AI Papers, 1988
© by The Society for Computer Simulation International
ISBN 0-911801-35-9

Towards automatic generation of experimental frames in simulation-based system design

Jhyfang Hu and Jerzy W. Rozenblit
AI and Simulation Group
Department of Electrical and Computer Engineering
University of Arizona
Tucson Arizona 85721

ABSTRACT

This paper presents a methodology for design performance evaluation. Such a methodology provides a basis for the formulation and realization of simulation experiments in system design studies. More specifically, the methodology addressed here concerns computer-aided development of the experimental frame concept for multi-objective design evaluation. In this framework design models can be tested within a number of objectives, taken individually or in trade-off combinations. The design specification description, organization of the experimental frame base, and methods for automatic retrieval and generation of frames are presented. These concepts form a basis upon which an integrated, knowledge-based design support environment is being developed.

BACKGROUND

The authors' recent research has focused on developing a knowledge-based engineering design methodology. This framework is based on the formal structures underlying the multifacetted modelling methodology, namely, that of the system entity structure and generic experimental frame (Rozenblit 1986; Rozenblit and Zeigler 1987). The system entity structure formalism is employed to structure the family of design configurations. Generic experimental frames reflect design objectives. They consist of variable types that express performance indices (Wymore 1980) associated with a given design objective. Generic frames applied to design entity structures in a process called pruning, produce a set of design model structures that conform to the design performance requirements and experimental frames. Design models are then constructed and evaluated in respective experimental frames (Rozenblit and Zeigler 1986). Briefly, an experimental frame defines a set of input, control, output, and summary variables, and input and control trajectories (Zeigler 1984). These objects specify conditions under which a model can be observed and experimented with.

A number of issues need to be addressed in the framework being developed. First, design objectives and requirements should be expressed in the form amenable to computer-aided generation of a design model. Secondly, design model synthesis should proceed automatically in design constraints. Also, design model evaluation should include multi-objective simulations. Rule-based model synthesis approach has been communicated in (Rozenblit and Huang 1987). Here, the issues of design representation and evaluation are addressed.

The research presented in this paper is part of a larger effort to develop an integrated, knowledge-based design support environment. A simulation environment called DEVS-Scheme is used as a vehicle for design performance evaluation.

DEVS-Scheme Simulation Environment

DEVS-Scheme (Zeigler 1986, 1987a) is a knowledgebased simulation environment for modelling and design that facilitates construction of families of models in a form easily reusable by retrieval from a model base. The environment supports construction of hierarchical discrete event models and is written in the PC-Scheme language which runs on IBM compatible microcomputers and on the Texas Instruments Explorer.

System Entity Structure Knowledge Representation

Model specification and retrieval in the DEVS-Scheme simulation environment is mediated by a knowledge representation component designed using the system entity structuring concepts (Zeigler 1984, 1986, 1987). The system entity structure incorporates decomposition, taxonomic, and coupling knowledge concerning a domain of real systems (Rozenblit et al. 1986; Sevinc and Zeigler 1987). A user prunes the entity structure according to the objectives of the modelling study obtaining a reduced structure that specifies a hierarchical discrete event model. Upon invoking the transform procedure, the system searches the model base for components specified in the pruned entity structure and synthesizes the desired model by coupling them together in a hierarchical manner. The result is a simulation model expressed in DEVS-Scheme which is ready to be executed to perform simulation studies. Related work focuses on developing automated writing of rule-based synthesis system from entity structure specifications (Rozenblit and Zeigler 1985, 1987; Rozenblit 1986; Rozenblit and Huang 1987).

In the ensuing sections, efforts towards incorporating the experimental frame-based evaluation into the above environment are described.

AUTOMATIC FRAME GENERATION

In a conventional simulation approach the experimental frames are usually designed by the modeller. The simulation environment is unable to generate and couple automatically an appropriate experimental frame to the design model. The designer must check the Input/Output (I/O) consistency and coupling schemes of the experimental frame. For different design objectives, the designer has to repeat this process several times. To reduce the total design cycle, the experimental frames should be generated automatically by the system itself based on the design constraints and objectives. Through the methodology of automatic frame generation, the responsibility for the design of experimental frames will be switched from the user to the system. The main objective of the automatic frame generation is to automate and optimize the design process and provide more appropriate evaluation environment for system design. This will ultimately facilitate the development of a Knowledge-Based Design Support System (KBDSS) for computer-aided design consulation.

The experimental frame base contains all atomic frames for the evalution of the design model. An atomic frame is essentially a DEVS atomic model, which is the basic component used to build an experimental frame component, such as Generator, Acceptor, or Transducer. The structural realization of an experimental frame is depicted in Figure 1. The methodology for experimental frame generation presented in this paper consists in:

- Specifying the design constraints, objectives, and criteria weighting schema.
- Identifying generic frames in the design specification and extracting all relevant atomic frames.
- Aggregating atomic frames and coupling the aggregated frames to the design model.
- Activating the simulators and collecting the simulation results.
- Deriving design alternatives by comparing the simulation results with the performance constraints. This will result in a number of design alternatives.
- Selecting the best design by evaluating design alternatives within a trade-off frame in which all conflicting objectives will be considered simultaneously.

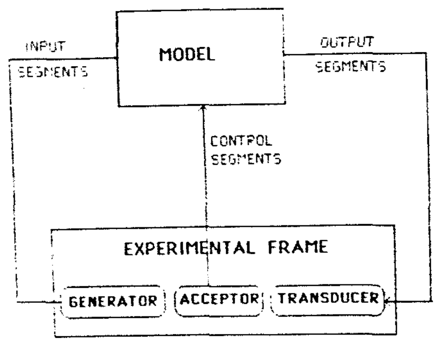


Figure 1 Structural realization of an experimental frame

DESCRIPTION OF DESIGN SPECIFICATION

In the Knowledge-Based Design Support System, the design alternatives are implied by the system entity structure. By giving the design specification, all design alternatives which conform to the design constraints can be pruned from the system entity structure. The design specification is defined as the collective information about design constraints, objectives, and criteria weighting. It is important to provide an efficient schema for design specification. The Design Specification Form (DSF) as shown in Figure 2 will serve as a user-friendly interface to accept the user's design specification.

The performance indices field of DSF is associated with a Performance Index Reference Tree (PIRT). This facilitates selecting appropriate performance indices for the specification of design constraints and objectives. The project title field of DSF is used for specifying the name of the project. The design constraints field of DSF implies the requirements that must be satisfied by the resulting system. Each performance constraint is expressed by a (relation performance-index value) triple as shown below:

$(Logic - Operator\ Performance - Index\ Value)$

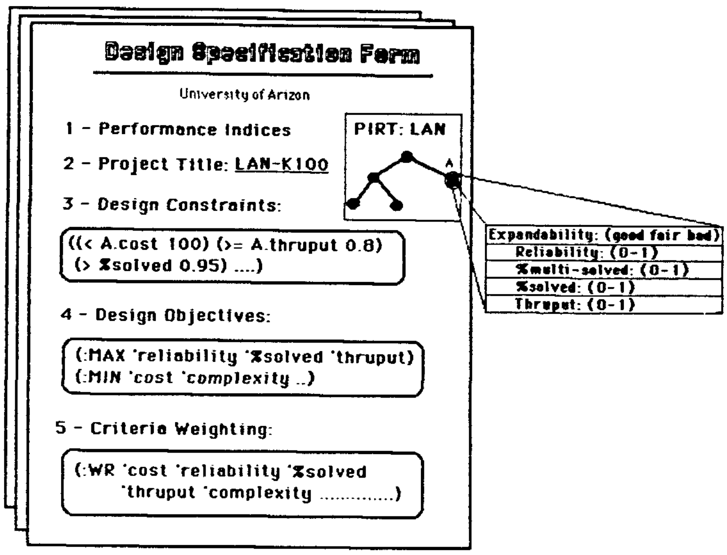
For example, in the design of Local Area Network (LAN), the constraint on a ring network can be expressed as:

$(< RING.Average - Transfer - Delay\ 280\mu sec)$

The "relation" indicates the requirement of the performance index over the specified value. The "performance index" is composed of "object" and "generic frame". The "object" indicates the associated model. The "generic frame" provides variable types, such as Throughput, Efficiency, Cost, etc.. The "value" is the index used to indicate the quality (good, fair, bad, high, medium, low, etc.) or quantity (100, 0.9, etc.) of the associated attribute. The performance constraints can be classified into two categories:

- Static performance constraints: This type of constraints is expressed by performance indices which are essentially design parameters used to characterize the physical properties of the design object. The static performance indices are usually predefined during the construction of models. The static performance constraints can be directly used as criteria to prune the entity structure without invoking any simulation since their values are known and are not modified by the system's dynamic behavior. Typical examples of static performance indices are the unit cost, maximum capacity, area, volume etc.
- Dynamic performance constraints: Dynamic constraints are expressed by performance indices which are relevant to the behavior of the system being designed. Simulation methods must be invoked to measure the performance of model components. For example, typical dynamic performance indices in the design of local area network are network throughput, average transfer delay, effective service time etc.

Design objectives imply the design goal. Specification of design objectives will be used to conduct the design opti-


Design Specification Form

University of Arizon

1 - Performance Indices

2 - Project Title: LAN-K100

3 - Design Constraints:

((< A.cost 100) (>= A.thruput 0.8)
(> %solved 0.95) ....)

4 - Design Objectives:

(:MAX 'reliability '%solved 'thruput)
(:MIN 'cost 'complexity ..)

5 - Criteria Weighting:

(:WR 'cost 'reliability '%solved
'thruput 'complexity ....)

Figure 2 Design specification form (DSF)

mization. Typical design objectives are stated as maximize or minimize one or more performance indices as shown below:

((MAX Performance-Indices ...) (MIN Performance-Indices ...))

For example, in the design of local area network, the objectives of a ring network may be expressed as:

((MAX RING.Thruput RING.Utilization ...)
(MIN RING.Cost RING.Average-Transfer-Delay ..))

The criteria weighting conveys the designer's preference over the set of evaluation measures. Different type of weighting schemes will be allowed to specify the relationship between criteria (Kmietowicz 1981; Osyczka 1984). Typical weighting schemes are:

- Unknown weighting: The unknown weighting is used when the user is unable to give any preference information about design criteria. Under this situation, all criteria are regarded as having equal importance.
- Complete weighting: The complete weighting is used when the user is able to specify exact preference for each criterion. An example of this category is to assign each criterion a positive weight and let the sum of total weights be equal to 1.
- Ranked weighting: The ranked weighting is used whenever partial preference information is available but is not comprehensive enough to give exact preference values of criteria. A typical example in this category is to list the trade-off order of criteria.
- Fuzzy weighting: The fuzzy weighting is used whenever the user is able to define the preference range (indicated by the lower-bound and upper-bound) of criteria.

EXTRACTION/AGGREGATION OF ATOMIC FRAMES

The first step toward the automatic frame generation is to identify the generic frame type from the design specification. Each generic frame will activate the extraction of one or more atomic frames. The extraction of atomic frames starts from the reference to the Performance Index Tree (PIT). The performance index tree is a semantic data structure indicating how the related performance index is evaluated within design parameters, atomic frames, and/or lower-level performance indices. A typical performance index tree used for evaluating the utilization of a local area network is illustrated in Figure 3. As shown in Figure 3, each node of the performance index tree may correspond to a design parameter, an atomic frame, or a performance index. The Transducer Aggregation Key (TAK) is used to indicate how the related performance index can be aggregated from its child nodes.Via referring to the performance index tree, all relevant atomic frames can be identified, extracted, and aggregated by the system. As shown in Figure 3, to build an experimental frame for utilization, the system starts searching its sub-level frames. Since the number of data bits in a packet (D) and number of overhead bits in a packet (H) are known as design parameters, only the frame of normalized throughput (S) will be extracted for aggregation.

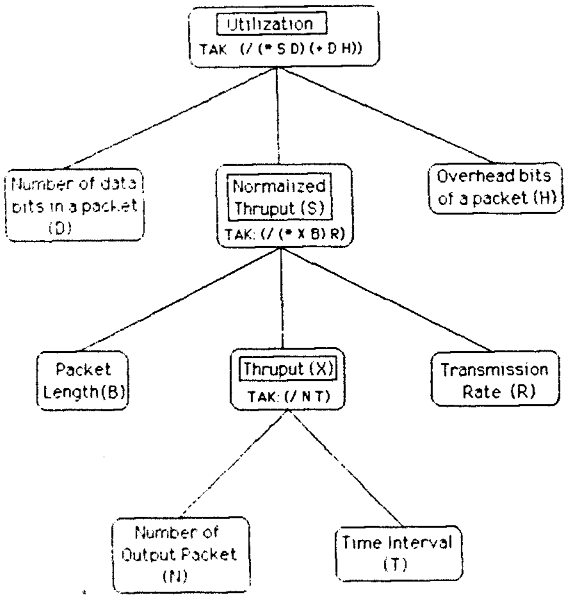


Figure 3 Performance index tree (PIT)

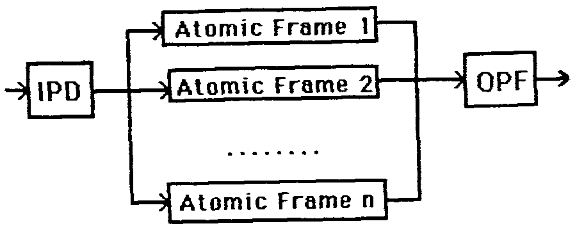


Figure 4 Structure of an aggregated frame

If the frame of normalized thruput (S) is not contained in the frame base, the system must aggregate the frame of normalized thruput before the frame of utilization is aggregated.

In order to ease the manipulation of Performance Index Trees (PIT), each system entity structure is associated with a Performance Index Reference Tree (PIRT). Each attribute contained in a PIRT node implies a performance index for the corresponding model. The performance index is evaluated by referring to the Performance Index Tree (PIT). All atomic frames related to the performance index being evaluated are then extracted from the frame base to form an experimental frame for evaluation purposes.

During the construction of an experimental frame, two special function modules, the Input Packet Distributor (IPD) and the Output Packet Formatter (OPF), will be installed in the frame. The IPD is used to distribute the contents of input packets to the relevant atomic frames based on detecting the attribute (or variable type) of performance indices. The evaluations of performance indices are accomplished by operating the aggregation key over the relevant outputs of atomic frames. The OPF is used to pack the experimental results into a valid packet based on the I/O specification of the design model. Finally these packets will be sent to the appropriate output ports for further propagation. The schematic structure of an aggregated frame is shown in Figure 4.

Due to the communication between the model structure and experimental frames, the system-generated frame must be *validated* by checking the I/O consistency with the design model. A *valid frame* is defined as the experimental frame which can be coupled to the design model for evaluation purpose without causing any I/O inconsistency. The I/O specification of a design model is expressed as a structure of:

$$< M \; IN_{ports} \; OUT_{ports} \; IN_{trajs} \; OUT_{trajs} \; T >$$

where:

M: name of the model
IN_{ports}: set of input ports
OUT_{ports}: set of output ports
IN_{trajs}: set of input trajectories
OUT_{trajs}: set of output trajectories
T: time scale

EVALUATION UNDER MULTIPLE CRITERIA

Given a set of design constraints and objectives, there may exist more than one design alternative which conforms to the user's design specification. Each design alternative will be evaluated under the system-generated experimental frame. Since design evaluations proceed under multiple, often conflicting, objectives, it is necessary to define methods that handle such situations adequately. In order to determine the best design from a number of design alternatives, the Trade-Off Frame (TOF) is designed to perform the selection purpose. A trade-off frame is a coupling of individual experimental frames (each corresponding to a single performance objective) with trade-off orderings defined over the set of output variables of each frame. The Multiple Criteria Decision Making (MCDM) methods will be integrated in the trade-off frame to make the best decision for the user. It is hard to say which MCDM method is best. Furthermore, no single MCDM method can be applied to solve all different types of decision making problems. To assure a reliable solution, the most appropriate MCDM method will be selected by a rule-based system according to the weighting scheme specified in the design specification. For example, if the fuzzy weighting is used to indicate the preference of criteria, simple application of MCDM methods, such as Maximin, Maximax, Minimax, or Bayes-Laplace (Kmietowicz 1981; Osyczka 1984) may cause incorrect selection of the best design. The schematic representation of a TOF evaluation is explained in Figure 5.

STRUCTURE OF THE DESIGN DATABASE

In order to facilitate the data manipulation for the design process, a database will be designed and organized into a hierarchical modular structure. Several modules will be included in this database:

- System entity structures: Each system entity structure represents a design problem domain.
- Pruned entity structures: Each pruned entity structure represents a design model structure.
- DEVS models: Each entity node of the system entity structure has a corresponding DEVS model for simulation.
- PIRT: Each node of the Performance Index Reference Tree (PIRT) contains performance indices. Each performance index is associated with a Performance Index Tree (PIT) to indicate all relevant atomic frames.
- MCDM: Each Multi-Criteria Decision Making (MCDM) method can be integrated in a trade-off frame to solve the multi-criteria evaluation problems.

CONCLUSIONS

Modern engineering design is a highly complex process involving consideration of multiple objectives, constraints, and configurations. The savings in human time/effort and in physical resources afforded by an AI-based simulation environment are potentially enormous. Through the methodology of automatic frame generation, the contributions to the system design are:

- Complicated engineering design process becomes transparent to the system designer.
- Design cycle is reduced by automating the design process. The automation includes generating design configurations, validating the design alternatives by comparing the simulation results with the design constraints, and selecting the best design configuration by considering all conflicting design objectives.
- The concept of associating the semantic structural trees to the behavior model/frame description facilitates the knowledge manipulation in the model-based expert design support system.

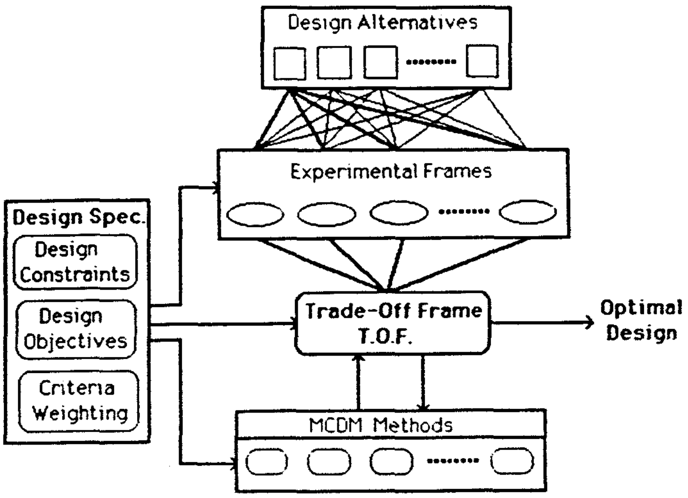


Figure 5 Trade-off frame (TOF) evaluation

REFERENCES

Kmietowicz, Z. W. 1981. *Decision Theory and Imcomplete Knowledge*. Gower Publishing Company Ltd., England.

Osyczka, Andrzej. 1984. *Multicriterion Optimization in Engineering*. Ellis Horwood Ltd., England.

Rozenblit, J. W. 1986. "A Conceptual Basis for Integrated, Model-Based System Design," Technical Report, Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona.

Rozenblit, J. W. and B. P. Zeigler. 1985. "Concepts for Knowledge based System Design Environments," *Proceedings of the 1985 Winter Simulation Conference*, San Francisco, California, December.

Rozenblit, J. W. and B. P. Zeigler. 1986. "Entity Based Structures for Experimental Frame and Model Construction," in: *Modelling and Simulation in the Artificial Intelligence Era.* (ed. M. S. Elzas, et. al.) North Holland, Amsterdam.

Rozenblit, J. W.; S. Sevinc; and B. P. Zeigler. 1986. "Knowledge Based Design of LANs Using System Entity Structure Concepts," *Proceedings of the 1986 Winter Simulation Conference*, Washington, D.C., December.

Rozenblit, J. W. and Y. M. Huang. 1987. "Constraint-Driven Generation of Model structures," *Proceedings of the 1987 Winter Simulation Conference*, Atlanta, Georgia, December.

Rozenblit, J. W. and B. P. Zeigler. 1987. "Design and Modelling Concepts," to appear in *Encyclopedia of Robotics*. John Wiley and Sons, New York.

Sevinc, S. and B.P. Zeigler. 1987. "Entity Structure Based Design Methodology: A LAN Protocol Example", To appear in *IEEE Transactions on Software Engineering.*

Wymore, W. A. 1980. "A Mathematical Theory of Systems Design," Technical Report, University of Arizona, Tucson, Arizona.

Yu, Po-Lung. 1985. *Multiple-Criteria Decision Making*. Plenum Press, New York.

Zeigler, B. P. 1984. *Multifaceted Modelling and Discrete Event Simulation*. Academic Press, London.

Zeigler, B.P. 1986. "DEVS-Scheme: A LISP-based Environment for Hierarchical, Modular Discrete Event Models", Technical Report AIS-2, Dept. of Electrical and Computer Engr., The University of Arizona, Tucson, AZ.

Zeigler, B.P. 1987. "Knowledge Representation from Newton to Minsky and Beyond", *Applied Artificial Intelligence*. Vol. 1, Jan. pp. 87-107.

Zeigler, B.P. 1987a. "Hierarchical, Modular Discrete Event Modelling in an Object Oriented Environment", *Simulation Journal*. November.