Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

KNOWLEDGE-BASED DESIGN OF LANs USING SYSTEM ENTITY STRUCTURE CONCEPTS

Jerzy W. Rozenblit, Süleyman Sevinç, Bernard P. Zeigler

Department of Electrical and Computer Engineering
The University of Arizona
Tucson, Arizona 85721

ABSTRACT

The paper illustrates an application of knowledge based system design concepts to design of local area networks. A system entity structure representation of a local area network (LAN) is developed. This representation unifies a variety of possibilities for LAN design architecture. In a design session, the LAN design domain is subsequently restricted by pruning the entity structure with respect to network design objectives. A LAN model is then synthesized using a production rule scheme. The benefit of the knowledge-based LAN design is briefly discussed.

1. INTRODUCTION

A Local Area Network (LAN) (Tanenbaum, 1981; Stallings, 1984) is a packet switching data communication network which is used to connect terminals, computers, printers and other auxiliary devices. Local Area Networks have proliferated considerably in the past few years due to the increasing needs for office automation (electronic mail, document processing and distribution), and for sharing expensive devices within a company. Information exchange and processing have become crucial issues for many corporations. Consequently, methods for LAN design and simulation are being developed to assist the users in making decisions about their communication needs.

Two major approaches to LAN design and performance evaluation are: analytic modelling, and discrete event simulation. In this paper we present a LAN design approach based on previously developed concepts for knowledge-based design incorporating modelling and simulation (Rozenblit and Zeigler, 1985; Rozenblit 1986). A brief summary of our design methodology is given in the ensuing section.

2. KNOWLEDGE-BASED SYSTEM DESIGN

A methodology for integrated, knowledge-based system design is being developed (Rozenblit, 1986). This methodology is intended to provide a theoretical basis for a uniform treatment of design process by providing concepts of structure and behavior, decomposition and hierarchy of specification. Appropriate schemes for representation of such concepts are being developed. These schemes constitute a basis for the construction of expert system design environments.

Our research employs multifacetted modelling and simulation concepts to unify system design activities and develop a methodology for systematic design model construction and evaluation. We consider the design process as a series of successive refinements comprising two types of activities (Fasang et. al., 1983; Gonauser and Sauer, 1983; Gonauser et. al., 1983; Rozenblit, 1986). The first type of "vertical" activity concerns the specification of design levels in a hierarchical manner. The design levels are successive refinements of the decomposition of the system under consideration. The first, and thus the most abstract level, is defined by the behavioral description of the system. Next levels are defined by decomposing the system into subsystems (modules, components), and applying decompositions to such subsystems until the resulting components are judged not to require further decomposition. At each level we also allow for classifying of components into different variants. This facilitates the representation of design alternatives.

The second type of activities is concerned with "horizontal" actions associated with design levels. Such actions include: requirements specification, system functional description, development of design models, experimentation and evaluation via simulation, and choice of design solutions.

Our methodology for supporting the design process bases itself on codifying appropriate decompositions, taxonomic, and coupling relationships. In other words, we seek to model the knowledge about the design domain by finding pertinent decompositions of the domain, the possible variants that fit within these decompositions, and the constraints that restrict the ways in which components identified in decompositions can be coupled together. This constitutes the declarative knowledge base. Beyond this, we provide the procedural knowledge base in the form of production rules which can be used to manipulate the elements in the design domain (Rozenblit, 1986; Rozenblit and Zeigler, 1985, 1986).

A formal object that meets the requirements stipulated above is the system entity structure (Zeigler, 1984). A system entity structure is a tree-like graph encomposing the system decompositions and boundaries. This structure is a basic means of specifying a family of possible design configurations. The entities represent system components. Alternative design structures are specified via aspects and specializations. An aspect is a mode of decomposition for an entity; a specialization is a mode of classification for it.

The system entity structure generates a set of substructures from which design models can be constructed. A process called pruning is needed to select those substructures which match the design requirements and objectives.

We define an object called a generic experimental frame to express formally design objectives. A generic frame consists of variable types that express performance indices associated with a given design objective. The pruning algorithms developed in (Rozenblit, 1986) select from the system entity structure entities whose attributes match the variable types given in a generic frame in which the pruning proceeds. The selected entities are used to construct design models.

The pruning algorithms generate all design model structures that correspond to the behavioral aspects of the design objectives. In order to deal with the structural constraints imposed on the system being designed we augment the design model development process with a process termed synthesis rule specification (Rozenblit, 1986, Rozenblit and Zeigler, 1986).

The constraints imposed by the design requirements are classified into two basic categories: convertible to active form, i.e., they can be converted into actions intended to satisfy them, and passive. The passive constraints do not guide or motivate any action. They do require satisfaction.

We conceive of the synthesis problem as a search through the set of all pruned structures. These are candidates for the solution to the problem. We can assume that for each active constraint we have a means of generating such candidates to test against the constraint. The passive constraints have no corresponding operators, and thus, we can only test for their satisfaction.

Based on these concepts, a canonical production rule scheme for a hierarchical design model synthesis has been developed (Rozenblit and Zeigler 1985, 1986).

We now summarize the basic activities of our design framework:

1.) A family of possible design configurations of the system being designed is organized by the system entity structure.

2.) The objectives and design requirements induce appropriate generic experimental frames.

3.) The system entity structure is pruned with respect to generic frames. This results in a family of design model structures conforming to design objectives.

4.) The pruned substructures serve as skeletons for generating production rules for synthesis of design models.

5.) Resulting design models are evaluated via simulation studies.

We now proceed to describe in detail how the above presented framework can be applied to the LAN design problem.

3. LAW SYSTEM ENTITY STRUCTURE REPRESENTATION

There are four types of knowledge needed to construct a design model of a LAN architecture:

1. Available physical resources: this type of knowledge must include quantitative and qualitative information about the physical resources of the LAN architecture.

2. Functions: functions are the routines that implement the communication tasks such as flow control, error control, etc.

3. Mapping knowledge: in an actual device, there are a number of processors. Mapping knowledge describes how functions are mapped onto the processors that execute them.

4. Coupling knowledge: in a LAN architecture data are processed by several components of the LAN and are sent from one component to another. In order to model this process in a modular way output ports of the LAN components must be appropriately coupled with corresponding input ports. Coupling knowledge describes the input-output port mapping for the component models of a LAN.

To represent the above categories of knowledge we first generate a system entity structure for a class of local area networks. Due to the diversity of LAN architectures our system entity structure for a local area network design is not built around a specific network architecture. Rather than confining ourselves to a particular scheme, we generate a structure that outlines a family of possible network configurations. A detailed description of the LAN entity structure is presented in the following section.

3.1 LAN Entity structure.

As illustrated in Figure 1, a local Area Network (LAN) is decomposed into two entities: COMMUNICATION SYSTEM and a multiple entity USERS. The multiple entity USERS represents a set of entities, each called USER. The term user denotes computers, terminals, mass storage devices, printers, plotters, monitoring equipment, etc.. The entity USER is further decomposed into its INTERNAL MODEL, QUEUES, TABLES, and PROTOCOLS. The entity structure for COMMUNICATION SYSTEM is described in detail in the following subsection.

Communication System Subentities

COMMUNICATION SYSTEM is decomposed into TRANSMISSION SYSTEM and a multiple entity INTERFACE DEVICES. As shown in Figure 1 a transmission system can have the following special types of topologies: BUS, TREE, and RING. Transmission techniques are specialized into BASEBAND (with a specialization SIGNALLING TECHNIQUES), and BROADBAND (with a specialization MODULATION TECHNIQUE). The logical decomposition of the entity TRANSMISSION SYSTEM yields ELECTRONIC COMPONENTS such as filters, amplifiers, and the communication media on which the signals are transmitted. Regardless of whether the physical communication media are fiber, coax, or twisted pair, they logically have a number of CHANNELS. Channels may be BIDIRECTIONAL OR UNIDIRECTIONAL, depending on the organization of the channels' electronic components. Channels can also be specialized into POINT-TO-POINT or BROADCAST types. Electronic components on the channels can be modelled as delay elements with appropriate error characteristics.

Interface Device Entity Structure

The multiple entity INTERFACE DEVICES represents the interface media between USERS and CHANNELS as well as interfaces between two or more channels. The devices coupling USERS and CHANNELS are called INTERFACE UNITS. The interfaces between channels are termed BRIDGES (Hawe et. al., 1985). Both BRIDGES and INTERFACE UNITS contain the same type of entities. Regardless of its type each interface unit physically contains a number of PROCESSORS and MEMORIES. Within an interface unit there are multiple entities: QUEUES, TABLES, and PROTOCOLS. Entities QUEUES and TABLES denote data structures. They are used for storage purposes.

PROTOCOLS are the actions that make use of the physical resources of the network to provide communication between the users of the network. They have ports that connect them externally to one another and the users. These ports may have speed limitations and are intended to represent physical device interfaces such as the standard RS-232 interface.

Due to space restrictions we have limited the LAN entity structure to that of Figure 1. The reader is referred to (Sevinc, 1986) for a complete example of the LAN representation.

3.2 Pruning the LAN Entity Structure for Network Model Construction

The LAN entity structure of Figure 1 specifies a family of possible network design configurations. The entities represent network components while aspects and specializations allow us to synthesize various alternative architectures. At this point we should apply the pruning algorithms in order to restrict the design domain to structures that conform to network design objectives. Recall from Section 1 that objectives are expressed in terms of generic experimental frames.

In outline, the pruning algorithm is based on the depth first tree traversal. Every entity in each aspect is searched for occurrences of input/output variable types that are present in the generic frame in which the pruning proceeds. The entities whose variable types correspond to those present in the generic frame are used to construct the model composition tree (Zeigler, 1984) as the search progresses. The coupling constraints associated with aspects selected by pruning are mapped onto the composition tree. The algorithm calls itself recursively for each entity being searched.

We now proceed to illustrate the pruning process by using the LAN design entity structure of Figure 1. For the sake of brevity let us focus on design of a subsystem of a LAN, namely the transmission system. Our objective is to design the transmission subsystem that attains a certain data transfer rate. Thus, the first step in our framework is to specify a set of generic variable types that express this performance measure. These variables define a generic frame called "Transmission Rate".

Generic Frame: Transmission Rate:

Input Variables:
- packet arrival

Output Variables
- packet.departure
- operation speed
- delay

Parameters:
- length
- bandwidth
- propagation speed

We proceed to prune the LAN entity structure of Figure 1, with respect to this generic frame. While pruning we select unique entities from every specialization of the entity TRANSMISSION SYSTEM. The choice of a specialized entity is made based on the designers's expertise, the design requirements and constraints that they impose. For instance, the choice between broadband and baseband depends on the distance, estimated data transfer rates, number of devices to be interconnected and cost. Topology type is affected by environment in which the network will operate, distribution of the devices, and access methods. To illustrate how constraints may affect the selection of specialized entities let us consider the LAN substructure depicted in Figure 2.

Constraints:

Coupling:

C1. For a pair of sending and receiving channels,
sending-channel-bandwidth <= receiving-channel-bandwidth.

C2. If base frequencies of receiving and sending channels that are to be connected are different then a frequency converter must be used.

C3. A receiving channel cannot be connected to another receiving channel.

C4. A sending channel cannot be connected to another sending channel.

C5. A frequency converter cannot connect to two channels of the same type (i.e., receiving or sending).

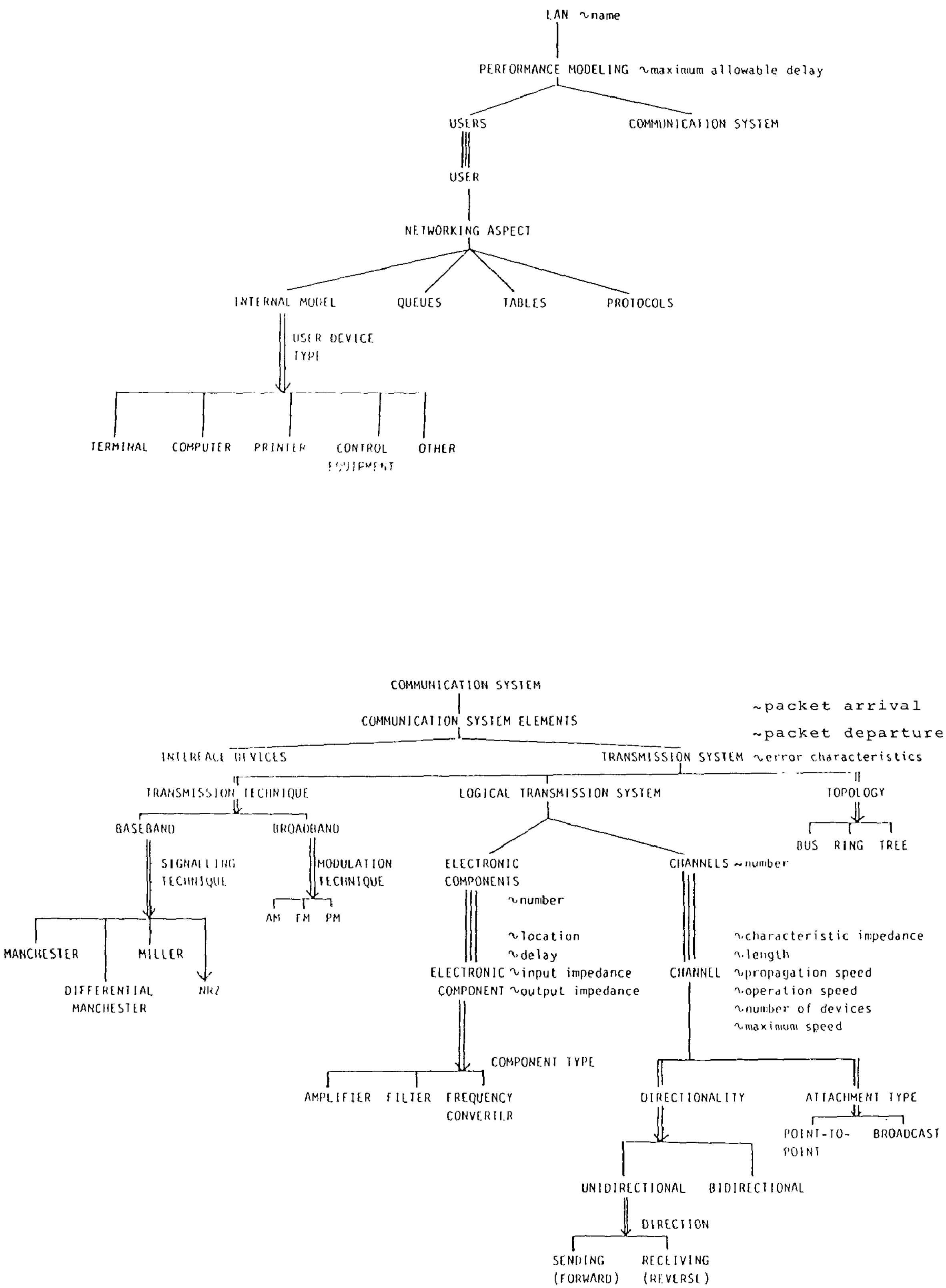

Figure 1. LAN Design Entity Structure

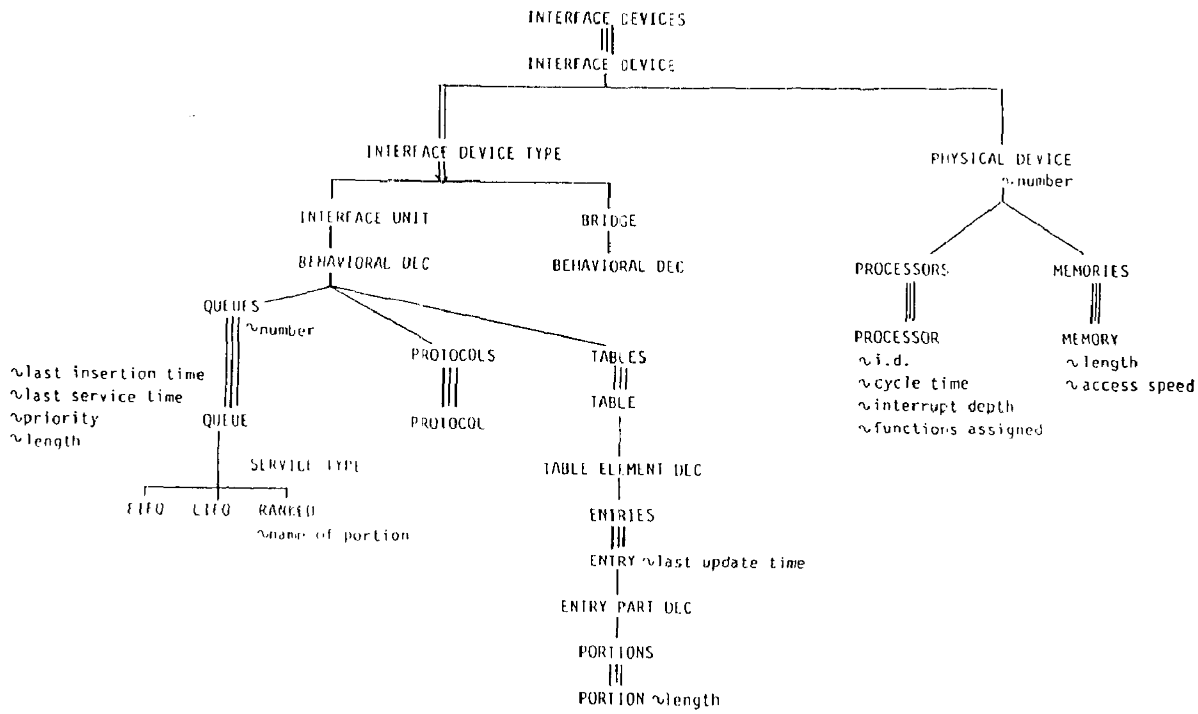

Figure 1. LAN Design Entity Structure - Cont'd

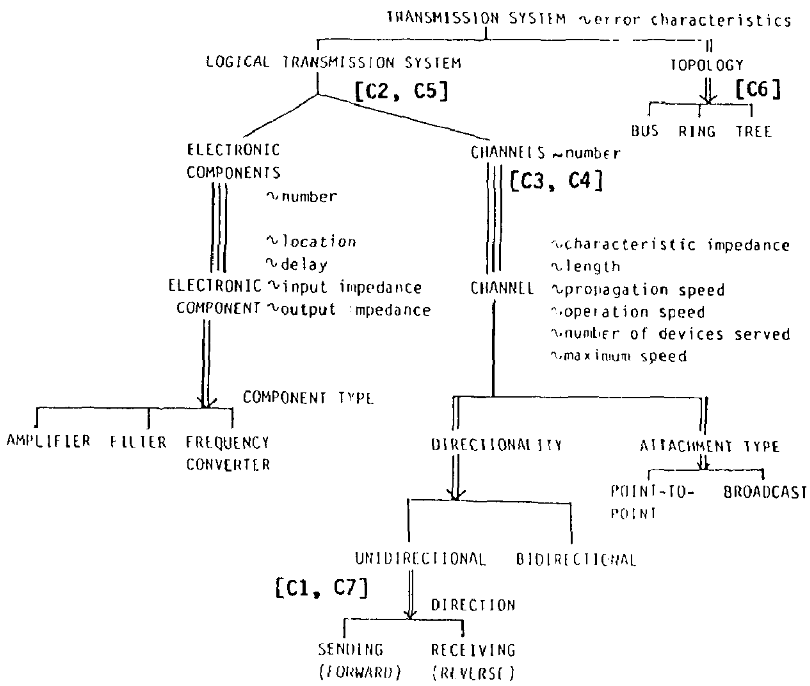

Figure 2. Transmission System Substructure with Coupling and Selection Constraints

Selection:

C6. A tree topology cannot hire Point-to-Point type channels.

C7. A sending and a receiving channel must be selected as a pair.

There are two types of constraints associated with aspects and specializations of Figure 2. The first type is termed <u>coupling constraints</u> (Zeigler, 1984), A coupling constraint specifies a manner in which components identified in an entity's decomposition can be coupled together. The second type is called selection constraints. Selection constraints are associated with a specialization of an entity. They restrict the way in which its subentities can replace it in the pruning process.

For example, the selection constraint 6., will force us to select Broadcast type channels if we decide to use a bus topology. It should be noted here that both coupling and selection constraints may be imposed by technical requirements, standards, resource availability, or cost considerations.

Both coupling and selection constraints have to be associated with the system entity structure in a manner consistent with their meaning. We assert that a coupling (selection) constraint must be attached to an aspect (specialization) which is the least upper bound over the set of entities involved in that constraint.

Although specialization is a distinct concept from that of aspect, there is a way of mapping a specialization hierarchy into an equivalent decomposition aspect. The transformation, described in detail in Zeigler (1984), involves the multiple decomposition concept. To illustrate the principle of such a mapping let us transform the UNDIRECTIONAL specialization of the entity CHANNEL of Figure 2, into a corresponding aspect UNDIRECTIONAL DECOMPOSITION. According to the transformation rules, UNDIRECTIONAL aspect must belong to the multiple entity CHANNELS. The transformed subentity structure is given in Figure 2a.

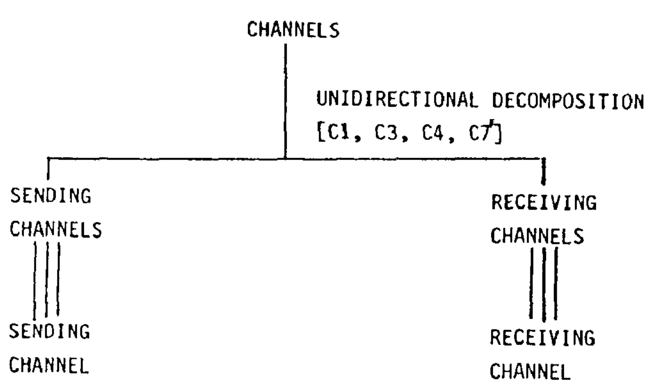

Figure 2a. Mapping Specialization into Decomposition Aspect

C7. The number of selected receiving channels must be the number of selected sending channels. (implied by constraints 3 and 4).

It is important to notice that the above transformation maps any selection constraint into a corresponding coupling constraint. Thus, the selection constraint C7 of Figure 2, becomes the coupling constraint C7' in Figure 2a.

The substructure resulting from pruning is illustrated in Figure 3. The transmission system depicted in Figure 3, contains channels and frequency converters. The channels may be allocated for sending or receiving operations. The number of channels and frequency converters will be determined during the network model synthesis process.

The entity structure of Figure 3 is converted to the transmission system composition tree of Figure 4. At this point we proceed to construct the model of the transmission system by generating rules for model synthesis.

4. LAN DESIGN MODEL SYNTHESIS

We now formulate structural constraints and convert them into a production rule scheme to synthesize a transmission system model for the pruned entity structure of Figure 3. To do this within the limited space of this paper we must make some simplifications and assumptions about the system being synthesized.

Let us define the problem and factors that impose constraints on design of a transmission system for a local area network. We assume that there are "n" devices, each with an expected data rate, that must be interconnected by using "m" channels. Every channel has a maximum number of devices that it can accommodate. Channels must be interconnected by "k" devices called bridges. At any time all the devices in the network must be fully interconnected.

For the sake of simplicity we disregard some details concerning channels. For example, we do not take into account the location of amplifiers, environmental constraints such as tunnel structures, and high temperature points which cannot be passed by a cable. Other factors such as distribution of devices, concurrency of sessions, protocol influence on the system capacity, and cost of maintenance and extensions are not considered either.

We assume, however, that some measure of interconnectivity is available. This measure guides the assignment of devices to channels. Within its limited scope the transmission system being designed must satisfy the following rules:

```
1.  If   there is no unconnected device
         there is no pair of channels that
         is unconnected
         number-of-bridges-used <= max-
         number-of-bridges
         number-of-channels-used <= max-
         numbers-of-channels

    then
         Print "Transmission System is
         Completed"

2.  If   state is null
         a channel is available

    then

         add this channel to the transmission
         system
         update number-of-channels-available
```

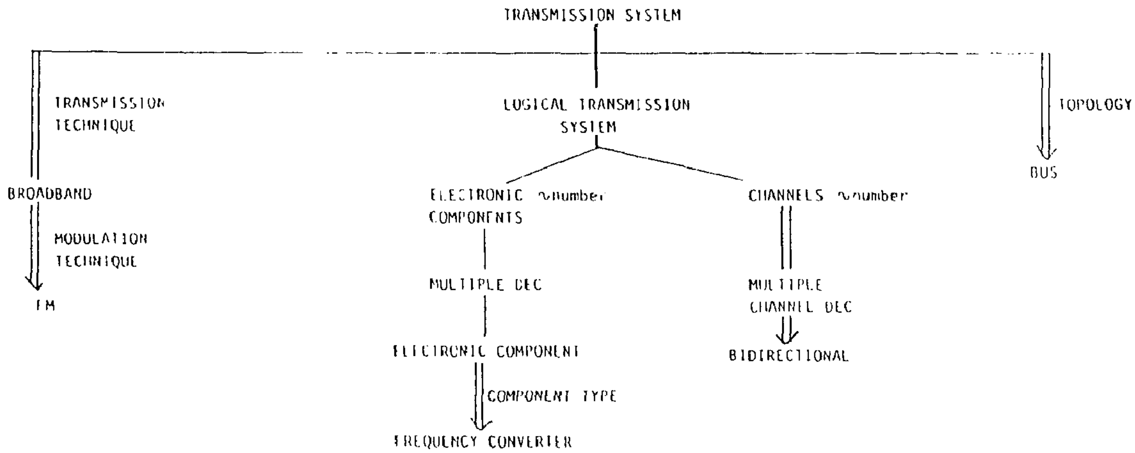

Figure 3. Pruned Entity Structure for Transmission System Design

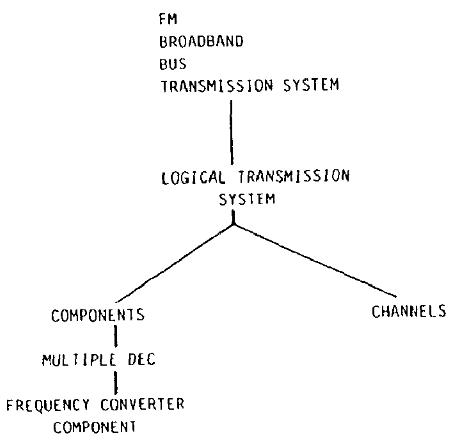

Figure 4. Transmission System Model Composition Tree

3. If there is an unconnected device
   there are channels with room for it

   then

   connect this device to the channel C(i) which maximizes the interconnectivity measure
   update number-of-devices-interconnected
   update available-capacity-of-channel-C(i)

4. If there is a pair of channels
   there is no channel with room for it
   a channel is available

   then

   add this channel to the transmission system
   update *number-of-channels-available*

5. If there is a pair of channels
   that are not interconnected
   a bridge is available
   (the channels are completely filled
   or
   (no more devices are left to be interconnected)

   then

   add a bridge between this pair of channels
   update number-of-bridges-available

The above rules are intended to generate all model structures that satisfy the design constraints. The model construction process will then proceed in a hierarchical manner as defined by Zeigler (1984).

Before a final design is selected, the model should be evaluated with the help of simulation studies. In previous work (Rozenblit, 1986) we have developed an experimental frame-based framework for performance evaluation of design models. To obtain meaningful performance data for our LAN design example, the transmission system should be embedded in a larger model, namely, the model of a LAN communication system. Such a model can be constructed using our pruning and synthesis framework.

Simulation would then proceded in an experimental frame constructed from the generic frame "Transmission Rate". The experimental frame would define input segments of packets arriving with a certain interarrival distribution. It would collect and summarize data concerning the speed of data transmission. The frame would also control the execution of simulation experiments (see Zeigler, 1984 for details concerning the experimental frame concepts).

5. SUMMARY

We have presented an application of the knowledge-based design framework to local area network design. The entity structure developed for LANs is used to organize the knowledge about the networks in a hierarchical manner. The pruning procedures and production rules restrict the design domain to network models related to the design objective and constraints.

Other approaches to LAN design are based on queuing models (Sauer and MacNair, 1983), or programming language-like descriptions of the network architecture (Chlamtac, et. al., 1984). The advantages of our approach are: a) its objectives driven nature, b) a knowledge representation scheme that enables us to generate a variety of design alternatives, c) the framework is easily amenable to computerization, thus provides a formal basis for the construction of an expert environment for LAN design support.

REFERENCES

Chlamtac, I., Jain, R., (1984). "A Methodology for Building a Simulation Model for Efficient Design and Performance Analysis of Local Area Networks", Simulation, Vol. 42, No. 2, pp. 57-66.

Fasang, P.P., Ulrich, P., Whelan, M., (1983). "A Perspective on the Levels of Methodolo gies in Digital System Design", Proc. of the 1983 IEEE Conference on Computer Design.

Gonauser, M., Sauer, A. (1983). "Needs for High-Level Design Tools:, Proc. of the 1983 IEEE Conference on Computer Design.

Gonasuer, M., Kober, R., Wenderoth, W. (1983). "A Methodology for Design of Digital Systems and Requirements for a Computer Aided System Design Environment", IFIP WG 10.0, Sept. 1983.

Hawe, B., et. al. (1984). "Transparent Interconnection of local area networks with bridges:, Journal of Telecommunication Networks, 3(2), pp.116-129.

Hutchinson, D., et. al. (1985). "Local Area Networks, an Advanced Course", Springer-Verlag.

Rozenblit, J.W. & B.P. Zeigler, (1986). "Design and Modelling Concepts", in: Encyclopedia of Robotics, John Wiley and Sons, New York. (in press).

Rozenblit, J.W. & B.P. Zeigler, (1985) "Concepts for Knowledge-Based System Design Environments", Proc. of the 1985 Winter Simulation Conference, pp. 223-231, San Francisco.

Rozenblit, J.W., (1986). "A Conceptual Basis for Intergrated, Model-Based System Design", Tech. Report. Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ.

Sauer, C.H., MacNair, E.A., (1983). "Simulation of Computer Communication Systems", Englewood Cliff, New Jersey: Prentice-Hall.

Sevinc, S., (1986). "Entity Structure Representation for Local Area Network Simulation", MS Thesis, Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ.

Stallings, W. (1984) "Local Networks:, MacMillan Pub. Co., New York.

Tanenbaum, A.S., (1981)."Computer Networks", Prentice-Hall, Englewood Cliffs, New Jersey.

Zeigler, B.P., (1984). "Multifacetted Modelling and Discrete Event Simulation", Academic Press, London.

JERZY W. ROZENBLIT is an assistant professor in the Electrical and Computer Engineering Department at The University of Arizona. He received his Ph.D. in Computer Science from Wayne State University in Detroit, in 1985. His research interests are in the areas of modelling and simulation, system design, and artificial intelligence. He is a member of ACM, IEEE Computer Society, and The Society for Computer Simulation.

Jerzy W. Rozenblit, Assistant Professor
Department of Electrical & Computer Engineering
The University of Arizona
Tucson, Arizona 85721
(602) 621-6177

Suleyman Sevinc was born in Turkey on November 27, 1960. He received his B.S. in Electrical Engineering from the Technical University of Istanbul, Turkey, in 1983. After being ranked first in a nationwide Electronics Engineering exam, he was awarded a full scholarship by the Turkish Ministry of Education to pursue his Ph.D. in the United States. He completed his M.S. degree in Electrical and Computer Engineering at The University of Arizona, Tucson, in May 1986. He is currently a doctoral student in the Department of Electrical and Computer Engineering at

The University of Arizona. His research interests include artificial intelligence and modelling, simulation, and design of local computer networks.

Suleyman Sevinc, Graduate Student
Dept. of Electrical and Computer Engineering
The University of Arizona
Tucson, Arizona 85721

BERNARD P. ZEIGLER is a professor of Electrical and Computer Engineering at The University of Arizona. He is author of "Multifacetted Modelling and Discrete Event Simulation", Academic Press, 1984, and "Theory of Modelling and Simulation", John Wiley, 1976. His interests include artificial intelligence, distributed simulation, and expert systems for simulation methodology.