GeoTeX: A GENERAL POINT DETERMINATION SYSTEM

I.Colomina, J.Navarro and A.Térmens
Institut Cartogràfic de Catalunya
Spain

ABSTRACT:

The GeoTeX system of the *Institut Cartogràfic de Catalunya* (ICC) is a general geodetic and photogrammetric point determination system which is able to deal with any type of geometric functional model. GeoTeX is suited for research as well as for production purposes and can be easily extended to incorporate new models. Examples of tasks that the system can master are: spatial triangulation —SPOT images—, aerial triangulation with kinematic GPS derived positions, conventional geodetic network adjustment, DTM surface information and combinations thereof. In the paper, two components of the system are described: the discrete model kernel based on discrete mathematical techniques and the I/O kernel based on a general formal geodetic model and its associated formal grammar based interface language.

KEY WORDS: abstract data types, combined adjustment, compiler-compilers, data standards, grammars, network discrete models.

1 INTRODUCTION

In a sense, network point determination is an old-fashioned topic in photogrammetry and geodesy. It can be stated at least that the subject is no longer appealing to the academic research community.

On the other hand, fast, accurate and reliable point determination is still a must for private companies and official agencies in the allied fields of geodetic surveying, photogrammetry and remote sensing. These organizations face a variety of problems ranging from huge data sets, to changing hardware/software environments, to the extension of the traditionally supported observable types (for instance the testing of new mathematical models or the use of new sensor types). As a consequence, remarkable efforts are still being made mainly by instrument/software manufacturers and by research/development departments in land surveying agencies in order to improve and keep the software systems for point determination up to date.

Two additional introductory remarks are in order here. First of all, the principles and concepts established by the research activities related to the former ISPRS Working Group III.1 (period: 1984-1988; subject: accuracy aspects of combined point determination) have not been transferred to practice with the exception of industrial photogrammetry and GPS aided aerial triangulation. Secondly, advantage of already existing progress in computer hardware and software must not only be taken to speed up the old software but also to develop superior methods and procedures.

At the ICC, besides the above general remarks, there are particular motivations behind the research and developments reported in this paper. A unique point determination system both suited for research and production was needed since the development and maintenance of two different systems cannot be afforded. Similarly, it is not cost feasible to develop and maintain a system for geodetic networks, a system for aerial triangulation and a system for spatial triangulation.

The authors claim that this kind of a contradiction between research/production software and tailored/general software can be overcome if proper mathematical and computer science tools are used. Another claim is that the benefits of this global approach are not solely of interest to small groups. They are: a cost reduction in software development, acquisition and maintenance; a closer collaboration between teams traditionally involved —as well as traditionally separated— in point determination tasks; and the introduction of factors of rationality and coherence in the corresponding point determination projects.

2 ON GENERALITY, ADAPTABILITY AND PORTABILITY

In the context of point determination systems, generality is the power of mastering a broad range of [network] observations, conditions and parameters; adaptability is the power to incorporate new ones; and portability, as everywhere else, is the power to run on different platforms with minor or no changes,

To say that generality, adaptability and portability are desirable properties of a software system is common place. Rather than restating this, this section aims at describing a specific example of how the pursuing of those properties has lead to certain design concepts.

2.1 Generality

Usually, in a same organization, there are several different groups involved in point determination tasks; namely, the geodesists, the photogrammetrists, the remote sensing specialists and the surveyors. There is no difference whether referring to *ppm*, *% of flying height*, μm, or *pixel*; they all perform similar computations to estimate position, orientation and other parameters. For each group, there is usually a different software system as well. This redundancy, besides being an increase in software costs, implies some additional complexity in data transfers and communication between processes. The situation is even in contradiction with the current evolution of technology towards more heterogeneous systems and procedures. Examples are the GPS supported aerial triangulation, the convergence of digital photogrammetry and remote sensing, and the democratization of geodetic surveying with the advent of GPS.

Since the statistical, functional and structural abstract concepts are the same for any set of observations —indeed, an elementary statement from adjustment theory—, the natural solution to the above mentioned problems is the development of general point determination systems. These systems are built around general network adjustment programs.[1] Credit for a remarkable and early realization of this idea goes to A.A.Elassal and his GALS software system [8]. At the ICC the idea has been realized through the development of the GeoTeX system, whose heart is the general network adjustment program ACX (see [4] for a first and short description and application).

The above discussion related to generality can be summarized in two items: a design concept and a realistic goal set to be achieved in the particular implementation of GeoTeX/ACX.

Design concept: definitions of observable, condition and parameter polymorphic abstract data type must be available; any particular abstract type of observable, condition and parameter must fit the former definition.

Realistic goal: that the system and the adjustment program be able to handle any combination of observations of the geodetic, photogrammetric, SPOT and DTM-surface type (in a 3 dimensional space).

2.2 Adaptability

Some commercial and operational software packages exhibit certain difficulties in keeping pace with technological and scientific progress. This constitutes a major inconvenience for users, specially for those involved in continous testing of new instruments, procedures or models.[2]

Moreover, it is quite frustrating that advanced users, perfectly aware of the mathematics of their experiments, cannot extend the software to fit their needs.

As in the former section, the above remarks lead to the following design and implementation items.

Design concept: The user must be allowed to define observable, condition and parameter new abstract data types. (Actually, ACX has been developed in this way by the authors.)

Realistic goal: that the addition of new models take only the formal definition of the observable and new parameters involved (definition of the particular observable and parameter abstract data types), the coding of the observation equations and their jacobian matrix, and linking; that very limited knowledge of the software be essential for the extension of the models.

2.3 Portability

Moving from one platform to another is unavoidable for many reasons: different collaborating organizations might have different platforms, the optimal development platform might not be the optimal production platform, the optimal field platform (PC) might not be the optimal office platform, the optimal platform today might not be the optimal platform tomorrow, etc.

Therefore, from the overall software architecture to the programming habits, portability has to be a main concern. It is a fact that highly qualified professionals waste too much time in transferring software to different platforms and plotting devices.

A last observation is that contrary to classic algorithmic programming languages, which happen to have well established standards, Graphic User Interfaces (GUI) are troublesome for scientific programmers and analysts. Actually, GUI standards appear to be competition weapons between manufacturers (see, for instance, [11]) rather than tools to easy the work of developers.

As before, two summarizing ideas are highlighted.

Design concept: define levels of interface complexity, built upon the same data structures, in such a

[1] In the photogrammetric literature general network adjustment programs are usually referred to as combined adjustment programs.

[2] In the opinion of the authors, it is a misunderstanding that operational software mean stiff software; and conversely, that "flexible" research oriented software mean a poor user interface.

way that a minimal sufficient set of functionalities for professional work are guaranteed in case of a platform migration. For GeoTeX, three levels, 0, 1 and 2, were defined. 0 is the lowest interface level with essentially no tools other than those provided by the operating system; 1 is the alphanumeric interface level and makes no other assumption than the ANSI standard; 2 is the graphic interface level, most likely to degrade after a migration.

Realistic goal: that the system and the adjustment program be able to run at least at the interface level 1 (see Section 5.2 and Figure 2) in the three *de facto* standard operating systems in scientific computing, DOS, UNIX and VMS.

2.4 Drawbacks of general approaches

A known effect of design principles similar to those of the preceeding sections is the proliferation of software shells which slow down the execution of some processes. Since that increase in computing time is approximately linear, it can be absorbed by faster hardware. In other words, the extra time budget offered by the new hardware can —and should— be partly invested in more complex software architectures. (Note that time and space algorithmic complexity remain a problem.)

The real danger of a general approach is that either a number of useful tailored functionalities may be lost or that irrelevant details for a particular subset of applications may burden the operation of the software.

In this respect, the following design concept must be fully realized in the implementation.

Design concept: generalization of concepts must be rigorous enough to make developers write tailored modules —always of an informative auxiliary nature— in rare ocassions only; privileged users like project or department managers must have the possibility to customize the software by setting defaults and abling/disabling options.

3 ON THE FORMAL STRUCTURE OF OBSERVABLES

The idea of defining the formal structure of observables and other elements participating in an adjustment dates back to the late seventies [15, 17] and it has been driven by the need for automation, either in the design of geodetic data bases [17] or in the design of adjustment systems [7, 14].

Probably, there is more than one approach well suited for the management and reduction of photogrammetric and geodetic data. Even for a given data structure approach, there might be different complexity levels; for instance, the general data base might contain information irrelevant to an adjustment.

For the sake of simplicity and because the GeoTeX data base is not completely defined a simple *adjustment oriented* point of view will be adopted. Then, the main data types are: observables, parameters, constraints and sensors (instruments may be called as well). The sometimes used term *formal structure* of observables, is nothing else than a rough expression for the definition of abstract data types. The data type *observable* is defined in GeoTeX as

$$< p_1 \dots p_i > < s_1 \dots s_j > < a_1 \dots a_k > \qquad (1)$$
$$< o_1 \dots o_l > < c_1 \dots c_m >,$$

where $p_1 \dots p_i$ are the identifiers of the parameters involved, $s_1 \dots s_j$ the identifiers of the instruments, $a_1 \dots a_k$ auxiliary information (meteorological, etc.), $o_1 \dots o_l$ the actual observed amounts, and $c_1 \dots c_m$ some representation of the covariance matrix.

The above definition is polymorphic in the sense that a particular observable data type is a particular case of (1). A photogrammetric observation data type would be defined as

$$< p_1 p_2 p_3 > < s_1 > < o_1 o_2 > < c_1 \dots >, \qquad (2)$$

where p_1 stands for an image orientation parameter, p_2 for a point, p_3 for a selfcalibration parameter, s_1 for a metric camera sensor, $o_1 o_2$ for the image coordinates, and $c_1 \dots$ for the statistical information. Last, a photogrammetric observation, i.e. an instance of the former abstract data type could be

$$8623 \;\; 13245 \;\; 1 \quad 9001 \quad 2345.7 \; -92356.6 \quad 7.5. \qquad (3)$$

A control point observable data type would look like

$$< p_1 > < o_1 o_2 o_3 > < c_1 \dots > . \qquad (4)$$

Analogously, the abstract data type *parameter* is defined in GeoTeX as

$$< p_1 > < a_1 \dots a_k > < o_1 \dots o_l > < c_1 \dots >, \qquad (5)$$

where the meaning of $p_1, a_1 \dots a_k, o_1 \dots o_l, c_1 \dots c_m$ is clear.

Constraint and *sensor* data type definitions follow the same philosophy.

4 DATA STANDARDS, ABSTRACT DATA TYPES AND DISCRETE NETWORK MODELS

It is necessary to differentiate three concepts used throughout the paper: data [transfer] standards, network abstract data types and network discrete models.

Data standards and some of their implications are discussed in Section 4.1. Note that data standards is a concept mainly related to data management and processing.

Network abstract data types, either for a data base or for a program, deal again with management and processing. Actually, Section 3 deals with just a small part of the problem of defining a consistent and comprehensive set of abstract data types for a network adjustment program.

A network discrete model is, on the contrary, a pure mathematical concept. It is motivated and introduced in Section 4.3.

4.1 Data standards, grammars and the GeoTeX I/O kernel

A data standard is nothing but a set of formal rules describing the structure and meaning of data, that is, a language. Geodetic data, as many other types of data, may be expressed by means of a formal language. This is important from many standpoints: aesthetics, automatic language recognition, automatic code generation, etc.

A language is defined by a grammar. Grammars are always formal and some of them can be *automatically* processed by a computer. In fact, it is possible to create a tool —a *compiler-compiler*— which, using a formal description of a grammar G —i.e. a metalanguage— as input, generates software able to recognize text files written in G [1]. A well known commercial example of such tools is the Lex & Yacc package [13].

As discussed in Section 3 and also in Section 2, the formal structure of the observables, parameters, etc. may be represented by means of polymorphic abstract data types. An immediate —and very important— consequence of this fact is that only a single limited grammar is required to define the language describing those geodetic items. Thus, only couples of modules have to be coded for the reading and writing basic operations.

The GeoTeX I/O kernel has been implemented following these principles. First of all, the AdIL grammar, which represents the formal geodetic data model, was defined. Then, a compiler-compiler tool, GDL, was developed. GDL uses that grammar to automatically generate the skeletons of the I/O modules. At this point, the software is able to decide whether a text is written in AdIL or not (that is, to decide if that text is syntactically correct).[3] Finally, all the semantic procedures were coded.

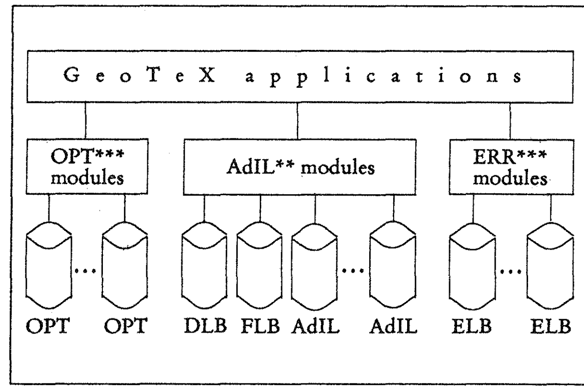

Figure 1: GeoTeX I/O kernel.

Specifically, the GeoTeX I / O kernel is composed of:

- A descriptor library (DLB), used to describe the particularities of the observables, parameters, sensors and constraints. One record —a descriptor— is stored in this file for each single instance of those geodetic elements.
- AdIL** modules. The GeoTeX I/O kernel consists of couples of modules (which are used to read and write the observables, etc.). The output modules also use a format library (FLB), created and tailored by the user, to write the output AdIL files according to the user's preferences.

These previous components have been specifically designed for the GeoTeX system. Additionally, other general input/output subsystems, shared by many other applications of the ICC, are used. Exactly the same rationale —formal grammars, abstract data types and automatic code generation— can be and has been applied to these types of data —not only to geodetic.

The additional general I/O subsystems are:

- OPT*** modules (OPT grammar). These two modules are used to read and write the "run option files" (that is, the set of options controlling the behavior of a program).

[3]Note that the tools known as compiler-compilers are only able to generate the *syntactical parsers*. That means that the modules created in this way can only recognize whether a text is written according to a grammar or not. The semantic process of the information contained in the input file has to be implemented manually. Nevertheless, the syntactical recognition of a text is a hard problem to solve, so the availability of those tools is of a great help.

The difference between syntactical and semantic processes is shown by the following example: if 182.4527 is a string of characters representing an angular magnitude, the syntactical process of such a string would check whether the format DDD.MMSS is used or not; the semantic treatment would transform, for instance, this magnitude from gons to radians.

- ERR*** modules (ERRMSG grammar). All the error messages related to the system have been stored into an "error message library" (ELB), identified by an error code number. The ERR*** modules gather the text of these error messages and display them according to a printing standard. The main advantage of such conception lies in the fact that it is possible to modify such messages with no changes in the software. Thus, multilingual versions of the system can be implemented with no additional development cost!

GeoTeX takes advantage of these subsystems as well (see Figure 1).

4.2 Abstract data types

A data type is any of the forms that information may adopt according to a classification criterion. The `real x` FORTRAN declaration states that x is an object whose type is real. More modern programming languages allow for a recursive construction of new —user defined— data types. Nevertheless, these programming languages are still third generation ones.

Thus, for instance, the practical implementation of the formal structure concept of observables in Section 3 could be

```
photogrammetric_observation x(n),
```

which would define the object x as an array of n photogrammetric observations. In this context, a photogrammetric observation is an abstract data type.

It is possible to go further up in the abstraction level. For example, the following sentence

```
observation x,
```

would redefine x as an observation of *any* type. Again within the scope of this context, this is a polymorphic abstract data type. It would be possible then to define polymorphic operators on observables which would take different actions —i.e. procedures— depending on the specific observation being processed; photogrammetric or control point observations.[4]

GeoTeX has been designed following the paradigm described above as far as allowed by the limitations imposed by the available tools (a FORTRAN compiler). Nevertheless, the polymorphic concepts of observation, parameter, sensor and constraint have been implemented using the techniques described in the previous sections. Note that there is also a close relation between polymorphic abstract data types and the data standards used to transfer the information.

[4]One could add two observations of different type. In such a case, an error condition would be returned.

Ideally, this abstraction process would lead to a final polymorphic abstract data type, the network —which would embody, among others, the observation data type. Powerful polymorphic operators on such an object could be defined, for instance as: `adjust_network`, `print_network`, `transfer_datum`, `create_subnetwork`, etc.

4.3 Discrete network models: the ACX discrete math model

The ACX network adjustment program, which incorporates most of the design considerations of Section 2, is the central component of the GeoTeX system.

There are many definitions available for networks in the context of least squares adjustment which more or less read *it is a set of points related through observations...*, but which are not definitions in any mathematical sense. On the other hand, for an adjustment the well known functional model —linear or not— and the stochastic model may do, though the structural (or topological) information of the network does not show up explicitely.

That the structural aspects are explicitely formulated is of practical importance.

For instance, structurally seen, a distance observation between two points is equivalent to a vector difference observation between the same two points. Both observations have the same influence on the numbering of unknowns in the normal equations and in their loading sequence. For the two purposes it is even irrelevant whether there are one or more repeated observations.

Another point in favor of explicite discrete models is based upon the following remark. For anyone who has ever written an operational production-valid adjustment program it is well known that the so called organizational tasks represent at least 70% of the analysis and coding effort. Many operations and algorithms thereof are of a discrete nature:

- extraction of parameters from observations,
- generation of the network graph,
- optimal graph vertex numbering for solving the normal equations,
- optimal numbering of the observations for the loading sequence of the normal equations,
- generation of the elimination graph (generation of the symbolic fill-ins),
- analysis of identification errors,
- generation of information for the sparse matrix numerical modules,

- ...

Now, if a look at existing software is taken, one will probably find out that most components (data bases; functional, statistical and numerical modules) are more or less alike. However, the discrete parts are either very different or, even worse, they are not clearly separated from the rest of the software. This is an indication that, still today, the discrete model and the corresponding discrete software modules are missing concepts in our systems.

As a discrete network model the concept of *hypergraph* is proposed [2][p. 389] (see as well [5, 6]). If V is a a finite non empty set and E, $E \subset \mathcal{P}(V)$, a family of non empty subsets of V such that

$$\bigcup_{E_i \in E} E_i = V,$$

then the couple H, $H = (V, E)$, is called a *hypergraph*. The elements of V are referred to as the *vertices* of the hypergraph. The *edges* or *hyperedges* are the elements of E.

The vertices of the hypergraph clearly correspond to the network parameters and the hyperedges to the observations. Note the n-to-1 correspondence between observations and hyperedges and, accordingly, the same correspondence between the design block [sparse] matrix of the adjustment. An additional advantage [5], is that if H is a network discrete model —i.e., a hypergraph— then the *representative graph* G of the *dual hypergraph* $H^\star$ is the graph of the network in the usual sense: an edge between two vertices (parameters) exists if the two parameters are involved in a same observation.

In short, all the structural information of the network is contained in its associated hypergraph. (For other additional properties see [5, 6].)

5 SYSTEM DESIGN

The coding of the first modules of GeoTeX/ACX started by the end of 1988. Their architecture is a compromise between the ideas described here, in [4, 5, 6], and the means available.[5]

5.1 GeoTeX architecture

From the architectural standpoint, GeoTeX is a system consisting of two types of software components: *heavyweight* and desktop applications. This classification is based on two criteria: the complexity of the functions to be implemented by the software — and therefore, its size— and the working environment required. Within the context of the GeoTeX system, a working environment is a standardized set of input/output files, user procedures and system resources. See Section 4.1 for a description of the GeoTeX I/O kernel and Section 5.2 for more information about procedures.

Type	*Complexity*	*Environment*	*Examples*
Main	High	Full	ACX, GAI
Utility	Medium	Full	Plotting
Desktop	Low	None	Coordinate transf.

Table 1: GeoTeX software components.

Heavyweight applications are the most complex and a full working environment is required. On the contrary, an almost non-existent environment is used to run the much simpler desktop applications (usually, screen interaction).

From a photogrammetric/geodetic point of view, the heavyweight applications are divided into main and utility applications. This classification is made for the sake of practical use. Main applications are much more demanding in terms of computer resources (usually main applications are executed in batch mode and utility applications in interactive mode).

The software components of the GeoTeX system are listed in Table 1.

The GeoTeX files may be classified into two groups: user and system files. The user files are those created and modified by the user during the life of each project (that is, the usual input/output files required by any system).

The system files are the implementation —using the available resources and tools— of the abstract data types used to model geodetic data. These files remain unchanged during the exploitation of the system but may be modified by the advanced users to upgrade or expand GeoTeX (correspond to the GeoTeX input files in Figure 3).

In a near future (see Section 6 and Figure 3), GeoTeX will be able to interface with geodetic/photogrammetric, topographic and other databases by means of utility applications.

A workstation (with the set of graphic/alphanumeric functionalities required to run the selected interface level) and a DIN A3 fast PostScript plotter —for work plots— is the minimum local configuration recommended to run GeoTeX. Additionally, a link to the LAN of the organization would be advisable —mainly when photogrammetric/geodetic etc.

[5] To be honest, the maturing of some of these ideas were not completely independent of the simultaneous realization of the software.

data bases are available. The photogrammetric analytical systems may be connected to the workstation of to the LAN system.

5.2 Interface levels

As discussed in Section 2.3, three interface levels were defined for GeoTeX. Level 0 consists of the set of tools offered by the platform's operating system —editors and batch commands; level 1 is composed of alphanumeric ANSI-based utilities —as for instance, syntax oriented file editors; level 2 is (or will be) based on Graphic User Interface (GUI) and Graphic System (GS) packages.

The reason for defining such interface levels is the authors' aim at creating a portable system and, at the same time, offering a helpful —also portable— set of tools to its users. To achieve that, the usage of widely spread GUI/GS packages has been carefully avoided. As it is well known, nowadays there is not a fully accepted GUI/GS standard. Therefore, the realistic goal proposed in Section 2.3, —running GeoTeX on DOS (the field platform), VMS (the current platform) and UNIX (the threatening platform) at the interface level 1— has been the authors' main objective.

A simple, character-oriented, ANSI-based, portable user interface is used by level 1 utilities. Minor or no changes are required to migrate such utilities to new platforms. Hence, a standard set of tools is guaranteed[6] to the user. Once this objective has been fulfilled, it is possible to develop the corresponding level 2 utilities for specific environments.

Note that "character-oriented" does not mean *poor interface*. The quality of a user interface depends on several factors. Of course, aesthetics play a very important role —shorter learning times, better usage, understanding and acceptance of the application by the users, etc.—, but there are other components which contribute to the quality of the final result. The concept behind the interfacing software, sometimes called the interface's foundation, is the key point here (see [10]). The mental data model the user has to assume, the available functions, the navigation scheme and the look & feel of the interface being used are the critical success factors to take into account.

It is recalled that when the GeoTeX interface foundation was conceived (see again *Design Concept* in Section 2.3), a clear objective was pursued: to offer the user a professional, upgradable interface, including the minimum set of functionalities required to perform his task. These functionalities —the *feel* of the interface— have been implemented at level 1 by

[6]Syntax oriented file editors, data screening, coordinate transformations, graph utilities, interface with data bases and plotting (PostScript). PostScript is becoming a *de facto* standard for plotting in scientific environments; see as well [3].

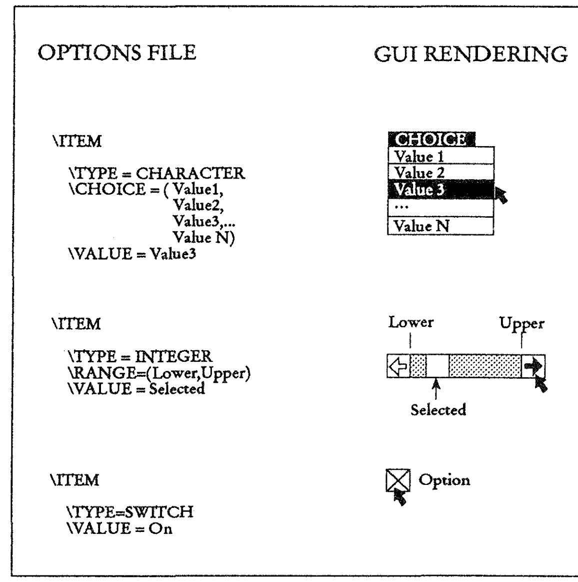

Figure 2: Option files and GUI rendering.

means of a character-oriented *look*, the ANSI portable package. Of course, it is possible to use more sophisticated tools, specific non portable GUI/GS environments, to upgrade that look (level 2); the feel, nevertheless, will remain unchanged (see Figure 2).

Thus, note that GeoTeX offers to its users a portable, professional level 1 interface, covering a basic range of functionalities. This interface may be upgraded from the "rendering" point of view using GUI/GS non-portable packages. Nevertheless, the concept behind the interfacing software is the same for all levels.

5.3 ACX architecture

A layout of the structure of ACX is depicted in Figure 3. ACX, like GeoTeX, has been developed as a compromise between the limited available means and the paradigm of object oriented programming. Figure 3 is almost selfexplanatory though somewhat simplified. The generation of initial approximations is, at the moment, done with a separate main application program (GAI) but the next version of ACX will embody GAI; this makes the program flow more complex than in Figure 3 since intermediate adjustments with simplified linear models have to be added.

Note the files in the dashed boxes in the same Figure 3. The upper left box contains the GeoTeX system files. The most important one contains descriptors which serve as links between abstract data types and data [transfer] standards The lower left box contains the ACX files whereby the most important one contains the network abstract data type definitions.

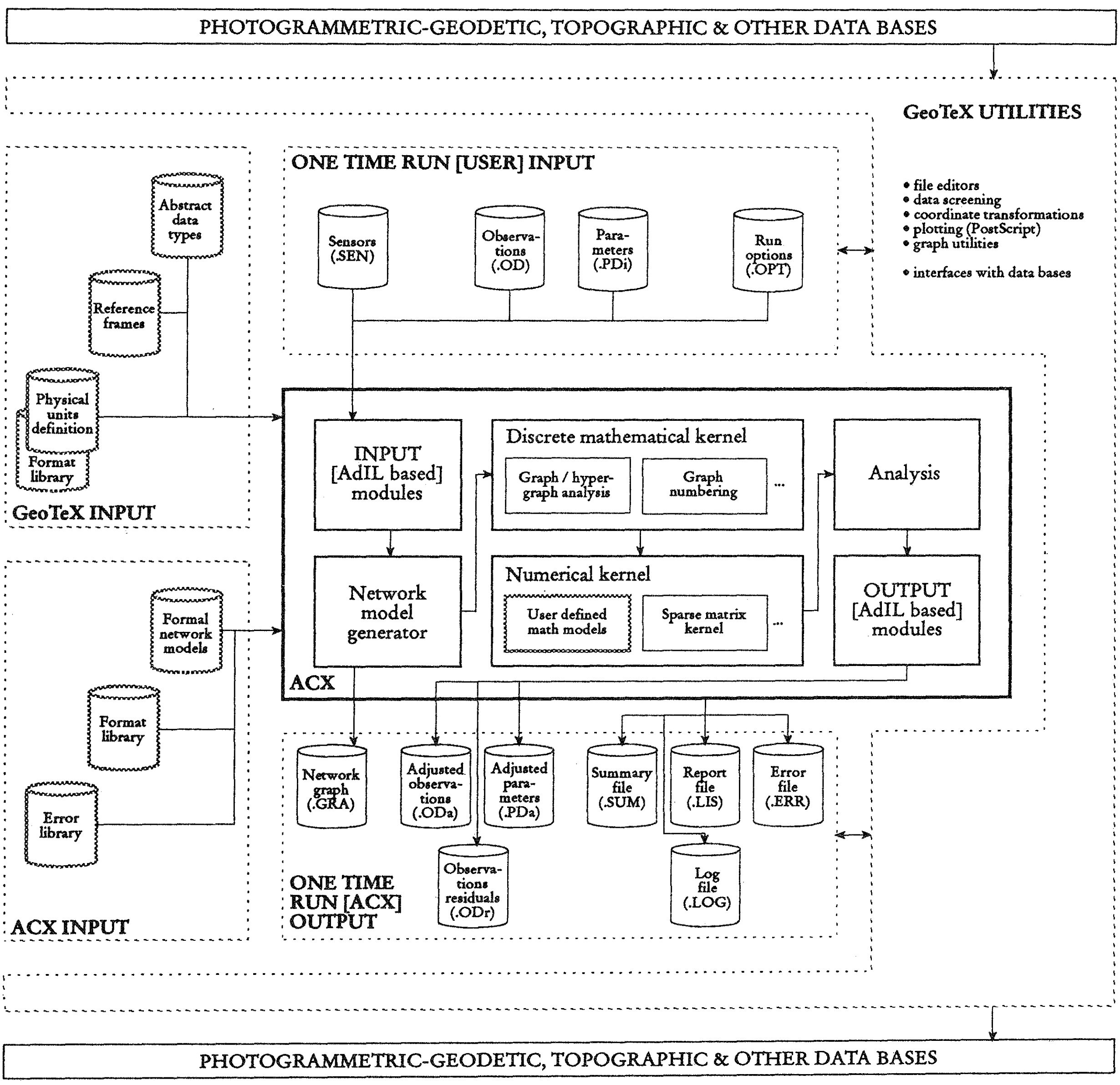

Figure 3: Schematical layout of the ACX general network adjustment program and its environment.

Thus, when the advanced user wants to implement a new model, she/he has just to edit the files within the dashed boxes; the two mentioned files (the major task) and other 5 remaining files (minor details like output formats that are not absolutely necessary since ACX will make default decisions). Then she/he has to program a FORTRAN subroutine for the observation equations and their derivatives according to certain calling conventions; this subroutine is added to the ACX object module library and a new executable module is generated. The pure photogrammetric or geodetic engineer or scientist needs to know very few about the program and just nothing about the discrete and I/O modules.

6 CURRENT STATUS AND FUTURE DEVELOPMENTS OF THE GeoTeX SYSTEM

GeoTeX is an operational photogrammetric/geodetic point determination system. Besides research, it is supporting production projects in geodesy and development projects in photogrammetry; the SPOT model is undergoing implementation in collaboration with the authors' colleagues of the Remote Sensing Group and with the team of the *Lehrstuhl für Photogrammetrie und Fernerkundung* at the Technichal University in Munich [9].

There are two missing pieces in the system. One is the interface to the photogrammetric, geodetic, topographic and other data bases which, with the exception of the topographic —DTM— data base, are undergoing definition.

The other missing piece of software is a level 2 interface (Section 5.2) which requires a GUI system including a GS.

Last, with the new generation of digital photogrammetric stereo workstations, a closer integration of the point determination system and the measuring instruments is foreseen. The stereoscopic display capabilities and the new fast hardware might change some working routines in the sense of easier network analysis and, in many cases, almost real time adjustments.

References

[1] Aho,A.V.,Sethi,R.,Ullman,J.D.,1986. Compilers: principles, techniques, and tools. Addison-Wesley, Reading.

[2] Berge,C.,1973. Graphs and hypergraphs. North-Holland, Amsterdam.

[3] Cavuoto,J.,1991. PostScript moves up. *Computer Graphics World.* 14(7): 105–108.

[4] Colomina,I.,1989. Combined adjustment of photogrammetric and GPS data. In: *Schriftenreihe des Instituts für Photogrammetrie*, Universität Stuttgart, Vol. 13, pp. 313-328, Stuttgart.

[5] Colomina,I.,1991. Structural aspects of hybrid networks in geodesy and photogrammetry. Ph.D. dissertation, Departament de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Barcelona.

[6] Colomina,I.,1992. Discrete mathematical techniques in the analysis and adjustment of hybrid networks. In: *International Archives of Photogrammetry*, Vol. 29, Comm. III.

[7] Crippa,B.,de Haan,A.,Mussio,L.,1989. The formal structure of geodetic and photogrammetric observations. In: *Proceedings of the Tutorial on Mathematical Aspects of Data Analysis*, ISPRS, Intercomission WG III/VI, Pisa, pp. 9–35.

[8] Elassal,A.A.,1983. Generalized adjustment by least squares (GALS). *Photogrammetric Engineering and Remote Sensing*, 49: 201–206.

[9] Heipke,C., Kornus,W., Strunz,G., Colomina,I., 1992. Automatic photogrammetric processing of SPOT imagery for point determination, DTM generation and orthoprojection In: *International Archives of Photogrammetry*, Vol. 29, Comm. IV.

[10] Hildebrand,J.D.,1991. Menus, mice & mom. *Computer Language.* 8(9): 42–48.

[11] F.U.W.,1991. The empire (IBM) strikes back (at MICROSOFT and the ACE consortium). Imaging technology report, 4(8).

[12] Judge,P.,1991. Guide to IT standards makers. Technology Appraisals, Isleworth (UK).

[13] Mason,T.,Brown,D.,1990. lex & yacc. O'Reilly & Associates, Sebastopol (California).

[14] Sarjakoski,T.,1988. Object-oriented approaches in the design of more capable (adjustment) systems. In: *International Archives of Photogrammetry*, Vol. 27, part B10, Comm. III, pp. 244–253.

[15] Schwarz,C.R.,Tscherning,C.C.,1982. Introduction to and report from the Symposium on Management of Geodetic Data, august, 1981. *Bulletin Géodesique*, 56: 106–114.

[16] Swanke,J.E.,1991. The realities of portable software. *Computer Language.* 8(11): 57–66.

[17] Tscherning,C.C.,1978. Defining the basic entities in a geodetic data base. *Bulletin Géodesique*, 52: 85–92.