ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/2416541

Improvement of Decomposable Systems.

Chapter · October 1996

Source: CiteSeer

CITATIONS
2

READS
15

1 author:

Mark Sh. Levin
Russian Academy of Sciences

190 PUBLICATIONS 885 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Scheduling View project

All content following this page was uploaded by Mark Sh. Levin on 12 February 2014.

The user has requested enhancement of the downloaded file.

IMPROVEMENT OF DECOMPOSABLE SYSTEMS

Mark Sh. Levin
The University of Aizu, Fukushima, 965-80 Japan
Email: mark@u-aizu.ac.jp

Abstract

The paper describes an improvement process of decomposable systems. We examine the following: an existing system; hypothetical new elements of the system and improvement actions; the change system (i.e., a hierarchy of improvement actions), and change schedule system (series-parallel schedule or trajectory). The design and analysis of the systems is based on hierarchical morphological multicriteria design (HMMD). A numerical example demonstrates the design, and improvement.

1 Introduction

Problems of the improvement in complex systems have been studied in various disciplines ([2], [6], [15], etc.). This paper addresses the description, and improvement of decomposable systems. Generally only two major approaches to the system design are well-known [33]:

(i) improvement of an existent system; and

(ii) designing a new system.

Usually, the first approach consists in the evolutionary improvement and multi-criteria selection of design alternatives ([1], [5], [7], [17], [32], etc.). We analyze the use of HMMD (designing a new system) [20] to represent and to design the improvement process. Note that hierarchical approaches to plan or to schedule have been studied many years, for example:

(1) hierarchical planning systems [8];

(2) hierarchical decision making in manufacturing [16];

(3) hierarchical tasks network (HTN) decomposition ([10], [11], etc.).

Here we use HMMD not only for the design and analysis of a system, but to design a change system (a hierarchy of improvement actions) and to plan a system improvement process too. Similar processes are basic ones in the quality improvement, and redesign or re-engineering. We analyze decomposable systems, main elements of the improvement process, our generalized framework of the improvement, and support combinatorial models to schedule improvement actions (e.g., clique, morphological clique, etc.). Our numerical example demonstrates stages of the improvement process.

2 Decomposable systems

In this paper, we examine decomposable systems, consisting of components and their interconnection (Is) or compatibility. Here we use basic assumptions of HMMD as follows [20]:

(1) decomposability of a system (i.e., tree-like structure);

(2) a system excellence is an aggregation of subsystems qualities and qualities of Is (compatibility) among subsystems;

(3) monotone criteria for the system components are used;

(4) qualities of subsystems and their Is are evaluated upon ordinal scales, which are coordinated.

And we assume the following hierarchical description of a system:

(1) tree-like system model;

(2) design alternatives (DAs) for leaf nodes of the model;

(3) priorities of DAs ($r = 1, ..., k$; 1 corresponds to the best one);

(4) ordinal compatibility for DAs ($w = 0, ..., l$, l corresponds to the best one).

Generally, we can examine the following kinds of requirements: criteria for nodes of the system model (DAs), constraints for DAs, factors of compatibility among DAs.

A basic version of HMMD involves the following phases:

(1) the design of system model (including a specification hierarchy);

(2) the generation of design alternatives for leaf nodes of the model;

(3) the hierarchical selection and composing of DAs into composite DAs;

(4) the analysis and improvement of composite DAs.

The composing of composite DAs is based on the following problem ([19], [20], [24]):

Find a composite design alternative

$S = S(1) * ... * S(i) * ... * S(m)$

of DAs (one representative for each system components) with non-zero Is, where $S(i)$ is a design alternative for ith component of the designed system. Fig. 1 depicts a design system. In addition, the following situation of a system change is presented in Fig. 1:

1. An initial system is: $S = A * B * C * D$ with corresponding DAs (A_1, A_2, A_3, A_4; B_1, B_2, B_3, B_4; C_1, C_2, C_3, C_4; D_1, D_2, D_3, D_4).

2. Change actions are the following:

(i) changing of the system structure: (a) removal of component D; (b) addition of component E;

(ii) changing of DAs: (a) removal of B_2, C_1; (b) addition of A_3, A_4, B_4, C_3, C_4.

$$S = A * B * C * D^- \Rightarrow S = A * B * C * E^+$$

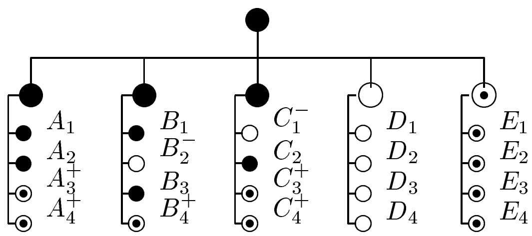

Fig. 1.Example of modified system

And we use a lattice of the system excellence on the base of the following vector:

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

where $w(S)$ is the minimum of pairwise compatibility in S, $n(S) = (n_1, ..., n_r, ...n_k)$, where n_r is the number of DAs of the rth quality in S. Thus we search for solutions which are nondominated by $N(S)$.

As a result, we can analyze the following layers of system excellence:

(1) an ideal solution;

(2) Pareto-effective points;

(3) a neighborhood of Pareto-effective DAs (e.g., a solutions of this set maybe transformed into a Pareto-effective point on the base of the only one improvement step).

The following kinds of elements (DAs,Is) with respect to solution S are considered: *S-improving*, *S-neutral*, and *S-aggravating* ones by vector N; where *S-aggravating* elements are examined as bottlenecks. An improvement of the system is illustrates in Fig. 2. Here we point out the following:

(a) points: initial point S_o; the ideal point I; four Pareto-effective points; target point S^*; S_{o1} and S_{o2}, that are intermediate points of improvements (these points maybe examined as the neighbors of the Pareto-layer);

(b) series trajectories of improvements:

$\alpha =< S_o, S_{o1}, S^* >$ and $\beta =< S_o, S_{o2}, S^* >$.

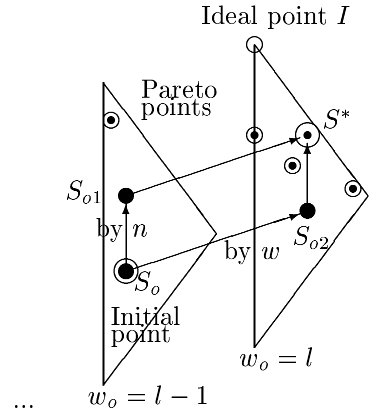

Fig. 2. Excellence lattice, improvements ($\rightarrow$)

In this paper we propose a similar stage for the statement and implementation of the improvement process on the base of HMMD. At this stage, we have to examine new kind of DAs as improvement actions, their interconnection (compatibility), and scheduling of these actions.

3 Improvement process

We examine the system improvement as series steps of the representation and processing of the following:

(a) initial system;

(b) hierarchical morphological design space;

(c) hierarchical change system (system of improvement actions);

(d) schedule of change actions.

Thus we need of the description of elements above, and methods to their processing. Note that a hierarchical approaches to plan are the basic ones ([8], [16], etc.).

3.1 Structure of improvement process

Let us consider the following interconnected levels:

(1) a space of system excellence, for example on the base of the lattice above;

(2) a set of compositions (composite DAs);

(3) a set of improvement trajectories, including a set of elementary improvement actions, and their series-parallel combinations (i.e., series-parallel trajectories).

Spaces of objects and their effectiveness are depicts in Fig. 3. Here we point out kinds of correspondences between elements of spaces above too.

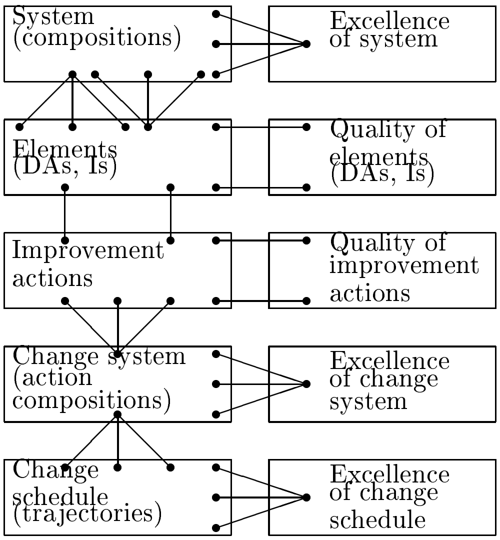

Fig. 3. Spaces of objects and effectiveness

Clearly that we have to take into account the following cases for a point of the excellence space:

(1) a corresponding composition does not exist;

(2) there exists the only one corresponding composition;

(3) there exist a set of corresponding compositions.

Analogically for a two compositions as start/end points of the improvement process we have got the same three cases.

We can point out several attempts to describe and use close multi-level descriptions of complex processes, for example:

(1) hierarchical task network planning ([10], [11], etc.);

(2) network languages for complex systems ([30], [31]).

In this paper, we examine main stages of the improvement process, which are shown in Table 1.

Generally, we can examine the following types of system changes:

1. Internal changes:

1.1 micro-level: (1) change of a subsystem (submodel, requirements) (2) change of DAs; (3) change of Is

1.2 macro-level: change of a system structure.

2. External changes:

2.1 requirements to the system;

2.2 searching for morphological solutions.

Table 1. Series improvement process

Object	Operations	Methods
1.Hierarchical description of existing system	Analysis of system, partitioning/ decomposition	Engineering techniques
2.Initial hierarchical morphological design space	Generation of new DAs (concurrently) Change of system structure	Engineering techniques Searching for new data
3.Extended hierarchical morphological design space	Generation of aggregate DAs (concurrently)	Clique problem with ordinal item compatibility
4.Hierarchical description of change system (improvement actions)	Analysis of system excellence and generation of improvement actions	HMMD for basic system, its analysis (morphological clique and improvement analysis)
5.Change system (selected improvement actions)	Design of change system and its analysis	HMMD for change system (morphological clique and improvement analysis)
6.Change schedule system (e.g., series-parallel schedule of improvement actions)	Design of series-parallel schedule or trajectory	HMMD for series-parallel schedule, dynamic programming, network planning, etc.

And now it is reasonable to investigate new types of requirements to new DAs (i.e., the improvement actions), their interconnection, and a structure of the system changes (Fig. 4).

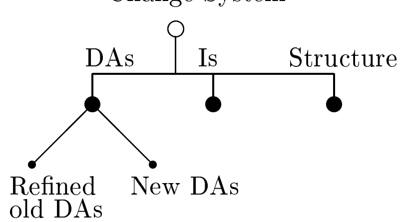

Fig. 4. Hierarchy of improvements

3.2 Phases and problems

Let us examine basic phases and problems of the improvement process:

Phase 1. Analysis of the initial system:

(1.1) analysis of the existing system;

(1.2) generation of new DAs and/or new system structure;

(1.3) generation of aggregate DAs;

(1.4) assessment of components (DAs, Is);

(1.5) evaluation of the system versions (i.e., composite DAs).

Phase 2. Generation of the improvement action set:

(2.1) generation of improvement actions on the base of the following: (a) expert judgment;, (b) examination of bottlenecks, (c) examination of the neighborhood of the Pareto-effective points, (d) examination of series neighborhood layers (i.e., the Pareto layer, the neighborhood of the Pareto layer, etc.);

(2.2) evaluation of the improvement actions including the following: (a) a profit of the actions, (b) required resources (time, etc.), (c) analysis of equivalent actions and their integration, (d) pair precedence relation between the actions;

(2.3) selection of admissible actions, and building of an action hierarchy (Fig. 3);

(2.4) building of a precedence digraph on the actions.

Phase 3. Design of an improvement implementation plan (trajectory) on the base of the following approaches: traditional network planning, dynamic programming, multistage planning, scheduling.

At the phase 3 we can examine the following types of problems:

(i) *optimization 1:* Find the best improvement plan with taking into account results (an excellence of the target system), and required resources;

(ii) *recognition:* Define the possibility (i.e., *Yes* or *No*) to reach a specified target solution(s) on the base of the specified set of the improvement actions;

(iii) *optimization 2:* Define the best improvement plan to reach the target system(s) in the case of the existence of the possibility (from problem *recognition*).

It is reasonable to use parallelization and/or coordination of improvement actions for problem (i). In this case we can design a multiperiod series-parallel improvement strategy on the base of HMMD ([21], [24]). The above-mentioned hierarchy of improvements maybe analyzed for each period with taking into account precedence relation of the actions (a basic morphological change system).

For problem (ii) and (iii), we can propose an analysis of series neighborhood layers and searching for an improvement trajectory on the base of two basic strategies (dynamic programming):

(a) from the target system(s) to the initial one;

(b) from the initial system to the target one(s).

3.3 Support methods

Here let us list support procedures as follows:

(1) mulricriteria ranking to obtain the ordinal priorities of DAs, or estimates of Is) ([4], [9], etc.);

(2) multicriteria clique problem with weighted compatibility of items to generate aggregate DAs ([21], [22]);

(3) morphological clique problem to find composite DAs ([22], [24]);

(4) multicriteria analysis of composite DAs ([20], [24]);

(5) generation of improvement actions [24];

(6) design of series-parallel schedule on the base of morphological clique problem ([21], [24]);

(7) searching for the best trajectory in an operational network on the base of operations management ([29], etc.), network methods and techniques (e.g., dynamic programming ([12], etc.), scheduling ([3], [12], etc.), etc.

4 Presentation issues

The importance of a complex objects presentation is increasing. In our case, we have to analyze several kinds of the systems (i.e., initial system, design space, change system, and improvement schedule), and their processing. Main presentation approaches for objects are as follows:

(1) structural modeling ([14], etc.);

(2) morphological presentation of complex objects and hierarchical alternatives ([20], [28], etc.);

(3) diagrams and flow-charts (e.g., for scheduling).

Techniques of the process presentation are mainly based on flowcharts, the use of languages, and special multi-media environments, for example:

(1) representation of complex technological processes (e.g., nets, bar diagrams, dataflow diagrams ([25], [27], etc.);

(2) morphological flow-chart presentation of operational environments [18];

(3) special languages ([13], etc.);

(4) complex presentation of algorithms/technique environments on the base of texts, animations, movements ([26], etc.).

5 Example

5.1 System and its analysis

We examine the following initial computer system S: hardware (J), software (V), information (Y), and personnel (H). A detailed investigation of an information center has been executed in [23]. In our case the initial system is a composition of DAs as follows: $S_o = J_0 * V_0 * Y_0 * H_0$.

At the next stage, we consider the following:

(a) generation of new DAs;

(b) design of a new system structure (an additional component *communication* C);

(c) generation of aggregate DAs; and

(d) removal of Y_2.

Clearly, that now $S_0 = J_0 * V_0 * Y_0 * H_0 * C_0$.

Table 2 contains descriptions of DAs (priorities are shown in brackets). Compatibility of DAs are presented in Table 3.

Table 2. DAs

DAs	
Several personal computers	$J_o(3)$
Workstation	$J_1(2)$
LAN	$J_2(2)$
Initial DBMS	$V_o(2)$
New DBMS	$V_1(1)$
Expert system	$V_2(3)$
New DBMS and expert system	$V_3 = V_1 \& V_2 (2)$
Initial database	$Y_o(2)$
Initial database and special intellectual interface	$Y_1(1)$
Knowledge base	$Y_2(3)$
Initial data base and knowledge base	$Y_3 = I_1 \& I_2 (2)$
Initial personnel	$H_o(2)$
Trained personnel	$H_1(1)$
Trained personnel and a knowledge engineer	$H_2(1)$
New personnel oriented to knowledge engineering	$H_3(2)$
None	$C_o(2)$
Access to external databases in certain time	$C_1(1)$
Real-time communication	$C_2(2)$

Table 3. Compatibility of DAs

	Y_o	Y_1	Y_3	Y_0	J_1	J_2	J_0	V_1	V_2	V_3	V_0	C_1	C_2
H_0	3	2	1	3	3	3	3	3	0	1	3	2	1
H_1	3	3	1	3	3	3	3	3	0	2	3	3	2
H_2	3	3	1	3	3	3	3	3	0	2	3	3	3
H_3	3	3	3	2	2	2	3	3	3	3	3	3	3
Y_o				2	3	3	3	3	0	3	3	1	1
Y_1				2	3	3	3	3	0	3	1	3	3
Y_3				1	3	3	2	2	1	0	2	3	3
J_o							3	3	2	2	3	2	1
J_1							3	3	3	3	3	3	3
J_2							3	3	3	3	3	3	3
V_o											3	0	0
V_1											3	2	2
V_2											0	3	3
V_3											2	3	3

Thus $N(S_o) = (2; 0, 4, 1)$. And Pareto-effective point set consists of the following elements:

1. $N = (2; 4, 1, 0)$:

$S_1 = J_1 * V_1 * Y_2 * H_1 * C_1$,

$S_2 = J_2 * V_1 * Y_2 * H_1 * C_1$,

$S_3 = J_1 * V_1 * Y_2 * H_2 * C_1$,

$S_4 = J_2 * V_1 * Y_2 * H_2 * C_1$.

2. $N = (3; 2, 3, 0)$:

$S_5 = J_1 * V_1 * Y_0 * H_1 * C_0$,

$S_6 = J_2 * V_1 * Y_0 * H_1 * C_0$,

$S_7 = J_1 * V_1 * Y_0 * H_2 * C_0$,

$S_8 = J_2 * V_1 * Y_0 * H_2 * C_0$.

Table 4 contains improvement actions, which are obtained on the base of bottlenecks (i.e., *S-aggravating* elements). We use the following types of improvements by results:

(1) generation of an ideal point;

(2) improvement of Pareto-effective points;

(3) refinement of neighbors of the Pareto layer;

(4) improvement and compression of the Pareto layer.

Table 4. Bottlenecks and improvements

Composite DAs	Bottlenecks		Actions	
	DAs	Is	w/r	Type
S_1	J_1		$2 \to 1$	1
S_1		V_1, C_1	$2 \to 3$	1
S_2	J_2		$2 \to 1$	1
S_2		V_1, C_1	$2 \to 3$	1
S_3	J_1		$2 \to 1$	1
S_3		V_1, C_1	$2 \to 3$	1
S_4	J_2		$2 \to 3$	1
S_4		V_1, C_1	$2 \to 3$	1
S_5	J_1		$2 \to 1$	2
S_5	Y_0		$2 \to 1$	2
S_5	V_0		$2 \to 1$	2
S_6	J_2		$2 \to 1$	2
S_6	Y_0		$2 \to 1$	2
S_6	V_0		$2 \to 1$	2
S_7	J_1		$2 \to 1$	2
S_7	Y_0		$2 \to 1$	2
S_7	V_0		$2 \to 1$	2
S_8	J_2		$2 \to 1$	2
S_8	Y_0		$2 \to 1$	2
S_8	V_0		$2 \to 1$	2

5.2 Change system

Generally the change system consists of the following subsystems (we point out possible improvement actions for our example):

1. Improvement of components: J, V, Y, H, C.

2. Improvement of compatibility: (J, V), (J, Y), etc.

Our consideration in previous section is the base to compress the change space, because we will examine only improvements of S_o, and Pareto-effective points.

5.3 Improvement trajectories

Now let us consider improvements at the space of compositions. We have to remember that our basic point is S_o, and for each other improvement trajectories it is necessary to add a start part as follows: from S_o to a point (e.g., S_1, etc.). Thus we can examine the following three kinds of improvement trajectories:

(1) from S_o directly to ideal point I.

(2) from S_o to points S_1 or S_2 or S_3 or S_4; and from the point to I;

(3) from S_o to points S_5 or S_6 or S_7 or S_8; and from the point to I.

We examine two types of improvement actions as follows: (a) a replacement of an element ($\to$), and (b) an improvement of an element ($\uparrow$). In our example basic improvement actions are the following: $J_0 \to J_1$, $J_0 \to J_2$, $V_0 \to V_1$, $Y_0 \to Y_1$, $H_0 \to H_2$, $H_0 \to H_2$, $C_0 \to C_1$.

The improvement actions for Pareto-effective points are presented in Table 4. Thus we can consider series-parallel trajectories of the 2nd kind above:

1. $\rho_1 = (S_o \to S_1) * (S_1 \uparrow) =$

(a) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * (J_1 \uparrow)$;

(b) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow)$.

2. $\rho_2 = (S_o \to S_2) * (S_2 \uparrow) =$

(a) $((J_o \to J_2)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * (J_2 \uparrow)$;

(b) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_1)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow)$.

3. $\rho_3 = (S_o \to S_3) * (S_3 \uparrow) =$

(a) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * (J_1 \uparrow)$;

(b) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow)$.

4. $\rho_4 = (S_o \to S_4) * (S_4 \uparrow) =$

(a) $((J_o \to J_2)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * (J_2 \uparrow)$;

(b) $((J_o \to J_1)\&(V_o \to V_1)\&(Y_o \to Y_2)\&(H_o \to H_2)\&(C_o \to C_1)) * ((V_1, C_1) \uparrow)$.

In addition, improvement trajectory $\rho_5 = (S_o \to S_5) * (S_5 \to I)$ is shown in Fig. 5.

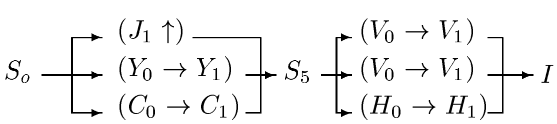

Fig. 5. Series-parallel imrovement trajectory

6 Conclusion

We have considered the hierarchical design, and improvement of decomposable systems. Our examination maybe used for various applications, for example:

(1) distributed information systems (modification, improvement, re-design);

(2) improvement of network systems through modification.

In addition, it is reasonable to point out the significance of the kinds of optimization problems, when we search for the best improvement of a combinatorial system. Similar approach maybe used for many well-known combinatorial problems on graphs.

Finally, let us emphasize the following significant research directions:

(1) development of special knowledge based systems to design of the change system;

(2) development of tools for the presentation of complex systems, and improvement processes;

(3) study of corresponding scheduling problems;

(4) development of special knowledge based systems to evaluate the change system with taking into account estimates of the schedule change system;

(5) application of examined issues in engineering education.

References

[1] R.U. Ayres, *Technological Forecasting and Long-Time Planning*, New York: McGraw-Hill Book Company, 1969.

[2] O. Berman , D.I. Ingco and A. Odoni, "Improving the Location of Minimax Facilities Through Network Modification". *Networks*, vol. 24, pp. 31-41, 1994.

[3] J. Blazewicz, K.H. Ecker, G. Schmidt, J. Weglarz, *Scheduling in Computer and Manufacturing Systems*, 2nd ed., Berlin: Springer-Verlag, 1994.

[4] D.M. Buede, "Software Review. Overview of MCDA Software Market". *J. of Multi-Criteria Decision Analysis*, vol. 1, no. 1, pp. 59-61, 1992.

[5] D.M. Buede and R.W. Choisser, "Providing an Analytical Structure for Key System Choices", *J. of Multi-Criteria Decision Analysis*, vol. 1, no. 1, pp. 17-27, 1992.

[6] D. Das, S.K. Gupta, and D.Nau, "Reducing setup cost by automatic generation of redesign suggestions. In: *Proc. of ASME Conference "Computers in Engineering"*, 1994.

[7] R.L. De Neufville, and D. Marks, (Eds.), *Systems Planning and Design Case Studies in Modeling, Optimization, and Evaluation.* Prentice Hall, Englewood Cliffs, NJ, 1974.

[8] M.A.H. Dempster, M.L. Fisher, L. Jansen, B.J. Lageweg, J.K. Lenstra, and A.H.G. Rinnooy Kan, "Analytical Evaluation of Hierarchical Planning Systems". *Operations Research*, vol. 29, no. 4, pp. 707-716, 1981.

[9] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius and S. Zionts, "Multiple Criteria Decision Making, Multi-attribute Utility Theory: The Next Ten Years". *Manag. Sci.*, vol. 38, no. 5, pp. 645-654, 1992.

[10] K. Erol, D. Nau, and J. Hendler, "HTN Planning: Complexity and Expressivity". In: *AAAI-1994*, Seattle, July 1994.

[11] K. Erol, J. Hendler, and D. Nau, "Complexity Results for HTN Planning". *Annals of Mathematics and Artificial Intelligence*, 1995.

[12] M.R. Garey and D.S. Johnson, *Computers and Intractability. The Guide to the Theory of NP-Completeness.* San Francisco: W.H.Freeman and Company, 1979.

[13] D. Garlan, C.W. Krueger and B.S. Lerner, "TransformGen: Automating the Maintenance of Structured Environments". *ACM Trans. on Programming Languages and Systems*, vol. 16, no. 3, pp. 727-774, 1994.

[14] A.M. Geoffrion, "An Introduction to Structured Modeling". *Management Science*, vol. 33, no. 5, pp. 547-588, 1987.

[15] A. Gunasekaran, A.R. Korukonda, I. Virtanen and P. Yli-Olli, "Improving Productivity and Quality in Manufacturing Organizations". Intl. *J. of Production Economics*, vol. 36, no. 2, 169-183, 1994.

[16] G. Harhalakis, C.P. Lin, R. Nagi, and J.M. Proth, "Hierarchical Decision Making in Computer-Integrated Manufacturing Systems". In: *Proc. of Third Intl. Conf. on CIM.* CS Press, pp. 15-24, 1992.

[17] J.C. Jones, *Design Methods.* New York: Wiley, 1981.

[18] M.Sh. Levin, "Hierarchical Components of Human-Computer Systems". In: L.E. Bass, J. Gornostaev, C. Unger (eds.), *Human Computer Interaction.* LNCS, vol. 753, Berlin: Springer, pp. 37-52, 1993.

[19] M.Sh. Levin, "Hierarchical Design of User Interfaces". In: B. Blumenthal, J. Gornostaev, C. Unger (eds.), *Human Computer Interaction.* LNCS, vol.876, Berlin: Springer, pp. 140-151, 1994.

[20] M.Sh. Levin, "Towards Hierarchical Morphological Design of Decomposable System". In: *Proc. of The Second Conf. on Concurrent Engineering*, McLean, pp. 65-75, 1995.

[21] M.Sh. Levin, "A Hierarchical Planning and Problem Solving: Approach to Interface Design" . In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'95*, Moscow, vol. 1, pp. 180-187, 1995.

[22] M.Sh. Levin, "Towards Hypertexts and Decision Making." In: *Proc. of Intl. Conf. on Human-Computer Interaction EWHCI'95*, July 2-8, 1995, Moscow, vol. 2, pp. 22-37, 1995.

[23] M.Sh. Levin, "Problem of Information Center Planning". *Nauchno-Tekhnicheskaya Informatsiya*, Ser. 2, no. 2 (in Russian) 1995.

[24] M.Sh. Levin, *Towards Combinatorial Models of Synthesis*, Technical Report 96-1-002, The University of Aizu, 47 pp., 1996.

[25] M.L. Lu, Y. Naka, K. Shibao, X.Z.Wang and C.McGreavy, "A Multi-Dimensional Object-Oriented Information Model for Chemical Engineering". In: *Proc. of The Second Conf. on Concurrent Engineering*, McLean, pp. 21-29, 1995.

[26] N. Mirenkov, Visualization and Sonification of Methods. In: *Proc. of The First Aizu Intl. Symposium on Parallel Algorithms/Architecture Synthesis*, March 1995, Japan, IEEE CS Press, Los Alamitos, 63-71, 1995.

[27] R. Ortiz and P. Dadam, "The Concurrency Model: Activating and Engineering database through and Integrated Product and Process Data Model". In: *Proc. of the 6th Intl. Conference and Workshop on Database and Expert Systems Applications DEXA'95*, London, 1995.

[28] U.K. Patel and A.G. Sutcliffe, "Three-Dimensional Visualization of Knowledge Structures: Prototyping for Design Evaluation". In: L.E. Bass, J. Gornostaev, and C. Unger (Eds.) *Human Computer Interaction*, LNCS, vol. 753, Berlin: Springer-Verlag, pp. 163-180, 1993

[29] J. Singhal and J.L. Katz, "A Branch-And-Fathom Algorithm for the Long Range Process Design Problem". *Manag. Sci.*, vol. 36, no. 4, pp. 513-516, 1990.

[30] B. Stilman, "Network Languages for Complex Systems". *Intl. J. Computers and Mathematics with Applications*, vol. 26, no. 8, pp. 51-79, 1993.

[31] B. Stilman, "A Formal Language for Hierarchical Systems Control". *Languages of Design*, vol. 1, no. 4, pp. 333-356, 1993.

[32] E.A. Sykes and C.C. White,III, "Multiobjective Intelligent Computer-Aided Design", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 6, p. 1498-1511, 1989.

[33] J.P. Van Gigch, *Applied General Systems Theory.* 2nd ed., Vol. 1 and 2, Harper and Row Publishers, 1978.

View publication stats