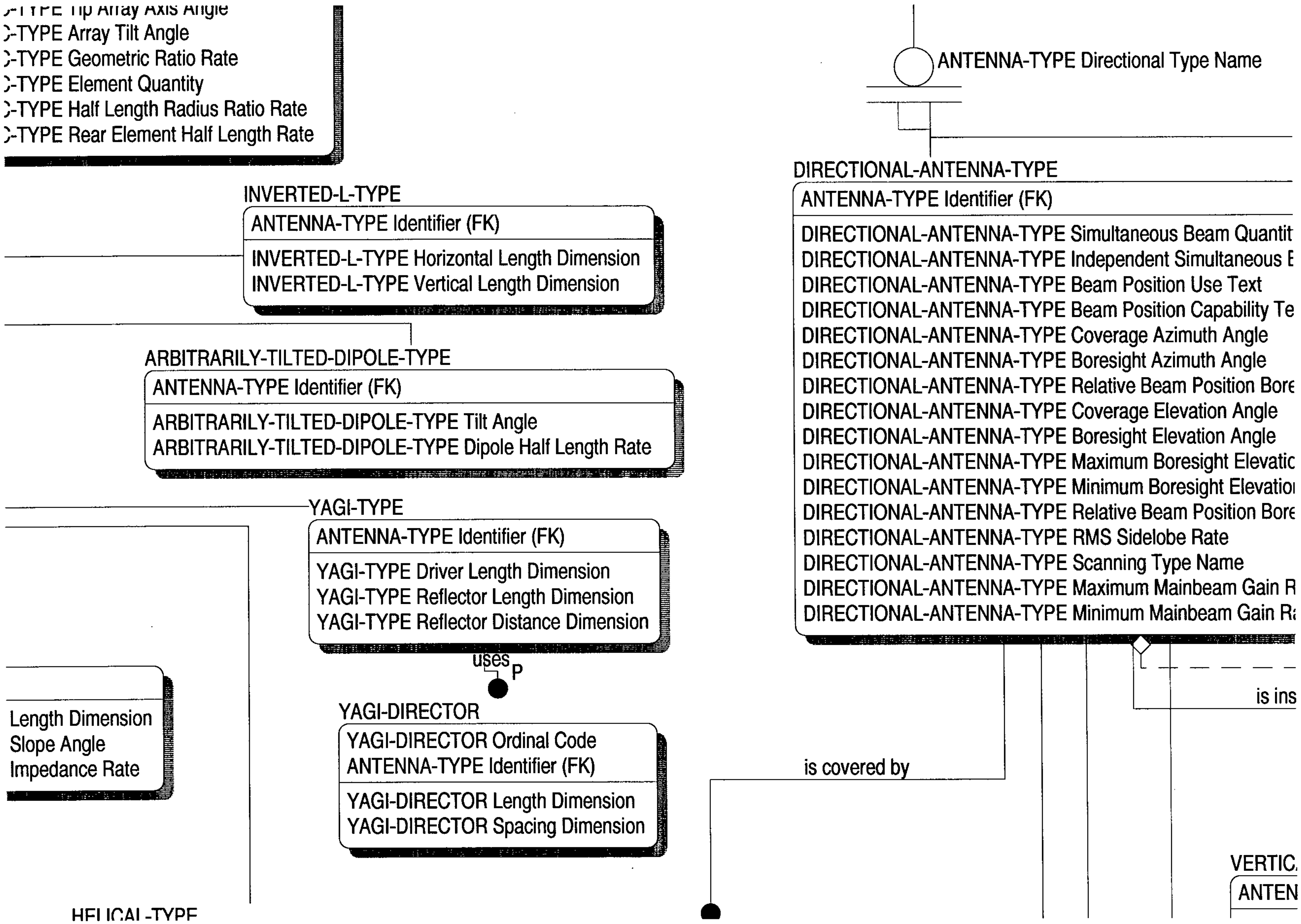

ANTENNA-TYPE-GROUND-PLANE
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-GROUND-SCREEN Ground Wire Length Dimension
ANTENNA-TYPE-GROUND-SCREEN Ground Wire Quantity
ANTENNA-TYPE-GROUND-SCREEN Ground Wire Angle
REFLECTOR-TYPE
REFLECTOR-TYP
REFLECTOR-TYP
REFLECTOR-TYP
REFLECTOR-TYP
REFLECTOR-TYP
directs energy for
ANTENNA-REFLECTOR-TYPE-USAGE
REFLECTOR-TYPE Identifier (FK)
ANTENNA-TYPE Identifier (FK)
MON
ANT
ANT
MON
MON
MON
ANTENNA-TYPE Physical Type Name
HORN-TYPE
ANTENNA-TYPE Identifier (FK)
LOG-PERIODIC-T
ANTENNA-TYPE
LOG-PERIODIC-T


FIELD-FEED-TYPE Width Dimension
FIELD-FEED-TYPE Height Dimension
CONNECTED-FEED-TYPE
ANTENNA-TYPE Identifier (FK)
CONNECTED-FEED-TYPE Type Name
CONNECTED-FEED-TYPE Impedance Rate
PE
/PE Identifier
/PE Length Dimension
/PE Width Dimension
/PE Diameter Dimension
/PE Shape Name
ANTENNA-TYPE-ECCM
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-ECCM Identifier
ANTENNA-TYPE-ECCM Polarization ECCM Code
ANTENNA-TYPE-ECCM Sidelobe Blanking Code
ANTENNA-TYPE-ECCM Sidelobe Canceller Code
ANTENNA-TYPE-ECCM Variation Technique Type Name
ANTENNA-TYPE-ECCM Type Code
ANTENNA-TYPE-ECCM Signal Combination Name
)NOPULSE-ECCM-TYPE
\TENNA-TYPE Identifier (FK)
\TENNA-TYPE-ECCM Identifier (FK)
ONOPULSE-ECCM-TYPE Antenna Feed Quantity
ONOPULSE-ECCM-TYPE Monopulse Null Depth Rate
ONOPULSE-ECCM-TYPE Aperture Distribution Text
utilizes
Z
HORIZONTAL-SECTOR-BLANKING
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-ECCM Identifier (FK)
HORIZONTAL-SECTOR-BLANKING Sector Quantity
HORIZONTAL-SECTOR-BLANKING Adjustability Code
HORIZONTAL-SECTOR-BLANKING Fixed Sector Quantity
HORIZONTAL-SECTOR-BLANKING Minimum Adjustable Sector Quantity
HORIZONTAL-SECTOR-BLANKING Maximum Adjustable Sector Quantity
VERTICAL-SECTOR-BLANKING
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-ECCM Identifier (FK)
VERTICAL-SECTOR-BLANKING Secto
VERTICAL-SECTOR-BLANKING Adjus
VERTICAL-SECTOR-BLANKING Fixec
VERTICAL-SECTOR-BLANKING Minin
VERTICAL-SECTOR-BLANKING Maxi
-TYPE
'E Identifier (FK)
>-TYPE Tip Array Axis Angle

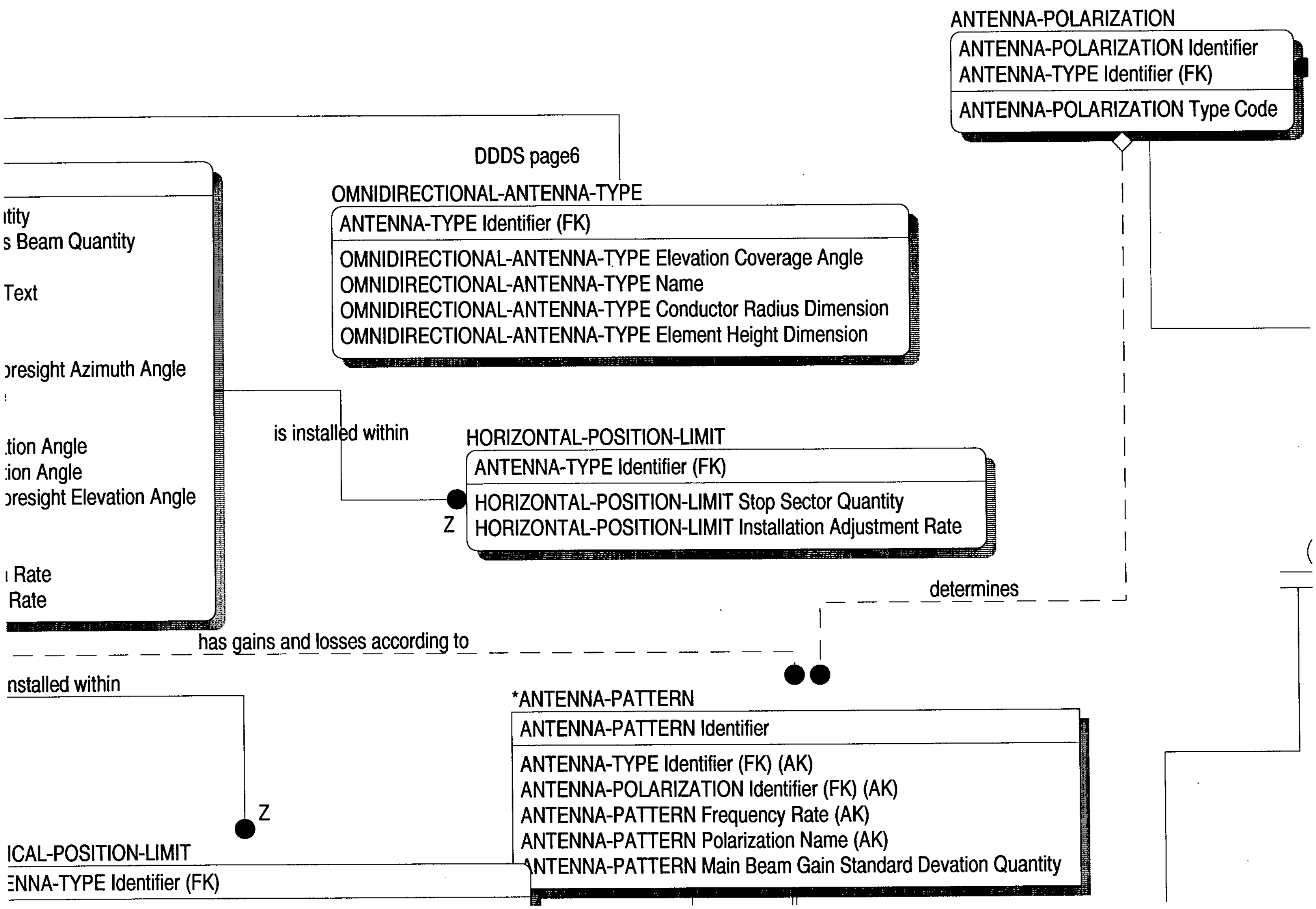

ACTIVE-NONLINEAR-COUPLING-TYPE Crossmodulation Conversion Loss Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE Crossmodulation Conversion Loss Standard Deviation Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE Desensitization Coefficient Rate
ACTIVE-NONLINEAR-COUPLING-TYPE Desensitization Power Breakpoint Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE First Power Breakpoint Crossmodulation Power Region Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE Second Power Breakpoint Crossmodulation Power Region Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE First Crossmodulation Power Region Slope Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE Second Crossmodulation Power Region Slope Absolute Value Quantity
ACTIVE-NONLINEAR-COUPLING-TYPE Crossmodulation Power Region Second Slope Sign Code
rac
rec
sig
ANTENNA-TUNER
LEAD-IN-COMPONENT Identifier (FK)
ANTENNA-TUNER Input Impedance Rate
ANTENNA-TUNER Output Impedance Rate
ANTENNA-TUNER Bandwidth Rate
ANTENNA-TYPE-ECCM Type Code
ANGLE-SCAN-DIVERSITY
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-ECCM Identifier (FK)
ANGLE-SCAN-DIVERSITY Type Code
WAVEGUIDE
LEAD-IN-COMPONENT Identifier (FK)
WAVEGUIDE Wave Mode Name
K)
ctor Quantity
justability Code
ed Vertical Quantity
imum Adjustable Sector Quantity
ximum Adjustable Sector Quantity


ANTENNA-RF-MULTIPLEXER Maximum Passband Rate
ANTENNA-RF-MULTIPLEXER Recovery Period Quantity
splits frequency into
P
ANTENNA-RF-MULTIPLEXER-DIVISION
ANTENNA-RF-MULTIPLEXER-DIVISION Identifier
LEAD-IN-COMPONENT Identifier (FK)
ANTENNA-RF-MULTIPLEXER-DIVISION Section Center Frequency Rate
ANTENNA-RF-MULTIPLEXER-DIVISION Section Bandwidth Rate
ANTENNA-RF-MULTIPLEXER-DIVISION Insertion Loss Quantity
operates on
MULTIPLEXER-PORT
MULTIPLEXER-PORT Identifier
LEAD-IN-COMPONENT Identifier (FK)
MULTIPLEXER-PORT Minimum Frequency Rate
MULTIPLEXER-PORT Maximum Frequency Rate
MULTIPLEXER-PORT Coupling Factor Rate
RF-ISOLATOR
LEAD-IN-COMPONENT Identifier (FK)
RF-ISOLATOR Maximum Isolation Provision Quantity

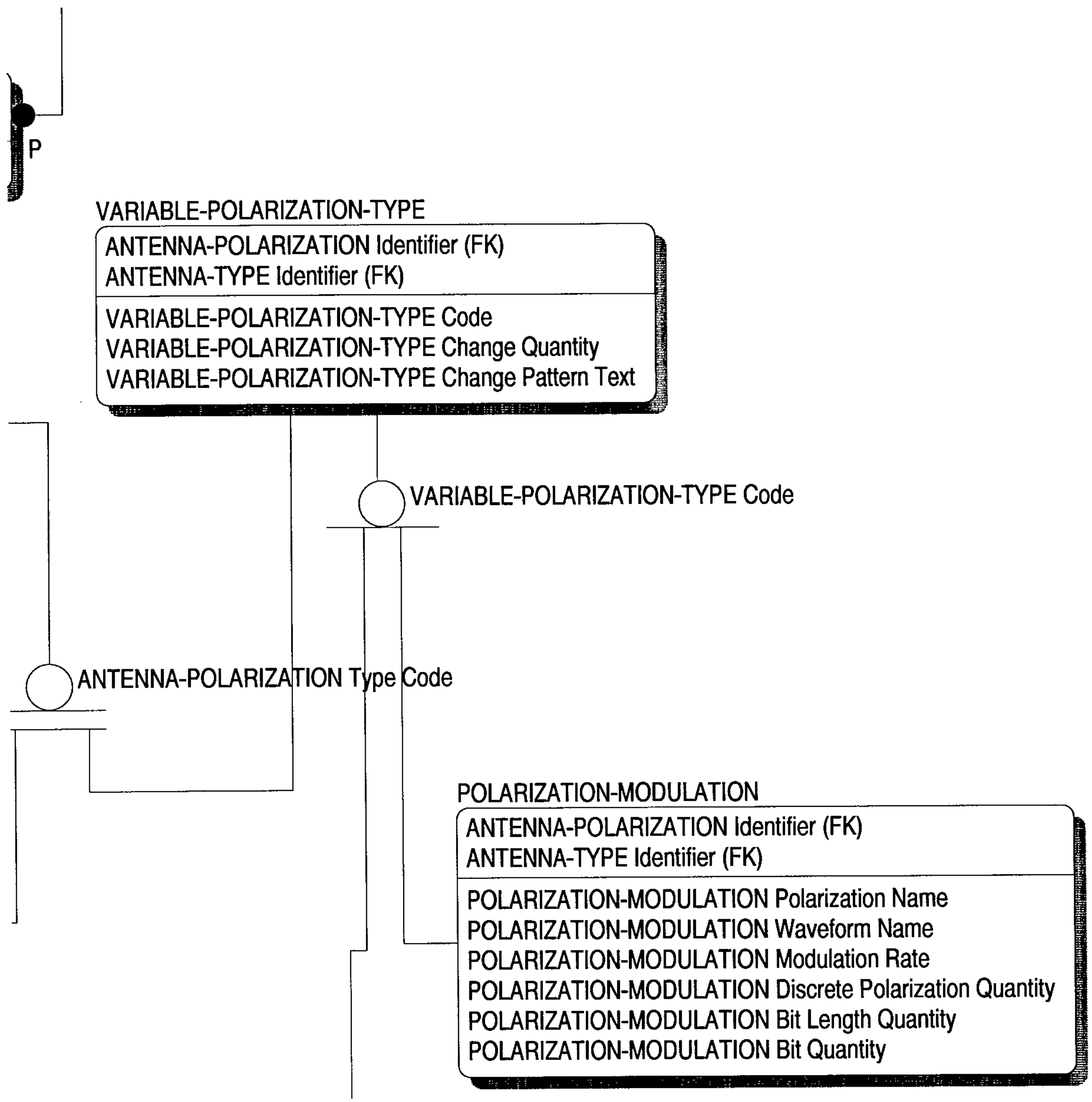

ANTENNA-TYPE Identifier (FK)
HORN-TYPE Height Dimension
HORN-TYPE Length Dimension
HORN-TYPE Type Name
HORN-TYPE Width Dimension
MONOPOLE-TYPE
ANTENNA-TYPE Identifier (FK)
MONOPOLE-TYPE Conductor Quantity
PARABOLIC-TYPE
ANTENNA-TYPE Identifier (FK)
PARABOLIC-TYPE Section Angle
SLOPING-V-TYPE
ANTENNA-TYPE Identifier (FK)
SLOPING-V-TYPE Half Apex Angle
SLOPING-V-TYPE Leg Length Dimension
SLOPING-V-TYPE Leg Slope Angle
SLOPING-V-TYPE Termination Height Dimension
SLOPING-V-TYPE Impedance Rate
SLOTTED-WAVEGUIDE-TYPE
ANTENNA-TYPE Identifier (FK)
SLOPING-LONG-WIRE-TYPE
ANTENNA-TYPE Identifier (FK)
COLLINEAR-ARRAY-TYPE

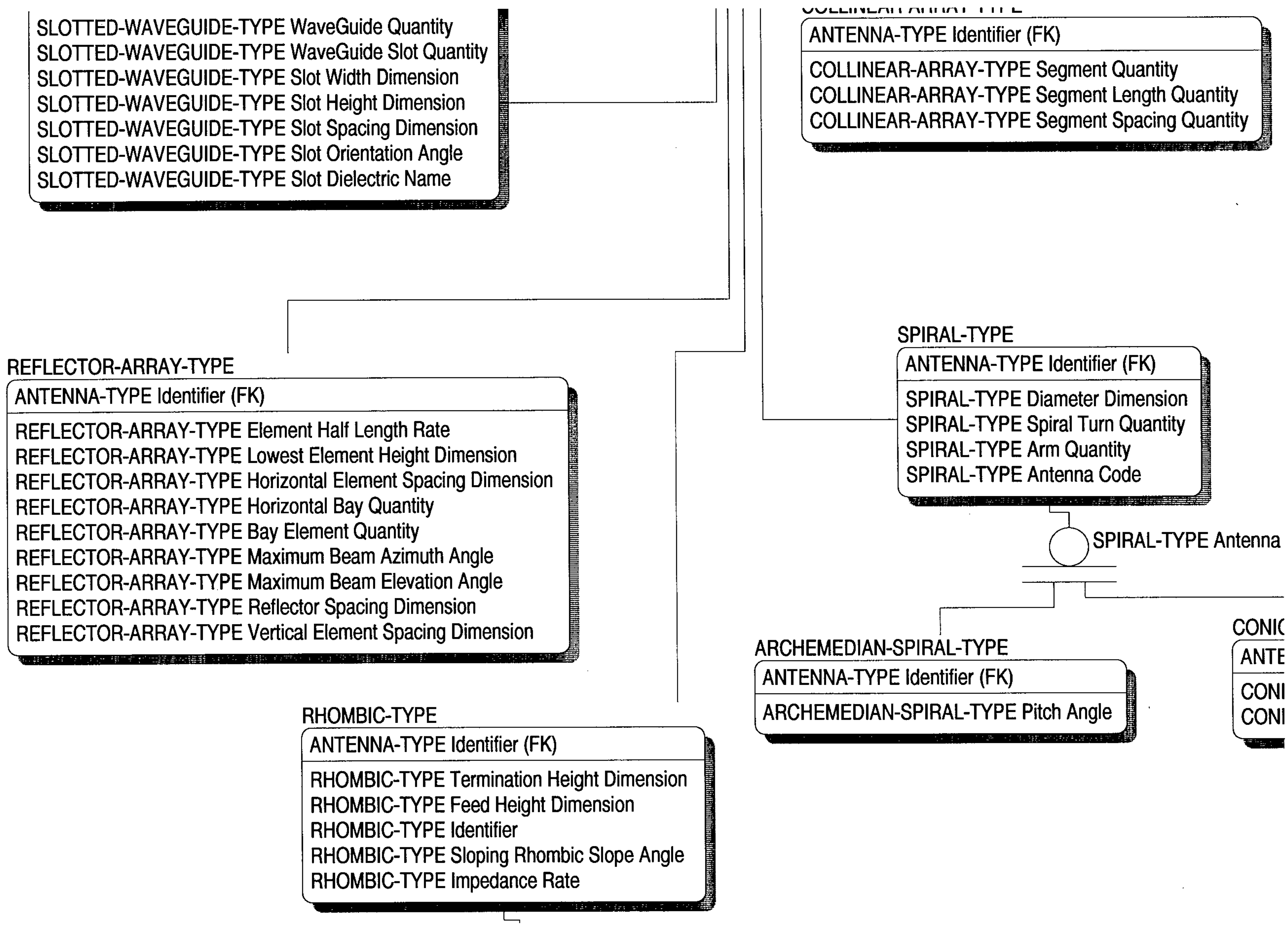


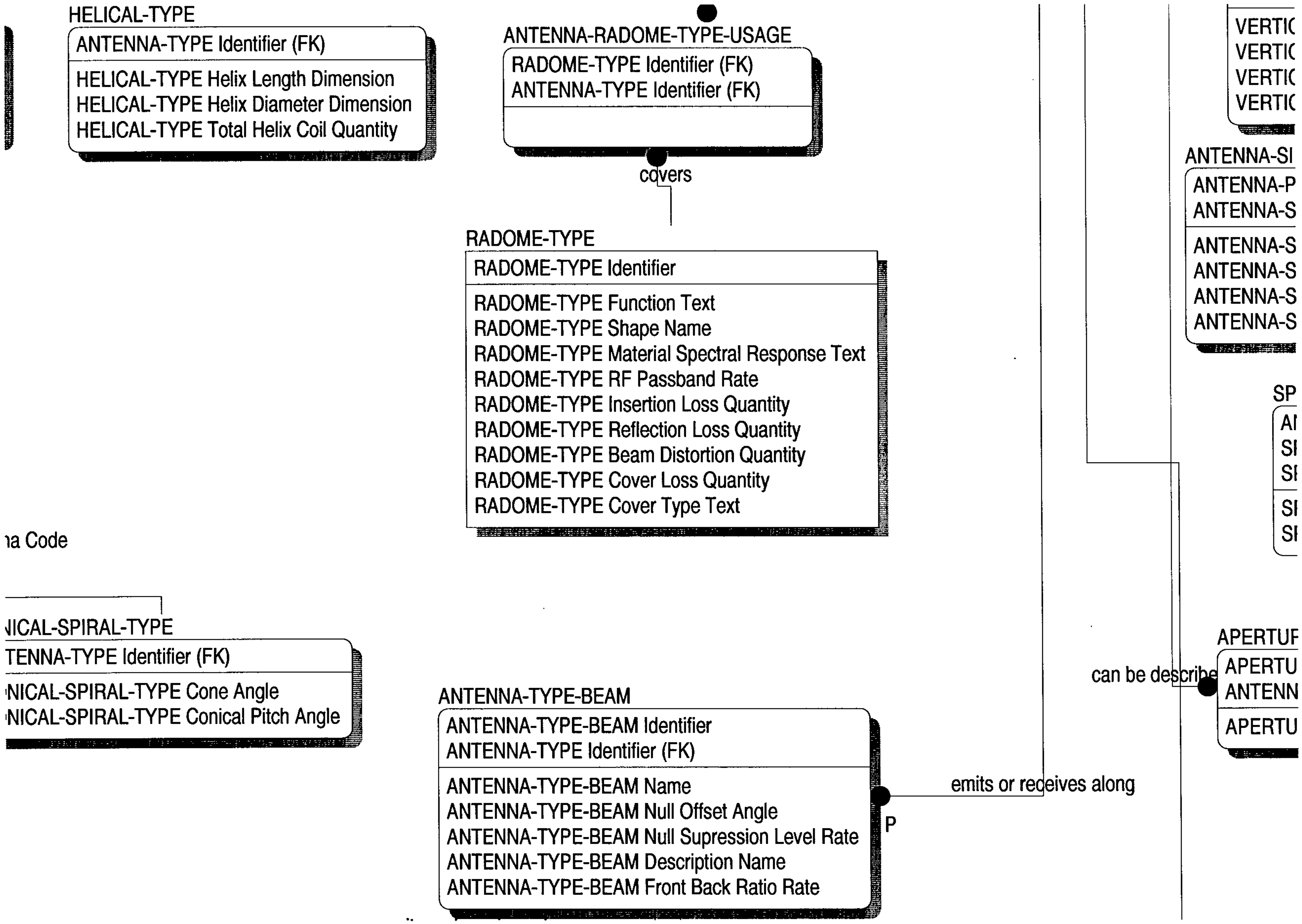
-TYPE Tip Array Axis Angle
-TYPE Array Tilt Angle
-TYPE Geometric Ratio Rate
-TYPE Element Quantity
-TYPE Half Length Radius Ratio Rate
-TYPE Rear Element Half Length Rate
INVERTED-L-TYPE
ANTENNA-TYPE Identifier (FK)
INVERTED-L-TYPE Horizontal Length Dimension
INVERTED-L-TYPE Vertical Length Dimension
ARBITRARILY-TILTED-DIPOLE-TYPE
ANTENNA-TYPE Identifier (FK)
ARBITRARILY-TILTED-DIPOLE-TYPE Tilt Angle
ARBITRARILY-TILTED-DIPOLE-TYPE Dipole Half Length Rate
YAGI-TYPE
ANTENNA-TYPE Identifier (FK)
YAGI-TYPE Driver Length Dimension
YAGI-TYPE Reflector Length Dimension
YAGI-TYPE Reflector Distance Dimension
uses
P
YAGI-DIRECTOR
YAGI-DIRECTOR Ordinal Code
ANTENNA-TYPE Identifier (FK)
YAGI-DIRECTOR Length Dimension
YAGI-DIRECTOR Spacing Dimension
Length Dimension
Slope Angle
Impedance Rate
HELICAL-TYPE
ANTENNA-TYPE Directional Type Name
DIRECTIONAL-ANTENNA-TYPE
ANTENNA-TYPE Identifier (FK)
DIRECTIONAL-ANTENNA-TYPE Simultaneous Beam Quantit
DIRECTIONAL-ANTENNA-TYPE Independent Simultaneous E
DIRECTIONAL-ANTENNA-TYPE Beam Position Use Text
DIRECTIONAL-ANTENNA-TYPE Beam Position Capability Te
DIRECTIONAL-ANTENNA-TYPE Coverage Azimuth Angle
DIRECTIONAL-ANTENNA-TYPE Boresight Azimuth Angle
DIRECTIONAL-ANTENNA-TYPE Relative Beam Position Bore
DIRECTIONAL-ANTENNA-TYPE Coverage Elevation Angle
DIRECTIONAL-ANTENNA-TYPE Boresight Elevation Angle
DIRECTIONAL-ANTENNA-TYPE Maximum Boresight Elevatic
DIRECTIONAL-ANTENNA-TYPE Minimum Boresight Elevatio
DIRECTIONAL-ANTENNA-TYPE Relative Beam Position Bore
DIRECTIONAL-ANTENNA-TYPE RMS Sidelobe Rate
DIRECTIONAL-ANTENNA-TYPE Scanning Type Name
DIRECTIONAL-ANTENNA-TYPE Maximum Mainbeam Gain R
DIRECTIONAL-ANTENNA-TYPE Minimum Mainbeam Gain R
is ins
is covered by
VERTIC
ANTEN

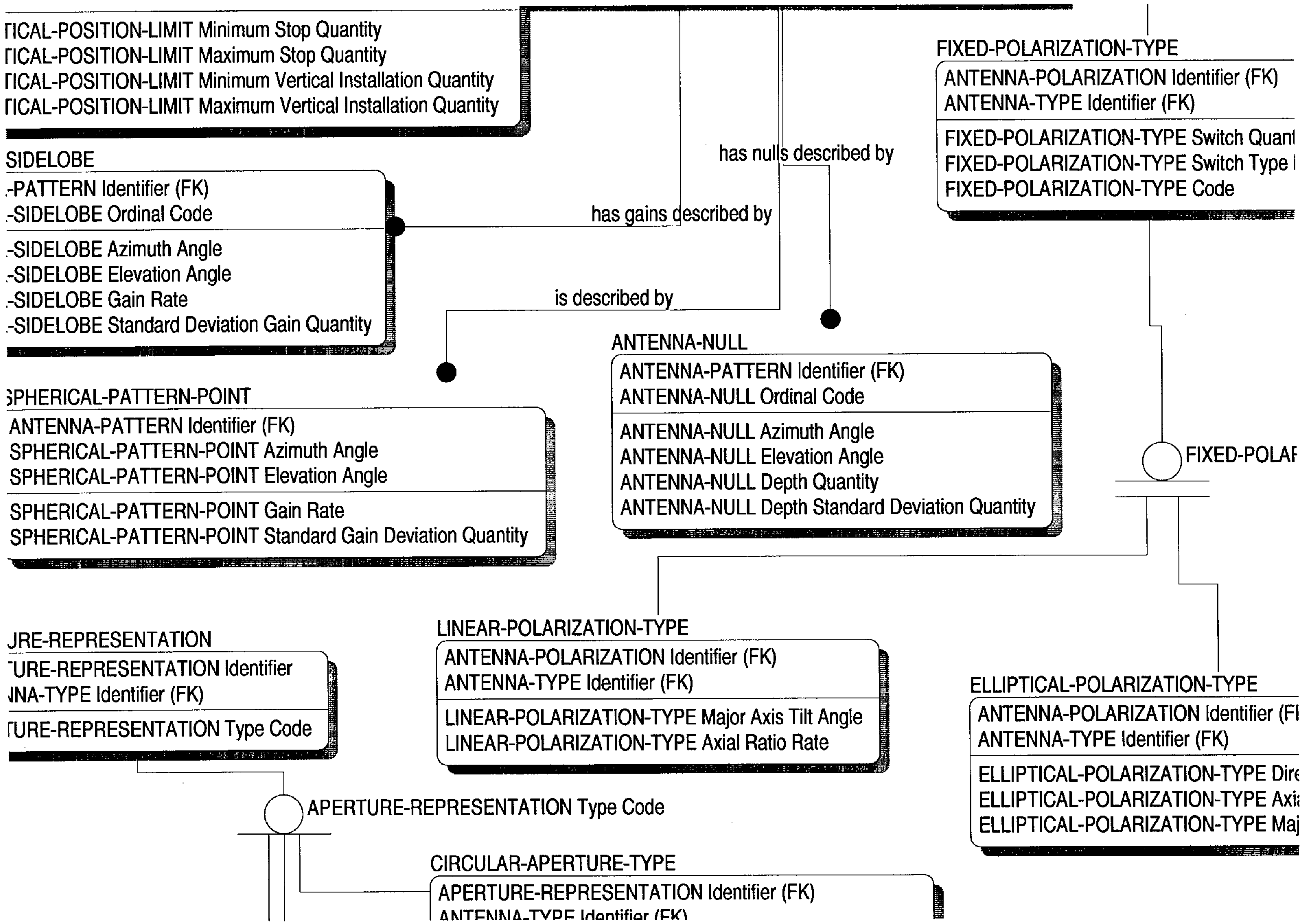


ANTENNA-POLARIZATION
ANTENNA-POLARIZATION Identifier
ANTENNA-TYPE Identifier (FK)
ANTENNA-POLARIZATION Type Code
DDDS page6
OMNIDIRECTIONAL-ANTENNA-TYPE
ANTENNA-TYPE Identifier (FK)
OMNIDIRECTIONAL-ANTENNA-TYPE Elevation Coverage Angle
OMNIDIRECTIONAL-ANTENNA-TYPE Name
OMNIDIRECTIONAL-ANTENNA-TYPE Conductor Radius Dimension
OMNIDIRECTIONAL-ANTENNA-TYPE Element Height Dimension
ıtity
s Beam Quantity
Text
ɔresight Azimuth Angle
.tion Angle
:ion Angle
ɔresight Elevation Angle
ı Rate
Rate
is installed within
HORIZONTAL-POSITION-LIMIT
ANTENNA-TYPE Identifier (FK)
HORIZONTAL-POSITION-LIMIT Stop Sector Quantity
HORIZONTAL-POSITION-LIMIT Installation Adjustment Rate
Z
determines
has gains and losses according to
nstalled within
Z
*ANTENNA-PATTERN
ANTENNA-PATTERN Identifier
ANTENNA-TYPE Identifier (FK) (AK)
ANTENNA-POLARIZATION Identifier (FK) (AK)
ANTENNA-PATTERN Frequency Rate (AK)
ANTENNA-PATTERN Polarization Name (AK)
ANTENNA-PATTERN Main Beam Gain Standard Devation Quantity
ICAL-POSITION-LIMIT
ΞNNA-TYPE Identifier (FK)


P
VARIABLE-POLARIZATION-TYPE
ANTENNA-POLARIZATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
VARIABLE-POLARIZATION-TYPE Code
VARIABLE-POLARIZATION-TYPE Change Quantity
VARIABLE-POLARIZATION-TYPE Change Pattern Text
VARIABLE-POLARIZATION-TYPE Code
ANTENNA-POLARIZATION Type Code
POLARIZATION-MODULATION
ANTENNA-POLARIZATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
POLARIZATION-MODULATION Polarization Name
POLARIZATION-MODULATION Waveform Name
POLARIZATION-MODULATION Modulation Rate
POLARIZATION-MODULATION Discrete Polarization Quantity
POLARIZATION-MODULATION Bit Length Quantity
POLARIZATION-MODULATION Bit Quantity


SLOTTED-WAVEGUIDE-TYPE WaveGuide Quantity
SLOTTED-WAVEGUIDE-TYPE WaveGuide Slot Quantity
SLOTTED-WAVEGUIDE-TYPE Slot Width Dimension
SLOTTED-WAVEGUIDE-TYPE Slot Height Dimension
SLOTTED-WAVEGUIDE-TYPE Slot Spacing Dimension
SLOTTED-WAVEGUIDE-TYPE Slot Orientation Angle
SLOTTED-WAVEGUIDE-TYPE Slot Dielectric Name
ANTENNA-TYPE Identifier (FK)
COLLINEAR-ARRAY-TYPE Segment Quantity
COLLINEAR-ARRAY-TYPE Segment Length Quantity
COLLINEAR-ARRAY-TYPE Segment Spacing Quantity
REFLECTOR-ARRAY-TYPE
ANTENNA-TYPE Identifier (FK)
REFLECTOR-ARRAY-TYPE Element Half Length Rate
REFLECTOR-ARRAY-TYPE Lowest Element Height Dimension
REFLECTOR-ARRAY-TYPE Horizontal Element Spacing Dimension
REFLECTOR-ARRAY-TYPE Horizontal Bay Quantity
REFLECTOR-ARRAY-TYPE Bay Element Quantity
REFLECTOR-ARRAY-TYPE Maximum Beam Azimuth Angle
REFLECTOR-ARRAY-TYPE Maximum Beam Elevation Angle
REFLECTOR-ARRAY-TYPE Reflector Spacing Dimension
REFLECTOR-ARRAY-TYPE Vertical Element Spacing Dimension
SPIRAL-TYPE
ANTENNA-TYPE Identifier (FK)
SPIRAL-TYPE Diameter Dimension
SPIRAL-TYPE Spiral Turn Quantity
SPIRAL-TYPE Arm Quantity
SPIRAL-TYPE Antenna Code
SPIRAL-TYPE Antenna
ARCHEMEDIAN-SPIRAL-TYPE
ANTENNA-TYPE Identifier (FK)
ARCHEMEDIAN-SPIRAL-TYPE Pitch Angle
RHOMBIC-TYPE
ANTENNA-TYPE Identifier (FK)
RHOMBIC-TYPE Termination Height Dimension
RHOMBIC-TYPE Feed Height Dimension
RHOMBIC-TYPE Identifier
RHOMBIC-TYPE Sloping Rhombic Slope Angle
RHOMBIC-TYPE Impedance Rate


HELICAL-TYPE
ANTENNA-TYPE Identifier (FK)
HELICAL-TYPE Helix Length Dimension
HELICAL-TYPE Helix Diameter Dimension
HELICAL-TYPE Total Helix Coil Quantity
ANTENNA-RADOME-TYPE-USAGE
RADOME-TYPE Identifier (FK)
ANTENNA-TYPE Identifier (FK)
covers
RADOME-TYPE
RADOME-TYPE Identifier
RADOME-TYPE Function Text
RADOME-TYPE Shape Name
RADOME-TYPE Material Spectral Response Text
RADOME-TYPE RF Passband Rate
RADOME-TYPE Insertion Loss Quantity
RADOME-TYPE Reflection Loss Quantity
RADOME-TYPE Beam Distortion Quantity
RADOME-TYPE Cover Loss Quantity
RADOME-TYPE Cover Type Text
na Code
NICAL-SPIRAL-TYPE
TENNA-TYPE Identifier (FK)
NICAL-SPIRAL-TYPE Cone Angle
NICAL-SPIRAL-TYPE Conical Pitch Angle
ANTENNA-TYPE-BEAM
ANTENNA-TYPE-BEAM Identifier
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-BEAM Name
ANTENNA-TYPE-BEAM Null Offset Angle
ANTENNA-TYPE-BEAM Null Supression Level Rate
ANTENNA-TYPE-BEAM Description Name
ANTENNA-TYPE-BEAM Front Back Ratio Rate
P
emits or receives along
can be describe
VERTIC
VERTIC
VERTIC
VERTIC
ANTENNA-SI
ANTENNA-P
ANTENNA-S
ANTENNA-S
ANTENNA-S
ANTENNA-S
ANTENNA-S
SP
AI
SI
SI
SI
SI
APERTUF
APERTU
ANTENN
APERTU



TICAL-POSITION-LIMIT Minimum Stop Quantity
TICAL-POSITION-LIMIT Maximum Stop Quantity
TICAL-POSITION-LIMIT Minimum Vertical Installation Quantity
TICAL-POSITION-LIMIT Maximum Vertical Installation Quantity
SIDELOBE
-PATTERN Identifier (FK)
-SIDELOBE Ordinal Code
-SIDELOBE Azimuth Angle
-SIDELOBE Elevation Angle
-SIDELOBE Gain Rate
-SIDELOBE Standard Deviation Gain Quantity
has gains described by
has nulls described by
is described by
SPHERICAL-PATTERN-POINT
ANTENNA-PATTERN Identifier (FK)
SPHERICAL-PATTERN-POINT Azimuth Angle
SPHERICAL-PATTERN-POINT Elevation Angle
SPHERICAL-PATTERN-POINT Gain Rate
SPHERICAL-PATTERN-POINT Standard Gain Deviation Quantity
ANTENNA-NULL
ANTENNA-PATTERN Identifier (FK)
ANTENNA-NULL Ordinal Code
ANTENNA-NULL Azimuth Angle
ANTENNA-NULL Elevation Angle
ANTENNA-NULL Depth Quantity
ANTENNA-NULL Depth Standard Deviation Quantity
FIXED-POLARIZATION-TYPE
ANTENNA-POLARIZATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
FIXED-POLARIZATION-TYPE Switch Quant
FIXED-POLARIZATION-TYPE Switch Type
FIXED-POLARIZATION-TYPE Code
FIXED-POLAR
URE-REPRESENTATION
URE-REPRESENTATION Identifier
NA-TYPE Identifier (FK)
URE-REPRESENTATION Type Code
LINEAR-POLARIZATION-TYPE
ANTENNA-POLARIZATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
LINEAR-POLARIZATION-TYPE Major Axis Tilt Angle
LINEAR-POLARIZATION-TYPE Axial Ratio Rate
ELLIPTICAL-POLARIZATION-TYPE
ANTENNA-POLARIZATION Identifier (F
ANTENNA-TYPE Identifier (FK)
ELLIPTICAL-POLARIZATION-TYPE Dire
ELLIPTICAL-POLARIZATION-TYPE Axi
ELLIPTICAL-POLARIZATION-TYPE Maj
APERTURE-REPRESENTATION Type Code
CIRCULAR-APERTURE-TYPE
APERTURE-REPRESENTATION Identifier (FK)

antity
e Name

ADAPTIVE-POLARIZATION

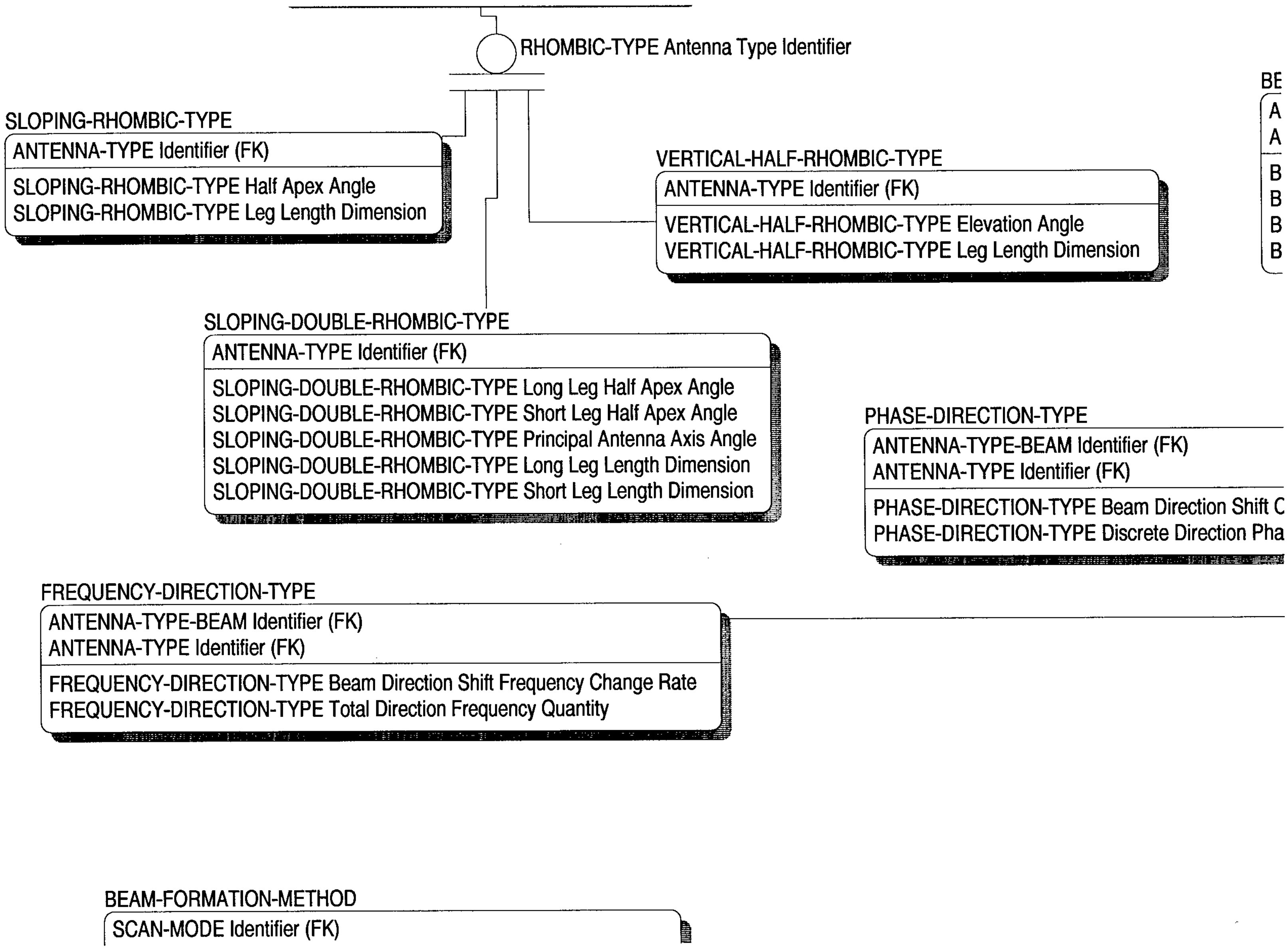
ANTENNA-POLARIZATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)

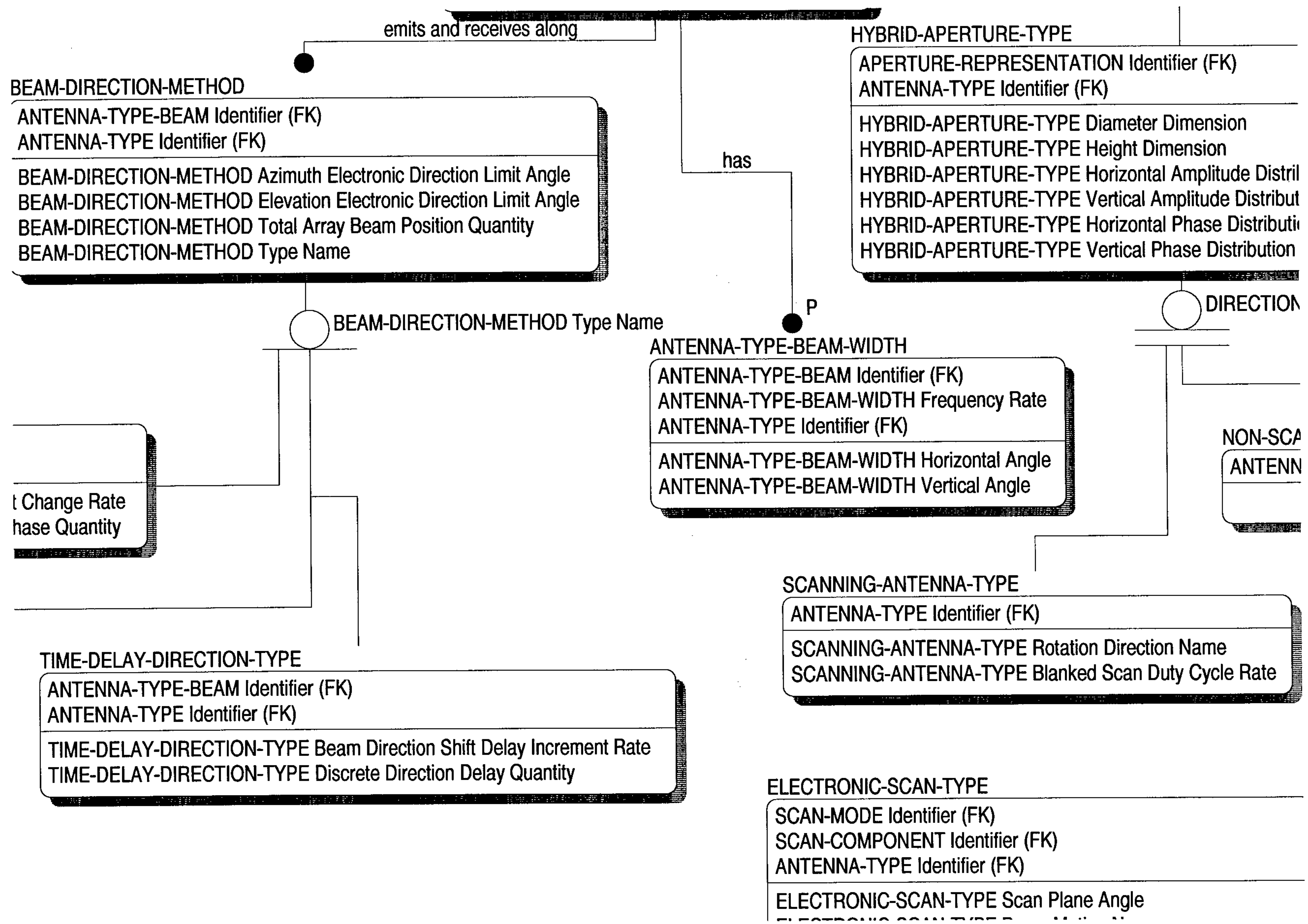
ADAPTIVE-POLARIZATION-TYPE Change Reason Name
ADAPTIVE-POLARIZATION-TYPE Code

ARIZATION-TYPE Code

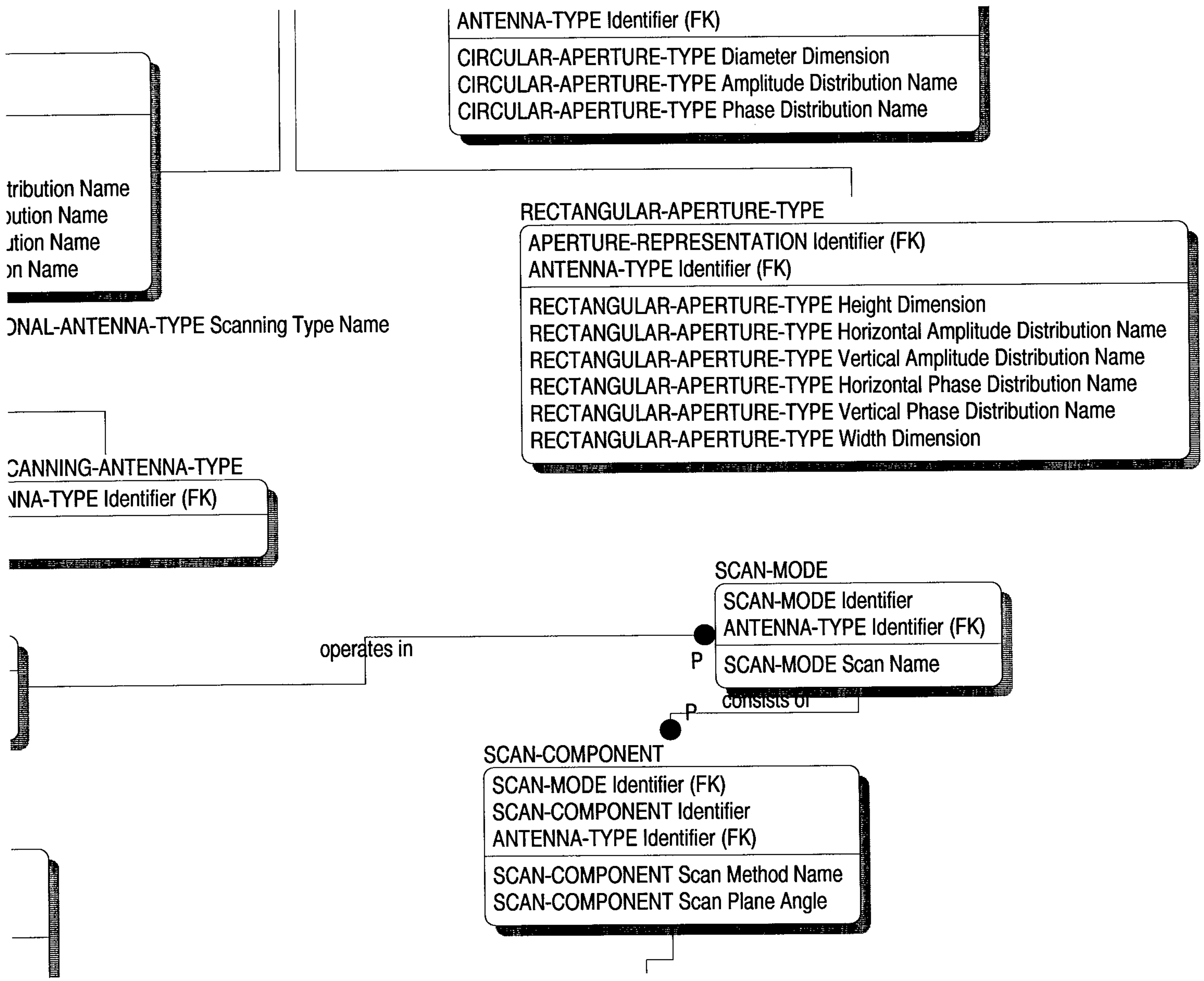
(FK)

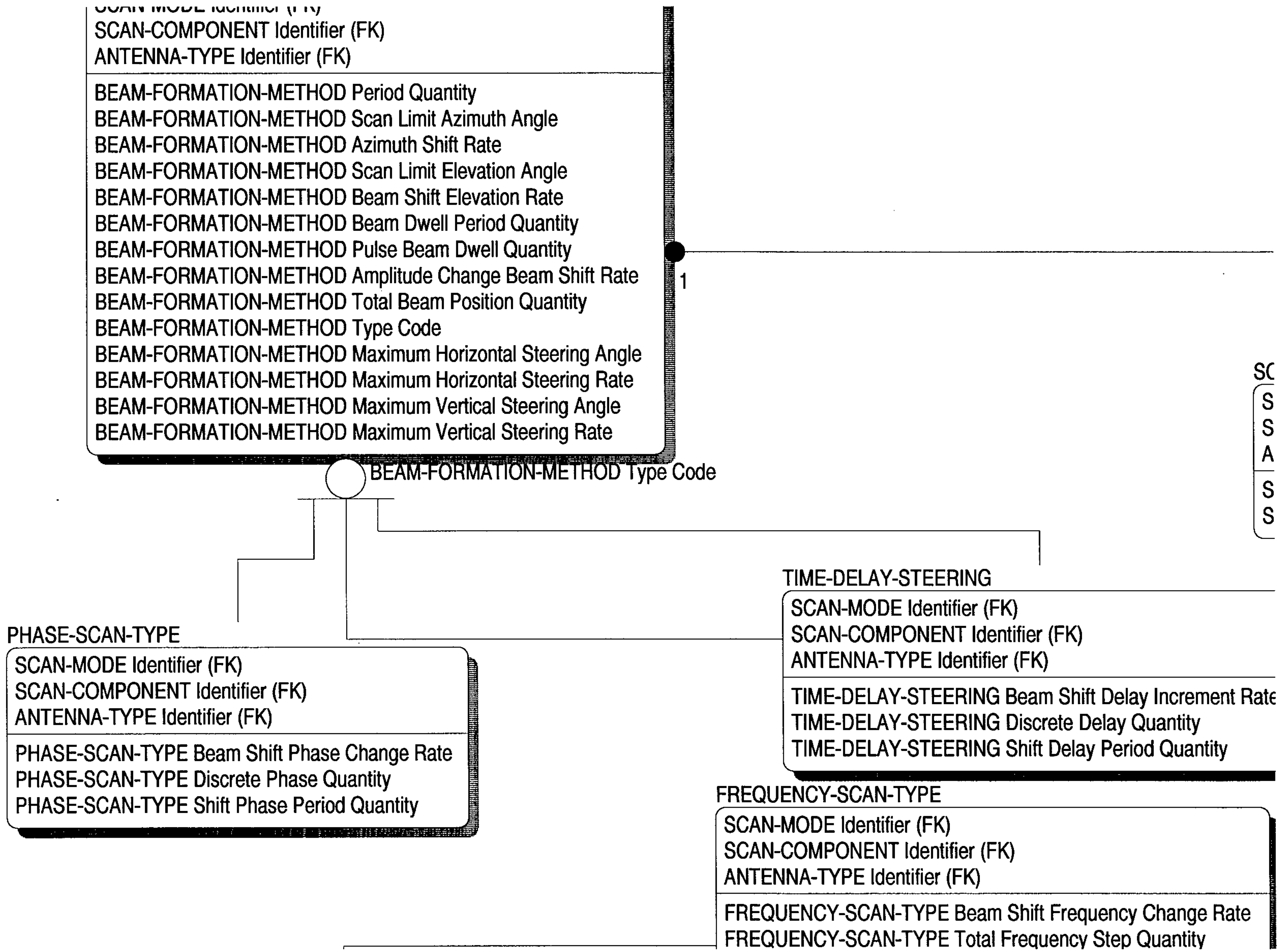
irection Sensor Code
xial Ratio Rate
lajor Axis Tilt Angle

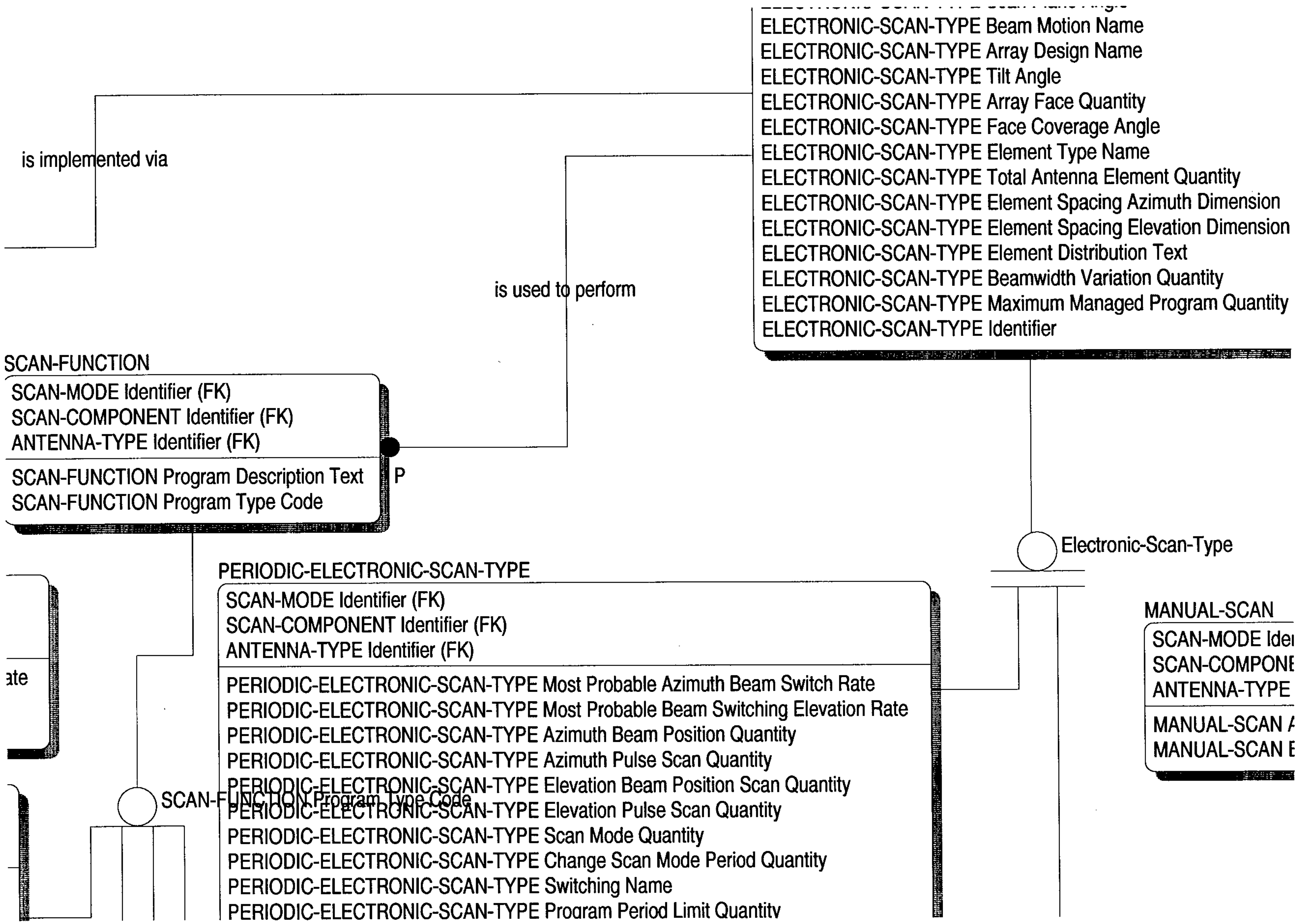

RHOMBIC-TYPE Antenna Type Identifier
SLOPING-RHOMBIC-TYPE
ANTENNA-TYPE Identifier (FK)
SLOPING-RHOMBIC-TYPE Half Apex Angle
SLOPING-RHOMBIC-TYPE Leg Length Dimension
VERTICAL-HALF-RHOMBIC-TYPE
ANTENNA-TYPE Identifier (FK)
VERTICAL-HALF-RHOMBIC-TYPE Elevation Angle
VERTICAL-HALF-RHOMBIC-TYPE Leg Length Dimension
BE
A
A
B
B
B
B
SLOPING-DOUBLE-RHOMBIC-TYPE
ANTENNA-TYPE Identifier (FK)
SLOPING-DOUBLE-RHOMBIC-TYPE Long Leg Half Apex Angle
SLOPING-DOUBLE-RHOMBIC-TYPE Short Leg Half Apex Angle
SLOPING-DOUBLE-RHOMBIC-TYPE Principal Antenna Axis Angle
SLOPING-DOUBLE-RHOMBIC-TYPE Long Leg Length Dimension
SLOPING-DOUBLE-RHOMBIC-TYPE Short Leg Length Dimension
PHASE-DIRECTION-TYPE
ANTENNA-TYPE-BEAM Identifier (FK)
ANTENNA-TYPE Identifier (FK)
PHASE-DIRECTION-TYPE Beam Direction Shift C
PHASE-DIRECTION-TYPE Discrete Direction Pha
FREQUENCY-DIRECTION-TYPE
ANTENNA-TYPE-BEAM Identifier (FK)
ANTENNA-TYPE Identifier (FK)
FREQUENCY-DIRECTION-TYPE Beam Direction Shift Frequency Change Rate
FREQUENCY-DIRECTION-TYPE Total Direction Frequency Quantity
BEAM-FORMATION-METHOD
SCAN-MODE Identifier (FK)

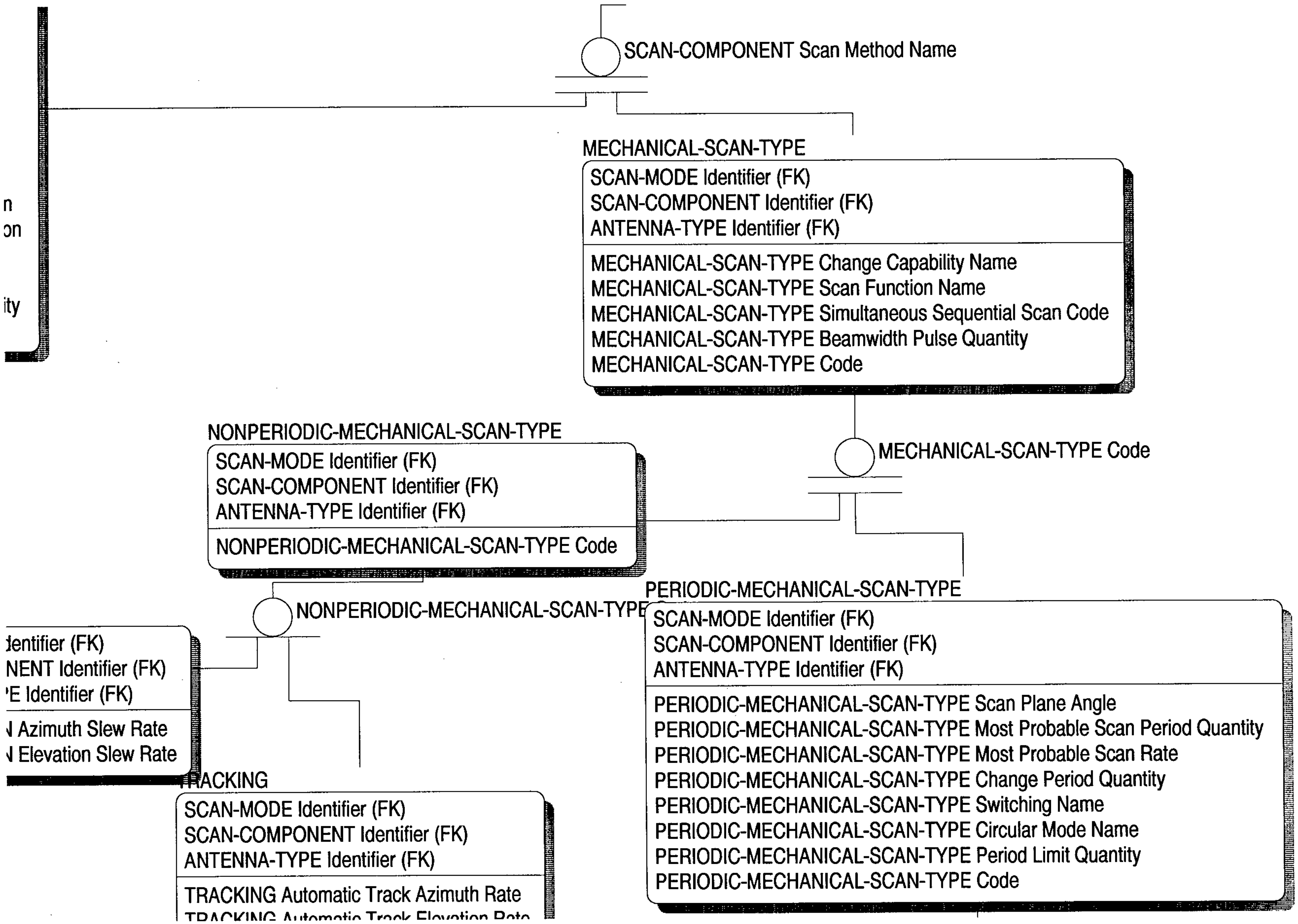


emits and receives along
has
P
BEAM-DIRECTION-METHOD
ANTENNA-TYPE-BEAM Identifier (FK)
ANTENNA-TYPE Identifier (FK)
BEAM-DIRECTION-METHOD Azimuth Electronic Direction Limit Angle
BEAM-DIRECTION-METHOD Elevation Electronic Direction Limit Angle
BEAM-DIRECTION-METHOD Total Array Beam Position Quantity
BEAM-DIRECTION-METHOD Type Name
BEAM-DIRECTION-METHOD Type Name
HYBRID-APERTURE-TYPE
APERTURE-REPRESENTATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
HYBRID-APERTURE-TYPE Diameter Dimension
HYBRID-APERTURE-TYPE Height Dimension
HYBRID-APERTURE-TYPE Horizontal Amplitude Distril
HYBRID-APERTURE-TYPE Vertical Amplitude Distribut
HYBRID-APERTURE-TYPE Horizontal Phase Distributi
HYBRID-APERTURE-TYPE Vertical Phase Distribution
DIRECTION
ANTENNA-TYPE-BEAM-WIDTH
ANTENNA-TYPE-BEAM Identifier (FK)
ANTENNA-TYPE-BEAM-WIDTH Frequency Rate
ANTENNA-TYPE Identifier (FK)
ANTENNA-TYPE-BEAM-WIDTH Horizontal Angle
ANTENNA-TYPE-BEAM-WIDTH Vertical Angle
NON-SCA
ANTENN
t Change Rate
hase Quantity
SCANNING-ANTENNA-TYPE
ANTENNA-TYPE Identifier (FK)
SCANNING-ANTENNA-TYPE Rotation Direction Name
SCANNING-ANTENNA-TYPE Blanked Scan Duty Cycle Rate
TIME-DELAY-DIRECTION-TYPE
ANTENNA-TYPE-BEAM Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TIME-DELAY-DIRECTION-TYPE Beam Direction Shift Delay Increment Rate
TIME-DELAY-DIRECTION-TYPE Discrete Direction Delay Quantity
ELECTRONIC-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
ELECTRONIC-SCAN-TYPE Scan Plane Angle


ANTENNA-TYPE Identifier (FK)
CIRCULAR-APERTURE-TYPE Diameter Dimension
CIRCULAR-APERTURE-TYPE Amplitude Distribution Name
CIRCULAR-APERTURE-TYPE Phase Distribution Name
tribution Name
ution Name
ution Name
n Name
NAL-ANTENNA-TYPE Scanning Type Name
RECTANGULAR-APERTURE-TYPE
APERTURE-REPRESENTATION Identifier (FK)
ANTENNA-TYPE Identifier (FK)
RECTANGULAR-APERTURE-TYPE Height Dimension
RECTANGULAR-APERTURE-TYPE Horizontal Amplitude Distribution Name
RECTANGULAR-APERTURE-TYPE Vertical Amplitude Distribution Name
RECTANGULAR-APERTURE-TYPE Horizontal Phase Distribution Name
RECTANGULAR-APERTURE-TYPE Vertical Phase Distribution Name
RECTANGULAR-APERTURE-TYPE Width Dimension
CANNING-ANTENNA-TYPE
NNA-TYPE Identifier (FK)
SCAN-MODE
SCAN-MODE Identifier
ANTENNA-TYPE Identifier (FK)
SCAN-MODE Scan Name
operates in
P
consists of
P
SCAN-COMPONENT
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier
ANTENNA-TYPE Identifier (FK)
SCAN-COMPONENT Scan Method Name
SCAN-COMPONENT Scan Plane Angle


SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
BEAM-FORMATION-METHOD Period Quantity
BEAM-FORMATION-METHOD Scan Limit Azimuth Angle
BEAM-FORMATION-METHOD Azimuth Shift Rate
BEAM-FORMATION-METHOD Scan Limit Elevation Angle
BEAM-FORMATION-METHOD Beam Shift Elevation Rate
BEAM-FORMATION-METHOD Beam Dwell Period Quantity
BEAM-FORMATION-METHOD Pulse Beam Dwell Quantity
BEAM-FORMATION-METHOD Amplitude Change Beam Shift Rate
BEAM-FORMATION-METHOD Total Beam Position Quantity
BEAM-FORMATION-METHOD Type Code
BEAM-FORMATION-METHOD Maximum Horizontal Steering Angle
BEAM-FORMATION-METHOD Maximum Horizontal Steering Rate
BEAM-FORMATION-METHOD Maximum Vertical Steering Angle
BEAM-FORMATION-METHOD Maximum Vertical Steering Rate
1
BEAM-FORMATION-METHOD Type Code
PHASE-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
PHASE-SCAN-TYPE Beam Shift Phase Change Rate
PHASE-SCAN-TYPE Discrete Phase Quantity
PHASE-SCAN-TYPE Shift Phase Period Quantity
TIME-DELAY-STEERING
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TIME-DELAY-STEERING Beam Shift Delay Increment Rate
TIME-DELAY-STEERING Discrete Delay Quantity
TIME-DELAY-STEERING Shift Delay Period Quantity
FREQUENCY-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
FREQUENCY-SCAN-TYPE Beam Shift Frequency Change Rate
FREQUENCY-SCAN-TYPE Total Frequency Step Quantity

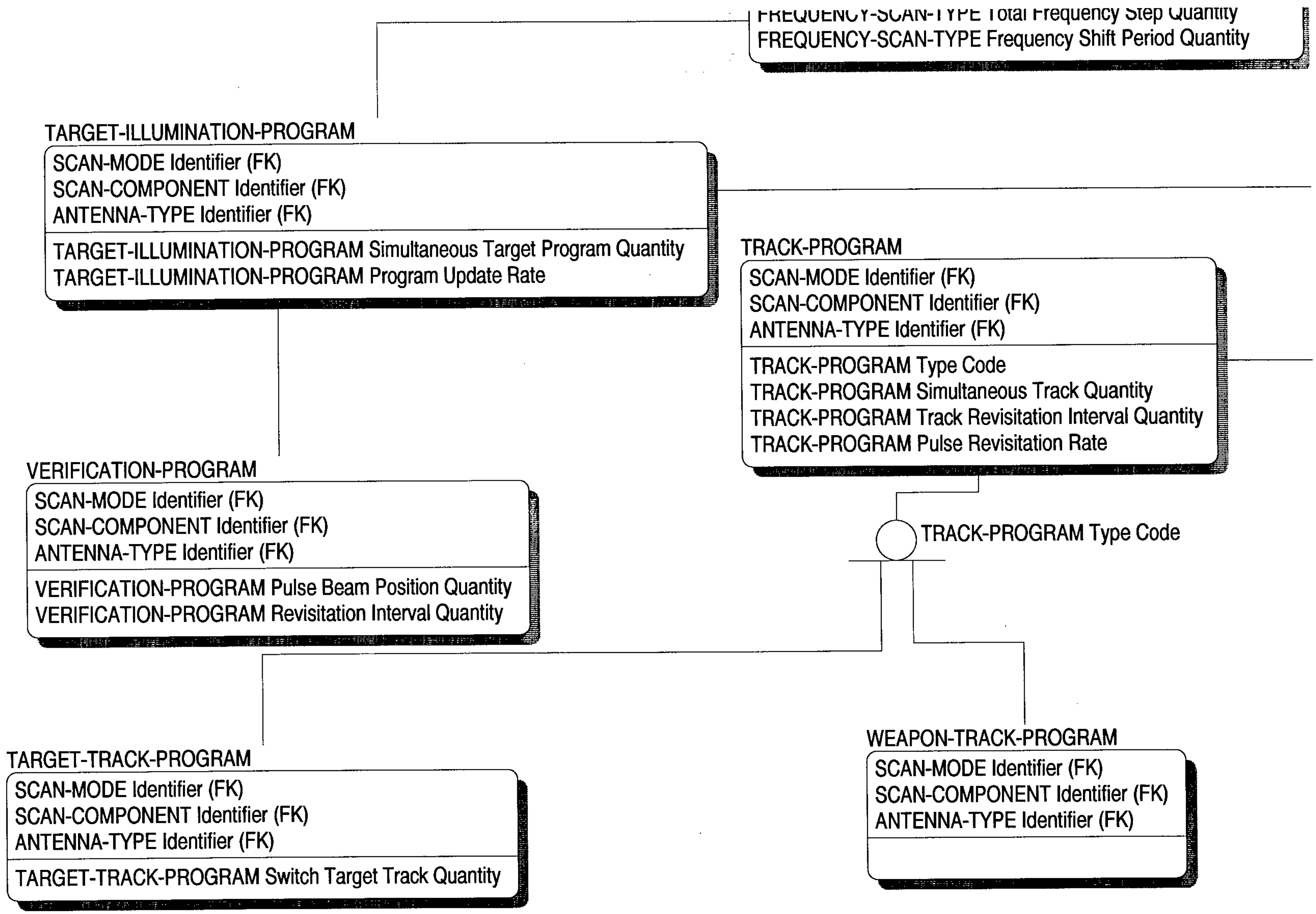

is implemented via
is used to perform
ELECTRONIC-SCAN-TYPE Beam Motion Name
ELECTRONIC-SCAN-TYPE Array Design Name
ELECTRONIC-SCAN-TYPE Tilt Angle
ELECTRONIC-SCAN-TYPE Array Face Quantity
ELECTRONIC-SCAN-TYPE Face Coverage Angle
ELECTRONIC-SCAN-TYPE Element Type Name
ELECTRONIC-SCAN-TYPE Total Antenna Element Quantity
ELECTRONIC-SCAN-TYPE Element Spacing Azimuth Dimension
ELECTRONIC-SCAN-TYPE Element Spacing Elevation Dimension
ELECTRONIC-SCAN-TYPE Element Distribution Text
ELECTRONIC-SCAN-TYPE Beamwidth Variation Quantity
ELECTRONIC-SCAN-TYPE Maximum Managed Program Quantity
ELECTRONIC-SCAN-TYPE Identifier
SCAN-FUNCTION
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
SCAN-FUNCTION Program Description Text
SCAN-FUNCTION Program Type Code
P
Electronic-Scan-Type
PERIODIC-ELECTRONIC-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
PERIODIC-ELECTRONIC-SCAN-TYPE Most Probable Azimuth Beam Switch Rate
PERIODIC-ELECTRONIC-SCAN-TYPE Most Probable Beam Switching Elevation Rate
PERIODIC-ELECTRONIC-SCAN-TYPE Azimuth Beam Position Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Azimuth Pulse Scan Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Elevation Beam Position Scan Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Elevation Pulse Scan Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Scan Mode Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Change Scan Mode Period Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Switching Name
PERIODIC-ELECTRONIC-SCAN-TYPE Program Period Limit Quantity
SCAN-FUNCTION Program Type Code
MANUAL-SCAN
ate

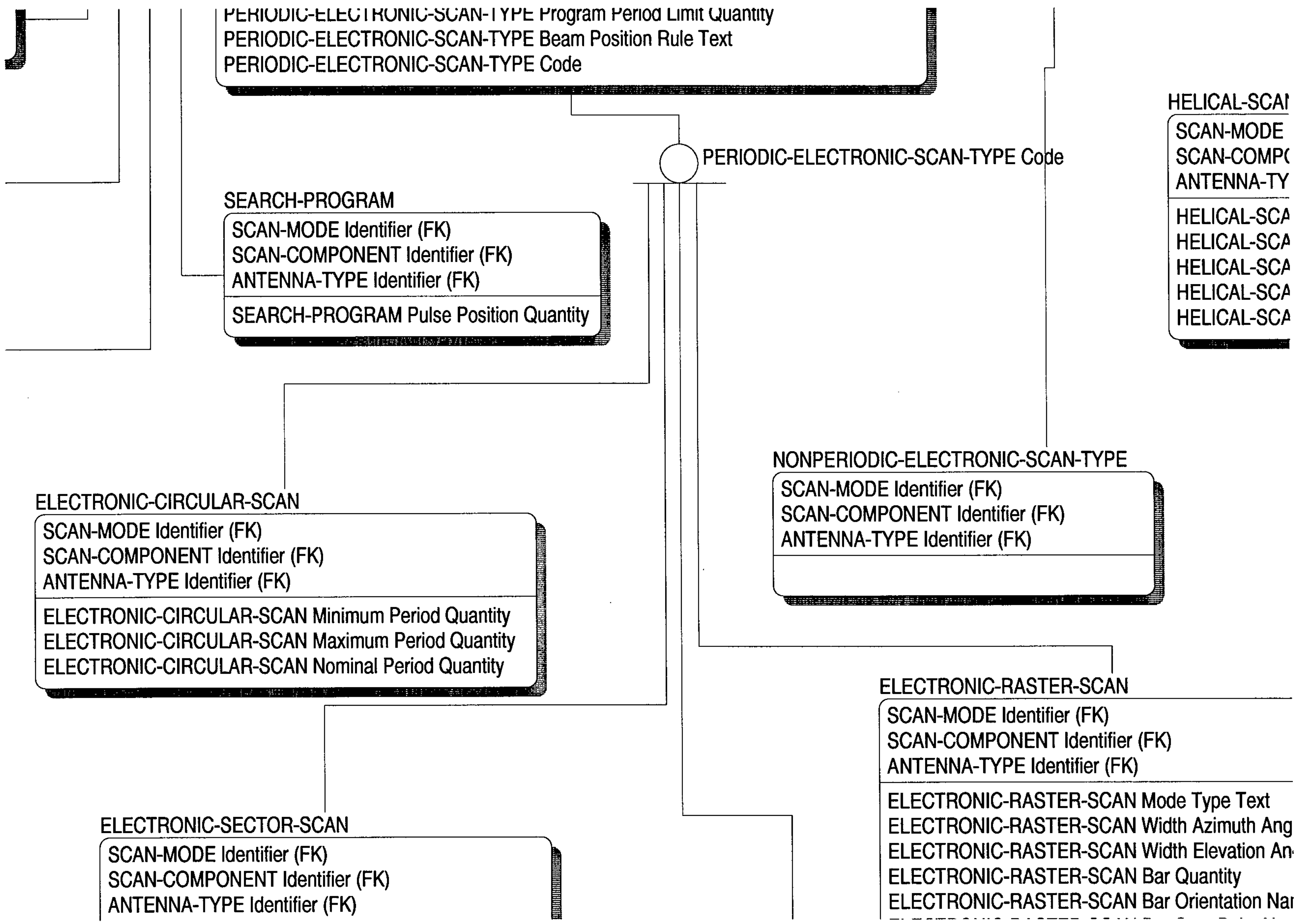

SCAN-COMPONENT Scan Method Name
MECHANICAL-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
MECHANICAL-SCAN-TYPE Change Capability Name
MECHANICAL-SCAN-TYPE Scan Function Name
MECHANICAL-SCAN-TYPE Simultaneous Sequential Scan Code
MECHANICAL-SCAN-TYPE Beamwidth Pulse Quantity
MECHANICAL-SCAN-TYPE Code
MECHANICAL-SCAN-TYPE Code
NONPERIODIC-MECHANICAL-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
NONPERIODIC-MECHANICAL-SCAN-TYPE Code
PERIODIC-MECHANICAL-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
PERIODIC-MECHANICAL-SCAN-TYPE Scan Plane Angle
PERIODIC-MECHANICAL-SCAN-TYPE Most Probable Scan Period Quantity
PERIODIC-MECHANICAL-SCAN-TYPE Most Probable Scan Rate
PERIODIC-MECHANICAL-SCAN-TYPE Change Period Quantity
PERIODIC-MECHANICAL-SCAN-TYPE Switching Name
PERIODIC-MECHANICAL-SCAN-TYPE Circular Mode Name
PERIODIC-MECHANICAL-SCAN-TYPE Period Limit Quantity
PERIODIC-MECHANICAL-SCAN-TYPE Code
NONPERIODIC-MECHANICAL-SCAN-TYPE
dentifier (FK)
NENT Identifier (FK)
E Identifier (FK)
Azimuth Slew Rate
Elevation Slew Rate
TRACKING
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TRACKING Automatic Track Azimuth Rate

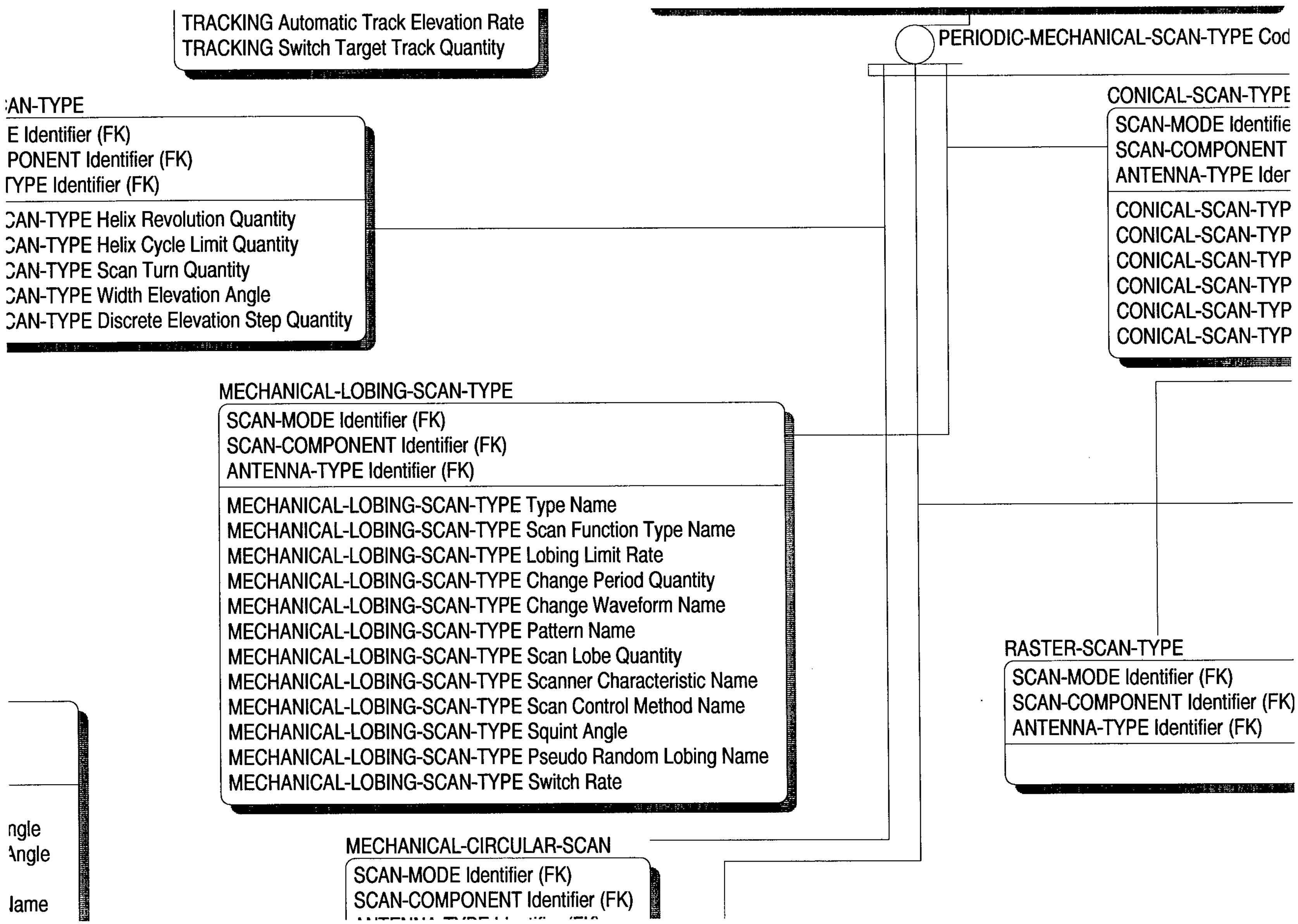
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE

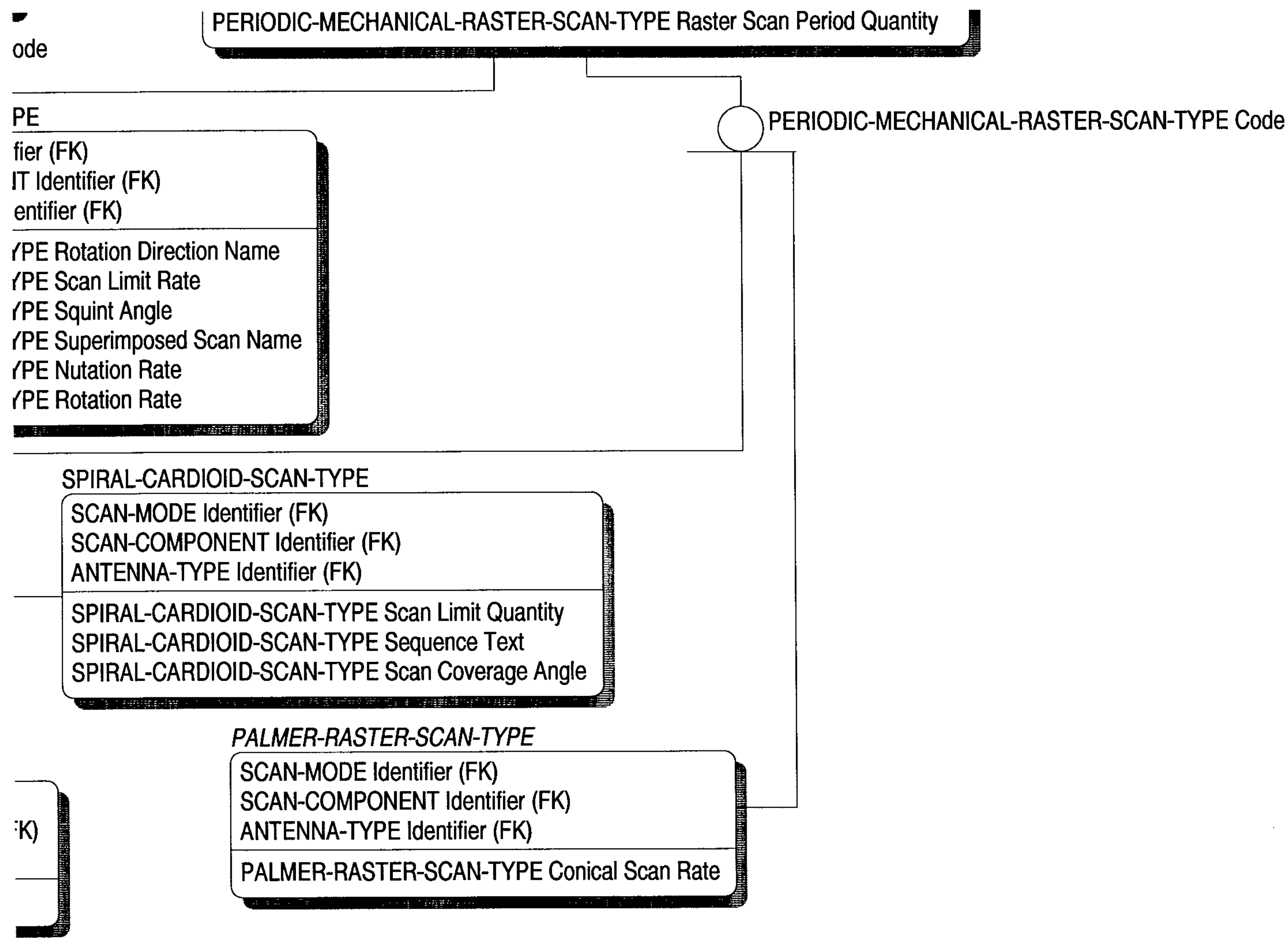
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)

PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Code
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Type Text
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Width Azimuth Angle
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Width Elevation Angle
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Bar Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Bar Orientation Name
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Bar Start Point Name
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Bar Period Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Bar Beam Position Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Blank Bar Period Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Flyback Period Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Scan Period Quantity


FREQUENCY-SCAN-TYPE Total Frequency Step Quantity
FREQUENCY-SCAN-TYPE Frequency Shift Period Quantity
TARGET-ILLUMINATION-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TARGET-ILLUMINATION-PROGRAM Simultaneous Target Program Quantity
TARGET-ILLUMINATION-PROGRAM Program Update Rate
TRACK-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TRACK-PROGRAM Type Code
TRACK-PROGRAM Simultaneous Track Quantity
TRACK-PROGRAM Track Revisitation Interval Quantity
TRACK-PROGRAM Pulse Revisitation Rate
VERIFICATION-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
VERIFICATION-PROGRAM Pulse Beam Position Quantity
VERIFICATION-PROGRAM Revisitation Interval Quantity
TRACK-PROGRAM Type Code
TARGET-TRACK-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
TARGET-TRACK-PROGRAM Switch Target Track Quantity
WEAPON-TRACK-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)


PERIODIC-ELECTRONIC-SCAN-TYPE Program Period Limit Quantity
PERIODIC-ELECTRONIC-SCAN-TYPE Beam Position Rule Text
PERIODIC-ELECTRONIC-SCAN-TYPE Code
PERIODIC-ELECTRONIC-SCAN-TYPE Code
SEARCH-PROGRAM
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
SEARCH-PROGRAM Pulse Position Quantity
HELICAL-SCAN
SCAN-MODE
SCAN-COMP
ANTENNA-TY
HELICAL-SCA
HELICAL-SCA
HELICAL-SCA
HELICAL-SCA
HELICAL-SCA
NONPERIODIC-ELECTRONIC-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
ELECTRONIC-CIRCULAR-SCAN
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
ELECTRONIC-CIRCULAR-SCAN Minimum Period Quantity
ELECTRONIC-CIRCULAR-SCAN Maximum Period Quantity
ELECTRONIC-CIRCULAR-SCAN Nominal Period Quantity
ELECTRONIC-RASTER-SCAN
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
ELECTRONIC-RASTER-SCAN Mode Type Text
ELECTRONIC-RASTER-SCAN Width Azimuth Ang
ELECTRONIC-RASTER-SCAN Width Elevation An
ELECTRONIC-RASTER-SCAN Bar Quantity
ELECTRONIC-RASTER-SCAN Bar Orientation Nai
ELECTRONIC-SECTOR-SCAN
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)


TRACKING Automatic Track Elevation Rate
TRACKING Switch Target Track Quantity
PERIODIC-MECHANICAL-SCAN-TYPE Cod
CONICAL-SCAN-TYPE
SCAN-MODE Identifie
SCAN-COMPONENT
ANTENNA-TYPE Ider
CONICAL-SCAN-TYP
CONICAL-SCAN-TYP
CONICAL-SCAN-TYP
CONICAL-SCAN-TYP
CONICAL-SCAN-TYP
CONICAL-SCAN-TYP
AN-TYPE
E Identifier (FK)
PONENT Identifier (FK)
TYPE Identifier (FK)
CAN-TYPE Helix Revolution Quantity
CAN-TYPE Helix Cycle Limit Quantity
CAN-TYPE Scan Turn Quantity
CAN-TYPE Width Elevation Angle
CAN-TYPE Discrete Elevation Step Quantity
MECHANICAL-LOBING-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
MECHANICAL-LOBING-SCAN-TYPE Type Name
MECHANICAL-LOBING-SCAN-TYPE Scan Function Type Name
MECHANICAL-LOBING-SCAN-TYPE Lobing Limit Rate
MECHANICAL-LOBING-SCAN-TYPE Change Period Quantity
MECHANICAL-LOBING-SCAN-TYPE Change Waveform Name
MECHANICAL-LOBING-SCAN-TYPE Pattern Name
MECHANICAL-LOBING-SCAN-TYPE Scan Lobe Quantity
MECHANICAL-LOBING-SCAN-TYPE Scanner Characteristic Name
MECHANICAL-LOBING-SCAN-TYPE Scan Control Method Name
MECHANICAL-LOBING-SCAN-TYPE Squint Angle
MECHANICAL-LOBING-SCAN-TYPE Pseudo Random Lobing Name
MECHANICAL-LOBING-SCAN-TYPE Switch Rate
RASTER-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
ngle
Angle
lame
MECHANICAL-CIRCULAR-SCAN
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)


ode
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Raster Scan Period Quantity
PERIODIC-MECHANICAL-RASTER-SCAN-TYPE Code
PE
fier (FK)
IT Identifier (FK)
entifier (FK)
/PE Rotation Direction Name
/PE Scan Limit Rate
/PE Squint Angle
/PE Superimposed Scan Name
/PE Nutation Rate
/PE Rotation Rate
SPIRAL-CARDIOID-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
SPIRAL-CARDIOID-SCAN-TYPE Scan Limit Quantity
SPIRAL-CARDIOID-SCAN-TYPE Sequence Text
SPIRAL-CARDIOID-SCAN-TYPE Scan Coverage Angle
PALMER-RASTER-SCAN-TYPE
SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)
PALMER-RASTER-SCAN-TYPE Conical Scan Rate
K)

ANTENNA-TYPE Identifier (FK)

ELECTRONIC-SECTOR-SCAN Type Name
ELECTRONIC-SECTOR-SCAN Scan Direction Text
ELECTRONIC-SECTOR-SCAN Scan Flyback Quantity
ELECTRONIC-SECTOR-SCAN Width Azimuth Angle
ELECTRONIC-SECTOR-SCAN Width Elevation Angle

ELECTRONIC-RASTER-SCAN Bar Orientation Na
ELECTRONIC-RASTER-SCAN Bar Start Point Nan
ELECTRONIC-RASTER-SCAN Bar Period Quantity
ELECTRONIC-RASTER-SCAN Beam Position Qua
ELECTRONIC-RASTER-SCAN Flyback Quantity

ELECTRONIC-LOBING-SCAN

SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)

ELECTRONIC-LOBING-SCAN Monopulse Type Name
ELECTRONIC-LOBING-SCAN Waveform Change Name

ame
tity
uantity

ANTENNA-TYPE Identifier (FK)

MECHANICAL-SECTOR-SCAN-TYPE

SCAN-MODE Identifier (FK)
SCAN-COMPONENT Identifier (FK)
ANTENNA-TYPE Identifier (FK)

MECHANICAL-SECTOR-SCAN-TYPE Scan Pattern Description Text
MECHANICAL-SECTOR-SCAN-TYPE Scan Relation Description Text
MECHANICAL-SECTOR-SCAN-TYPE Scan Component Azimuth Period Quantity
MECHANICAL-SECTOR-SCAN-TYPE Scan Component Elevation Period Quantity
MECHANICAL-SECTOR-SCAN-TYPE Sector Scan Rate
MECHANICAL-SECTOR-SCAN-TYPE Sector Beamwidth Ratio Rate
MECHANICAL-SECTOR-SCAN-TYPE Sector Type Name
MECHANICAL-SECTOR-SCAN-TYPE Scan Direction Name
MECHANICAL-SECTOR-SCAN-TYPE Flyback Period Quantity
MECHANICAL-SECTOR-SCAN-TYPE Sector Width Azimuth Angle
MECHANICAL-SECTOR-SCAN-TYPE Sector Width Elevation Angle

Entity Name: ANTENNA-LEAD-IN-ASSEMBLY
Entity Definition: The equipment that connects an antenna with its possible receivers, transmitters, and transceivers.
Table Name: ANTENNA_COUPLING_U
Attribute Names: ANTENNA-TYPE Identifier (PK) (FK)
Column Names: Materiel_ID CHAR(35) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: ANTENNA-TYPE Identifier References: ANTENNA-TYPE
FK Referential Integ: U - Restrict I - Restrict D - Restrict CU - Restrict PI - CD -

Attribute Names: RF Equipment ID (PK) (FK)
Column Names: Materiel_ID CHAR(18) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: RF Equipment ID References: TRANSMISSION-RECEPTION-EQUIPMENT-TYPE
FK Referential Integ: U - Restrict I - Restrict D - Restrict CU - Restrict PI - CD -

Entity Name: ANTENNA-TYPE
Entity Definition: The classification of a device for the collection or radiation of electromagnetic signals.
Table Name: ANTENNA
Attribute Names: ANTENNA-TYPE Identifier (PK) (FK)
Column Names: Materiel_ID CHAR(35) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: ANTENNA-TYPE Identifier References: EQUIPMENT
FK Referential Integ: U - Restrict I - Restrict D - Cascade CU - PI - CD -

Attribute Names: ANTENNA-TYPE Use Name
Column Names: ant_use_name CHAR(35)

Attribute Names: ANTENNA-TYPE Nomenclature Identifier
Column Names: Ant_Nomenclature CHAR(35)

Attribute Names: ANTENNA-TYPE Application Mode Name
Column Names: Ant_Application CHAR(20)

Attribute Names: ANTENNA-TYPE Distance Dimension
Column Names: Ant_Distance FLOAT(6,2)

Attribute Names: ANTENNA-TYPE Tracking Name
Column Names: Ant_Tracking CHAR(15)

Attribute Names: ANTENNA-TYPE Transmission Feed Name
Column Names: Ant_Transmission CHAR(15)

Attribute Names: ANTENNA-TYPE Drive Type Name
Column Names: Ant_Drive_Type CHAR(250)

Attribute Names: ANTENNA-TYPE Maximum Slew Rate
Column Names: ant_max_slew_rate FLOAT(10)

Attribute Names: ANTENNA-TYPE Maximum Acceleration Rate
Column Names: ant_max_accelerati FLOAT(7)

Attribute Names: ANTENNA-TYPE Length Dimension
Column Names: ant_length FLOAT(6,2)

Attribute Names: ANTENNA-TYPE Weight
Column Names: Ant_Weight FLOAT(10)

Attribute Names: ANTENNA-TYPE Directional Type Name
Column Names: ant_direction_type CHAR(250)

Attribute Names: ANTENNA-TYPE Physical Type Name
Column Names: Ant_Physical_Type CHAR(250)

Attribute Names: ANTENNA-TYPE Low Range Frequency Rate
Column Names: Ant_Low_Freq FLOAT(7)

Attribute Names: ANTENNA-TYPE High Range Frequency Rate
Column Names: Ant_High_Freq FLOAT(7)

Attribute Names: ANTENNA-TYPE Mount Feed Point Dimension
Column Names: Ant_Mount_Feed FLOAT(6,2)

Attribute Names: ANTENNA-TYPE Mount Point Dimension
Column Names: Ant_Mount_Point FLOAT(6,2)

Attribute Names: ANTENNA-TYPE Efficiency Rate
Column Names: Ant_Efficiency FLOAT(10)

Attribute Names: ANTENNA-TYPE Maximum Effective Radiated Power Rate
Column Names: Ant_Max_ERP FLOAT(7)

Attribute Names: ANTENNA-TYPE Ground Stake Code
Column Names: Ant_Ground_Stake CHAR(1)

Entity Name: ANTENNA-TYPE-BAND
Entity Definition: The operation of an antenna in a frequency range.
Table Name: ANTENNA_BAND_SELEC
Attribute Names: ANTENNA-TYPE Identifier (PK) (FK) (AK)
Column Names: Materiel_ID CHAR(35) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: ANTENNA-TYPE Identifier References: ANTENNA-TYPE
FK Referential Integ: U - Restrict I - Restrict D - Restrict CU - Restrict PI - CD -

Attribute Names: ANTENNA-TYPE-BAND Selection Identifier (PK)
Column Names: Band_Selection_ID CHAR(35) NOT NULL

Primary Key: Band_Selection_ID

Attribute Names: ANTENNA-TYPE-BAND Low Frequency Rate (AK)
Column Names: Ant_Band_Low_Freq FLOAT(7) NOT NULL

Attribute Names: ANTENNA-TYPE-BAND High Frequency Rate (AK)
Column Names: Ant_Band_High_Freq FLOAT(7) NOT NULL

Attribute Names: ANTENNA-TYPE-BAND Antenna Gain Rate
Column Names: Ant_Band_Gain FLOAT(7)

Attribute Names: ANTENNA-TYPE-BAND Impedence Rate
Column Names: Ant_Band_Impedence FLOAT(7)

Entity Name: ANTENNA-TYPE-INTEGRAL-MAST
Entity Definition: The supporting structure for raising an antenna above the ground or the platform provided as an integral part of the antenna assembly.
Table Name: Integral_Mast
Attribute Names: ANTENNA-TYPE Identifier (PK) (FK)
Column Names: Materiel_ID CHAR(35) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: ANTENNA-TYPE Identifier References: ANTENNA-TYPE
FK Referential Integ: U - Restrict I - Restrict D - Restrict CU - Restrict PI - CD -

Attribute Names: ANTENNA-TYPE-INTEGRAL-MAST Identifier (PK)
Column Names: Mast_ID CHAR(35) NOT NULL
Primary Key: Mast_ID

Attribute Names: ANTENNA-TYPE-INTEGRAL-MAST Antenna Mast Section Quantity
Column Names: Mast_Num_Sections INTEGER

Attribute Names: ANTENNA-TYPE-INTEGRAL-MAST Minimum Height Dimension
Column Names: Mast_Height_Min FLOAT(6,2)

Attribute Names: ANTENNA-TYPE-INTEGRAL-MAST Maximum Height Dimension
Column Names: Mast_Height_Max FLOAT(6,2)

Entity Name: EQUIPMENT
Entity Definition: Materiel that is not consumed in use, retaines it original identity during the period of use, and requires accountability. All non- expendable items needed tc
Table Name: EQUIPMENT
Attribute Names: Materiel Identifier (PK)
Column Names: Materiel_ID CHAR(18) NOT NULL
Primary Key: Materiel_ID

Attribute Names: EQUIPMENT Type Code
Column Names: Equipment_Type CHAR(18)

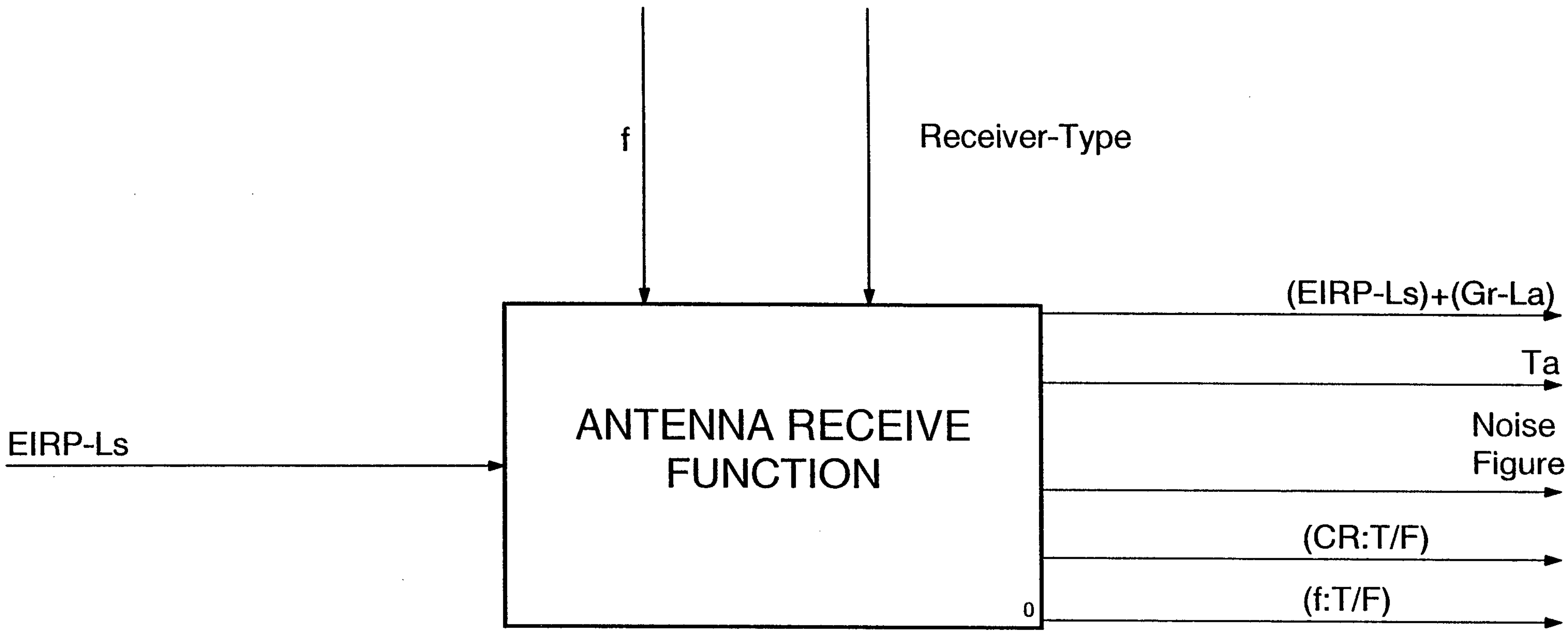
Entity Name: TRANSMISSION-RECEPTION-EQUIPMENT-TYPE
Entity Definition: The classification of the equipment that is a transmitter, receiver or transceiver.

Table Name: RF_EQUIPMENT
Attribute Names: RF Equipment ID (PK) (FK)
Column Names: Materiel_ID CHAR(18) NOT NULL
Primary Key: Materiel_ID
Foreign Keys: RF Equipment ID   References: EQUIPMENT
FK Referential Integ: U - Cascade   I - Restrict   D - Cascade   CU - Restrict   PI -   CD -

Appendix B. IDEF0 Antenna Model

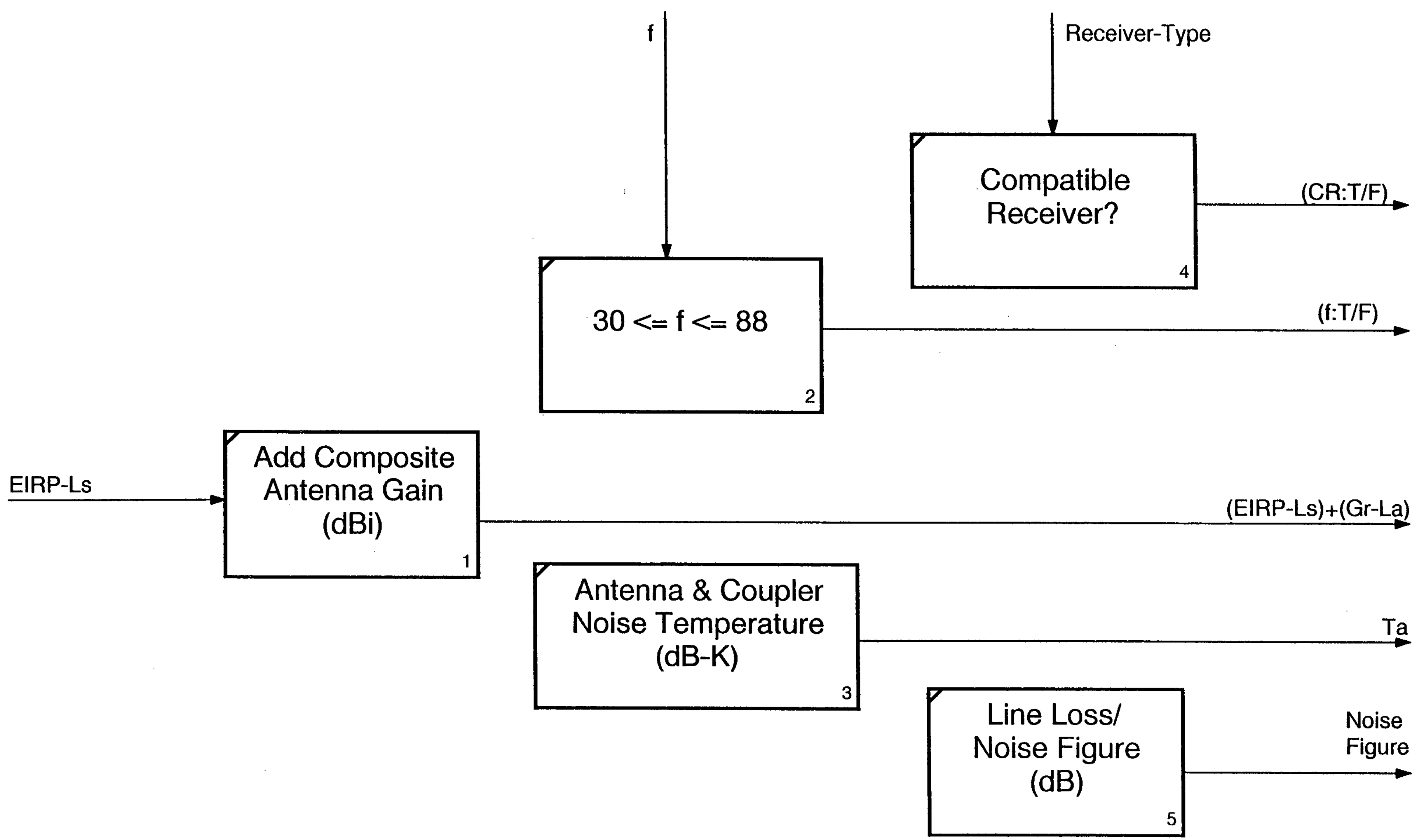
The following diagrams show the activity model of the OE-254()/GRC antenna and schema for each activity, input, control and output.

USED AT:	AUTHOR: Timothy R. Schmoyer, CPT	DATE: 15 Feb. 1997	WORKING	READER	DATE	CONTEXT:
	PROJECT: OE-254()/GRC Model Project	REV: 17 Feb. 1997	DRAFT			TOP
			■ RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10		PUBLICATION			



NODE: A-0	TITLE: ANTENNA RECEIVE FUNCTION	NUMBER:

USED AT:	AUTHOR: Timothy R. Schmoyer, CPT	DATE: 15 Feb. 1997		WORKING	READER	DATE	CONTEXT:
	PROJECT: OE-254()/GRC Model Project	REV: 17 Feb. 1997		DRAFT			TOP
			■	RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION			A-0



NODE: A0	TITLE: ANTENNA RECEIVE FUNCTION	NUMBER:

Model Name: OE-254()/GRC
Definition: This model represents the activities of the antenna when receiving a signal across a path and providing that signal to the receiver to determine bit energy per noise power spectral density and link margin.
Scope: OE-254()/GRC operations; path to receiver.
Viewpoint: At the antenna.
Purpose: To standardize antenna models for for net centric simulation.
Source: MIL-A-49204A
TC 24-24
Sklar, Bernard; DIGITAL COMMUNICATIONS Fundamentals and Applications; PTR Prentice Hall, Englewood Cliffs, NJ 07632; 1988
Author Name: Timothy R. Schmoyer, CPT

Activity Name	Activity Definition	Input Name	Input Definition
ANTENNA RECEIVE FUNCTION	The process required in RF reception to add antenna effects to path loss and noise for determining bit energy per noise power spectral density and link margin	EIRP-Ls	Effective Isotro Power (dBW) from minus the path l
Add Composite Antenna Gain (dBi)	Antenna gain is a function of frequency and effective area. G=4(pi)Ae/(3E8/f)^2 Ae=(efficiency)Ap	EIRP-Ls	Effective Isotr Power (dBW) fro minus the path
30 <= f <= 88	Is the center frequency greater than 30 (MHz) and less than 88 (MHz)? Returns true if it is, and false if f falls outside		

ɔn	Control Name	Control Definition	Output Name
ropic Radiated ɔm the transmitter loss (dB).	f	Center frequency of transmitted signal over the radiowave propagation path which is compared with the frequency range of the antenna.	(EIRP-Ls)+(Gr-La)
	Receiver-Type	Indentity of Receiver which is compared with an array of Receivers to verify compatability.	Ta
			Noise Figure
			(CR:T/F)
			(f:T/F)
ropic Radiated ɔm the transmitter loss (dB).			(EIRP-Ls)+(Gr-La)
	f	Center frequency of transmitted signal over the radiowave propagation path which is compared with the frequency	(f:T/F)

Output Definition

Antenna gain (dBi) minus the line and coupler loss (dB) (which must be 3 dB or greater for this antenna) plus the Effective Isotropic Radiated Power (dBW) from the transmitter minus the path loss (dB).

Noise Temperature of the Antenna (dB-K).

Noise Figure of the cable and coupler (dB).

Returns a boolean value depending if Receiver-Type matched a value (NSN) in the Compatible Receiver Array.

Returns a boolean value depending if f is within antenna's frequency range.

Antenna gain (dBi) minus the line and coupler loss (dB) (which must be 3 dB or greater for this antenna) plus the Effective Isotropic Radiated Power (dBW) from the transmitter minus the path loss (dB).

Returns a boolean value depending if f is within antenna's frequency range.

	this range.
Antenna & Coupler Noise Temperature (dB-K)	The Noise Temperature into the Receiver Pre-amplifier.
Compatible Receiver?	Compares Receive-Type with list of compatible Receivers: 5820-00-857-0759 (AN/PRC-25) 5820-00-223-7412 (AN/VRC-12) 5820-01-151-9915 (AN/PRC-119) 5820-01-151-9916 (AN/VRC-87) 5820-01-151-9917 (AN/VRC-88) 5820-01-151-9918 (AN/VRC-89) 5820-01-151-9919 (AN/VRC-90) 5820-01-151-9920 (AN/VRC-91) 5820-01-151-9921 (AN/VRC-92)
Line Loss/ Noise Figure (dB)	Added to Losses/Noise in Receiver for Lo in Link Margin equation.

	range of the antenna.	
		Ta
Receiver-Type	Indentity of Receiver which is compared with an array of Receivers to verify compatability.	(CR:T/F)
		Noise Figure

Noise Temperature of the Antenna (dB-K).

Returns a boolean value depending if Receiver-Type matched a value (NSN) in the Compatible Receiver Array.

Noise Figure of the cable and coupler (dB).

Appendix C. C4I Modeling and Simulation

In this appendix, the hierarchy of communications systems is described and a brief discussion of types of simulations is provided. A detailed discussion is also provided concerning the modeling of radiowave propagation to arrive at accurate received power and link margin calculations. The importance of developing simulations based on accurate models is discussed in the need for automation and determining the perishability of information.

C.1 Network Layering

Communications networks are modeled according to the network architecture. The most common network architecture for computer networks is the International Standards Organization (ISO) Open Systems Interconnect (OSI) model (Figure C.1). There are other network architectures based on the OSI model, such as the Internet Protocol (IP) architecture, Systems Network Architecture, and Digital Network architecture, but the OSI model provides the framework(34). The network control functions are performed in the session, transport, network, and data link layer. The radios, antenna and associated hardware are the physical layer.

Application
Presentation
Session
Transportation
Network
Data Link
Physical

Figure C.1 Open Systems Interconnection Reference Model Architecture(34)

C.2 Abstract vs. Detail Modeling

The tension between detailed and abstract modeling (often referred to as the fidelity of the model) is a fundamental issue for the network simulator. Detailed models provide accurate results, but can take a long time to develop and consume large amounts of computer resources to generate results. Abstract models typically generate results much more quickly, but their results may not be useful if too many simplifying assumptions have been made in the abstraction(16, 22). The key input from the network engineer is recognition of the key components of the system that must be modeled in detail and what abstractions can be used to reduce simulation run times(4). An important consideration when modeling communications systems is communication realism in the network due to changes on the battlefield. The temporal effects of the environment on a radio path is often left out of many battlefield simulations encompassing communications. Environmental effects are modeled in the links, and are measured in data errors and received signal levels.

C.3 Real-Time vs. Analytical

There are two types of simulations in use today, real-time and analytical. Real-Time simulations provide output in near real-time (approximately every 5 seconds) for use by other simulations (distributed interactive simulations) or humans (training and war-gaming). Real-time simulations often required more abstraction to meet the 5 second requirement, however, communications realism in Real-Time simulations is necessary in a dynamic environment to provide feedback when communications are operating reliably, stressed or no longer possible. If a communications link experiences a loss for a fraction of a second, the digital message which had been transmitted over several seconds or minutes may have to be resent. However, in a voice message, the same error may cause an insignificant or even unnoticeable chirp. Analytical simulations can "elongate" time and be more detailed, but are less useful in interactive or distributed simulations. Communications realism is also important in these simulations to determine reliability and the causes of failures in the network.

Expanding the previous discussion, the use of "visual physics", the practice of making something look realistic, whether or not physically correct, is common in real-time simulations. By contrast, analytic simulations rely heavily on deterministic, discrete, closed, event-driven models which are reproducible and accurate. Many of the analytic models have undergone a rigorous validation and verification process to ensure the accuracy and truthfulness of their algorithm. In order to verify that the real-time simulation is accurately reflecting real world expectations, analytical simulations based on measured data must be completed(31). The verification, validation and accreditation process of a *realistic communications* algorithm is the basis for this thesis.

C.4 Constructive, Virtual and Live Simulations

Simulations are also described as constructive, virtual and live. Constructive simulations require little or no interaction from humans. These are usually purely analytical. Virtual simulations require some interaction between the computer and human. Manned simulators and war-gaming exercises are examples of virtual simulations. Live simulations are real-time experiments, training exercises, demonstrations and tests that take place in a field like environment(5).

C.5 Modeling Radiowave Propagation

Figure C.2 is a block diagram of a satellite communications link, emphasizing the sources of signal loss and noise. In the figure, a signal loss is distinguished from a noise source by a dot pattern or line pattern, respectively. The contributors of *both* signal loss and noise are identified by a crosshatched line pattern. There are 21 sources of degradation cataloged by the figure. This is only a partial list(32). This section will discuss the sources of signal loss and noise as a result of the environment.

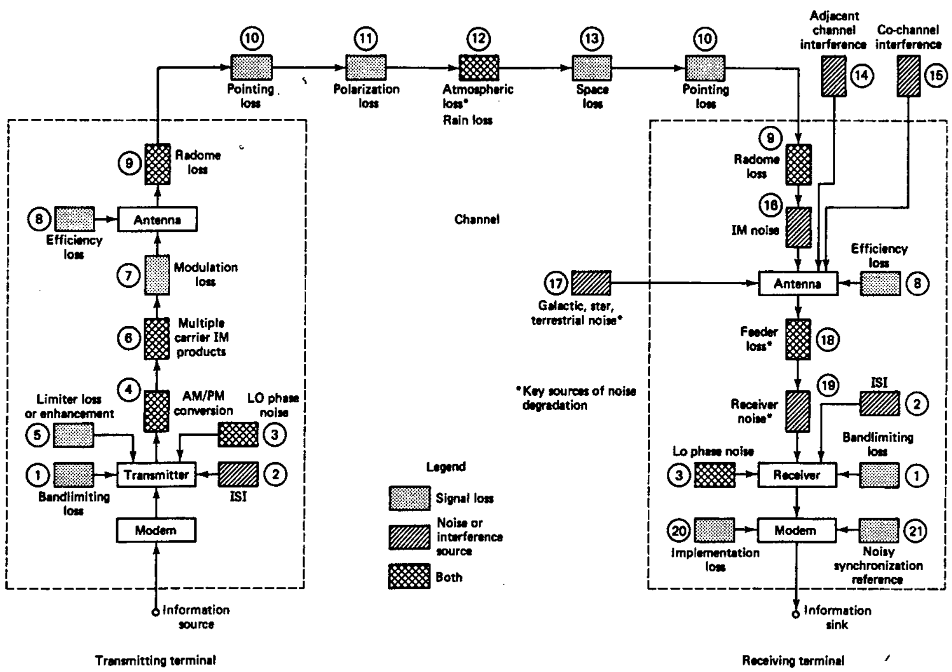


Figure C.2 Satellite Transmitter-to-Receiver Link with Typical Loss and Noise Sources(32)

There are many environmental factors which degrade communications over the path. Many of these factors are temporal and cause severe fading or increased noise when they occur. Figure C.3 shows some of the environmental factors which can affect communications(14).

C.6 Ionospheric effects

Radio frequency waves propagating in the ionosphere experience dissipative attenuation which becomes increasingly important with decreasing frequency. A principal mechanism of attenuation is collisions of free electrons with neutral atoms and molecules. An electromagnetic wave propagating in a plasma imparts an ordered component of velocity to the electrons but the electrons lose some of the associated energy in the collision process. Hence the electromagnetic wave is attenuated. For frequencies above about 30 MHz or for transverse propagation of the ordinary wave, the attenuation constant varies inversely with frequency squared and can be found using Equation C.1 (13):

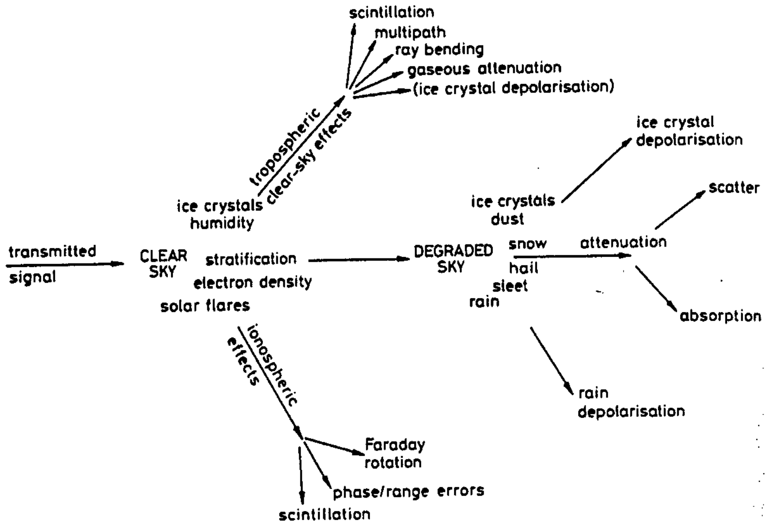


Figure C.3 Schematic Presentation of Propagation Impairment Mechanisms(14)

$$\alpha = 8.686 \left(\frac{N q^2 \nu}{2 m \epsilon_o n_r c \omega^2} \right) \qquad \mathrm{dB/m} \tag{C.1}$$

where:

N is the electron density of the ionosphere in electrons/m^3

n_r is the real part of the index of refraction

$q \; = 1.6022x10^{-19}$ C

$m \; = 9.1096x10^{-31}$ kg

$\epsilon_o \; = 8.854x10^{-12}$ F/m

$c \; = 2.9979x10^{8}$ m/s

$\omega \; = 2\pi f$ with f in Hz

and ν is the collision frequency in Hz.

C.7 *Clouds and Fog*

Clouds and fog are both composed of minute water droplets or ice crystals suspended in air. Fog forms near the Earth's surface, and clouds occur at higher levels. The water droplets in fog and clouds cause attenuation of electromagnetic waves and add noise by acting as an elementary antenna that radiates energy. The attenuation caused by clouds for frequencies between 1GHz and 50GHz can be found by using Equation C.2,(13)

$$\alpha_p = \left(0.4343\frac{6\pi}{\lambda}Im\left[-\frac{K_c-1}{K_c+2}\right]\right)\rho_l \quad \text{dB/km} \tag{C.2}$$

where

α_p is the attenuation coefficient (dB/km)

λ_{cm} is the wavelength (cm)

K_c is the complex relative dielectric constant of water

ρ_l is the liquid water content of the cloud (g/m^3)

and the noise caused by the cloud can be found using Equation C.3,

$$T_b = T_s e^{-\tau} + T_i(1 - e^{-T}) \tag{C.3}$$

where T_b is the brightness temperature when a source at a temperature of T_s is viewed through an absorbing region having an effective temperature of T_i. The parameter τ represents the integral of the power attenuation constant across the path. This noise, T_b is then added to the system temperature, T_{sys}, through the antenna(13). The attenuation and noise temperature as a result of clouds as a function of frequency is shown in Table C.1.

Case	Lower Cloud				Upper Cloud				Remarks	S-Band (2.3 GHz) Zenith		X-Band (8.5 GHz) Zenith		X-Band (10 GHz) Zenith	
	Density g/m^3	Base km	Top km	Thick-ness km	Density g/m^3	Base km	Top km	Thick-ness km		T(K)	A(dB)	T(K)	A(dB)	T(K)	A(dB)
1	-	-	-	-	-	-	-	-	Clear Air	2.15	.035	2.78	0.45	3.05	.049
2	0.2	1.0	1.2	0.2	-	-	-	-	Thin Clouds	2.16	.036	2.90	.047	3.22	.052
3	-	-	-	-	0.2	3.0	3.2	0.2		2.16	.036	2.94	.048	3.28	.053
4	0.5	1.0	1.5	0.5	-	-	-	-		2.20	.036	3.55	.057	4.12	.066
5	-	-	-	-	0.5	3.0	3.5	0.5		2.22	.037	3.83	.062	4.50	.073
6	0.5	1.0	2.0	1.0	-	-	-	-	Medium Clouds	2.27	.037	4.38	.070	5.27	.084
7	-	-	-	-	0.5	3.0	4.0	1.0		2.31	.038	4.96	.081	6.06	.098
8	0.5	1.0	2.0	1.0	0.5	3.0	4.0	1.0	Heavy Clouds	2.43	.040	6.55	.105	8.25	.133
9	0.7	1.0	2.0	1.0	0.7	3.0	4.0	1.0		2.54	.042	8.04	.130	10.31	.166
10	1.0	1.0	2.0	1.0	1.0	3.0	4.0	1.0		2.70	.044	10.27	.166	13.35	.216
11	1.0	1.0	2.5	1.5	1.0	3.5	5.0	1.5	Very Heavy Clouds	3.06	.050	14.89	.245	19.66	.326
12	1.0	1.0	3.0	2.0	1.0	4.0	6.0	2.0		3.47	.057	20.20	.340	26.84	.457

Table C.1 Sample Cloud Models and S-, X-, K-Band Zenith Effects(14)

C.8 Rain

Raindrops, like clouds but more severe, cause attenuation of radio waves by both absorption and scatter. Absorption involves dissipation of some of the energy of an electromagnetic wave as heat. Scatter involves diversion of some of the energy of the wave into directions other than the forward direction. The attenuation coefficient from rain can be found using Equation C.4 (13)

$$\gamma = kR^{\alpha} \quad \text{dB/km} \tag{C.4}$$

where R is the rain rate in mm/hr, and k and α have been recorded in Table C.2 for the index of refraction of water at 20^{o} C and spheroidal drops in the range of 1-150 mm/h.

The rain rate for different geographic areas around the world have been measured and regions have been identified by their rain rate climate. These eight regions are shown in Figures C.4 and C.5 as well as the rain rates exceeded as a percentage of the year. Table C.3 provides a more detailed chart for the rain rates exceeded in regions A to H as a percentage of the year.

Frequency (GHz)	k_H	k_V	α_H	α_V
1	0·0000387	0·0000352	0·912	0·880
2	0·000154	0·000138	0·963	0·923
4	0·000650	0·000591	1·121	1·075
6	0·00175	0·00155	1·308	1·265
7	0·00301	0·00265	1·332	1·312
8	0·00454	0·00395	1·327	1·310
10	0·0101	0·00887	1·276	1·264
12	0·0188	0·0168	1·217	1·200
15	0·0367	0·0335	1·154	1·128
20	0·0751	0·0691	1·099	1·065
25	0·124	0·113	1·061	1·030
30	0·187	0·167	1·021	1·000
35	0·263	0·233	0·979	0·963
40	0·350	0·310	0·939	0·929
45	0·442	0·393	0·903	0·897
50	0·536	0·479	0·873	0·868
60	0·707	0·642	0·826	0·824
70	0·851	0·784	0·793	0·793
80	0·975	0·906	0·769	0·769
90	1·06	0·999	0·753	0·754
100	1·12	1·06	0·743	0·744
120	1·18	1·13	0·731	0·732
150	1·31	1·27	0·710	0·711
200	1·45	1·42	0·689	0·690
300	1·36	1·35	0·688	0·689
400	1·32	1·31	0·683	0·684

Laws and Parsons dropsize distribution [7]
Gunn and Ginzer terminal velocities [8]
Index of refraction of water at 20°C after Ray [9]
Values of k_H, k_V, α_H, and α_V for spheroidal drops [10, 11] in the range 1–150 mm/h

Table C.2 Regression Coefficients for Estimating the Attenuation Coefficients(14)

Percent of Year	RAIN CLIMATE REGION												Minutes per Year	Hours per Year
	A	B_1	B	B_2	C	D_1	$D=D_2$	D_3	E	F	G	H		
0.001	28.5	45	57.5	70	78	90	108	126	165	66	185	253	5.26	0.09
0.002	21	34	44	54	62	72	89	106	144	51	157	220.5	10.5	0.18
0.005	13.5	22	28.5	35	41	50	64.5	80.5	118	34	120.5	178	26.3	0.44
0.01	10.0	15.5	19.5	23.5	28	35.5	49	63	98	23	94	147	52.6	0.88
0.02	7.0	11.0	13.5	16	18	24	35	48	78	15	72	119	105	1.75
0.05	4.0	6.4	8.0	9.5	11	14.5	22	32	52	8.3	47	86.5	263	4.38
0.1	2.5	4.2	5.2	6.1	7.2	9.8	14.5	22	35	5.2	32	64	526	8.77
0.2	1.5	2.8	3.4	4.0	4.8	6.4	9.5	14.5	21	3.1	21.8	43.5	1052	17.5
0.5	0.7	1.5	1.9	2.3	2.7	3.6	5.2	7.8	10.6	1.4	12.2	22.5	2630	43.8
1.0	0.4	1.0	1.3	1.5	1.8	2.2	3.0	4.7	6.0	0.7	8.0	12.0	5260	87.7
2.0	0.1	0.5	0.7	0.8	1.1	1.2	1.5	1.9	2.9	0.2	5.0	5.2	10520	175
5.0	0.0	0.2	0.3	0.3	0.5	0.0	0.0	0.0	0.5	0.0	1.8	1.2	26298	438

Table C.3 Rain Rates Exceeded as a Function of Percentage of Year, for Regions A to H(13)

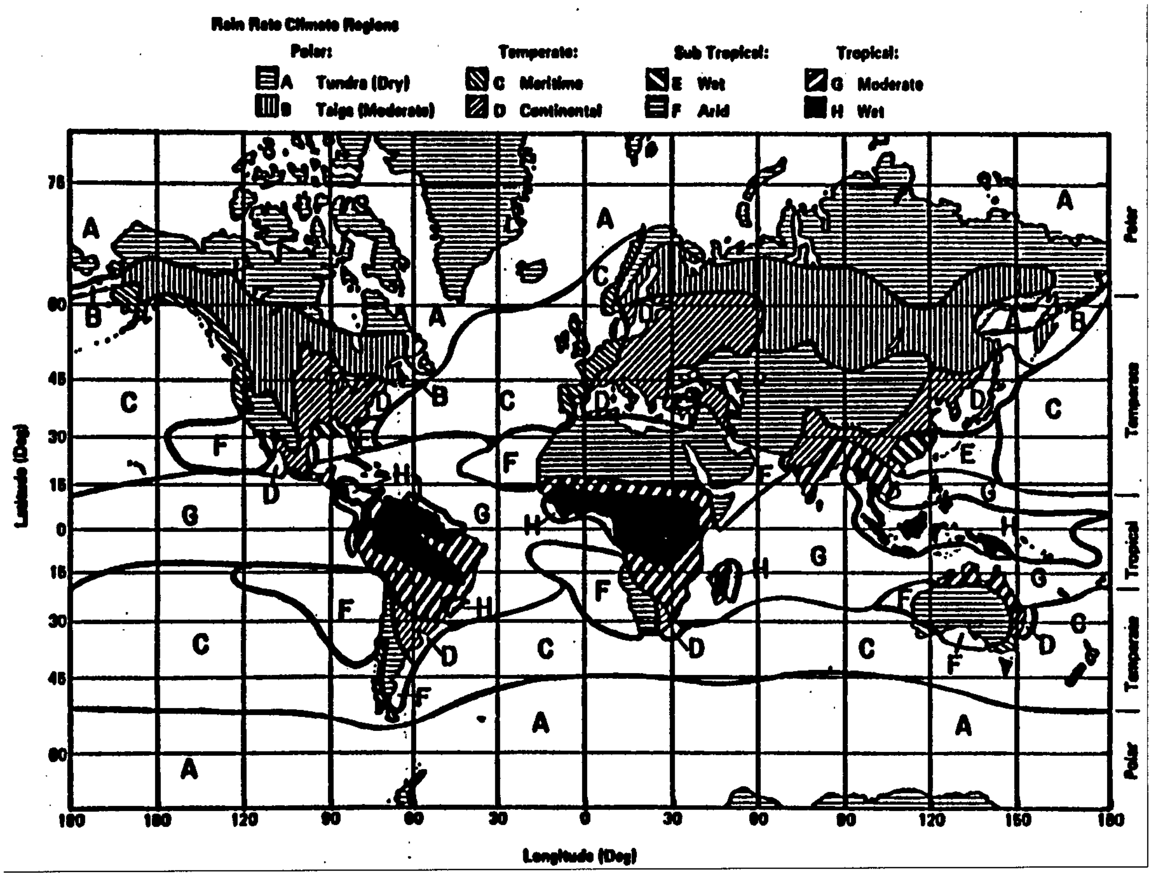


Figure C.4 Global Rain Rate Regions(13)

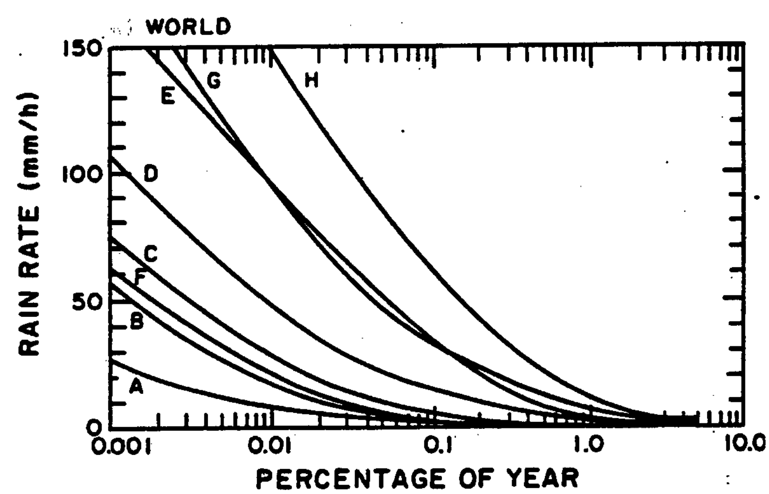


Figure C.5 Rain Rates as a Function of Percent of Year Exceeded(13)

The attenuation over the path is a function of the path length traveling through the rain. For satellite paths, it is necessary to measure the path length as a function of the height of the rain storm. For latitudes below 36^o, use a rain height of 4 km. For latitudes of 36^o and above, use the Equation(13)

$$H = 4.0 - 0.75(\Theta - 36^o) \tag{C.5}$$

where Θ is the latitude in degrees.

Then by using trigonometry, the path length can be found as a function of the height and elevation angle, θ (13)

$$L = \begin{cases} \frac{H}{\sin\theta}, & \theta \geq 10^o \\ \frac{2H}{(\sin^2\theta + \frac{2H}{Kr_o})^{1/2} + \sin\theta}, & \theta < 10^o \end{cases} \tag{C.6}$$

where θ is the elevation angle and Kr_o is the effective earth radius.

Therefore we can solve for the attenuation across the length of the path affected by the rain using (13)

$$A = \gamma L r_p \tag{C.7}$$

where r_p is a factor to account for the variability of the rain rate along the path (known as an empirical path reduction factor). We can first solve the attenuation problem for a probability of 0.01, for which r_p is given by

$$r_p = \frac{1}{1 + 0.045(L\cos\theta)} \tag{C.8}$$

Then we can determine the attenuation, A_p, equaled or exceeded with a probability other than 0.01 using (13)

$$A_p = A_{0.01} 0.12 p^{-(0.546+0.043 \log p)} \tag{C.9}$$

The degradation of the signal to noise ratio can be found using the same Equation as was used for clouds in the previous section. The effect of atmospheric absorption from rain can be large, as Figure C.6 demonstrates (SNR is shown as C/X in this figure).

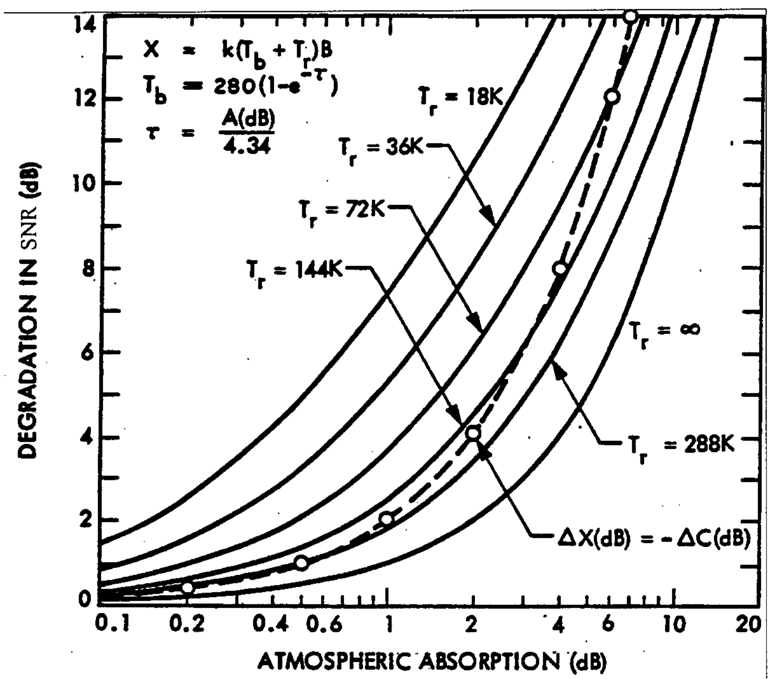


Figure C.6 Degradation in Signal-to-Noise Ratio (C/X) versus Atmospheric Absorption, for Various Values of T(14)

C.9 Sand and Dust Storms (Haboobs)

Haboobs are storms of particulates which act on radio waves with the same mechanisms as rain and clouds. A haboob is a unique storm primarily affecting equatorial regions in the rain rate climate regions F, G, and H (refer back to Figure C.4). Strong gusts from the cold down draft within the thunderstorm produce the dust storm. The air is usually fairly humid, leading to a significant uptake of water by the dust which will increase the propagation effects (Figure C.7).

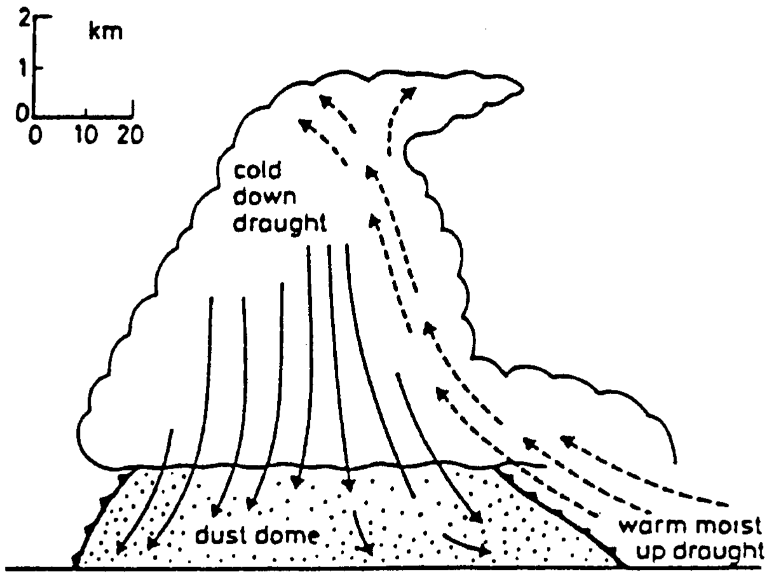


Figure C.7 Cross-Section through a Thunderstorm that is Producing a Haboob(14)

To find the attenuation as a result of a haboob, a model of the dust cylinder must be created to calculate the amount of dust in the path (Figure C.8). The dust is assumed to be of uniform density horizontally with all of the significant dust contained within a symmetrical right cylinder of diameter 10 km. The dust density falls off with height, following a power law decay, the index of which is determined by the initial visibility at a height of 2m.

The attenuation can be found using the same Equations as rain for the path length and replacing γ by (13)

$$\gamma = \frac{189}{V}\frac{a}{\lambda}\left[\frac{3K_i}{(K_r+2)^2+K_i^2}\right] \tag{C.10}$$

where V is the visibility in km, a is the average particle radius in m, λ_m is the wavelength in m, and $\epsilon_r = K_r - jK_i$ is the relative dielectric constant of the dust.

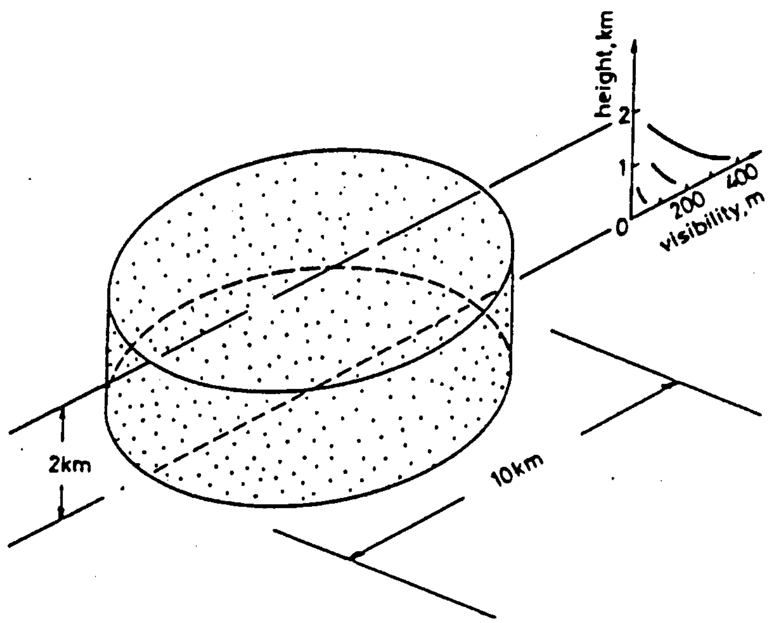


Figure C.8 Schematic of the Dust Cell Model(14)

Assuming a visibility of 10m and all dust particles are spherical and uniformly distributed along the path, the attenuation in dB/km is shown in Figure C.9. The lower line (with dots) is for a 0.3% mixture of water and dust (g H_2O/g soil). The upper line (with x's) is for a higher concentration of water (10% (g H_2O/g soil)).

C.10 Determining Received Power and Link Margin

The power received at the communication receiver is a function of the transmitted power, gain of the transmit antenna, path loss and antennuation, and losses from the antenna, cable, connector and receiver. The received power is expressed as C.11 (32, 38):

$$P_r = \frac{(EIRP)G_r}{L_s L_o} \tag{C.11}$$

where:

P_r is the received signal power in Watts;

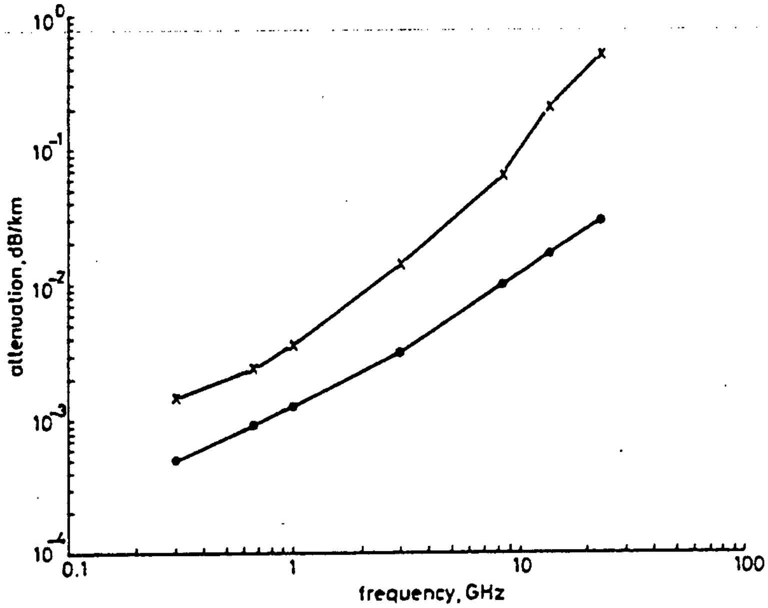


Figure C.9 Attenuation as a Function of Frequency for Falling Dust Particles(14)

EIRP is the Effective Isotropic Radiated Power in Watts and can be expressed as C.12:

$$EIRP = P_t G_t \tag{C.12}$$

where P_t is the transmitted power in Watts and G_t is the gain of the transmit antenna;

G_r is the gain of the receive antenna;

L_s is the path loss expressed as C.13:

$$L_s = \left(\frac{4\pi}{(c/f)}\right)^2 \frac{K_s}{R_s^{l_s}} \tag{C.13}$$

where:

c is the speed of light ($\simeq 3 \times 10^8$ m/s),

f is the frequency in Hz,

K_s is the constant of attenuation over the radiowave path,

R_s is the communication path distance in meters, and

l_s is the antennuation exponent on the communication path ($l_s = 2$ for free space communications such as air to air and air to ground, and $l_s = 4$ for ground to ground communications (38)); and L_o are losses from the antenna, cable, connector and receiver.

The link margin (or safety margin) is an expression which relates the received power and the receiver thermal noise power spectral density with a required bit energy per noise power spectral density to yield a specified error probability. The received signal power (no noise) is the product of the received energy per bit and the information rate C.14:

$$P_r = E_b R_b \tag{C.14}$$

where E_b is the received energy per bit and R_b is the received information rate or roughly the baseband bandwidth.

Therefore, the link margin, (M), can be expressed as C.15:

$$M = \left(\frac{E_b}{N_o}\right)_{recvd} (\text{dB}) - \left(\frac{E_b}{N_o}\right)_{reqd} (\text{dB}) \tag{C.15}$$

where N_o is the noise power spectral density to the Predetection output of the receiver and is expressed as C.16 (32):

$$N_o = G\kappa T_s^{\circ} W \tag{C.16}$$

where:

G is the gain of the receiver,

κ is Boltzmann's constant (1.38 x 10^{-23} W-K/Hz),

T_s° is the system effective temperature, and

W is the bandwidth of the received signal in Hertz.

The system effective temperature is expresses as:

$$T_s^\circ = T_A^\circ + (10^{F_L F_R/10} - 1)290\text{K} \tag{C.17}$$

where T_A° is the antenna temperature and F_L and F_R are the noise figure of the antenna cable and coupler and the noise figure of the receiver, respectively. Figure C.10 represents a simplified schematic of a receiving system (32).

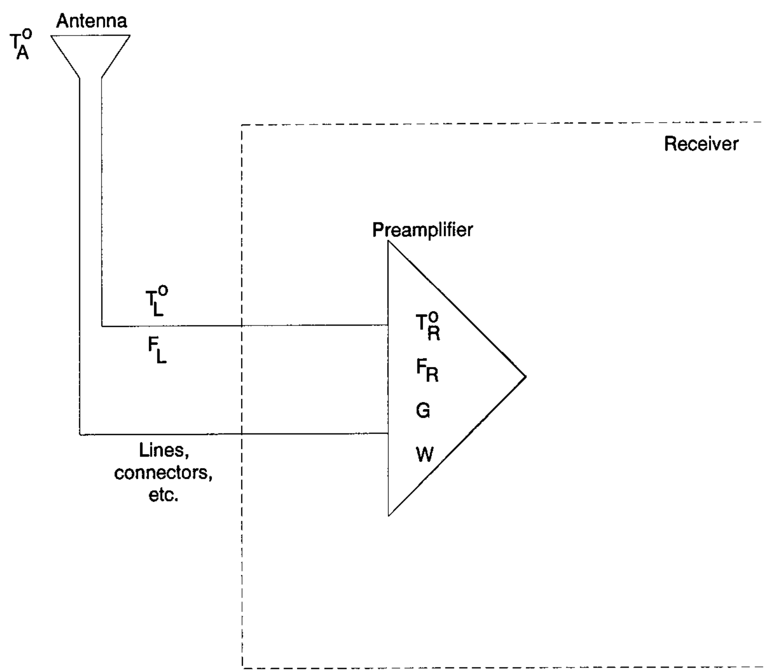


Figure C.10 Major Noise Contributors of a Receiving System(32)

C.11 Need for Automation

Current methods of manual network design and analysis use acetate overlays on paper maps, self-adhesive dots for unit location, push pins for nodes, and tape, grease pencil or colored string for transmission routes. Link performance is driven solely on basic specifications for line of site distance, terrain obstacles, and average power available at the transmitter. Network topology is often changed several times during the planning stage, and then reactive during mission operation. For example, a site which has been selected to maximize subscriber access may be shifted to satisfy terrain constraints, and then this site may be adjusted again because of logistics considerations. By this time, the subscriber access has been degraded and must be revisited. This problem will be many times more difficult with the implementation of the Army's Integrated System Control (ISYSCON) which will require the coordination of all communications assets including Enhanced Position Locating and Ranging System (EPLRS), Joint Tactical Information Distribution System (JTIDS), Satellite Communications System (SATCOM), Combat Net Radio (CNR) and other communications assets at all levels of command. Additionally, as the operation develops, the network becomes mobile on the ground(6).

Computer assistance in the network design process will increase the planner's effectiveness and reduce the major part of the work load. It will allow planners to develop plans in the reduced time necessary to support the mobility, flexibility, and robustness required of today's communications equipment in the rapidly changing tactical environment(6). Additionally, the plan that is developed will be available for direct transfer to others and the communications network can be simulated in advance (simulated war game) based on operational plans as they develop. Applying realism in the simulation avoids mistakes commonly made during network planning when using basic specifications in the manual method and gives decision makers more realistic expectations of the communications backbone.

Despite the extensive research during the past decade on network performance assessment and optimization, the need to develop automated tools to assess system performance in more than one network layer, responsive to the dynamic battlefield environment still exists.

C.12 Perishability of Information and Situational Age

Certain types of information is perishable and loses usefulness if not updated frequently. An example of this is situational age. Consider a sensor that takes a snapshot in time of an enemy tank. The tank is moving, so the sensor reports the tank's direction and speed. However, immediately after this information is sent, the degree of uncertainty of the tanks position increases with time (Figure C.11). The tank may have changed direction or speed.

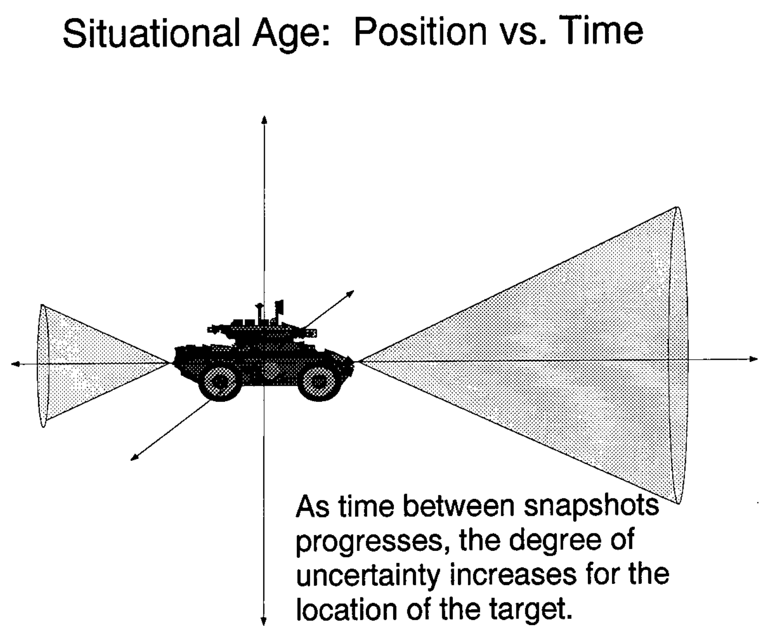


Figure C.11 Situational Age, an Example of the Perishability of Information.

Unless the sensor updates this information, it becomes useless as the degree of uncertainty grows. In order for the information to be useful, it must be received and acted upon before the tank could leave the kill radius of the weapon system chosen to destroy it, or the sensor must update the decision maker and/or weapon system to ensure the tank is within the kill radius of the weapon when it arrives. Information perishability is an important metric of C4I systems and relies on an accurate model of the radiowave propagation path. A haboob which hides the target from the sensor is important to include in the model and may result in a different result during a wargame simulation. This case is also true if the target is initiated by an observer or sensor;

however, information needed to update the weapon system and achieve a "kill" upon arrival is lost as a result of environmental conditions between the observer/sensor and weapon system.

C.13 Modeling Temporal Events

Environmental conditions discussed in this appendix are time dependent. IDEF2 provides a model by which these events can be captured and described(15). This model can be included in the architectural design process recommended by this thesis, to capture and include in C4I simulations, the radiowave propagation effects discussed in this appendix.

Appendix D. IDEF4 Overview

The following overview of the IDEF4 Object-Oriented Design Method by Knowledge Base Systems, Inc. is provided, with permission, to encourage further standardization of the simulation object development process. Simulation developers, as computer programmers, need the documentation produced by IDEF4 to identify existing classes, instances, and object parameters and behavior. The IDEF4 products would also be used to validate and approve objects for storage in an ODBMS in the distributed simulation network.

IDEF4 Object-Oriented Design Method

The intuitive nature of object-oriented programming makes it easier to produce code. Unfortunately, the ease with which software is produced also makes it easier to create software of poor design, resulting in systems lacking re-usability, modularity, and maintainability. The IDEF4 method is designed to assist in the correct application of this technology.

With over thirty object-oriented design methods in existence today, why should we chose IDEF4? Most importantly, IDEF4 views object-oriented design as part of a larger system development framework, rather than an object-oriented analysis and design method that is ambiguous. IDEF4 stresses the object-oriented design process over the graphical syntax, using the graphical syntax and diagrams as aids to focus and communicate important design issues. IDEF4 is significantly different from other object design methods, primarily in its support of "least commitment" strategies and its support for assessing the design impact of the interaction between class inheritance, object composition, functional decomposition, and polymorphism.

IDEF4 Concepts

IDEF4 divides the object-oriented design activity into discrete, manageable chunks. Each subactivity is supported by a graphical syntax that highlights the design decisions that must be made and their impact on other perspectives of the design. No single diagram shows all the information contained in the IDEF4 design model, thus limiting confusion and allowing rapid inspection of the desired information. Carefully designed overlap among diagram types serves to ensure compatibility between the different submodels. The IDEF4 method allows the designer to easily make trade-offs between class composition, class inheritance, functional decomposition, and polymorphism in a design. IDEF4 is more than a graphical syntax--the graphical syntax provides a convenient framework for navigating an evolving object-oriented design that is ultimately specified on class invariant data sheets and method set contracts.

Figure 1 shows the basic organization of an IDEF4 model. Conceptually, an IDEF4 model consists of two submodels, the class submodel and the method submodel. The two submodels are linked through a dispatch mapping. These two structures capture all the information represented in a design model. Due to the size of the class and method submodels, the IDEF4 object designer never sees these structures in their entirety. Instead, the designer makes use of the collection of smaller diagrams and data sheets that effectively capture the information represented in the class and method submodels.

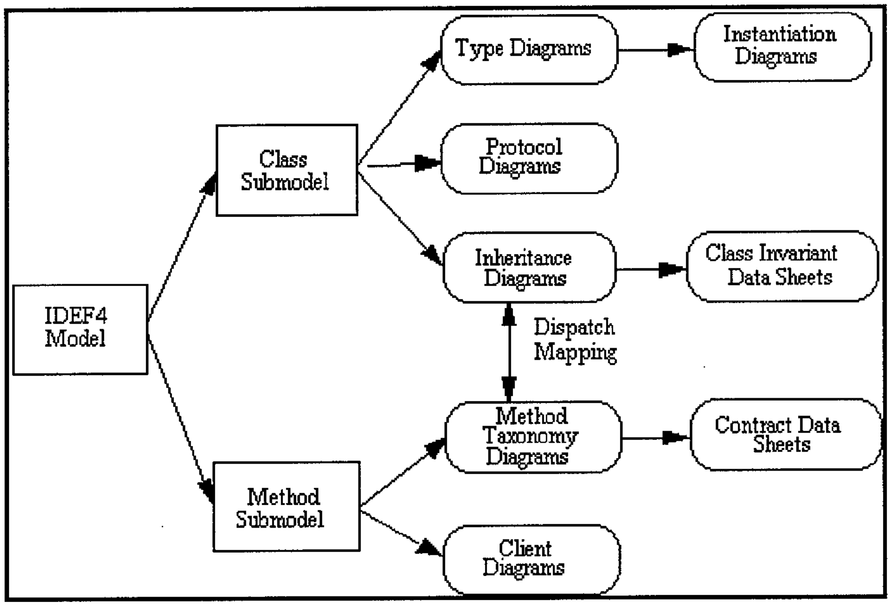


Figure 1. Organization of the IDEF4 Model

The class submodel is composed of the following diagram types: 1) Inheritance diagrams that specify class inheritance relations; 2) Type diagrams that specify class composition; 3) Protocol diagrams that specify method invocation protocols; and 4) Instantiation diagrams that describe object instantiation scenarios that assist the designer in validating the design.

The method submodel is composed of the following two diagram types: 1) Method taxonomy diagrams which classify method types by behavior similarity and 2) Client diagrams which illustrate clients and suppliers of methods, to specify functional decomposition.

IDEF4 Class Submodel

The class submodel consists of inheritance diagrams, type diagrams, instantiation diagrams, protocol diagrams, and class-invariant data sheets. The class submodel shows class inheritance and class composition structure.

IDEF4 Inheritance Diagrams

Inheritance diagrams specify the inheritance relations among classes. For example, Figure 2 shows the class Filled-rectangle inheriting structure and behavior directly from the classes Rectangle and Filled-object and indirectly from the class Object.

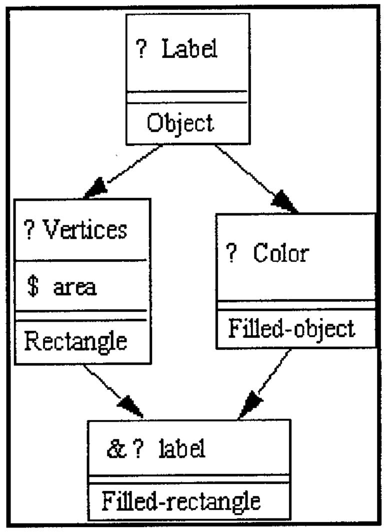


Figure 2. Inheritance Diagram

IDEF4 Instantiation Diagrams

Instantiation diagrams are associated with type diagrams in the class submodel. The instantiation diagrams describe the anticipated situations of composite links between instantiated objects that are used for validating the design.

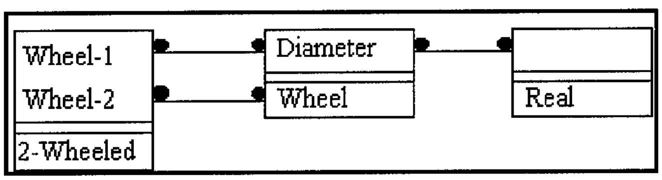


Figure 3. Type Diagram

IDEF4 Protocol Diagrams

Protocol diagrams specify the class argument types for method invocation. Figure 4 illustrates a protocol diagram for the Fill-closed-object: Polygon routine-class pair. From the diagram, the reader can to tell that Fill-closed-object will accept an instance of the class Polygon as its primary (self) argument and an instance of the class Color as a secondary argument, and will return an instance of a Polygon.

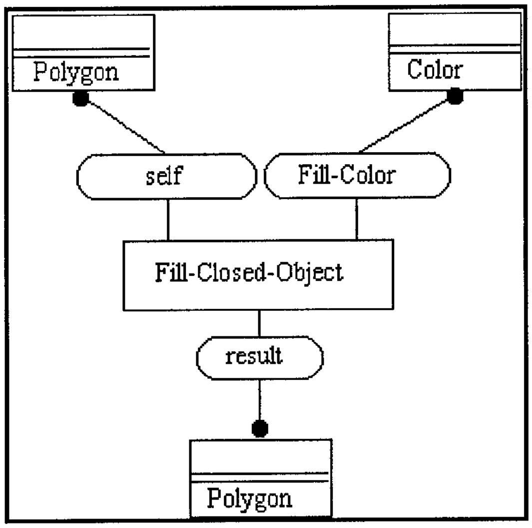


Figure 4. Protocol Diagram

The IDEF4 Method Submodel

The method submodel consists of method taxonomy diagrams, client diagrams, and data sheets.

Method Taxonomy Diagrams

Method taxonomy diagrams classify method types by behavioral similarity. A method taxonomy diagram classifies a specific system behavior type according to the constraints placed on the method sets represented in the taxonomy. The arrows indicate additional constraints placed on the method sets. Figure 5 shows a Print method taxonomy diagram. The method sets in the taxonomy are grouped according to the additional contracts placed on the methods in each set. In the example, the first method set, Print, has a contract that states that the object must be printable. The Print-text method set contract would have constraints such as "the object to be printed must be text."

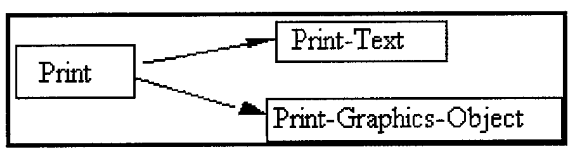


Figure 5. Method Taxonomy Diagram

Client Diagrams

Client diagrams illustrate clients and suppliers of routine-class pairs. Double-headed arrows point from the routine that is called to the calling routine. Figure 6 shows a client diagram where the Redisplay routine attached to the class Redisplayable-object calls the Erase routine of the Erasable-object class and the Draw routine on the Drawable-object class.

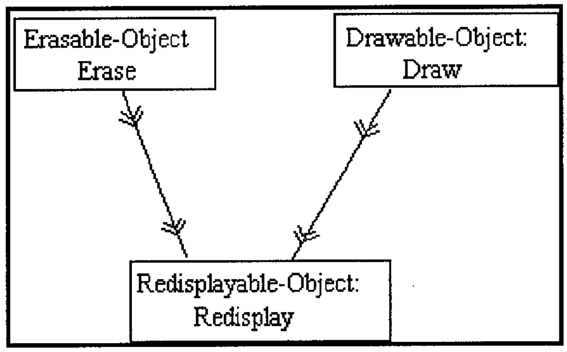


Figure 6. Client Diagram

Data Sheets

Because IDEF4 is not just a graphical language, additional information about the inheritance diagrams, method taxonomy diagrams and type diagrams is maintained in detailed data sheets.

Class Invariant Data Sheets

Class-invariant data sheets are associated with inheritance diagrams and specify constraints that apply to every instance of a particular class of objects. There is one class-invariant data sheet for each class. The constraint, "Every triangle has three sides," is a class-invariant constraint on the class Triangle.

Contract Data Sheets

Contract data sheets are associated with the method sets in method taxonomy diagrams and specify contracts that the implemented methods in a method set must satisfy. There is one contract data sheet for each method set. A method set called Pop, for popping values off a stack, would have a contract that specified that it would not attempt to "pop" a the stack if the stack was empty.

Summary

The IDEF4 method, developed under sponsorship of the U.S. Air Force Armstrong Laboratory, is designed to assist in the correct application of object-oriented technology. IDEF4 was developed by professional object-oriented designers and programmers. The most important reason for choosing the IDEF4 method is that it views object-oriented design as part of a larger system development framework, rather than an object-oriented analysis and design method that is everything to everyone. It stresses the object-oriented design procedure over the graphical syntax, using the graphical syntax and diagrams as aids to focus on and communicate important design issues.

KBSI has developed an automated Object-Oriented Design tool, SmartClass, to support the IDEF4 method.

This page was last updated on December 13, 1996.

Bibliography

1. "The COMPASS Story." Science Applications International Corporation (SAIC), August 1995. `http://compass.saic.com/files/compass-story.html`.
2. "Network Computing Architecture; An Oracle White Paper," September 1996. `http://www.oracle.com/nca/html/nca_wp.html`.
3. Alexander H. Levis, Didier M. Perdu, Robert F. Phelps. *Architecting Information Systems* (gmu/c31-165-r Edition). Center of Excellence in Command, Control, Communications and Intelligence, George Mason University, Fairfax, VA 22030-4444, December 1995.
4. Alta Group of Cadence Design Systems, Inc., 919 East Hillsdale Blvd, Foster City, CA 94404. *BONeS DESIGNER Modeling Guide* (Des8005 Edition), December 1994.
5. Army Digitization Office, Washington, DC. *Army Digitization Master Plan*, January 1995.
6. Atwater, Frederick M. "Application of Neural Network Computing to Tactical Communications Planning," *1991 IEEE Military Communications Conference, MILCOM 91*, 4.7.1–4.7.5 (1991).
7. Barkakati, Naba. *UNIX Webmaster Bible*. IDG Books Worldwide, 1996.
8. Bruce, Thomas A. *Designing Quality Databases with IDEF1X Information Models*. 353 West 12th St., New York, NY 10014: Dorset House Publishing, 1992.
9. Cattell, R.G.G., editor. *Object Database Standard: ODMG-93*. 2929 Campus Drive, Suite 260, San Mateo, CA 94403: Morgan Kaufmann Publishers, 1994.
10. Center for Architecture, Defense Information Systems Agency. *Department of Defense Technical Architecture Framework for Information Management (TAFIM), Version 2.0: Technical Reference Model and Standards Profile Summary.*
11. Date, C. J. *An Introduction to Database Systems*. Addison-Wesley Publishing Company, August 1995.
12. Department of the Army, Washington, DC 20310-5200. *Army Enterprise Implementation Plan*, 1994.
13. Flock, Warren L. *Propagation Effects on Satellite Systems at Frequencies Below 10 GHz: A Handbook for Satellite Systems Design* (nasa rp-1108(02) Edition). Scientific and Technical Information Division, National Aeronautics and Space Administration, Washington, DC 20546, December 1987.
14. Gerald C. Gerace, MAJ, "Radiowave Propagation Theory and Models." Lecture handouts, September 1995.
15. Hanrahan, Robert P. "The IDEF Process Modeling Methodology," *Cross Talk*, 19–23 (June 1995). Robert P. Hanrahan is a Software Technology Support Center techology domain lead for Process, Software Engineering Environment and Requirements Analysis and Design technologies.
16. Jerry Banks, John S. Carson II and Barry L. Nelson. *Discrete-Event System Simulation*. Upper Saddle River, NJ 07458: Prentice Hall, 1996.
17. Joint Interoperability and Engineering Organization, Defense Information Systems Agency. *JIEO Report 8125, Joint Task Force Tactical Communications Architecture*, March 1995.
18. Ken Arnold, James Gosling. *The Java Programming Language*. One Jacob Way, Reading, MA 01867: Addison-Wesley Publishing Company, Inc., May 1996.
19. Khoshafian, Setrag. *Object-Oriented Databases*. John Wiley and Sons, Inc., 1993.

20. Kim, Won, editor. *Modern Database Systems*. ACM Press, New York, New York: Addison-Wesley Publishing Company, January 1995.

21. Korth, Henry F. and Abraham Silberschatz. *Database System Concepts*. McGraw-Hill, Inc, 1986.

22. Lewis, P. A. W. and E. J. Orav. *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*. Pacific Grove, CA: Wadsworth & Brooks/Cole Advance Books & Software, 1989.

23. McFarland, Dave, "A Technical Overview of the J-MASS Architecture." J-MASS Program Office, June 1995. `http://www.jmass.wpafb.af.mil/overview.html`.

24. Muirragui, Lieutenant Colonel Richard A. *Communications, The Forgotten Element of C3I: A Study of WarGaming, Modeling, and Simulations*. Individual Study Project DTIC AD-A239 302, Carlisle Barracks, PA 17013: U.S. Army War College, June 1991.

25. National Institute of Standards and Technology, National Information Service, Dept. of Commerce, Springfield VA 22161. *Integrated Definition for Information Modeling (IDEF1X)*, December 1993.

26. National Institute of Standards and Technology, National Technical Information Service. Dept. of Commerce, Springfield VA 22161. *Integration Definition for Function Modeling (IDEF0)*, December 1993.

27. Object Management Group (OMG), OMG Headquarters, 492 Old Connecticut Path, Framingham, MA 01701. *The Common Object Request Broker: Architecture and Specification* (2.0 Edition), July 1996.

28. Office, Defense Modeling & Simulation, "Annotated Briefing on the DoD High Level Architecture for Simulation," September 1996. `http://www.dmso.mil`.

29. Office of the Director for Information Systems for Command, Control, Communications Computers and Intelligence. "Systems Architecture for Battlefield Digitization and Assorted Slides." Provided by MAJ Renner, ODISC4, July 1995.

30. Paul F. Reynolds, Jr., "DISorientation." Department of Computer Science, University of Virginia, 1994. `http://www.mystech.com/~smithr/elecsim94/disori/disori.txt`.

31. Pratt, David R. and Matthew A. Johnson. "Constructive and Virtual Model Linkage," *Proceedings of the 1995 Winter Simlation Comference*, 1222–1229 (December 1995).

32. Sklar, Bernard. *Digital Communications*. Englewood Cliffs, NJ 07632: Prentice Hall, 1988.

33. Sollins, Karen R. and Jeffrey R. Van Dyke. *Linking in a Global Information Architecture*. MIT, 1996. `http://www.lcs.mit.edu/web_project/anaweb/links.html` and `http://www.w3.org/pub/Conferences/WWW4/Papers/193/`.

34. Tarek N. Saadawi, Mostafa H. Ammar and Ahmed El Hakeem. *Fundamentals of Telecommunication Networks*. NY, NY: John Wiley & Sons, Inc., 1994.

35. Teter, LTC William A. *Technical Architecture, Version 4.0*. Department of the Army.

36. Tomlinson, Bob, "CORBA and OMG Information Resources." Advance Computing Laboratory, Los Alamos National Laboratory, January 1997. `http://www.acl.lanl.gov/CORBA/`.

37. Wagner, Richard. *JavaScript Unleashed* (First Edition). 201 W. 103rd St., Indianapolis, IN 46290: Sams.net Publishing, 1996.

38. William J. O'Brien, Jeffrey Seibert, Brian Vogel. *On Achieving Network LPI for Spread Spectrum Communications*. Final DTIC AD-A163 985, 11225 Waples Mill Rd., Fairfax, VA 22030: E-Systems, Inc., Melpar Division, Special Processing Group, October 1985.

39. Ye, Zhou and Ankur Sharma, "Java - CORBA Binding." Oracle Corporation, June 1996. `http://www.w3.org/pub/WWW/OOP/9606_Workshop/submissions/64.html`.

Vita

Captain Timothy R. Schmoyer is a 1987 graduate of Worcester Polytechnic Institute and the Reserve Officer Training Corps. He received a Bachelor of Science degree in Electrical Engineering and Regular Army commission. He started his career as an ROTC Recruiter at Fitchburg State College, MA.

His engineering assignments include Test Engineer at Electronic Proving Ground, Ft. Huachuca, AZ; Quality Assurance Engineer for Wideband Defense Communications Systems (Information Systems Engineering Command-Europe), Germany; and New Equipment Training and Development, USA John F. Kennedy Special Warfare Center and School, Ft. Bragg, NC.

His tactical assignments include Platoon Leader and Company Executive Officer with the 1st Signal Battalion, Germany; Company Commander, 112th Signal Battalion (Special Operations) (Airborne); and Communications-Electronics Officer, 3rd Battalion, 7th Special Forces Group (Airborne), Ft. Bragg, NC. Captain Schmoyer's military schooling includes: the Airborne Course, Signal Officer Basic Course, the Ranger Course, Signal Officer Advance Course, the Tactical Signal Staff Officer Course and the 3rd Special Forces Group (Airborne) Jumpmaster Course.

His awards and decorations include: the Parachutist Badge, Ranger Tab, Meritorious Service Medal (1 oak leaf cluster), Army Commendation Medal (2 oak leaf clusters), Army Achievement Medal (3 oak leaf clusters), National Defense Service Medal, Armed Forces Expeditionary Medal, Southwest Asia Service Medal (3 bronze stars), Humanitarian Service Medal, Army Service Ribbon, Overseas Service Ribbon, Kuwaiti Liberation Medal-Saudi Arabia, Kuwaiti Liberation Medal-Kuwait, Joint Meritorious Unit Award, Army Superior Unit Award and the German Army Proficiency Badge (Leistungsabzeichen) in Bronze.

He is married to the former Rachel L. Kardon and is currently assigned to the Air Force Institute of Technology for his Masters of Science in Electrical Engineering. They reside in Dayton, Ohio.

Permanent address: [redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	20 March 1997	Master of Science Thesis

4. TITLE AND SUBTITLE
Net-Centric Design and Analysis of Information Systems

5. FUNDING NUMBERS

6. AUTHOR(S)

Timothy R. Schmoyer, Captain, U.S. Army

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Departmnet of Electrical Engineering
2950 P. Street
Wright-Patterson AFB, OH 45433

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/97M-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of the Director of Information Systems for Command, Control, Communications and Computers
107 Pentagon (Army)
Washington DC 20310-0107

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

A

13. ABSTRACT *(Maximum 200 words)*

This thesis presents an unique methodology merging state of the art Internet and distributed database technology to support distributed simulations with programming language and platform independence. Standardized models of Command, Control, Communications, Computers, and Intelligence (C4I) systems using Integrated Definition (IDEF) models and executable simulation objects are placed in database repositories which can be accessed and implemented over a distributed simulation network using the Common Object Request Broker Architecture (CORBA). The CORBA distributed simulation network accesses heterogeneous distributed databases, performs distributed processes, and supports portability and reuse of simulation objects and interoperability across operating systems and programming languages.

14. SUBJECT TERMS
Simulation, Distributed, Object-Oriented, CORBA, C4I, IDEF, Communication, Database, Modeling, and Networks

15. NUMBER OF PAGES
150

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLAS	UNCLAS	UNCLAS	UNLIMITED