Applied Intelligence 16, 235–247, 2002

Towards Combinatorial Analysis, Adaptation, and Planning of Human-Computer Systems

MARK SH. LEVIN
Department of Mechanical Engineering, Faculty of Engineering, The Ben-Gurion University of the Negev, Beer Sheva, Israel
marksh@bgumail.bgu.ac.il

Abstract. The paper addresses a modular system approach to the analysis, design, and improvement of human-computer systems (HCSs). The approach is based on ordinal expert information and optimization models. A modular description of HCSs (system components and their interconnection), some corresponding requirements to them, and improvement actions are described. The following stages have been examined: design of a basic system morphology, modification of the morphology, analysis, and planning.

Our combinatorial approach (two-level hierarchical morphological design) consists of two problems: (i) multicriteria analysis of primitives (design alternatives), and (ii) combinatorial synthesis. The hierarchical combinatorial synthesis is based on a "design morphology" which corresponds to an initial hierarchical knowledge structure (design alternatives, their estimates, etc.). Ordinal scales for initial information are used.

Two basic numerical examples illustrate the approach: (i) modular analysis, adaptation, and improvement of HCSs; (ii) series planning the user interfaces for knowledge engineering.

Keywords: system design, design & modeling, planning & scheduling, system adaptation, knowledge-based methodologies, knowledge engineering, optimization problems, human-computer systems

1. Introduction

In recent decades, majority of applied problems are very complicated and require some special composite efforts including the following:

I. System approaches to design and planning.
II. Knowledge of domain experts (engineering and design skills of specialists) as a basis of the approaches including corresponding cycle of knowledge technology/engineering (knowledge acquisition, modeling, testing, correction, and utilization).
III. Hybrid methods which involve, for example, traditional optimization techniques, multicriteria decision making technology, and artificial intelligence approaches.

This article addresses and discusses the above-mentioned issues on the basis of hierarchical morphological multicriteria design (HMMD) (a combination of combinatorial optimization techniques, multicriteria analysis, and ordinal information of domain experts). Human-computer systems (HCSs) are used as representative systems for our examination.

In recent years, significance of approaches for complex systems design is increasing. In our opinion, the following methods can be listed as main ones:

1. Formal methods for design [1–4].
2. Optimization on the basis of complex mixed integer non-linear programming (e.g., design/synthesis in chemical engineering, in process systems engineering) [5–7].
3. Non-linear multicriteria (multiobjective) optimization [8–11] including evolutionary multiobjective optimization techniques [12, 13].
4. Multidisciplinary optimization in aerospace and structural engineering [14, 15].

5. Parameter Space Investigation (PSI) for the design of various complex systems and financial planning [9, 15, 16].
6. Various methods of global optimization [5, 17, 18].
7. Hierarchical system design [19–23] and modular system design [24–27] including HMMD [26].
8. Design on the basis of grammar description for composable systems as in software engineering [28].
9. Special artificial intelligence approaches on the basis of expert systems (knowledge based systems), e.g., R1/MICON [29–31] for computer engineering, VLSI design, etc. [1, 32].
10. Hybrid methods [31].

At the same time, it is reasonable to point out some specific design problems which are important for complex composite systems [26]:

1. Basic system design:
   1.1. Design of a system as a wholeness.
   1.2. Synthesis or integration of a composite system.
   1.3. Multistage planning as the design of multistage (series, parallel-series) composite strategy.
2. System improvement, reengineering (revelation of system bottlenecks, generation of improvement actions, scheduling of the actions).
3. Strategic system design (under uncertainty):
   3.1. One-stage strategic system design.
   3.2. Multistage system design.

From this viewpoint, some of the above-mentioned system design methods are oriented only to the design problem 1.1 and it can be very difficult to apply them for other design problems. Mainly, methods of AI, hybrid methods (including HMMD) can be considered as basic ones for more complicated and/or multistage design/planning problems above. In our opinion, many important system properties (e.g., reliability, adaptability, flexibility) are based on special system redesign (reengineering) and strategic design problems which can be implemented as the above-mentioned complex design problems.

In the paper, a modular HCS is examined as a decomposable system consisting of the basic parts (tasks, techniques, information, user). These parts, their interconnection, and possible improvement actions are under our consideration. The synthesis design problem 1.2 is examined in [26]. Here we consider the analysis and adaptation of HCSs (example 1: problems 1.2 and 2) and planning problem which corresponds to 1.3, 3.2, and 2 (example 2).

In recent years, special *User Interface Development Environments* (*UIDE*) and *User Interface Management Systems* (*UIMS*) have been used for the design and management of user interfaces [33–36]. Functional/task analysis and models of human computer interaction are applied as the main approaches to the user interface design [35, 37]. Main human-computer interaction models are the following [35, 37]: (a) Command Language Grammar (CLG); (b) Task Action Language (TAL); (c) Task Analysis for Knowledge Description (TAKD); (d) GOMS approach (Goals, Operations, Methods and Selection Models); and (e) Task Mapping Model (TMM). Mainly, approaches to interface design are based on the analysis, evaluation, and selection of design alternatives [26, 33, 35, 36]. The following techniques are used for the comparison and selection of user interface components (icon, text, direct manipulation, menu, etc): (1) experimental investigation [38]; (2) AI methods as rule bases [33, 36]; and (3) Analytic Hierarchy Process [39]. Object-oriented development approaches are widely used for user interface design too [40].

In our opinion, the main trend consists in an implementation of synthesis approaches. On the other hand, significance of system adaptation and intelligence is increasing. Recently, adaptation of user interfaces is a central problem in human-computer interaction [41–46]. Some authors have been studied adaptation of specific computer systems (e.g., hypermedia, CAD systems, decision support systems) or their components. For example, Brusilovsky examined issues of adaptive hypermedia [47].

In our article, Hierarchical Morphological Multicriteria Design (HMMD) is used as a basic approach to analyze and to redesign (improvement, transformation, adaptation, multistage planning) an initial HCS [26]. Note hierarchical approaches to complex systems have been studied by many authors [35, 48, 49]. Our combinatorial method is based on two-level morphological design [26]. The paper focuses on the following three problems:

(a) combinatorial (modular) description and analysis of HCS;
(b) adaptation of HCS; and

(c) planning on the basis of multicriteria analysis of primitives (design alternatives) and combinatorial synthesis (two-level hierarchical morphological design).

Ordinal scales are used for our evaluation of the design alternatives and their compatibility. As a result, the approach is useful for domain experts from viewpoint of knowledge acquisition and understandability of the solving process.

The hierarchical combinatorial synthesis is based on a "design morphology" which corresponds to an initial hierarchical knowledge structure (design alternatives, their interconnection, etc.). The "design morphology" is a specific new hierarchical representation for domain knowledge. The synthesis approach consists of the following two main phases:

1. Design of the basic "design morphology" including assessment of its design alternatives and their interconnection.
2. Bottom-Up selection and synthesis of the design alternatives to obtain a resultant system (composite solution as a target system, a parallel-series strategy).

In the paper, two numerical examples are presented:

(a) analysis and adaptation of HCS, and
(b) planning an user interface in knowledge engineering (a kind of the design problem for solving strategy).

The first example consists in the modular analysis, synthesis, and improvement of HCS. The second example is oriented to the modular design of a composite series strategy. The planning process (i.e., problem solving) has been examined by well known authors [49–51]. The decomposition of problems is perhaps the most important approach towards complex situations in various domains [50]. For example, problems of composing some models are often examined. In recent decades, various investigations in the field of model management have been conducted [52–55]. These researches address the development and use of model management systems as model bases or libraries for application domains as follows: (a) databases; (b) decision support systems; and (c) expert systems [55]. Building the solving strategies usually is based on traditional optimization models, e.g., linear programing, non-linear programming [24, 55] or artificial intelligence approaches, e.g., hierarchical task network planning [56]. Our planning example corresponds to the multi-stage modular design of human-computer interaction. The problem is:

Build the "best" series scheme on the basis of some standard representation operations (primitives).

Thus we examine designing the series strategies for supporting a dialogue in the development and usage of knowledge base systems.

2. Hierarchical Morphological Approach

HMMD is described in [26] as a fundamental of combinatorial engineering. In this section, we briefly consider a generalized scheme of HMMD and its usage for the combinatorial analysis, transformation/adaptation, and planning.

From viewpoint of knowledge engineering (knowledge acquisition, structuring/modeling, testing, correction, utilization), HMMD realizes a special ordinal domain expert knowledge based technology for the description, design, and improvement of composite systems. Here engineering skills is modelled as a tree-like system structure, design alternatives for the structure nodes, interconnection the design alternatives, and multicriteria description of structure parts above. At the same time, HMMD involves optimization techniques (multicriteria ranking and combinatorial synthesis) and can be considered as a hybrid knowledge & optimization based system.

2.1. Hierarchical Morphological Multicriteria Design

The following basic assumptions are used in HMMD:

(1) the target system is decomposable one, i.e. it has tree-like structure;
(2) an excellence of the system is an aggregation of subsystems qualities, and of qualities of components compatibility;
(3) criteria for design alternatives (DA's) are monotone;
(4) estimates of DA's for model nodes upon criteria can be transformed into priorities, and priority scales of components are coordinated ($r = 1 \ldots k$, 1 corresponds to the best group, here $k \leq 4$); and

(5) interconnection of DA's (Is) is evaluated on ordinal scale $(0 \ldots l; l$ is the best one, 0 corresponds to impossible Is, here $l \leq 5)$.

The generalized scheme of HMMD involves the following: *Top-Down* design of system model; generation of DA's for leaf nodes of the system model; *Bottom-Up* hierarchical selection and composing of composite DA's; analysis of composite DA's. Here DSS COMBI is used for multicriteria ranking [57]. At the same time, some other ranking procedures can be used too, for example, an additive utility function [58] or well-known Analytic Hierarchy Process AHP [39]. The basic problem of composing is to find a composition (composite DA's, morphological scheme)

$$S = S(1) \star \cdots \star S(i) \star \cdots \star S(m)$$

of DA's (one representative from each morphological class, a morphological class is a set of DA's for a component) with non-zero Is, where $S(i)$ is ith system component. Figure 1 depicts an example of our composition problem.

Our basic composition problem is based on the following system excellence for S:

$$N(S) = (w(S); n(S)),$$

where $w(S)$ is the minimum of pairwise compatibility in S,

$$n(S) = (n(1), \ldots, n(r), \ldots, n(k)),$$

where $n(r)$ is the number of components of the rth quality in S. Thus we search for solutions which are nondominated by $N(S)$. In HMMD, we consider vector domination (preference) for solutions on the basis of Pareto-rule as follows [8, 10]:

Definition 1. Point $\hat{x} = (x_1, \ldots, x_i, \ldots, x_n)$ dominates by Pareto-rule point $\hat{y} = (y_1, \ldots, y_i, \ldots y_n)$ $(\hat{x} \succ^P \hat{y})$ if the following two conditions are correct:

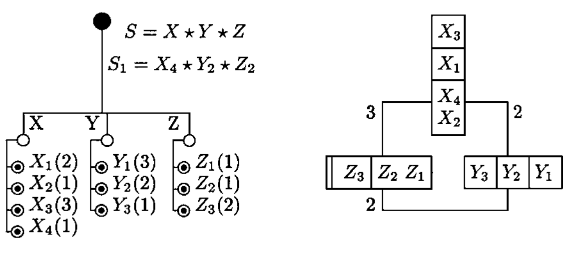

Figure 1. Example of composition problem (priorities of DA's are shown in brackets).

(1) $\forall i = 1, \ldots, n \quad x_i \succeq y_i$ (e.g., $x_i \geq y_i$) and
(2) $\exists i_o \in \{1, \ldots, n\}$ such that $x_{i_o} \succ y_{i_o}$ (e.g., $x_{i_o} > y_{i_o}$).

Let X be a set of points.

Definition 2. Point $\hat{x}_o \in X$ is called Pareto-effective point if there does not exist $\hat{x} \in X$ such that $\hat{x} \succ^P \hat{x}_o$.

Figure 2 illustrates Pareto rule and Pareto-effective points for two-dimensional case.

For composite decision in Fig. 1, we get $N(S_1) = (2; 2, 1, 0)$. Figure 3 depicts an example of the lattice of system quality.

Figure 4 depicts the lattice-like space of system quality (for $(N(S))$ and examples of decisions (Pareto-effective points and ideal point). Note the space consists of three ordered lattices each of them corresponds to the lattice from Fig. 3.

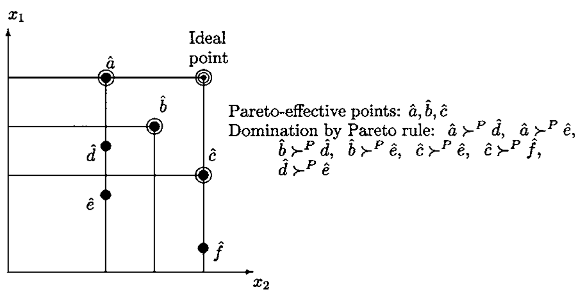

Figure 2. Pareto rule and Pareto-effective points.

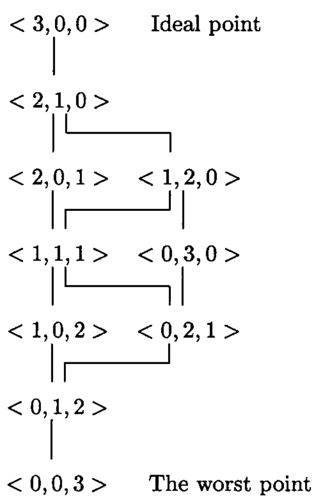

Figure 3. Position (histogram) presentation of the lattice of system quality for $N = (w; n(1), n(2), n(3))$, w =const, $m = 3$, $l = 3$.

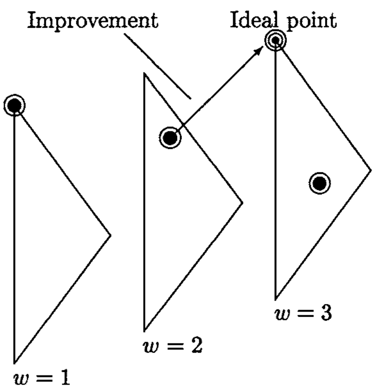

Figure 4. Space of system excellence for $N(S)$.

The following types of elements (DA's, Is) with respect to solution S are used to analyze composite DA's: *S-improving*, *S-neutral*, and *S-aggravating* ones by vector N, where latter elements are considered as 'bottlenecks'. Thus we can generate layers of the system excellence:

(1) ideal solutions (from the best components);
(2) Pareto-effective points by N; and
(3) neighborhood of Pareto-effective DA's (a decision from this set can be transformed by only one improvement step into a Pareto-effective point).

2.2. *Generalized Glance*

A transformation of decomposable systems on the basis of HMMD has been described in [26]. The following ways of the improvement can be investigated:

(1) to improve DA's;
(2) to add new DA's;
(3) to improve Is; and
(4) to improve system structure (i.e., model).

Some conceptual improvement actions for human-computer systems will be presented later (see Section 3). Thus one can build a change system (i.e., an ordered set of improvement actions).

Generally, it is reasonable to analyze multi-trajectory transformation of HCS on the basis of a hierarchical change system, to design composite actions, and to plan the improvement process. Close planning problems have been studied in hierarchical task network planning [56] and in linguistic geometry [59]. Figure 5 depicts a generalized glance to objects and corresponding problems.

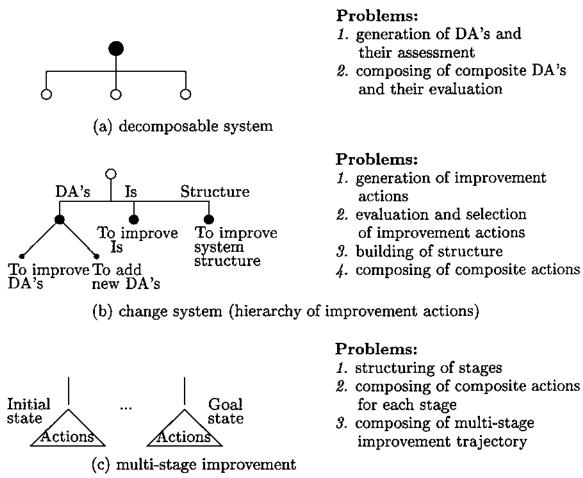

Figure 5. Generalized glance: objects and problems.

3. Human-Computer System

3.1. *Structure, Requirements, Improvements*

Here we assume that HCS consists of four main parts as follows [35, 42, 57, 60]:

(a) goals (tasks);
(b) operational part;
(c) factual part (information as data and/or knowledge); and
(d) human (user's) part.

There exist various classifications of basic tasks and user's activities. Unfortunately, the tasks and user's activities are interconnected, and it is reasonable to examine them together. From the viewpoint of creativity Altshuller proposed several linear ordered levels of creative activities as follows [61]: selection, modification, design, design of a new composition. In addition, he differs types of objects as follows: data, idea, construction or algorithm, version of implementation. As a result, we obtain a 2-dimensional Cartesian space:

$$\text{kinds of creative actions} \times \text{types of objects}.$$

Norman has been studied four stages of user's activities: (1) forming the intention; (2) selection of an action; (3) execution of the action; and (4) analysis of outcome [62]. Dix has investigated similar information life cycle [63]. Rouse has considers several kinds of behaviors [64]: (1) information seeking; (2) information processing; (3) meta behaviors, and (4) products of creativity.

Many authors have been examined specific roles in a team work (manager, coordinator, etc.) [22, 65, 66]. Cross has been analyzed styles of activities in a design process (convergent, divergent, etc.) [67]. Table 1 demonstrates some examples of activities. Here we do not consider collaborative tasks (conference, co-working, etc.) [68].

Thus let us assume the following list of basic kinds of tasks/activities:

(1) routine activities (e.g., input/output of data);
(2) basic activities as a simple analysis and decision making: observation (analysis as comparison); control: comparison/analysis and selecting a basic activity; and
(3) creative activities: (a) diagnosis of a situation; (b) comparison/analysis and selecting the activities; (c) planning and generation of a composite activity; (d) implementation of decisions; and (e) analysis in on-line mode.

Usually an analysis of users is based on the following taxonomy: (i) novice; (ii) trained user; and (iii) expert. Some investigators have been considered levels of user's knowledge and experience on components of applied computer technology [57, 69]:

Table 1. Examples of basic activities.

Activity	Element	Type of presentation	Person & example of application
1. Edition (operations of data transformation)	(a) Data (b) Modes	(a) Text (b) Table (c) Menu, icon (d) Flowchart	Operator; database
2. Supervision (comparison with standard situation)	Control parameters of data processing	(a) Text (scrolling) (b) Icon (c) Flowchart (d) Animation	Process operator; real-time control
3. Analysis (comparison, matching, clustering, etc.)	(a) Data (e.g., partitions)	(a) Histogram (b) Bar charts (c) Star charts (d) Graphics (e) Animation	Analyst; analysis
4. Design (construction, creation)	(a) Product (b) Strategy (c) Plan	(a) Text (b) Data (c) Structure (a) Graphics (b) Animation	Designer; creative design, planning, etc.

(a) preobjective, preoperational; (b) concrete objects and operations or specific detailed knowledge; and (c) abstract objects and operations or global, general knowledge.

Generally, the levels above correspond to levels of Piaget [70]. Gardner has examined 7 types of intelligence [71]: (i) logical-mathematical; (ii) vision-spatial; (iii) musical; (iv) linguistic; (v) bodily kinesthetic; (vi) intrapersonal; and (vii) interpersonal. Clearly, in the case of an user team, it is necessary to examine more complex cases.

The factual part can be considered, for example, from the following viewpoints: (1) volume (small, middle, large); (2) correspondence to tasks; (3) quality of presentation; (4) complexity of availability (required time, complexity of requests, etc.); and (5) needs of additional processing.

Finally, we point out some basic requirements to HCS components, and their interconnection:

I. COMPONENTS:

1.1. Tasks:

Requirements: understandabity, simpleness, certainty.

Improvements (improvement of DA's): to reduce, to improve a description, to define more precisely.

1.2. Operations:

Requirements: adequacy, easy to use.

Improvements:

(a) improvement of DA's: to describe, to test, to modify;

(b) addition of new DA's: to compose aggregate/composite operations, to design new operations.

1.3. Information:

Requirements: sufficiency, certainty, easy to search for and/or to acquire, easy to manipulate, quality of representation.

Improvements:

(a) improvement of DA's: to improve presentation, process;

(b) addition of new DA's: to search for, to process, to design new information.

1.4. User:

Requirements: experience, creativity, learnability, system thinking.

Improvements:

(a) improvement of DA's: to train user;

(b) addition of new DA's: to add trained user or expert, to combine a team.

II. INTERCONNECTION:

2.1. Task-Operations:

Requirements: correspondence, minimum of modification of operations.

Improvements (interconnection): to solve test tasks.

2.2. Task-Information:

Requirements: correspondence, minimum of information processing, minimum of solving iterations.

Improvements (interconnection): to solve test tasks.

2.3. Task-User:

Requirements: experience of user, understandability.

Improvements (interconnection): to train user, to improve a description of a task, to improve task presentation, to improve user interface.

2.4. Operations-Information:

Requirements: correspondence (by format, by type), required resources, etc.

Improvements (interconnection): to add helper, to solve test tasks, to improve user interface.

2.5. Operations-User:

Requirements: experience of user, easy to use, easy to learn.

Improvements (interconnection): to add helper, to solve test tasks, to improve user interface.

2.6. Information-User:

Requirements: experience of user, easy to use, quality of representation, understandability.

Improvements (interconnection): to add helper, to improve data presentation, to solve test tasks.

The requirements to HCS involve effectiveness, cost, usability, adaptability, etc. [35, 40].

3.2. Scheme of Analysis and Adaptation

In recent years, many authors have been investigated adaptation of user interfaces [41, 42, 45, 46]. Moreover, Malinowski et al. have compared two approaches: adaptive user interface and intelligent interface [44]. In this paper, we describe our attempt to examine adaptation not only for the user interface, but for all HCS components and their interconnection. Our scheme is depicted in Fig. 6.

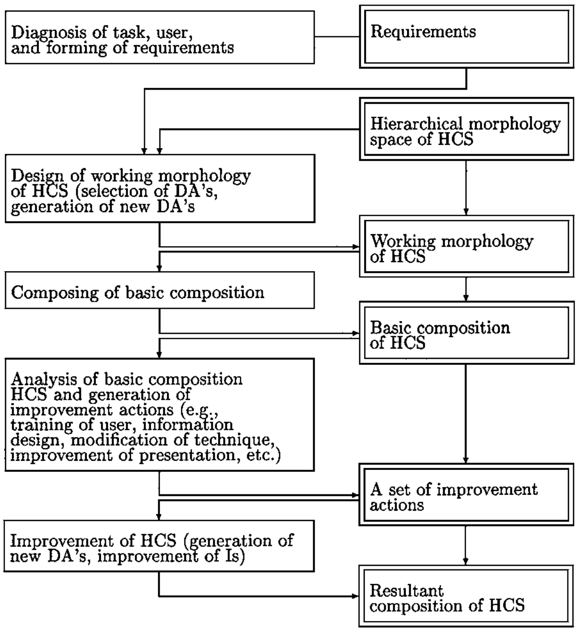

Figure 6. Scheme for analysis and adaptation of HCS.

4. Example 1: Analysis, Adaptation, Improvement

The basic morphology is depicted in Fig. 7. Basic DA's are presented in Table 2. A certain task is considered as an input, for example, T_2 (we reject T_1 and T_3). In addition, we add aggregate DA's as follows: $U_5 = U_1$ & U_2, $U_6 = U_2$ & U_3, $O_4 = O_1$ & O_3, $I_5 = I_3$ & I_4.

An example of compatibility is presented in Table 3.

Resultant composite DA's are the following (ideal point $N = (4; 4, 0, 0)$, Fig. 8):

1. $N = (3; 4, 0, 0) : S_1 = T_2 \star I_3 \star O_4 \star U_2$ and $S_2 = T_2 \star I_5 \star O_4 \star U_2$;
2. $N = (4; 2, 1, 1) : S_1 = T_2 \star I_5 \star O_3 \star U_3$.

Now we can consider an improvement process. For an improvement of S_1 and S_2 it is necessary to increase

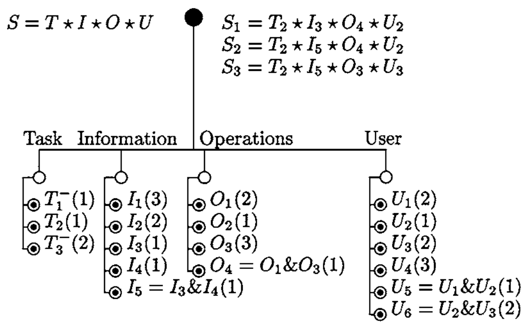

Figure 7. Structure of HCS (hypothetical priorities of DA's are shown in brackets).

Table 2. Basic DA's.

DA's	Description
I_1	Basic information
I_2	Results of an additional search
I_3	Results of an information design
I_4	Expert information
O_1	Basic techniques
O_2	Additional selected techniques
O_3	New designed techniques
U_1	Novice
U_2	Trained user
U_3	Specialist
U_4	Expert

Table 3. Compatibility.

	I_1	I_2	I_3	I_4	I_5	O_1	O_2	O_3	O_4	U_1	U_2	U_3	U_4	U_5	U_6
T_2	4	3	3	2	4	1	2	4	3	1	3	4	4	3	3
I_1						2	3	3	3	2	1	2	1	2	2
I_2						2	2	2	3	1	3	3	1	3	3
I_3						2	3	3	3	3	3	3	3	3	3
I_4						0	2	3	3	0	1	2	3	1	2
I_5						2	3	4	3	3	3	4	3	3	3
O_1										3	2	3	1	2	3
O_2										2	2	3	3	2	3
O_3										1	2	4	3	2	3
O_4										2	3	3	3	2	3

compatibility between all their elements. Let us examine an improvement for S_3. Evidently, elements O_3 and U_3 are bottlenecks in S_3. The following three possible actions are possible:

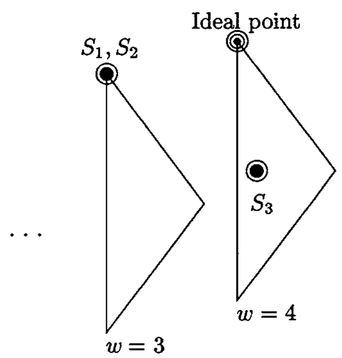

Figure 8. Space of system excellence and Pareto-effective points.

(a) improvement of priority for O_3 $(3 \Rightarrow 1)$ (for example, by development of a special helper for the methods), result: $(S_3^a, N = (4; 2, 2, 0)$;
(b) improvement of priority for U_3 $(2 \Rightarrow 1)$ (for example, by additional training of specialist), result $S_3^b, N = (4; 3, 0, 1)$; and
(c) composite improvement $((O_3, 3 \Rightarrow 1)$ & $(U_3, 2 \Rightarrow 1))$, result: $S_3^c, N = (4; 4, 0, 0)$, the ideal point.

5. Example 2: Planning an User Interface

Here the following basic components are considered:

(i) operations of data processing,
(ii) operations of knowledge acquisition or transformation, and
(iii) training of user and data/knowledge representation.

The design framework consists of the following stages:

1. Forming a basic hierarchical morphological space of operations (HMSO).
2. Analysis of an initial situation (user, task) and adaptation of HMSO (design of a working version of HMSO) as follows:
   (a) selection or identification of appropriate operations on the basis of constraints;
   (b) parallelization of operations on the basis of parametrization of techniques, and use of different experts; and
   (c) training the user.
3. Design of a composite solving strategy including the following two phases:
   (a) selection of operations (multicriteria ranking);
   (b) synthesis on the basis of steps as follows: composing a series strategy or constructing an operation chain (morphological clique).

5.1. Solving Scheme

There exist sources of a user interface adaptation as follows:

(a) user;
(b) task; and
(c) intermediate results of problem solving (information for feedback).

Our viewpoint to adapted objects, basic operations, and the adaptation processes is described in Section 3. Note studies of the user modeling and adaptation are considered in detail in [42, 72]. In the main, the following techniques have been applied for the model selection, and sequencing (synthesis of series strategies):

(a) artificial intelligence techniques [53, 73, 74];
(b) linear programming problems [55];
(c) integer programming problems as searching for an optimal path to a required output [54]; and
(d) nonlinear integer programming for the modular design with redundancy, i.e., with parallel fragments [24].

Here we examine conceptual design, and combinatorial models for the selection, modification, and composing of the solving strategies and their elements. The generation of HMSO is based on the conceptual analysis and design [75, 76].

5.2. *Example*

Our example is based on the examination of a system for the knowledge acquisition and use. We examine HMSO as follows:

1. Prototyping P:
   - 1.1 Problem identification I.
   - 1.2 Selection/generation of terminology T.
   - 1.3 Knowledge acquisition V.
2. Design of work version D.
   - 2.1 Enhancement of problem description (type, terminology) E.
   - 2.2 Acquisition of additional knowledge W.
3. Utilization U.

Generally, HMSO is based on the following kinds of operations: (1) knowledge acquisition (interaction); (2) data/knowledge processing (computation); (3) data/knowledge representation (interaction); and (4) training of user (interaction).

The following criteria for the evaluation of DA's are used [26, 40, 54]: required computer resources; required human resources; quality of ranking (robustness, etc.) possibility for data representation; possibility for an analysis of intermediate data; usability (easy to learn, understanding, acceptability, habits, etc.).

In our example, we apply representation elements or design alternatives (DA's) for leaf nodes as follows: command language (X_1); menu (X_2); icons (X_3); and graphic/animation (X_4). In addition, we consider the following parallel aggregate (parallel combined) DA's:

$$X_5 = X_1 \ \& \ X_2; \quad X_6 = X_3 \ \& \ X_4; \quad \text{and}$$
$$X_7 = X_2 \ \& \ X_3 \ \& \ X_4.$$

Here the index corresponds to the number of DA's, and instead of X we use a certain notation of the leaf (e.g., T_1, E_3). Our obtained working HMSO is presented in Fig. 9. Thus the problem is to select DA's (kinds of alternative representation elements: $\{X_1, \ldots, X_7\}$) for each leaf node (tree-like system model, Fig. 9) to compose a series combination (a series solving strategy) of the DA's.

Let us construct a series solving strategy on the basis of morphological clique problem. Priorities of DA's ($r = 1, \ldots, 4$), obtained by expert judgment, are presented in brackets in Fig. 9. Table 4 contains basic compatibility for basic DA's. As a result, we obtain compatibility which is presented in Tables 5 and 6.

Thus we get the following composite DA's:

(a) $P_1 = I_5 \star T_2 \star V_6$, $N(P_1) = (4; 2, 0, 1, 0)$;
(b) $P_2 = I_3 \star T_5 \star V_4$, $N(P_2) = (4; 1, 2, 0, 0)$; and
(c) $D_1 = E_4 \star W_4$, $N(D_1) = (5; 2, 0, 0, 0)$.

Table 7 contains compatibility for the higher level of system hierarchy.

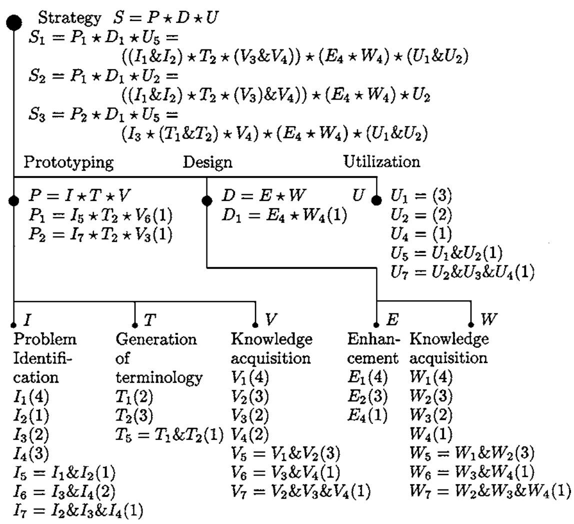

Figure 9. Design morphology.

Table 4. Compatibility.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
X_1	1	2	3	4	1	4	5
X_2	2	4	5	5	3	4	4
X_3	3	4	2	5	4	3	4
X_4	4	5	5	1	4	3	2
X_5	1	3	4	4	3	4	3
X_6	4	4	3	3	4	2	2
X_7	5	4	4	2	3	2	1

Table 5. Compatibility.

	T_1	T_2	T_3	V_1	V_2	V_3	V_4	V_5	V_6	V_7
I_1	1	2	1	1	2	3	4	1	4	5
I_2	2	4	3	2	4	5	5	3	4	4
I_3	3	4	4	3	4	2	5	4	3	4
I_4	4	5	4	4	5	5	1	4	3	2
I_5	1	3	3	1	3	4	4	3	4	3
I_6	4	4	4	4	4	3	3	4	2	2
I_7	5	4	3	5	4	4	2	3	2	1
T_1				1	2	3	4	1	4	5
T_2				2	4	5	5	3	4	4
T_3				1	3	4	4	3	4	3

Table 6. Compatibility.

	W_1	W_2	W_3	W_4	W_5	W_6	W_7
E_1	1	2	3	4	1	4	5
E_2	2	4	5	5	3	4	4
E_3	3	4	2	5	4	3	4

Table 7. Compatibility.

	D_1	U_1	U_2	U_4	U_5	U_7
P_1	4	2	4	4	3	3
P_2	3	2	4	3	4	3
D_1		4	5	1	4	2

Now we combine composite Pareto-effective (by N) DA's:

(i) $S_1 = P_1 \star D_1 \star U_5 =$ $((I_1\ \&\ I_2) \star T_2 \star (V_3\ \&\ V_4)) \star (E_4 \star W_4) \star (U_1\ \&\ U_2)$, $N(S_1) = (3; 3, 0, 0, 0)$;

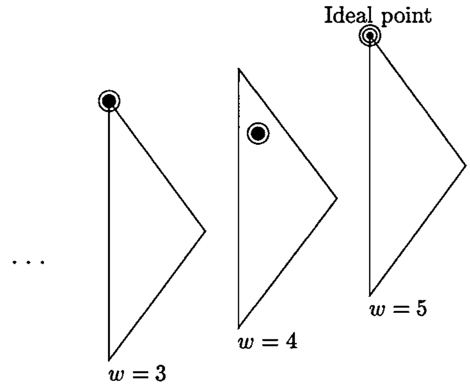

Figure 10. Space of system excellence and Pareto-effective points.

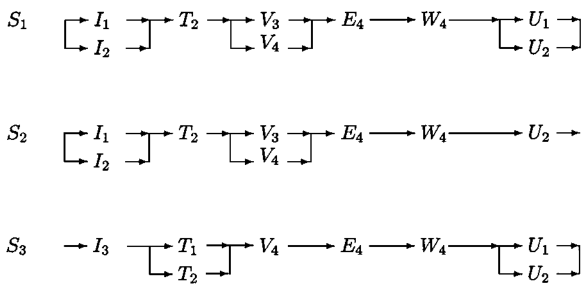

Figure 11. Series planning strategies.

(ii) $S_2 = P_1 \star D_1 \star U_2 =$ $((I_1\ \&\ I_2) \star T_2 \star (V_3\ \&\ V_4)) \star (E_4 \star W_4)$ $\star\, U_2$, $N(S_2) = (4; 2, 1, 0, 0)$;

(iii) $S_3 = P_2 \star D_1 \star U_5 =$ $(I_3 \star (T_1\ \&\ T_2) \star V_4) \star (E_4 \star W_4) \star (U_1\ \&\ U_2)$, $N(S_3) = (3; 3, 0, 0, 0)$;

Figure 10 illustrates two above-mentioned Pareto-effective points for S.

Let us point out a couple of bottlenecks and improvement actions for composite DA's:

(a) (P_1, U_5) (compatibility $3 \Rightarrow 4$) for S_1, new $N =$ $(4; 3, 0, 0, 0)$ (ideal point) and
(b) U_2 (priority $2 \Rightarrow 1$) for S_2, new $N = (4; 3, 0, 0)$.

Resultant series planning strategies for S are depicted in Fig. 11.

6. Conclusion

In our article, we have focused on the combinatorial system approach to the system analysis and design (including modeling, analysis, adaptation, improvement, multi-stage planning). This is our attempt to consider modular system approach to the system analysis, design, and improvement. Human-computer system (a representative of complex systems) is examined as a composite system. We proposed a structural description of human-computer systems (system components and interconnection), some corresponding requirements to them, and improvement actions. This is a basis for our modular system model.

Our HMMD (hierarchical system analysis and composing the system from design alternatives) based on a "design morphology" for the examined system has been used. Evidently, the article has a discussion character and is oriented to promote hybrid system design approaches for many applied domains. Note human-computer systems are useful and understandable examples (as representatives of complex systems) for many applied specialists to feel some contemporary systems approaches.

The article involves only a couple of basic system design problems. A movement from detail design (or design of system component) to the system design problems is a basic crucial contemporary direction of design community in all engineering domains. Our presented problems can be considered as some steps for the movement.

Clearly, our HMMD has many limitations for the use. For example, HMMD includes simple linear system decomposition approach. Many systems (and HCSs too) are very complicated and require complex non-linear dynamic system modeling and this direction is a basic one for other studies. On the other hand, linear system decomposition is very useful as an initial step to describe a complex system and for educational goals.

It is reasonable to list the following important future investigations:

1. to examine an usage of other analysis/design methods including various hybrid methods with AI components;
2. to study other applied representative kinds of complex systems;
3. to study various kinds of system design problems including system design problems above under uncertainty;
4. to analyze special issues of design/engineering skills acquisition for composite systems analysis/design;
5. to study some special problems for the composite systems (e.g., technical diagnostics and maintenance);
6. to use more complicated kinds of expert information, for example, on the basis of poset-like scales and composite spaces of system excellence; and
7. to use system analysis/design methods in engineering education courses, e.g., student research projects including composite multidisciplinary student teams for composite multidisciplinary systems.

Acknowledgments

A preliminary material of the article was presented at conferences EWHCI'95 and EWHCI'96. The author thanks reviewers of the article for their significant and helpful comments. The first version of the article was prepared when the author was visiting at The University of Aizu (Japan). The final stage of the research was supported by The Ministry of Absorption (Center of Absorption in Science) and The Ministry of Science, State of Israel.

References

1. D. Braha and O. Maimon, *A Mathematical Theory of Design: Foundations, Algorithms and Applications*, Kluwer: Dordrecht, 1998.
2. D.M. Buede, *The Engineering Design of Systems: Models and Methods*, J.Wiley & Sons: New York, 1999.
3. J. Gero and F. Sudweeks (eds.), *Advances in Formal Design Methods*, Kluwer: Boston, 1996.
4. A. Kusiak, *Engineering Design: Products, Processes, and Systems*, Academic Press: New York, 1999.
5. C.S. Adjiman, I.P. Androulakis, and C.A. Floudas, "Global optimization of mixed-integer nonlinear problems," *AIChE Journal*, vol. 46, no. 9, pp. 1769–1797, 2000.
6. C.A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications (Topics in Chemical Engineering)*, Oxford University Press: New York, 1995.
7. I.E. Grossmann, "Mixed-integer non-linear programming techniques for the synthesis of engineering Systems," *Research in Engineering Design*, vol. 1, no. 2/3, pp. 205–228, 1990.
8. K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer: Boston, 1999.
9. R.B. Statnikov and J. Matusov, *Multicriteria Optimization and Engineering*, Chapman and Hall: New York, 1995.
10. R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, J. Wiley & Sons: New York, 1986.
11. E.L. Ulungu and J. Teghem, "Multi-objective combinatorial optimization problems: A survey," *J. of Multi-Criteria Decision Analysis*, vol. 3, no. 2, pp. 83–104, 1994.

12. C.A. Coello Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1995.
13. C.M. Fonseca and P.J. Fleming, "An overview of evolutionary algorithms for multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.
14. N.M. Alexandrov and M.Y. Husssaini (eds.), "Multidisciplinary design optimization: State of the art," in *Proceedings in Applied Mathematics Series; no. 80*, SIAM: Philadelphia, 1997.
15. R.B. Statnikov, *Multicriteria Design: Optimization and Identification*, Kluwer: Dordrecht, 1999.
16. I.M. Sobol, "An efficient approach to multicriteria optimum design problems," *Surveys on Mathematics in Industry*, vol. 1, pp. 259–281, 1992.
17. C.A. Floudas and P. Pardalos (eds.), *State of the Art in Global Optimization: Computational Methods and Applications*, Kluwer: Boston, 1996.
18. C.A. Floudas, *Deterministic Global Optimization*, Kluwer: Boston, 1999.
19. G. Booch, *Object Oriented Development*, The Benjamin/Cummings Publishing Company: Redwood City, California, 1991.
20. G. Harhalakis, C.P. Lin, R. Nagi, and J.M. Proth, "Hierarchical decision making in computer integrated manufacturing systems," in *Proc. of the Third Intl. Conf. on CIM*, IEEE CS Press: Los Alamitos, CA, 1992, pp. 15–24.
21. P.S. Krasnoshekov, V.V. Morozov, and V.V. Fedorov, "Decomposition in design problems," *Engineering Cybernetics*, no. 2, pp. 7–17, 1979 (in Russian).
22. D.C. Liebisch and A. Jain, "JESSI COMMON FRAMEWORK design management—the means to configuration and execution of the design process," in *Proc. of European Design Automation Conference EURO-DAC'92*, CS Press: Los Alamitos, 1992, pp. 552–557.
23. E. Yourdan, *Techniques of Program Structure and Design*, Prentice-Hall: Englewood Cliffs, NJ, 1975.
24. O. Berman and N. Ashrafi, "Optimization models for reliability of modular software systems," *IEEE Trans. on Software Engineering*, vol. 19, no. 11, pp. 1119–1123, 1993.
25. C.C. Huang and A. Kusiak, "Modularity in design of products and systems," *Trans. on Systems, Man, and Cybernetics, Part A*, vol. 28, no. 1, pp. 66–77, 1998.
26. M.Sh. Levin, *Combinatorial Engineering of Decomposable Systems*, Kluwer Academic Publishers: Dordrecht, 1998.
27. F. Zwicky, *Discovery Invention, Research Through the Morphological Approach*, McMillan: New York, 1969.
28. M. Shaw and D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall: New York, 1996.
29. A.P. Gupta, W.P. Birmingham, and D.P. Siewiorek, "Automating the design of computer systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 4, pp. 473–487, 1993.
30. J. McDermott, "R1: A rule-based configurer of computer systems," *Artificial Intelligence*, vol. 19, no. 2, pp. 39–88, 1982.
31. E.A. Sykes and C.C. White, III, "Multiobjective intelligent computer-aided design," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, no. 6, pp. 1498–1511, 1989.
32. J. Gero and F. Sudweeks (eds.), *Artificial Intelligence in Design'98*, Kluwer: Dordrecht, 1998.
33. L. Bass and J. Coutaz, *Developing Software for the User Interface*, Addison Wesley: Reading, MA, 1991.
34. D. Hix, "Generations of user-interface management systems," *IEEE Computer*, vol. 5, no. 5, pp. 77–87, 1990.
35. P. Johnson, *Psychology, Task Analysis and Software Engineering*, Prentice-Hall: London, 1992.
36. J.A. Larson, *Interactive Software—Tools for Building Interactive User Interfaces*, New York, Yourdan Press Computing series, Prentice Hall: Englewood Cliffs, 1992.
37. K.A. Mayo and H.R. Hartson, "Synthesis-oriented situational analysis in user interface design," in *Human-Computer Interaction*, edited by L.E. Bass, J. Gornostaev, and C. Unger, LNCS, vol. 753, Springer-Verlag: Berlin, pp. 53–67, 1993.
38. I. Benbasat and P. Todd, "An Experimental investigation of interface design alternatives: Icons vs. text and direct manipulation vs. menus," *Intl. J. Man-Machine Studies*, vol. 38, pp. 369–402, 1993.
39. T.L. Saaty, *The Analytic Hierarchy Process*, MacGraw Hill: New York, 1988.
40. J. Nielsen, *Usability Engineering*, Academic Press: New York, 1993.
41. D. Browne, P. Totterdell, and M. Norman, *Adaptive User Interface*, Academic Press: London, 1990.
42. M. Debevc, D. Donlagic, R. Svecko, S. Spasovski, and B. Meyer, "Flexible user support with adaptive bar," in *Proc. of Intl. Conf. on Human Computer Interaction EWHCI'94*, St. Petersburg, vol. 1, 1994, pp. 129–134.
43. E. Kantorowitz and O. Sudarsky, "The adaptable user interface," *Comm. of the ACM*, vol. 32, pp. 1352–1358, 1989.
44. U. Malinowski, T. Kuhme, M. Schneider-Hufschmidt, and H. Dieterich, "Adaptive user interfaces," in *Intelligent Workstations for Professionals*, edited by A.H. Rubenstein and H. Schwartzel, Springer-Verlag: Berlin, pp. 268–278, 1992.
45. A.F. Norcio and J. Stanley, "Adaptive human computer interfaces: A literature survey and perspective," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 19, no. 2, pp. 399–408, 1989.
46. M. Schneider-Hufschmidt, T. Kuhme, and U. Malinowski, *Adaptive User Interfaces: Principles and practice*, Elsevier Science Publisher: Amsterdam, 1993.
47. P. Brusilovsky, "Adaptive hypermedia: The state of the art," in *Proc. of Intl. Conf. Multimedia, Hypermedia and Virtual Reality*, Moscow, 1993, pp. 24–29.
48. R.D. Hill and M. Herrmann, "The composite object user interface architecture," in *User Interface Management and Design*, edited by D.A. Duce, M.R. Gomes, F.R.A. Hopgood, and J.R. Lee, Springer-Verlag: Berlin, 257–271, 1990.
49. H.A. Simon, *The Science of Artificial*, MIT Press: Cambridge, MA, 1969.
50. A. Newell and H.A. Simon, *Human Problem Solving*, Prentice Hall: Englewood Cliffs, NJ, 1972.
51. N.J. Nilsson, *Problem Solving Methods in Artificial Intelligence*, MacGraw-Hill Book Company: New York, 1971.
52. L.M. Applegate, B.R. Konsynski, and J.F. Nunamaker, "Model management systems: Designs for decision support," *Decision Support Systems*, vol. 2, no. 1, pp. 81–91, 1986.
53. D.R. Dolk and B.R. Konsynski, "Knowledge representation for model management systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 10, no. 6, pp. 619–628, 1984.
54. D. Grosh and R. Agarwal, "Model selection and sequencing in decision support systems," *OMEGA*, vol. 2, no. 2/3, pp. 157–167, 1991.

55. W.A. Muhanna and R.A. Pick, "Meta-modeling concepts and tools for model management: A system approach," *Management Science*, vol. 40, no. 9, pp. 1093–1123, 1994.
56. K. Erol, J. Hendler, and D. Nau, "Complexity results for HTN planning," *Annals of Mathematics and Artificial Intelligence*, vol. 18, pp. 69–93, 1996.
57. M.Sh. Levin, "Hierarchical components of human-computer systems," in *Human-Computer Interaction*, edited by L.E. Bass, J. Gornostaev, and C. Unger, LNCS, vol. 753, Springer-Verlag: Berlin, pp. 37–52, 1993.
58. R.L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, J. Wiley & Sons: New York, 1976.
59. B. Stilman, "Network languages for complex systems," *Intl. J. Computers and Mathematics with Applications*, vol. 26, no. 8, pp. 51–79, 1993.
60. T.J. Teorey and J.P. Fry, *Design of Database Structures*, Prentice-Hall: Englewood Cliffs, NJ, 1982.
61. G.S. Altshuller, *Creativity as Exact Science. The Theory of the Solution of Inventive Problems*, Gordon and Breach Science Publishers: London, 1984.
62. D.A. Norman, "Four stages of user activity," in *Human-Computer Interaction INTERACT'84*, edited by B. Shackel, Elsevier Science Publishers B.V. North Holland: New York/ Amsterdam, 1985, pp. 507–511.
63. A. Dix, "Information processing, context and privacy," in *Human-Computer Interaction INTERACT'90*, edited by D. Diaper, Elsevier Publishers B.V. North Holland: New York/ Amsterdam, 1990, pp. 15–20.
64. W.B. Rouse, "A note on the nature of creativity in engineering: Implications for supporting system design," *Information Processing & Management*, vol. 22, no. 4, pp. 279–285, 1986.
65. C. Hales, "Designer as chameleon," *Design Studies*, vol. 6, no. 2, pp. 111–114, 1985.
66. D.V. Morse and C. Hendrickson, "A communication model to aid knowledge-based systems," *AI EDAM*, vol. 4, no. 2, pp. 99–115, 1990.
67. N. Cross, "Styles of learning, designing and computing," *Design Studies*, vol. 6, no. 3, pp. 157–162, 1985.
68. H. Marmolin and Y. Sundblad, "An analysis of design and collaboration in a distributed environment," in *Proc. of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson, and K. Schmidt, Kluwer: Amsterdam, 1991.
69. W.B. Rouse, J.A. Cannon-Bowers, and E. Salas, "The role of mental models in team performance in complex systems," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, no. 6, pp. 1296–1308, 1992.
70. J. Piaget, *Biology and Knowledge*, The University of Chicago Press: Chicago, 1971.
71. H. Gardner, *Creating Minds: An Anatomy of Creativity*, Basic Books, New York, 1993.
72. A. Kobsa and W. Wahlster (eds.), *User Models in Dialog Systems*, Springer-Verlag: Berlin, 1989.
73. J. Fedorovicz and G.B. Williams, "Representing modeling knowledge in an intelligent decision support system," *Decision Support Systems*, vol. 2, no. 1, pp. 3–14, 1986.
74. T.-P. Liang, "Development of a knowledge-based model management system," *Operations Research*, vol. 36, no. 6, pp. 849–863, 1988.
75. T.A. Gavrilova and E.V. Zudilova, "Conceptual design of interfaces for knowledge acquisition systems," *Engineering Cybernetics*, no. 2, pp. 3–11, 1994 (in Russian).
76. F.W. Rook and J.W. Croghan, "The knowledge acquisition activity matrix: A systems engineering conceptual framework," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 19, no. 3, pp. 586–597, 1989.

Mark Sh. Levin received the M.S. degree in Radioengineering from Moscow Technological University for Communication and Informatics (1970), the M.S. degree in Mathematics from Moscow State University (1975), the Ph.D. degree in Systems Analysis and Combinatorial Optimization from Institute for Systems Analysis of Russian Academy of Sciences (1982). Many years he was with Moscow R&D organizations as Engineer, Senior Engineer, and Head of Laboratory. After that he was Senior/Leading Research Scientist in Russian State Committee of Standards and in Russian Academy of Sciences, the Author of Project with Ofakim Center of Technology (Israel), an Indepedent Researcher (Moscow, Russia), a Visiting Researcher (status of Professor) at The University of Aizu (Fukushima, Japan), a Visiting Scholar at University of Ottawa and University of Toronto (Canada), a Research Scientist at The College of Judea and Samaria (Israel). Now Dr. Levin is a Researcher at Department of Mechanical Engineering, The Ben-Gurion University of the Negev (Israel).

Dr. Levin is interested in decision making technology, systems engineering, combinatorial optimization, and applications in engineering, management, and education. He has published over 60 papers and two books: "Decision Making: Combinatorial Models of Information Approximation" (with co-author, Nauka, 1990, in Russian) and "Combinatorial Engineering of Decomposable Systems" (Kluwer, 1998). Dr. Levin is member of IEEE, ACM, Intl. Society of Applied Intelligence, and OR Society of Israel.