БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, 52
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 52

София . 2001 . Sofia

An Approach for Intelligent Decision Support in Research Knowledge Network

Tatiana Atanasova

Institute of Information Technologies, 1113 Sofia

1. Introduction

In the Internet and Intranet there is a lot of valuable resources for scientific research – digital libraries, conference materials, journal papers, software collections, mails in newsgroups, images, demos, data files, etc [1]. The question is how to use them for building research models on the base of distributed and heterogeneous knowledge. The solution may be a management information system, which visually presents the information in the research knowledge base, and provide decision support to optimize the results of searching.

The decision support systems are characterized with the human-centered principles of designing for environment of decision support – the principle of user as designer, the integration of subjective opinion with objective knowledge and putting the decision-maker in the center of the process. This is opposite to techno centric vision of expert systems. Traditional, model based, decision support systems provide calculations based on assumptions and formulae. The result often is adjusted then by an expert. Other forms of decision support systems include multicriteria models that allow more suitable adjustments.

In this paper the research process is considered. The aim for the investigation is formulated as follows:

For the given research task it is need to recommend a list of approaches, methods, documents, software libraries and other relevant knowledge objects that satisfy some criteria for fitness. The task may be defined in different manner ranging from simple set of keywords to structured task description.

The rest of the paper is organized as following. First we consider the research process phases and knowledge structuring for them. Then approach for intelligent decision making is discussed. An algorithm for decision support in research network is proposed. Implementation tools are mentioned.

2. Knowledge structuring for research process

The research process involves several phases. The **first phase** is concerned with significant literature studies, long negotiations and first experiments.

The **second phase** is concerned with developing of structured task knowledge base and client-server architecture for communications in the network. The knowledge base consists of models for different solving methods. All concepts have to be instances of the defined metamodel.

Third phase: it is necessary to ensure the coherence between tasks. Analysis of formal conflicts (inconsistency) is performed via queries.

To assist researchers in their research problems an intelligent system can be built on a base of common approach in representing and processing knowledge.

Knowledge representation and processing for decision support

We consider the research process as one that consists of TASKS and OBJECTS consumed and produced by those TASKS.

METHOD is used because there may be different ways for resolving of a specific task.

METHODS and TASKS form an AND/OR decomposition structure. In this structure a task can be supported by one or several methods and a method consists of some sub-tasks.

The following scheme TASK $\rightarrow$ METHOD $\rightarrow$ OBJECT represents a metamodel for decision making.

Let T be a set of tasks, $D = \{d_1, d_2, ..., d_N\}$ is a set of possible solutions, $M = \{m_1, m_2, ..., m_L\}$ is a set of possible methods.

The task may be represent in a notation of constraint net:

$T_i = (V_i, D_i, C_i)$,

$V_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$ – object variable set,

$D_i = D_{i1} \times D_{i2} \times \ldots D_{ij} \times \ldots \times D_{iNi} = \times D_{ij}$ – set of object variable values,

$C_i = \{c_{i1}, c_{i2}, \ldots, c_{ik}\}$ – constraints set,

C_i is a KDB of correct states of DB.

This is a static model [2]. Dynamic model is a sequence of static models in which changes of V_i reflect on including or excluding of elements from C_i.

Knowledge and databases (KDB) consist of theoretical and empirical knowledge. Current and previous problems are described in the KDB using a vocabulary of terms from the domain ontology.

The information in the KDB is accessed through:

- formulation questions based on information needs,
- identification of potential sources of information,
- developing of successful search strategies.

3. Decision scenarios

Decision about retrieving suitable knowledge objects is made according 3 possible scenarios (Fig. 1):

1) search by keywords,
2) tasks comparing – on the base of task description,
3) comparing of structured task – on the base of structured task descriptor.

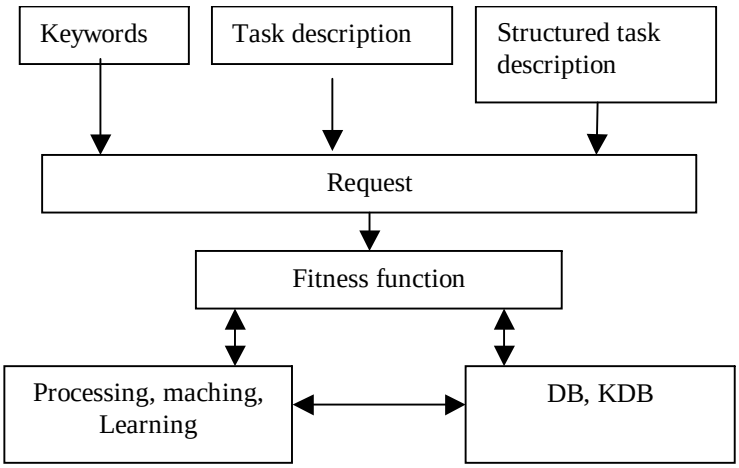

Fig. 1. Decision scenarios

3.1. Description by keywords

The search by keywords is the most familiar case. The metadata representation is used for describing knowledge objects. This is extensible, scalable, and supports the requirements for decision making. For the hypertext documents the Dublin Core standard is developed. The Dublin Core (DC) [3] is a 15-element metadata element set intended to facilitate discovery of electronic resources. DC serves as a core element set for resource discovery. The same elements are used to describe text and images, for example.

This representation provides a template for sets of objects. It also serves a very important role for supporting distributed queries based on knowledge object type as a content-based document catalog and search tool.

The document may be found by identifying one or more fields from the given template (DC):

1. Title;
2. Author;
3. Description;
4. Given keywords;
5. Input date;
6. Publisher;
7. Type;
8. Source;
9. Contributor;
10. Format;
11. Language;
12. Rights;
13. Identification;
14. Coverage;
15. Relation.

To make the results more useful for the user it is need to optimise them. The optimisation is proceeding on the base of clustering "Keyword/Value Notation" within the given ontology. Different clustering techniques in the feature space can be used.

3.2. Task Description

Searching by general text description is another possibility. The user may describe his research task in free text, for example: "How and by which tools to make dynamic simulation of turbo-generator group in presence of disturbances?" In this case it is necessary first to provide pre-processing of this text in order to extract description by keywords and after that to compare it with existing clusters. The pre-processing may be done by neural networks with Kohonen rule: $i = \text{find}(A(i) = 1)$,

$$W(i, j) = \text{lr}(P(j) - W(i, j)) \; \forall \; i, A(i) = 1,$$
$$W(i, j) = W(i, j) + \Delta W(i, j),$$

where i is a raw vector, $W(i, j)$ is a weight matrix, $P(j)$ – input vector, lr – learning rate.

3.3. Structured task description

The space of structured task is shown in Fig. 2. The metadata representation is used for describing the structured task. Some modifications need to be made to make the set more suitable for our purposes. In this work an extension of Dublin Core is proposed for metadata description of structured task (Fig. 2'). In such a way it is allowed to embed structured content representation in KDB.

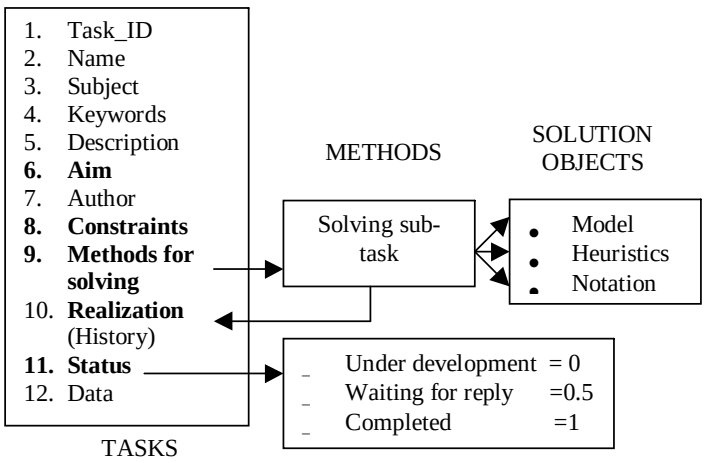

Fig. 2. Space of structured task

Thus the structured task descriptor can be described by Dublin Core metadata set plus the task extension, given in Fig. 2'.

Dublin Core

1. Title
2. Creator
3. Subject
4. Description
5. Publisher
6. Contributor
7. Date
8. Type
9. Format
10. Identifier
11. Source

DC extension

12. Language
13. Relation
14. Coverage
15. Rights
16. Aim
17. Constraints
18. Methods (models) for solving
19. Realization (history)
20. Status
21. Domain

Fig. 2'. Metadata for description of structured task

In information management system it is necessary to have a task editor and a task manager.

The structured task editor has the following functions: Input, edit (correct record), delete record, view (display), print.

The task manager performs:

- Define task;
- Select task;
- Delete task;
- Load task;
- Quit.

4. Algorithm for the decision support

Searching of a solution for such defined tasks is provided by database search utilities for the local knowledge base. If the local search gives no result then the search in the web can be proceeded. The system collects the knowledge objects using several autonomous web agents, which automatically analyse, index, and assign the knowledge objects to subject classes. Ontological filtering engine based on clustering algorithms is proposed (Fig. 3). The result of clustering is organising of the information in the knowledge base into clusters. Every cluster consists of the detailed information:

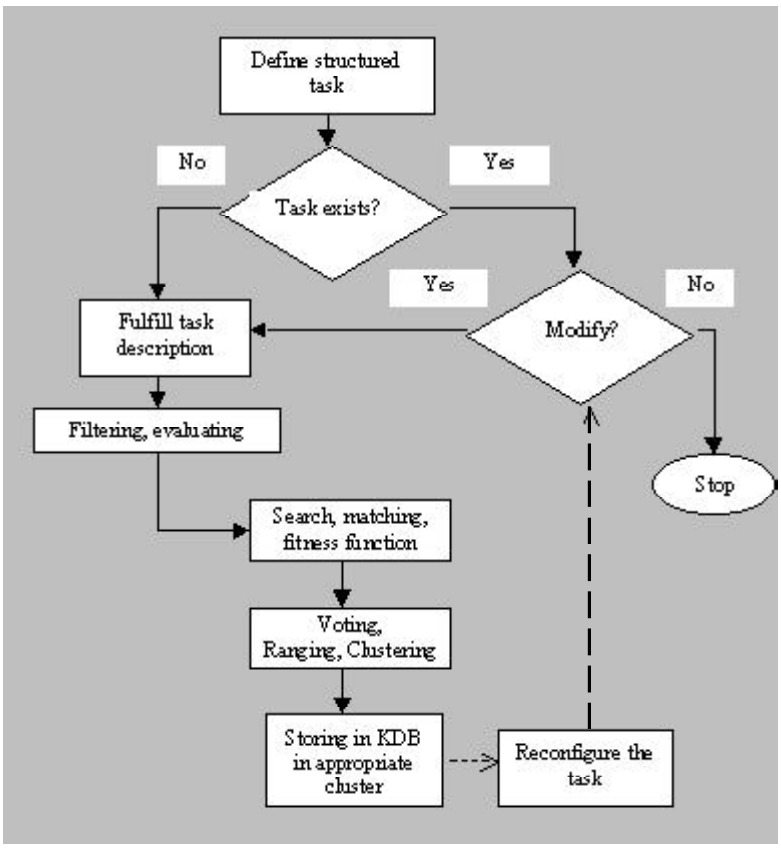

Fig. 3. An algorithm for decision support

- the list of members;
- summary description (set of keywords);
- related clusters.

This allows:

- Efficient indexing data structure for high-dimensional feature space.
- Efficient user interface for query specification and visual browsing.
- Association with domain knowledge and other datatypes.
- Evaluation criterion and methodologies.

The algorithm for the decision support can be summarised as follows:

1) define the task;
2) define the fitness function – (set of keywords, or: matching parts of Structured Task Description Vector with metadata tags in DB);
3) search in DB (and web);
4) criteria for stopping (in internet-based searching);
5) show the results.

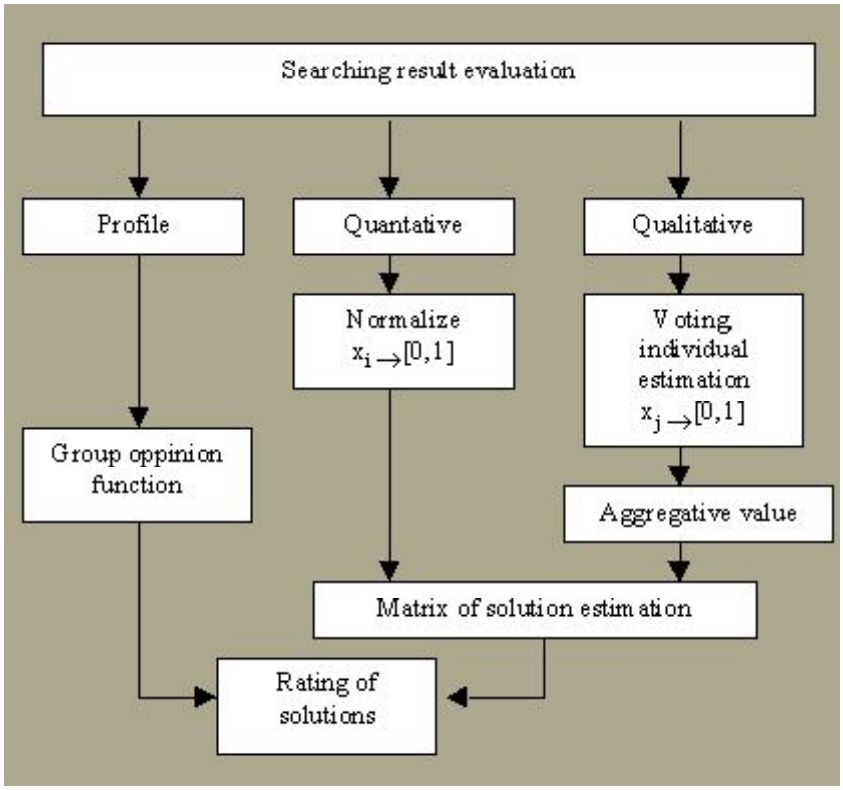

Fig. 4. Result evaluation

The decision results received on the basis of this approach can be tested and evaluated (Fig. 4). If their quality is good then the results are saved into knowledge base. The history of best results can be made visible and the contest-based information can be designed locally into the task. Voting and feedback systems allow estimating the quality of the solving task. These systems are parts of learning module of DSS. Statistical factors are included in the results. There is a matrix calculated from the data of the database that is associated with each link of the matrix. Knowledge object taxonomy is constructed in a semi-automatic way.

5. Implementation issues

It is proposed to use 3-tier architecture for realization of the developed algorithm. It consists of:

- Database;
- Java servlets (on a web server);
- Scripts (in a web browser).

For representation of the content of knowledge objects (documents, graphics, program codes) an XML [4] may be used [5, 6]. Content representation allows manipulating content, to make search by analogy, specialisation, similarity, etc. XML enables to insert content representation within the knowledge documents.

The complete system would possess several powerful functionalities: searching using content-based techniques, query modification using content-based relevance feedback and text-based searching.

6. Conclusion

In the paper an approach for searching and retrieval of relevant knowledge objects in distributed heterogeneous network environment is proposed. The search is intelligently guided by ontologies and metadata for ontologies. Ontological filtering engine is used to answer simple request of the user in the form of keywords. If the request is made in form of free text then the intelligent processing is necessary to obtain the weighted keywords that allows using the ontological filtering engine. Searching and browsing activities range from a well-defined search for a specific document to a non-specific task to see which information is available. To support these activities, it is proposed that the system uses metadata information to provide services that help refine user queries to focus a search, automatically route queries to relevant servers, and cluster related items.

The structure of research task according to the metadata standard for hypertext documents is derived. The structured task description vector is used for searching in the local KDBs and distributed information sources.

A query refinement is needed to overcome the problem of large result sets that often are returned by search engines. This is provided by suggesting modifications to focus user queries.

Query refinement is based on clustering algorithms. The organization of information into clusters of related items assists both the users and the system when dealing with large information spaces. The cluster abstraction allows a large information space to be treated as a unit, without regard for the details of its contents. Clusters also provide convenient units for the partitioning of work and resource allocation among the distributed components of the system.

References

1. Honavar V., Les Miller, J. Wong. Distributed Knowledge Networks. http://www.cs.iastate.edu/~honavar/aigroup.html
2. Смирнов, А. В. Управление конфигурацией производственных систем: методы и информационные технологии. Теоретические основы и прикладные задачи интеллектуальных информационных технологий. Под ред. дтн проф. Р.М.Юсупова. Санкт-Петербург, СПИИРАН, 1998,111–123.
3. Dublin Core metadata set recomendation (1999). http://purl.oclc.org/metadata/dublin_core/
4. Extensible Markup Language (XML), http://www.w3.org/XML/

5. A t a n a s o v a, T. XML and RDBMS representation of Fuzzy system knowledge. – In: Fifth International Conference "Information Theories and Applications". – In: ITA 2000, Varna, Bulgaria, 1-15.09.2000.
6. A t a n a s o v a, T. XML view of fuzzy system knowledge. – In:The Third International Workshop on Personal Computer and Particle Accelerator Controls, 9-12 October 2000, Hamburg, Germany, http://www.desy.de/pcapac

Подход для интеллигентного принятия решений при исследовании сетей знаний

Татяна Атанасова

Институт информационных технологий, 1113 София

(Р е з ю м е)

Предлагается подход для интеллигентного принятия решений при исследовании сетей знаний. Цель исследования формулируется следующим образом. Для данной исследовательской задачи выбираются документы, методы, программные средства и другие объекты, удовлетворяющие данным критериям соответствия. Задача может быть представлена разнотипно – от списка ключевых слов до структурированного описания.