[8] J.Hoffmann,B.Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (14): 253-302*, 2001.

[9] J.Hoffmann. Extending FF to numeric state variables. *In Proc. of ECAI-02*, 2002.

[10] A.Jonsson et al. Planning in the interplanetary space : theory and practice. *In Proc. of AIPS-00*, 2000.

[11] J.Koehler,B.Nebel,J.Hoffmann,Y.Dimopoulos. Extending planning graphs to an ADL subset. *In Proc. of ECP-97*, 1997.

[12] J.Koehler. Planning under resource constraints. *In Proc. of ECAI-98*, 1998.

[13] P.Laborie,M.Ghallab. Planning with sharable resource constraints. *In Proc of IJCAI-95*, 1995.

[14] X.Nguyen,S.Kambhampati,R.S.Nigenda. Planning graph as the basis for deriving heuristics for plan synthesis by state space and csp search. *ASU Technical Report*, 2002.

[15] J.Rintanen,H.Jungholt. Numeric state variables in constraint based planning. *In Proc. of ECP-99*, 1999.

[16] S.Wolfman,D.Weld. The LPSAT engine and its application to resource planning. *In Proc. of IJCAI-99*, 1999.

Authors information

Marco Baioletti – Dipartimento di Metodi Quantitativi, Univeristà degli Studi di Siena, P.zza S.Francesco 5, Siena, Italy

Alfredo Milani - Dipartmento di Matematica e Informatica, Università degli studi di Perugia, Via Vanvitelli, 06100 Perugia, Italy

Valentina Poggioni – Dipartimento di informatica e Automazione, Università degli Studi di Roma Tre, Via della Vasca Navale 79, 00146 Roma, Italy

A PLANNING MODEL WITH RESOURCES IN E-LEARNING

G. Totkov, E. Somova

Abstract: This work proposes a model for planning of education based on resources and layers. Each learning material or concept is determined by certain characteristics: a layer and a list of resources and resource values. Models of studied subject domain, learner, information and verification unit, learning material, plan of education and education have been defined. The plan of education can be conventional, statical, author's and dynamic. Algorithms for course generation, dynamic plan generation and carrying out education are presented. The proposed model for planning of education based on resources and layers has been included in the system PeU.

Keywords: planning education, e-learning.

Introduction

The e-learning action plan has been created by the Commission of the European Communities in 2001. The development of this plan shows the actuality of the present work. A lot of attempts for creation the e-learning standards with necessary requirement (SCORM, CMI, LOM, IMS, ARIADNE) and systems for creation of e-learning environments (Learning Space, WebCT, Top Class, First Class, Blackboard, Virtual-U, Web course in a Box, CourseInfo, Learning Landscapes, CoSE, CoMentor, ARIADNE, Asymetrix Librarian, Norton Connect, Allaire Forum, Team Wave, WebBoard, Asymetrix ToolBook, etc.) [Britain] have been done. In order to satisfy the requirements of the e-learning environments, the model of planning of education have to be done.

The known models for planning of education are classical [Koffman, 1975; Пасхин, 1985; Савельев, 1986; Зайцева, 1989; Grandbastien, 1994; Grant, 1997] and resource [Milani, 2000; Milani, 2001a; Milani, 2001b] respectively based on the classical and recourse models of problem planning.

The classical model for planning [Chien] is based on the initial and the final state of the problem and the operators for transformation of one state into another. The Chien model can not represent multiple used educational environment, where learners are interested in optimizing the path for passing over learning materials, not only from the point of view of the length of the path, but also depending on the time for passing over, price, level of difficulty, etc.

The classical model is expanded in [Koehler, 1998; Rintanen, 1999; Milani, 2001a] by adding resources (level, time, price, etc.). The disadvantage of the exiting models with resources is that they do not consider the structure and content of learning materials, the studied concepts and their representation (introduction, definition, example, classification, comparison, application, etc.). These representations we will call layers.

This work proposes a model for *planning of education based on resources and layers*. Each learning material or concept is determined by certain characteristics: a layer and a list of resources and resource values [Somova, 2002].

Studied subject domain

Let $N = \{N_v\}_{v=1}^{h}$ be a set of learning matatearials' names from a different type, $L = \{L_j\}_{j=1}^{m}$ be a set of all layers, $R = \{r_k\}_{k=1}^{u}$ be a set of all resources.

Definition 1. Let S be a random finite set. The family of the sets $T = \{T_s : s \in S\}$ is called **S-set**. The S-set $T = \{T_s : s \in S\}$ is finite, if $|T_s| < \infty$ for each $s \in S$ and it is with one element, if $|T_s| = 1$, for each $s \in S$.

Let $\{S_i\}_{i=1}^{n}$ be the S-sorted sets, corresponding to the possible values of the resources. For example S_i ($i = \overline{1,n}$) can be the set of real numbers, an interval $\{[a,b] : a,b \in R\}$, a set (with full order) $\sum^*$ from all words above random alphabet $\sum$, or lexicograpf interval $\{[c,d] : c,d \in \sum^*\}$.

Definition 2. *Model of studied subject domain* (SSD) M_d is oriented graph $G_d = (V_d, E_d)$, where the set V_d consists of nodes, introducing concepts of SSD, and the set E_d consists of arcs, determining the relationships (of type *predecessor_of*) between the concepts.

Definition 3. Let SSD M_d be presented by graph $G_d = (V_d, E_d)$. *Model of learning course* M_c in M_d is orientented graph $G_c = (V_c, E_c)$, where: a) $V_c = V_c^{'} \cup B$, the set of nodes $V_c^{'}$ ($V_c^{'} \subseteq V_d$, $B \cap V_c^{'} = \varnothing$) introduces the concepts, studied in M_c, the set B consists of four type nodes – *and*, *or*, *yes* and *not*; b) the set of arcs E_c introduces the relationship between the concepts, included in M_c and

$E_c = E_c^{'} \cup E_B$, $E_c^{'} \subseteq E_d$,

$E_B \subseteq \{(v,b) : v \in V_c^{'}, b \in B\} \cup \{(b_1,b_2) : b_1, b_2 \in B\} \cup \{(b,v) : b \in B, v \in V_c^{'}\}$.

Introducing nodes from B allows a model of different strategies for carrying out the education for learning concepts $V_c^{'}$, included in the learning course. Introduced type nodes – *and*, *or*, *yes* and *not* [Somova'2002] give the opportunity to set respectively parallel actions without order of passing over, variety of education, additional supporting materials, etc. during the learning process.

Definition 4. *Model of the learner* M_s^t at the moment t of learning on a given course is determined by the triple (V_b, V_t, R_t), where V_b is the initial set of known concepts, V_t is the current set of learned concepts (including V_b) in the process of learning ($V_b \subseteq V_t \subset V_c$), and R_t is the list of couples resource-value, specifying the current characteristics of the learner: $R_t = \{(r_{i_k}, s_{i_k}) \in R \times S\}_{k=1}^{n}, i_k \neq i_p, k \neq p$.

The models of initial and final (target) state of the learner M_s^b and M_s^e are determined respectively by the triples (V_b, V_b, R_b) and (V_b, V_e, R_e), where V_e is the final (target) set of concepts included in the course, and R_b и R_e – the initial and final set of characteristics of the learner.

Learning materials

The *learning materials* can be either *learning units* (text, multimedia, etc.) or *verification units* (tasks or tests for verification or self-assessment). Every learning material regards one or several concepts. The full representation and verification of a concept in the common case can be made in some learning and verification units.

Definition 5. *Model of information unit* M_l in SSD M_d is determined by the couple (N_l, H_l), where $N_l \in N$ is the name of information unit, and H_l is a finite set of triples (C, L_C, f_C), where $C \in V_d$ is a concept from the SSD M_d, $L_C \in L$ is a layer for the concept C, and $f_C : C \times L_C \to S_1^\varepsilon \times S_2^\varepsilon \times ... \times S_n^\varepsilon$ is a resource function. Here $S_i^\varepsilon = S_i \cup \varepsilon$, $i = \overline{1,n}$ and ε is used in cases when the value of the respective resource is not determined.

Note: In the realization of the model, each information unit M_l is determined by the triple $(N_l, H_l, Type(N_l))$, where $Type(N_l)$ is the type of the physical file containing presentation of M_l. The interpretation of the information unit M_l during the learning process is different according to the file type (for example – a visualization of the file on the display in proper way). For each information unit the characteristics layer and list of the couples resource-value are determined.

Definishon 6. *Model of verification unit* (**test assignment)** M_v is determined by the order sextuple (S, K, T, p, q, H_v), where K is a random with one element S-set (named c*ontext* or S-frame of the assignment); $T = \{T_s : s \in S\}$ is S-sorted set (presenting sorted sets of the particular slots of the S-frame); p – unary predicate (determining *potential answers* or limitations in filling in the slots at the same time); q – binary predicate (determining *decisions* of the assignment), and H_v is a finite set of couples (C, f_C), where $C \in V_d$ is concept of SSD M_d and $f_C : C \to S_1^\varepsilon \times S_2^\varepsilon \times ... \times S_n^\varepsilon$ is a resource function.

Definition 7. *Learning material* is an information unit or a verification unit.

Planning of education

Definition 8. *The Plan of education* P on a given learning course $M_c = (V_c, E_c)$ is oriented graph $G_p = (V_p, E_p)$, where $V_p = [v_1,..., v_k]$ is a finite set of lists of learning materials, and $E_p \subseteq E_c$. For each $i = \overline{1,k}$, if learning material $m \in v_i$, then: a) if m is an information unit (n, h), then the concept $c \in V_c$, the layer $l_c \in L$ and the resource function f exist, for which $(c, l_c, f) \in h$; b) if m is a verification unit (s, k, t, p, q, h), then the concept $c \in V_c$ and the resource function f exist, for which $(c, f) \in h$.

Definition 9. *The Model of education* on a given course M_c in SSD M_d is the sextuple $(R, S, L, \mu_l, \mu_v, P)$, where R is the set of all resources, S – the set of resource types, L – the set of all layers, μ_l – the set of information units, μ_v – the set of verification units, P – the plan of education.

The plan of education of a particular learner is determined individually from M_s^b and M_s^e.

Actions, which are caused by a learning and a verification unit, are respectively a ***learning*** and a ***verification action***. The ***learning action*** is realized when the material(s) is(are) offered to the learner and it is supposed that the knowledge of the learner is changed (increased). The ***verification action*** is realized when the material(s) are offered to the learner and they interact with the learner in order to evaluate learner's knowledge and skills.

After forming (modeling) the content of the learning course by pointing out the concepts and relationships from the SSD, which students have to learn, a respective learning course is automatically generated in the DB.

In the *algorithm for course generation* (Algorithm 1) the following procedures are used:

> **procedure** Create_Course (**var** Course; Author; Subject_Domain; Course_Node; Course_Link) – creates a course of the given author in the given SSD with the given nodes and relationships between them;
>
> **procedure** Domain_Concepts (Subject_Domain; **var** Domain_Concept[]) – finds and shows the concepts in the given SSD;

procedure Select_Concepts (Domain_Concept[]; **var** Selected_Concept[]) – chooses the concepts from the given SSD;

procedure Course_Nodes (**var** Course_Node[]) – finds and shows the nodes of the given course;

procedure Course_Links (**var** Course_Link[]) – finds and shows the relationships between the nodes of the griven course;

procedure Select_Nodes (Course_Node[]; **var** Selected_Node[]) – chooses the node/s of the course;

procedure Select_Links (Course_Link[]; **var** Selected_Link[]) – chooses the relationship/s between the course nodes;

procedure Add_Concept (Selected_ Concept[]) – adds concept/s in the course;

procedure Special_Node (Selected_Node[]; Node_Type) – links the selected nodes through special node of type Node_Type;

procedure Link (Selected_Node; Selected_Node[]) – links the node with its predecessor nodes;

procedure View_Node (Selected_Node) – shows the type of the node;

procedure Del_Node (Selected_Node[]) – deletes the node/s from the course, and

procedure Del_Link (Selected_Link[]) – delete relationship/s between two nodes from the course;

```
procedure Author_Course(Course, Author);
begin
  Domain_Concepts (Subject_Domain, Domain_Concept[]);
  Course_Node[]=empty; Course_Link[]=empty;
  Create_Course (Course, Author, Course_Node, Course_Link);
  repeat
    Course_Nodes (Course_Node[]);
    Course_Links (Course_Link[]);
    case Proc of
      View_Node: begin
        Select_Nodes (Course_Node, Selected_Node);
        View_Node (Selected_Node);
      end;
      Add_Node: begin
        Select_Concepts (Domain_Concept[], Selected_Concept[]);
        Add_Concept (Selected_Concept[]);
      end;
      Add_Special_Node: begin
        Select_Nodes (Course_Node, Selected_Node[]);
        Special_Node (Selected_Node[], Node_Type);
      end;
      Del_Node: begin
        Select_Nodes (Course_Node, Selected_Node[]);
        Del_Node (Selected_Node[]);
      end;
      Add_Link: begin
        Select_Links (Course_Link[], Selected_Link[])
        Link (Selected_Node, Selected_Node[]);
      end;
      Del_Link: begin
        Select_Links (Course_Link[], Selected_Link[])
        Del_Link (Selected_Link[]);
      end;
    end;
  until end;
end;
```

❖ **Algorithm 1. Course generation**

The plan of education is a sequence of learning and verification actions. The plan can be *statical* (it is represented with one learning material, containing learning information, responding to the initial state of the particular group of learners) and *dynamical* (for learners with different knowledge and skills, which are updated on the base of graph of learning materials and history of education). The education can be carried out *statically* (through using the static and conventional plan), *in certain methodology* (for example through passing over a given plan in depth, in width or in some other rules) and *freely* (through dynamic or author's plan).

The conventional plan of education is built only of one learning material, which responds to the requirements of the exact particular group of learners. This plan corresponds to the conventional learning courses and does not change according to the progress of the learner. In the common case, the material is in the form of hypermedia document and except following the content of the material, it can carry out navigation through hiperlinks realized in the document. In this case, it is not necessary to determine the relationships with other information units.

The static plan of education is generated on the base of other type plans of education. The plan is presented as a graph of learning materials (eventually with an indication of recommended relationships between them). However, the education is realized through following the learner's will, not following the indicated relationships between the learning materials.

Each author can propose his/her own plan of education (*author's plan*) by his/her own learning materials, i.e. a recommended path for learning the materials. The student is free to choose the order of learning (from actions unified with *and*) and to miss some actions in the presence of or-nodes. The author's plan is developed without using a graph of concepts. This path is accepted to be the best and is used by the dynamic plan for giving some recommendations about the choice between several nodes from the graph.

When choosing which learning materials to be included in the author's plan, resource and layer restrictions are not used. The process of automatic generation (procedures and algorithm) is analogical to the generation of the learning course with one difference – instead of building a graph of concepts a the graph of learning materials is built, and the initial set of concepts from SSD Domain_Concept[] is changed by the set of learning materials from a given author Author_Material[].

In order to generate the *dynamic plan of education* on the base of the graph of concepts, it is necessary to determine the initial and final state in the process of learning for each learner. During the generation of the plan about a given learner on the learning course, the teacher chooses the author (authors) from whom the learning materials should be found, and which should respond to the given characteristics (which resources, with what value and which layers). For each concept from the graph of the course suitable learning materials in the DB are searched. The suitable materials are shown, sorted by the number of characteristics, which have been satisfied; the number of the additional concepts (in and out of the thematic of the course), that are presented; etc. From the found learning materials (information and verification units), some materials are selected and ordered in a linear structure, which replaces the node-concept. The special nodes and relationships between all course nodes remain unchanged in the graph of the plan.

The found learning materials, explaining/verifying extra concepts (besides the searched concepts) which are not in the learning course, are recommended for additional learning/solving. It is recommended to search for a material in conventional bearers or a help from the teacher for the concepts, about which materials are not found in the DB. During the educational process some learning materials are proposed only for concepts, for which predecessor concepts are passed over with their respective materials. This way of generation can bring to a large number of learning materials, linked with one concept, if the resource and layer restrictions of the course are not defined well (for example, if in the DB, there is a lot of authors and materials for using in the particular case).

When selecting among the materials, created from different authors, in order to generate the plan of education, it is possible to assign coefficients for recalculation of authentic author valuations (levels). Therefore, the learning materials representing layers for one concept, created from different authors can be combined.

On the base of the author's plan of education one of the materials is always recommended as the best-fitting one [Somova, 2002]. However, the learner himself/ herself has the opportunity to choose the next material.

In the *algorithm for generation of dynamic plan* (Algorithm 2) the next procedures are used:

> **procedure** Initial_Concepts (Course; **var** Initial_Concept[]) – determines the initial concepts (which do not have predecessor concepts);

procedure Learn_Materials (Concept; Layer_Restrictions; Resource_Restrictions; **var** Learn_Material[]) – determines the information units for a given concept, which responds to the layer and resource restrictions;

procedure Assignments (Concept; Resource_Restrictions; **var** Assignment[]) – determines the assignments for a given concept, which responds to the layer and resource restrictions;

procedure Next_Concepts (Concept; **var** Next_Concept[]) – finds the next nonpassed concepts, which are direct predecessors of the given concept;

function count (Array): Number – returns the number of the array elements;

procedure Recursion (Concept_Array[], Prev_Material) – recursively passes over the course in order to generate the plan, finds the next level concepts and their materials and assignments, and makes the relationship between materials/assignments;

procedure Choose_Order (Material[]; **var** First_Material; **var** Last_Material) – the necessary materials and assignments from the array are chosen, they are ordered, then linked in the proper order and the first and last material/assignment are returned, and

procedure Link (Prev_Material, First_Material) – links one node with another with the relationship of type *predecessor_of*.

```
procedure Generation_Dynamic (Course; Layer_Restrictions; Resource_Restrictions);
begin
  Initial_Concepts (Course, Initial_Concept[]);
  Recursion (Initial_Concept [], “”);
end;
procedure Recursion (Concept_Array[], Prev_Material);
begin
  for i:=1 to count (Concept_Array) do
  Begin
    if (Concept_Array [i]<>”and”) and (Concept_Array [i]<>”or”) and (Concept_Array [i]<>”not”) and (Concept_Array [i]<>”yes”) then
    begin
      Learn_Materials (Concept_Array [i], Layer_Restrictions, Resource_Restrictions, Material[]);
      Assignments (Concept_Array [i], Resource_Restrictions; Material[]);
      if count (Material)=0 then
      begin
        write “Find materials/assignments on conventional bearers for the concept”, Concept_Array [i]);
        First_Material:= “”;
        Last_Material:= “”;
      end
      else
        Choose_Order (Material[], First_Material, Last_Material);
    end
    else
    begin
      First_Material:= Concept_Array [i];
      Last_Material:= Concept_Array [i];
    end;
    Link (Prev_Material, First_Material);
    Next_Concepts (Concept_Array [i], Next_ Concept[]);
    if count (Next_ Concept[])=0 then exit
    else
      Recursion (Next_Concept[], Last_Material);
  end;
end;
```

❖ **Algorithm 2. Generation of the dynamic plan**

Carrying out the education

The process of education consists of proposition of learning information units (for introduction) and verification units (for solving and testing) according to the respective plan. Because of the simplicity of the conventional plan (one unit, in which there can be hyperlinks) and the statical plan (characterized with free navigation among the materials without accounting the relationships among them), the algorithms for carrying out the education are not presented.

The algorithm for carrying out the education through author's and dynamic plan (Algorithm 3) oblige the learners to respect given relationships among the learning materials. In the algorithm the following procedures are used:

procedure Initial_Nodes (Plan; **var** Initial_Node[]) – determines the initial nodes (these which do not have predecessor nodes);

procedure Recursion (**var** Pass_Node[]; **var** Potential_Node[]) – recursively passes over the education plan, gives and finds the passed and the next potential nodes;

function Show_Material (Material): Success – gives the access (as hyperlink) to the material and returns as a result: 0 – unsuccessfully passed node, 1 – successfully passed node, 3 – it is not known how the node has been passed);

procedure Next_Nodes (**var** Next_ Node[]); – finds the current potential nonpassed nodes;

procedure Show_Nodes (Potential_Node[]); – shows all the potential nodes;

procedure Select (Array_Node[]; **var** Node); – the user selects one of the nodes;

procedure Branch_Pass (Nodes[], Type); – the branches, beginning with nodes of the pointed type are marked as passed;

procedure Find_Successors (Node; **var** Successor[]); – finds the successors of the pointed node, and

procedure Choose_Node (Nodes[]; **var** Node); – selects one of the nodes.

```
procedure Dynamic_Learning (Plan);
begin
  Initial_Nodes (Plan, Initial_Node[]);
  Recursion ([],Initial_Node[]);
end;
procedure Recursion (Pass_Material[], Potential_Material[]);
begin
  write ("Choose one of learning materials");
  Show_Nodes(Potential_Material[]);
  Select (Potential_Material[], Material);
  if Material="and" or Material [i]="or" or Material [i]="not" or  Material [i]="yes" then
    begin
      if Material="or" then
        begin
          write ("Choose one of materials: ");
          Pass_Material[count(Pass_Material)+1]:= Material;
          Find_Successors (Material, Successor[]);
          Choose_Material (Successor[], Material);
          Success := Show_Material (Material);
          Branch_Pass (Successor[]-Material,"material");
          Branch_Pass (Successor[]-Material,"and");
          Branch_Pass (Successor[]-Material,"or");
          Branch_Pass (Successor[]-Material,"yes");
          Branch_Pass (Successor[]-Material,"not");
        end;
    end
```

```
    else
      Success := Show_Material (Material);
    Pass_Material[count(Pass_Material)+1]:=Material;
    Next_Nodes (Next_ Material[]);
    Find_Successors (Material, Successor[]);
    if Success=3 then
      begin
        Branch_Pass (Successor[],“yes”);
        Branch_Pass (Successor[],“not”);
      end
    else
      begin
        Branch_Pass (Successor[],“or”);
        Branch_Pass (Successor[],“and”);
        Branch_Pass (Successor[],“material”);
        if Success=0 then Branch_Pass (Successor[],“yes”);
        else if Success=1 then Branch_Pass (Successor[],“not”);
      end;
    Recursion (Pass_Material[],Next_Material[]);
  end;
```

❖ **Algorithm 3. Education through author’s and dynamic plan**

Conclusion

The proposed model for planning of education based on resources and layers has been included in the system PeU (http://peu.pu.acad.bg). The experiments with the system confirm that the model gives to the learner the opportunity – the generated plan for the course to vary dynamically on the base of learner’s progress passing over each learning phase and to correspond to the learner’s initial knowledge and way of learning.

Bibliography

[Зайцева, 1989] Л. Зайцева, Л. Новицкий, В. Грибкова, Разработка и применение автоматизированньх обучающих систем на базе ЕВМ, Зинатне, Рига, 1989.

[Пасхин, 1985] Е. Пасхин, А. Митин, Автоматизированная система обучения ЕКСТЕРН, Издательство Московского университета, 1985.

[Савельев, 1986] А. Савельев, В. Новиков, Ю. Лобанов, Подготовка информации для автоматизированньх обучающих систем, Вьсшая школа, Москва, 1986.

[Britain] S. Britain, O. Liber, A Framework for Pedagogical Evaluation of Virtual Learning Environments, http://www.jtap.ac.uk/reports/html/jtap-041.html

[Chien] S. Chien et al., The multimission VICAR planner: A knowledge-based system for automated image processing, NASA Science Information Systems, Issue 38.

[Grandbastien, 1994] M. Grandbastien, E. Gavignet, ECSA: An Environment to Design and Instantiate Learning Material (Handbook of Design and Production of Multimedia and Simulation-based Learning Material; de Jong T. and L. Sarti; eds.), Kluwer Academic, Publishers, Dordrecht, The Netherlands, 1994.

[Grant, 1997] T. J. Grant, M. Verhoef, L. P. Gale, AI planning for just-in-time training in space: ESA’s integrated learning system, in Proceedings of 1st NASA Workshop on Planning and Scheduling for Space, Oxnard, CA, Oct 1997.

[Koehler, 1998] J. Koehler, 1998. Planning under resource constraints, in Proceeding of ECAI-98, 489-493, 1998.

[Koffman, 1975] E. B. Koffman, J. M. Perry, Description and evaluation of a model for generative CAI, Computers in education: Proc. II-d Conf. IFIP, Amsterdam, 1975.

[Milani, 2000] Alfredo Milani, Stefano Marcugini, Marco Baioletti, DPPLAN: An algorithm for fast solutions extraction from a planning graph, Proceedings of AIPS2000, 5th Int. Conf. on AI Planning and Scheduling, Chien & Kambhampati eds, AAAI Press, Menlo Park, USA, 2000.

[Milani, 2001a] Alfredo Milani, Valentina Poggioni, Automated planning of educational goals, Proceedings of the ICT&P 2001 Conference, Sofia, 49-60 p., 2001.

[Milani, 2001b] Alfredo Milani, Valentina Poggioni, RDDPlan: a system for planning with resources, Tech. Rep. 12-2001, Department of matematics and computer science, University of Perugia, Italy, 2001.

[Rintanen, 1999] J. Rintanen, H. Jungholt, 1999. Numeric state variables in constraint-based planning, in Proceedings of the 5th European Conference on Planning, ECP99, Durham, UK, 1999.

[Somova, 2002] E. Somova, G. Totkov, Modelling of Education in the Environments of Type "Virtual University", Conf. EDEN: Open and Distance Learning in Europe and Beyond Rethinking International Co-operation, Granada, Spain, 16-19 Jun, 2002, p. 275-280.

Author information

George Totkov - The University of Plovdiv, Dept. of Computer Science, 24 Tzar Assen st., 4000 Plovdiv, Bulgaria; e-mail: totkov@pu.acad.bg

Elena Somova - The University of Plovdiv, Dept. of Computer Science, 24 Tzar Assen st., 4000 Plovdiv, Bulgaria; e-mail: eledel@pu.acad.bg

PLANNING OF INTELLECTUAL ROBOT ACTIONS IN REAL TIME

N. Romanenko

Summary: In article the mathematical model of the mobile robot actions planning at recognition of situations in extreme conditions of functioning is offered. The purpose of work is reduced to formation of a concrete plan of the robot actions by extrapolation of a situation and its concrete definition with the account a priori unpredictable features of current conditions.

Key words: the mobile robot, recognition of a situation.

Introduction

Creation of the intellectual mobile robots, capable to adapt and plan the actions in conditions of aprioristic uncertainty of dynamically changing habitat, is one of the important strategic problems of modern techniques. Absence of preliminary environment formalization, and also presence of any way moving obstacles and purposes in it complicates the use of automatic control traditional methods. The given circumstance stimulates development of new control systems with presence on mobile robots (MR) board of situation recognition system on the basis of the multiprocessing computer with elements of artificial intelligence that provides adaptability of MR behavior in an environment.

Recognition of situations is the new area of cybernetics. The closest area is images recognition. But there is a basic distinction of these concepts: the image is "static", and the situation is dynamical, recognition of situations is always connected to a prediction (extrapolation) that usually does not happen in the theory of images recognition. At situation recognition there is no aprioristic classification as the number of possible situations is unlimited, but results are classified and have the final alphabet.

For MR control system we shall understand that the situation is a set of events, developing in time and space limited in radius of its action, and having the important consequences from the point of view of the chosen criterion function. The situation includes three basic components:

- the ground conditions fixed during the certain moment of time (presence of obstacles in a way);
- processes which can occur both with its condition, and with MR condition (dynamism of obstacles and the robot);
- result or possible consequences (planning of actions and forecasting).

To distinguish a situation by control system - means to develop decision about result of further MR movement on the basis of environment and proceeding process information.

Traditional mathematical models of MR management in extreme conditions of functioning becomes insufficiently as MR proper response to change of situations is not described, especially at occurrence of obstacles. The given problem was examined by many researchers, and there exist various ways of its decision [1-3].