Yugoslav Journal of Operations Research
1 (1991), Number 2, 179–194

TIME-SERIES MODELING IN A BLACKBOARD-BASED COUPLED EXPERT SYSTEM

Sanja PETROVIĆ

Knowledge Engineering Laboratory, "Mihajlo Pupin" Institute, P.O. Box 15, 11000 Belgrade, Yugoslavia

Ivan OBRADOVIĆ

University of Belgrade, RGF, Đušina 7, 11000 Belgrade, Yugoslavia

Radovan KRTOLICA

Knowledge Engineering Laboratory, "Mihajlo Pupin" Institute, P.O. Box 15, 11000 Belgrade, Yugoslavia

Abstract. This paper is devoted to the selection of an appropriate model for a time-series. It is a complex problem which requires both types of processing: numerical and symbolic. The idea was to develop a coupled expert system FILTEX which upgrades a professional digital signal processing software package SIG by symbolic knowledge representing expert knowledge used for time-series modeling. FILTEX blackboard architecture enables opportunistic problem-solving. The appropriateness of using frames for the blackboard implementation is briefly discussed.

Key words and phrases: time-series modeling, coupled expert systems, opportunistic problem-solving, blackboard architecture, frames.

1. INTRODUCTION

The selection of an appropriate model for a given time-series is a problem that appears in various fields of research. A typical application is the selection of a filter model in signal processing when a discrete signal is represented by a time-series

$$y(t), \qquad t \in I,$$

where t is the discrete-time index from the set of integers I.

The application of computers in signal processing (digital signal processing — DSP, (see e.g. Oppenheim and Schafer, 1975)) gradually developed in the last two decades ending in a wide range of numerical DSP software packages. A typical package consists of a library of numerical algorithms for time-series analysis and manipulation, a special data base for storing the various discrete signal representations, a graphics module and a user-oriented interface. However, the use of a DSP package requires expert knowledge, since model selection, even in the case when model class has previously been narrowed, remains a complex problem. Thus, a novice can hardly be expected to solve successfully the time-series modeling problem with no previous experience with such a tool.

A possible solution for this problem can be an upgrading of the available numerical package, by adding symbolic knowledge which represents the heuristics used by experienced time-series specialist when selecting an appropriate model. The outcome is a *coupled system* (Kowalik, 1986; Kitzmiller and Kowalik, 1988), i.e. a system that couples symbolic and numerical computing for expert time-series modeling.

This paper describes some basic ideas used in the development of FILTEX — an expert system for the design of filter models for given time-series (Krtolica et al., 1991). The approach was based on symbolic upgrading of a professional DSP software package SIG (Lager and Azevedo, 1985), achieved by coupling the numerical algorithms represented by Fortran routines in SIG with symbolic knowledge represented in IF/Prolog.

The FILTEX expert system was developed on a VAX/VMS machine in the Mihajlo Pupin Institute Knowledge Engineering Laboratory. It implements a specially designed opportunistic modeling strategy (Obradović et al., 1991) realized through a blackboard architecture of the system (Obradović et al., 1990).

The following Section outlines the arguments supporting the use of coupled systems in time-series modeling as well as the salient features of these systems. Section Three covers the specially developed strategy of model selection in FILTEX based on opportunistic reasoning and its implementation in the blackboard architecture. The frame-based approach used for the realization of this architecture is also briefly discussed. The Fourth Section is dedicated to an illustrative example of the way the system works, while the final Section gives the main conclusions.

2. TIME-SERIES MODELING: THE NEED FOR A COUPLED SYSTEM

The modeling of a time-series $y(t)$ representing a discrete signal often achieves suboptimal performance in practice. This can be viewed as a result of the superposition of following factors:

(a) lack of mathematical theory in cases when the time-series formation process is not completely known,

(b) introduction of assumptions which simplify the model in order to make the numerical processing easier,

(c) the effects of computer resource limitations (time, memory),

(d) the choice of non-optimal (arbitrarily estimated) parameter values when optimal values for the algorithm are not a priori known,

(e) measurement errors of the input data.

In order to overcome some of the resulting difficulties, generally of a numerical character, the knowledge and experience of an expert is necessary. The expert controls the time-series modeling operations by a selective activation of available algorithms and choice of appropriate parameters.

The building of systems emulating the behavior of an expert is the main goal of *knowledge engineering* i.e. the technology of expert systems. However, the classical structure of expert systems (see e.g. Waterman, 1986) is not suitable for problems requiring a large amount of sophisticated numerical computation. A need for this kind of expert systems motivated the development of expert systems which couple numerical and symbolic computing — coupled systems, in brief.

2.1. Coupling Numerical and Symbolic Computing

It is well known that the procedural programming style is not an optimal solution for an expert system data base (Rolston, 1988). Similarly, the declarative style is unsuitable for the programming of numerical routines and procedures in general. Therefore, when a data base has to include both rules and procedures (usually numerical), a purely declarative programming language (e.g. Prolog) is as uncomfortable as a purely procedural one (e.g. Fortran). In such a case it is desirable to combine the use of declarative and procedural languages, thus coupling numerical and symbolic computing.

Any system encompassing both numerical processes and symbolic reasoning could be considered as coupled. Many well-known expert systems based on symbolic reasoning, like Mycin, Prospector, Dendral or Hearsay-II, use numerical algorithms as well. However, there are important differences in the way an expert system approaches the use of numerical processes. Many existing expert systems treat numerical processes as simple functions which are "blindly" invoked to establish a numerical value. There is no particular knowledge within such systems pertaining to the numerical process itself, nor does the need for such systems, however, view numerical processes as important for reaching the solution, and the knowledge base component of the system reasons both about their application and their results. Only these systems are considered to be "true" coupled systems, the depth of coupling being measured by the amount of knowledge about the involved numerical processes.

2.2. Coupled Systems in Time-Series Modeling

Many time-series modeling algorithms combine techniques based on formal mathematical theory with non-formal heuristics and available knowledge about the environment in which the time-series was sampled. The design of algorithms using well-known mathematical concepts, such as convolution, domain transformation, etc., is usually based on some kind of idea about the process generating the time-series. However, during the development and testing of these algorithms shortcomings in performances appear, due to an incomplete theoretical model and erroneous assumptions about the attributes of data, i.e. their discordance with characteristics of observed time-series. This leads to an adjustment of algorithm parameters based on experience, as well as the knowledge about the underlying mathematical theory and the knowledge about the desired output. The outcome of this adjustment can be a considerable correction of algorithm performances. The interactive use of formal time-series mathematical models and non-formal heuristics used for their modeling can end in successful strategies for problem solving. The architecture of coupled systems provides for a corresponding feedback control loop supporting this interaction.

Coupled systems for time-series modeling can be divided into at least three classes according to the adopted strategical paradigm (Sharman and Durrani, 1990). These paradigms are:

(a) selection of data ("focusing of attention"),

(b) choice of algorithms and parameter adjustment, and

(c) generation and execution of plans for time-series modeling.

The main goal of the systems characterized by data selection is to limit the amount of data to be processed, i.e. to "focus the attention" of the data processing operations on parts of the available time-series. This strategy is suitable for applications where time-signal operations require a great amount of computing resources, when these operations can be used in a selective manner. The symbolic component of the system decides when and how a certain operation can contribute to the time-series modeling goal and realizes them accordingly. This strategy reduces resource requirements avoiding to blindly process all data all the time.

The systems whose strategic paradigm is based on the choice of algorithms and parameter adjustment improve the quality and reliability of the time-series processing operations by choosing the most suitable algorithms and/or their parameters on the basis of available information. This strategy can be applied in the case of an incomplete model of the time-series modeling process, when informal tuning of the process can improve system performance for the time-series at hand. The symbolic component in this class of coupled systems controls the use of time-series modeling operations on the basis of the knowledge about the observed time-series and/or available modeling heuristics.

When the time-series modeling task is ill-defined or not clearly stated, an interactive approach is used, in which an experienced expert decides what operations from the available DSP package should be applied, and in what order. The strategy used by the expert, based on the "generate and test" paradigm (Rich, 1983), is known as *interpretation guided time-series processing*. The expert tries to formulate and realize dynamically a time-series modeling plan in the context of the current assumption about the source generating the time-series, either to strengthen this assumption or to point out to errors in interpretation.

The outlined strategic paradigms are not mutually exclusive and can be combined in different ways to achieve the desired performance of the coupled system when solving a particular problem.

Finally, let us briefly state the advantages of time-series modeling by means of a coupled system. These advantages lie in:

(a) a reduction of resource requirements,
(b) possibilities for error correction, and
(c) performance improvement.

A coupled system reduces computer resource requirements by a careful choice of algorithms and their use, and by the mechanism of "focusing of attention", i.e. selection of data to be processed, which can considerably reduce numerical computation. Error correction in time-series interpretation is provided in coupled systems by their cyclic character: the successive activation of numerical and symbolic processes. This means that the results obtained by time-series processing can be independently evaluated and if an error is detected, the time-series operators can be regrouped and reused. Finally, coupled systems can improve the robustness and performances of time-series modeling operations by controlling the choice of algorithms and their parameters as well as their use. The possible criteria are expected algorithm performances on the basis of *established* attributes of the time-series (data-driven selection), or on the basis of *assumed* attributes (model-driven selection).

3. REALIZATION OF THE TIME-SERIES MODELING STRATEGY IN FILTEX

The coupled expert system for the design of time-series filter models FILTEX is based on a specially developed opportunistic strategy realized through the blackboard architecture of the system. We shall now give a brief review of this strategy and the main architectural characteristics of the system.

3.1. Time-Series Modeling Strategy

The main goal of the model selection strategy in FILTEX is the choice of a linear predictive filter, on the basis of a mathematical model for the observed time-series. For practical purposes, the allowable filter models were restricted to the class of linear finite-dimensional parametric SISO (single-input single-output) models. Even so, the number of possible models for a given time-series is almost infinite. The design of a predictive filter, being such a complex problem, allows for multiple problem-solving strategies. Consequently, the representation of relevant theoretical and empirical knowledge in the system leads to interesting knowledge-base design problems.

The general model of the time-series used is the *autoregressive moving average model with exogenous input* (ARMAX) (Candy, 1986):

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t)$$

where A, B and C are polynomials and q represents the backward-shift operator with the property that $q^{-k}y(t) = y(t-k)$. Namely:

$$A(q^{-1})y(t) = y(t) + a_1 y(t-1) + \cdots + a_{N_a} y(t - N_a)$$

$$B(q^{-1})y(t) = b_0u(t) + b_1u(t-1) + \cdots + b_{N_b}u(t-N_b)$$
$$C(q^{-1})y(t) = e(t) + c_1e(t-1) + \cdots + c_{N_c}y(t-N_c)$$

where $y(t)$ stands for the output time-series, $u(t)$ for the exogenous input and $e(t)$ for a white-noise time-series. The ARMAX model, usually abbreviated by ARMAX(N_a, N_b, N_c), represents the general form of the model. Other model forms can be deduced from ARMAX, such as the autoregressive (AR) model: $B(\cdot) = 0$, $C(\cdot) = 1$, the moving average (MA) model: $A(\cdot) = 1$, $B(\cdot) = 0$, or the autoregressive model with exogenous input (ARX): $C(\cdot) = 1$.

The modeling problem can be viewed as the selection of the ARMAX model *structure* (N_a, N_b, N_c) and determination of appropriate parameters. In addition, the user usually sets his requirements for filter performances, e.g. by imposing limits on model complexity ($s = N_a + N_b + N_c$) or final prediction error.

The adopted strategic paradigm for predictive filter design in FILTEX is the search for the least complex acceptable solution (the principle of parsimony). In an attempt to satisfy this goal a model from each of the elementary classes (AR, MA, X) is proposed according to local optimization rules. The obtained models which satisfy user constraints form the set of model-candidates. If no elementary model-candidates are found, more complex classes (ARMA, ARX, MAX, ARMAX) are investigated by combining the two (three) elementary models. If this doesn't give a model-candidate, then a search for a model-candidate is performed in the "vicinity" of the model "closest" to the solution, i.e. a model which satisfies at least one of the requirements. For example, if a model with allowable complexity but unacceptable final prediction error exists, lowering or increasing of its complexity might lead to a model with acceptable error.

However, if there are no models "close" to the solution, an extensive search is performed in the limits of allowable model complexity. If this final attempt doesn't succeed, the user is asked to change the imposed constraints. If at any point in the aforementioned strategy at least one model-candidate appears, then the set of model-candidates is widened with models obtained by lowering the complexity of the least complex model. Finally, the set of model-candidates is offered to the user with models sorted according to a desirable criterion: complexity, final prediction error or variance.

3.2. A Blackboard Approach to Model Selection

The initial experience with the FILTEX system indicated that the solving of the model selection problem could not be based on a single method of reasoning, e.g. backward or forward chaining, but required their opportunistic combination. The model selection problem is characterized by lots of intermediate results to which various algorithms and heuristics can potentially be applied. There is a need for careful choice of action at each problem-solving step. Such an approach to the search for the solution, by selecting from the set of possible actions the action which will contribute to the solution mostly and at the same time uses the most favorable method of reasoning, is referred to as *opportunistic problem solving*

(Erman et. al., 1988). Opportunistic problem solving is based on *opportunistic reasoning*, which is defined as reasoning combining backward and forward chaining according to problem solving needs.

The classical expert system structure appears to be inadequate for opportunistic problem solving for several reasons (Engelmore, 1988). Knowledge representation in such an architecture is dependent on the nature of the inference engine (for example a rule interpreter requires knowledge expressed as rules). Inference engines are usually designed to support single methods of reasoning. The control of knowledge application is not very flexible because it lies implicitly in the structure of the knowledge base (for example in the ordering of the rules for a rule based system).

It is well known that techniques and methods for the decomposition of complex programs into a number of less complex modules are in wide use. The communication among modules becomes efficient if there is a global data base which stores information generated during the problem-solving process. Following these ideas a new problem-solving model called the *blackboard* was proposed (Nii, 1986).

The blackboard model is based on two main components:

1. *Knowledge sources* (KS). The domain knowledge needed to solve a problem is partitioned into logically independent parts — knowledge sources. Independence means that knowledge sources have no information about each other's behavior, but they rely on other knowledge sources results of expertise. Problem-solving is achieved by cooperative activity of various knowledge sources.
2. *Blackboard*. The global data base called the blackboard has a double role:
   - The blackboard holds all input data, partial solutions, alternatives and final solution. Information available on the blackboard represents the current state of the solution.
   - The blackboard enables communication among knowledge sources. The current state of the blackboard determines which of the knowledge sources can contribute to the forming of the solution in the next step. Knowledge sources take the information they need from the blackboard and create new information as a result of their activation. The new state of the blackboard enables participation of other knowledge sources in problem-solving.

3.3. Coordination of Heterogeneous Knowledge Sources

The model selection strategy used in FILTEX is oriented towards objects — knowledge modules determined by the model classes: AR, MA, ARMA, etc. These modules are realized through knowledge sources which propose models from appropriate classes. The knowledge sources which propose models from appropriate classes. The knowledge sources are divided into two groups: ELementary MODels (EL_MOD) and COMplex MODels (COM_MOD). They propose models from one of the corresponding classes: AR, MA, X and ARMAL, ARX, MAX, ARMAX, respectively (Figure 1). Knowledge sources use SIG numerical routines to

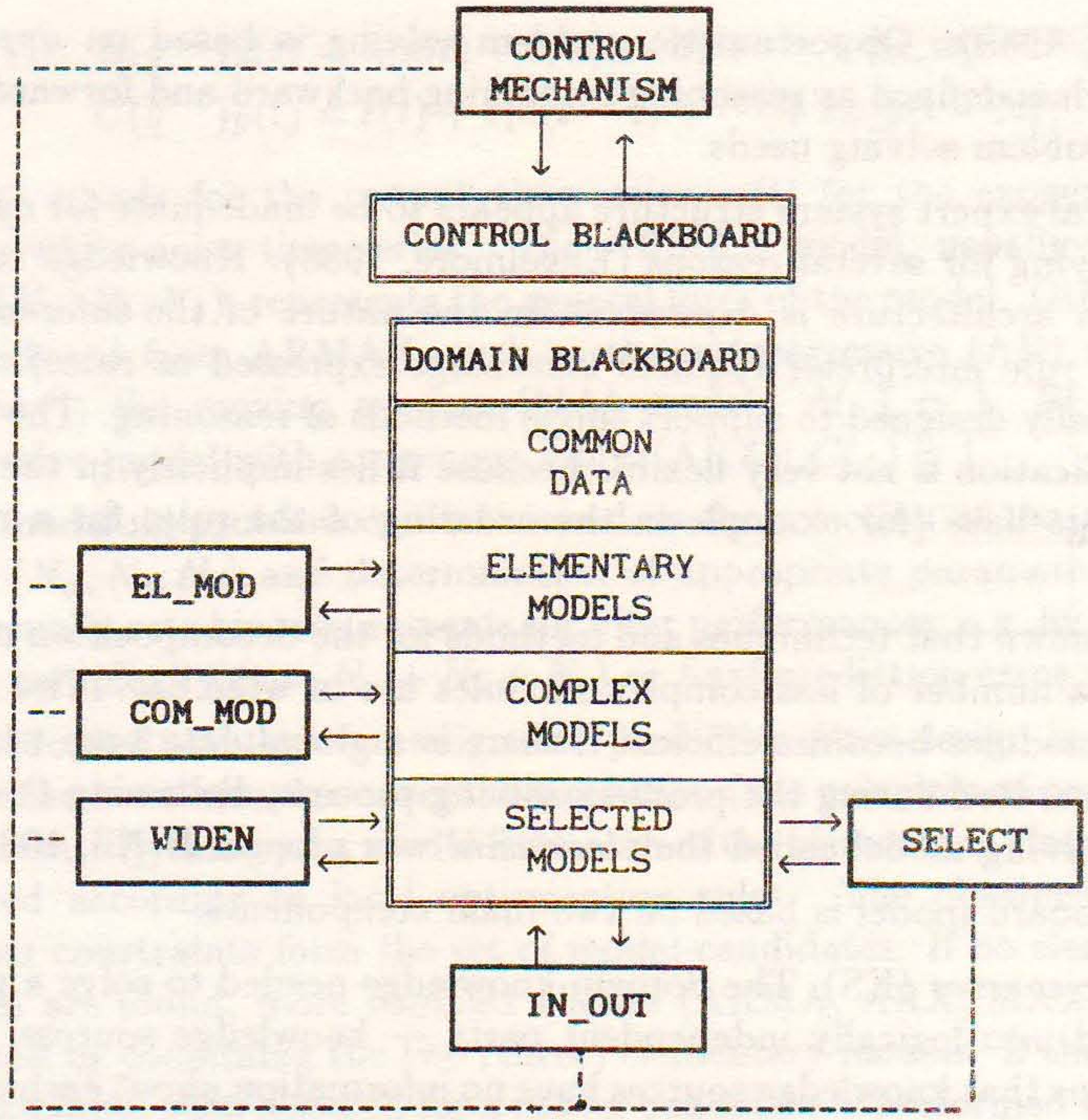

Figure 1. The FILTEX blackboard structure

estimate model parameters. The proposed models, written on the blackboard, represent different hypotheses about the solution. These hypotheses are integrated by knowledge sources from the *SELECT* group. They examine the proposed models and select those which satisfy the imposed constraints. Knowledge sources in the *WIDEN* group contain heuristic rules related to the part of the strategy aimed at producing additional model candidates. This can be realized in several ways: by looking for model-candidates in the "vicinity" of particular models which are not candidates but are considered to be "close" to one of them, by lowering the complexity of a selected model-candidate and by an extensive search within the limits of the allowable model complexity. Communication with the user is handled by the *IN_OUT* knowledge source group.

The outlined knowledge partitioning suggests an organization of the blackboard into hierarchically related levels, corresponding to inferences of knowledge source groups (Figure 1). Each knowledge source group has access to information on the associated level as well as on all higher levels. The hierarchical organization enables a faster and more transparent functioning of the system.

Coordination of heterogeneous knowledge sources is in the competence of the control mechanism. In order to separate the flows of domain data and control data, the blackboard is divided into two parts: domain and control blackboard. All information related to model selection is kept on the domain blackboard. Information needed by the control mechanism is stored on the control blackboard.

The control mechanism is realized through a loop which invokes three control procedures: *Precondition_test*, *Select_KS*, *Activate_KS* (Figure 2).

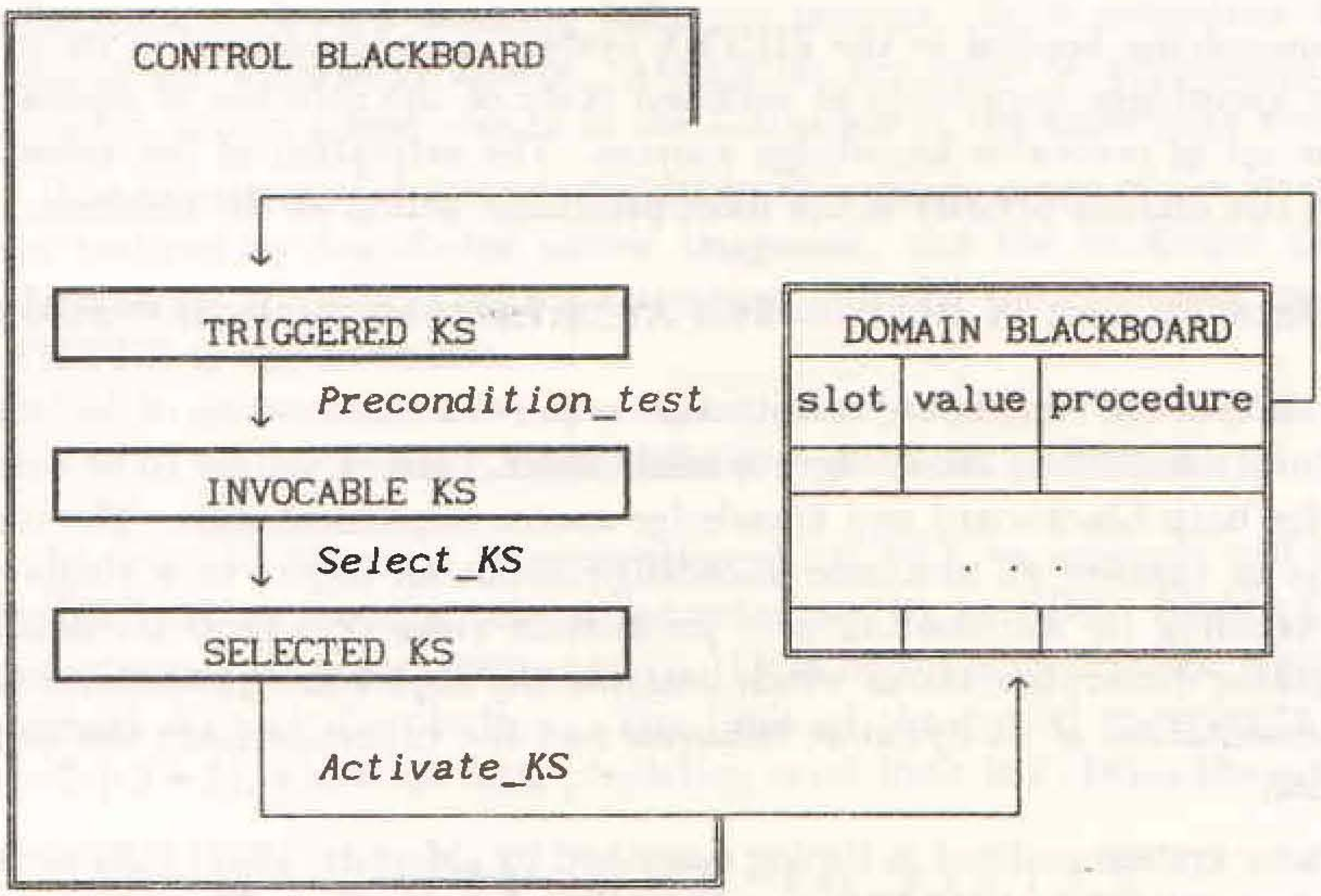

Figure 2. The FILTEX control mechanism

The activation of a knowledge source results in adding or deleting the values of selected blackboard attributes. These events trigger knowledge sources that are potentially applicable in the next step of the problem-solving process. The reasoning process of a knowledge source may need particular domain blackboard data. The existence of these data is a precondition for its activation. The fulfillment of preconditions is tested by the control procedure *Precondition_test*. Only triggered knowledge sources which pass the precondition test can participate in the problem-solving process. These knowledge sources are called "invocable". Activation of the remainder of triggered knowledge sources is postponed until the domain blackboard reaches the appropriate state.

The control procedure *Select_KS* selects one knowledge source from the set of invocable knowledge sources. The selection is based on priorities assigned to knowledge sources. These priorities reflect the adopted strategy which will be used for problem-solving.

The activation of the selected knowledge source is performed by the control procedure *Activate_KS*. The knowledge source activation modifies the state of the domain blackboard. This triggers new knowledge sources and the loop restarts.

In each problem-solving step, control procedures change the values of three control blackboard attributes: *triggered*, *invocable* and *selected*.

The loop ends in two cases:

1. If there are no more invocable knowledge sources.

2. When the activation of a knowledge source is unsuccessful due to an error which makes further functioning of the system impossible (for example if there is a severe error in time-series data).

Problem-solving applied in the FILTEX system is opportunistic in its proper sense. The knowledge applicable at attained state of the solution is represented through the set of invocable knowledge sources. The activation of the knowledge source with the highest priority is the most profitable action at the moment.

3.4. Using Frames in Blackboard Architecture Implementation

Realization of the blackboard architecture requires a combination of declarative and procedural methods of knowledge representation. Frames appear to be suitable structures for both blackboard and knowledge source implementation. The idea of the frame is to capture all available knowledge about an object in a single data structure, avoiding its distribution over production rules or logical formulas. A frame integrates declarative terms which describe the object and procedural terms for the representation of its dynamic behavior and the expertise from the domain of application.

The frame system realized in Prolog described by (Merritt, 1989) fully satisfied all the requirements of the FILTEX components implementation. The internal data structure holding frame information is a predicate *frame(Name,Slot_list)* with two arguments, the first being the name of the frame and the second — the list of slots. Slots describe the properties — attributes of the object represented by the frame. The list of slots contains the values of slots and facets which describe the nature of these values. There are five possible facets:

1) *val* denotes the current value of the slot,
2) *def* denotes the default value of the slot which is valid in the absence of other information,
3) *calc* denotes that the procedure attached to the slot calculates the value of the slot when it is needed,
4) *add* denotes that the procedure attached to the slot will be activated when a value is added to the slot,
5) *del* denotes that the procedure attached to the slot will be activated when the slot value is deleted.

Access to data stored in the frame is controlled by three predicates:

1) *get_frame(Frame,Slot-Value)* which retrieves a slot value,
2) *add_frame(Frame,Slot-Value)* which adds or updates a slot value,
3) *del_frame(Frame,Slot)* which deletes a slot value.

The procedures denoted by *add* or *del* facets, which are activated when the slot value is changed, enable the forward chaining inference process. This property is used for knowledge sources triggering. After setting or deleting the value of the slot, procedures attached to the slots of the domain blackboard frames report to the control mechanism which knowledge sources are triggered. The procedure in the *calc* facet, invoked when the value of a slot is required, demonstrates the behavior similar to a backward chaining inference process. Such procedure describes the action of the knowledge source. Asking for the value of a particular slot in the knowledge source frame results in the activation of the knowledge source.

The existence of slot facets results in alternative usage of the forward chaining steps realized by knowledge source triggering, and the backward chaining steps performed by knowledge source activation. Simply, the reasoning method applied in FILTEX is opportunistic.

4. AN EXAMPLE OF A FILTEX SESSION

In order to illustrate the operation of FILTEX an example will be presented. Suppose that we model a time-series generated by an PRBS ARMAX model, with a pseudorandom binary sequence $(0;10)$ as input and discrete white noise with a normal $N(0;1)$ distribution. The limit of the model complexity is set to 16 $(s = 3+2+3)$, while the final prediction error limit is 2 (twice the noise variance).

W E L C O M E T O F I L T E X

EXpert system for FILTer design

Enter the data store number of the output time-series [1,899]: **221**

Does the input time-series exist? (y/n) **y**

Enter the data store number of the input time-series [1,899]: **2**

Do you want to set a limit to model complexity? (y/n) **y**

Enter the allowable model complexity [1–150]: **16**

Do you know the variance of the white-noise? (y/n) **n**

Do you want to set a limit to final prediction error? (y/n) **y**

Enter the acceptable final prediction error: **2**

According to the selected strategy, FILTEX is looking for an elementary model which satisfies the imposed constraints. The identification is based on a set of rules for local optimization.

Since the input time-series exists, the identification of the X-model is possible. The time of impulse response disappearing is accepted for the upper limit of the B-order. The maximum value is 4, and if this limit is surpassed the B-order is set to 1. This is in accordance with the principle of parsimony.

Examination of the structure [0,1,0].

Structure: $[0, 1, 0]$ Final prediction error = 195.867

The following model is extracted:

X-model
Final prediction error = 195.867 Complexity = 1 Variance = 194.343
$N_a = 0$ Parameters: 1.0
$N_b = 1$ Parameters: 0.0, 1.38837
$N_c = 0$ Parameters: 1.0

The following elementary model is obtained:

X [0, 1, 0] Final prediction error = 195.867 Complexity = 1

Models with allowable complexity are listed below:

X [0, 1, 0] Final prediction error = 195.867 Complexity = 1

THERE ARE NO MODELS WITH ACCEPTABLE FINAL PREDICTION ERROR!

None of the proposed elementary models satisfies both constraints: allowable model complexity and acceptable final prediction error. The search for the model is continued in complex classes that combine the two (three) elementary models.

Identification of the ARX-model is based on the strategy: B-order (1) is taken from the identified X-model, A-order is increased until a local minimum is reached.
Structure: [1, 1, 0] Final prediction error = 10.4964
Structure: [2, 1, 0] Final prediction error = 3.33649
Structure: [3, 1, 0] Final prediction error = 3.11888
Structure: [4, 1, 0] Final prediction error = 3.14028

The following model is extracted:

ARX-model
Final prediction error = 3.11888 Complexity = 4 Variance = 3.02291
$N_a = 3$ Parameters: 1.0, 1.4158, 0.393337, -0,1239925
$N_b = 1$ Parameters: 0.0, 1.04291
$N_c = 0$ Parameters: 1.0

Identification of the MAX-model is based on the strategy:
B-order (1) is taken from the identified X-model, C-order is increased until a local minimum is reached.

Structure: [0, 1, 1] Final prediction error = 120.398
Structure: [0, 1, 2] Final prediction error = 116.313
Structure: [0, 1, 3] Final prediction error = 1.437056e+07

The following model is extracted:

MAX-model
Final prediction error = 116.313 Complexity = 3 Variance = 113.618
$N_a = 0$ Parameters: 1.0
$N_b = 1$ Parameters: 0.0, 1.15459

$N_c = 2$ Parameters: 1.0, - 1.0241, 0.893791

Identification of the ARMAX-model is based on the strategy:
B-order (1) is taken from the identified X-model, the total complexity of the A and C polynomials is increased (till 2, because this is the minimum of two values: A-order in the ARX-model and C-order in the MAX-model) until a local minimum is reached.

Structure: [1, 1, 1] Final prediction error = 8.95441

The following model is extracted:

ARMAX-model
Final prediction error = 8.95441 Complexity = 3 Variance = 8.74697
$N_a = 1$ Parameters: 1.0, 0.916513
$N_b = 1$ Parameters: 0.0, 1.05488
$N_c = 1$ Parameters: 1.0, -0.508699

The following set of complex models is obtained:

ARMAX	[1, 1, 1]	Final prediction error = 8.95441	Complexity = 3
MAX	[0, 1, 2]	Final prediction error = 116.313	Complexity = 3
ARX	[3, 1, 0]	Final prediction error = 3.11888	Complexity = 4

Models with allowable complexity are listed below:

ARMAX	[1, 1, 1]	Final prediction error = 8.95441	Complexity = 3
MAX	[0, 1, 2]	Final prediction error = 116.313	Complexity = 3
ARX	[3, 1, 0]	Final prediction error = 3.11888	Complexity = 4
X	[0, 1, 0]	Final prediction error = 195.867	Complexity = 1

THERE ARE NO MODELS WITH ACCEPTABLE FINAL PREDICTION ERROR!

The set of the models with allowable model complexity is not empty:

ARMAX	[1, 1, 1]	Final prediction error = 8.95441	Complexity = 3
MAX	[0, 1, 2]	Final prediction error = 116.313	Complexity = 3
ARX	[3, 1, 0]	Final prediction error = 3.11888	Complexity = 4
X	[0, 1, 0]	Final prediction error = 195.867	Complexity = 1

The model with the smallest final prediction error is chosen:

Structure: [3, 1, 0] Final prediction error = 3.11888

MODEL WITH THE STRUCTURE [3, 1, 0] IS CLOSEST TO THE SOLUTION!

Variation of complexity of the structure [3, 1, 0] is performed in order to find the model with acceptable final prediction error.

The complexity of the structure [3, 1, 0] is lowered.

Structure: [2, 1, 0] Final prediction error = 3.33649
(identified before)

The complexity of the structure [3, 1, 0] is increased.

Structure: [4, 1, 0] Final prediction error = 3.14028
(identified before)

Structure: [3, 2, 0] Final prediction error = 2.04834
Structure: [3, 1, 1] Final prediction error = 1.89387

Model with the structure [3, 1, 1] has the smallest final prediction error: 1.89387.

In the set of models obtained by variation of the complexity of the structure [3, 1, 0], the model with the structure [3, 1, 1] has the smallest final prediction error.

The following model with allowable model complexity and acceptable final prediction error is obtained:

ARMAX-model
Final prediction error = 1.89387 Complexity = 4 Variance = 1.8213
$N_a = 3$ Parameters: 1.0, 1.52648, 0.551559, -7.273768e-02
$N_b = 1$ Parameters: 0.0,1 0252
$N_c = 1$ Parameters: 1,0, 0.832533

The set of models with allowable model complexity and acceptable final prediction error is not empty:

ARMAX [3, 1, 1] Final prediction error = 1.89387 Complexity = 5

FILTEX will try to widen this set with less complex models.
The complexity of the structure [3, 1, 1] is lowered.

Structure: [2, 1, 1] Final prediction error = 1.98117
Structure: [3, 1, 0] Final prediction error = 3.11888
(identified before)

Model with the structure [2, 1, 1] has the smallest final prediction error: 1.98117.

The following model is added to the set:

ARMAX-model
Final prediction error = 1.98117 Complexity = 4 Variance = 1.92021
$N_a = 2$ Parameters: 1,0, 1.5754, 0.668532
$N_b = 1$ Parameters: 0.0, 1.02625
$N_c = 1$ Parameters: 1.0, 0.782808

The complexity of the structure [2, 1, 1] is lowered.

Structure: [1, 1, 1] Final prediction error = 8.954441
(identified before)

Structure: [2, 1, 0] Final prediction error = 3.33649
(identified before)

Model with the structure [2, 1, 0] has the smallest final prediction error: 3.33649.

Final prediction error of the model surpassed the allowable limit!

The lowering of complexity stops!

CHOOSE ONE OF THE FOLLOWING CRITERIA
FOR THE RANKING OF MODELS:

1. Final prediction error
2. Model complexity ($N_a + N_b + N_c$)
3. Variance of the residual sequence
4. End

Chosen criterion [1–4]: 1

MODELS RANKED ACCORDING TO FINAL PREDICTION ERROR:

ARMAX-model
Final prediction error = 1.89387 Complexity = 5 Variance = 1.8213
$N_a = 3$ Parameters: 1,0, 1.52648, 0.551559, -7.273768e-02
$N_b = 1$ Parameters: 0.0, 1.0252
$N_c = 1$ Parameters: 1.0, 0.832533

ARMAX-model
Final prediction error = 1.98117 Complexity = 4 Variance = 1.92021
$N_a = 2$ Parameters: 1.0, 1.5754, 0.668532
$N_b = 1$ Parameters: 0.0, 1.02625
$N_c = 1$ Parameters: 1.0, 0.782808

5. CONCLUSIONS

The FILTEX system whose main features were described in this paper was tested on a set of representative time-series generated by various ARMAX models using the simulating facilities offered by SIG. Thus, the experimental environment could be fully controlled and the obtained results carefully evaluated. These results justified the approach, since an appropriate model was proposed by the system in more than 90% of cases.

The importance of the methodology used for building FILTEX lies in the fact that it can be used in similar situations, i.e. when a problem requiring a large amount of numerical computation by an existing software package needs symbolic upgrading. By this we mean the coupling of the numerical package with Prolog, the development of an appropriate strategy, and the realization of this strategy in the blackboard architecture.

The fact that FILTEX was developed on a VAX/VMS machine was a consequence of the numerical package SIG being available only on this system. However, in the Mihajlo Pupin Institute Knowledge Engineering Laboratory this numerical package was successfully ported to an IBM PC computer, thus opening the possibility of realizing a PC version of the FILTEX expert system. This project is now in progress, and the coupling of ported SIG (named SIG+) with Arity/Prolog is under way.

REFERENCES

Candy, J. (1986), *Signal Processing. The Model-Based Approach*, McGraw-Hill Book Company New York.

Engelmore, R. and Morgan, T. (eds.) (1988), *Blackboard Systems*, Addison-Wesley.

Erman, L.D., Hayes-Roth, F., Lesser, V. R. and Reddy, D.R. (1988), *The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty*, in: Engelmore, R. and Morgan, T. (eds.), *Chapter 3*, 31–86.

Kowalik, J.S. (ed.) (1986). *Coupling Symbolic and Numerical Computing in Expert Systems*, North-Holland, Amsterdam.

Kowalik, J. S., and C. T. Kitzmiller (eds.) (1988), *Coupling Symbolic and Numerical Computing in Expert Systems, II*, North-Holland, Amsterdam.

Krtolica, R., I. Obradović, and R. Božinović (1991), *FILTEX: Toward an expert system for time-series filter design*, Journal of Applied Statistics, Special Issue on Expert Systems and Statistics **18** (1), 121–137.

Lager, D., and S. Azevedo (1985), *SIG, a general purpose signal processing program*, User's manual, Lawrence Livermore National Laboratory, UCID-19912-Rev 1, Livermore, California.

Merritt, D. (1989), *Building Expert Systems in Prolog*, Springer-Verlag.

Nii, P. H. (1986), *The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures (Part 1)*, AI Magazine 7 (2), 38–53.

Obradović, I., R. Krtolica, S. Petrović, and D. Radojević (1990), *A Blackboard Approach to Problem-Solving Strategies in Filter Selection*, in: *Mathematical and Intelligent Models in System Simulation*, R. Hanus, P. Kool and S. Tzafestas (eds.), J. C. Baltzer AG, Scientific Publishing Co., Basel, 331–336.

Oppenheim, A.V. and R.W.Schafer (1975), *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

Rich. E. (1983), *Artificial Intelligence*, McGraw-Hill, New York.

Rolston, D. W. (1988). *Principles of Artificial Intelligence and Expert Systems Development*, McGraw-Hill, New York.

Sharman, D. B., and T. S. Durrani (1990). *Coupled systems for signal processing*, IEEE Proceedings **137**, Pt.F, No. 5, 331–336.

Waterman, D. A. (1986), *A Guide to Expert Systems*, Addison-Wesley, Reading, Mass.