CONCEPTS AND SOFTWARE FOR ADVANCED SIMULATION METHODOLOGIES

Bernard P. Zeigler[1]
Department of Computer and Communication Sciences
University of Michigan
Ann Arbor, MI 48104

ABSTRACT: The sine qua non of advanced simulation methodology is the objective of providing comprehensive and integrated assistance in all aspects of the modelling and simulation process. This paper reviews the activities that can be assisted by the computer and an architectural framework for software (and hardware) systems intended to provide such assistance. Trends in state-of-the-art simulation software, progress in the design of software for advanced methodologies, and urgent large scale, multifacetted problems are surveyed. It is concluded that we may expect rapid development and uptake of comprehensive and integrated simulation support systems in the next decade.

1. INTRODUCTION

1.1 What is Advanced Simulation Methodology?

If 'simulation methodology' means roughly 'how to do simulation' then the modifier 'advanced' is intended to suggest a relativity to the current state of affairs i.e., 'how it should be possible to do simulation in the future'. (What will be called 'advanced' in the nineties will most certainly be different from what we call advanced today.)

Perhaps the most descriptive keyword for characterizing the envisioned simulation practice of the future is 'comprehensive'. That is, practitioners will be doing the same things they do now, except that, individually and collectively, they will be able to carry out more of these activities, to greater effect, in the time and money available. Stated another way, currently time pressures and cost constraints greatly limit our ability to carry out all the activities necessary to achieve reliable simulations. Software for advanced simulation methodology should remove such limitations as a reason for simulation studies to fall short of their expectations.

Of course going through all the motions will be of little consequence if they are not done well. The methodologist seeks to disentangle the various activities involved in modelling and simulation and to disentangle the various activities involved in modelling and simulation and to suggest concepts and procedures for their proper execution. The software developer may then create systems which embody the suggested concepts and support the indicated procedures. It is in this light that this paper attempts to survey the state-of-the-art in software for advanced simulation methodologies.

1.2 Modelling and Simulation Activities

While definitions of modelling and simulation abound (Pritsker, 1979) few if any suggest the comprehensiveness of the required activities just alluded to. Such a definition however, is implicit in the top down structuring of the numerous possibilities for computer-aided modelling provided by Ören (1980).

---

[1]Research under the author's direction reported here was supported by Grant No. DAERO-78-G-088 of the European Research Office of the U.S. Army. The dedication of David Belogus and Alexander Bolshoi in this effort are deeply appreciated.

TH0079-4/80/0000-0025$00.75 C 1980 IEEE.
Simulation with Discrete Models: A State-of-the-Art View
T.I. Ören, C.M. Shub, P.F. Roth (eds.)

Modified slightly to emphasize the activities (rather than their support by the computer), Ören's structure is reproduced in several layers of detail in Figure 1 as output by ESP (Zeigler 1980), a system structuring tool of which more will be said later. The simulationist will no doubt be struck by the depth at which the activity called 'simulation' appears. This lowly status is in keeping with Ören's definition of simulation as "experimentation with models" (Ören, 1979). Indeed, reducing the simulation activity to its bare and proper essentials leaves one free to elaborate, and place in their rightful status, the various activities often associated with simulation but truly part of the larger modelling and simulation process.

We may interpret this hierarchy as stating that the modelling and simulation process may be decomposed into the following activities:

---model generation/referencing

---model processing

---behavior processing

---real system experimentation

---model acceptability assessment

Each of these activities is further decomposable as is illustrated in Figure 1. We shall briefly review Ören's treatment of the above categories and refer the reader to the original paper for more details.

1. Model generation/referencing refers to the generation of models either by construction from scratch or by employing models retrieved from a model bank as components to be coupled together.

2. Model processing refers to the manipulation of model descriptions (e.g., to produce documentation, to check consistency, etc.) and to the generation of model behavior (of which simulation is a predominant form).

3. Behavior processing refers to the analysis and display of behavior in a static setting (e.g., in equilibrium) or in a dynamic mode (i.e., concerning state trajectories) or in a structural mode (i.e., changes in model structure are tracked).

4. Real system experimentation refers to the gathering of behavioral data from the real system or any of its component systems of interest.

5. Model acceptability assessment refers to the verification and validation of models but also to a host of other tests of the relationships in which models participate.

We can now refine our earlier definition:

Software systems for advanced modelling and simulation methodologies are sets of tools which attempt to provide comprehensive and integrated computer assistance in carrying out the activities outlined above. (Let us stress the caveat 'attempt to provide' as signifying 'move in the direction of' knowing full well that completeness and full integration are none too likely achievements.)

1.3 Trends in State-of-the-art Simulation Software

What then is the state of the art of such software systems? It is fair to say that no such systems currently exist so that the state of the art refers more to the concepts and designs being developed rather than to operational experience.

It is also fair to say that a number of software systems do exist which point to the feasibility of the advanced systems. Indeed in a recent paper (Zeigler, in press) we pointed out certain trends in contemporary simulation software development which pointed to a natural evolution in the direction being advocated. Among the trends identified were:

```
1 :  ENTITY MODELLING AND SIMULATION PROCESS
  2 :  ASPECT ACTIVITIES DECOMPOSITION
    3 :  ENTITY MODEL GENERATION/REFERENCING
      4 :  ASPECT MODEL GENERATION/REFERENCING DECOMPOSITION
        5 :  ENTITY MODELLING          MODEL_GENERATION
                --DEPTH  5 TRUNCATION
        5 :  ENTITY MODEL_REFERENCING
                --DEPTH  5 TRUNCATION
    3 :  ENTITY MODEL_PROCESSING
      4 :  ASPECT MODEL_PROCESSING_DECOMPOSITION
        5 :  ENTITY BEHAVIOR_GENERATION
                --DEPTH  5 TRUNCATION
        5 :  ENTITY MODEL_MANIPULATION       ALGORITHMIC_MODEL_PROCESSING
                --DEPTH 5 TRUNCATION
        5 :  ENTITY SENSITIVITY ANALYSIS
    3 :  ENTITY  BEHAVIOR PROCESSING
      4 :  ASPECT BEHAVIOR_PROCESSING_DECOMPOSITION
        5 :  ENTITY BEHAVIOR_ANALYSIS
        5 :  ENTITY BEHAVIOR_DISPLAY
    3 :  ENTITY REAL_SYSTEM_EXPERIMENTATION
    3 :  ENTITY ACCEPTABILITY ASSESSMENT
      4 :  ASPECT ACCEPTABILITY_ASSESSMENT_DECOMPOSITION
        5 :  ENTITY OBJECTIVES/MODELS_RELATION
        5 :  ENTITY REAL_SYSTEM/MODEL_RELATION
        5 :  ENTITY SIMULATOR/MODEL_RELATION
        5 :  ENTITY MODEL/MODELS_RELATION
        5 :  ENTITY MODELLING          MODEL GENERATION
          6 :  ASPECT  MODELLING_DECOMPOSITION
            7 :  ENTITY  SPECIFICATION_OF_PARAMETRIC MODELS
            7 :  ENTITY  SPECIFICATION_OF_PARAMETER_VALUES
              8 :  ASPECT S_O_P_V_DECOMPOSITION
                9 :  ENTITY  CALIBRATION
                9 :  ENTITY IDENTIFICATION
                9 :  ENTITY OPTIMIZATION
          5 :  ENTITY  MODEL_REFERENCING
            6 :  ASPECT  MODEL_REFERENCING_DECOMPOSITION
              7 :  ENTITY STORAGE
              7 :  ENTITY RETRIEVAL
    3 :  ENTITY  MODEL_PROCESSING
      4 :  ASPECT  MODEL_PROCESSING_DECOMPOSITION
        5 :  ENTITY  BEHAVIOR_GENERATION
          6 :  ASPECT  BEHAVIOR_GENERATION_DECOMPOSITION
            7 :  ENTITY  POINT_BEHAVIOR_GENERATION       SOLUTION_OF_STATICS
            7 :  ENTITY  TRAJECTORY_BEHAVIOR_GENERATION  SOLUTION_OF_DYNAMICS
              8 :  ASPECT  T_G_B_DECOMPOSITION
                9 :  ENTITY  SIMULATION
                9 :  ENTITY  SIMULATOR_DESIGN
                9 :  ENTITY  OPERATIONAL_GAMING
            7 :  ENTITY  STRUCTURAL_BEHAVIOR_GENERATION
        5 :  ENTITY  MODEL_MANIPULATION      ALGORITHMIC_MODEL_PROCESSING
          6 :  ASPECT  MODEL_MANIPULATION_DECOMPOSITION
            7 :  ENTITY  MODEL_DOCUMENTATION
            7 :  ENTITY  CONSISTENCY_CHECKING
            7 :  ENTITY  MODEL_REDUCTION  SIMPLIFICATION
            7 :  ENTITY  MODEL_COUPLING   INTERFACING
            7 :  ENTITY  MODEL_COMPARISON
```

Fig. 1 Decomposition of Modelling and Simulation Into Activities.

1. Trend toward independence of model specification from procedural and machine required specification (providing the user with model oriented languages which separate him to an increasing extent form implementation details).

2. Trend towards modularity of functional elements in simulation programming (providing clear segmentation of parts of a program devoted to distinct tasks e.g., structure declarations, process descriptions, experimental control, etc.)

3. Trend toward flexibility in modelling formalism (providing the user with a wider range of formalisms in which to describe components of his model).

4. Trend toward tools which provide specialized support of modelling and simulation activities (e.g. statistics, graphics and optimization packages).

5. Trend toward the integration of such tools (e.g., the linking of standard simulation languages to statistical and optimization modules).

6. Trend toward increased interactivity (providing the user with greater immediacy in perceiving the status of computer activity and controlling its direction).

7. Trend toward integration and comprehensiveness in specific application domains (e.g., aircraft design, econometric models, etc.).

8. Trend toward incorporation into larger contexts (employing modelling and simulation modules as parts of larger decision support systems).

(The reader is refered to the above mentioned paper (Zeigler, in Press) for references to software which embody these trends.)

1.4 Advances in Related Software Systems

These trends indicate both that simulationists are asking for more advanced simulation tools and that simulation software designers have responding to this interest. However, it is also fair to say that the intensity of research into next generation software systems in the simulation context has not attained the levels characteristic of other areas. Indeed, methodological research in modelling and simulation methodology has much in common with developments in the areas of software engineering and in data base/information systems design.

One the one hand simulation programs and tools constitute a class of software (with some quite distinctive characteristics) and should therefore benefit from the concepts and tools being developed in structured programming (e.g. top down design; Dykstra, 1976), data strueturing (e.g. abstract data types; Liskov and Zilles, 1974) and data base design (e.g. data model concepts; Nijssen, 1977).

On the other hand, modelling and simulation methodology is playing an increasingly important role in software and data base design methodologies. Indeed there is an intimate interreliance of design and modelling methodologies on each other and their activities are also partially analogous (with some essential differences; see editor's introduction to Methodology in Systems Modelling and Simulation, (Zeigler et. al. eds., 1979). It is not surprising then that computer based software design systems (Teichrow and Hershey, 1978 ; Estrin, 1978 ; Yeh, 1977) resemble the simulation support systems to be described below in architectural philosophy, if not in facilities provided. Likewise data models developed in the data base field resemble the schemes for data representation provided by simulation languages (indeed, a data base management system built upon SIMSCRIPT principles is nearing completion (Markowitz, personal communication)).

As developments proceed in these areas, points of commonality can be shared to mutual advantage (Sanguinetti, 1979 ; Cutler, 1980 ; Beauchamp and Field, 1979 ; Ryan, 1979 ; Rzevski, 1980).

2. DESIGNS FOR ADVANCED SIMULATION SUPPORT SOFTWARE

We have shown then that the needs for advanced simulation methodologies have been recognized and, to some extent, accommodated, but also that the state-of-the-art of computer science suggests that much more can be achieved. In the sequel we shall review some of the systems being designed explicity to support advanced modelling and simulation methodologies First we shall describe an architectural scheme which will serve as a framework for such a review.

2.1 Architecture for Advanced Simulation Methodologies

An architectural framework for advanced software systems was suggested recently (Zeigler, 1980) and is represented in Figure 2 again employing the output of ESP.

In such a scheme, users interact with the computer system through interfaces which enable them to initiate or engage in activities. The sequencing of activities may be partly fixed and partly open to users' control. An activity is executed by one or more processors (in conjunction with the user) and acts upon one or more data bases (where the "data" may be of various kinds to be described). In executing an activity, information is stored in the bases. The information so generated is accessable to the user through the interfaces.

```
1 :  ENTITY  MODELLING_AND_SIMULATION_PROCESS
  2 :  ASPECT  ARCHITECTURAL_DECOMPOSITION
    3 :  ENTITY  USERS
      4 :  ASPECT  USER_CLASSIFICATION
        5 :  ENTITY  MODELLERS
        5 :  ENTITY  EXPERIMENTALISTS
        5 :  ENTITY  STATISTICIANS
        5 :  ENTITY  DECISION_MAKERS
        5 :  ENTITY  SUPPORT_SYSTEM_DESIGNERS
    3 :  ENTITY  INTERFACES
      4 :  ASPECT  INTERFACE_CLASSIFICATION
        5 :  ENTITY  LANGUAGES
        5 :  ENTITY  DIALOGS
        5 :  ENTITY  FORMALISMS
    3 :  ENTITY  ACTIVITIES
      4 :  ASPECT  ACTIVITIES_DECOMPOSITION
        5 :  ENTITY  MODEL_GENERATION/REFERENCING
                  -- DEPTH 5 TRUNCATION
        5 :  ENTITY  MODEL_PROCESSING
                  -- DEPTH 5 TRUNCATION
        5 :  ENTITY  BEHAVIOR_PROCESSING
                  -- DEPTH  5 TRUNCTAION
        5 :  ENTITY  REAL_SYSTEM_EXPERIMENTATION
        5 :  ENTITY  ACCEPTABILITY_ASSESSMENT
                  -- DEPTH  5 TRUNCATION
    3 :  ENTITY  PROCESSORS
      4 :  ASPECT  PROCESSOR_CLASSIFICATION
        5 :  ENTITY  SEQUENTIAL_PROCESSORS
        5 :  ENTITY  PARALLEL_PROCESSORS
    3 :  ENTITY  BASES
      4 :  ASPECT  BASES_CLASSIFICATION
        5 :  ENTITY  MODEL_BASE
        5 :  ENTITY  EXPERIMENTAL_FRAME_BASE
        5 :  ENTITY  SIMULATION_PROGRAM_BASE
        5 :  ENTITY  SOFTWARE_TOOL_BASE
        5 :  ENTITY  BEHAVIORAL_DATA_BASE
```

Fig. 2 Architectural Decomposition of Modelling and Simulation.

2.2 Proposed Software Systems for Advanced Methodology

As suggested earlier no system is likely to provide a complete selection of facilities in each of the components of the above framework. Designs differ in the components that they emphasize and the consequent elaboration that is given to them. Our review will proceed from this point of view.

Model and Experimental Frame Bases

Oren and Zeigler (1979) emphasize certain of the activities and bases of the general scheme. As illustrated in Figure 3, they view the modelling process as being initiated by the incidence of 'objectives' and terminating in the construction of models to meet these objectives. If we think of the modelling process as part of the overall process of decision making, then the objectives derive from requests by decision makers for models with which to assess the efficacy of proposed policies. ('Decision maker' and 'policy' are intended here in a broad sence to include engineer, designer, manager, etc., and implementation, design, tactic, strategy, etc., respectively.) Such

Such objectives are supposed to be formulated as series of questions regarding a real system or its components and ultimately to be formulated as experimental frames. An experimental frame is a specification of the kind of data a model should produce in order to answer the questions of interest. The concept however must be meaningful both for real system as well as model experimentation since in principle the same data could be obtained from the real system (although there are many reasons why models are preferred in practice).

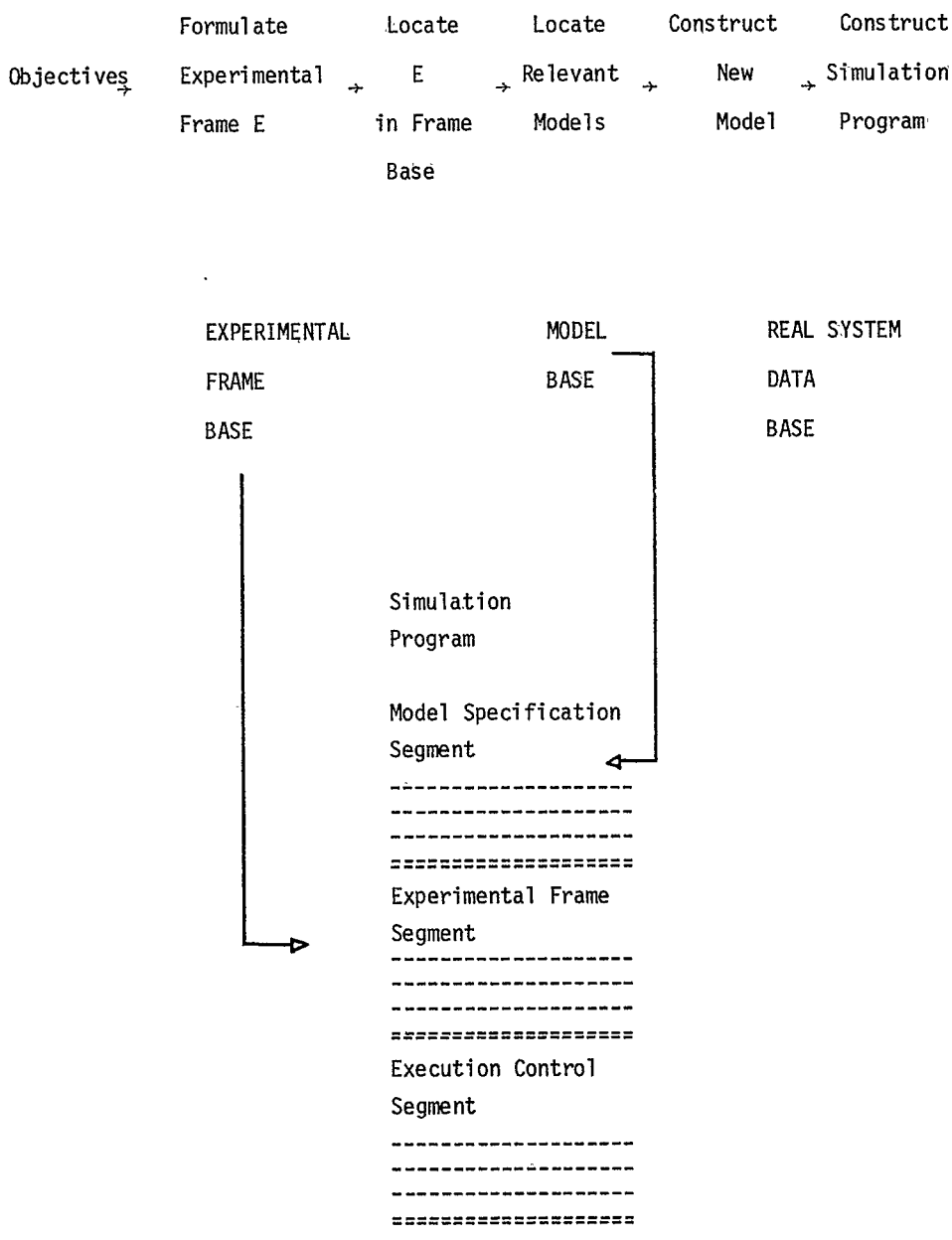

Fig. 3 Concepts for Computer Support of Advanced Methodologies

The support system should maintain a base of previously defined experimental frames and help to locate a new frame among them. A model base of previously developed models should also be maintained which should be referenceable from the frames base. That is, knowing which should be referenceable from the frames base. That is, knowing which resident frames are similar to the new frame should provide an entree to the existing models which are relevant to it. Such models, after adaptation and simplification, should serve as components to be interfaced to form a model to which the new frame is applicable.

It should be noted that the experimental frame is key concept in model assessment since model validity is properly formulated as a relation involving a model, a real system, and a frame in which the behavioral data of the two are compared. We shall expand somewhat more fully on these concepts when discussing multifacetted systems (see major section 3).

Model oriented language interface

In such a modelling support system, a simulation language may be regarded as interface through which the user directs the writing of simulation programs. The suggested structure of simulation program, as illustrated in Figure 4, comprises model specification, experiemental frame, and execution control segments. The GEST78 simulation language (Ören, 1978; Elzas, 1979) is being designed to reflect these fundamental distinctions in a simulation program's tasks. Ultimately, the model specification and experimental frame segments should originate from their respective bases.

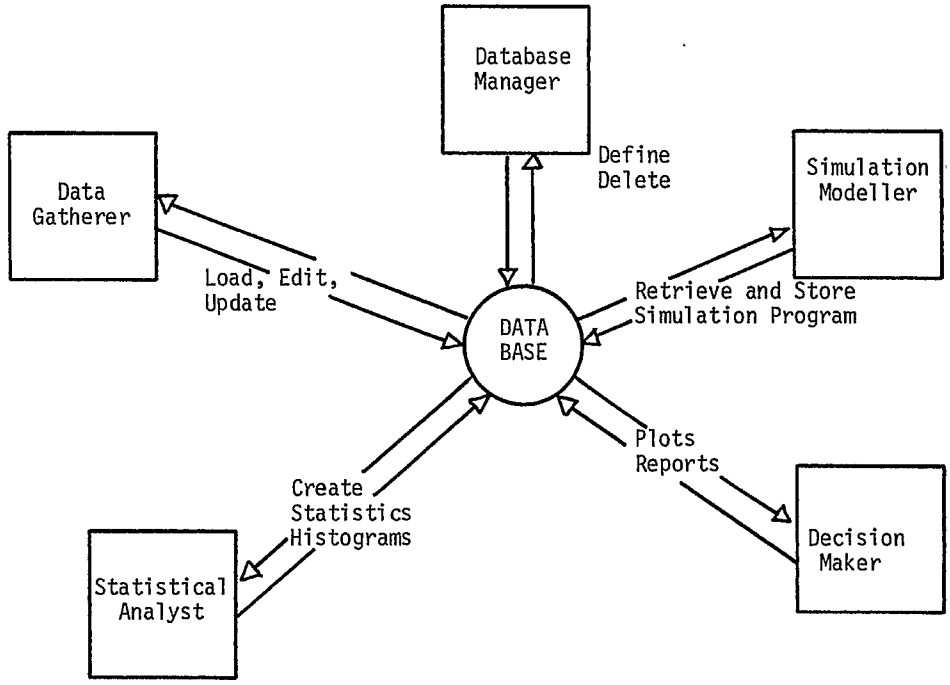

Fig. 4 Users and their relation to behavioral data base (from Figure 2 of Standridge and Pritsker (1980)).

Other languages which should be compatible with the interface role are PROSIM78 (Sierenberg and de Gans) and COSY (Cellier, 1979). It is characteristic of all three that they are model oriented (i.e., support the model specification/execution control distinction) and truly general purpose in the sense of providing constructs for combining discrete event and continuous formalisms.

Behavioral data base

The role of the behavioral data base has been emphasized by Standridge and Pritsker (1980). Their design of SDL/1 is aimed at integrating a convential data base management system with a state-of-the-art simulation software system. The relational data base can be defined to hold both simulation generated data as well as real system data. Simulation programs can store and retrieve data on line by invoking FORTRAN subprograms, statistics and plots can be produced and operations invoked for statistical analyses and validation purposes. The integrated data base/simulation/statistical software system is thus open to a variety of users as illustrated in Figure 4 extracted from the Standridge article. Pritsker and his co-workers intend to "develop simulation software for conducting an entire simulation study" by basing themselves on the "concepts of SDL/1, of GERT-type networks (Standridge and Pritsker,

1979) and of Ören and Zeigler (1979)". Since this group has been rather successful in infusing simulation practice with its products in the past, there is good reason to believe that a version of a comprehensive simulation support system will be commercially available well within this decade.

Processors
Dekker et. al. (1980) emphasize the role of processors in a support system for advanced simulation methodology. They give an extensive high level design of a hardware-based simulation support system emphasizing parallel architecture. The objective is to achieve easily reconfigurable and rapidly operating simulators of large, spatially disaggregated highly parallel models. As in all parallel schemes, one main problem to overcome is the design of human understandable co-ordination and control schemes.

While the foregoing designs emerge from a simulation orientation, it is also good to gain some perspective from other modelling approaches as the following should show.

Activities
GSPS (Klir, 1979; Cavallo and Klir, 1978) is a software system under design for assisting in the solution of general systems problems. It accepts a fundamental a hierarchical taxonomy of systems descriptions (Klir, 1979; or models in the preceeding terminology (Zeigler, 1976)) which ascend upwards from purely observational to behavioral and increasingly structural levels. Problems, as formulated within this taxonomy are of two kinds:

i) given a particular system, determine another system of a given taxonomical type satisfying given requirements,

ii) given two particular systems, determine some property, satisfying given requirements, of the relation between the two.

GSPS will provide a set of tools appropriate to each problem type and an interface able to suggest the tools appropriate to the activity currently undertaken.

Figure 5 drawn from Klir (1979) illustrates the decomposition of the modelling process which the software is intended to support. In practice, Klir's group has been concentrating on the problem of structure induction in which a space of structures is systematically searched for a candidate able to reconstitute a given behavior. In this case, the behavior is probabilistically described relation and a structure is a set of (smaller) such relations (Uyttenhove, 1978).

Users
Modelling within the context of decision support systems is explicitly considered by Bonczek et. al. (1980) and by Stohr and Tannira (1978). The sense of modelling to be understood here is that of financial planning models used for projections of income, cost, etc. Such systems aim to put the development and use of such models directly into the hands of the manager. Thus they conceive of a knowledge base incorporating the knowledge of operations experts and a data base containing real system data.

Bonczek et. al. (1979) propose that such a knowledge base could consists of computation modules (akin to our model base). A user request for a computation (e.g., sales forecast) would be formulated as a theorem in the predicate calculus and a resolution method would be employed to automatically produce a composition of modules able to answer the request. Judging from the experience in the field of artificial intelligence with such systems however, it is unlikely that realistic requests can be handled without encorporating further methodological principles and/or human interaction.

3. SUPPORT OF THE MULTIFACETTED SYSTEMS MODELLING APPROACH

Advanced simulation methodology has a natural affinity for large scale systems where its development is most urgently needed. We have suggested (Zeigler, 1979a) that the term 'multifacetted' better captures the issues usually involved in large scale modelling and indeed, signifies a more general approach which is not necessarily scale dependent. The multifacetted modelling approach recognizes that a given real system may be approachable from different points of view by different people with different objectives at different times. Computer support must provide facilities for maintenance of a multiplicity of partial models in an environment in which the real system boundaries and its facets of interest are constantly evolving. While such a view is appropriate to large scale systems it is less the scale, than the multiplicities just refered to, which is its distinguishing mark. Among the systems addressable from this point of view are: lake (large scale) and laboratory (small scale) exosystems, digital control systems, business systems, transportation systems, neural networks, cellurar biochemical systems, etc.

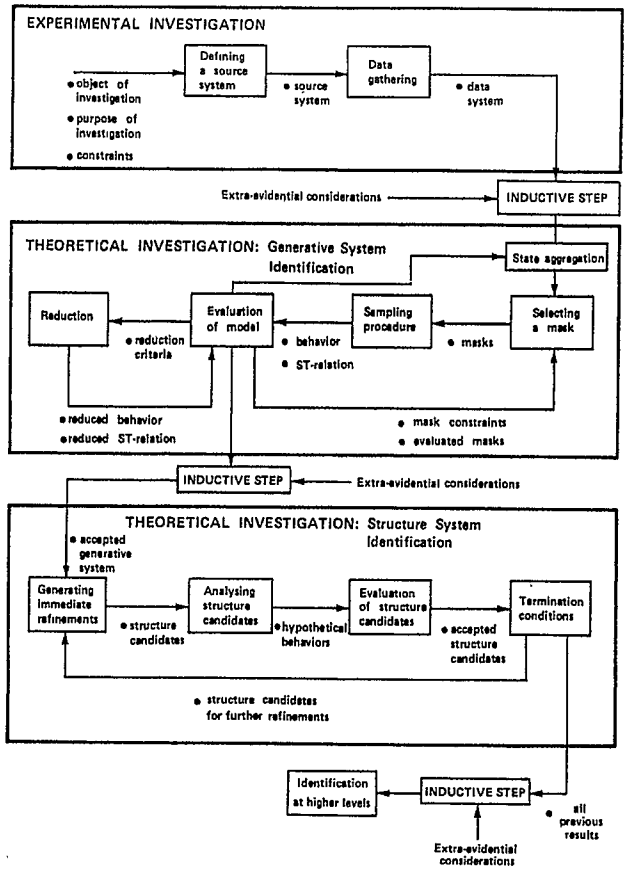

Fig. 5 Klir's Modelling Methodology (from Fig. 1 of Klir (1979)].

3.1 Design Goals for Multifacetted Support Systems

The basic goal in designing multifacetted simulation support systems is to provide a much improved link between the objectives of a user and the models that can be provided to achieve these objectives.

Ultimately the user's interface with the system should be, in the first instance, at the level where he can formulate the questions about the system entities of interest to him. The support system should then produce models from a model base which can answer these questions if such exist or else should assist in synthesizing appropriate models using building block components in the model base. More explicitly,

1. The models retrieved or synthesized should be as computaitonally inexpensive as is permitted by the stated questions so as to admit quick and extensive simulation experimentaiton.

2. The user should be able to flexibly explore the subspace of models of interest by modifications of the questions and model descriptions rather than having to manually reprogram for each modification.

3. The results of such exploration should be stored in a form which facilitates subsequent summary and analysis.

4. The models in the model base should form a consistent whole so that calibration and assessment of new models is assisted by extant models rather than done in isolation as is currently the case.

In order to illustrate the urgency of achieving the preceding goals we shall discuss an area (energy models) where multifacetted issues are paramount. This will be followed by a review of the concepts and software which we have been developing for supporting such an approach.

3.2 Energy Planning Models

In response to the energy crisis a large number of models (in the order of a thousand) have been developed to help governmental and industrial energy planning. Since they were largely developed independently of one another, the set of models does not form a systematic whole and thus is far from a satisfactory basis for decision making. Many of the problems involved were well brought out in the proceedings of a recent meeting (Gass, 1980) under the rubric of 'assessment and validation issues of energy models'. Among the problems discussed are the following:

1. Selection. How can a planner/decision maker know which ones of the many models potentially at his disposal are suited to the particular issue he is being faced with at the time?

   Hudson and Jorgenson (1980) analyse four models "intended to provide a reasonable coverage of methodology and applicability from among the many existing models relating to energy and/or the economy." They find that each model has "different strengths and different areas of applicability" and represent the result in a 'model capability' matrix shown in Table 1 (reproduced from Table 1 of Hudson and Jorgenson (1980); for the model references please see the said paper).

TABLE 1

MODEL - CAPABILITY MATRIX

Application	Model			
	Almon	Pilot	DRI	Hudson-Jorgenson
Economic projections, analyses				
Level of activity, short run			X	
Detailed projections	X		X	
Economic growth				X
Economic structure				X
Energy Projections, analyses				
Demand		X		X
Supply and conversion		X		
Conservation				X
Technologies		X		
Energy-economy analyses				
Short run impacts			X	
Full impacts				X

Table 1 from Hudson-Jorgenson (1980)

An example of error arising from application of a model to a task not in its capability is given in the paper. The authors conclude that the "correct strategy for analysis is not to proceed from the model to the problem but rather the reverse: the problem should dictate the selection."

By inverting the model capability matrix the authors produce a 'problem - model mapping' shown in Table 2 (reproduced from their Table 2) to guide the user in the selection of a model suitable to his problem.

TABLE 2

PROBLEM - MODEL MAPPING

Problem	Model
Disaggregation to find specific economic impacts	Almon
Strategic choice in energy sources and technologies	Pilot
Short-run economic forecasts	DRI
Medium and long run economic analysis	Hudson-Jorgenson
Short run economic impacts of energy	DRI
Full economic impacts of energy	Hudson-Jorgenson

Table 2 from Hudson-Jorgenson (1980)

2. Model Comparison. In general, the problem - model mapping may not be single valued and the decision maker is still faced with a set of models formally capable of dealing with his problem but potentially differing in the reliability with which they do so. Cherry (1980) discusses this difficulty and reports on the evaluation of ten models which forecast electric energy demand. A set of scenarios was developed and each scenario was applied to the subset of models it suited (see Table 3 reproduced from Figure 3 of Cherry (1980)).

   Comparison within such a subset shows models sometimes differing guite widely in their predictions (see Figure 6 reproduced from Cherry's Figure 5). To the extent that models aggree, one may have more confidence in their common prediction. Conversely, areas of disagreement should signal the necessity of further development al well as giving rise to uncertainty and caution as a basis for policy testing.

   Cross model comparison may also be approached at more structural levels. Crissey (1975) suggests a methodology in which models are designed so that "points of contention" (structural issues in dispute) are more easily analysed for their "criticality" (effect on model conclusions).

3. Model Integration. As Jones (1979) has argued, energy models, having been developed under a variety of different methodologies, for a variety of purposes, and from a variety of limited disciplinary views, could well profit from integration into a coherent system. Such a system would facilitate model selection and cross comparison (as discussed above) as well as model composition (the coupling of component models to form a more comprehensive model). Composition and integration may be desirable across many lines including:

   a) methodological (e.g., a linear programming model is coupled to an input-output model (Jones (1979)),

   b) disciplinary (e.g., an energy supply model is coupled to an econometric model),

   c) sectorial (e.g., an electricity demand model is coupled to a natural gas demand model),

   d) spatial (e.g., regional models are coupled to form a national model), and

   e) scope/aggregational (e.g., a detailed sectorial model is coupled to a coarser multisectorial model).

	Model	Scenario 1	2	3	4	5	6	7	8
1.	Comm. Ed.	X	X	X	X	X			
2.	ORNL-REDM	X	X			X	X		
3.	ORNL-SLED	X	X			X			
4.	TVA	X	X	X	X	X	X	X	
5.	CPC	X	X			X	X	X	
6.	FPL	X					X		
7.	NU	X					X		X
8.	Baughman-Joskow	X				X			
9.	WEPCO	X			X				
10.	GPU	X							

Scenarios: 1. Reference Case 2. Price of Electricity (avg. price increase) 3. Price of Electricity (demand charge increase) 4. Price of Electricity (energy charge increase) 5. Competing Fuels Price 6. Appliance Efficiency Standards 7. Technological Change Cogeneration 8. Time-of-Day Pricing.

Table 3. Applicability of scenarios to models (from Table 3 of Cherry (1980)).

An extensive example of the last type is discussed by Hopkins and Rubin (1980) in which a detailed residential energy model is incorporated into a national energy forecasting model. To obviate excessive computation time and cost, the residential model had to be "cartooned" i.e., simplified to equivalent summary form. Moreover, differences in variable definitions, such as whether a certain energy use is to be counted as commercial or residential, had to reconciled. Suffice it to say that integration of existing models may involve difficult simplification and reconciliation problems. The task is notably more straight forward when a collection of models is built upon a uniform multifacetted methodology (see Zeigler (1979a, b) for examples).

3.3 Approach and Concepts

Our approach to the design of such systems is to develop general and universally applicable concepts and software tools. Such tools must be portable (easily installed in different computer systems) and readily specializable in particular contexts. To the extent that our concepts have captured the invariants of the various activities involved we can expect that our tools will achieve the requirements stated earlier. Beginning with a fundamental theory of modelling and simulation (Zeigler, 1976), a set of concepts have been elaborated (Zeigler, 1978; 1979b) which we believe to be fundamental to the design of software support systems for multifacetted modelling. Zeigler (1979a) presented a set of structuring principles on which a concrete design can be based. Figure 7 summarizes the concepts which were suggested. Space does not permit an extensive dicussion of the concepts but we shall attempt to breifly relate them to the energy modelling situation described above (see also Zeigler (1980) for a review).

1. ENTITIES and their organization are intended to capture subsystems and system boundaries e.g., counties, states, nations, etc.

2. VARIABLES which are associated with entities have values which may change over time, e.g., demands, supplies, etc.

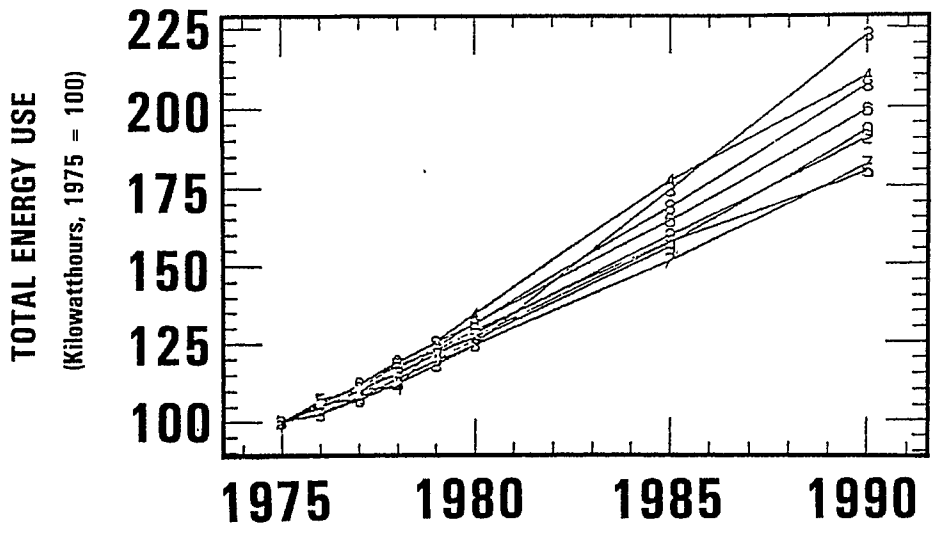

Fig. 6 Forecasted energy use in Scenario 1 (from Fig. 5 of Cherry (1980)).

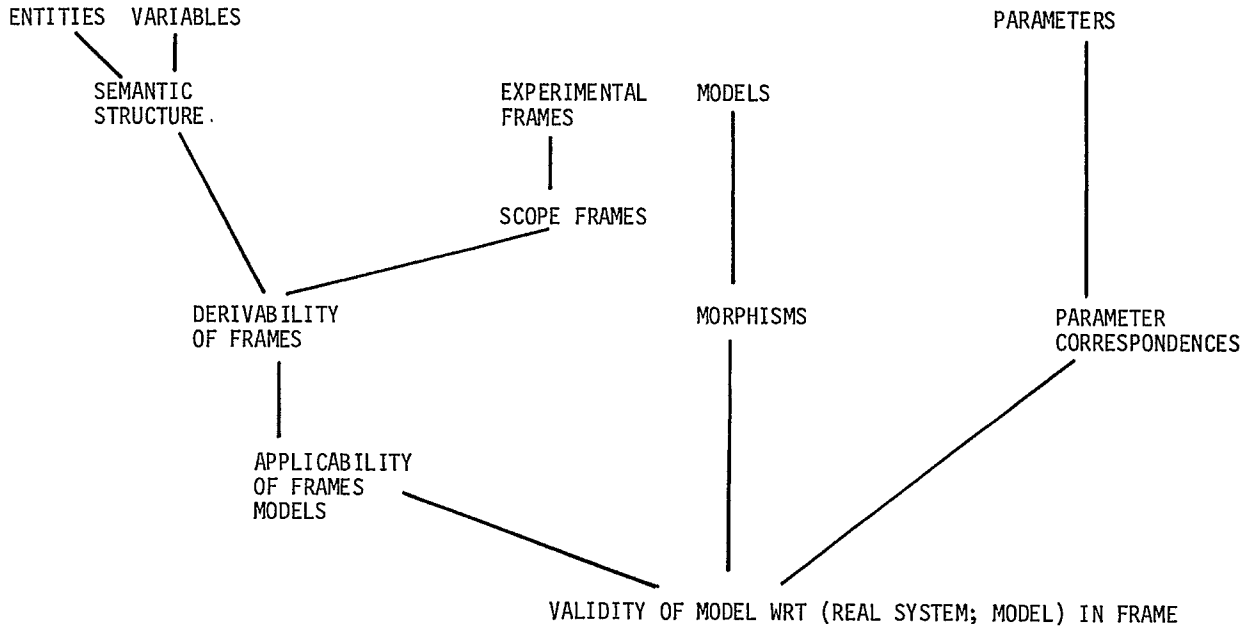

Fig. 7 Concepts for Multifacetted Systems Support.

3. SEMANTIC STRUCTURE represents the interrelations of the variables arising from their definitions. In the integration of models, the entities, variables and expecially, the semantic structures must be reconciled.

4. EXPERIMENTAL FRAMES, as indicated earlier, formalize the behavioral data space suited to a set of questions.

5. SCOPE FRAME of a model is the frame representing the data space within which the model operates i.e., it represents the most extensive set of questions the model is suitable for.

6. DERIVABILITY OF FRAMES is a relation on frames which partially orders them according to the extent of the data they demand (roughly, according to the level of detail demanded by the questions).

7. APPLICABILITY OF FRAMES TO MODELS is a relation holding between a frame and a model when the frame is derivable from the scope frame of the model. Informally, a frame applies to a model if the questions it represents can be answered by the model. Thus the applicability relation is a formal version of the capability matrix illustrated in Table 1.

8. VALIDITY, as indicated earlier, is a relation involving an experimental frame and either a pair of models (relevant to the simplification or "cartooning" of component models before coupling) or a model and a real system (relevant to the acceptability assessment of a model).

9. MODELS, an open-ended set of models i.e., system descriptions at various levels in the behavior-structure hierarchy, for a particular real system.

10. MORPHISMS are preservation relations between models which provide a partial order mirroring that of derivability.

11. PARAMETERS are variables in a model specification which must be determined in order to select a particular model from a class; after a calibration process parameters have empirical content.

12. PARAMETER CORRESPONDENCES are relations between parameter sets of pairs of models which accompany morphisms between the models; such correspondences enable propogation of empirical content from a model to its morphic relatives and enable the MODELS to be assessed as a whole rather than as individuals.

3.4 A System Structuring and Documentation Tool

The above framework serves to guide our development of software tools for modelling and simulation (Hopfeld, 1978; Zeigler, 1980). Since the entity base constitutes the core of the envisioned arhcitecture, we chose to concentrate upon its development. An interactive tool called ESP - Entity Structuring Program - was completed. Operating in a conversational manner, a modeller, or team of modellers, can conveniently set up, modify, interogate, and document system entity structures. The program allows one to associate a multiplicity of decompositions, called aspects, with any entity. This enables one to accomodate models based on different decompositions of the same real system. The program enforces a set of axioms, which are straightforward and once understood provide a powerful means of representing the underlying static structure of a model or family of models. Although intended as a basis for further development of multifacetted system architecture, ESP also has utility as a stand-alone tool for assisting top-down model structuring and documentation. The ESP philosophy is consistent with a number of program and model development methodologies curently being advanced (c.f. the "Conical Methodology" referenced in Nance (1980)).

System entity structure axioms

We shall now briefly review the axioms behind ESP (which was developed as part of a doctoral dissertation by Belogus) and illustrate its application. The command structure of ESP is described by Zeigler et. al. (1980) and its computer science aspects by Belogus (1980).

It is desirable to specify an entity structure in a representation independent way. This method of specifying data structures is called an Abstract Data Type, and can be used both for program specification as well as in modern programming languages (Liskov and Zilles, 1974).

An ENTITY is an abstract object which refers to a component of a real system being modelled, and has both variables and substructure, i.e. further decomposition. Each ASPECT of an entity selects one decomposition and one set of variable types for the entity. An item (entity or aspect) may occur more than once in the entity structure. Each such occurrence is called an item occurrence (ITOC) of that item.

1. The Uniformity axiom requires that all such ITOC's of the same item have the same variable types, mode (entity or aspect) and substructure (further decomposition).

2. The Alternating Mode (Duality) axiom requires that entities have aspects as immediate subordinates while the reverse is true for aspects.

3. The Strict Hierarchy (Finite Descendents) axiom prohibit any item from having an occurrence of itself in its substructure.

Illustrating some Features of ESP

To illustrate some of the features of ESP we shall discuss two examples.

Figure 8 displays an entity structure which characterizes a GPSS like simulator. This simulator is decomposed via the STANDARD_DECOMPOSITION aspect into an EXECUTIVE which directs the simulation, the TRANSACTIONS and the BLOCKS they traverse, and the equipment such as FACILITIES, STORAGES, etc. whose states are altered by the transactions.

```
1 :  ENTITY  GPSS      GENERAL_PURPOSE_SIMULATION_SYSTEM
  2 :  ASPECT  STANDARD_DECOMPOSITION   STND.DEC
    3 :  ENTITY  EXECUTIVE          EXEC
         VAR:       NUMBER
         VAR:       CEC       CURRENT_EVENTS_CHAIN
         VAR:       FEC       FUTURE_EVENTS_CHAIN
         VAR:       CLOCK
         VAR:       XACTS_REMAINING
    3 :  ENTITY  TRANSACTIONS      XACTS
         VAR:       NUMBER
      4 :  ASPECT  XACTS.DEC
        5 :  ENTITY  TRANSACTION       XACT
             VAR:       NAME.     IDENTIFIER
             VAR:       LOCATION
             VAR:       DEPARTURE_TIME
          6 :  ASPECT  XACT.PARAMETERS
            7 :  ENTITY  REGISTERS
                 VAR:       NUMBER   NUM
              8 :  ASPECT  REGISTERS.DEC
                9 :  ENTITY  REGISTER REG
                     VAR:       NAME.     IDENTIFIER
                     VAR:       RANGE.
    3 :  ENTITY  BLOCKS   BKS
                 .
                 .
                 .
    3 :  ENTITY  STORAGES STORES
         VAR:       NUMBER
      4 :  ASPECT  STORES.DEC.
        5 :  ENTITY  STORAGE  STORE
             VAR:       CAPACITY
             VAR:       CONTENTS
             VAR:       REMAINDER
             VAR:       EMPTY
             VAR:       FULL
                .
                .
                .
    3 :  ENTITY  REGISTERS
         VAR:       NUMBER   NUM
       ------- SUBSTRUCTURE APPEARED PREVIOUSLY
                 .
                 .
                 .
```

Fig. 8 An entity structure for the GPSS simulation language

One form of documentation for an entity structure is shown in Figure 9. This is part of the file produced by ESP under the SAVE command from which the entity structure can be reconstructed. Note that the variables attached to an item may be quite fully characterized.

```
                1:  GPSS

  ENTITY
                    SYNONYM (S) :
  GENERAL_PURPOSE_SIMULATION_SYSTEM
            SUB ITEMS ARE:
  STANDARD_DECOMPOSITION
           WITHOUT VARIABLES
================================================================================
              2 : STANDARD_DECOMPOSITION
  ASPECT
                    SYNONYM (S):
  STND.DEC
            SUB ITEMS ARE:
  EXECUTIVE
  TRANSACTIONS
  BLOCKS
  FACILITIES
  QUEUES
  STORAGES
  REGISTERS
  RANDOM_VARIABLES
  FUNCTIONS
           WITHOUT VARIABLES
================================================================================
              3 : EXECUTIVE
  ENTITY
                    SYNONYM (S) :
  EXEC
           WITHOUT DECOMPOSITIONS
           VARIABLE:
  FEC      FUTURE_EVENTS_CHAIN
            RANGE OF VARIABLE:
      CEC.RANGE
            MEANING OF VARIABLES:
      FUTURE EVENTS LIST:EACH TRIPLE IS AN XACT,ITS LOCATION AND DEPART.T
            PHYS. DIMENSION OF VARIABLE:
      NONE.
            UNITS OF VARIABLE:
      NONE.

          VARIABLE :
  CLOCK
            RANGE OF VARIABLE:
      NON NEGATIVE INTEGERS
            MEANING OF VARIABLE:
      THE SIMULATION CLOCK TIME
            PHYS. DIMENSION OF VARIABLES:
      ARBITRARY TIME UNIT
            UNITS OF VARIABLE:
      NONE.
```

Fig. 9 Document Produced by the ESP save command

Let us see how the uniformity axiom lets us easily represent the fact that REGISTERS may be global, i.e. attached to the GPSS system as a whole, or local, i.e., associated with particular transactions. In either case a REGISTER (generalization of SAVEVALUE) has the same structure. This is evident in the display where REGISTERS appears first at level 7 and then at level 3. By the uniformity axiom the second occurrence has the same substructure and variable types as the first. In its occurrence at level 3, REGISTERS is a subentity of TRANSACTIONS.

This example will illustrate how very natural naming conventions can be used to automatically generate variable names appropriate to the entity occurrence. Its singular form REGISTER is given the variables NAME and RANGE. This means that every XACT can have zero, one or more of its own (local) REGISTERS. Each such occurrence of REGISTERS can be distinguished by the label REGISTERS.in. XACT j and each occurrence of REGISTER gives rise to variables such as (REGISTER i.in.XACT j).NAME and

(REGISTER i.in.XACT j).RANGE. The global occurrence of REGISTER (at level 3) in contrast has variables such as REGISTER i.NAME and REGISTER i.RANGE.

Figure 10 a) displays an entity structure for models of bus system planning and management. Figures 10 b) and 10 c) show various substructures of the ES. The intention here is to illustrate how the process of extending system boundaries is reflected in ESP operations.

```
1 :  ENTITY  BUS_SYSTEM
  2 :  ASPECT  TRANSPORTATION_ASPECT
    3 :  ENTITY  BUSSES
      4 :  ASPECT  BUSSES_DECOMPOSITION
        5 :  ENTITY  BUS
          6 :  ASPECT  PROCESS_DECOMPOSITION
            7 :  ENTITY  LOADING_PROCESS
            7 :  ENTITY  UNLOADING_PROCESS
    3 :  ENTITY  STATIONS
      4 :  ASPECT  STATIONS_DECOMPOSITION
        5 :  ENTITY  STATION
          6 :  ASPECT  STATION_DECOMPOSITION
            7 :  ENTITY  ENTRANCE
            7 :  ENTITY  WAIT_AREA
            7 :  ENTITY  EXIT
    3 :  ENTITY  PERSONS
      4 :  ASPECT  PERSONS_DECOMPOSITION
        5 :  ENTITY  PERSON
```

a)

```
1 :  ENTITY  BUS
  2 :  ASPECT  PROCESS_DECOMPOSITION
    3 :  ENTITY  LOADING_PROCESS
    3 :  ENTITY  UNLOADING_PROCESS
```

b)

```
1 :  ENTITY  STATION
  2 :  ASPECT  STATION_DECOMPOSITION
    3 :  ENTITY  ENTRANCE
    3 :  ENTITY  WAIT_AREA
    3 :  ENTITY  EXIT
```

c)

Figure 10. Various substructures of the bus system entity structure.

Suppose a modelling team was split into subteams which were assigned the tasks of modelling a BUS and a STATION respectively. The structures of Figure 10 b) and c) might then result. To obtain models for the BUS_SYSTEM, models for BUSSES and STATIONS are to be employed as components. Such a composition process is mirrored in ESP by amalgamating ES's to form a larger ES such as that in Figure 10 a).

As hinted previously Figure 1 and 2 are also examples of ESP output. Indeed, we can now adopt the view that Figures 1 and 2 represent decompositions of the modelling and simulation process into its component activities and its computational subsystems, respectively. The two aspects are amalgamated into a single entity structure as shown in Figure 11. The reader is invited to test his understanding of the entity structure axioms by examining closely Figures 1, 2 and 11.

ESP: Its place in the larger picture

In summary, ESP is a conversational program for constructing and manipulating entity structures. ESP builds structures which adhere to the uniformity, alternating mode and strict hierarchy axioms. The user needs to understand these axioms and the basic commands of ESP but need know nothing further to build syntactically correct structures. While it is useful in stand-alone form, ESP is intended as the fundamental structuring tool in a multifacetted simulation support system. The entity structure is envisioned as the key principle around which the activities and computational subsystems are organized. Written in PASCAL for portability, it has been successfully installed in a number of institutions and is available from the author.

```
1 :  ENTITY MODELLING_AND_SIMULATION_PROCESS
  2 :  ASPECT ARCHITECTURAL_DECOMPOSITION
    3 :  ENTITY USERS
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY INTERFACES
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY ACTIVITIES
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY PROCESSORS
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY BASES
              -- DEPTH  3 TRUNCATION


  2 :  ASPECT ACTIVITIES_DECOMPOSITION
    3 :  ENTITY MODEL_GENERATION/REFERENCING
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY MODEL_PROCESSING
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY BEHAVIOR PROCESSING
              -- DEPTH  3 TRUNCATION
    3 :  ENTITY REAL_SYSTEM_EXPERIMENTATION

    3 :  ENTITY ACCEPTABILITY ASSESSMENT
              -- DEPTH  3 TRUNCATION
```

Figure 11. Entity structure for decomposition of MOdelling and Simulation Process

4. CONCLUSIONS

We have surveyed some of the concepts and software emerging in the direction of advanced simulation methodology. While much of the state of the art is in the research rather than the production stage, both the natural evolution of simulation software as well the urgent needs of societal problems point to a rapid uptake of comprehensive and integrated support systems.

Indeed, since they aim for comprehensiveness and integration, we can expect such systems to tax the outer limits of contemporary computer resources and to exhibit various generality/efficiency compromises. Clearly, performance evaluation will become a major issue (at the end of the decade?) and perhaps now is the time to begin the establishment of standards and criteria.

The vistas are wide open!

REFERENCES

Beauchamp, J.N., and Field, R.C. (1979), "Simulation Modelling by Stepwise Refinement," In: Proceedings of the Winter Simulation Conference, San Diego, CA.

Belogus, D. (1980), Concepts and Software Tools for Multifacetted System Simulation, Internal Report, Weizmann Inst., Rehovot, Israel.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. (1979),"The integration of network data base management and problem resolution," Infor. Syst. 4, 143-154.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B. (1980),"The evolving roles of models within decision support systems,"Decision Sciences," 11.

Cavallo, R., and Klir, G.J. (1978), "A conceptual foundation for systems problem solving," Int. J. Sys. Science, 9, 2.

Cellier, F.E. (1979), Combined Continuous/Discrete system simulation by use of digital computers: techniques and tools, Doct. Dissertation, ETH, Zurich, Switzerland.

Cherry, B.H. (1980), "Electric load forecasting: probing the issues with models," In: (Gass, 1980).

Crissey, Brian L. (1975), A rational framework for the use of computer simulation models in a policy contex, Doctoral Dissertation, John Hopkins University.

Cutler, M.M. (1980), A formal program model for discrete event simulation and its use in the verification and validation of system models and implementations, Doct. Dissertation, UCLA, Los Angeles, CA.

Dekker, L., et. al. (in press), Parallel Simulation Systems, Proceedings of the IMACS Simulation of Systems Conference held in Sorrento, Italy, 1979.

Dykstra, E.W. (1976), A Discipline of Programming, Prentice Hall, N.J.

Elzas, M.S. (1979), "What is needed for robust simulation?," in (Zeigler et. al. 1979).

Estrin G. (1978), "A method for design of digital systems supported by SARA at the age of one," AFIPS Conference Proceedings, NCC.

Gass, S.I. (1980), Validation and assessment issues of energy models, NBS special publ. 569.

Hopfeld, A. (1978), A software package for checking the existence of homomorphic relations between models, Masters thesis, Weizmann Inst., Rehovot, Israel.

Hopkins, F. and Rubin, L. (1980), "Validating the HIrst residential energy use/ Mid-range energy forecasting system interface," In: (Gass, 1980).

Hudson, E.A. and Jorgenson, D.W. (1980), "Assessment and selection of models for energy and economic analysis," In: (Gass, 1980).

Jones, W.T. (1979), "Integrated modelling systems: application to energy systems and considerations of software engineering," In: (Zeigler et. al., 1979).

Klir, G.J. (1979), "General systems problem solving methodology," in (Zeigler et. al, 1979).

Liskov, B. and S. Zilles (1974), "Programming with abstract data types," ACM SIGPLAN Notices, 9 1974.

Nance, R.E. (1980), The time and state relationships in simulation modelling, Technical Report, Combat Systems Dept., Dahlgren VA 22448.

Nijssen, G.M. (ed.) (1977), Architecture and Models in Data Base Management Systems, North Holland Pub. Co., Amsterdam.

Ören, T.I. (1979), "Concepts for advanced computer assisted modelling," In: (Zeigler et. al., 1979).

Ören, T.I. (1980), "Computer-aided modelling systems," In: Proceedings of Simulation '80, Interlaken, Switzerland.

Ören, T.I. and Zeigler, B.P. (1979), "Concepts for advanced simulation methodologies," Simulation, 32, 3.

Pritsker, A.A.B. (1979), "Compilation of definitions of simulation," Simulation, 33, 2.

Ryan, K.T. (1979), "Software Engineering and Simulation," In: Proceedings of the Winter Simulation Conference, San Diego, CA.

Rzevski, G. (1980), "Systematic design of simulation software," In: Proceedings of Simulation '80, Interlaken, Switzerland.

Sanguinetti, J. (1979) "A technique for integrating simulation and systems design, in: Conference on Simulation, Measurement and Modeling of Computer Systems," Sigmetrics/Simuletter, Fall.

Sierenberg, R. and O. de Gans, PROSIM78 Textbook, Delft Inst. of Technology, (in prep.).

Standridge, C.R., and A.A.B. Pritsker (1980), "Using Data Base Capabilities in Simulation," In: Proceedings of Simulation '80, Interlaken, Switzerland.

Stohr E.A. and Tannira, M. (1978), "The design of a corporate planning system simulator," Proc. of Winter Simulation Conf. p. 919-930, (USA).

Teichrow,D. and E. Hershey (1978), "PSL/PSA: A Computer-aided Technique for Structure Documentation and Analysis of Information Processing Systems," IEEE Trans. Soft. Eng. 3/1.

Uyttenhove, H.J.J. (1978), Computer-aided systems modelling: an assemblage of methodological tools for systems problem solving, Doct. Dissertation, SUNY, Binghamton, N.Y.

Yeh, R. (ed.) (1977), Current trends in programming methodology, volume 1: Software specification and design, Prentice Hall, N.J.

Zeigler, B.P. (1976), Theory of modelling and simulation, Wiley, N.Y.

Zeigler, B.P. (1978), "Structuring the organization of partial models," Int. J. General Systems, 4, 1.

Zeigler, B.P. (1979a), "Structuring principles for multifacetted system modelling," in (Zeigler et. al., 1979).

Zeigler, B.P. (1979b), "Multi-level, multi-formalism modelling: an ecosystem example," in Theoretical Systems Ecology (ed: E. Halfon), Academic Press example," in Theoretical Systems Ecology (ed: E. Halfon), Academic Press N.Y.

Zeigler, B.P. (1980), Introduction to GARF - A language for modelling formalisms, Internal Report, Weizmann Inst.

Zeigler, B.P. (in press), "Modelling and Simulation Methodology: State of the Art and Promising Directions," in Proceedings of the IMACS Simulation of Systems Conference held in Sorrento, Italy, Sept. 1979.

Zeigler, B.P., Elzas, M.S., Klir G.J., and Ören, T.I., (1979) Methodology in systems modelling and and simulation, (North Holland, Amsterdam.).