PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 34, NO. 3, PP. 205–220 (1990)

ON IMPLEMENTATION OF ROBUST AUTOTUNING OF TRANSMISSION ELECTRON MICROSCOPES

Z. PAPP[1]

Department of Applied Physics
Delft University of Technology, The Netherlands

Received October 31, 1990; Revised December 20, 1990.

Abstract

Practice shows that the current implementations of automatic tuning of transmission electron microscopes suffer from not satisfactory robustness, and this seriously limits their applicability. The paper presents a software architecture which provides a framework for the realization of a real-time automatic tuning system with improved robustness. First the transmission electron microscope tuning as general measuring/modelling process is characterized and the consequences of the improvement in robustness are identified in this context. It is concluded that both extending the models of image formation of the electron microscope into qualitative and heuristic directions, and the continuous model validation with sophisticated control are necessary for coping with these problems. Then a two-layer software architecture is presented which helps satisfying the above requirements to a considerable extent: the lower layer contains the conventional and symbolic data/image processing components (with data/control interfaces), the upper layer — using knowledge based approach extensively — realizes the higher level control based on the partial results of the processing on the lower level. (Hence, the upper level is responsible for the robustness in system-wide sense.) Main subsystems of the autotuning software are shown. A short survey of the hardware background is also given. A summary closes the paper.

Keywords: transmission electron microscope, autotuning, artificial intelligence techniques in instrumentation.

Introduction

Transmission electron microscopes (TEMs) are important measuring instruments in the research on biological and inorganic structures as well. Practice shows that among the limiting factors of TEM application the most serious one is the skill of the operator, who should tune the instrument accurately — eventually under difficult imaging conditions (e.g. low intensity, noisy image, limited observation time). Consequently, a demand for automatic TEM tuning (autotuning) has arisen. Intensive research is

---

[1]On leave from the Department of Measurement and Instrument Engineering, Technical University of Budapest, Hungary.

carried out by several groups and laboratories all over the world to realize autonomous real-time measuring systems to support TEM operators by taking over certain parts of the tasks and responsibilities of designing and implementing TEM tuning.

This work is based on the results of the long term and intensive research on TEM autotuning carried out at the Delft University of Technology, Department of Applied Physics, and on intelligent measuring systems at the Technical University of Budapest, Department of Measurement and Instrument Engineering. Our contribution concerns how knowledge based (KB) and other artificial intelligence (AI) based techniques can be integrated with the traditional signal and image processing techniques, how they can support the robust, in some sense intelligent operation (PAPP, 1990b).

The results of the first phase of the research are summarized in PAPP (1990a). This paper can be considered as a short extract of the technical report cited.

The outline of the paper is as follows. First the autotuning of TEMs as measuring/modelling process is characterized, then the theoretical and implementational problems of the autotuning are summarized. The second part shows the proposed software architecture for a real-time autotuning measuring system, which — probably — can serve as a basis for algorithm and methodology research and for application systems as well, for a long term. Next, a concise survey of the underlying hardware background follows, finally conclusions close the paper.

TEM Autotuning as a Measuring/Modelling Process: an Overview

The goal of the tuning of TEMs is to minimize the artifacts in the image of the specimen under observation. This minimisation is carried out by setting optimal TEM parameters (REINER, 1984). In the routine, every-day use, the TEM

—has to be *aligned*,

—the *astigmatism* has to be corrected,

—known *defocus* has to be set.

Several methods based on completely different approaches have been developed for TEM autotuning in the last 10–15 years. They differ from each other in many respects, such as

—instrumentation requirements,

—computational requirements,

—complexity of the image formation model applied,

—achievable precision,

—dose efficiency, etc.

(An extensive historical overview and assessment of the methods can be found in KOSTER, 1989.)

The advances in the computer and image processing technology have given a basis to the application of the more sophisticated (consequently more precise) autotuning methods. One of the most promising approaches is the *deliberate beam tilt induced image shift measurement* based tuning, which is thoroughly investigated at the Delft University of Technology, Department of Applied Physics (e.g. KOSTER, 1989; RUIJTER, 1988). From now on the application of this tuning approach is assumed.

From the signal processing point of view the tuning process can be represented as shown in *Fig. 1*. The operation conditions of the TEM are determined by the $\boldsymbol{P}$ parameter set. $\boldsymbol{P}$ can be divided into two subsets: $\boldsymbol{P}_1$ contains the TEM parameters, which are not modified during the tuning process (but are relevant from the image formation point of view) and

$$\boldsymbol{P}_2 = \{\mathbf{misalignment},\ \mathbf{astigmatism}, defocus\},$$

elements of which are the controlled parameters[2].

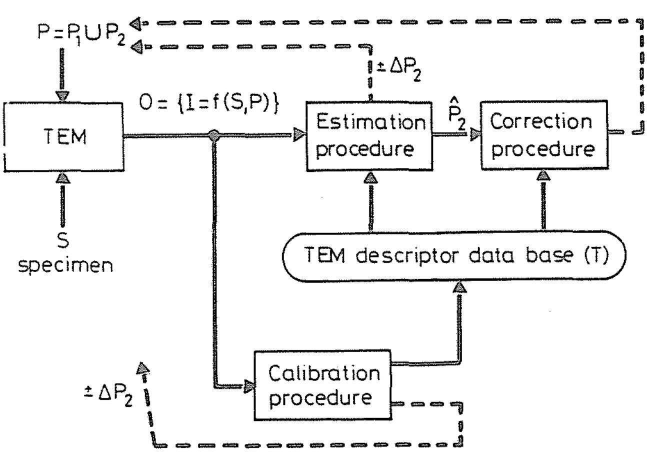

Fig. 1. The TEM tuning process

[2]The **misalignment** and the **astigmatism** are 2-dimensional vector parameters (interpreted as misalignment and astigmatism in two orthogonal directions). The *defocus* is scalar.

Inserting a specimen into the TEM (specimen model is denoted by $\boldsymbol{S}$), image $\boldsymbol{I}$ can be observed on the screen. The image observed depends on the specimen examined and the actual values of the parameters in $\boldsymbol{P}$.

A sequence of images (recorded at different parameter values in $\boldsymbol{P}$, eventually at different $\boldsymbol{S}$) serves as input ($\boldsymbol{O}$, observation) of the *estimation procedure*. The estimation procedure, using the TEM description data base $\boldsymbol{T}$, delivers in some sense optimal estimator for parameters in $\boldsymbol{P}_2$ (denoted as $\widehat{\boldsymbol{P}}_2$). Based on $\widehat{\boldsymbol{P}}_2$ the *correction procedure* corrects (or sets a known value for) misalignment, astigmatism and defocus.

The elements of the $\boldsymbol{T}$ data base are calculated by the *calibration procedure*. Both the estimation and the calibration procedures use deliberate beam tilt induced image shift measurements to deliver their outputs. (In the figure $\Delta\boldsymbol{P}_2$ means known, deliberate 'detuning' from the actual parameter values.)

The computerized implementation of the autotuning process has a number of difficulties. It can easily be shown that each of them is — to a certain extent — responsible for the not satisfactory level of robustness of the existing implementations. The most characteristic difficulties can be categorized into three classes, as

—image formation,
—parameter estimation and
—practical limitations.

Image Formation

The autotuning procedure cannot be derived without the modelling of the image formation of the TEM. Derivation of the model claims theoretical efforts and presently it seems that the modelling cannot be solved in general. Quantitative models have been derived for particular classes of specimens, e.g. for weak phase objects and amplitude contrast objects (KOSTER, 1989; REINER, 1984)[3]. Unfortunately, even for these specimen classes, only simplified linear models have been developed and, as a consequence, the validity of these models is strongly limited in the case of large scale detuning or TEM parameter variations. Since the autotuning is based on the estimation procedure, which inherently depends on the image formation model (and its validity), the image formation model is of key importance in the robustness point of view.

[3]According to these models the effect of the adjustment of a parameter $p \in \boldsymbol{P}_2$ results in mere image shift on the screen, and by satisfying certain precedence constraints the value of the parameters in $\boldsymbol{P}_2$ can be estimated separately. For details see KOSTER, 1989.

Parameter Estimation

The estimation procedure realizes a mapping between the observations (sequences of images recorded at different (partially unknown) TEM parameters) and the estimations of the relevant parameters:

$$\widehat{\boldsymbol{P}}_2 = \text{mapping}_{S_i}(\boldsymbol{O}; \boldsymbol{P}, \boldsymbol{F}),$$

where

$\boldsymbol{O}$: observations,

$\boldsymbol{P} = \boldsymbol{P}_1 \cup \boldsymbol{P}_2$: the current TEM parameters,

$\boldsymbol{F}$: free variables of the experiment.

The structure of the estimation procedure is determined by the specimen model assumed ($\boldsymbol{S}_i$). The estimated value ($\widehat{p} \in \widehat{\boldsymbol{P}}_2$) of a certain parameter ($p \in \boldsymbol{P}_2$) depends on the current values of other parameters in $\boldsymbol{P}_1 \cup \boldsymbol{P}_2$, which — in part — are also unknown, since they are also subjects of estimation. Moreover, the estimation procedure has its own parameters also. One of the subsets of these parameters can be considered as *free variables* ($\boldsymbol{F}$) of the experiment. The proper setting of this subset can provide the optimality of the estimation.

These facts imply that the optimal estimation procedure for parameters in $\boldsymbol{P}_2$ cannot be designed off-line and — the estimation-correction process is inherently cyclical: the TEM autotuning cannot be done in one step.

Practical Limitations, Secondary Effects

There are several factors which influence the image formation of the TEM to a considerable extent, but — because of their unmanageable nature — cannot be taken into consideration in the estimation procedure design. These factors include among others the mechanical instability of the TEM, the charging of the specimen and its environment, the hysteresis of the magnetic components of the TEM, the non-linear behaviour for extreme variations of TEM parameters.

The tuning steps significantly differ from each other considering the *type and amount of available information* (knowledge) and the way of using this information. As a consequence, the tuning process cannot be considered as a mere iterative one, the tuning algorithm applied in a certain tuning step is completely different from the algorithm applied in another step. The robustness of the autotuning claims for data/information processing procedures (DIP for short), which are able to cover the full scale of available relevant knowledge about the particular tuning situation and about TEM tuning in general. KB and other AI based symbolic data processing techniques have promising characteristics in this respect, but probably even the most sophisticated implementations of autotuning will not work properly without intensive 'manual' help in the starting tuning steps.

Software Architecture for Real-Time Autotuning

Motivations, the Higher Level Control Concept

As the previous section shows, the tuning of a TEM is a complex modelling/measuring process. A general modelling/measuring process can be divided into several subprocesses on functional basis (EYKHOFF, 1974) such as:

—collecting a priori knowledge,
—(preliminary) model building,
—experiment design,
—execution and data processing, and
—result/model evaluation.

In designing autonomous, robust measuring systems, the key issue is to provide sufficiently wide coverage of these subprocesses with computerized algorithms.

This requires extensive use of knowledge about the problem domain, specific areas of measurement theory and measurement technology. The central problem of realizing autonomous intelligent measuring systems is the formulation, efficient implementation and efficient use of knowledge/expertise required for solving the measuring problem in hand. Roughly speaking, the architectural complexity of the realization of a computerized measuring system to a great extent depends on what cannot be done off-line (i.e. cannot be 'hard-wired' into the program code) among the steps of the measuring process to be realized (SZTIPÁNOVITS, 1984; PAPP, 1990b).

The 'fragility' of a computerized measuring system is not independent of the facts mentioned above. Most of the problems originate from the not sufficient model evaluation and — partly connected with this — operating outside the validity of the model. Of course, instead of doing wrong actions it is much more acceptable — if the measuring system is not able to do the right things — to do nothing. This shows the importance of continuous model validation. In the traditional way of measuring systems design, this model — derived from the a priori knowledge — is implicitly coded into the program of the measuring procedure. This model formation process has its own problematics and is usually based on sophisticated mathematical methods (EYKHOFF, 1974). New kinds of problems arise, when the application of this refined, well developed formal approach becomes impossible (PAPP, 1990b).

On the other hand, concerning the measuring process in general, it should also be mentioned that clear separation exists between the operations *(actions)* and the knowledge of using these *(control)*. The former represents how to do things, the latter defines what to do. An autonomous control — in case of complex measuring problems — has to take into consideration the current state of the data processing (e.g. available partial results, results of model validity assessment) to achieve an acceptable level of liability and robustness. As a consequence, the transparent coupling between the data processing and control is of primary importance.

In respect of coping with the difficulties mentioned above the results of the research in certain specific fields of AI are promising. There is a common root for both solving the modelling and the control/action separation problem. Here the evolution of *symbolic data processing techniques* and the *knowledge centred approach* should be emphasized concerning our application domain[4].

Shortly summarizing the results of investigating the problems of autotuning it can be said that robust autotuning can only be realized by running properly chosen, properly parameterized DIP algorithms combined with continuous model validation and feedback (adaptivity). These activities are beyond the scope of responsibility of a particular data processing algorithm, these decisions and operations cannot be based on local information only, i.e. which is available for or generated by a single data processing element. The decision-making process has to take into consideration all the partial results coming from different data processing elements, has to integrate these results and, according to these, has to determine a suitable action providing system-wide optimality and robustness.

In our approach these tasks are covered by the higher level control (HLC) which is realized by a well separated system component, the higher level controller (HLCer). The HLCer is an intelligent *planner, executor* and *evaluator* for solving autotuning measuring/modelling problems: the HLCer supports the operator by asking questions for the sake of collecting all the relevant a priori knowledge, according to this automatically designs a (sub)optimal (possibly combined quantitative, qualitative and heuristic)

[4]Though the continuous referencing of the basic concepts is omitted, for readers who are not familiar with the AI related concepts some excellent survey literature is mentioned: NILSSON, 1980; BOBROW, 1984; ACM, 1985.

data processing scheme, executes the tuning step designed and depending on the outcome realizes feedback in the measuring process (by asking new questions, modifying the model, collecting new observations, giving advice, etc.), if it is necessary.

The Information Processing Scheme

From the information processing point of view the HLC is an intermediate layer between the TEM operator and the (traditional) DIP subsystem, the components of which are regularly based on predefined model structures. The lower level of the information processing, the DIP subsystem, is responsible for realizing the proper actions (i.e. DIP subsystem represents 'how to do things'), the upper level, realized by the HLCer, is responsible for the control (that is for 'what to do') with tight cooperation with the user. The 'usefulness' of the HLC for the autotuning system can be measured by the 'amount' of the responsibility taken over by the program from the user (in this respect the presence of the HLCer can really be interpreted as a way of realizing some kind of 'machine intelligence').

The information processing scheme of the complete TEM autotuning system is shown in *Fig. 2.* The measuring system can be divided functionally into two subsystems.

The HLCer, using different knowledge sources, attempts to realize a (sub)optimal measuring procedure by setting the parameters of, and controlling the DIP subsystem.

The other one, the DIP subsystem, consists of a set of well-defined data processing modules (DPs), each of which implements a basic building element for measuring procedures built up by the HLCer. A DPi receives input from the TEM (observations) or other DPs, and generates output (as well as status information) for the common data base and/or for other DPs. DPs can control the TEM if the data/image processing algorithm implemented requires active (excited) measurements. It should be emphasized that symbolic data processing algorithms can also be realized by DPs.

The only way for communication between the subsystems is reading and writing two common data bases (*Parameters & Control and Results & Status*). The synchronization of the subsystems is also realized via the data bases according to the data-driven and demand-driven control paradigms: the appearance of a data item or the request for a data item trigger the corresponding subsystem to process or to produce that item, respectively (see details later). In this approach the explicit programming of the situation dependent control of the cooperation between the subsystems can be completely omitted, the control automatically becomes adaptive.

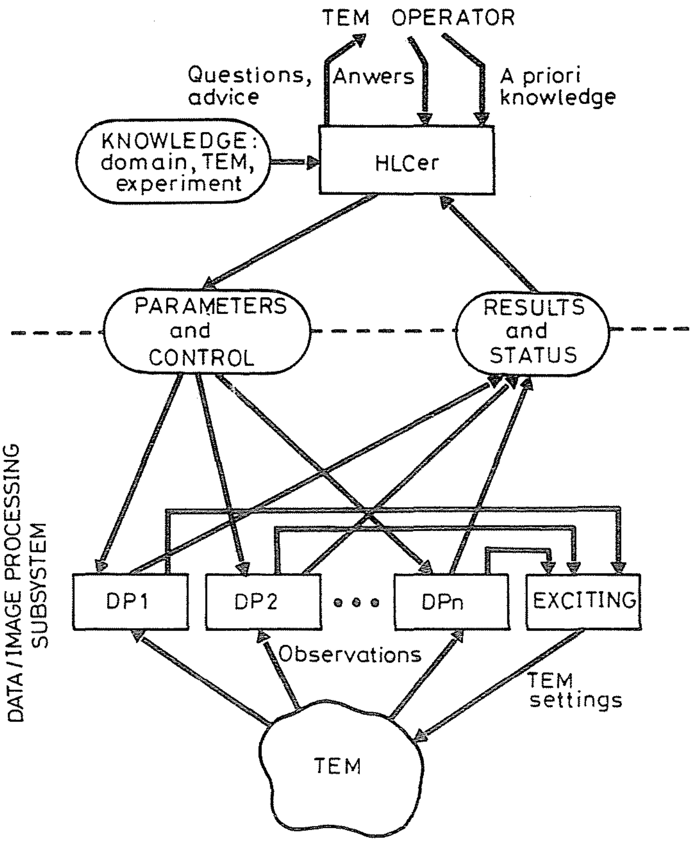

Fig. 2. The information processing scheme

The Software Architecture

The software architecture and the structure of its components to be presented are influenced by several AI related concepts and techniques, however, these concepts and techniques can hardly be identified clearly. In our approach the AI methods and AI programming paradigms were not considered as ready-made 'building blocks', but a way of coping with the task of structuring, describing and realizing complex systems (BOBROW, 1985). Thus unambiguous references to the methods applied cannot be given, some related ones are mentioned.

The software architecture of the autotuning system is surveyed in three parts according to the functional decomposition. The description is started with the structure of the common data base (CDB) which can be considered as a particular implementation of the blackboard communication paradigm (NII, 1986). Then this is followed by a deeper insight into the HLCer and the DIP subsystem and some remarks about the real-time operation close the section.

The Common Data Base

The function of the common data base (CDB) is to realize a communication platform between the HLCer and the DIP subsystem by

- — storing and providing access to the parameters of the DIP procedures, the control information for the DIP procedures and the status information from the DIP procedures,
- — providing synchronization between the activities of the HLCer and the DIP subsystem.

(The proposed measuring system software architecture contains two CDBs (Parameters & Control, Results & Status). All the statements below are relevant for both of the data bases.)

The former function is the common and well-known function of the data base managers and in our application it could easily be implemented by interface procedures acting on common memory array. The latter makes the implementation a little complicated. Having in mind the requirement of flexibility and the need for eventual modification, the structure shown in *Fig. 3* is proposed. The whole data base is a sequence of *data items* with some type of indexing mechanism provided for fast access (the figure shows only one data item in detail).

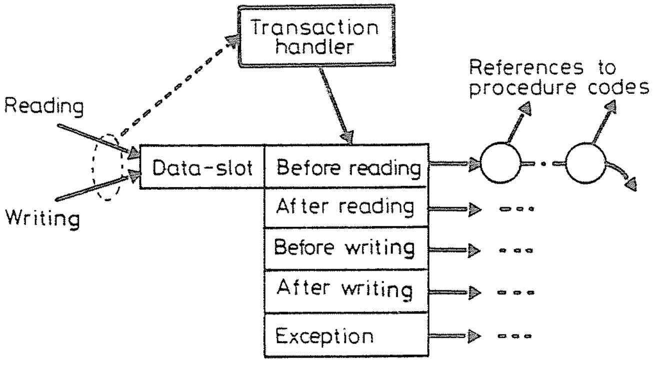

Fig. 3. The Common Data Base structure

The *data slot* is to store the information bound to the item. To each data slot a set of action slots is connected. These slots contain *activity descriptors* (typically a linked list of references to procedure codes), which are activated by the *transaction handler* according to the operation carried out (or intended to be carried out) on the data item.

In this scheme the data-driven and the demand-driven control paradigms can easily be implemented. In the data-driven case all the activities, which have to be carried out when a particular data item is delivered, have to be defined as a procedure and the references to these procedures have to be linked to the *AfterWriting* action list of this data item. After writing the new value of the data item into the data base, the procedures — realizing the operations to be accomplished — are automatically activated, so the first element of the data-driven chain is being started. Any of the procedures linked can execute data base operations, including writing, which can trigger the next element of the data-driven processing chain.

In a similar manner, in demand-driven cases the procedures linked into the *BeforeReading* action list define the way of acquiring the value of the related data item (i.e. the missing information). These activities can also be chained via data base elements.

The flexibility of this data base handling scheme is also shown by the fact that *exception handling, adaptive user communication* and even the *distributed data base handling* can easily be integrated. (The distributed handling becomes a crucial question if a two-computer hardware architecture is chosen for realization. This approach has several advantages, the current experimental set-up at the Delft University of Technology, Department of Applied Physics belongs to this category also.)

The Higher Level Controller

The structure of the HLCer subsystem is shown in *Fig. 4*. The key elements of the HLCer are the *knowledge interpreters*. Each of them realizes the processing of a certain type of knowledge representation formalism (i.e. realizes a measuring subprocedure based on a given type of system model). The situation independent ('general') knowledge is stored in the related knowledge base (knowledge source, KSi, in the figure). The situation specific knowledge for each knowledge interpreter is a subset of the CDBs. These subsets are not to be disjunct, so in this way communication among the KB subsystems can be established. The control of the cooperation can be defined by means of activity descriptors (see previous section). It should be emphasized that the system architecture is symmetric, that is the KB subsystems and the DIP subsystem are managed in a unified way. There

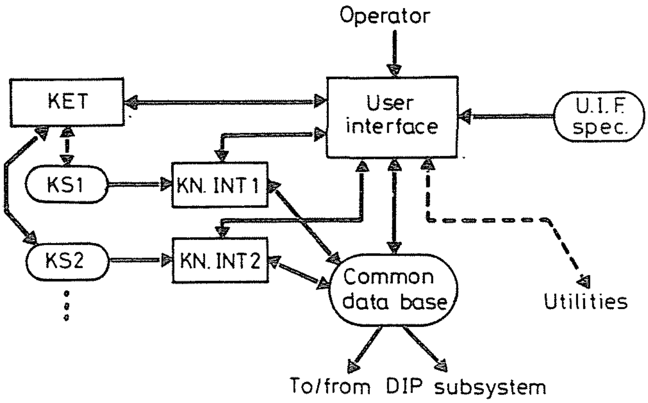

Fig. 4. The Higher Level Controller

is no difference between the KB subsystems and the DIP subsystem with respect to the cooperation and the relation to other system components.

The knowledge engineering tool (KET) is an optional subsystem of the HLCer. The KET supports the interactive knowledge base modification and development. Besides the dedicated knowledge base editors, KET can include (and usually does) special knowledge base compilers to speed up the knowledge interpretation (the running of measuring subprocedures) by means of knowledge base preprocessing (FORGY, 1982; PAPP, 1989).

In this architecture several types of HLCer can easily be implemented — depending on the results of further investigations — without any modification of the DIP subsystem. At one of the extrema, the HLCer realizes only a 'shiny' user interface (e.g. with built-in context sensitive help) — in this case the built-in knowledge is negligible. Going to the other end — where the idealistic completely autonomous and robust solution exists — the HLCer contains and applies more and more relevant knowledge (e.g. KB expert system, artificial neural network with learning facility, etc.). This incremental development matches very well the proposed HLCer architecture, which is important for coping with hard-to-formalize, continuously refining specification.

In the introductory phase of the implementation presumably it is sufficient to realize a pattern directed inference system with integrated forward/backward control and efficient truth (reason) maintenance mechanism (MCALLESTER, 1980). This symbolic data processing scheme is flexible enough to support the description and running both the heuristic and the qualitative model based measuring subsystems on a considerably high level, with acceptable compromise (NII, 1978).

The Data/Image Processing Subsystem

The DIP subsystem consists of functionally independent data processing elements (DPE) and interface components as *Fig. 5* shows. A DPE realizes a building block for the DIP algorithms to be applied and based on structurally determined (specimen, TEM, observation channel) models.

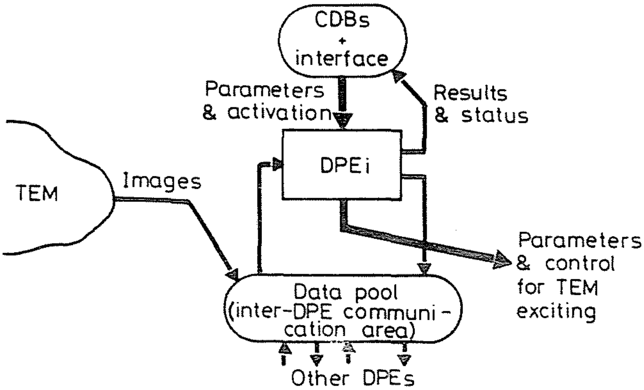

Fig. 5. An element of the DIP subsystem

Complete data processing algorithms can be composed as running a sequence of parameterized DPEs. The parameters arrive from, and the results are stored in the CDBs providing the communication with the HLCer. The *Data Pool* serves as an inter-DPE communication area. For the sake of flexibility standardized data representation for DPE input/output has to be developed and applied allowing a formally unlimited combination of DPEs.

The Experimental Set-up

The computational requirement of the autotuning algorithms even on the DIP subsystem level is demanding (e.g. the image shift induced is measured by cross-correlation technique (using floating point two-dimensional FFT to achieve the sufficient precision), for low-dose and high resolution experiments several types of low/high pass filtering are necessary). To provide acceptable reaction time, the need for sophisticated hardware background is unambiguous.

In the previous phase of the research which mostly concentrated on the theoretical foundation of TEM autotuning and on data/image processing algorithms development and testing, a TIETZ VIDEO Digital Image Processing System (Tietz) was used. This is a powerful VME bus based computer with Motorola 68020 processor, 5 Mbytes main memory and 250 Mbytes hard disk. For the sake of supporting the TEM autotuning application, the system is equipped with a frame grabber, image ALU, dedicated FFT processor, array processor and digital-analogue converters (for exciting the magnetic lenses of the TEM). Using this hardware installation, a tuning step is performed within a few seconds. The C, Pascal and assembly language support with development environments run under OS/9 multi-tasking operating system. A special purpose library and shell for elementary image processing operations (e.g. image averaging, diffractogram calculation, contrast enhancement) is also available.

This image processing system is proved to have sufficient computational power, thus it forms a reasonable basis for further development. Because of software engineering considerations (for details see PAPP, 1990a), a two-computer loosely coupled hardware architecture serves as a basis for the robust autotuning development. In this architecture the HLCer is allocated on the IBM PC, and consequently distributed CDBs have to be realized. The implementation of the HLCer started with a dialect of LISP under MS-DOS operating system.

Conclusions

In the paper a software architecture for a TEM autotuning system was shown. Since the implementation of the robust autotuning software is just started, experimental results verified in practice cannot be presented, but some more general statements can be made and should be emphasized.

Investigating the reasons of the fragility of experimental autotuning softwares it became clear that the continuous model evaluation and the adaptivity depending on the outcome of this evaluation are the key issues. The accurate and dose effective tuning requires the application of complex, sophisticated specimen and TEM imaging models. This modelling cannot be solved in general, models can be derived for particular specimen classes and TEM imaging modes, moreover these are simplified models and their validity is strongly limited in case of large scale detuning. In certain set-up analytical models are not available at all, due to theoretical problems and/or resource constraints. In addition, optimal TEM autotuning procedures — based on these models — cannot be designed off-line.

With respect to coping with the difficulties mentioned above, the results in certain specific fields of AI research are promising. Our investigations showed that the most convertible results are on the 'deep level' of AI research, such as symbolic data processing, knowledge centred approach, heuristic algorithms, declarative programming.

The autotuning software can embed both quantitative, qualitative and heuristic data processing algorithms, and provides well-defined, transparent coupling among them. A valuable feature of the architecture presented is that it gives a framework for automating the control and cooperation of the different styles of data processing, and in this way makes the program development (both in algorithmic and methodological sense) easier to a considerable extent. The software architecture also shows that carefully designed architecture can embed AI related components even in real-time application domain in a hardware independent way.

Though the hardware background for real-time autotuning has a heavy price tag today, the new generation of TEMs, which inherently contains image processing facilities, gives importance to the development and provides solid basis for commercial systems.

Acknowledgements

Thanks are due to Prof. P. Kruit, Dr. ir. A. J. Koster for their valuable comments and suggestions on the improvement of the preliminary version of the technical report referred to and A. Bruin for his clear explanations and demonstrations on transmission electron microscopy. The long term and intensive research on intelligent instrumentation was carried out with the REALEX Group of the Department of Measurement and Instrument Engineering, Technical University of Budapest, namely Dr. T. Dobrowiecki, B. Vadász, Dr. K. Tilly, B. Pataki and I. Kerese.

References

ACM (1985): Special Issue on Architectures for Knowledge Based Systems. *Communication of the ACM*, Vol. 28, No. 9, September 1985.

BOBROW, D. G. (ed) (1984): Special Volume on Qualitative Reasoning about Physical Systems. *Artificial Intelligence*, Vol. 24, No. 1–3, December 1984.

BOBROW, D. G. (1985): If the Prolog is the Answer, What is the Question? Or What it Takes to Support AI Programming Paradigms? *IEEE Trans. on Software Engineering*, Vol. SE–11, No. 11, pp. 1401–1408, November 1985.

EYKHOFF, P. (1974): System Identification; Parameter and State Estimation. John Wiley & Sons, 1974.

FORGY, C. L. (1982): A Fast Algorithm for Many Pattern / Many Object Pattern Matching Problem. *Artificial Intelligence*, Vol. 22, pp. 17–37, 1982.

KOSTER, A. J. (1989): Autotuning of a Transmission Electron Microscope. PhD Thesis, 1989, Dept. of Applied Physics, Delft University of Technology, The Netherlands.

McALLESTER, D. M. (1980): An Outlook on Truth Maintenance. MIT AI Lab Memo 551, 1980.

NII, H. P. – FEIGENBAUM, E. A. (1978): Rule-Based Understanding of Signals. In D.A. Waterman, F. Hayes-Roth (eds): Pattern Directed Inference Systems. Academic Press, 1978. pp. 483–501.

NII, H. P. (1986): Blackboard Systems. *AI Magazine*, pp. 1–13, Spring 1986.

NILSSON, N. J. (1980): Principles of Artificial Intelligence. Tioga Publ. Co., 1980.

PAPP, Z. – DOBROWIECKI, T. – VADÁSZ, B. – TILLY, K. (1989): Expert System Architecture for Real-Time Process Supervisor Applications. In G. Rzevski (ed): Artificial Intelligence in Manufacturing. Springer-Verlag, 1989. pp. 223–240.

PAPP, Z. (1990a): Implementation of Robust TEM Autotuning. Report 9001, Delft University of Technology, Dept. of Applied Physics, Physics Informatics Research Group.

PAPP, Z. (1990b): Programming Tool for Integrating Numerical and Knowledge Based Signal Processing Techniques. *Proc. of the International IMEKO Symposium on Knowledge Based Measurement*, Karlsruhe, FRG, pp. 181–188, September 1990.

REINER, L. (1984): Transmission Electron Microscopy. Springer-Verlag, 1984.

RUIJTER, W. J. DE (1988): Autotuning a Transmission Electron Microscope for Low Dose and High Resolution Applications. Master's Thesis, 1988, Dept. of Applied Physics, Delft University of Technology, The Netherlands.

SZTIPÁNOVITS, J. – BOURNE, J. (1984): Design of Intelligent Instrumentation. *Proc. of the 1st IEEE Conf. on Artificial Intelligence Applications*, 1984, Denver, USA, pp. 490–495.

TIETZ Video and Image Processing Systems (product descriptions). Tietz Video GmbH, Herbstrasse 7, D–8035 Gauting, FRG

Address:

Dr. Zoltán PAPP
Department of Applied Plysics
Delft University of Technology
NL-2600 Ga Delft, P.O.Box 5046
The Netherlands