Computer Assisted Mechanics and Engineering Sciences, 9: 105–121, 2002.

Towards personalised software in machine design

Jerzy Pokojski
Institute of Machine Design Fundamentals, Warsaw University of Technology
ul. Narbutta 84, 02-524 Warsaw, Poland

(Received September 7, 2001)

The paper presents a concept of knowledge based software supporting a long period machine design analysis – intelligent personal assistant. The concept of intelligent personal assistant is based on the maze model and optimisation model of the design process. The functional structure of the whole system is shown.

1. INTRODUCTION

The last thirty years have brought a relatively high development of computer tools supporting engineers in their design activities. CAD, CAE systems have become industrial standards. Many other engineering problems like simulation or analysis have been supported by commercial and self made software as well.

Design teams, working on their projects, use various computer systems and many computer codes. These systems and codes often function separately. They are not integrated and exist in several versions. The number of computer tools used by a design team can be as high as a few thousands [3, 9, 27, 54].

Each designer can use the whole plurality of computer tools in his own way [1, 2, 4–7, 10–13, 16, 28, 30, 45, 46, 55, 56, 58]. That is what we can also observe with design teams [8, 22–24, 45, 46, 57].

Designers' work is very individual [34, 35, 44]. Designers use their own approaches and their own methods. They have an individual way of seeing a problem. When we observe how these people solve their problems; we first notice that they use mental modelling, mental problem identification and mental problem solving procedures. They do mental exercises with the problem. For them design can exist as a project in their heads or as a project on the paper (or as a project in computer memory). The information printed on the paper or stored on the disk doesn't reflect the whole information concerning the particular project. A lot of important knowledge remains in designers' heads.

Designers have a wide range of tools supporting the design process. Computer tools are one of them. Information processed by computer tools is not always integrated. But the integration of computer tools can make the whole design process more efficient. In addition the plurality of the existing software tools supporting the design process allows to create the design process in a very flexible way. When we look at the actual state of the development of computer tools supporting the design process, accepted by the designers, we see quite a wide plurality and flexibility. Nevertheless all these features allow the user to maintain an individual personalised character of the whole computer support.

The problem of fast, effective and flexible integration of design tools is still actual. It seems to be the only way to exploit new achievements in science. Some authors underline the contradiction between a very precise analysis dealing with details and a very rough analysis of global problems and global dependencies.

We can observe a growing number of commercial systems offering the possibility of fast and efficient integration [3, 9, 26, 27]. These tools enable the user to integrate computer systems fast and flexibly. They make it possible to store processes, repeat processes, optimise the whole process and to make parallel analysis with alternative models or software.

The trend mentioned above concentrates on the aspect of "how" a particular design process was achieved. During the last years many research teams have worked on the methodology and on tools whose main goal is to store the "why" of the procedure [45, 51, 52].

The information which is behind the project is called design rationale. During the last twenty years a lot of research has been accomplished in this field. Some approaches and applications are known already. Most of them were done ad hoc. Later the problem of data exchange between different applications appeared.

Consequently the standardisation of design rationale information became the next step of this development. We can observe attempts of creating knowledge oriented standards which are the natural development of data oriented standards (like STEP, IGES) [51, 52].

The acquired knowledge can be re-used by other designers, as it is accessible in specially made design knowledge repositories [51, 52].

The main goal of collecting information, which is behind some decisions, is to exploit thoroughly existing knowledge, and to better organise the work of co-operating teams enabling world-wide distribution [14, 22, 23, 29].

As a result the following problems occur: how to motivate people to share their knowledge with others; who should do the knowledge acquisition; who should do the servicing and management of the design knowledge repositories. But all the tasks mentioned above require additional effort. Our concept, presented in this paper, is based on the idea of an intelligent personal assistant [5, 6, 11–13, 38–44]. We assumed that knowledge is delivered, serviced and managed by the designer personally for himself, in his own personal computer notebook and for his own personal purposes.

Every designer uses some kind of notes in his (her) professional practice. These notes are stored in some paper notebooks. The information stored in the notebook doesn't need any formal structure. It can be composed of different data pieces. The data coming to the notebook are stored chronologically. The system of searching for some data is based on a similar idea. Sometimes keywords are important. They can be connected with a particular project, the kind of analysis, results, comments, dates – functioning as headers and others. Designers often make notes which do not only deal with the actual problems but which are also more general comments to some technical problems and situations. They store their conceptual visions of products or some aspects of their functioning.

Designer's notes often contain comments to other data structures like formal project documentation, formal analysis, formal contacts, etc. And in most of the cases paper notebooks are treated as personal tools.

Today designers use computer systems supporting the process of designing. The results of the design process are stored on computers with the help of some standard formats.

Designers who have to solve new design problems use their paper notes. The selected content of the notes is again transformed to a practical form – to some piece of action in the design process, and it gives some valuable results. The whole procedure described above is obviously done by the designer personally. Looking at the process from the operation point of view we notice

1. knowledge storage – complete and incomplete form in the notebook,
2. knowledge retrieval from the notebook,
3. knowledge application in the design process.

The expert knowledge can be acquired and used in some computer knowledge based system. We can imagine the situation that an expert adds knowledge (created by him) to an existing knowledge base in a dynamic way for his own (expert) purposes. In this case the system fulfils the role of the expert's active notes. This knowledge may contain comments to different models, parameters, achieved results and successful plans etc. The stored knowledge can be used by the expert himself and the people collaborating with him.

Let's assume we not only want to collect the text documents but we also want the user's personal opinions, his personal knowledge chunks which have different representations. The users can have different associations connected with different pieces of knowledge or information.

They employ different computer tools. These tools have some structure, parameters, model of user's profile etc. Often tools can work as a sequence of tools when there is some know-how to organise this process and to build data exchange connections. All the above information can have practical value for the user.

In the paper we try to connect two ideas:

1. computer system integrating computer design tools,
2. computer system fulfilling the role of an intelligent personal assistant.

The paper presents concepts of a knowledge based system supporting machine design called intelligent personal assistant.

Today there are several papers presenting software which can be classified as an intelligent personal assistant [1, 7, 10–13, 16–18, 55, 56]. Some of these systems are specialised and intended for specific domains, others are more universal and often simpler. Each system has its own concept and goal of its development.

The following Sections show concepts of systems representing a class of intelligent personal assistants in the domain of machine design.

2. THE MAZE MODEL OF DESIGN PROCESS

In our concept of an intelligent personal assistant we assume that the system will be based on the ideas of the maze [2, 38] (Fig. 1) and optimisation [31–33] (Fig. 2) models of the design process.

The maze model is a system of nodes connected by links. Every node connects some important activity (Fig. 3). Activities can be procedures of some model generation, analysis etc. Links reflect the possibility of moving from one activity to another. The maze model has an open structure. The design process can be initiated from different initial nodes. It can go via feasible paths, and it can finish in different nodes. New nodes and links can be added to the actual system (Fig. 4). Nodes and links which are not needed any more can be removed. The actual state of the maze can be stored as a plan (Fig. 5). The stored plans can be loaded. We can store plans which were used in any design process. Later we can return to the processes which were used earlier (Fig. 6). Activities, computer codes or commercial applications can be connected to the maze model as new nodes. Data integration can be achieved with different tools, starting from data files and finishing with the blackboard architecture [34, 36, 37].

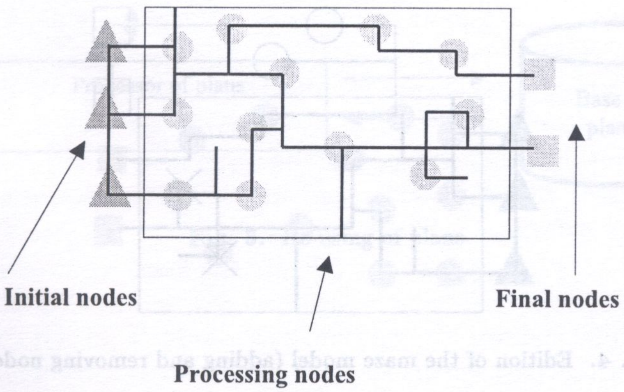


Fig. 1. The maze model of design process

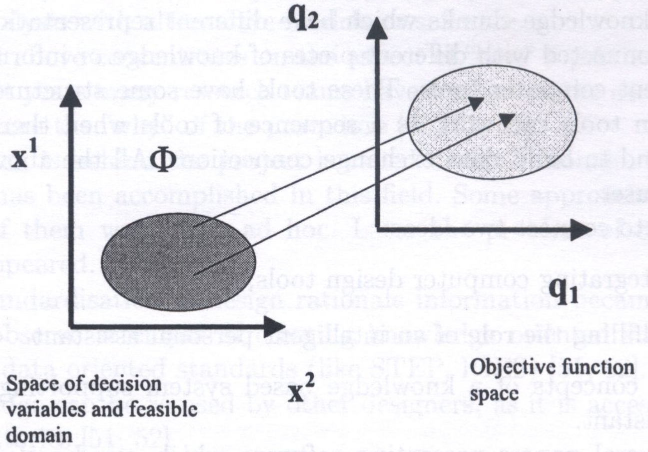


Fig. 2. The optimization model of design process

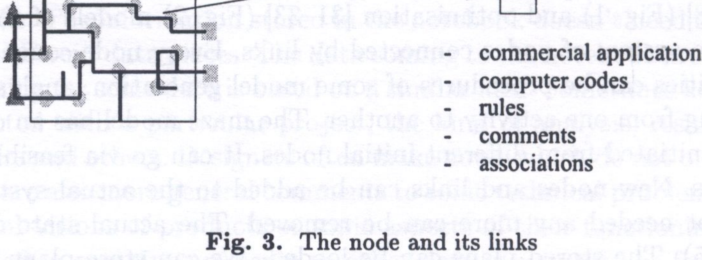


Fig. 3. The node and its links

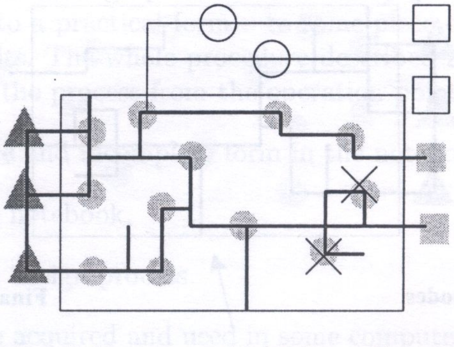


Fig. 4. Edition of the maze model (adding and removing nodes)

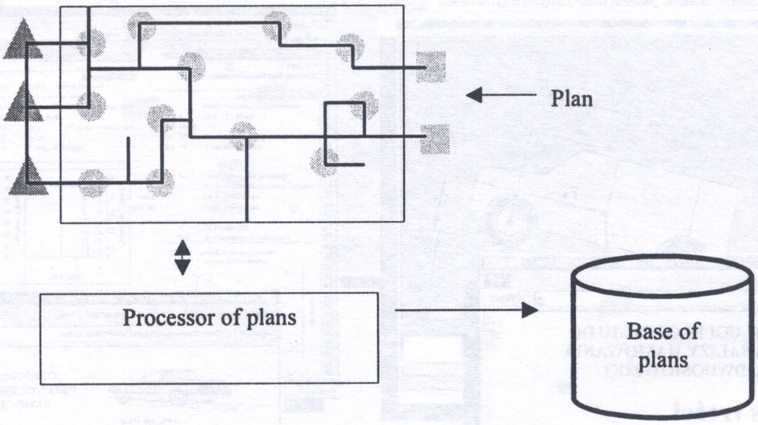


Fig. 5. Storage of plans

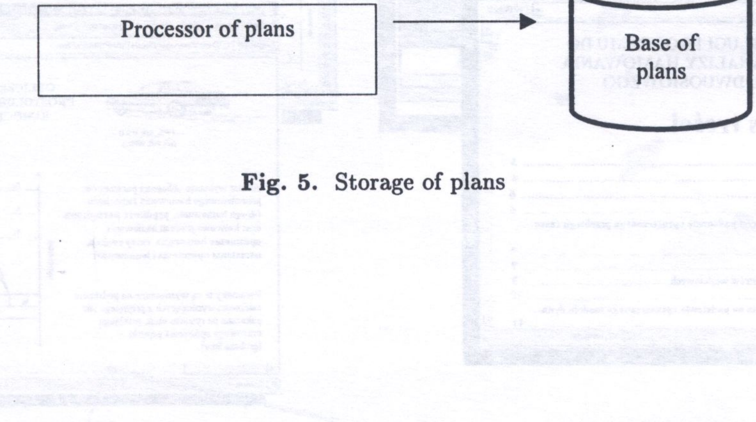
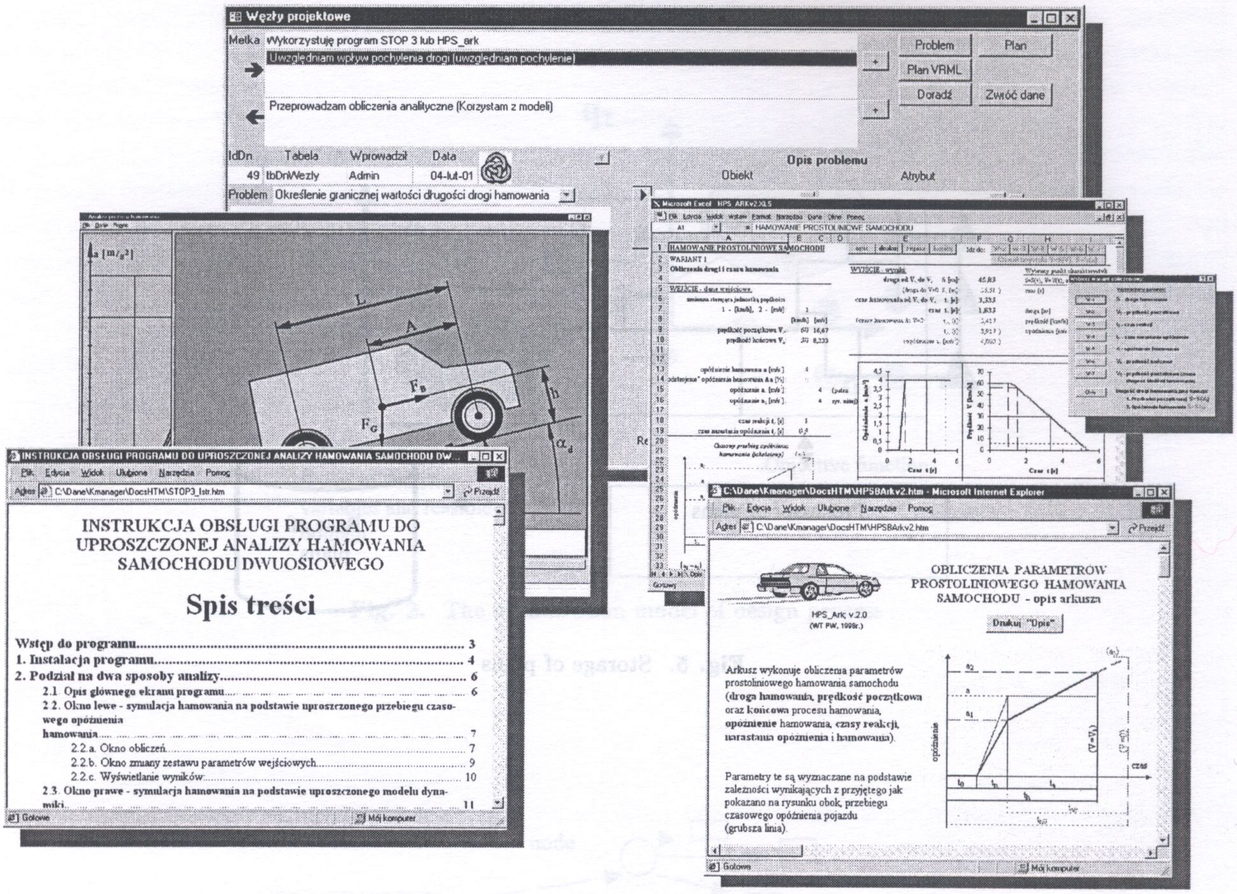


Fig. 6. Re-using of plans



Metka Wykorzystuję program STOP 3 lub HPS_ark

Uwzględniam wpływ pochylenia drogi (uwzględniam pochylenie)

Przeprowadzam obliczenia analityczne (Korzystam z modeli)

IdDn	Tabela	Wprowadził	Data
49	tbDnWezly	Admin	04-lut-01

Problem Określenie granicznej wartości długości drogi hamowania

Adres C:\Dane\Kmanager\DocsHTM\STOP3_Istr.htm

INSTRUKCJA OBSLUGI PROGRAMU DO
UPROSZCZONEJ ANALIZY HAMOWANIA
SAMOCHODU DWUOSIOWEGO

Spis treści

Wstęp do programu........ 3
1. Instalacja programu........ 4
2. Podział na dwa sposoby analizy........ 6
  2.1. Opis głównego ekranu programu........ 6
  2.2. Okno lewe - symulacja hamowania na podstawie uproszczonego przebiegu czasowego opóźnienia hamowania........ 7
    2.2.a. Okno obliczeń........ 7
    2.2.b. Okno zmiany zestawu parametrów wejściowych........ 9
    2.2.c. Wyświetlanie wyników........ 10
  2.3. Okno prawe - symulacja hamowania na podstawie uproszczonego modelu dynamiki........ 11

Adres C:\Dane\Kmanager\DocsHTM\HPSBArkv2.htm

OBLICZENIA PARAMETROW
PROSTOLINIOWEGO HAMOWANIA
SAMOCHODU - opis arkusza

HPS_Ark; v.2.0
(WT PW, 1998r.)

Arkusz wykonuje obliczenia parametrów prostoliniowego hamowania samochodu (**droga hamowania, prędkość początkowa** oraz **końcowa** procesu hamowania, **opóźnienie** hamowania, **czasy reakcji, narastania opóźnienia i hamowania**).

Parametry te są wyznaczane na podstawie zależności wynikających z przyjętego jak pokazano na rysunku obok, przebiegu czasowego opóźnienia pojazdu (grubsza linia).

Fig. 7. Multi-window graphic interface

The whole software structure of the maze model can be done in one of the following technologies [39, 43]: 1) as relational data base, 2) as an expert system, 3) as dedicated application written in programming language. In each case the user interface can be presented in the typical multi-window form (Fig. 7).

The problem of navigation in the maze model is supported by a special visualisation module based on virtual reality software [39], where nodes are bodies and links are pipes (Fig. 8). By changing the visualisation parameters it is possible to analyse and catch the whole structure of the maze model.

The maze model is quite useful in conceptual design, where the designer concentrates on some important aspects of a problem which is directly solved by him and where he quickly wants to restart and to build processes with a more complicated structure [38, 39, 42–44].

The whole procedure of moving in the maze model has, as mentioned above, an operational aspect and an inferencing (knowledge based) aspect. Parallel to the moving in the maze model the designer has to make decisions; what to do next, how to select parameters and how to evaluate results. He supports his inferencing with his knowledge. The main problem is how to operate efficiently with an environment which allows different actions and where different actions can be supported with tools helping to create new solutions or adapt past solutions.

The process of moving from one node to the other in the maze requires testing to assure that all data really exist. If not, a special procedure is initiated and the missing data are created. Tools of artificial intelligence can fulfil the role of data generators. In [4, 5, 38, 43] expert systems were used to support this process. In this case expert systems contain knowledge which provides the missing data. The knowledge bases contain domain knowledge connected with a particular node. In [43] modules supporting particular node applications were the following (Figs. 9 and 10):

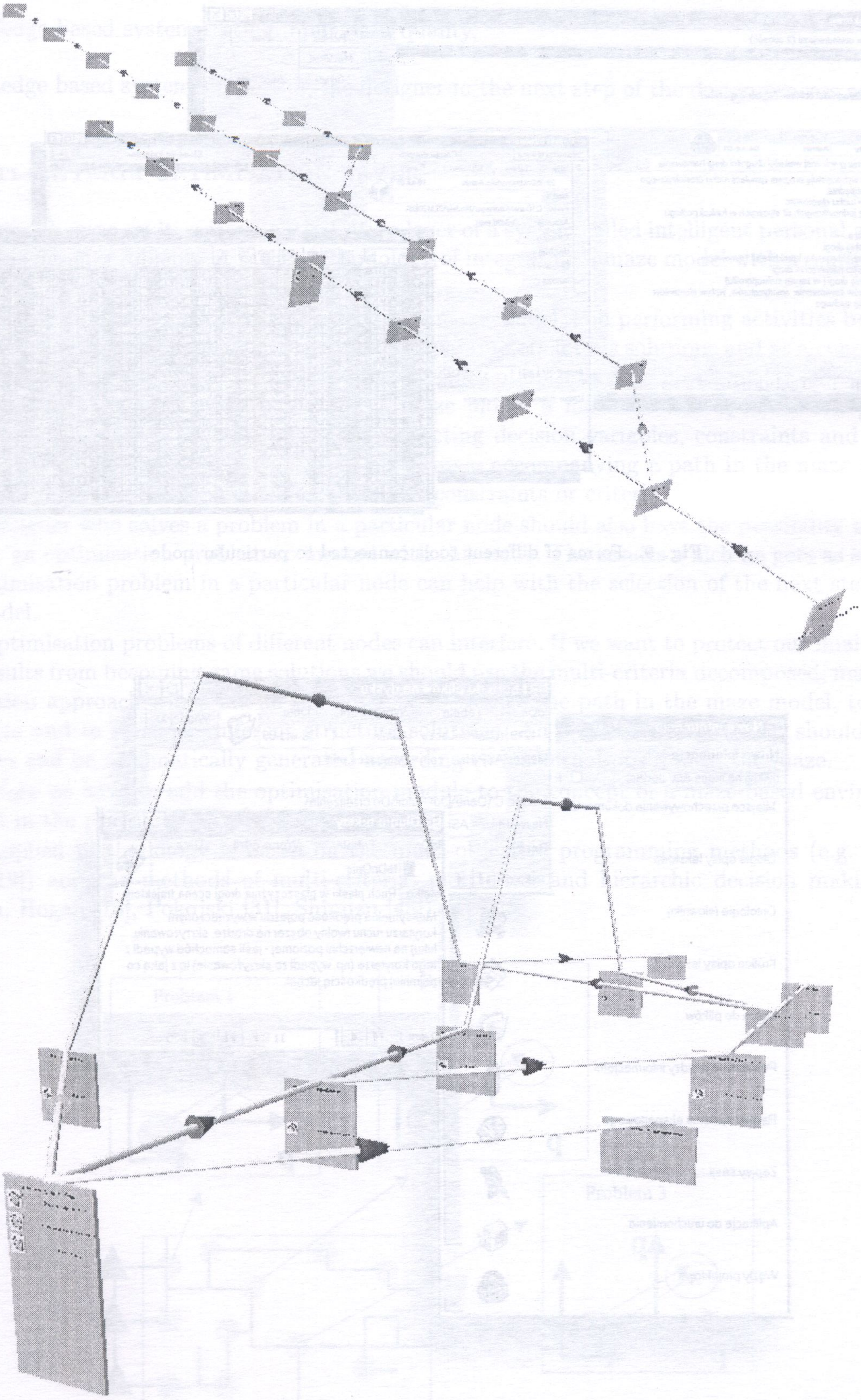
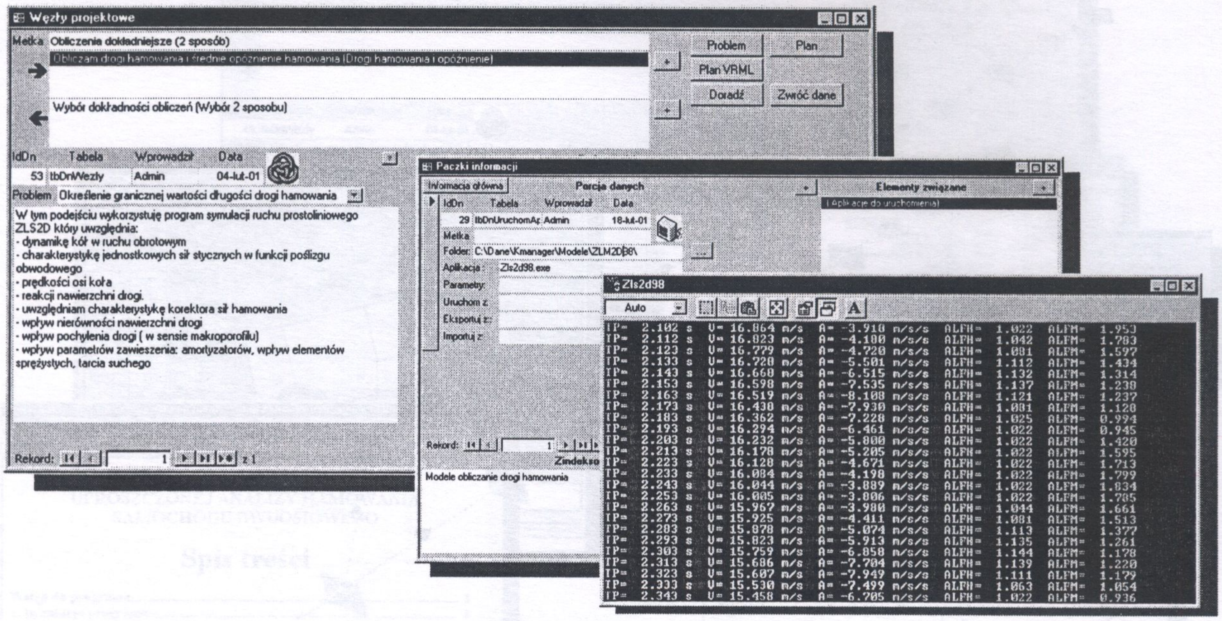


Fig. 8. Schemes of plans made in virtual reality tools



Metka: Obliczenia dokładniejsze (2 sposób)

Obliczam drogi hamowania i średnie opóźnienie hamowania (Drogi hamowania i opóźnienie)

Wybór dokładności obliczeń (Wybór 2 sposobu)

IdDn	Tabela	Wprowadził	Data
53	tbDnWezly	Admin	04-lut-01

Problem: Określenie granicznej wartości długości drogi hamowania

W tym podejściu wykorzystuję program symulacji ruchu prostoliniowego ZLS2D który uwzględnia:
- dynamikę kół w ruchu obrotowym
- charakterystykę jednostkowych sił stycznych w funkcji poślizgu obwodowego
- prędkości osi koła
- reakcji nawierzchni drogi.
- uwzględniam charakterystykę korektora sił hamowania
- wpływ nierówności nawierzchni drogi
- wpływ pochylenia drogi (w sensie makroporofilu)
- wpływ parametrów zawieszenia: amortyzatorów, wpływ elementów sprężystych, tarcia suchego

Rekord: 1 z 1

IdDn	Tabela	Wprowadził	Data
29	tbDnUruchomAp	Admin	18-lut-01

Metka:
Folder: C:\Dane\Kmanager\Modele\ZLM2D98\
Aplikacja: Zls2d98.exe
Parametry:
Uruchom z:
Eksportuj z:
Importuj z:

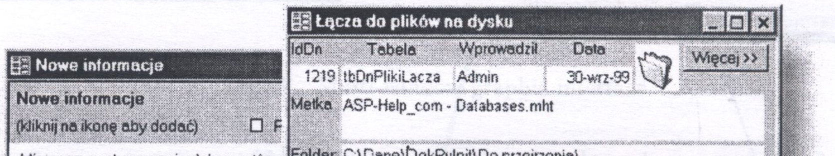
Rekord: 1

(Aplikacje do uruchomienia)

Modele obliczanie drogi hamowania

```
TP=  2.102 s  V= 16.864 m/s  A= -3.910 m/s/s  ALFH=  1.022  ALFM=  1.953
TP=  2.112 s  V= 16.823 m/s  A= -4.180 m/s/s  ALFH=  1.042  ALFM=  1.783
TP=  2.123 s  V= 16.779 m/s  A= -4.720 m/s/s  ALFH=  1.081  ALFM=  1.597
TP=  2.133 s  V= 16.720 m/s  A= -5.501 m/s/s  ALFH=  1.112  ALFM=  1.434
TP=  2.143 s  V= 16.660 m/s  A= -6.515 m/s/s  ALFH=  1.132  ALFM=  1.314
TP=  2.153 s  V= 16.598 m/s  A= -7.535 m/s/s  ALFH=  1.137  ALFM=  1.230
TP=  2.163 s  V= 16.519 m/s  A= -8.108 m/s/s  ALFH=  1.121  ALFM=  1.237
TP=  2.173 s  V= 16.430 m/s  A= -7.930 m/s/s  ALFH=  1.081  ALFM=  1.120
TP=  2.183 s  V= 16.362 m/s  A= -7.228 m/s/s  ALFH=  1.025  ALFM=  0.994
TP=  2.193 s  V= 16.294 m/s  A= -6.461 m/s/s  ALFH=  1.022  ALFM=  0.945
TP=  2.203 s  V= 16.232 m/s  A= -5.800 m/s/s  ALFH=  1.022  ALFM=  1.420
TP=  2.213 s  V= 16.178 m/s  A= -5.205 m/s/s  ALFH=  1.022  ALFM=  1.595
TP=  2.223 s  V= 16.128 m/s  A= -4.671 m/s/s  ALFH=  1.022  ALFM=  1.713
TP=  2.233 s  V= 16.084 m/s  A= -4.198 m/s/s  ALFH=  1.022  ALFM=  1.799
TP=  2.243 s  V= 16.044 m/s  A= -3.889 m/s/s  ALFH=  1.022  ALFM=  1.834
TP=  2.253 s  V= 16.005 m/s  A= -3.806 m/s/s  ALFH=  1.022  ALFM=  1.785
TP=  2.263 s  V= 15.967 m/s  A= -3.980 m/s/s  ALFH=  1.044  ALFM=  1.661
TP=  2.273 s  V= 15.925 m/s  A= -4.411 m/s/s  ALFH=  1.081  ALFM=  1.513
TP=  2.283 s  V= 15.878 m/s  A= -5.074 m/s/s  ALFH=  1.113  ALFM=  1.377
TP=  2.293 s  V= 15.823 m/s  A= -5.913 m/s/s  ALFH=  1.135  ALFM=  1.261
TP=  2.303 s  V= 15.759 m/s  A= -6.858 m/s/s  ALFH=  1.144  ALFM=  1.178
TP=  2.313 s  V= 15.686 m/s  A= -7.704 m/s/s  ALFH=  1.139  ALFM=  1.220
TP=  2.323 s  V= 15.607 m/s  A= -7.949 m/s/s  ALFH=  1.111  ALFM=  1.179
TP=  2.333 s  V= 15.530 m/s  A= -7.499 m/s/s  ALFH=  1.063  ALFM=  1.054
TP=  2.343 s  V= 15.458 m/s  A= -6.705 m/s/s  ALFH=  1.022  ALFM=  0.936
```

Fig. 9. Forms of different tools connected to particular node



IdDn	Tabela	Wprowadził	Data
1219	tbDnPlikiLacza	Admin	30-wrz-99

Metka: ASP-Help_com - Databases.mht
Folder: C:\Dane\DokPulpit\Do przeirzenia\
NazwaPliku: ASI
Rozmiar:

IdDn	Tabela	Wprowadził	Data
11	tbDnOpis		

Metka: Ruch płaski w płaszczyźnie drogi ocena trajektorii

Maksymalna prędkość pojazdu w wyznaczonym korytarzu ruchu (wolny obszar na drodze, skrzyżowaniu, łuku) na nawierzchni poziomej - jeśli samochód wypadł z tego korytarza (np. wypadł ze skrzyżowania) to z jaką co najmniej prędkością jechał.

Rekord: 11 z 35

Fig. 10. Notes retrieved from the system

- knowledge based system which supports missing data generation,
- knowledge based system testing input data quality,
- knowledge based system supporting the designer in the next step of the design process selection.

3. MULTI-CRITERIA OPTIMISATION LAYER

This Section shows a multi-criteria optimisation layer of a system called intelligent personal assistant for machine design problems. It is the methodology of integrating a maze model with a multi-criteria optimisation (Fig. 11).

Parallel to the process of creating a path in the maze model, and performing activities belonging to this path, the designer can try to select the best parameters for his solutions and as a consequence create a sequence of activities belonging to the category of optimisation problems [31–33] (Fig. 12). We can assume that with every node of the maze model a multi-criteria optimisation problem created by the designer can be connected by selecting decision variables, constraints and criteria function. The sequence of the optimisation problems – accompanying a path in the maze model – can be mutually interacted via decision variables, constraints or criteria.

The designer who solves a problem in a particular node should also have the possibility to create and solve an optimisation problem connected with this node. The results which he gets as solutions of an optimisation problem in a particular node can help with the selection of the next step in the maze model.

But optimisation problems of different nodes can interfere. If we want to protect our final optimisation results from becoming game solutions we should use the multi-criteria decomposed, multi-level optimisation approach which allows the designer to change the path in the maze model, to repeat some parts and to consider different structure solutions. An optimisation problem should always follow him and be automatically generated according to his actual position in the maze.

Therefore we have to add the optimisation module to the concept of a maze-based environment presented in the previous Section.

The applied methodology is based on the multi-objective programming methods (e.g. Hwang, Masud [19]) and the methods of multi-criteria, multi-level and hierarchic decision making (e.g. Geoffrion, Hogan [15], Pokojski [31], Shimizu [48]).

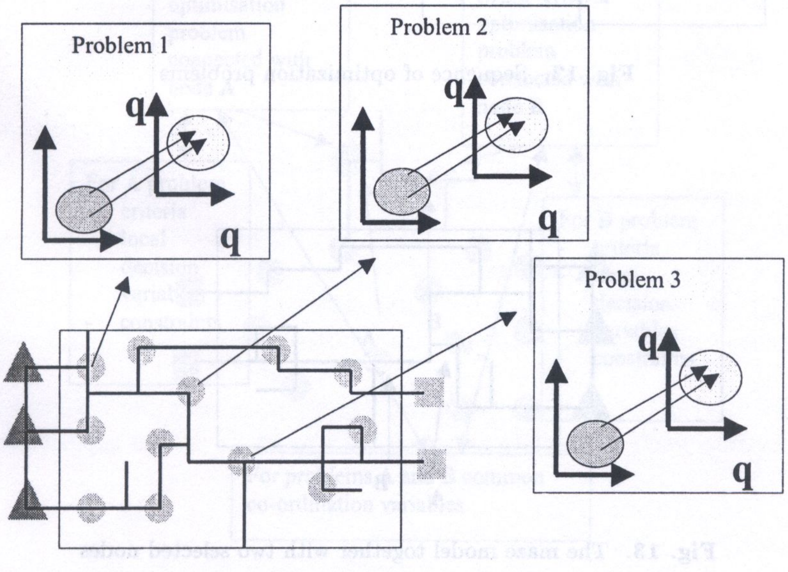


Fig. 11. Optimization models connected to every node on selected path

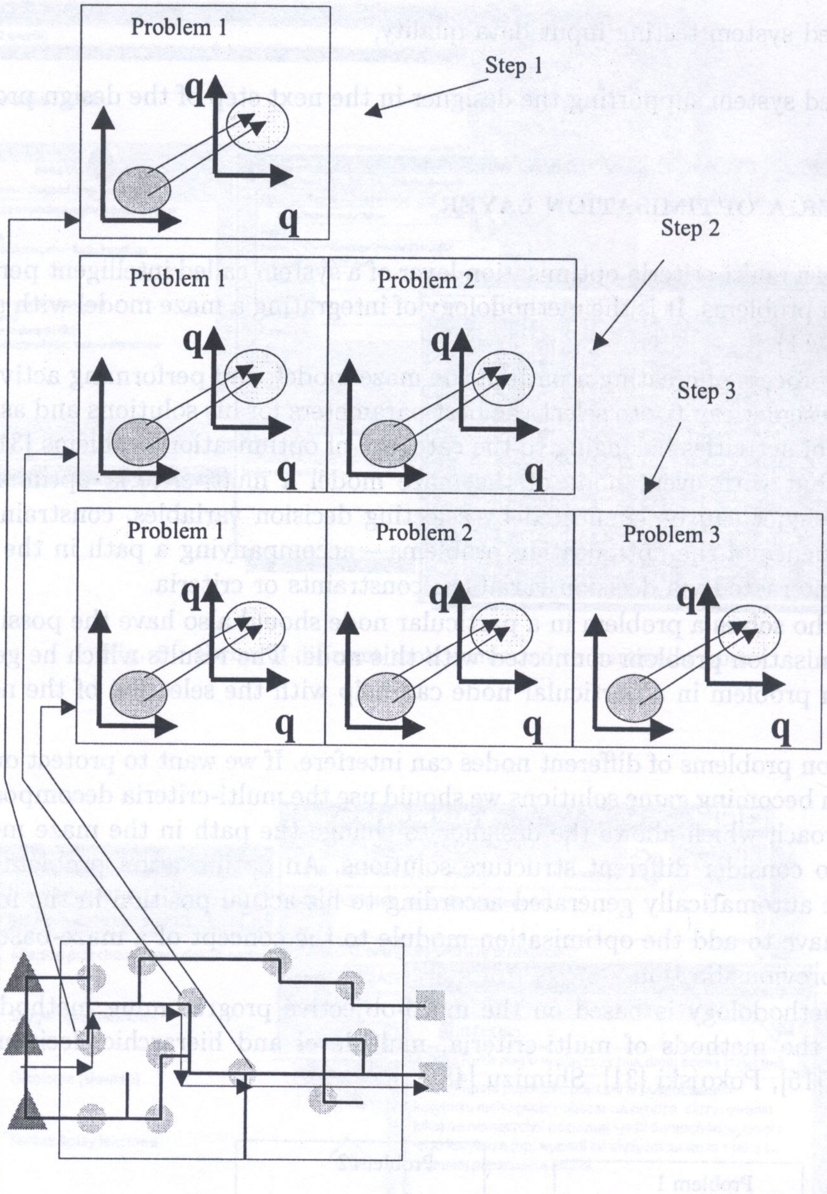


Fig. 12. Sequence of optimization problems

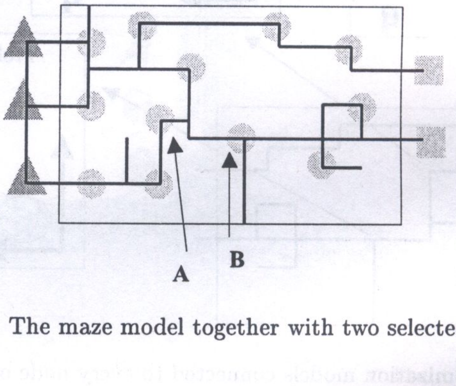


Fig. 13. The maze model together with two selected nodes

Let us consider two optimisation problems connected with two neighbour nodes. Let us assume that the designer started from the node signed with **A** (Fig. 13). Let us assume he formulated an optimisation problem connected with node **A**. And he selected the decision variables, constraints and criteria and solved his problem with the help of one of the multi-criteria optimisation methods and received some satisfying results.

After analysing the whole problem at its actual stage (in node **A**) he decided to solve, as next, a problem connected with node **B**. Then he moved to node **B** in the maze and defined a multi-criteria optimisation problem connected with this node (Figs. 14 and 15). Let us assume that as before he solved his problem (ignoring the decision variables selected in problem **A**) with the help of some multi-criteria optimisation method and received satisfying results.

Now we want to look at the structure of the whole problem. First we solve problem **A**, later problem **B**. The results of problem **A** can influence problem **B**. We can select two groups of decision variables: 1) connected only with a particular node of the maze model, 2) connected with more than one particular node of the maze model. Group 1 we call local decision variables, group 2 we call co-ordination decision variables (Fig. 16).

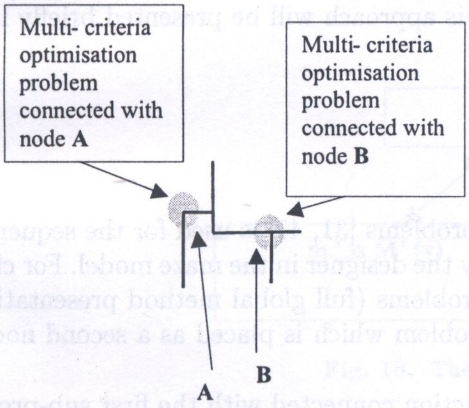


Fig. 14. The maze model together with two connected optimization problems

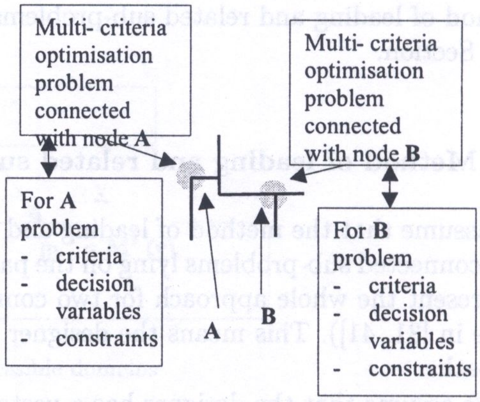


Fig. 15. Two nodes together with their optimization problems

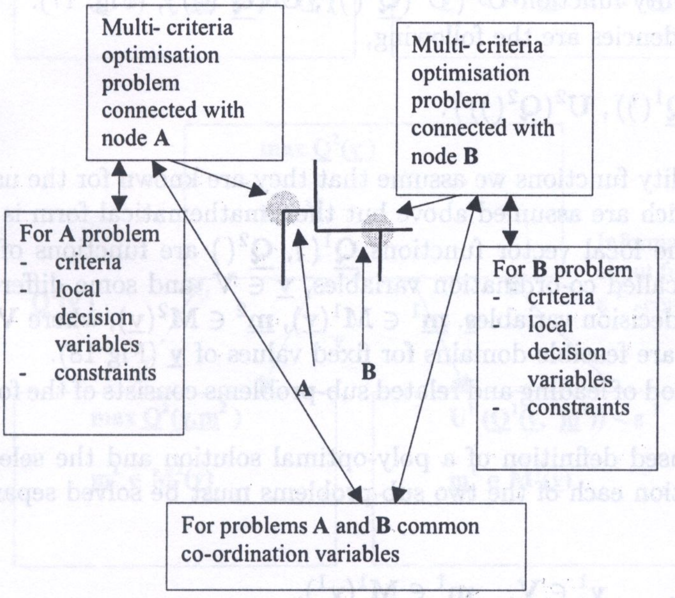


Fig. 16. Two nodes together with structure of decision variables

Looking at our problem from the point of view of the multi-criteria optimisation theory we notice the following conflicts: in the selection of local decision variables for problem **A**, as in the selection of local decision variables for problem **B**, and additionally, in the selection of co-ordination variables which are common for problems **A** and **B**.

The rational solution is to find some compromise. But a compromise can only be found after having solved the whole problem together as one multi-criteria optimisation problem. If we tried to exploit this idea with a maze model it would not be possible because it is assumed that we select our sub-problems while being in the maze model. At the moment when we start to solve the first sub-problem we don't know what the next problem will look like. Results of our work with this sub-problem can influence our decision in the next step – when we go further. If we want to solve our optimisation problem we first have to solve the optimisation problem of the node of the maze model, which we selected first. Later we can select the next node and try to solve the optimisation problem connected with this node, ignoring what was fixed in the first node. As a next step we have to solve the two problems together, analysing the conflicts between them and finding the compromise. To move to the next nodes in the maze model is a similar procedure.

As a way of solving the above optimisation problem the author suggests the application of the method of leading and related sub-problems [31, 41]. This approach will be presented briefly in the next Section.

3.1. Method of leading and related sub-problems

We assume that the method of leading and related sub-problems [31, 41] is used for the sequence of interconnected sub-problems lying on the path selected by the designer in the maze model. For clarity we present the whole approach for two connected sub-problems (full global method presentation is made in [31, 41]). This means the designer solves the problem which is placed as a second node on his path.

We assume that the designer has a vector criteria function connected with the first sub-problem $\underline{\mathbf{Q}}^1(\,)$ and that he has a vector criteria function connected with the second sub-problem $\underline{\mathbf{Q}}^2(\,)$. We also assume that he has clear preferences connected with each of these two sub-problems which can be modelled as utility functions: $\mathbf{U}^1(\underline{\mathbf{Q}}^1(\,))$, $\mathbf{U}^2(\underline{\mathbf{Q}}^2(\,))$, [21]. We assume as well that his preferences dealing with utility functions $\mathbf{U}^1(\underline{\mathbf{Q}}^1(\,))$, $\mathbf{U}^2(\underline{\mathbf{Q}}^2(\,))$, and treated as one problem together, can be modelled as global utility function $\mathbf{U}^G(\,\mathbf{U}^1(\underline{\mathbf{Q}}^1(\,))\,,\,\mathbf{U}^2(\underline{\mathbf{Q}}^2(\,))\,)$, (Fig. 17).

Then global dependencies are the following,

$$\mathbf{U}^G(\,) = \mathbf{U}^G(\,\mathbf{U}^1(\underline{\mathbf{Q}}^1(\,))\,,\,\mathbf{U}^2(\underline{\mathbf{Q}}^2(\,))\,). \tag{1}$$

Concerning the utility functions we assume that they are known for the user implicitly. It means they have features which are assumed above but their mathematical form is not known.

We assume that the local vector functions $\underline{\mathbf{Q}}^1(\,)$, $\underline{\mathbf{Q}}^2(\,)$ are functions of some common vector of decision variables, called co-ordination variables, $\underline{\mathbf{v}} \in \mathbf{V}$, and some different vectors of decision variables, called local decision variables, $\underline{\mathbf{m}}^1 \in \mathbf{M}^1(\underline{\mathbf{v}})$, $\underline{\mathbf{m}}^2 \in \mathbf{M}^2(\underline{\mathbf{v}})$, where $\mathbf{V}$ is the feasible domain for $\underline{\mathbf{v}}$; $\mathbf{M}^1(\underline{\mathbf{v}})$, $\mathbf{M}^2(\underline{\mathbf{v}})$ are feasible domains for fixed values of $\underline{\mathbf{v}}$ (Fig 18).

The proposed method of leading and related sub-problems consists of the following steps (Fig. 19):

1. According to the used definition of a poly-optimal solution and the selected method of multi-objective optimisation each of the two sub-problems must be solved separately from other local sub-problems,

$$\max \underline{\mathbf{Q}}^1(\underline{\mathbf{m}}^1, \underline{\mathbf{v}})\,, \qquad \underline{\mathbf{v}}^1 \in \mathbf{V}, \quad \underline{\mathbf{m}}^1 \in \mathbf{M}^1(\underline{\mathbf{v}}^1), \tag{2}$$

$$\max \underline{\mathbf{Q}}^2(\underline{\mathbf{m}}^2, \underline{\mathbf{v}})\,, \qquad \underline{\mathbf{v}}^2 \in \mathbf{V}, \quad \underline{\mathbf{m}}^2 \in \mathbf{M}^2(\underline{\mathbf{v}}^2). \tag{3}$$

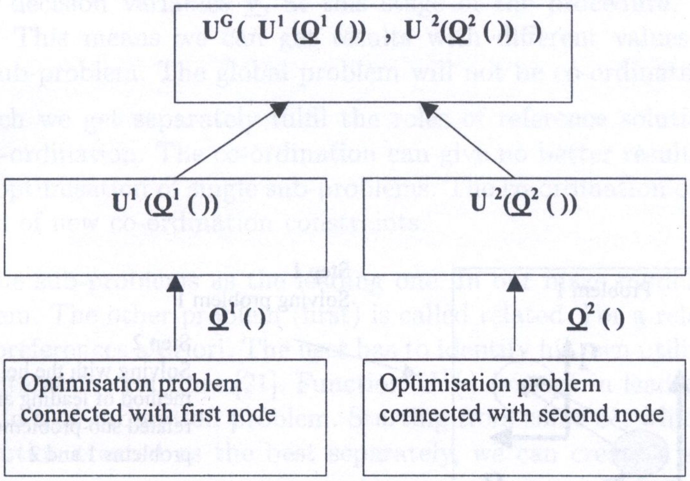


Fig. 17. The structure of utility functions

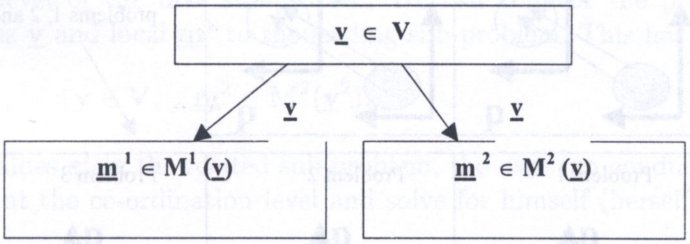


Fig. 18. The structure of feasible domains

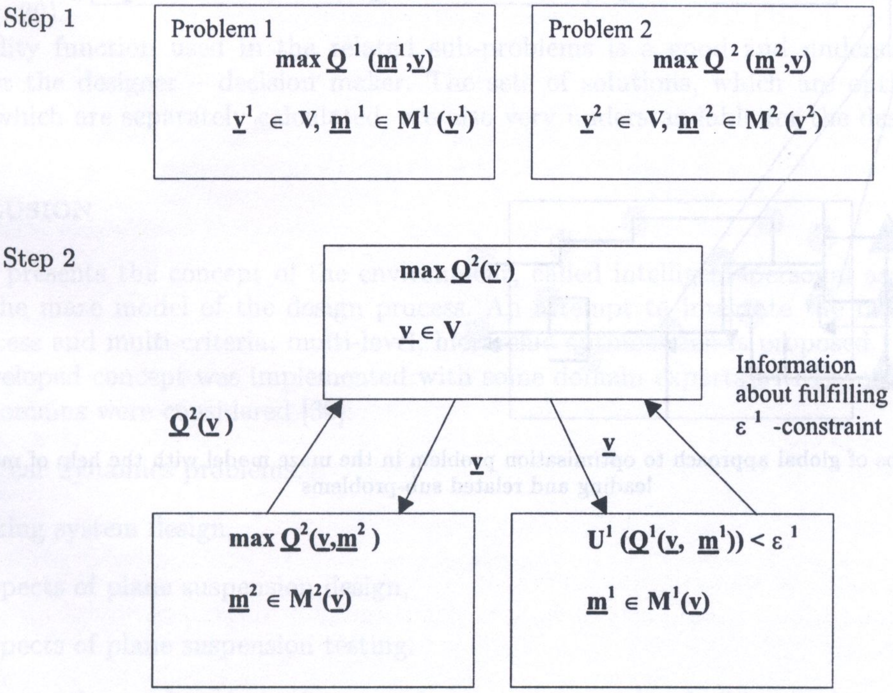


Fig. 19. The method of leading and related sub-problems

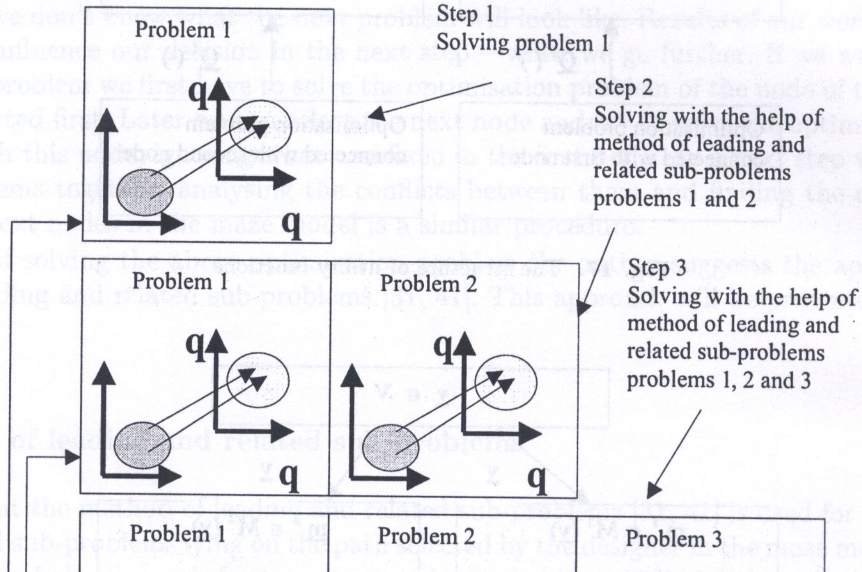


Fig. 20. Steps of global approach to optimisation problem in the maze model with the help of method of leading and related sub-problems

The co-ordination decision variables $\underline{\mathbf{v}}$, at this stage of the procedure, are treated like local decision variables. This means we can get results with different values of the co-ordination variables in each sub-problem. The global problem will not be co-ordinated.

The solutions which we get separately fulfil the roles of reference solutions (utopia point) at the stage of the co-ordination. The co-ordination can give no better results than those achieved from the separate optimisation of single sub-problems. The co-ordination of a single sub-problem means the addition of new co-ordination constraints.

2. We select one of the sub-problems as the leading one. In our maze model it will be the second (last solved) problem. The other problem (first) is called related. For a related problem the user has to express his preferences a'priori. The user has to identify his own utility function $\mathbf{U}^1(\,)$ with the help of some interactive routines [21]. Function $\mathbf{U}^1(\,)$ (except in leading problem) fulfils the role of constraints in a co-ordination problem. Starting from some ε^j, which is worse in Pareto's sense than the solution treated as the best separately, we can create a series of the following problems,

$$\mathbf{U}^1(\underline{\mathbf{Q}}^1(\underline{\mathbf{v}},\underline{\mathbf{m}}^1)) < \varepsilon^1, \qquad \underline{\mathbf{m}}^1 \in \mathbf{M}^1(\underline{\mathbf{v}}), \tag{4}$$

for ε^1 aspiration level of the first sub-problem We can connect the process of selecting co-ordination variables $\underline{\mathbf{v}}$ and local $\underline{\mathbf{m}}^1$ to the leading sub-problem. This has the following form,

$$\max \underline{\mathbf{Q}}^2(\underline{\mathbf{v}},\underline{\mathbf{m}}^2), \qquad \underline{\mathbf{v}} \in \mathbf{V}, \qquad \underline{\mathbf{m}}^2 \in \mathbf{M}^2(\mathbf{v}^2). \tag{5}$$

By changing the values ε^1 in the related sub-problem, the user can conduct the process of scanning compromises at the co-ordination level and solve for himself (herself) the most important – leading problem.

The whole procedure can start with two sub-problems and later it can be increased to more sub-problems, but the co-ordination should always be conducted as the global problem. It means the global optimisation problem increases according to the growing number of steps in the maze model (Fig. 20).

The utility function used in the related sub-problems is a good and understandable quality measure for the designer – decision maker. The sets of solutions, which are optimal in Pareto's sense and which are separately calculated, are also very understandable for the designer.

4. CONCLUSION

The paper presents the concept of the environment, called intelligent personal assistant, which is based on the maze model of the design process. An attempt to integrate the maze model of the design process and multi-criteria, multi-level, hierarchic optimisation is proposed.

The developed concept was implemented with some domain experts with several examples. The following domains were considered [39]:

1. selected car dynamics problems,
2. car braking system design,
3. some aspects of plane suspension design,
4. some aspects of plane suspension testing.

Up to now author and co-operating team have not managed to build a single, universal environment according to the presented concept. Some of the ideas shown in the paper are functioning in separate implementations done for different domains.

REFERENCES

[1] U.M. Borghoff. *Information Technology for Knowledge Management.* `http://www.iicm.edu/jucs3_8`, 1998.

[2] S. Boulanger, I. Smith. Models of design process. *Applications of Artificial Intelligence in Structural Engineering. Proceedings of the First EG-SEA-AI Workshop, Lausanne*, pp. 30–46, 1994.

[3] R.D. Braun. NASA's Intelligent synthesis environment program. *Integrated Enterprise*, **2**(1): 2–6, 2001.

[4] M.I. Campbell, J. Cagan, K. Kotovsky. A-design: an agent-based approach to conceptual design in a dynamic environment. *Research in Engineering Design*, **11**: 172–192, 1999.

[5] P. Cichocki, M. Gil, J. Pokojski. Heating System Design Support. In: *Artificial Intelligence in Structural Engineering. Information Technology, Collaboration, Maintenance, and Monitoring.* Springer-Verlag, *Lecture Notes in Artificial Intelligence 1454*, pp. 60–68, 1998.

[6] P. Cichocki, M. Gil, J. Pokojski. Knowledge based heating system design support. In: *Proceedings of the 12th International Conference on Engineering Design, TU Munchen*, pp. 1901–1904, 1999.

[7] P.J. Clarkson, J.R. Hamilton. "Signposting", a parameter-driven task-based model of the design process. *Research in Engineering Design*, **12**: 18–38, 2000.

[8] G. Coates, R.I. Whitfield, A.H.B. Duffy, B. Hills. Coordination approaches and systems – Part II: An operational perspective. *Research in Engineering Design*, **12**: 73–89, 2000.

[9] R.L. Dreisbach. Product simulation integration. *Integrated Enterprise*, **2**(1): 7–10, 2001.

[10] T. Dybala, G. Tecuci. Shared expertise space – a learning-oriented model for computer aided engineering design. *Proc. of the IJCAI-95 Workshop on Machine Learning in Engineering*, Montreal, 1995.

[11] S. Fenves, G. Turkiyyah. Knowledge-based assistance for finite element modelling. *IEEE Expert Intelligent Systems and Their Applications*, pp. 23–32, June 1996.

[12] S. Fenves. Towards personalised structural engineering tools. In: I. Smith, ed., *Lecture Notes in Artificial Intelligence, V EG-SEA-AI Workshop*, pp. 86–91. Springer-Verlag, 1998.

[13] S.J. Fenves, H. Rivard, N. Gomez. SEED-Cofig: a tool for conceptual structural design in a collaborative building design environment. *Artificial Intelligence in Engineering*, **14**: 233–247, 2000.

[14] D. Frank. The importance of knowledge management for BMW. *International Conference on Engineering Design, Proceedings ICED99, Munich, Germany*, pp. 33–40, 1999.

[15] A.M. Geoffrion, W. Hogan. Coordination of two-level organizations with multiple objectives. In: A. Balakrishnan, ed., *Techniques of Optimization*, pp. 455–466. Academic Press, 1972.

[16] H.P. Hildre. CAE from first day designing complex machines. *International Conference on Engineering Design, Proceedings ICED99, Munich, Germany*, pp. 715–720, 1999.

[17] J. Hong, G. Toye, L.J. Leifer. *Personal Electronic Notebook with Sharing, Centre for Design Research.* Stanford University, `http://www-cdr.stanford.edu/people/hong/papers/wetice/`

[18] J. Hong, G. Toye, L.J. Leifer. Engineering design notebook for sharing and reuse. *Computers in Industry*, Vol. **29**, Elsevier, 1996.

[19] Ch. Hwang, A. Masud. *Multiple Objectives Decision Making – Methods and Applications.* Springer-Verlag, 1979.

[20] Ch. Hwang, K. Yoon. *Multiattribute Decision Making.* Springer-Verlag, 1981.

[21] R. Keeney, H. Raiffa. *Decision with Multiple Objectives, Preference and Trade-offs.* Wiley, 1976.

[22] R. Klein. *Knowledge Modeling in Design – the MOKA framework.* Daimler–Chrysler Research Dept. Knowledge Based Engineering Group (FT3/EW).

[23] R. Klein. *Towards an Integration of Engineering Knowledge Management and Knowledge Based Engineering.* Daimler–Chrysler Research Dept. Knowledge Based Engineering Group.

[24] G. Li, R.H. Bracewell, K.M.W. Wallace. A knowledge framework to support engineering design. *Advances in Concurrent Engineering CE2001, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim, USA*, pp. 243–252, CD, 2001.

[25] J.G. Lin. Proper inequality constraints and maximization of index vectors. *JOTA*, **21**(4): 505–521, 1977.

[26] M. Macias, M. de Souza. Integrated analysis architecture. *Integrated Enterprise*, **2**(1): 1–18, 2001.

[27] B. Malone. Building automated processes using ModelCenter. *Integrated Enterprise.* **2**(1): 25–27, 2001.

[28] S. Nishida. A case-based layout design support system utilizing design rationale. *Computer Integrated Manufacturing*, **II**: pp. 18–26. WNT Warszawa, 2001.

[29] G. Pahl, W. Beitz. *Design Theory* (in Polish). WNT, 1984.

[30] H. Park, M.R. Cutkosky. Framework for modeling dependencies in collaborative engineering processes. *Research in Engineering Design*, **11**: 84–102, 1999.

[31] J. Pokojski. *Multi-criteria Optimization of Large Design Problems in Machine Design on the Basis of Car Gear-Box* (in Polish). PhD Thesis. Warsaw University of Technology, 1982.

[32] J. Pokojski. Computer aided decision making in vehicle dynamics. *Modelling, Simulation and Control*, B, **17**: 1–11, 1988.

[33] J. Pokojski. Computer aided multi-criteria decision making in machine dynamics, (in Polish). *Prace Naukowe Politechniki Warszawskiej, Seria Mechanika*, z.134, 1–105, Wyd. Pol. Warsz., 1990.

[34] J. Pokojski. The integrated product and process model for the design of gear box. *Proc. of Intern. Conf. on CIM, Zakopane*, pp. 367–372, 1996.

[35] J. Pokojski. Expert system Technology in machine design. Advances in Engineering, *Proceedings of the IX German-Polish Seminar, Köln*, pp. 118–125, 1997.

[36] J. Pokojski. Product model transformations in maze model of design. *Computer Integrated Manufacturing. Proceedings of Int. Conf. CIM 99, Zakopane, 9–12.03.1999*, pp. 121–128. WNT, Warsaw, 1999.

[37] J. Pokojski. Blackboard integration of design tools. *Computer Integrated Manufacturing. Proceedings of Int. Conf. CIM 99, Zakopane, 9–12.03.1999*, pp. 112–120. WNT, Warsaw, 1999.

[38] J. Pokojski. Knowledge based support of machine dynamics analysis. In: P.K. Chawdhry , P. Ghodous, D. Vandorpe, eds., *Advances in Concurrent Engineering CE99, Sixth ISPE International Conference on Concurrent Engineering: Research and Applications, Bath, UK, 1999*, pp. 336–344. Technomic Publishing Co. Inc., Lancaster, USA, 1999.

[39] J. Pokojski, ed. *Intelligent Support of Design Environment Integration in Machine Design.* (in Polish). WNT, Warszawa, 2000.

[40] J. Pokojski. Knowledge based systems in machine design. Methods of artificial intelligence in mechanics and mechanical engineering. *AI Mech 2000, Proceedings, Gliwice*, pp. 265–272, 2000.

[41] J. Pokojski. Towards personalised software in machine design – multi-criteria optimisation aspect. *Advances in Concurrent Engineering CE2001, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim, USA*, pp. 261–271, CD, 2001.

[42] J. Pokojski, P. Cichocki, M. Gil. Intelligent personal assistant for machine dynamics problems. In: A. Borkowski, ed., *Artificial Intelligence in Structural Engineering. Proceedings of the 6th EG-SEA-AI Workshop, Wierzba 99*, pp. 159–164. WNT, Warszawa, 1999.

[43] J. Pokojski, P. Cichocki, M. Gil. Intelligent personal/team assistant. In: P. Ghodous, D. Vandorpe, ed., *Advances in Concurrent Engineering CE2000, Seventh ISPE International Conference on Concurrent Engineering: Research and Applications, Lyon, France 2000*, pp. 644–649. Technomic Publishing Co. Inc., Lancaster, USA, 2000.

[44] J. Pokojski, P. Cichocki. Towards personalised software in machine design – comparative study. *Advances in Concurrent Engineering CE2001, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim, USA*, pp. 255–261, CD, 2001.

[45] W.C. Regli, X. Hu, M. Atwood, W. Sun. A survey of design rationale systems: approaches, representation, capture and retrieval. *Engineering with Computers*, **16**: 209–235, 2001.

[46] M. Rosenman, F. Wang. CADOM. A component agent-based design-oriented model for collaborative design. *Research in Engineering Design*, **11**: 193–205, 1999.

[47] P. Sainter, K. Oldham, A. Larkin, A. Murton, A. Brimble. Product knowledge management within knowledge-based engineering systems. *Proc. of DETS'00, ASME 2000 Design Engineering Technical Conference, Baltimore*, pp. 1–8, 2000.

[48] K. Shimizu, E. Aiyoshi. Hierarchical multi-objective decision systems for general resource allocation problems. *JOTA*, **35**(4): 517–533, 1981.

[49] S.B. Shooter, W.T. Keiroz, S. Szykman, S.J. Fenves. A model for the flow of design information in product development. *Engineering with Computers*, **16**: 178–194, 2001.

[50] R. Sriram. *Intelligent Systems for Engineering.* Springer-Verlag, 1997.

[51] R. Sriram. Standards for collaborative product development. *Advances in Concurrent Engineering CE2001, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim, USA*, pp. XXXVII–XLV, CD, 2001.

[52] S. Szykman, R.D. Sriram, Ch. Bochenek, J.W. Racz, J. Senfaute. Design repositories: Engineering design's new knowledge base. *IEEE Intelligent Systems*, May/June, pp. 48–54, 2000.

[53] Ch. Tong, D. Sriram, ed. *Artificial Intelligence in Engineering Design.* Vols. 1,2,3. Academic Press, 1992.

[54] S.S. Tong. Driving optimized products through automated software collaboration. *Advances in Concurrent Engineering CE2001, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim, USA*, pp. XIX–XXVI, CD, 2001.

[55] H. Werner, S. Ahmed. Design with a model system using event triggered procedures. *International Conference on Engineering Design, Proceedings ICED99*, pp. 1011–1014. Munich, Germany, 1999.

[56] H. Werner, C. Weber. Ligo – an object-oriented modelling for integrated product development. *Integrated Product Development Workshop*, Magdeburg, 1998.

[57] R.I. Whitfield, G. Coates, A.H.B. Duffy, B. Hills. Coordination approaches and systems – Part I: A strategic perspective. *Research in Engineering Design*, **12**: 48–60, 2001.

[58] M.C. Yang, M.R. Cutkosky. Machine generation of thesauri: adapting to evolving vocabularies in design documentation. *International Conference on Engineering Design, Proceedings ICED99, Munich, Germany, 1999*, pp. 143–148, 1999.