On the Automation of the MarkIII Data Analysis System

W. Schwegmann and H. Schuh
Deutsches Geodätisches Forschungsinstitut (DGFI)
Marstallplatz 8
D-80539 München
Germany
Tel.:+49/89-23031-202, Fax: +49/89-23031-240
e-mail: schwegmann@dgfi.badw.de

Abstract: A faster and semiautomatic data analysis is an important contribution to the acceleration of the VLBI procedure. A concept for the automation of one of the most widely used VLBI software packages the MarkIII Data Analysis System was developed. Then, the program PWXCB, which extracts weather and cable calibration data from the station log-files, was automated supplementing the existing Fortran77 program-code. The new program XLOG and its results will be presented. Most of the tasks in the VLBI data analysis are very complex and their automation requires typical knowledge-based techniques. Thus, a knowledge-based system (KBS) for support and guidance of the analyst is being developed using the AI-workbench BABYLON, which is based on methods of artificial intelligence (AI). The advantages of a KBS for the MarkIII Data Analysis System and the required steps to build a KBS will be demonstrated. Examples about the current status of the project will be given, too.

1 Introduction

Based on the inertial reference system of extragalactic radiosources VLBI is an extremely precise geodetic technique for monitoring the Earth rotation and for the realization of the global reference system. However, the costs are relatively high and there is still a delay of five days minimum between the time of observation and the availability of the results. If this gap could be shortened considerably this would allow to monitor the Earth rotation parameters (ERPs) and the relative station motions in near real-time. This could be extremely useful for high precision positioning and navigation. Results in near real-time could also be used as references for other 'on-line' observing methods, e.g. laser gyros or supra-fluid helium gyros, as well as for observing transient astronomical phenomena.

The paper presents a concept for the automation of the MarkIII Data Analysis System, a widely used VLBI software system (Ryan et al., 1980). This concept has been developed at DGFI, Munich, within a research project called 'Applications of Methods of AI for the VLBI Data Analysis'. The goal of the project is to accelerate and automate the VLBI data analysis. It belongs to a general proposal for VLBI in near real-time that was presented by Schuh (1995), where the means for automation and acceleration of the VLBI procedure are described for all steps of the VLBI procedure, i.e. scheduling, observation, correlation and data analysis. When looking at the work done by the analyst during the course of VLBI data analysis, it can be seen that almost all tasks can be supported by typical AI methods, as proposed by Schuh (1991, 1993). Some tasks can be realized by additional algorithms or crosschecks conventionally programmed within the existing software. This paper gives examples for both, the automation with conventional methods and with methods of AI, and shows the advantages of knowledge-based techniques.

A first successful realization of an on-line VLBI-System is the Japanese Keystone project. In this project four radiotelescopes in the Tokyo area produce geodetic results in near real-time. It is part of the Japanese Earthquake Prediction Programme (Yoshino, 1994, Koyama et al., 1998).

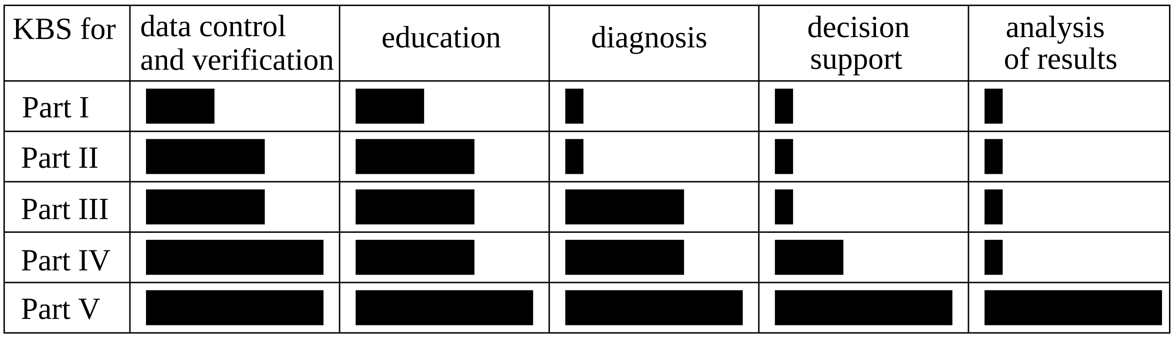
2 Reasons for the development of a Knowledge-Based System

In the next few years the number of geodetic VLBI observing sessions will increase considerably. The degree of automation within the present MarkIII Data Analysis System is very low. Most steps during the data analysis have to be carried out manually and the data analysis is very complex and often requires the analyst's (expert's) decisions. Already today there is a shortage of qualified experts and the analyst's training takes a long time. Moreover, even experienced analysts make time-consuming errors. Wrong results could lead to misinterpretations and to incorrect conclusions. This leads to the necessity of a faster and even semiautomated data flow during the data analysis, what can only marginally be achieved by faster computers. Thus, the goal should be to make the VLBI data analysis faster and safer by ensuring that less failures and less errors are made even by less experienced analysts. According to the description of a knowledge-based system (KBS) (Sundermeyer, 1991) a KBS can be used to model the knowledge needed for the VLBI data analysis and to use this knowledge for solving the complex problems within the data analysis automatically. A KBS is less susceptible to errors, allows to conserve the analyst's knowledge and to check his decisions. More information about KBS can be found in Schnupp et al. (1989).

3 Automation of VLBI data analysis

To examine the possibilities for the automation of the different steps within the data analysis, the data flow inside the MarkIII Data Analysis System was analysed. Especially the applicability of knowledge-based techniques was investigated but it was also considered, to automate some steps with conventional programming techniques by supplementing the existing program-code. Therefore the data flow inside the MarkIII Data Analysis System was subdivided into five parts each related to a superior task: correlator output, a-priori information, generation and modification of databases, log-file processing and data analysis (see Fig. 1).

The demands on a software which should automate the data analysis differ for each of the parts. Table 1 shows to which degree a KBS would be useful in the different parts of the data analysis. A KBS can be used as a teaching system and for special aspects of the data processing. The main application, however, will be for decision-support, diagnosis and analysis of results in *Part V*.



Tab. 1: Possibilities for the application of a KBS in the VLBI data analysis.

After having thoroughly analysed the data flow inside the MarkIII Data Analysis System the program PWXCB in *Part IV* was automated. This could be done so far with conventional programming techniques that is by supplementing the existing program-code written in Fortran77, because the knowledge needed for these tasks is relatively simple. The new program is called XLOG and is described in the next section.

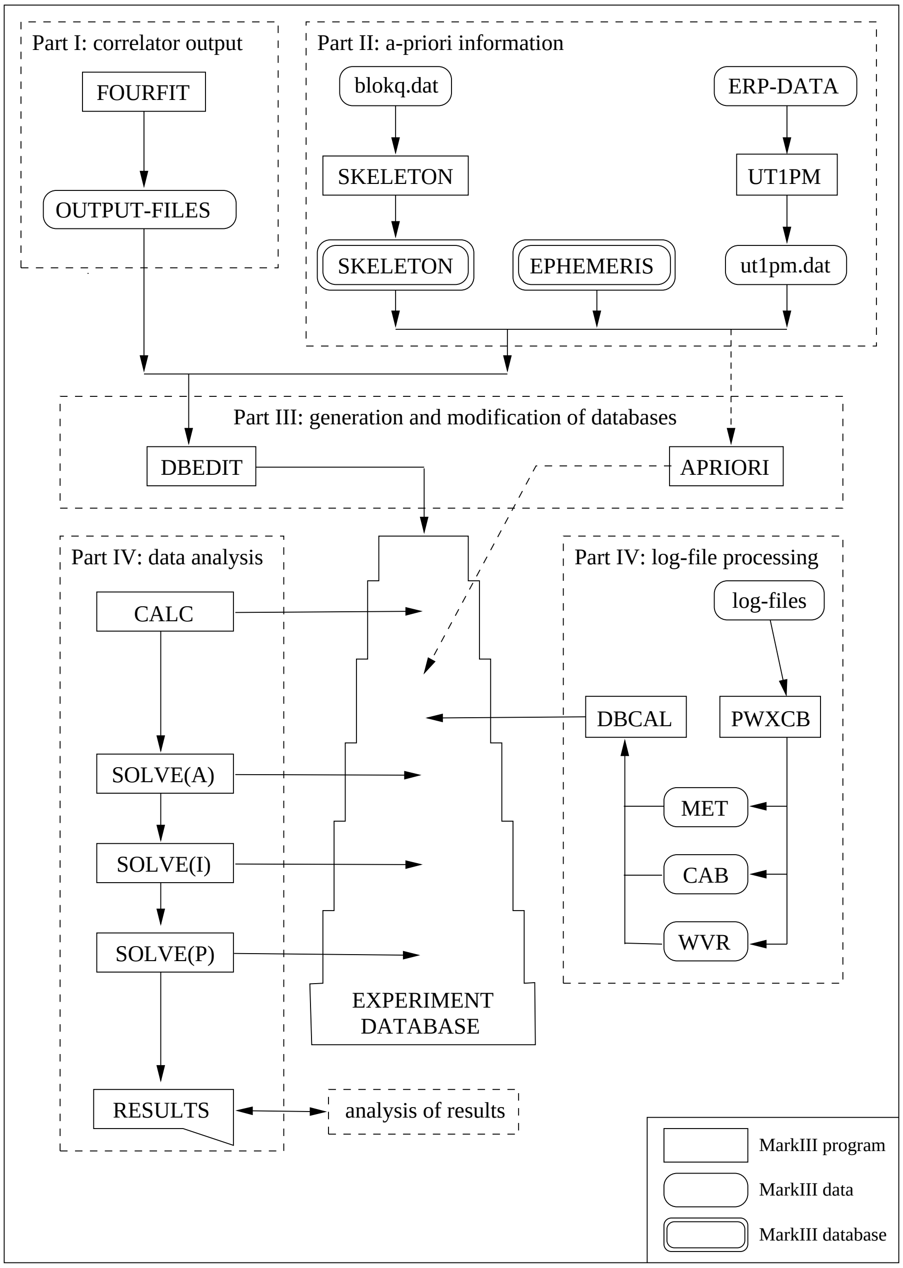


Fig. 1: Flow chart of the standard MkIII Data Analysis System.

3.1 Automatic processing of station log-files

During the analysis of VLBI experiments the meteorological data and the cable calibration data have to be read from the station log-files and are entered into the experiment database. For that purpose the program PWXCB extracts the meteorological and cable calibration measurements from the station log-files and stores the data in separated files, the so called 'calibration files' (see Fig. 1, *Part IV*). The data from these files are entered into the database by the program DBCAL. Creating the calibration files can be very time-consuming because of wrong entries in the log-files which have to be corrected manually. In addition, the extracted data have to be checked with respect to their plausibility. These tasks shall be performed now by an automated version of PWXCB called XLOG. All possible errors had to be considered when developing the software, even if they appear quite implausible, to obtain a high reliability of the automation. XLOG considers offsets, gaps and single outliers in the data, too few data, wrong entries in the log-files (e.g. missing keywords, mixing up temperature and pressure, etc.) and obvious met sensor problems (e.g. constant temperatures entered in a log-file which is an indication for a sensor failure).

The program can be started by several user-modes: In the semi-automatic mode the analyst has to confirm all actions done by the program, i.e. original and edited data are plotted by XLOG and the analyst is asked to decide which data should be used or if he wants to edit the data manually with PWXCB. In the fully automatic mode the analyst is asked for a confirmation only if there are problems with the automatic processing of the data, e.g. if suspicious data were detected but not deleted, if more than 10 % of all data points were deleted, etc. In a third user-mode the program runs without user interaction and no messages are printed on the screen. In this case the analyst is able to check the results by looking at the output files of XLOG which contain information about the automatic processing of the log-files. More details about the user-modes can be found in the users-guide of XLOG. The data flow inside XLOG is shown in Fig. 2. The program starts reading the control-file which contains all necessary information to process the log-files automatically, e.g. user-mode, input-filelist, experiment descriptor, etc. Then the program runs through the following steps for every input-file. First the program identifies the type of the input file (MarkIII-, MarkIV-log-file or an existing calibration file). The data extracted from the input-file will be checked with respect to their plausibility and will be saved if they are o.k. and the next file from the input-filelist is going to be processed. If there are problems during the automatic processing or if XLOG runs in semi-automatic mode, warning messages are sent to the analyst and the original and edited data are plotted. Then the analyst is able to check the plausibility with stronger or weaker conditions to find wrong data. If the results are still not o.k., the analysts is able to decide that the original data should be used and be processed with PWXCB. Finally XLOG creates the control-file for the program DBCAL.

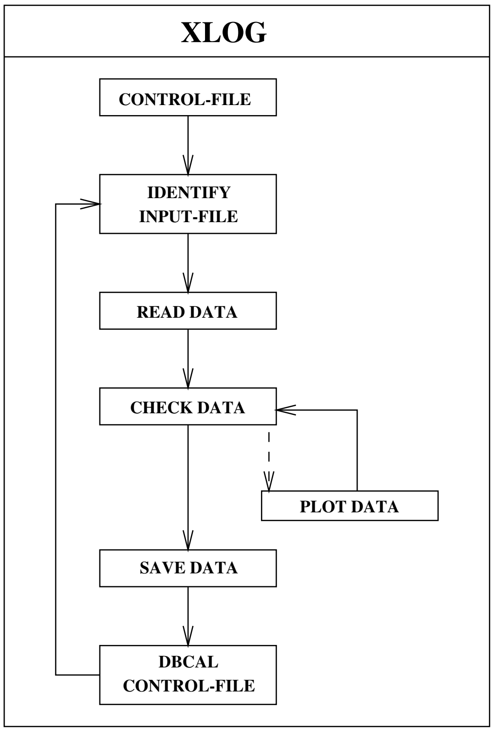


Fig. 2: Data flow of the program XLOG.

The automatic processing of the log-files was checked on real data. Thereto about 200 station log-files which had been already processed manually were reprocessed with XLOG. Almost 25 percent of the 200 log-files have had wrong entries and had to be edited manually. The comparison of the results showed a success rate of better than 96 percent of the automated version compared to manual editing which means that basically the same results were achieved. For four percent of all cases the log-files were not processed correctly by XLOG, but in all cases the program wrote warning messages into the output files and on the screen. An example for successful automatic editing of cable calibration data is shown in Fig. 3. The cable sign was detected by XLOG and written into the cable calibration file. The offset within the data (see left part of Fig. 3) was deleted, too. To demonstrate which errors could be detected by XLOG Fig. 4 shows the automatic editing of intentionally falsified meteorological data (see left part of Fig. 4) by the present version of the program. All obvious errors were detected by XLOG. Merely at the end of the experiment there is a problem when no data are available for three hours which reveals some weaknesses of the conventionally programmed version of XLOG. Hence there is a difference of almost eight degrees centigrade between the last data point before the gap and the first point after. At present XLOG cannot decide if this is a measurement error or if the data after the gap are correct because this requires the knowledge and the experience of the analyst about the usual weather conditions at the station. In such case, the analyst will be warned by XLOG and he will edit the data manually. Anyway, using the current version of XLOG a considerable amount of time and effort can be gained. At present, a beta-version of XLOG is being tested for routine application. It is planned to release version 1.0 in Spring 1999.

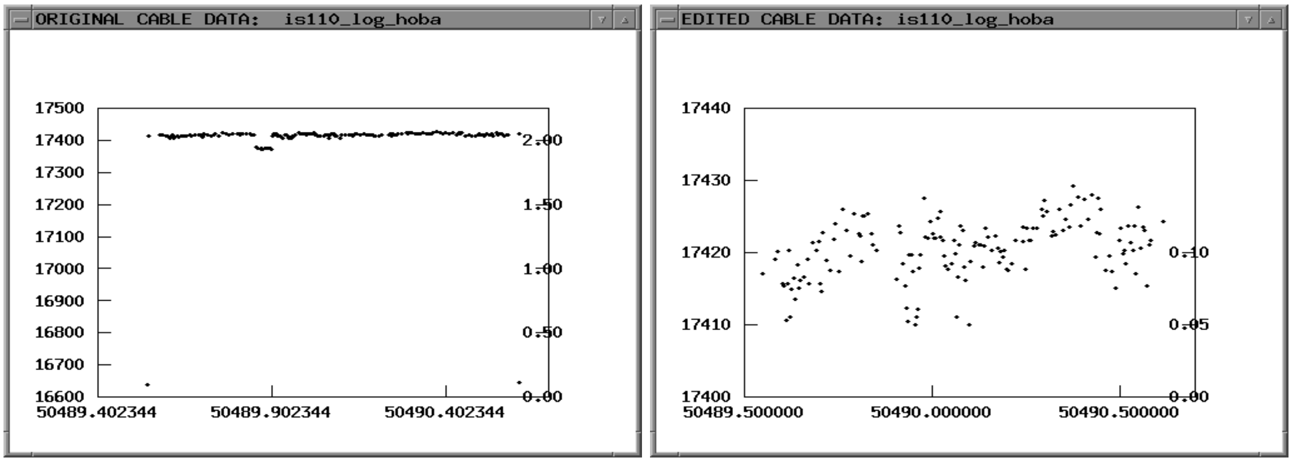


Fig. 3: Comparison of original cable measurements at a VLBI station from the log-file (left) and the data after automatic processing with XLOG *(right).*

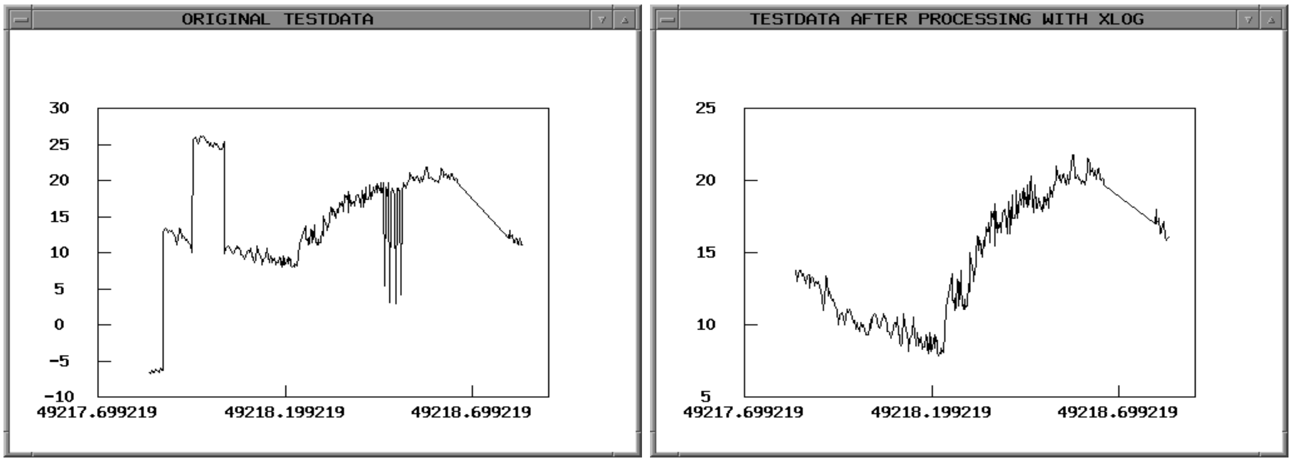


Fig. 4: Comparison of intentionally falsified temperature measurements at a VLBI station (degrees centigrade) from the log-file (left) and the data after automatic processing with XLOG *(right).*

3.2 A knowledge-based system for the VLBI data analysis

3.2.1 General remarks

The KBS for the automation of VLBI data analysis with the MarkIII Data Analysis System will be developed to support the analyst in an implicit and explicit manner. Implicit means that all steps done by the analyst are checked by the system in the background. The analyst only notes these checks, if the system sends a warning message. Explicit means that the analyst is able to consult the system anytime to solve a problem.

The development of a KBS can be supported efficiently by AI software tools. Thus the AI-workbench BABYLON (Christaller et al., 1992) is going to be used for our purpose. BABYLON contains formalisms for knowledge representation, a problem solving component to process the knowledge and an explanation component to tell the user how the system arrived at a solution. BABYLON is implemented in LISP, the most common AI programming language. It was developed by the 'Gesellschaft für Mathematik und Datenverarbeitung (GMD)' in Sankt Augustin, Germany. To construct a KBS using BABYLON, the knowledge needed for the problem must first be collected, structured and organized (knowledge acquisition). This knowledge has to be formally represented and stored in the knowledge base (knowledge representation). The efficiency of a KBS depends on the quality of the knowledge within the knowledge base, which should be modular, flexible and extensible. Because the general procedure of VLBI data analysis is very similiar in all VLBI analysis programmes, the system is being designed to be transferable to other VLBI software packages like Occam (developed at the Universities of Bonn, Madrid and Saint Petersburg) or MASTERFIT/MODEST (developed at Jet Propulsion Laboratory, Pasadena, CA). Therefore the knowledge base will be built to a high degree independently from the MarkIII Data Analysis System. A system interface will be needed to acquire data from the MarkIII Data Analysis System and to control the data analysis with the KBS.

3.2.2 Automation of the standard VLBI data analysis

At present the KBS is going to be developed for the automation of a single VLBI experiment according to the standard procedure applied for analysing geodetic VLBI data (Nothnagel, 1995). The system should automate the steps of *Part V* (see Fig. 1) of the VLBI data analysis:

1. Group delay ambiguity shifting
2. Ionosphere calibration
3. Update of the data base
4. Final parameterization

Every step contains several subsequent tasks, which are sometimes trivial from the point of view of an expert but have to be considered by the KBS. The main problem when automating these tasks is the representation of the knowledge, because the methods used by the problem solving component are in principle the same for both, trivial and complex tasks. A trivial task is for example the correct parameter setting of the clock parameters during the group delay ambiguity shifting. The following examples are given to show how the different knowledge representation formalisms of BABYLON can be used to represent the knowledge about this task. The knowledge needed for this task according to Nothnagel (1995) can be summarized by the following rules:

- One station of a VLBI experiment has to be used as reference station for which no clock parameters will be estimated.
- The reference station should be in the central area of the network.
- The reference station should not be known for strong clock variations.
- For ambiguity shifting only the first two clock parameters (clock offset and rate) are estimated at each station except the reference station.

To represent this knowledge and to define the steps to control the correct setting of the clock parameters the following formalisms are used:

- Frames to represent the stations and their attributes (e.g. name of a station, characterization of the clock (*bad* if it is known for strong clock variations otherwise it is *good*), etc.).
- Behaviours to change the attributes of a station, for example to set the clock parameters to [0 0], which means that the station is going to be used as reference station:

  ```
  (defbehaviour (vlbi-station :set-clock-reference) ()
    ← :put 'clock-para [0 0])
  ```

- Prolog to analyse the situation, for example to check if by mistake a station is set as reference station. The result of the 'Prolog clause' `is-wrong-reference-clock` is *TRUE* if the clock is described as *bad* and if the clock parameters at the station are set to [0 0]:

  ```
  ((is-wrong-reference-clock _station)
    ←
    (is [0 0] (← _station :get 'clock-para))
    (is bad (← _station :get 'clock)))
  ```

- Rule-sets to check the clock parameter settings and to control the analysis, for example to check if by mistake more than one station was set as reference station:

  ```
  (defrule-set :test-multiple-reference-clock
     (rule-1 ($and (is-reference-clock _station1)
                   (is-reference-clock _station2))
             ($execute
               (SAY ''There are multiple defined clock references'')
               (← _station1 :set-clock-2p)
               (← _station2 :set-clock-2p)
               (find-implications :define-clock-reference :do-all))))
  ```

  The rule `test-multiple-reference-clock` uses the behaviour `is-reference-clock` to check if a specific station is set as reference. If more than one station is set as reference a message is plotted to the screen, the behaviour `set-clock-2p` is used to set the clock parameters of all stations to [1 1] and the rule-set `define-clock-reference` is started to define a reference station.

The knowledge is stored in two knowledge bases. One contains all knowledge with respect to the stations and the other contains the knowledge for the clock parameter settings. This is necessary because the informations about the stations are going to be used in other steps of the data analysis, too. Thus the modularity of the knowledge bases is guaranteed.

To check the clock parameter settings the system performs the following steps:

1. check if more than one station is defined as reference, if yes:
   (a) set the clock parameters of all stations to [1 1]
   (b) define one reference station
2. check if the reference station can be used as clock reference, if not:
   (a) set the clock parameters to [1 1]
   (b) find a correct reference station

To automate the tasks of the VLBI data analysis described above for each step several knowledge bases have to be created. These knowledge bases contain the knowledge which is necessary to solve the specific problem and can be merged to automate the data analysis of single VLBI experiments.

4 Conclusions

To automate the VLBI data analysis either conventional programming techniques or methods of artificial intelligence can be used. The automation of *Part IV* of the MarkIII Data Analysis System by supplementing the existing Fortran77 program-code was successful but has also shown the weaknesses of conventional methods. Knowledge-based techniques are much more practical for the automation of the VLBI data analysis, because they allow an explicit modelling of the multifaceted knowledge needed for these tasks. At present the KBS is going to be built for the most complex and time consuming step in the VLBI data analysis: the selection of parameters inside *Part V* in Fig. 1.

5 Acknowledgements

The research project 'Applications of Methods of AI for the VLBI Data Analysis' is supported by the Deutsche Forschungsgemeinschaft (DFG), project Schu 1103/2-1.

References

Christaller, T., di Primio, F., Schnepf, U. and Voss, A., eds., The AI Workbench Babylon, *Knowledge-Based System Book Series*, Academic Press, London/New York, 1992.

Koyama, Y., Kurihara, N., Kondo, T., Sekido, M., Takahashi, Y., Kiuchi, H. and Heki, K., Automated Geodetic Very Long Baseline Interferometry Observation and Data Analysis System, Submitted to *Earth, Planet and Space*, 1998.

Nothnagel, A., VLBI Data Analysis User's Guide, Geodetic Institute of the University of Bonn (GIUB), Internal Report, 1995.

Ryan, J. W., Ma, C. and Vandenberg, N. R., The Mark III VLBI Data Analysis System, NASA Goddard Space Flight Center, publication X-945-80-25, 1980.

Schnupp, P., Huu, C. T. N. and Bernhard, L. W., *Expert Systems Lab Course*, Springer-Verlag, Berlin/Heidelberg, 1989.

Schuh, H., Design of an Expert System for VLBI Data Analysis, *Proceedings of the 8th Working Meeting on European VLBI for Geodesy and Astrometry, Dwingeloo, June 1991*, Report MDTNO-R-9243, Survey Department of Rijkswaterstaat, Delft, The Netherlands, pp. IV53–IV58, 1991.

Schuh, H., An Expert System for the Selection of Parameters in Solve, *Proceedings of the 9th Working Meeting on European VLBI for Geodesy and Astrometry, Sept. 1993*, Mitteil. a. d. Geodätischen Instituten der Rhein. Friedr. Wilh. Univ. Bonn, No. 81, pp. 79–86, 1993.

Schuh, H., A concept for real-time VLBI, *Proceedings of the 10th Working Meeting on European VLBI for Geodesy and Astrometry, May 1995*, edited by R. Lanotte and G. Bianco, agenzia speziale italiano, pp. 51–56, 1995.

Sundermeyer, K., Knowledge-Based Systems: terminology and references, BI-Wissenschaftsverlag, Mannheim/Wien/Zürich, 1991.

Yoshino, T., Crustal Deformation Monitoring in Tokyo Metropolitan Area by Space Geodetic Observation, *Journal of the Communication Research Laboratory*, *41*(3), pp. 1–9, 1994.