ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Rainer Manthey and Viacheslav Wolfengagen (Eds)

Advances in Databases and Information Systems 1997

Proceedings of the First East-European Symposium on Advances in Databases and Information Systems, (ADBIS'97), St Petersburg, 2-5 September 1997

The Sybil Database Integration and Evolution Environment: An Overview

Roger King, Michael Novak, Christian Och, Richard Osborne, and Fernando Velez

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

ISBN: 3-540-76227-2

Designing an Intelligent Decision Aiding System Using a Multiagent Approach

Beatrice Rumpler, Sylvie Calabretto

LISI-INSA de LYON

20, avenue Albert Einstein

69621 Villeurbanne Cedex

France

Tel : (33) 4 72 43 83 92 / (33) 4 72 43 88 94

Fax : (33) 4 72 43 85 18

E-mail : rum@if.insa-lyon.fr , cala@if.insa-lyon.fr

Abstract

The project described in this paper deals with the design of a software product aimed to help users during the tuning process. It gives suggestions about a few crucial points, such as suitable hardware, database parameters and object distribution on the disks. Two scenarios have prevailed : if the database already exists, statistics automatically collected by Oracle will be extensively used. In the case of database creation, an existing module consisting of a theoretical performance evaluation tool will serve as a basic framework. Knowledge modelling is to be based on KADS, a well-known technique for knowledge-based systems. This method will be used particularly for specific tuning tasks, while a multiagent approach will be preferred for designing the general application model.

1 Introduction

This project deals with the design of a Software Intelligent Aiding System for Oracle Database Configuration and Tuning. This system is for designers of ORACLE transactional applications and for administrators of ORACLE databases. It offers to users the possibility of optimising interactively the ORACLE database configuration, including software and hardware aspects. The proposed system can analyse interactively the configuration parameters provided by the user or defined in the database itself. It returns information about the performance of the system according to the provided elements of configuration. It can decide to perform a new evaluation of the performance using the new parameters defined by the user. This interactive process can go on until the user is satisfied by the performance of the system.

This project is the following step to previous research work [11], in which we first extract the most influential parameters of the response time of a transaction in an ORACLE database system under UNIX. In a second step, we have designed a tool for theoretical evaluation of the critical resources consumed during the execution of the requests and for the evaluation of the associated performances. This previous work constitutes an important contribution to the methodology proposed for the final Intelligent Decision Aiding System. Concretely, two scenarios of analysis have prevailed:

- a suitable sizing of the system proposed during the creation phase
- the optimisation of an existing system

So, two types of evaluation can be proposed:

- during the creation phase, only a theoretical evaluation can be performed, according to the future use of the database (data storage, requests executed)
- for an existing database, it is possible to explore statistics, automatically collected by ORACLE, which give information about the execution of processes and about the use of the resources.

We have used the KADS method (Knowledge Analysis and Design Support) to design the conceptual model describing our approach for optimisation. If this method is adapted to the modelling aspect of our study, we have chosen a multiagent approach to implement the model itself. The multiagent concept permits to distribute intelligence and knowledge between "actors" or "agents" able to communicate, to exchange information and to propose an optimal solution. The functions assigned to the agents are defined in the modelling phase, according to the KADS concept.

2 Modelling optimisation knowledge from KADS

The techniques used for optimisation [10], [4] can be described by the KADS method [12] [2]. The main point of this method is its ability to extract comprehension of knowledge and the mechanisms of reasoning, both used to perform expert task assistance [3] [5]. The KADS method proposes a three levels structure to build models : domain knowledge, inference knowledge and task knowledge.

Domain knowledge describes the static knowledge using concepts, attributes, objects. The relations between these elements are represented by functions, laws or heuristics.

Inference knowledge describes how the domain knowledge is used in elementary reasoning inferences, how existing information can be turned into new information. An inference structure formalises a succession of transformations, using the knowledge domain.

Tasks knowledge describe the phases of reasoning or the aims, and the way used to achieve these aims according to the inference structures used. A recursive decomposition of a top-level task in sub-task can then be proposed.

2.1 Expertise domain

The KADS expertise model [2], designed according to the optimisation technique can be represented by figure 1.

Figure 1: Expertise Domain

Database description : concepts describing the architecture and the utilisation. We can mention :

- the logical structure (tables, index, rows, tables size, data distribution).
- the model of workload (description of SQL requests, frequencies, the number of users connected to the system)
- the statistics of the system (automatically collected by Oracle system), which enables the activity of the system to be spied on and to evaluate the use of allocated resources.

This data can only be obtained from an existing system.

Effective or real configuration : It gives the main adjustable parameters defined with the database logical structure and the workload. Particularly we can find :

- the hardware characteristics of the system (CPU, memory, disks ...)
- the physical structure (distribution of tables and index on disks)
- the parameters of the ORACLE configuration (for memory management)
- the UNIX parameters for memory and processes management.

Most of these parameters cannot be used in the case of the creation of a database.

Theoretical configuration : It gives the same parameters as a real configuration, but this configuration is obtained by the tool we have defined to evaluate theoretical performances [11]. This is an ideal configuration, in which ORACLE database could give the best performance.

Problems : The differences between the effective and the theoretical configuration indicate the bottlenecks of the transactional application according to the used resources. The aim of our study is to find the mal functioning of the database. It can be detected by an analysis of the statistics collected during the execution of the tested application.

Actions or modifications : enabling solving the problems tied to the configuration or tied to the database implementation. In the first case, it is an optimisation fault, in the second case, we try to reduce the system workload to permit correct functioning of the database.

2.2 Tasks

The relations between the tasks (figure 2) enable to represent the main steps of the design of our application to be represented. At the first level, the optimisation task corresponds to the global application. This main task is composed of three distinct sub-tasks:

Data collection : this task is decomposed according to the origins of the data. The data provided by users are generally the workload model and the hardware characteristics. The logical and physical structures, the parameters and the statistics can be collected from the database. If the database does not exist, the logical structure and the other database parameters are unknown. A checking on data consistency is checked to stop the analysis process in case of anomaly.

Analysis: this is the main part of the study. This task collects information about the problems induced by an inadequate configuration and proposes solutions to solve them. It includes four phases, each associated to a specific domain : CPU capacities, memory management, Input/Output management.

Resolution : the resolution of the problem is executed by users who select a solution among all those proposed.

In fact, this framework (figure 2) does not exactly represent the logical sequence of the tasks because the conditional and iterative structures are not mentioned. So we propose to implement a multiagent approach which can integrate the possible re-routings.

Figure 2: Distribution of Tasks

2.3 Inferences structures

Each elementary task can be described by an inference structure which specifies the logical sequence of inferences or of the actions performed. The data used and produced by each inference, called knowledge role, are defined by the model of the domain presented in the previous section. The resources needed by the reasoning task are also mentioned. In fact, only the tasks for analysis must be detailed, the other one depend on the interface domain rather than the expertise domain. Each analysis phase has its owns specific characteristics, however we have defined a generic model including the main inferences (figure 3).

Figure 3: KADS Inferences Structure

Calculate : We use an existing theoretical model to define an optimal theoretical model according to the data describing the database.

Compare : The theoretical configuration obtained is compared with the real configuration of the database. So we can detect the differences between the two configurations.

Generate : The differences in the mal functioning of the database can be generated.

Detect : The statistics enable detection of the origin of future mal functioning.

Resolve : For each problem previously detected, a set of solutions enables some modifications of the database design and configuration to be suggested.

User choice : The possible actions are proposed to users. They can select and implement some of them. Their choice depends first, on their material and financial capacities but also on the performance expected from the database.

This KADS structure, built on three levels, enables the model we propose for the system to be described. To implement it, several approaches can be used, for example traditional knowledge-based system or a multiagent system.

3 Agents and Decision Aiding System

The numerous parameters useful for database configuration and analysis in real time of their impact on the response time cannot be treated instantaneously by an operator because of the large number of operations to be performed. The agent based systems are a good solution to solve this difficulty. So we have decided to propose a multiagent approach to implement our model design from the KADS methodology.

3.1 Agents

An agent is an autonomous entity, able to perform a specified task [6], [7]. It must be able to do one or several operations and achieve one or several ends in a determined context. Moreover, it must satisfy several concepts:

- accommodation to the context,
- specificity to a task,
- autonomy for decision,
- capacity to exchange information with other agents.

”Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realise a set of goals for which they are designed” [9] [8].

An agent can react to external events, analyse information provided by others agents and decide the next action to perform [1]. In this application, for example, agents named ” expert agents ” will use the decision rules provided by the model to execute the appropriate action. The agents co-operate by exchange of messages. The model of a multiagent system decomposes an interactive system in several co-operative processing units but it does not give any indication about organisation.

3.2 The proposed system

We have built the architecture of the multiagent system and organised the agents according to the diagram of the tasks proposed by the KADS method (figure 4).

According to the KADS diagram in figure 2, we have associated a task defined in the KADS model to an agent, then we have grouped them by families of agents:

- Agent for User Interface

- Agent of Checking Consistency
- Agent for Decision
- Agent For Database Connection
- A group of agents for analysis of tasks

Each agent manages a specific function, it can solve its own problems and can ask for some more information it needs to take decision, from other agents. I.e. the "Agent for Decision" has to organise his collaboration with the group of "Agents for analysis of tasks" (CPU agent, Memory agent, Input/output agent, etc.). It must decide which of them has to begin analysis, then it has to process the information sent by these agents. After that, it must take decision about the collected information, propose solutions and send them to the "User Interface Agent", or it can decide to ask for some extra information from the "Agent of Checking Consistency".

The multiagent system processes parallel activities, by way of agents. Each agent processes information according to predefined internal rules.

The communication between each agent is organised, an agent cannot get information from any other agent. The dynamic description of the system is presented in figure 4. For example, we notice that the "Agent for User Interface" cannot exchange information with the "CPU agent", but it can communicate with the "Agent for Decision", with The "Agent of checking consistency".

Figure 4: The multiagent architecture

If an agent must respect the organisation defined for the operations of communication, it is free to take decisions about the next operations to perform. For example, the "Agent for decision" can decide to suggest

some final solutions to the "User Agent Interface". If the "User Interface Agents" does not accept these propositions and modifies some parameters of the configuration of the database, the "Agent for Decision" has to activate one of the "Agent for analysis", it must decide which of them is best suited to solving the problem.

The "Agent of Checking Consistency" and the "Agent for Decision" manage the main functions of the system. The first one has to control the coherence of the data proposed for the analysis. It can decide to get additional information from the "Database Agent" or the "User Interface Agent". The second one decides to activate the "Agents for Analysis" or to communicate with the "User Interface Agent"

4 Conclusion

This paper presents an intelligent aiding system for configuring an ORACLE database, based on the KADS method and a multiagent concept for the implementation. We can extract two main points from this approach :

- about the knowledge of optimisation of the configuration of ORACLE applications :
Beyond the capitalisation and the automation of the knowledge of designers of ORACLE applications and of the administrators of database, the projection of the knowledge in a KADS structured model incites the expert to organise his knowledge and to make a best analysis of the tasks.
- about the multiagent concept:
This approach is an excellent solution for the representation and the distribution of the knowledge. Moreover, it permits to manage interactive treatments, an essential characteristic of an Intelligent Decision Aiding System.

References

- [1] Boy, G. "Cooperation homme-machine dans les systmes complexes : Agents et aide la dcision" . Ingnierie de l'interaction homme-machine". Toulouse: Cpadrs ditions, 1995, pp.69-79
- [2] Breuker, J., Van de Velde, W. "CommonKADS library for expertise modelling". Amsterdam: IOS Press, 1994, p.1-30
- [3] W. Clancey. "Heuristic Classification". Artificial Intelligence Journal, Vol. 27, 1985
- [4] Corrigan, P., et Gurry, M. "Oracle performance tuning". Sbastopol: O'Reilly and Associates Inc., 1993, pp.201-356
- [5] L. Eshelman. "MOLE : A Knowledge- Acquisition Tool for Cover-and-Differentiate Systems". Automating Knowledge Acquisition for Expert Systems, Sandra Marcus editor, Kluwer Academic Publishers, 1988
- [6] Ferber, J. "Les systmes multi-agents". Paris: Inter ditions, 1995, 522 p.
- [7] Ferber, J. "Technologie multiagent". Conf. France Telecom Recherche : Interfaces intelligents et images. Octobre 1996, pp.63-79
- [8] Franklin S. And Graesser A.. "Is it an Agent, or just a Program?: A Taxonomy fot Autonomous Agents". Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- [9] Maes Patties, "Artificial Life Meets Entertainment : Life like Autonomous Agents", Communication of the ACM, 38,11, pp 108-114
- [10] Oracle Corporation. "ORACLE RDBMS Performance Tuning Guide Version 6.0". USA, Fvrier 1990, 180 p.

- [11] Polo, M. "Mthodes et outils d'aide la configuration et l'optimisation d'applications transactionnelles". Thse de Doctorat en Ingnierie Informatique de l'INSA de Lyon, Juillet 1995, pp.72-104
- [12] Wielinga, B.J., Schreiber, A.T. and Breuker, J.A. "KADS : a modelling approach to knowledge engineering". Knowledge Acquisition Review, Special Issue : The KADS approach to knowledge engineering, Vol.4, N1, Mars 1992, pp.5-53