SUPPORTING THE SEARCH FOR THE OPTIMAL LOCATION OF FACILITIES

A. Biancardi*, **R. De Lotto**+, **E. Ferrari***

*Department of Computer Engineering (DIS)
+Department of Civil and Territorial Engineering (DIET)
University of Pavia, 1 Via Ferrata
I-27100 Pavia
Italy
alberto@vision.unipv.it, robiurb@unipv.it, ferrari5@odino.unipv.it

ABSTRACT

Solving the problem of locating services in a given context requires a methematical abstraction so that its complexity can be managed by means of an iterative search through context simulations. Keeping the interface between planners and the model within planners' knowledge domain is achieved by an interactive tool that encourages exploration and comparison among different possible solutions. Additionally workspaces, for managing conveniently multiple parameter-sets, are introduced together with other tools to improve the ability of getting a proper insight of each change to the plan.

Keywords: urban planning, interactive analysis, visual data-managment

1. INTRODUCTION

Solving the problem of locating services in a given context is a complex task. Urban planners have to take into account a large typology of factors, always inter-dependent, ranging from technical and numerical quantities to social and political implications. In all the typical cases of selecting functional changes in the urban context, the planner must be supported by the complete knowledge and management of the parameters to find an optimized solution.

Simulations do help designers find better solutions: by creating a model of the context under study, the attention is focused to the actual changes that each variation induces in the global context. However, given the complexity of the system, the search space will be so extended that an exhaustive search cannot be performed and a closed solution can not be computed. The only way to manage complexity is to exploit the expertise and experience of planners by giving them hints about the underlying inter-dependences; the solution becomes the final point of an iterative procedure where environmental parameters and facility locations are changed according to the planners' need and outcomes are constantly compared.

Even if the underlying model may be deeply abstract, model data refer to the specific context they belong to; hence highly visual interfaces are needed by planners to interact with model data, but unnecessary as far as the model computation is concerned. In this paper ULISSE (Urban Location Interactive System for SErvices) is presented: after an overview of the planning model used by the program, the design issues of its graphical interface and the tool for managing multiple parameter-sets are described.

2. FACILITIES IN CONTEXT

Defining a model for a pragmatic strategy leads to the natural formalization of a cost function and its minimization to reach the sought-after solution [Drezn95]. The cost function can rely, for instance, on the actual cost to build and run the facility together with the costs paid by the facility clients. But while the first type of cost can be reasonably regarded as independent of the site chosen for location, that is, it is not so relevant in the minimization process, the costs paid by the facility users determine a distribution of the potential users on the urban area, significantly affecting the traffic flows through the underlying transportation networks. These costs can be quantified in terms of the time to reach and access the facility in question: however to overcome the limitations imposed by a generic approach, it is necessary to use a model built on the abstraction of the transportation network.

Making reference to a real urban context, the concept of *user* can be naturally replaced with the concept of *vehicle* or *transportation means*, since it is impossible to ignore the presence of an underlying transportation network [Casce90]. According to this basic consideration, the model used in this paper describes the road network as a directed graph, making it possible to evaluate traffic changes due to the access to services. It describes each lane of a road as an oriented link, characterized by time-varying parameters, such as the travel time at given unsaturated conditions. Graph nodes may represent sources or drains of users (population areas or facility locations) and act as road intersections, too. The propagation delays due to the presence of traffic lights and non homogeneous flows can be taken into account in the modeling of links, while the presence of additional services at or nearby a location may be modeled as a *quality index* that is equivalent to a (negative) travel time.

Clearly, in searching the optimal facility location, it is the average access time to be crucial rather than the access time of a single client. This is the reason why a macroscopic continuous-time model turns out to be the correct choice. The graph defined in the previous sub-section can be translated into an electrical network and solved by the program SPICE [Nagel75]. The resulting information defines optimal traffic flows for the given scenario (parameter set) and travel times for every direction (which are meaningful only if an actual vehicle flow exists).

It is importanto to state that no circuit simulation program has a user interface that can be adapted to the urban context — the only option is to translate the graph into a net-list describing the whole graph element by element and totally loosing any graphical information. Graphical information systems (GIS) cannot be of any help either, since the problem lies in the way circuit parameters are managed and related to geographical information.

3. *ULISSE*: A PLANNING TOOL

From the discussion in the previous section it is easy to reckon that in practical planning situations it is the support tools (more than an optimal solution for a too simplified model) that let urban planners keep a meaningful overview throughout the decision process until they reach a satisfying solution. The core task of such tools is to encourage the exploration of as many options as possible and analyse the different outcomes, while managing (unnoticed) all the housekeeping that makes this exploration possible.

The main purpose of ULISSE (Urban Location Interactive System for SErvices) is to make as direct as possible the interaction with context data. To this end three kinds of items are used:

- ❏ a map that gives the graphical foundation for the model graph;
- ❏ source nodes for population or crossroads and facility nodes for services;
- ❏ a number of links that connect existing nodes.

A typical working session would start by selecting a map, adding the necessary user and facility nodes and all the necessary links to describe the traffic network. Of course input data may be changed at any time thanks to contextual menus: nodes and links may be added or removed, parameters of links or nodes may be changed, . . . After entering all the data, the planner can run the simulation, watch the results, change the parameter and go on with the planning loop.

4. VISUAL AIDS FOR DENSE DATA

In addition to the input parameters (at least 1 for the nodes and 2 for each direction of the links), the following information can be extracted from Spice results:

- ❏ for any node, the time to access the nearest facility;
- ❏ the nodes “captured” by a facility, and the corresponding burden in terms of clients (this, in turn, allows one to identify the influence area of each facility delimited, on the nodes map, by the border lines connecting the nodes with associated longer access time);
- ❏ the number of clients reaching each facility, and hence the degree of utilisation of each service;
- ❏ the induced traffic variation in each link of the transportation network;

Even if the number of new parameters per graph-item is limited, when it is added to the number of input parameters it is clear that there is no easy way of displaying four of five numbers for each and every item. Moreover some operations like the visualization of influence areas are incompatible with other kinds of operations and must be locked into their own mode as long as the user wishes.

4.1. Graphical Rooms

Modality is considered a limitation of a program interface: it may prevent users from doing things with no visual clue, leaving a sense of frustration and inability. The problem faced by ULISSE is that in some well characterised cases modality is unavoidable. Starting from the fact that, for instance, we do not expect to be able to perform the same actions we would do in a kitchen if we were in a living-room, it is possible to state that humans are perfectly accustomed to modal behaviours when they move from one room to another. Using a virtual room metaphor, reminiscent of the rooms window manager [Hende86], the concept of graphical rooms (gRooms henceforth) was introduced.

A gRoom is a particular state of the program which is clearly demarked by immediately visible clues, such as the background color of the information area, and by other actions such as the enabling or disabling of related menu items. The main gRoom is the *edit* one and is active by default; only when users are in this gRoom they can edit the graph and alter its parameters. Other gRooms are used for handling influence area analysis and comparison analysis as explained later.

4.2. Extended Layering

One way to manage the spatial density of information is to turn on or off the visualization of some subset: information is still stored, but is not shown. There is no special processing or modality, only a boolean choice. When this choice is grouped over the graph, i.e. the same kind of information throughout the graph, we user the term (visualization) *layer*. Layers are a convenient way to select globally which parameters are displayed and, being orthogonal to gRooms, they can be effectively used when performing the analysis tasks. Additionally they can be used together with ULISSE Postscript output to create highly informative annotated maps.

Layers are effective, but they are limited in the sense that they work globally. By letting the user specify at the item level, *watcher* add a finer control that can be exploited by the user to focus its attention on a selected subset of items as shown in Fig. 1.

5. SCENARIOS MANAGMENT

Exploratory planning requires an active support from the planning tool. The optimal choice may be the final outcome of a complex process that evaluates a huge number of possible solutions with a number of variations both in the graph and in the relative parameter sets. Such big number of alternatives can easily overwhelm the planner and hide away the right path to the solution. The complexity that the planner faces when tackling a new facility-location problem arises from at least three reasons: every interesting parameter set should be saved for future analysis; all the parameter set should be easily managed for classification, retrieval, and comparison of results; parameter set have

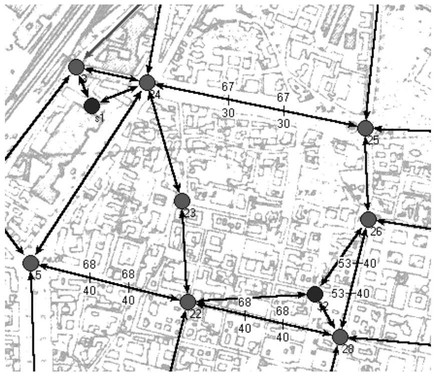

Figure 1: Visualization of numerical values using layers and watchers

to be analysed and compared to evaluate the overall impact of changes in the parameter set.

Prioritizing these data-management requirements, it is clear how analysis and storage, while being difficult *per se*, derive from the central task that eases the handling of all the parameter sets. Switching from one parameter set to another should be almost immediate; without any constraints the whole data set (background map, graph, and parameters) should be retrieved and this would cause a noticeable delay (up to a few seconds), which would stop any user from working on any serious planning problem.

Even comparisons may cause interface difficulties. If the graph is wide and dense, for instance, a change in one part may cause a saturation far off in the graph: if there is no easy way to highlight changes in the computed travel times, the saturation may go unnoticed and the proposed solution will be far from optimal.

To this end the widely-known abstraction of grouping information by a hierarchy of folders was used to create a new tool that can handle the different needs of a planner in a direct way. The word *workspace* has been used to define a named graph together with its parameter set and its results (if computed). A family of workspaces is any set of workspaces that share a common map; a folder (group) of workspaces is any set of workspaces that share a common graph[1]. The formalization of workspaces led to the following constraint: as changing the underlying map means changing problem, one and only one family of workspaces may be worked upon at one time; this avoids the need of updating the map and reduces to negligible the time taken to switch from one workspace to another.

[1] Notice that by this definition workspaces with common graph may be split over a number of folders, but there is no way for workspaces with different graphs to be placed in the same folder!

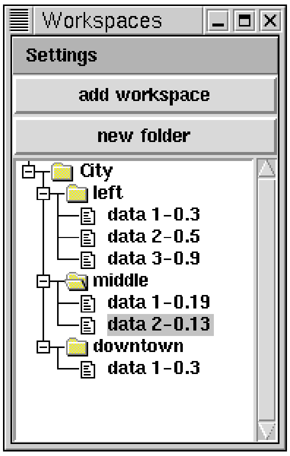

Figure 2: Graphical Management of Workspaces

5.1. Designing Workspaces

The design of a new feature such as workspaces has highlighted a number of peculiar requirements that the ULISSE tool implements:

- provide the user with a nested, hierarchical grouping of parameter sets to prevent problem complexity from making exploration unmanageable;
- use a metaphor for the hierarchy of workspaces, which the user is familiar with as much as possible;
- keep track of many parameter sets without saving to file at each and every change;
- keep track of graph changes so that parameter sets belonging to different graph do not get mixed in the same group (as noted before);
- let the user move and operate easily on the hierarchy of workspaces;
- minimize access and display times for any workspace change;
- let the user name workspaces so that workspace labels may carry any additional semantics that is valuable for the planner;
- let the user save all or part of the workspaces;
- continuous visual feedback for any workspace related operation (changes may occur on invisible layers);

The proposed interface is based on the tree metaphor. Fig. 2 shows a family of workspaces, named *city*, with tree first level groups and some workspaces inside. As explained before the root item cannot be removed; its name, like the names of all the other items, may be changed by the user and is given, by default, the name of the map. Every time the graph is changed the "`Add workspace`" button is disabled, forcing the creation of a new folder if the user wishes to store the workspace. Available operations are handled by contextual menus and depend on the item class. In addition to the usual `Delete`, `Rename`, and `Save` (as a single workspace), the menu entry `Add to save list` allows users to select which workspaces they want to be saved (as part of a family).

5.2. Aiding Analysis

The main task in a *what-if* analysis is finding out the impact of newly made changes by comparing output data; the following step would be to check what caused the result variation by comparing input data. Comparison between two workspaces, hence, is the key operation during the analysis stage.

Workspace comparisons are easily triggered by selecting the other workspace to be compared with the current one. After making selection, the planner is entered the comparison gRoom. All the numerical values may be visualised using the layers and watchers combination; additionally a threshold level may be set by the user so that corresponding items that differ more than the threshold value are highlight automatically.

6. CONCLUSIONS

This paper presented a graphical tool to support the search for the optimal location of facilities. Layered data visualization and a new hierarchical management of parameter sets were implemented to ease the analysis of complex problems with very extended search spaces. Future work will extend the tool to manage multiple transportation networks and to make installation even more straightforward.

ULISSE program is available upon request to the authors, together with information on how to reach our extra-net server for computations or on how to set up a new computing server under the Linux operating system.

REFERENCES

[Casce90] E. Cascetta. *Metodi Quantitativi per la Pianificazione dei Sistemi di Trasporto*. CEDAM, Padova, 1990. (In Italian).

[Drezn95] Z. Drezner. *Facility Location. A Survey of Applications and Methods*. Springer-Verlag, New-York, N.Y., 1995.

[Hende86] A. Henderson and S. K. Card. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics*, 5(3):211–243, 1986.

[Nagel75] L. W. Nagel. Spice: a computer program to simulate semiconductor circuits. Electronics Research Laboratory ERL Memo M520, University of California, Berkeley, 1975.