SITIE: A Health Care Workstation Integration Architecture for Epidemiologists

LAURENT TOUBIANA, JEAN-FRANÇOIS VIBERT, PHILIPPE GARNERIN, AND ALAIN-JACQUES VALLERON

INSERM U263, Institut Fédératif Saint-Antoine de Recherches sur la Santé, 27 rue Chaligny, 75571 Paris cedex 12, France

Received October 12, 1994

SITIE is a workstation devoted to epidemiological data selection, analysis, and representation. Epidemiological data are collected using the French Communicable Diseases computer Network and stored in an Oracle database. Disease selection and spatial and temporal representations can be done using a user friendly graphic interface. SITIE automatically generates SQL requests to the Oracle database, extracts the data, processes it according to the user’s choice, and represents the data. SITIE is built upon the AVS graphic package and allows the creation of new applications using previously developed modules linked together to form a network. SITIE can be viewed as a tool box for epidemiological data representations. Three example applications are detailed in order to make clear the way a new application can be built using SITIE. The reusability of modules is exemplified by a compound application. © 1995 Academic Press, Inc.

INTRODUCTION

The French Communicable Diseases computer Network (FCDN) was established in France in 1984 for the national surveillance of communicable diseases (1–3). An epidemiological database is maintained and updated practically in real time. This database provides information for the French public health system, allowing for a rapid response. Briefly, FCDN collects, in 95 French departments, the statistics of 12 (of 20) notifiable diseases. FCDN collects data from approximately 500 sentinel general practitioners (SGP) spread throughout the country, using terminals or personal computers with modems. SGP collect clinical and demographic data on, among other diseases, influenza, hepatitis, measles, urethritis, mumps, diarrhea, chicken pox, and HIV tests.

This system is accessed by many users for different purposes. For example, a doctor or a health officer might request statistical information on a disease in a particular region using different modes of representation, or an epidemiologist researcher might access to the system to provide data for the development of a model. They all utilize the system to extract, treat, and represent data which
can be performed in many different ways. We have observed that users lose time when rebuilding general tasks that others have already done.

A large number (385 Mb) of epidemiological data have been gathered since 1984 and stored in an Oracle relational database. For example, more than 150,000 individual cases of flu have been described. At the beginning, data were extracted using hand written SQL requests, stored in files, and processed using specific tools and Mathematica to detect epidemics and represent the spread of diseases (4). A first attempt to enhance the data processing led the use of SAS Access to Oracle to process the FCDN data. However, it limited the user system interaction.

As an information and research center, the response to users' demands required a comprehensive change. The goal was to develop a system with an integrative architecture which can keep the old developments active and be adapted to new requirements.

In order to give a more interactive access to the FCDN database, we developed a new system for workstations, structured around the concept of modularity. This workstation offers, in a friendly graphic environment and in a coherent platform, a set of general procedures to ensure rapid access to pertinent epidemiological information.

Inexperienced computer users can have an immediate FCDN database visibility without any knowledge of numerous complex softwares. This workstation concept is based on an integration of various tools to selectively extract data, process, and then represent them using either spatial or temporal representations. More advanced computer users can connect these tools graphically to constitute their own developments (applications). Finally, C programmers can develop specialized tools that can be used to build new applications.

This software package is called SITIE,¹ a French acronym for Station Intégrée de Traitement des Informations Epidémiologiques, which roughly translated means "integrated workstation for the processing of epidemiological information."

**Concept of SITIE**

SITIE is a tool box that provides the user with a set of software tools called *applications*, which are full-featured procedures easily allowing a complete data processing. An application involves a large number of different tasks: automatic construction of SQL requests, communication with the FCDN database, downloading of the requested data, statistical computations, and graphic representations.

For developing the epidemiological workstation SITIE, we defined a users' request as a procedure called *application*. Each procedure is made of a set of elementary tasks. We tried to find out the fundamental commons among users' demands. We, in fact, discovered that almost all the tasks were common to

¹ Pronounced "city."
users' demands and only a few marginal tasks could not be classed within this group. We developed the common tasks so that users do not have to re-create them each time they want a new application. Thus, users can develop only the interests that are specific to their application.

A task is designed to be a flexible and powerful processing component, named module. Modules are objects representing the building blocks of an application; they must be linked together to create a complete application. A module can be viewed as a computational unit. Each of them can accept data as input, compute them, and generate data as output. Under certain conditions the computing can be modified by parameters. These parameters can be modified by users using a push-button and menu-oriented graphic user interface (GUI). From the user's side, modules are quickly and transparently executed.

Modules can be grouped together according to their function and archived as module libraries. It is possible to individualize modules devoted to the data extraction from the database. Other modules are devoted to the data processing, such as statistics or filtering, and finally the last group is devoted to the graphic representation of the previous module categories output (histograms, curves, maps, etc.). At the beginning of the processing chain, we must identify at least one data acquisition module, followed by one or several processing modules, and finally, one or several representation modules.

The structured approach allows the utilization of the same existing modules in several different applications. For example, the selection of an analyzed disease may be common to other analyses and is independent of the way the data will be represented. A module allowing this choice (called select_maladie in SITIE) is shown in Figs. 2, 4, and 6, as well as the module allowing to select the time step during which the data will be analyzed (called select_temps_deb_fin in SITIE). These modules are nevertheless specific to the data source and representation and were specifically developed for SITIE in ANSI C language.

**Development of SITIE**

The concept of module is similar to the concept of subroutine within a structured programming language. Development of modules requires experience in C or FORTRAN programming language.

However, each module can be run individually and can be considered an independent process that has the ability to get information from other modules. They are unable to produce any treatment without other processes. However, when a module is selected, it can run a function if it receives a synchronization signal. The module modifies the data located in a shared memory location. Therefore, the communication between modules is ensured by synchronization signals and shared memory management.

The main advantage of such a system is to create a new application by adding one or several modules in a preexisting application without a complete compilation of the whole application (including linking) to get this new applica-
tion. For the new module, the user is required only to indicate the shared memory available and the events that start this process. We will see later that this can be done graphically. In this way, "intermediate" users can quickly create and check a new application from a previous one. This ensures flexibility and reliability because we do not have to redesign the old functionality; we need only add a module and check immediately if it works or not.

The next step toward efficiency is the manipulation of modules for building or modifying applications. Each module is symbolically associated with a graphic icon. All valid modules belong to a graphic-specialized library where users can pick up the module they need and drag it into a working window, the network editor. Each module has input(s) and/or output(s), graphically represented by small rectangles whose color represents a given input or output data type. This facilitates the construction of the application: a given output color can only be connected to the same input color. The application building is thus reduced to the module selection and to their connection in order to obtain a network. These links correspond symbolically to a data flow, but actually are semaphore gateways in order to operate on data fields located in shared memory. This task may require some training in building complex applications, but this can be done without any knowledge of advanced programming languages.

Modules and links constitute a network that is a graphical representation of an application. Users can save networks and store them in an application library that can be accessed by the means of application menus. These menus can be personal or can be accessed by a large public if its applications are seen as general interest applications.

SITIE applications are grouped together into spatial and temporal families, and for each family, a menu bar proposes several pull-down menus, each of which has 3 to 8 subchoices, for a total of 23 different applications. These applications, as recurrently used by the FCDN team, constitute only a small subset of the possible applications using SITIE and are listed in Table 1. Seventy-five other applications have been developed for specific purposes, but are also available for any user through a file selection box or from a customized personal menu.

**Three Examples of Standard SITIE Applications**

Three examples of applications will be developed in this section: (i) a spatial application allowing the user to choose a disease and to show its incidence at a given time for a given time step (Table 1*); (ii) a temporal application allowing the user to choose a disease and to show its incidence through time during a given time period for a given time resolution (Table 1**); and (iii) a more complex application combining two spatial representations, one identical to the first application, supplemented with a second representation in which the same data were computed with a spatial filtering using the kriging method (5) (Table
TABLE 1
SITIE APPLICATIONS

<table>
<thead>
<tr>
<th>Type</th>
<th>Domain</th>
<th>Application</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases/SGP</td>
<td>Region</td>
<td>Mean incidence by region</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Department</td>
<td>Mean incidence by department*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physician</td>
<td>Incidence by SGP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kriging</td>
<td>Kriged mean incidence by department</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Region-kriging</td>
<td>Region + kriged maps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dept-kriging</td>
<td>Department + kriged maps***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dept-physician</td>
<td>Department map + Incidence by physician</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Region-physician</td>
<td>Region map + Incidence by physician</td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td>Region</td>
<td>Physician FCDN participation ratio by region</td>
<td></td>
</tr>
<tr>
<td>Participation to the network</td>
<td>Department</td>
<td>Physician FCDN participation ratio by department</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physician</td>
<td>Physician FCDN participation ratio</td>
<td></td>
</tr>
<tr>
<td>Demography</td>
<td>Localization</td>
<td>FCDN physicians localization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Population</td>
<td>Urban population of main cities</td>
<td></td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Gopher</td>
<td>Internet access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Demo</td>
<td>Demonstration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Animation</td>
<td>Spatio temporal maps animation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>France</td>
<td>Mean incidence by physician for France versus time**</td>
<td></td>
</tr>
<tr>
<td>Cases/physician</td>
<td>Region</td>
<td>Mean incidence by physician by region versus time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Department</td>
<td>Mean incidence by physician by department versus time</td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>France</td>
<td>Participation ration of physicians for France versus time</td>
<td></td>
</tr>
<tr>
<td>Participation</td>
<td>Region</td>
<td>Participation ration of physicians by region versus time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Department</td>
<td>Participation ration of physicians by department versus time</td>
<td></td>
</tr>
</tbody>
</table>

Note. Applications are indicated as they are partitioned in the menu bars and pull-down menus of the application submenu of SITIE. Here, for the sake of brevity, SGP is indicated for sentinel general practitioner and incidence for number of cases per SGP. Asterisks indicate the illustrated applications (see text).

***For each of these applications, both the network and the GUI will be shown and discussed.

Spatial Application

Figure 1 shows the GUI of the exemplified spatial application. It displays the spatial representation of a selected disease incidence at a given time and
Fig. 1. Spatial application graphic user interface: Departmental representation. Screen copy of the departmental representation graphic user interface. This application produces a departmental map of the incidence of clinical flu. It allows for easy selection of the represented illness, the way it is represented, and the time period considered, using buttons and dials. This screen is functionally structured into five zones artificially delimited by dash frames labeled A to E. Zone A, parameter selection for data access (time step, illness, data filter type, date considered); zone B, process control; zone C, graphic display of the analysis results; zone D, graphic handler, zone E, printer control. Labels are in French: pas de temps, time step; annee, year; semestre, semester; trimestre, quarter; mois, month; semaine, week; maladie, illness; hepatite, hepatitis; rougeole, measles; grippe, influenza; uretrite, urethritis; oreillons, mumps; diarrhee, diarrhea; varicelle, chicken pox; test_VIH, HIV test; annee_debut, beginning year; pas_temps_debut, beginning time step; resultats, results; legende, legend; haut, up; bas, bottom; rien, nothing; mise a jour, update; fichier a imprimer, file to print; imprime, print.

graphically represents the French departments as a spatial unit on a map of France. Figure 2 displays the network that has produced the corresponding GUI including the resulting map displayed in Fig. 1. Both the network and the GUI have letter-labeled frames in order to indicate the correspondence between the network and the resulting screen output. In this network (Fig. 2), frame A groups together modules devoted to data selection. The modules select_
Fig. 2. SITIE network modules for a spatial application. The modules used to build the application of Fig. 1 are linked using thick lines and constitute a so-called network. The figure shows the network editor interface, where all modules and links can be modified using a mouse as a pointer device. Letter-labeled dashed frames delimit the modules corresponding to the same letter labeled frames described in the legend to Fig. 1. Lines correspond to the information flow between modules.

Connecting lines are in color on the actual screen in order to color code the flowing data types and to help the user to make coherent interconnections between modules.

maladie and select_temps_deb_fin, described above, respectively, display the disease radio-box menu (only one push-button may be selected at once) (Fig. 1A, right) and both the time step radio-box menu (Fig. 1A, left) and the two dials (Fig. 1A, bottom). The module select_espace allows one to select a French department as the space unit, as well as the module select_type_data, which selects the analyzed values as the disease incidence. Disease incidence and department space units are fixed parameters in this application. All these modules feed a unique module named SQL_Multi. This module builds the SQL request to access the data stored in the Oracle database (since in a typical configuration, this database is located on another machine connected by a network, the SQL request is actually an SQL-net request). SQL requests are built using the inputs received from the four upstream modules. The module, upon sending a request, waits for the answer, and stores the data locally in a specific SITIE format in order to be available
for the graphic representation modules. Graphic output is synchronized by the data: the graphic representation begins only when all the data are available. The apply module displays an apply pushbutton (Fig. 1B) whose role is thus to validate the data choice and to start the database interrogation and all the subsequent processes. Frame D groups together modules allowing data representation scaling and color coding of data. The module extract_scalar collects the numerical data while extract_min_max determines minimum and maximum values and mx_float is used to choose between automatically computed minimum and maximum values or custom values. Module get_real displays input panels on the screen allowing the customization of these maximum and minimum values (Fig. 1D), while generate_colormap prepares the parameters necessary for the color representation of numerical values, as well as to edit the colormap if necessary. AVS, being devoted to the graphic representation of data, provides the possibility for interactive modification of representation, by zooming, rotating, etc.; set_view displays the "Top" push-button to renormalize the data in order to go back to the default representation. The C frame is mainly constituted by the module polyg, which prepares data for the graphic output. It reads in a file or in a geographic database a set of points to generate a background map using any spatial resolution (in this application the map of France, with its departments), creates a color code according to the epidemiologic data, and fills in with the right color the polygons representing the chosen spatial units (here the departments) as well as their legend. To summarize, polyg produces a geometric colored object (the map) along with the corresponding PostScript file for hardcopy printing if needed. The geometry_viewer module maps the graph on the screen. Finally, the E frame represents the printing facility (printer module) and displays a push-button on the screen (Fig. 1D). This module allows the choice of a printer using a specifically developed Xwindow/Motif widget and spools the prepared PostScript file of the represented map.

Temporal Application

Figure 3 shows the GUI of the exemplified temporal application, displaying the temporal representation of a selected disease incidence during a given time period and for a selected time resolution and the resulting graph. Figure 4 displays the network that produced this screen. This application illustrates the building block concept implemented in SITIE. The A frame groups five modules, three of which are exactly the same as those described in the spatial application (select_type_data, select_temps_deb_fin, and select_maladie). The module select_espace_num provides the choice of the space unit among department or region (a group of departments: France is divided into 95 departments and 22 regions). While the select_temps_deb_fin module is exactly the same as those used in the spatial application, the corresponding part of the screen looks different, since four dials are displayed instead of two (Fig. 3A, bottom). Careful comparison of the A frame of both networks shows differences
in the links between modules. Namely, in the spatial application network, only three outputs are used (two dials are omitted), while in the temporal application network the five outputs are connected. The four modules feed a unique new module named SQL_incid_Temp, which builds the SQL request for the Oracle database from the input received from the four upstream modules, as did the SQL_Multi module of the spatial application. It displays also the apply push-button that starts the database request using the chosen parameters (Fig. 3B).

The graphic output is a standard curve representation available in the AVS graph subsystem. The C frame is devoted to the graphic representation and takes advantage of the AVS standard representations. Module extract_scalar and generate_colormap are the same as previously described, color_range compute colors in order to modify the curve color according to the incidence value, and color_courbe actually applies the colors. The resulting color changing curve is then given to graph_viewer, an AVS standard module that maps the actual curve along with its x and y axes, legends, etc. The D frame printout visible in Fig. 3D is the result of the print_field module, while the printer module is the same as those used in the spatial application.

**Composite Application: Kriging Representation**

Figure 5 shows a composite application that displays two maps: the disease (here influenza) incidence in each French department at a given date and a spatial filtering of the same data using a method known as the kriging method (5), where the disease incidence is represented as a contour map with colors indicating the incidence range. Figure 6 displays the network producing this
Fig. 4. SITIE network modules of a temporal application. The network represents the modules used to build the application of Fig. 3. Labeled zones delimit the modules corresponding to the same labeled zones described in the legend of Fig. 3. Lines correspond to information flow between modules. The gray level (in color on the actual screen) determines the color-coded data types facilitating coherent interconnections.

This composite application. This application illustrates the reusability concept used in SITIE, since the left part of the application screen (Fig. 5A-E) is exactly the same as that of the spatial application (Fig. 1A-E). Correspondingly, at the network level, the whole frame labeled A–E is also exactly the same as the frames A to E described in the spatial application network and therefore reused. The right-hand part of the screen is produced by the modules grouped in the F frame. Module filter_krig does the kriging computations using the epidemiologic data (a collection of scarce geographic locations using x–y coordinates and their corresponding incidence value) as input and produces a matrix completely filled with the computed incidence in every geographic location as output. From this matrix, intervals_field computes the number of incidence slices, while isoband_slice computes and draws the colored bands and the AVS standard module graph_view draws the isocontour lines. The module masque_to_geom overlays a black mask outside the French borders and seashores in order to make the French territory apparent. The module pave_legends prepares the legend block and Generate_label prepares a label for the entire map. The output of these five modules feeds two AVS standard modules, render_geometry and display_pixmap. The concatenation of these two modules is functionally equivalent to the geometry_viewer module already described. The use of two separate modules allows the storage of multiple maps.
Fig. 5. An example of a complex spatial application graphic user interface using kriging method representation. This application produces two different map representations of the same data (the clinical flu incidence for the 48th week of 1993). The left dashed frame is the same described in the legend to Fig. 2 (zones A–E). The right one (zone F) is an isovalue representation. Frame G includes controls for dynamic display controls for the zone F graphic display.

corresponding to several sequential dates in order to combine them into a movie using standard AVS features, in order to visualize the incidence dynamics through time. In this application, we stress that a part of another application was reused, but other parts were also imported from previous developments. Namely, filter_krig was previously developed and was part of a previous computer program, used by Carrat and Valleron (5) for their work. These functions were encapsulated in another function (a kind of transduction function providing some mandatory parameters and adjusting the existing parameters to the correct format) in order to fit the special SITIE data structure. Thus, SITIE allows the integration of already existing functions implementing specific algorithms rather easily for the knowledgeable C programmer.

Utilization

Redistribution of information is the priority of an efficient surveillance system (6). When completed, SITIE was used to produce, for example, diseases incidence maps and replaced time consuming steps previously needed to produce such maps. The map producing process was about 4–5 hr, involving several computers such as MicroVAX running VMS and Macintoshes running Mathe-
Fig. 6. SITIE network modules of a composite application. The “SQL multi” module is simultaneously connected to “filter krig” and “polyg” modules, in order to produce two representations with only one data base access (compare to Fig. 2).

Mathematica and Pascal programs and needed several Kermit file transfers between these computers. These steps included the database extraction, the kriging process, and the map generation. Using SITIE, the whole process is now done efficiently in one step on one workstation (even if the database is hosted on another computer on the network) in less than 1 min even for the longest
application. Evidently, the machines are much more rapid now, and the architecture of the computer local area network is more efficient. Nevertheless, SITIE takes advantage of these new technologies and allows us to use them transparently.

Currently, SITIE is used daily by our epidemiologist in order to watch for a possible epidemic alert. The flu incidence map is published weekly by two daily MD French newspapers (Impact Médecin and Le Quotidien du Médecin) in epidemic season and when an alert for one of the other notifiable diseases appears. The flu map is also posted weekly on our World Wide Web server (www.b3e.jussieu.fr) together with the map of the preceding week in order to allow comparisons.

Finally, SITIE is also used to visualize the output from models of epidemic dynamics, using the same tools to compare the simulated epidemic as those for the actual one.

**Implementation**

SITIE was developed using AVS V5.01 on a DECstation 5000/240 running Ultrix V4.0. AVS is a proprietary product and needs a license. SITIE can run on all platforms running AVS, almost all Unix platforms, and in the future on PC platforms running MS-Windows. Any Macintosh or PC platform running an X emulator or any X terminal can also access an Unix server running SITIE in a client–server mode. Only a few standard AVS functions were used. Some functions were down-loaded from the public domain AVS user group library. All other functions (122) were specifically developed for SITIE applications in ANSI C, representing approximatively 56,000 lines of C code and the majority of the utilized functions. Seventy-five different applications were developed (23 are accessible using the SITIE menus (see Table 1), the other 52 by loading them directly from files), representing about 45,000 lines of AVS code. Most hard-copy graphic outputs were PostScript files produced using the public domain Plot_PS V5.0 C library, a general purpose graphic C library developed in our laboratory (7).

**Discussion**

The total number of users of FCDN is around 1000, including national, regional, and departmental health officers. As the treatment of data is rather complex, and as many MD users do not have the computer or statistical skills, we developed a project of an interactive workstation which could be as easily used by the "noncomputer scientist–nonstatistician" MD health officer and by those who are computer oriented. For the latter we wanted a system which could help them build their own library of requests to the system. Finally, this project was designed to decentralize the information system. The problem of managing large data collections inevitably leads to the development of interactive tools using a visually interactive and user friendly interface. The FCDN
was faced with the problem of large data collections, leading to the conception and development of SITIE. The basic idea was to build a tool allowing the user to easily select the data available in the database, to easily select a data processing or analysis, and to represent the data either temporal representation or, more interestingly along with their geographic correspondence, spatially. A few geographic information systems have been used as epidemiologic surveillance tools, mainly in epizootia studies, e.g., to study the resurgence of malaria in Israel in 1992 (8), to study the trypanosomia in Africa (9), and some in veterinary fields (10–12). In the domain of more common pathology, a graphical approach was used by Rizzardi et al. (13), who coupled the detailed geographic maps of the United States Bureau of Census to the data of AIDS incidence in San Francisco and lung cancer case location relative to the petrochemical sites in Contra Costa Country, using the S statistical software package. SITIE is a step further since it also integrates the database access and the processing and representation in a user friendly way, including animated spatio temporal representations.

As stressed by Silva and Ball (14), professional workstations are complex systems which must satisfy several needs that are often contradictory. The architecture we developed for SITIE appears to satisfy such needs, since it can be used by several categories of users: from the computer naïve epidemiologist, thanks to a friendly graphic user interface, to the advanced user who will be able to develop new applications, ensuring its evolutionary capability. SITIE provides a rapid access to selected data and several ways to manipulate and display them in various domains, thus easily providing support for decision making. SITIE takes advantage of a modular architecture, allowing its customization, and is based on a commercial product available on a wide variety of Unix running hardwares, ensuring its perenniality.

SITIE appears to be in agreement with the need to integrate both existing (commercial) products and de novo applications, specifically designed for health care purposes (15). In agreement with this statement, when the SITIE project was started, it was decided to use existing tools as the workstation backbone, in order to be efficient. Several systems were explored using both data from the literature (16) and personal experience (17). The large number of graphic outputs necessary for SITIE led to selection of a graphic-based system. AVS was chosen because (i) it is a graphic representation system using a realistic rendering based on high-level three-dimensional graphics and normalized primitives using PHIGS, and (ii) it is a development platform using a tool-box approach, where the basic object (the module) is implicit. But finally, AVS served mainly as an integrating environment, able to link custom modules, and as an image mapper and handler. Other commercial products, or public domain software, such as the Tcl and Tk toolkit (18) could have been chosen to build SITIE, but would have necessitated the development of more codes since they are generally at a lower level than AVS.

The initial idea for developing the epidemiological workstation SITIE was to find powerful solutions on a unique platform for specific needs of the epidemi-
ologists team running the FCDN. We decided to create an architecture based
on the concept of modularity. This concept provided enough flexibility to be
easily adapted and complemented new needs. The future development of SITIE
is contingent on its usage. Most of the work developing the custom modules
was done with care on high modularity. The select_temps.deb_fin is a good
example of a module where the maximum functionalities were implemented, but
not necessarily used. A number of standard applications were developed to
allow the new user to begin working efficiently, using SITIE as it is. However,
when new users become experienced users, they can begin to create new
networks and develop their own applications.

ACKNOWLEDGMENTS

We thank Eyyatar Av-Ron for his helpful comments and help with the English composition and
Wanda Glowinska for having read and corrected our English. We also thank the referees for their
helpful comments, which contributed greatly to improving this paper.

REFERENCES

LEFAUCHEUX, J. A computer network for the surveillance of communicable diseases: The
2. GARNERIN, P H., SAIDJ, Y., AND VALLERON, A-J. The French Communicable Diseases com-
puter Network: A seven-year experiment. In Extended Clinical Consulting by Hospital Com-
3. VALLERON, A-J., AND GARNERIN, P H. Computer networking as a tool for public health surveil-
4. GARNERIN, P H., AND VALLERON, A-J. The French Communicable Diseases computer Network:
5. CARRAT, F., AND VALLERON, A. J. Epidemiologic mapping using the “Kriging” method.
Application to an influenza-like illness epidemic in France. Am. J. Epidemiol. 135(11),
6. THACKER, S. B., CHOI, K., AND BRACHMAN, P. S. Surveillance of infectious diseases. JAMA 249,
7. VIBERT, J.-F., AND BOUSSARD, E. Plot.ps et Plot.xpm: Libraries graphique X et PostScript
1994.
8. KITRON, U., PENER, H., COSTIN, C., ORSHAN, L., GREENBERG, Z., AND SHALOM, U. Geo-
graphic information system in malaria surveillance: Mosquito breeding and imported cases in
9. ROGERS, D. J., AND WILLIAMS, B. G. Monitoring trypanosomiasis in space and time. Parasitol-
ology 106(Suppl), S77-92 (1993).
10. SANSON, R. L., PFEIFFER, D. U., AND MORRIS, R. S. Geographic information systems: Their
11. MALONE, J. B., FEHLER, D. P., LOYACANO, A. F., AND ZUKOWSKI, S. H. Use of LANDSAT MSS
imagery and soil type in a geographic information system to assess site-specific risk of
12. ZUKOWSKI, S. H., WILKERSON, G. W., AND MALONE, J. B., Jr. Fascioliasis in cattle in
Louisiana. II. Development of a system to use soil maps in a geographic information system


