

Components of Information Technology

2.1 Properties of Information Technology

First, let us point out the basic properties of information technology:

1. Various kinds of sources: (a) statistical information; books, databases; (b) specialists, population.
2. Preservation of initial information and possibility for reprocessing.
3. Possibility for parallel/concurrent processing.
4. Possibility for usage of many processing methods.
5. Possibility to accumulate various results of information processing.
6. Very high ecological awareness.
7. High requirements for professional skills of specialists.
8. Unique role of humans.
9. High requirements for information presentation and human–computer interaction.
10. Integration of many domains (*e.g.*, exact science, engineering, management and economics, psychology and cognitive science, education, art).
11. Wide range of users and professional domains (*e.g.*, science, engineering, management and economics, education, art, private life).

2.2 Information and Hierarchical Structuring

An integrated research direction of knowledge engineering and management is a key task [336, 372, 436, 558, 900]. This section addresses some issues of this direction, including the main problems, approaches, and applied systems.

2.2.1 Schemes and Methods

In recent years, issues of skill /knowledge acquisition have played a central role in many domains (*e.g.*, cognitive modeling, information technology, problem solving, knowledge-based systems, knowledge management, system design, engineering design, systems engineering, process systems engineering, team distributed design processes) [28–32, 50, 76, 97, 98, 102, 107, 182, 189, 263,

327, 372, 395, 478, 485, 614, 617, 619, 677, 733, 825]. Let us point out some other important research directions which are close to the above-mentioned domain:

- (1) cognitive modeling of design processes [7, 30, 32];
- (2) representation and study of design spaces [497, 555, 758];
- (3) modeling of specialists and his/her behavior (*e.g.*, mental models) [444, 698, 699, 807];
- (4) relation between cognitive processes and operations research [439];
- (5) models for representing experience in computers [447];
- (6) usage of object-oriented structures to represent expert knowledge [159];
- (7) visualization of knowledge and specialist experience [150];
- (8) design of hybrid intelligent systems with combined representation of various knowledge kinds [563];
- (9) organization and management of engineering information on complex products/systems and their reengineering [58, 151, 152, 495, 497, 619, 890];
- (10) studying and developing the knowledge and skills of specialists [182, 189];
- (11) acquiring engineering knowledge from design processes [395, 555];
- (12) design and maintenance of engineering concept spaces [155, 414, 577], including concept lattices [865, 882];
- (13) cognitive modeling for conceptual design [538];
- (14) engineering history bases [813]; and
- (15) information modeling for evolution/development of products/systems [177, 178, 497, 505, 512, 513, 591, 626].

In addition, it is reasonable to point out the contemporary research direction on *intellectual capital* [228, 438, 543, 639, 795]. In my opinion, engineering skills are a crucial part of national and international *intellectual capital*. Figure 2.1 depicts the above-mentioned components.

In this chapter, a hierarchical morphological framework is examined which is a useful organizational basis for systems engineering processes: to execute operations for the description, design, analysis and improvement of composite systems. In the case of composite systems or composite decisions, hierarchical structures are used for two goals (Figure 2.2): (a) hierarchy of the experts collaboration; (b) hierarchy of the resultant information description (for an engineering domain/system).

J. Conklin [176] has pointed out that a hierarchical representation of information is more natural for users with an engineering background. Note, construction of a generalization space for domain knowledge is a contemporary approach in the field of conceptual structuring [100, 101]. M. Minsky has pointed out the following architecture of representations [580]: (1) stories written in natural language; (2) story-like scripts; (3) transframes; (4) frame-arrays and picture-frames; (5) semantic nets; (6) *K*-lines; (7) neural nets; and (8) micronemes.

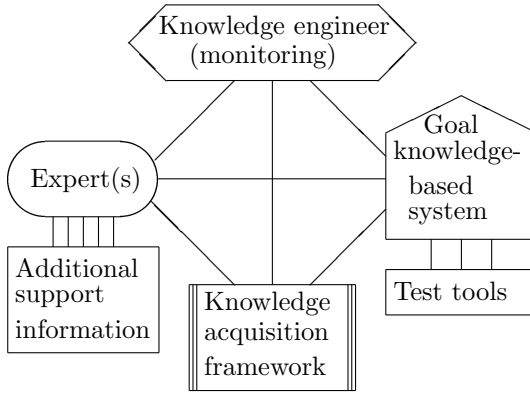


Figure 2.1 Knowledge acquisition: components

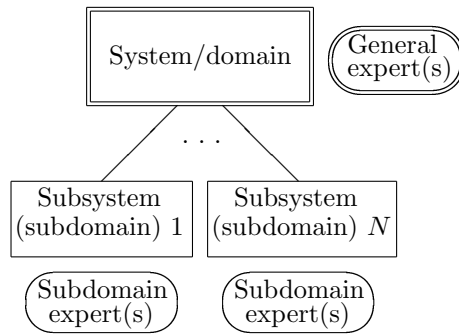


Figure 2.2 System hierarchy

2.2.2 Approaches to Knowledge Acquisition

The following four basic interconnected cognitive acts are examined [609]: (a) action; (b) observation; (c) reasoning; and (d) evaluation. Thus, the above-mentioned four components form a fundamental for knowledge acquisition methods. Traditional methods of knowledge acquisition are described, for example, in [162, 360, 434, 484, 780]. Surveys on knowledge acquisition are contained in [97, 727]. In our opinion, knowledge acquisition techniques are connected with domains as follows: (i) problem domains; (ii) domain experts, including their properties (type of thinking, experience, *etc.*); (iii) methods for knowledge representation; and (iv) possible usage of the knowledge. Note that the knowledge life cycle involves stages as follows [87]: 1. creation; 2. mobilization; 3. diffusion; and 4. commoditization.

The main kinds of system for knowledge acquisition are the following:

1. The knowledge acquisition activity matrix (a systems engineering conceptual framework) [695];
2. KADS technology [731, 878];

3. Systems of conceptual knowledge and ontologies [250, 437, 780]:
 - 3.1. CODE4: a unified system for managing conceptual knowledge [764];
 - 3.2. systems with a classification model that gives a structure to a knowledge acquisition environment (*e.g.*, STEPCLASS technique [278], M-Class model [562]);
 - 3.3. construction of ontologies [329, 381, 618, 780, 784, 846];
4. Knowledge acquisition techniques on the basis of ordinal classifications [278, 474];
5. Knowledge acquisition with *linguistic geometry* method [798];
6. Knowledge acquisition tools based on object-oriented approach (*e.g.*, SOOKAT [638]);
7. Multistage knowledge acquisition (*e.g.*, CAL Expert System [830]);
8. Systems for acquisition of various skills:
 - 8.1. systems for the creating and maintenance of a skill catalogue on the basis of two kinds of initial information [617]: (a) skill-related discussion database, and (b) person-related electronic communications (e-mail database).
 - 8.2. our approach based on hierarchies as morphological AND-OR trees [495, 497] (it is similar to ontological approach).
9. Other approaches:
 - 9.1. systems based on prototypical knowledge (*e.g.*, as prototype situations in system CENTAUR) [16];
 - 9.2. a system for representing and using real-world knowledge NETL [240];
 - 9.3. The knowledge acquisition and representation language KARL [251].

In addition, it is reasonable to point out the contemporary ontology design approaches to organization of knowledge as follows [381]: (a) inspirational approach; (b) inductive approach; (c) synthetic approach; and (d) collaborative approach.

2.2.3 Some Hierarchical Information Systems

The main functional operations for the design, utilization, and maintenance of design information systems are the following [206, 492, 668]: (1) acquisition of new data and knowledge; (2) structuring, modeling; (3) representation; (4) learning; (5) access, control; (6) analysis, evaluation, correction; and (7) maintenance. Support information systems may be oriented to various components of a systems engineering (system design) process. Also, it is reasonable to list reference examples of some information systems:

- (1) information design model (EDM) for engineering design [227];
- (2) distributed hypermedia system for managing knowledge in organizations (KMS) [18];
- (3) hierarchical hypertext system (HHS) involving components of different kinds and their attribute descriptions for various problem domains [490, 492];
- (4) *Designer's Electronic Guidebook* for mechanical engineering in Cambridge Engineering Centre [229];

- (5) information model for cooperative product development SHARED [890];
- (6) concept lattices and conceptual knowledge systems [856, 865, 882];
- (7) *system version spaces* [178];
- (8) systems as *engineering concept spaces* [155];
- (9) special combinatorial design spaces [758];
- (10) *large design spaces* [414];
- (11) various kinds of ontology-based information systems [250, 616];
- (12) information bases for evolution of products/systems [178, 497, 505, 512, 513, 626]; and
- (13) special *engineering history bases* [813].

Note, special logical studies of the hierarchical approach to modeling of knowledge are described in [238, 239, 346].

2.2.4 Our Framework

The basic contextual scale for information consists of four levels: (i) *data*, (ii) *information*, (iii) *knowledge*, and (iv) *decision*. Evidently, our main goal consists in the transformation of data, information, and knowledge into decisions. Here, *structuring* is the basic support operation [625, 867]. In our case, the decisions are based on a hierarchical morphological description (modeling) of some applied domains/systems. Figure 2.3 illustrates the structure of our decision environment for skills acquisition (on the basis of HMMD) [501].

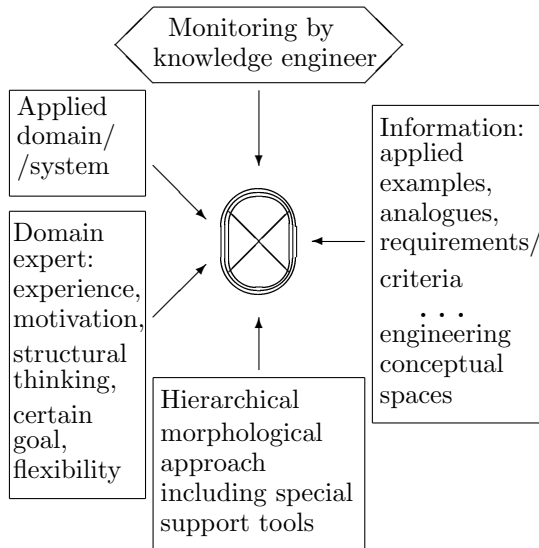


Figure 2.3 Structure of Environment

In our case, support tools involve the following: (a) systems for multicriteria ranking, *e.g.*, DSS COMBI PC [492, 497], and (b) a system for hierarchical design (DSS for hierarchical engineering design SED [492, 497]). Information

support is based on the following: (a) traditional scientific sources; (b) HHSs (requirements, criteria, situations, models, *etc.*) [490, 492]; (c) special systems as *engineering concept spaces* [155] and *design spaces* [414]; (d) *system version spaces* [178]; (e) Internet sources and tools; and (f) consulting of colleagues.

2.3 Decision-making Paradigm

2.3.1 Decision-making Framework

The basic decision-making paradigm consists of several basic stages [274, 390, 701, 759, 762, 836]. The first intellectual stage involves the following: analysis of a problem and its identification, information search, including revealing the main resource constraints, *etc.* The second *design* stage includes a multicriteria description of the problem (generation of alternatives, specification of criteria, and design of scales). The stage of *choice* is oriented to the evaluation of alternatives (accumulation of data on alternatives, acquisition of preference of specialists, information processing), selection of the best alternatives, and analysis of results including sensitivity analysis. And the final stage is an implementation of the decision (*i.e.*, the best alternative(s)). Clearly, special tools are required to support all the stages above. DSSs are oriented to these objectives [274, 497, 781, 836]. Now let us point out a set of main components of decision-making technology [74, 274, 836]: (1) alternatives, (2) criteria (and corresponding scales), (3) data (estimates of alternatives upon criteria, and expert preferences), (4) techniques (multicriteria choice, selection, *etc.*), (5) specialists (experts for the decision-making process, domain experts for the evaluation process, decision maker), and (6) resultant decisions.

Each component above maybe specified *a priori* or formed in the solving process. For example, alternatives may be specified or may be generated (corrected) as a result of solving procedures. Table 2.1 contains basic stages of the traditional decision-making paradigm of H.A. Simon [274, 390, 759, 762, 836]. Note, the concept of decision-aiding methodology as an extension of decision making was suggested in [701, 832, 833].

2.3.2 Towards Architecture of Decision Support Systems

A systemic view of decision support is described in [37]. The basic components of decision support procedures are the following: initial information base of models and methods, knowledge base, base of typical situations, and analysis and learning of user. A discussion of the design and use of DSSs is contained in [708], including the following: (a) generic technological components, (b) system requirement analysis and specification, (c) the use of alternative analytical methods, and (d) evaluation of such systems. The following technological way is examined: (1) user needs, (2) system requirements, (3) hardware and software allocation of the requirements, and (4) development of suitable hardware and software architectures.

Table 2.1 Stages of decision making and information support

Stages of decision making	Concepts, roles	Basic processes of information support
1. Problem identification	Problem types Resources Bottlenecks Objectives Constraints	Search and selection of analogous problems Search and/or selection of objectives, design of objective spaces
2. Structuring	Criteria Scales	Search and/or composing of criteria Choice/design of scales
3. Generation of alternatives	Alternatives Experts	Search and choice of analogous alternatives Composing of composite alternatives
4. Evaluation of alternatives	Techniques of assessment Estimates Experts	Search of data choice for alternatives Selection of experts Forming of expert groups
5. Choice (selection, ranking)	Techniques Experts Decision makers	Search and choice of techniques Search and selection of experts
6. Analysis	Techniques Experts Decision makers	Search and choice of techniques Search and selection of experts

The basic structure of a DSS involves the following parts [274, 426, 489, 497, 505, 648, 724, 781, 827, 836]: user interface, database, knowledge base, model base, HELPER. In addition, contemporary DSSs include tools for standard decision-making problems, and their combinations. In recent years, DSSs have been developed as integrated environments which are oriented to several stages of the decision-making process. The list of components for modern DSSs is as follows:

1. Models (multicriteria techniques, processing of interval estimates and fuzzy descriptions of alternatives, on-line design of problem-solving schemes, statistical data processing, knowledge bases, procedures of the result analysis).
2. Support of preliminary stages of decision making (preliminary data processing, procedures for identification and structuring of problems).
3. Tools for human-computer interaction (graphical user interfaces, training subsystems, expert procedures, support of group/distributed work).
4. Information support (*e.g.*, communication with databases, hypermedia systems).

Modern DSSs are based on the above-mentioned structure, and involve libraries of techniques, artificial intelligence components, procedures for the analysis of results, hyper-information subsystems, and special procedures for

problem identification and structuring [274, 435, 492]. In addition, visualization of multiattribute information (*e.g.*, by 2D and 3D graphics) is applied for user interfaces [408, 545, 566, 857]. Finally, important tendencies are the following:

1. support of group work [121, 467];
2. knowledge-based subsystems (*e.g.*, production subsystems, neural networks) with orientation to some auxiliary problems as follows: selection of techniques [808, 836, 876];
3. subsystems for scheduling and model management [594];
4. graphical interfaces [497, 827];
5. hypermedia information components [274, 435, 490];
6. subsystems for intellectual problems (generation of composite alternatives, design of criterion hierarchies, forming of specialist groups, generation of composite models) [497, 706].

2.3.3 Four-domain Decision Making

Generally, decision-making processes are based on a four-domain (four *spaces*) framework as follows (Figure 2.4): (i) a domain of alternatives, (ii) a domain of criteria and attributes, (iii) a domain of solving methods (algorithm, interactive procedures and their combinations), and (iv) a structural domain of the resultant decisions. Each above-mentioned domain can be modeled on the basis of some structures and hierarchies (from simple structures to complicated ones). Note, a closely integrated investigation is described in [827], including a multidimensional visual and interactive support environment for multicriteria decision making.

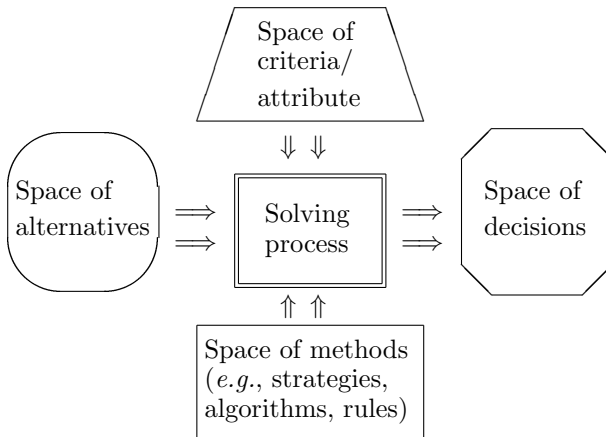


Figure 2.4 Four-domain decision-making framework

2.3.4 Basic Structures

Figures 2.5, 2.6, 2.7, and 2.8 illustrate some basic structures for the parts of the decision-making framework considered (alternatives, criteria/attributes,

decisions, and methods). The following simple structures are examined: (a) a single element, (b) a set of elements, (c) a chain of elements (*e.g.*, series structure, linear order), (d) a tree-like structure over elements, (e) a parallel-series structure, (f) partitioning of an element set (a set of clusters), (h) a multi-layer hierarchy of elements, and (f) the general case of a digraph. Note, the kind of element structure is a crucial from the viewpoint of complexity issues. For example, many combinatorial problems are polynomial ones on simple structures (*e.g.*, at the level of parallel-series graphs, interval graphs). This is based on the property of decomposability.

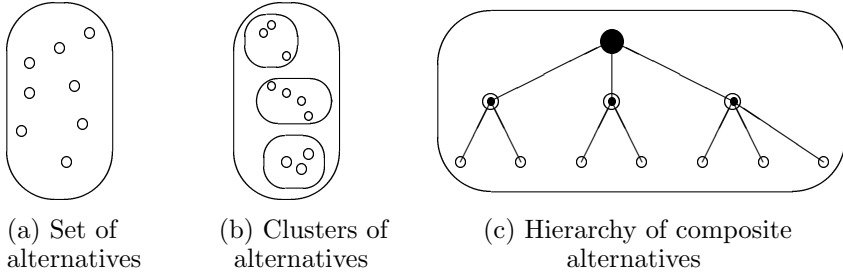


Figure 2.5 Structures of alternative space

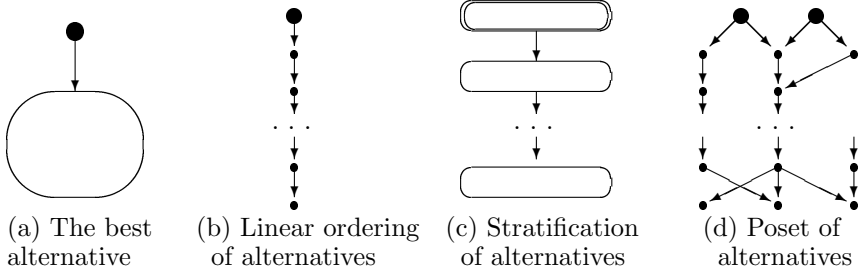


Figure 2.6 Basic structures for decision space

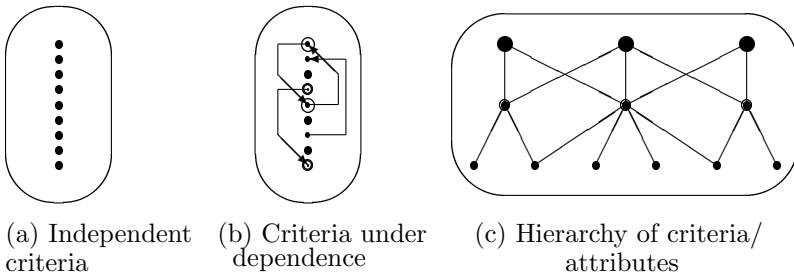


Figure 2.7 Structures of criteria/attributes

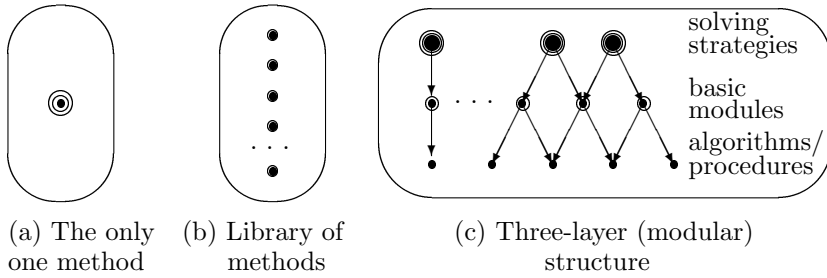


Figure 2.8 Structures of solving method space

2.3.5 Multicriteria Ranking

The following approaches/methods are basic ones for multicriteria ranking alternative ones:

- (1) utility function techniques [261, 427];
- (2) AHP [84, 706];
- (3) outranking techniques (*e.g.*, methods ELECTRE, Promethee) [104, 700, 701];
- (4) interactive techniques of multicriteria analysis [446, 449];
- (5) knowledge bases and neural networks [521, 869];
- (6) a special approach on the basis of a *technological viewpoint* (as multi-agent systems, generation of a solving strategy on the basis of libraries of algorithms and procedures) [492, 497, 510];
- (7) hybrid techniques [245, 350].

2.4 Specialist

2.4.1 Role of the Specialist

In previous sections we have considered types of specialist in a problem-solving process (*e.g.*, in decision-making):

- (1) decision maker (main person, who is responsible for all results);
- (2) domain expert (initial data accumulation, generation of alternatives, assessment of alternatives and preference relations);
- (3) specialist for organizational support of decision-making technology;
- (4) knowledge engineer (organization of a cycle knowledge acquisition, knowledge verification, knowledge modification, maintenance, and utilization).

It is reasonable to emphasize the unique role of the specialist in organizational support of decision-making technology. In this kind of technology, an object under analysis and transformation is a very difficult multi-factor decision problem (*e.g.*, engineering, organizational, social-economical, ecological, political). Note here that the contemporary viewpoint is targeted to *intellectual capital* [880].

2.4.2 Professional Levels

The classification of specialists is a crucial issue. Traditional classes of specialists are the following: (a) novice (beginner), (b) professional (competent), and (c) expert. P.J. Denning suggested a prospective seven-level classification of specialists [207]:

1. *Novice* (beginner): (a) common situations are unfamiliar; (b) practice in simple situations.

2. *Advanced beginner* (rookie): (a) problem solving and practice with rules and strategies; (b) need supervising for more complex tasks.

3. *Professional* (competent): (a) practice in many standard situations; (b) advanced problem solving; (c) extensive practice in both common and exceptional situations; (d) apprenticeship to more advanced professionals and teams; (e) membership in professional networks.

4. *Proficient professional* (star): (a) deals with complex situations effortlessly; (b) individual performance is a benchmark for others; (c) considerable experience and practice across a wide range of situations over years of work; (d) apprenticeship to experts; (e) coaching; (f) putting self into wide range of situations; (g) membership and contribution to professional networks.

5. *Expert* (virtuoso): (a) appears to solve difficult, complex problem effortlessly; (b) enormous breadth and depth of knowledge; (c) routinely forms and leads high-performance teams; (d) admired by others as a benchmark of team performance; (e) performance standards are beyond those of most practitioners; (f) apprenticeship to masters; (g) advanced coaching, development of breadth; (h) teaches others; (i) years or decades of practice.

6. *Master*: (a) capacity for long-range strategic thinking and actions; (b) sees historical drifts and shifting clearings; (c) has studied with many different teachers and has developed own distinctive style; (d) has produced innovations in the standard practices of others; (e) altered the course of history in the field, and knows how to do this again; (f) teaches others to be experts and masters; (g) learning continues by working with other masters as teachers; (h) creates and leads professional networks; (i) teaches others.

7. *Legend*: (a) work has widely accepted impact; (b) same as for master with emphasis on public appearance.

2.5 Solving Frameworks

Now, the selection/design of algorithms and planning the solving processes are central and the most important research directions. For example, intelligent technologies are considered as searching for algorithms for dynamical optimization problems (including flexible execution strategies, *e.g.*, any-time algorithms) [454, 910]. The basic trend is targeted to *problem-solving environments* [685].

2.5.1 Search Spaces and Strategies

State-space search is the basic approach to problem solving and design [760, 903] on the basis of the following components: (1) goals, (2) state spaces, and

(3) operations for various design domains and design problems. The search methodology can be considered as fundamental in several domains, *e.g.*, mathematics, computer science, operations research and its significance is increased [430]. Here, the key problems are the following: (i) design of search space; (ii) design of a search strategy in the search space. Let us list the basic search strategies as follows:

1. Constraint-based space-decreasing methods [850] (Figure 2.9).
2. Gradient-like methods. Figure 2.10 depicts a search space and an improvement search strategy (*e.g.*, local optimization, greedy algorithms, k -exchange technique).
3. Grid-based methods (*e.g.*, design method of parameter space investigation (PSI) suggested by I.M. Sobol and R.B. Statnikov [768, 787] (Figure 2.11).
4. Region-approximation-based methods (*e.g.*, spanning tree-based algorithms, ellipsoid method).
5. Decomposition-based methods: dynamic programming approach as series multistage design of solution parts, step-by-step.
6. Parallel methods (Figure 2.12).
7. Random methods (*e.g.*, random walks).
8. Partitioning–synthesis methods on the basis of HMMD (Figure 2.13).
9. Hybrid strategies.

In this book, the partitioning–synthesis method is the basic one used: (a) partitioning the initial search space into subspaces; (b) search for the best local decision for each subspace; and (c) combination (composition, synthesis) of the local decisions into the global resultant decision.

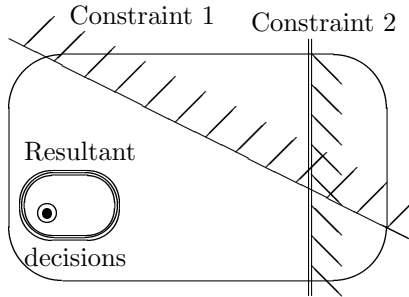


Figure 2.9 Constraint-based strategy

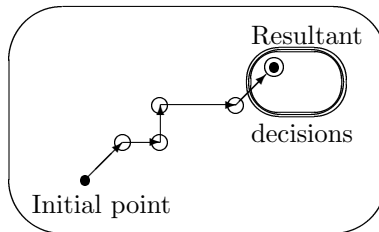


Figure 2.10 Improvement strategy

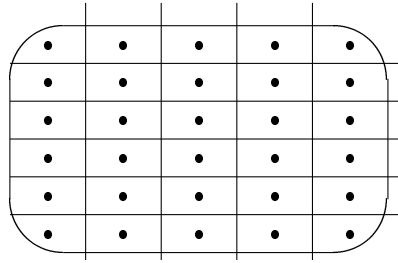


Figure 2.11 Grid-based strategy

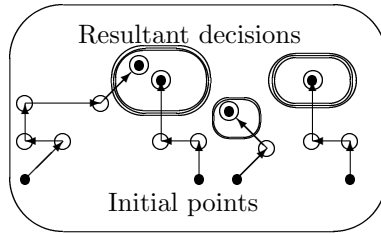


Figure 2.12 Parallel strategy

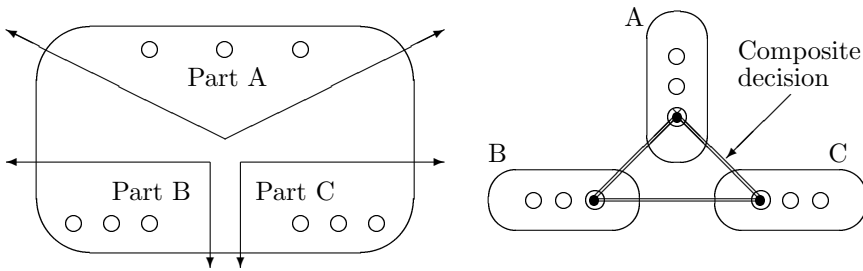


Figure 2.13 Partitioning-synthesis strategy

2.5.2 Heuristics

In recent decades, the significance of heuristic approaches has increased. The situation is based on the fact that the majority of real applied problems are very complicated and require composite heuristic solving schemes which can be considered as hierarchical multi-stage algorithm systems. Thus, it is reasonable to examine a systems engineering approach to the design of algorithm systems.

The list of traditional well-known heuristic approaches involves the following [42, 61, 67, 127, 281, 306–308, 312, 377, 379, 424, 473, 537, 569, 643, 675, 679, 844]: 1. Lagrangian relaxation; 2. greedy algorithms; 3. tabu search; 4. simulated annealing; 5. artificial neural networks; 6. approximation solving

schemes; 7. genetic algorithms and evolutionary computing; 8. constraint satisfaction approach; and 9. probabilistic approaches. In my opinion, a basic trend in the field of heuristics design is the following:

FROM algorithms (*e.g.* [61]) **TO** algorithm frameworks (*e.g.* [602]).

A three-stage algorithm framework is described for a sports timetabling problem by C.L. Nemhauser and M.A. Trick [602]. Multi-stage algorithm systems have been proposed, for example, in [307, 308]. Moreover, the heuristic algorithm systems can be considered as specific problem-solving environments which have been studied in recent years [215, 280, 685]. Here, the structure of the systems, the kinds and the basic algorithm frameworks are proposed which can be considered as a step to the design of algorithm systems.

2.5.3 Metaheuristics

Many real-work problems require a complex solving approach, such as *meta-heuristics* [127, 306] and *hyper-heuristics* [430]. The basic materials and surveys on metaheuristics are contained in the following sources: [61, 66, 67, 140–142, 172, 280, 281, 440, 497, 643, 675, 679, 684, 761, 863, 911]. The list of main metaheuristic kinds involves the following: 1. ant colony optimization; 2. tabu search; 3. simulated annealing; 4. genetic/evolutionary computing including evolutionary multiobjective optimization; 5. GRASP; 6. memetic algorithms; 7. neural networks; 8. variable neighborhood search; 9. reactive search; 10. scatter search; 11. anytime algorithms; and 12. hybrid methods.

2.5.4 Algorithm Systems

Here, let us consider structures of algorithm systems [498]. The essence of the contemporary novelty in this field is based on the following:

effectiveness of simple algorithmic schemes is limited and it is reasonable to build more complex multi-level, hierarchical solving frameworks which involve series-parallel and adaptive solving process.

The structure of the algorithm system can be analyzed as follows:

1. Control unit (planning, adaptation): (a) selection of corresponding basic algorithms, and (b) design of composite solving strategy.
2. Level of execution: (a) analysis of an initial problem, (b) execution of a solving process, and (c) analysis of results.
3. Level of bases/repositories: (a) problems and examples, (b) algorithms and procedures, and (c) solving strategies.

Here, let us examine the basic kinds of algorithm systems structures. First, let us point out the main parameters of the algorithm systems:

1. *Kind of algorithm base*: (a) one-algorithm framework; (b) algorithm set with parallel execution of the algorithms; and (c) complex multistage algorithm framework (*e.g.*, series, series-parallel).

2. *Feedback*: (a) without feedback; (b) with feedback of two types as follows: (i) analysis of resultant solution; and/or (ii) analysis of intermediate results.

3. *Kind of algorithm framework (scheme, composite strategy) synthesis*: (a) fixed algorithm framework (algorithm strategy); (b) flexible design of algorithm framework (*e.g.*, selection of standard algorithm strategies, synthesis of new algorithm frameworks).

Thus, the following functional operations have to be examined: (1) analysis of problem data (input), (2) solving process, (3) analysis of results, (4) modification/selection of algorithm, (5) selection/synthesis of an algorithm set, (6) aggregation of results, and (7) selection/synthesis of an algorithm scheme.

Clearly, the simplest solving scheme involves the initial data, the solving process, the result, and the analysis of the result. Figures 2.14, 2.15, 2.16, and 2.17 depict possible more complicated situations (*e.g.*, parametric modification/selection of algorithm, selection/design of algorithm scheme).

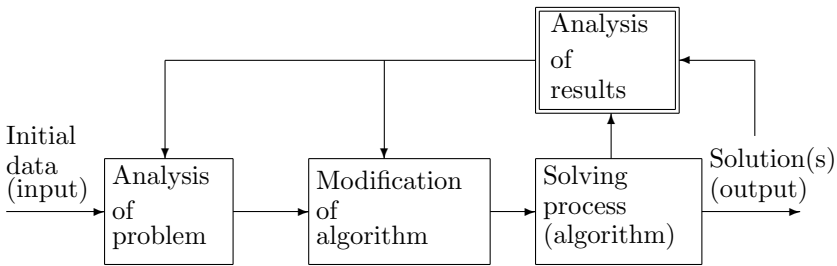


Figure 2.14 Simple one-algorithm framework

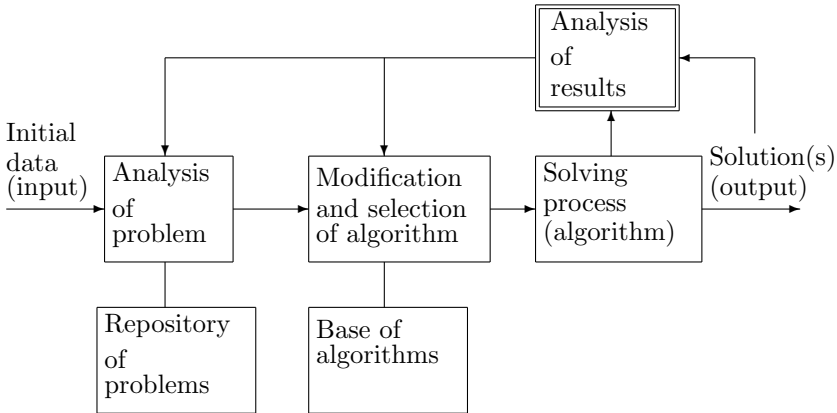


Figure 2.15 One-algorithm framework with algorithm base

Note, it is possible to add into the multi-algorithm framework at Figure 2.15 a block for aggregation of results which were obtained on the basis of different algorithms.

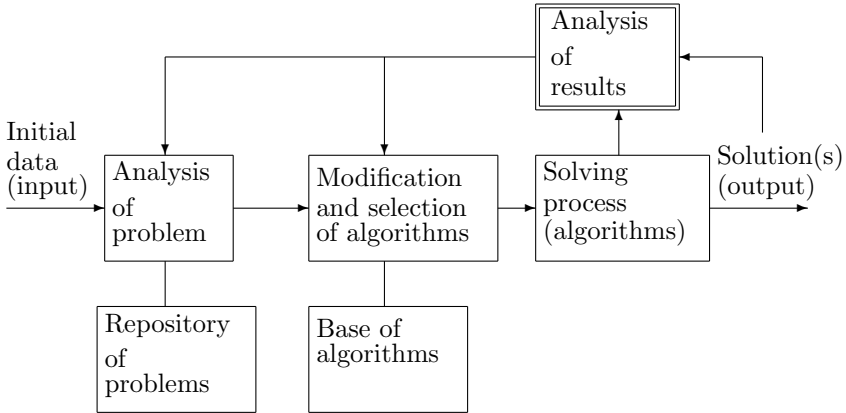


Figure 2.16 Multi-algorithm framework with algorithm base

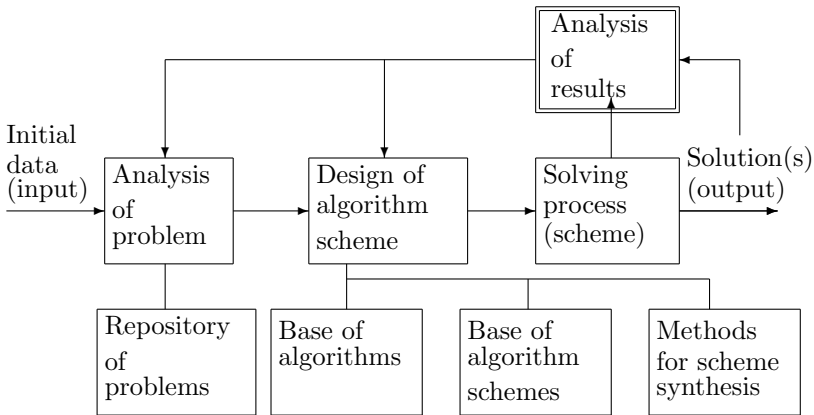


Figure 2.17 Multi-algorithm scheme framework

2.6 Summary

We have examined some components of information technology: (a) information, (b) knowledge acquisition/management, (c) decision-making framework, (d) specialists, and (e) solving strategies. A successful design for complex composite decisions is based on background and experience in the components above.



<http://www.springer.com/978-1-84628-001-6>

Composite Systems Decisions

Levin, M.S.

2006, XVI, 254 p. 126 illus., Hardcover

ISBN: 978-1-84628-001-6