

Artemis: An Event Operation Tool for Telecommunication Management Systems

José Marcos Nogueira, Antonio A.F. Loureiro, Ana Paula R. da Silva,
Daniel Fernandes Macedo, Leonardo Barbosa e Oliveira, Fernando Augusto Teixeira

Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, MG, Brazil

{jmarcos,loureiro,anapaula,damacedo,leob,teixeira}@sis.dcc.ufmg.br

Abstract. *The management of both computer and telecommunication networks is an activity that has as main requirement the integration of heterogeneous systems. This work presents the Artemis tool, which has as key feature the capability to combine distinct systems of network management as well as their information concerning the network in order to exhibit them concisely and through a unique interface to users. The tool is part of the SIS management platform used by TELEMAR, the largest Brazilian telecommunications company.*

Keywords: *Management platform, Operation interface, Integration tool.*

1 Introduction

In the operation of large scale telecommunication and computer networks, one of the main requisites is the need to integrate a large number of heterogeneous systems and equipments. The networks and network elements are managed by protocols and systems that are not always standardized or compatible among themselves. In this scenario, it is important to have a common operation environment for telecommunication providers. In particular in the area of fault management, where the human operator is involved in the management process, a common operation environment is fundamental, since there is no need to make operators proficient with specialized platforms and systems.

A unique procedure to manage different systems, query databases, visualize alarms and interact with faulty elements is essential to ensure quality and response through management actions. However, fault management is not the only activity that benefits from an integrated environment. In fact, event management in general (e.g., security and configuration) and execution of commands to activate services are activities in a management system that can also be improved with the use of a common working environment.

This heterogeneous scenario of a telecommunications network is exactly the scenario where TELEMAR—the largest telecommunications company of Brazil—operates. The plant is composed of a large variety of switching equipments and transmission systems, differing not only by their vendors but also by their technology, age and way of monitoring and control. In order to solve that management problem, a distributed management platform called SIS (System for Integration of the Supervision) was developed [7]. This platform is fully operational in the 16 state companies where TELEMAR operates. This company has 80 million users and is spread over 60% of the Brazilian territory, which corresponds to 53% of the U.S. territory.

The SIS platform is a system in constant evolution. The platform is in operation, and new functionalities are added frequently. Due to this fact, the volume of information and tasks performed by the system is always increasing. Also the number, diversity and locations of managed elements is increasing. Among all the facts that motivated the construction of a new operation tool, the most important ones are the following:

- Management systems are meant to be used uninterruptedly, 24 hours a day. Most operators will use this tool during all the working period, executing tasks that will become routine after a few working days. In such a scenario, usability must be carefully taken into account in the project of a software.

- For large or middle-sized telecommunication plants, the amount of simultaneous events allows the specialization of operators by technology, such as commuting, transmission, and infrastructure. The supervision can be even more specialized in some cases, where operators are responsible for equipments from a certain manufacturer, for example.
- With the use of computer networks and both reliable and high speed links, it is more feasible to centralize the operation of the telecommunications plant, decreasing the number of management centers. Those centers will be bigger, and can be more generic or specialized in determined technologies or categories of equipments.
- An administrative reorganization, merging regional companies into a national company forced the management systems in operation to be used on larger scales.

This change in the dimension of the managed plant forced the use of new ideas and practices that were being faced by the operators (the users of the system) and the software development team. In the case of a large and complex plant, such as the one present at TELEMAR, those ideas were responsible for the creation of the new event operation software called *Artemis*, described in this paper. The main requisites of this tool are:

- Enable an integrated operation of an extensive, heterogeneous, multi-functional communication plant (telephony, data, mobile phones).
- Improve the efficiency of management tasks (e.g., faster response time for complex queries sent to databases, and interface responsiveness) compared to the previously developed and available tools.
- Improvements in usability and ergonomics, due to the intensive use (24×7 modality) of operational software.
- Portability for different computational systems, with minimum administrative cost. Beyond these objectives, the developed software was designed to be easily extensible, to integrate with other management systems from different manufacturers with minimum cost.

The development of the tool followed various phases and considered strict requisites. The observations of daily needs, together with performance measurements of common tasks were the starting point for the development of this software. The previous event operation tool used on the SIS platform, SISTerm [9], was one of the many tools used in the plant. There are other applications for configuration, database navigation and event analysis. However, the focus of the changes was on the event operation application. This is by far the most used application in the platform, and was the most affected during the evolution of the plant. A preliminary requisite specification was presented to the users, and later on a prototype was developed. After the acceptance of the prototype, a detailed requisite specification was written. Those documents and the prototype served as sources of information for the development of the *Artemis* tool. The initial development was conducted in-house on a laboratory, using test databases, followed by tests on real databases. A minimum set of requirements was implemented, and an alpha version was released to the users, for evaluation and testing purposes. New versions were developed, improving the software and adding new functionalities, until the software was considered ready for operation. It is important to note a particular requisite of the development process: the impact of the new application in the plant should be minimum. Furthermore, tests on-site, during normal operation, should not produce errors, nor induce operators to take wrong actions, nor introduce errors to databases. These restrictions led to a careful development, forcing the process to be slower. It is important to mention that this software tool is not bound to the SIS platform. *Artemis* was designed to be easily extended to other management systems as well.

The rest of the paper is organized as follows. Section 2 presents the requisites and goals of *Artemis*. Section 3 discusses some project issues, and presents the architecture and modules of the software. Section 4 describes the implementation of the tool. Examples of use are presented in Section 5. Finally, Section 6 presents our concluding remarks.

2 Project Requisites

This section presents the main requisites for devising the *Artemis* tool. During its specification, the most important points were integration capacities, specialization, centralization and decentralization of the operation, performance, portability, usability, and security. The reference system for the *Artemis* project was, as mentioned before, the SIS platform.

In order to better understand the development process, a brief description of the SIS operating model is presented. In the SIS platform, a plant to be managed—composed of equipments, systems, and telecommunication

networks, all of which called supervised entities—is seen as an hierarchical organization that has geographic administrative regions. A supervised entity belongs to a sub-region that in turn belongs to a region. The regions, when combined, constitute an area or core. The set of geographical areas form the total area to be managed. There are other relationships among supervised entities and system entities. For instance, building stations belong to sub-regions and they have supervised entities. Building stations also are associated with management centers, where operators are located.

The structures of the real plant and of the system itself (configuration data) are stored in a distributed database. Using configuration data it is possible to obtain hierarchical views of the managed plant through *Artemis*, that obtains all information from the database. These views, in the SIS context, are named topologies.

The different entities of the system have different attributes. For example, a supervised entity has, among its attributes, a specialty, a class, domain(s), and a station that owns the entity. Domain is an attribute built arbitrarily with the aim of gathering entities based on a criteria. An alarm has a severity associated with an acknowledgment status, the ID of the operator who carried out the acknowledgment, the occurrence time, etc. Thus, applying filters over attributes combined with topologies allows views under a variety of aspects.

The typical operating model for SIS management, in particular in the case of faults and events, consists of the following cycle: an alarm is collected from the plant by an agent that in turn sends it to its manager that stores the alarm data into the database; one or more operators, through operating terminals, send queries (periodically or on demand) to the database about events present in the system and become aware of the alarms (acknowledgment operation); actions can be taken in order to solve reported problems, e.g., opening trouble tickets or sending commands directly to equipments; whenever a fault is normalized, i.e., corrected, the corresponding entry in the database is updated.

Characteristics and Functionalities. *Artemis* is a graphical operating tool used to manage faults and events in a telecommunication network. It must have a good performance, a friendly interface, and must be robust and correct.

The tool provides a wide and hierarchical view of the entire telecommunications plant to be managed. The interface *Artemis* allows an operator to view different levels of a plant, ranging from a country region, to a unique supervised entity. For instance, a network element, a process, and so forth. This property allows a smooth transition among distinct views.

The access to information in the operation units, for queries or modifications, follows restrictions based on access rights that are defined externally in the management system for each user. Besides the entire visualization of the plant, *Artemis* provides the following functionalities:

- Presentation of event lists, supervised entities and stations where they are located.
- Possibility of defining different filters over lists, as filters of event severity, equipment specialty, acknowledgment of alarms, etc.
- Presentation of statistics about events.
- Possibility of acknowledging alarms by operators with records of who are responsible for them.
- Capacity of obtaining information, besides those presented in the lists, such as the name of the operator who acknowledged an alarm.
- Presentation of event history.
- Possibility of opening terminals to execute commands in the supervised equipments.
- Automation of the update frequency of the lists.
- Ability of opening trouble tickets in external systems.

It is possible to open lists in several operation levels at the same time. The search for information in distinct databases or management systems, if necessary, occurs simultaneously. The capacity of monitoring events from different areas or regions at once requires *Artemis*' performance to be an aspect extremely importance, due to the great amount of data involved in these cases.

Filters applied to lists are performed independently for each list, that is, each list can have its own filter expression – which is applied to the context that is defined for the list.

Statistics about events are displayed individually or by group. Each list contains statistical graphics related to their events. In case it is desirable to observe statistics about the number of events of a wider level, operators have only to request a list from the level in question – which might include the entire telecommunication plant that is being managed. This requirement is necessary due to the dynamicity of the collected data.

It is allowed to perform list combinations originated from distinct systems or databases. These combinations should be done under domain filters, or with the aim of easing operation specialization. A specialist user can work in a view of just a specific domain, for instance, or of a particular manufacturer of equipments.

Concerning usability, it was assumed that the operator will use *Artemis* continuously during many hours. Because there are too much possibilities of operation, and a great amount of information is presented, it is important to have a user-friendly interface. Users can ask for the most relevant information or more details if desirable. They also can easily have a general notion of all operation capacities. The access to each desired operation or information is done quickly and intuitively, by means of icons rather than plain text.

The data organization in an operation screen is flexible so that windows can be deployed in a customized way. Features such as *pop-up menus*, *tool-tips* in icons, representative colors in lists as well as sorted lists by any field and movement of rows are offered in order to facilitate the *Artemis* operation.

One of the strongest aspects of *Artemis* is its customization. This property is obtained by means of a functionality that allows users to specify which attributes, from which types or supervised entities, the monitored data belongs to. For this, system operators make use of boolean and regular expressions, attribute filters (type, class, and so on), and individual choices of nodes that represent devices or a set of devices, which in turn belong to the topology of the monitored network.

Considering the heterogeneity of the operation environment, the application had also portability as a requisite. The compiled code is able to run on different operating systems (Linux, Unix or Windows) and architectures (Intel, Sparc, etc).

The interface also enables improvements in its own functionalities as well as integration of new systems quickly and efficiently. In order to achieve this, *Artemis*' design pursued modularity by means of an object-oriented language and *Design Patterns* [4].

The system was devised in order to reduce costs concerning both development and utilization (licenses). The *Tool* is independent of geographical or logic location.

3 Project

The *Artemis* architecture was designed as a system composed of inter-operating modules. Basically, there are four modules: *data acquisition*, *data management*, *visualization*, and *authentication*. The modules communicate among themselves through pre-defined interfaces. Such a module division makes the system easy to be adapted to new functionalities, such as new management protocols. Such a strategy is critical in large telecommunications plants. The developed system can adapt easily to changes in the user interface, and in the interface to management systems. Figure 1 shows the high-level architecture of *Artemis*.

Data acquisition

The *data acquisition* module communicates with management systems to make queries or to execute commands. This module has communication subsystems that obtain data through protocols or specific API's. The subsystems can connect directly to the network devices using, for instance, SNMP, as well as requesting data from a server. In this case, using database access (JDBC, ODBC) or remote operations (RPC, RMI, CORBA). Connection to proprietary systems can be done using vendor's libraries or through some application. The acquired data are translated to an internal data model.

Data management

The control of *Artemis* operation is done by the *data management* module that has a data structure to store data that represent the managed network topology and the managed entities with their attributes and states, including the associated events. Data are obtained by the *data acquisition* module in response to periodical or non-periodical requests from the *data management* module. All information is stored in internal data structures, in order to be easily handled. Such data structures have been designed considering the possibility of integration with distinct management systems. This module is also in charge of data processing and of generating reports and statistics.

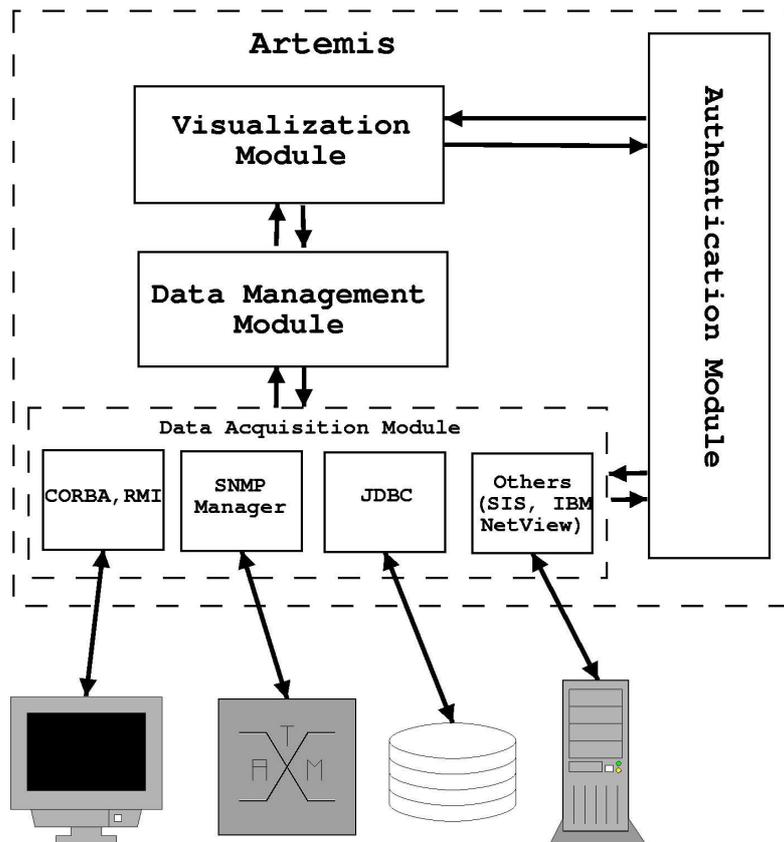


Figure 1: *Artemis*' architecture

Authentication

This module manages the software security and the process of user identification in the various management systems. It asks for the user identification in the beginning of the operation, to provide control on the use of the tool. The information obtained is stored to be used later by the *data acquisition* module when a service or a library module needs to be authenticated.

Visualization

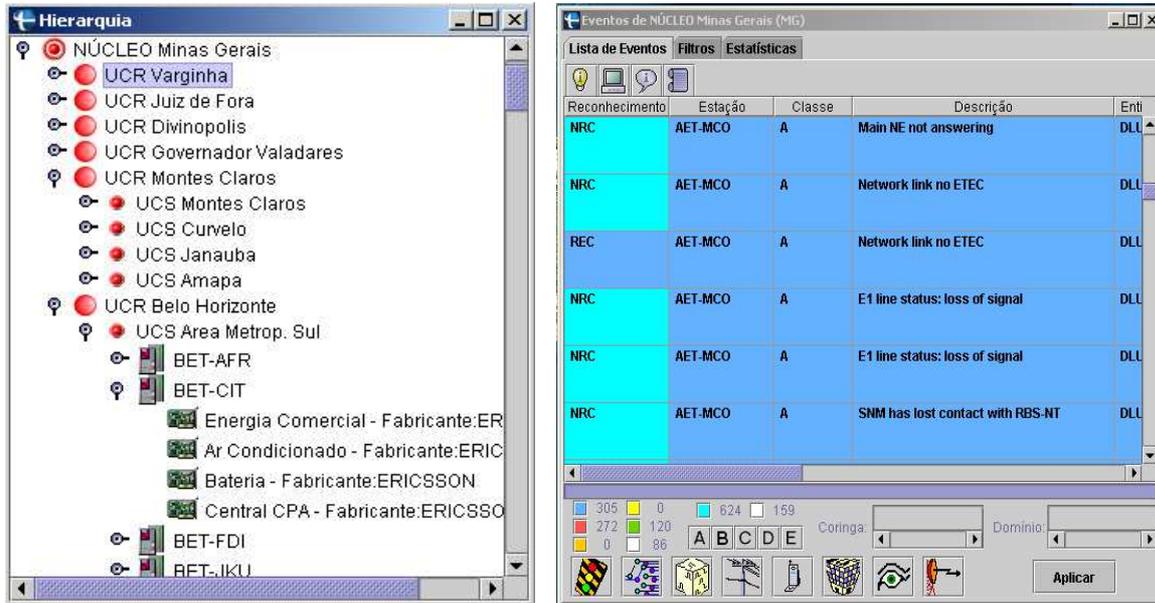
This module implements the operator's interface. It is responsible for showing all information to users. It provides a simplified and intuitive use of *Artemis*. This module shows graphical representations of the topology, event lists, statistical graphics, reports, etc. Furthermore, it makes possible the customization of *Artemis*' interface.

4 Implementation Aspects

In the following we present some aspects concerning the implementation of the interface, and discuss some implementation decisions.

Programming Language. The programming language chosen to develop the *Artemis* tool was Java [5]. Since it is object-oriented, Java has characteristics such as encapsulation and modularity, which in turn are needed in large projects as *Artemis*. The language also makes use of modern programming techniques that allow the increment of code modularity (e.g., Model-View-Controller [3], *Design Patterns* and *Observer/Observable* [4]). In addition, Java has a wide range of APIs, which make possible interactions among heterogeneous systems. For instance, RMI, JDBC, Remote Tea ONC/RPC, etc [8].

Usability. The friendly environment was obtained through the employment of tool-tips, and icons and buttons with well defined functions. Besides that, the customization achieved by using filters and merged data enables users to



(a) Hierarchy Window

(b) Event List Window

Figure 2: Operator Interfaces: (a) hierarchy and (b) event list

create their own “desktops”, which in turn exhibit elements of the hierarchy and the equipments most utilized by operators. These customized “profiles” make tasks easier, specially when the volume of data is too much as it can show only relevant information.

Security. Currently, user authentication is carried out via a request to the operating system asking who is the logged user. Once it possesses the answer, *Artemis* then, through another request, now transmitted to *Network Information System* (NIS), verifies whether the user is valid and which are their permissions.

Robustness and Performance. Several tools that measure the amount of CPU consumed, memory, bandwidth, task executions, and network speed, such as *WMnet*, *Gkrellm*, *XLoad*, *Top*, *MRTG* [6], were used in order to identify and improve weak spots in the tool.

Development Effort. The development, from the specification to running tests, consumed a great number of resources and people. During the entire project, approximately 3000 hours were spent, being most of them employed in the implementation. The final code has about 30000 lines of code.

Software Correctness and Tests.

Successive releases, user suggestions and employment of auxiliary tools during the code development were some of the strategies used in order to achieve a higher degree of software correctness. The tool *Concurrent Version System* (CVS) [2] was used for boosting software updates from all codes that were being developed simultaneously. For managing bugs and future tasks, we opted to employ the *Bugzilla* tool [1].

5 The Interface in Operation

This section presents some screen-shots of the user interface, as well as a brief performance study of the software. The example screen-shots were taken using data from the SIS telecommunication managed plant.

The *Hierarchy Window*, as shown in Figure 2 (a), is used to present an hierarchical view of the telecommunications plant. In the top of the hierarchy, we have a representation of the entire plant (Core). Below it, there are subdivisions — Regional Central Units (UCRs) that are divided further into Secondary Central Units (UCSs). UCSs encompass the stations, where the Supervised Elements are located. Through this window the operator is able to request lists of any of the mentioned levels.

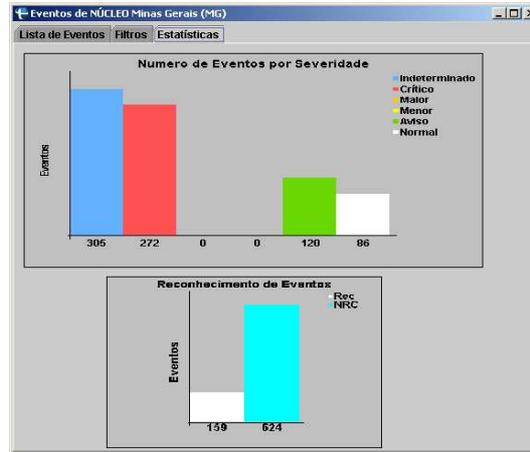


Figure 3: Event statistics

Area MG	Station List		Equipment List		Event List	
No. of elements	1597		17699		783	
Software	SISTerm	<i>Artemis</i>	SISTerm	<i>Artemis</i>	SISTerm	<i>Artemis</i>
Latency time	1min 40s	6s	3min 48s	1 min 20s	10s	8s

Table 1: Comparison of exhibition times between *Artemis* and SISTerm

The *Event Window* is the *Artemis* interface where the user can visualize and interact with the events occurring in the telecommunications plant. This interface has several tabs. The main tab, shown in Figure 2 (b), displays a list of tuples. Every tuple is related to a single event, and its attributes are characteristics of the event, such as time of occurrence, severity, type, description, and if an operator has already taken the responsibility for the resolution of the event. The list displayed in the screen-shot contains 783 tuples, and it took only 3 seconds to show it, including querying and exhibition times.

The operator can use this window to execute several actions on the events displayed in the list. Above the list there are 4 buttons, which execute the following actions: *Acknowledgment*, *Event History*, *Terminal* and *Event Details*. The *Acknowledgment* action records that the operator is aware of that event. The *Event History* function lists the events that happened in a given supervised element. The *Terminal* functionality provides the possibility of remote access to the network element, so the operator can issue commands. The last button, *Event Details*, shows more information related to a given event. All those functions are also accessible through pop-up menus.

The lists can also be filtered. The *Filter Tab* enables the use of *boolean* and regular expressions to control what events will be displayed. Those filters can also be modified at the “Filter Summary”, located below the *Event List* on the *Event Window*.

The *Statistics Tab*, shown in Figure 3, presents graphs related to the severity and acknowledgment of events. Those statistics are also present in the “summary” of events located at the *Event Window*, below the event list.

A number of experiments were made to measure the performance of the tool and compare it with the tool used previously in the SIS platform, called SISTerm. In most cases, our software performed better than SISTerm. In some tests, we had an improvement of 15 times over SISTerm, as shown in Table 1. The impact in response time is important on the daily use of the software.

6 Concluding Remarks

This paper presented *Artemis*, a tool that provides network operators involved in the management process, a common operation environment. This kind of tool plays a key role in large scale telecommunication and computer networks, which need to integrate a large number of heterogeneous systems and equipments through a common operation environment. In particular the tool deals with fault management, a fundamental area that

network operators are permanently involved during the management process. However, other areas such as event management and execution of commands to activate services are activities in a management system that can also be improved with the use of a common working environment.

Using *Artemis*, there is no need to make operators proficient with specialized platforms and systems. Furthermore, *Artemis* was designed in such a way that is not limited to the SIS platform. In fact, the tool was designed to be easily extended to other management systems as well. This is probably the best benefit a tool can provide in a company like TELEMAR, where new network elements and management platforms need to be integrate frequently.

We can say for sure that the development of *Artemis* was very successful, and achieved the proposed goals. We plan to extend the tool in other aspects as well. For instance, to incorporate new functional areas. The development of *Artemis* was a complex activity and the results obtained are very stimulating.

Team and Thanks.

The design and development of *Artemis* was the result of hours of work from a large team of professionals and students. Besides the authors, the following people worked on some parts of the project: Sérgio de Oliveira, Cláudio Márcio de Souza Vicente, Waldir Ribeiro Pires Júnior, Daniel Scaldaferri Lages, Rodrigo Tadeu Gonçalves Silveira, Elton Corrêa de Almeida, Fernando Lages Rodrigues.

This work was conducted in the Network and Distributed Systems Lab at the Federal University of Minas Gerais.

References

- [1] Bugzilla. <http://www.bugzilla.com/>.
- [2] Concurrent versions system. <http://www.cvshome.org/>.
- [3] Cay S. Horstmann & Gary Cornell. *Core Java - Volume 1 - Fundamentos*. Makron Books, 1999.
- [4] Mark Grand. *Patterns in Java - Volume 1*. John Wiley & Sons, Inc, 1998.
- [5] <http://java.sun.com/>.
- [6] Mrtg - multi router traffic grapher. <http://www.mrtg.org>.
- [7] Dilmar M. Nogueira, Jose Marcos S. & Meira. The sis project: A distributed platform for the integration of telecommunication management systems. *IEEE/IFIP 1996 Network Operations and Management Symposium - NOMS'96*, 1996.
- [8] Remote tea onc/rpc for java. <http://acplt.plt.rwth-aachen.de/ks/english/remotetea.html>.
- [9] Equipe SIS. *Sistema Integrado de Supervisão - Manual do Operador*. SIS-DCC-UFGM, 2000. <http://www.sis.dcc.ufmg.br/sis2/utills.cgi>.