Design of a view-based DSS for location planning

Citation for published version (APA):

Arentze, T. A., Borgers, A. W. J., & Timmermans, H. J. P. (1996). Design of a view-based DSS for location planning. *International Journal of Geographical Information Systems*, 10(2), 219-236.
<https://doi.org/10.1080/02693799608902075>

DOI:

[10.1080/02693799608902075](https://doi.org/10.1080/02693799608902075)

Document status and date:

Published: 01/01/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Research Article

Design of a view-based DSS for location planning

THEO A. ARENTZE, ALOYS W. J. BORGERES and
HARRY J. P. TIMMERMANS

Eindhoven University of Technology, Urban Planning Group, P O Box 513,
postvak 20, 5600 MB Eindhoven, The Netherlands
email: T.A.Arentze@bwk.tue.nl

(Received 19 October 1993; accepted 26 July 1994)

Abstract. The traditional approach in DSS falls short of providing a highly interactive problem solving environment for planning. Often, cumbersome procedures are required to implement optional plans and obtain feedback information. In dynamic graphic systems, the user is able to view different linked graphic representations (e.g., spatial or statistical graphs) of statistical data and interact (e.g., selecting items) with these graphics. In this paper we describe the design of a DSS for planning facility locations, which uses principles of dynamic graphics to achieve a highly interactive problem solving environment. As in dynamic graphic systems, the user interacts with the DSS through active and linked views. However, where views in dynamic graphics are different representations of a given dataset, the views in the DSS are active data structures describing the facility system to be planned from different perspectives. The declarative and procedural forms of knowledge involved are identified by a logical analysis of planning problems. A frame-based formalism is proposed to represent the knowledge contained in the views. The main advantage of this view-based approach is that it offers the user a highly flexible and interactive environment for performing 'what-if' analyses.

1. Introduction

Both local governments and commercial firms (e.g., retailers) continuously monitor developments in a market area which affect existing relations between the demand and supply of urban facilities. Location strategies or plans are developed when disequilibria or opportunities for expanding the firm (facility chain) arise. In planning facility locations often several, possibly, conflicting goals are pursued. Generally, the primary goals of both public and private planners are economically healthy facilities and coverage of consumer demands (e.g., Arentze *et al.* 1993a, Davies 1984, Van der Heijden *et al.* 1984, Wrigley 1988). Several types of uncertainties are typically involved in such planning problems. First, the decision maker is often uncertain about the formulation of goals and their relative weights (e.g., Armstrong *et al.* 1991, Densham 1991). Second, in many cases uncertainties exist on the occurrence and nature of developments on the supply side (e.g., opening of competing outlets) or on the demand side (e.g., changing consumer preferences), which affect the performance of the system but are outside the control of the decision maker (e.g., Davies 1984, Van der Heijden *et al.* 1984). Third, the decision maker is often unable to assess with certainty the consequences of optional actions (e.g., entering a new

market) or possible developments for the performance of the system. Because of these uncertainties facility planning is typically an ill-structured problem.

Spatial Decision Support Systems (SDSS) are explicitly designed to help decision makers solve ill-structured spatial problems by providing a flexible and interactive problem solving environment (Densham 1991). Such environments provide facilities for implementing optional actions, reformulating goals and evaluating the effects of these changes on system performance. These kinds of 'what-if' analyses help the decision-maker to understand the problem and to assess possible trade offs between conflicting objectives. In this paper we describe the design of an SDSS for 'what-if' analyses; the intention is to introduce a new approach aimed at improving the interactive properties of SDSS.

2. Existing approaches

GIS technology offers a potentially valuable basis for developing spatial SDSS (Clarke 1990, Fedra and Reitsma 1990). Several SDSS for facility planning (e.g., Kohsaka 1993, Amer *et al.* 1993, Lin *et al.* 1993, Grothe and Scholten 1992, Armstrong *et al.* 1990, 1991), which use standard GIS functions for database management, display and interface purposes, are described in the literature. However, as Owen (1993) has argued, traditional geoprocessing systems fall short of providing the highly interactive environment needed for effectively supporting 'what-if' analyses. In the traditional approach, the user specifies a dataset and activates functions operating on this dataset. Often, a combination of functions must be activated in a series of steps to generate the information needed. 'What-if' analyses are supported by these *procedural* systems, but they require for each simulation cumbersome procedures of editing data to implement the optional action and activating the appropriate combination of functions to assess the consequences. The time and effort involved in closing the feedback-loop (the turnaround test) may have the effect that trade-off relations in the system to be planned remain obscured (Keen 1983). Another drawback of this approach is that analytic or procedural knowledge is required to use the system, which may be a bottleneck in the accessibility or acceptance of these systems.

To improve the interactive properties of GIS, a new system technology has been introduced, which is based on the principle of dynamic data definitions (e.g., Owen 1993, Haar 1993, Wessels 1993, Clementini *et al.* 1993). In dynamic systems, attributes of spatial objects are typically defined in a dynamic way in terms of procedures for calculating attribute values. If a predefined condition is met, procedures attached to attributes are triggered to update their value. For example, consider 'distance to nearest facility centre' as an attribute of residential zones. Rather than assigning constant distance values, this attribute may be defined by a procedure for calculating nearest-centre distances. This procedure can be made to trigger when facility locations are changed. Owen (1993) shows that systems based on this dynamic data model overcomes the shortcoming of procedural, static systems; the objects of an application update themselves automatically as a response to users' actions. In the example, the nearest-centre-distance attribute of residential zones is updated each time the user relocates facility centres. Thus, the user is able to obtain immediate feedback without the need to initiate procedures. The dynamic principle corresponds to the automatic re-calculation of spreadsheet data as the user enters new values.

Recently, a dynamic approach has also been introduced in the field of graphic representations of statistical data to improve the interactive properties of graphics

(e.g., Beshers and Feiner 1993, Cleveland and McGill 1988, Haslett *et al.* 1990). Dynamic graphic systems allow users not only to view the data graphically, but also to interact with these graphs. Users can have different views (e.g., geographical or statistical graphs) of the data. These views are active in the sense that the user can select and highlight items or subsets of items in each view. The views are linked so that selecting items in one view causes the selection of corresponding cases in all other views. Thus, graphic views are active interfaces to the data, which allow the user to interactively explore the data.

The development of interactive DSS for planning has also received attention in the field of Operational Research (OR) and management science. In OR applications computer-based models are used to generate the plan that optimises a specified objective function. In the interactive modelling approach (Anthonisse *et al.* 1988, Dror *et al.* 1991, Hurion 1986) systems are developed that allow users to specify a model of some problem situation and to interact with the model to explore alternative decision scenarios. Densham and Armstrong (1993) describe the data structures and user interface of a spatial DSS that allows users to interact with spatial models through active and linked representations of the problem. Users are free to specify model coefficients, but the model structure (e.g., a linear programme) is fixed. Therefore, to use the system, problems must meet a certain format, which limits the flexibility of the system. The approach we suggest aims to provide a more flexible DSS environment, where the user is to a larger extent free to determine the model structure and to interact with the model.

In the approach we propose, the principles of dynamic data definitions and linked views are combined to develop a flexible and highly interactive SDSS for planning. In our approach views are not different represents of data, as in dynamic graphics, but instead different descriptions (subsets of data) of the facility system to be planned. Each view describes the facility system from a certain perspective, for example, the present state, anticipated future state or goal state of the facility system. The user is able to specify a standard set of linked views in terms of variables relevant for a specific planning problem and attached procedures for evaluation. To investigate optional actions, possible developments, goals or methods, the user changes variable values or procedure specifications in the views specified. Variables are dynamically defined such that dependent views are automatically updated in response to users' actions.

The concept of views introduced in this paper is similar to the generally used concept of views in Database Management Systems (DBMS) where views are specific cross sections of a database (Armstrong and Densham 1990), which present the information queried by the user in a way independent of the physical aspects of information storage and processing. The purpose of these views is to provide an interface to the data that insulates users from the specifics of data storage. However, in contrast to views in DBMS, views in the context of this paper are active and linked data structures. The approach we suggest integrates the principle of dynamic data definition with conventional data views.

In the next section we describe a model of planning problems to derive the set of relevant views and their logical interrelationships. Next, we discuss in general terms the knowledge required for solving planning problems, which a SDSS should incorporate, and subsequently, the characteristics of the decision making process, which is supported by a view-based DSS, is discussed. In the sections that follow, we describe the basic approach and implementation of a view-based DSS, and we

consider a retail planning problem to illustrate the way the system can be used. Finally, we discuss some conclusions and directions for future research.

3. A model of planning

In developing a new facility network or reorganizing an existing facility network commercial firms or local governments are faced with the problem of developing a location strategy or location plan. In this section we introduce a conceptual model of such planning problems. The model describes the elements of planning problems and their logical interrelations.

In the planning model the present state of the system under investigation and the goals of the decision maker are given. The set of goals defines the ideal or goal state of the system at some future moment in time. The system is continuously changing in time due to factors not under control of the decision maker and to actions undertaken by the decision maker. From the perspective of the decision-maker, the factors beyond his or her control are exogenously given and are, therefore, considered as autonomous developments. The decision-maker predicts the future state of the system under conditions of planned actions and anticipated autonomous developments, for comparison with the goal state. The discrepancy between the predicted state and the goal state indicates the effectiveness of planned actions.

The planning problem is defined as one of generating the set of actions that minimises (or reduces to an acceptable level) the discrepancy between the future state (given anticipated developments) and the goal state. The planning model is schematically shown in figure 1. The boxes in the scheme represent descriptions of

Figure 1. Basic scheme of planning tasks.

the system and the arcs denote dependency relations between descriptions. For example, the future state of the system is a function of the present state, autonomous changes and planned actions. The dashed arc connecting the Action to the Discrepancy indicates that actions do not logically follow from discrepancies, but rather involve a choice among alternative options. If autonomous developments and actions are absent, the Future State equals the Present State. Then, comparing the Future State with the Goal State means evaluating the performance of the present state of the system. Therefore, the scheme also covers the problem of monitoring the system.

In most planning problems the present and goal state of the system are described in terms of different sets of variables. In spatial planning, the present state of the system is typically described in terms of locational, topological and thematic variables (Armstrong and Densham 1990), say X -variables, and the goal state in terms of performance criteria (Arentze *et al.* 1993b), say C -variables, where $C=f(X)$. Then, the future state described in X -variables must be evaluated in C -variables before comparing it with the goal state. Often, decision makers do not have the means to directly manipulate the system state (in X -variables). Instead, they must rely on measures (instruments) for *influencing* developments. Then, actions are defined in terms of instrument variables, say I -variables, where $X=f(I)$.

The introduction of different sets of variables to define actions (I) and goals (C) complicates the planning problem, because an action in I does not directly follow from a discrepancy in C . Generally, different actions in I are possible for reducing a given discrepancy in C . First, because a given goal in C can often be realised in more than one way by changes in X and, second, because a given change in X can be accomplished by various actions in I . Consequently, the generation of actions is a divergent and tentative process.

At least in the Dutch context, governmental planners typically split a complex planning problem into two phases. The first phase involves planning actions in X (rather than in I), to arrive at a desirable future state of the system. Then, in the second, the problem reduces to finding actions in I that realise this future state. So, the plan developed in the first phase is operationalised in the second phase. Both phases are themselves planning tasks and can be described by the planning model (figure 1). The first phase is described by the model where actions are defined in X -variables, as shown in the lower part of figure 2. This task is less complicated than the original task, since the extra step of selecting the appropriate instruments to accomplish the changes in X , needed to reduce the discrepancy, is omitted. The second phase fits in the model where the future state in X (designed in the first phase) serves as the goal state, as shown in the upper part of figure 2. Now, the desirable changes in X follow directly from the discrepancy (in X), rather than being abduced from the discrepancy in C .

We emphasize that the planning model does not describe the *process* of planning, but rather the declarative components involved and their logical interrelationships. The model defines the set of views that is generally relevant to planning problems.

4. Knowledge in planning

The procedural and declarative forms of knowledge required to solve planning problems can be identified based on the scheme of figure 2. In this section, we will summarise these forms of knowledge, to indicate in general terms the contents of the knowledge base of an effective DSS.

Figure 2. Scheme of planning tasks: the development of the Future State (lower part) and the planning of actions in *I* (upper part).

Table 1 summarises the declarative forms of knowledge that describe the system from different perspectives (the boxes in figure 2). Tables 2 and 3 list the procedural forms of knowledge required to specify and evaluate *X*, *C* and *I*-planning variables. Procedural knowledge may take the form of mathematical models or solution strategies developed in the spatial sciences (Arentze *et al.* 1993c) or operational research. For example, we can think of spatial interaction models or discrete choice models for assessing the impact of actions, population forecasting models or statistical projection techniques for predicting autonomous developments, performance measurement methods for evaluating the future system state and combinatorial algorithms for generating optional actions. Alternatively, procedural knowledge may take the

Table 1. The declarative knowledge involved in planning tasks, as derived from the planning model shown in figure 2.

Declarative knowledge
Autonomous developments
Present state of the system in <i>X</i> -variables
Goal state of the system in <i>C</i> -variables
Future state in <i>X</i> -variables
Future state in <i>C</i> -variables
Discrepancy in <i>C</i> -variables
Actions in <i>X</i> -variables
Actions in <i>I</i> -variables

Table 2. Forms of procedural knowledge required for the identification of planning variables.

Task	Knowledge
Which <i>X</i> -variables are required to evaluate the given set of <i>C</i> -variables?	Inferential knowledge to identify causal factors (<i>X</i> -variables); Analytic (methodological) knowledge to identify relevant analytic models and their data needs;
Which <i>I</i> -variables are relevant (available) to manipulate the then given set of <i>X</i> -variables?	Instrumental knowledge to identify relevant instruments and conditions for their use.

Table 3. Forms of procedural knowledge required for the evaluation of planning variables.

Task	Knowledge
The evaluation of <i>C</i> -variables based on <i>X</i> -variables; Determining discrepancies in terms of <i>C</i> -variables;	Conceptual knowledge or, more specifically, methods for performance analysis and evaluation;
Generating an (optional) action in <i>X</i> -variables given discrepancies in <i>C</i> -variables;	Heuristic knowledge, or more specifically, methods for plan generation;
Generating an (optional) action in <i>I</i> -variables given discrepancies in <i>X</i> -variables;	Instrumental knowledge or, more specifically, methods for instrumentalisation;
Inferencing the impact of autonomous factors on the system state in <i>X</i> -variables; Inferencing the impact of an action in <i>I</i> -variables on the system state in <i>X</i> -variables.	Inferential knowledge or, more specifically, methods for forecasting and impact assessment.

form of logic-based models (e.g., sets of if-then rules). These qualitative models can be used for judging system performance (evaluating *C*-variables) and for the selection of actions in *X* or *I*, through knowledge intensive reasoning. Of course mixed models comprising both quantitative and qualitative knowledge may also be included.

The declarative and procedural knowledge components summarised in tables 1, 2 and 3 *describe* the planning problem. Higher level knowledge is required for using this descriptive knowledge to solve planning problems (Breuker and Wielinga 1989). First, analytic knowledge is needed to specify and select models for solving subproblems. Second, task knowledge is needed to apply these models for solving specific classes of problems. In the context of facility location planning, task knowledge may take the form of algorithms developed in the field of operational research for solving location-allocation problems. In terms of the planning model (figure 2), location-allocation algorithms (e.g., the interchange algorithm) generate the future state in *X*-variables (the solution), given the goal of optimising a single *C*-variable (objective function) and a heuristic for generating optional actions (e.g., substitutions) within given constraints. Therefore, these optimisation algorithms fit in the model as an extra knowledge layer which controls the planning process. In turn, algorithms at this level may be controlled by knowledge at an even higher level. This higher level knowledge would prescribe or suggest ways of using algorithms, e.g., varying specifications of the objective function to generate interesting plan alternatives.

In sum, to support both the specification of the problem (the set of views) and the problem solving process (controlled by either the user or an algorithm), a SDSS must incorporate or accommodate the descriptive, analytic and task forms of knowledge mentioned above.

5. The decision-making process

The DSS described in this paper is designed to support decision-making through generating and evaluating optional actions in an iterative process. In terms of the scheme of figure 2, the decision-maker goes repeatedly through cycles of (1) initiating or adjusting an action based on observed discrepancies between the anticipated and goal state of the system, and (2) comparing the changed future state with the goal state. The immediate feedback of changes in decision variables (X or I -variables) on goal variables (C -variables) helps the decision-maker to acquire insight into the problem. Governed by this insight the process is likely to converge into a final plan.

Alternatively, the decision maker may wish to simulate various autonomous developments (keeping actions constant), to explore possible future states of the system. A prediction analysis consists of cycles of (1) initiating or adjusting autonomous factors and (2) comparing the changed future state with the goal state. Thus, the consequences of various equally valid assumptions regarding autonomous changes are investigated.

The decision-making process may be embedded in the prediction analysis, resulting in an action plan for each possible autonomous development. Thus, the decision-maker is able to respond adequately to possible developments. Also, the reverse strategy may be followed, where the prediction analysis is embedded in the decision-making process. Then, each optional action is evaluated across a series of possible autonomous developments, allowing the decision-maker to investigate the strategic properties of alternative decisions. For example, options can be adjusted to minimise the maximum discrepancy between future state and goal state (that is maximise the performance under the worst condition).

In many cases the decision-maker is not able to exactly specify the goal state of the system. Goals and plans often develop simultaneously (Armstrong *et al.* 1991). In terms of the scheme: the decision-maker varies C -variables defining the goal state and initiates actions in X -variables for each goal setting. A subjective evaluation of the future states in X -variables (configurations of the facility network) under different goal settings (e.g., an accessibility standard) indicates the validity of goal settings. This process helps the decision maker to articulate goals (and at the same time to generate actions).

If uncertainties exist regarding methods for evaluating dependent variables, the decision maker may wish to simulate the effects of different method specifications. For example, different methods can be used to forecast the consequences of assumed autonomous developments or to assess the impact of proposed actions. This kind of simulation gives an indication of the sensitivity of outcomes to method specifications. If different, equally valid method specifications give substantially different outcomes, then it makes sense to embed the sensitivity analysis in the decision-making process. This strategy allows the decision-maker to assess the robustness of optional action plans for arbitrary variation in method specifications.

6. The view-based approach

The user interacts with the data and models of the DSS through views. Views are frame-like structures that describe the system to be planned from different

perspectives. The DSS provides a standard set of linked views corresponding to the scheme of figure 2. The user first defines an application by specifying the set of views dependent on the given problem. Then, the user interacts with the specified set of views to solve the problem. Different users may be involved in the specification phase and application phase. In the specification phase, the user customises the generic DSS to a specific problem. The intended users of the generic DSS (the specification phase) are researchers or consultancy agencies who are experts in the field of location analysis. The target users of specific DSS (the application phase) are agencies preparing plans for policy-makers in a political context or managers in an organizational context. In this section we describe the structure, specification and application of views, respectively.

6.1. *The structure of views*

Figure 3 shows the general structure of views. Each view contains slots for the specification of variables describing the system from that perspective (e.g., Future State, Action and so on). A variable usually takes the form of a vector of scores that describes some characteristic of the objects (e.g., residential zones or facilities) of the system. A slot has fields for storing the name, value and evaluation procedure of the variable concerned. For example, the Future State in *C* contains the set of *C*-variables which describes the future state and stores for each variable the name, the vector of object-specific scores, and the specification of a performance measurement method.

Views are active data structures in the sense that variable values are dynamically defined by means of slot procedures. Furthermore, views are linked in the sense that changes of variable values (e.g., closing a retail outlet) in one view (Action) automatically leads to changes in all linked views (e.g., the turnover of existing outlets in the Future State) by triggering slot procedures (e.g., a consumer choice model). Therefore, views are comparable to spreadsheets.

6.2. *Supporting the specification phase*

In the specification phase, the user is able to specify a set of views (an application) either by opening and editing an existing application or by defining a new one. In the latter case, the system internally creates the structure of views according to the planning scheme in figure 2.

A view is specified in terms of relevant variables and attached evaluation

view type		
application name		
variables		
- name1 - name2 - ...	- value1 - value2 - ...	- procedure1 - procedure2 - ...

Figure 3. The structure of views.

procedures. The specification of variables is supported by presenting view-specific lists of variable dimensions (e.g., performance criteria) for which evaluation procedures (e.g., performance measurement methods) are available in the modelbase of the system. Based on analytic knowledge, the system supports this phase by suggesting variables which might be relevant, given the planning goals specified and the available data and model resources.

To specify attached procedures, the user activates view-specific dialogues. In a dialogue the user (1) selects a function from a list of options, (2) sets values of the function parameters, if any, and (3) specifies the data needed to evaluate the function. Data are specified by selecting slots of linked views or, in the case of independent views (the Present State, Goal State or Autonomous Developments), by importing or editing data. For example, to define procedures for evaluating *X*-variables that describe the future state of the system, the user (1) selects an item from a list of optional forecasting methods, (2) sets values of parameters involved, and (3) selects variables from the Present-State, Action and Autonomous-Developments view. Based on analytic knowledge the system suggests optional procedure specifications (functions, parameter values and data) for evaluating the variable concerned, dependent on available data and model resources.

A view can be thought of as a program for analysing the system with respect to a certain perspective. Views can be stored separately so that the user is able to build for each perspective a base of views (prototypes). In specifying an application the user can select a view from this base and adjust the settings to the specifics of that application. The system supports this option by providing a standard base of prototypical views. Views can also be stored as a complete set. A set of views can be seen as a computer-based model of a planning problem. Users may wish to use a model specified for analysing a number of cases or case variants. Just as at the level of separate views, the user is able to develop a base of models by storing specified sets of views.

An application can be seen as a specific DSS and, consequently, the system for supporting the specification phase can be thought of as a DSS generator (Sprague 1980). This concept of DSS generator is comparable to some dynamic graphic systems which allow users to define views on statistical data (Beshers and Feiner 1993).

6.3. *Supporting the application phase*

In the application phase the user interacts with a set of views specified for solving a specific planning problem. In this phase the system provides functions for (1) editing variable values or procedure specifications within views (e.g., adjusting an optional action by changing variable values in the Action view) and (2) displaying variable values (e.g., discrepancy score vectors). Different ways of representing data can be used for both editing and displaying the data contained in views. The most suitable form will depend on the data type, problem type and preferences of the user (Densham 1991). Therefore, the application phase is optimally supported when different optional representations of data (views on data) are provided. Then, the user can choose to edit or inspect data represented on a map, or in various graphic or tabular formats. A high-interaction environment for exploring data would be obtained by using active and linked graphics (the dynamic graphics approach).

In many spatial problems map images are an appropriate medium for user interactions. When this medium is used, views take the form of thematic maps that

display the variables of interest. Actions like relocating facility centres are realised by moving entities in map space. Attribute variables are edited in dialogue boxes that pop up as users click on a map entity. An interface like this allows users to inspect and manipulate data both easily and intuitively.

6.4. Reporting analysis results

The application phase results in a set of fully specified views. This set not only describes the analysis results (variable values) of a set of views, but also the analysis method (attached procedures) used. Therefore, an application comprises the information needed for a full report of analysis results. So, in the view-based approach generating a report simply involves making hard copies of the set of views concerned.

7. The representation of knowledge

The former section described the semantic and functional properties of views. This section focuses on a method of knowledge representation for implementing these views. The implementation method used is not a key issue in the view-based approach. Obviously, various implementation methods can be used. In this section, we will briefly outline a method that could be used. The method is based on principles of organising knowledge (data and procedures) derived from the hybrid frame-rule formalism as described in Parsaye and Chignell (1988) and Pracht (1990).

The declarative (variables) and procedural (slot procedures) elements of views are organised in frame structures (see figure 4). A frame consists of a local database and a series of slots. The database contains (pointers to) the data needed to determine slot values. Since all data are stored in frames, the database consists of pointers to other frames. For example, the database of the frame representing the future state in *X* contains pointers to the Autonomous-Development, Action and Present-State frame of the same application. A frame slot has fields for the name and value of a view variable and, furthermore, a field for the (specification of the) attached procedure.

A simple inference engine manages the frame processing. The engine executes attached procedures when (1) the slot value concerned is queried while it is unknown

Figure 4. The proposed structure of frames.

and (2) elements of the local database are changed. Since procedures query needed slot values, the first trigger condition may cause a backward chaining of procedures. For example, a procedure for evaluating a Discrepancy variable queries needed Future State *C*-values. If these values are unknown, then the attached procedures are triggered (executed). In the same way these procedures may cause the triggering of impact-assessment procedures for evaluating Future State *X*-variables and so on. Eventually, the evolving chain of procedures will generate the needed discrepancy value (and all intermediate values). Furthermore, since procedures may change attached slot values, the second trigger condition may cause forward chaining of procedures. For example, a change in the value of an Action variable triggers impact-assessment procedures for evaluating Future State *X*-variables. If these procedures result in new slot values then procedures for evaluating Future State *C*-variables are triggered. Similarly, these procedures may trigger procedures for evaluating Discrepancy *C*-variables and so on. In sum, the *feed-backward* mechanism manages data retrieval processes, so that information can be queried at any time and at any level. The *feed-forward* mechanism manages data updating, so that the data in the set of views remain up-to-date across data manipulations.

Finally, a change in the specification of a procedure causes the slot value concerned to be reset. If resetting implies altering the slot value (i.e., if the slot value had been evaluated), then through the feed-forward mechanism all forwardly-linked frames are re-evaluated. Re-evaluation implies querying backwardly-linked frames. Therefore, due to the feed-backward mechanism the slot value that was reset is also re-evaluated. Through the combined workings of the feed-backward and feed-forward mechanisms, the data in frames remain consistent across changes of procedure specifications.

In sum, frames of the type described above are active data structures. Slot values are dynamically defined by the attached procedures and the elements in the internal database. Frames are linked in the sense that slots of one frame are input to other frames. The links are realised by the frame databases, which contain pointers to other frames. Procedures activate each other to obtain needed data and to assess the effects of generated results. Consequently, data in the system remains up-to-date and internally consistent. The set of linked frames that define an application can be seen as an active database.

The representation method described in this section is an object-oriented method, in the sense that declarative and procedural forms of knowledge are organised in computational entities (frames or objects). However, the frames described here differ in some formal and semantic aspects from objects in most object-oriented systems. First, in frames (of this type) all procedural elements take the form of functions that return a value (slot value), whereas object procedures ('methods') encode more generally for the behaviour of objects. Furthermore, the procedures in these frames are not capable of directly activating each other, e.g., by means of sending messages, as in objects. Instead, procedures activate each other through querying or changing data items. Finally, there is an important semantic difference. Frames (in this context) represent certain perspectives or aspects of the system, whereas objects typically stand for entities of the system. So, in the view-based approach knowledge about an entity (e.g., its location, performance, discrepancy) is distributed over all frames, whereas the knowledge about an aspect (e.g., the performance of all entities) of the system is localised in a single frame. The opposite holds in most object-oriented systems. In a view-based DSS, the view-oriented way of organising knowledge is

efficient, since the information needed relates to aspects of the system (e.g., system performance) rather than to individual entities (e.g., the attributes of a specific outlet).

8. Illustration of the approach

In this section we will discuss a hypothetical example to illustrate the way a view-based DSS can be used. We consider a governmental agency who is concerned with the provision of daily shopping facilities to a population in an urban area. The example reflects the main concerns and standards of local governments in retail planning in the Dutch context. The primary criterion the retail system must meet is that all people are able to reach a neighbourhood centre within walking distance (i.e., approximately 750m). Individual centres should be economically viable to ensure service coverage in the longer term. So, this economic performance criterion is not an end in itself, but is derived from the primary coverage-criterion. The problem is complicated since the coverage and economical criterion are in conflict. Generally, adding centres improves the coverage of demand, but, at the same time, decreases the economic performance of centres.

The present functioning of the retail system is evaluated in response to observed nationwide trends that threaten the viability of the small-scale neighbourhood centres. An increase in mobility and changes in consumer preferences are the main factors that have caused a shift from daily shopping in nearby centres towards weekly bulk-shopping in large-scale centres. When no actions are undertaken, these trends will probably lead to a drop out of the smaller and more vulnerable centres. The gaps in the network of centres that arise as a consequence will disadvantage less mobile consumers. If these anticipated problems appear to occur in this specific case, then corrective actions can be initiated. The government does not have the means to open, close or relocate centres nor to determine centre attributes (e.g., floorspace). However, through financial support, provision of public services and zoning regulations, the government is able to influence the development of the system.

Table 4 shows the contents of the set of views that describes the problem situation. The goal pursued by the planner is a network (spatial pattern) of viable neighbourhood centres that cover all demand locations within walking distance. The viability of centres is evaluated by determining the floor productivity, that is the retail turnover per square foot of floorspace, and comparing the result with a normative planning standard. The coverage of demand locations is evaluated by comparing the distance to the nearest centre with a pre-specified critical distance (walking distance). Therefore, the relevant *C*-variables are floor productivity (a vector of centre related scores) and nearest-centre distance (a vector of consumer related scores). In real-world problems of this type additional *C*-variables will be considered, such as the total travel distance, the completeness of the branch mix present in centres, the proximity of centres to other centres, and the quality of centres. However, these two *C*-variables correspond to the most important planner's goals and suffice to illustrate the view-based approach.

The *X*-variables include the location, attribute and topological data that are needed to evaluate these *C*-variables. The *X*-variables are subject to changes caused by either autonomous developments or planner's actions. The planner anticipates changes in sizes of neighbourhood populations (an *X*-variable) caused by future mutations in the supply of houses. In the context of this problem, these mutations are considered autonomous developments. Developments with respect to the supply side are (indirectly) controlled by the planner. The optional actions in *X*-variables

Table 4. Variables and attached procedures of the views of an example application.

Variable	Attached procedure
Goal state	
normative floor productivity	
normative nearest-centre dist.	
Present state in <i>X</i>	
neighbourhood locations	
neighbourhood attributes	
neighbourhood-centre locations	
neighbourhood-centre attributes	
distances	
Future state in <i>X</i>	
consumer locations	
consumer attributes	impact_assessment(Developments, Present State)
neighbourhood-centre locations	impact_assessment(Action, Present State)
neighbourhood-centre attributes	impact_assessment(Action, Present State)
distances	
Future state in <i>C</i>	
floor productivity by centre	consumer_choice_model(Future State in <i>X</i>)
nearest-centre distance by consumer locations	measurement(Future State in <i>X</i>)
Discrepancy	
discrepancy floor production by centre	difference(Future State in <i>X</i> , Goal State)
discrepancy nearest-centre distance by consumer location	difference(Future State in <i>X</i> , Goal State)
consumer location	
Autonomous developments	
mutations house supply by neighbourhood	
Actions in <i>X</i>	
centre relocation	
centre closure	
centre opening	
centre attributes	

include the relocation, opening and closure of centres and the specification of centre attributes. Having determined the desirable future state in *X*-variables (the configuration of the network), the planner will generate actions in *I* (e.g., financial support), to operationalise the plan.

The data regarding the Present State, Goal State and Autonomous Developments are exogenously given. Consequently, the evaluation procedures attached to the variables of these views consist of retrieving data from a specified database. Generally, the *X*-variables of the Future State are evaluated by assessing the impact of Autonomous Developments (mutations in housing supply) and Actions in *X* (e.g., relocating centres) on the Present State. A consumer-choice model is used to predict the floor productivity of centres (a *C*-variable) of the Future State. The parameters of the model are set to values that reflect the changed consumer preferences. The discrepancy variables are defined simply as the difference between measured and normative values of *C*-variables. Actions are initiated by the decision maker, rather than by pre-specified heuristic (knowledge-based) procedures, since the problem of designing the network is considered too complex to be formalised.

The set of views is used to evaluate the present functioning of the network and, if problems are detected, to reorganise the network by initiating actions. To evaluate the Present State, the user sets Autonomous and Action-based changes to zero and queries the Discrepancy variables. The floor-productivity and nearest-centre-distance discrepancies are represented on a thematic map of the area. Poorly performing centres and badly covered consumer locations are identified, localised and highlighted. The user activates the Future State (= Present State) to query supplementary attribute data of the highlighted centres possibly represented in tables. Then, based on discrepancy and attribute information, the user specifies actions in the Action view. Actions are first specified in terms of X -variables to determine the desirable future state in X -variables. Relocating, deleting and creating centres are realised through a map interface. A table editor is activated to change centre attributes. The effects of the actions are inspected by querying the Discrepancy view. Going through these cycles of editing the network and querying discrepancies will reveal local trade offs between the viability of centres and the coverage of demand. Based on this insight, compromise solutions may be found and complementary actions may be planned to compensate negative effects.

To test the robustness of possible action plans, the planner investigates the effects of different assumptions regarding possible developments, trends or goals for each plan. First, the planner simulates possible trends in consumer preferences (e.g., the relative weights of distance or price) by varying parameter values in the consumer choice model. Second, the planner investigates the effects of different mutations in housing supply on the floor productivity of centres by editing the Autonomous Developments view. Finally, the effectiveness of each plan is evaluated under various critical levels of floor productivity and nearest-centre distance by editing the Goal State view. The Discrepancy view is consulted each time a change is made and the plan (set of actions) that proves to be effective and robust for arbitrary specifications is selected.

The resulting design of the network (Future State in X -variables) serves as the goal state in a similar procedure for planning actions in terms of instrument variables (implementable measures). The set of views that result is used as a report of the planning results. The views show the designed Future State (under Actions in X), the expected Future State (under Actions in I), the Goal State and the corresponding anticipated and planned developments and discrepancies. The attached procedures inform the decision maker on the methods used to obtain the results.

9. Conclusions and discussion

The DSS described in this paper supports facility planning as an iterative process of generating and evaluating optional plans. A flexible and highly interactive problem solving environment is realised by views that represent the system to be planned from different perspectives. The set of views that are generally relevant are identified by a logical analysis of planning problems. The user is able to define an application by specifying variables and attached procedures relevant to the given problem. The DSS incorporates analytic knowledge to support the specification of views. The user is able to simulate optional actions, possible developments, goal settings or method specifications by changing variable or procedure settings within views. Views are active and linked so that changes in one view automatically lead to updating all linked views. The application phase can be supported by using dynamic graphic

techniques. The DSS design introduced in this paper is best viewed as an attempt to apply the principles of dynamic graphics for the development of an effective SDSS.

The view-based approach has some features in common with the interactive modelling approach, which has emerged from the field of Operational Research (OR) applications (Anthonisse *et al.* 1988, Dror *et al.* 1991, Hurion 1986). In both approaches, the analyst builds a computer-based model of some problem situation and the decision maker uses the model to investigate alternative decision scenarios by manipulating data or parameters of the model and inspecting the effects on model outcomes. In OR applications, however, the model has a fixed structure and is restricted to solving optimisation problems (e.g., linear programming problems) of a certain format. In a view-based system, in contrast, the user is to some extent free to determine the model structure (set of views) by selecting combinations of procedures. Second, in OR applications the user typically varies specifications of the objective function (the goal variable) and inspects the corresponding solutions (actions) generated by the model, whereas in view-based systems the user typically initiates actions (solutions) and inspects the impacts on goal variables assessed by the model.

An essential feature of the view-based approach is the separation of decision support in a specification and application phase. In the first phase, the system is best viewed as a DSS generator, which allows the user to generate a problem specific DSS. In this phase, analytic knowledge is required to specify variables and attached procedures, given the characteristics of the problem and available data and model resources. The extent to which the system is capable of supporting the specification phase depends on the degree to which it incorporates the analytic knowledge required. At the one extreme, the system has extensive knowledge for completely specifying the set of views (analysis framework), given specifications of the present state and goal state of the system to be planned. At the other extreme, the system offers only an interface to the modelbase and the user selects appropriate models (procedures). In a useful variant in between these extremes, the system has an extensive base of pre-specified sets of views and incorporates analytic knowledge for selecting from this base the set of views that best fits a given problem specification. Then, the user would only have to adjust the set of views selected so that the system meets specific information needs.

There is no analytic knowledge required to use the DSS in the application phase. The user can concentrate on the planning process, without the need to specify and initiate procedures to obtain information. The automatic triggering of procedures increases the speed of feedback and, therefore, improves the effectiveness of decision support. The set of views makes up an active database, which guards the integrity and consistency of the database across data manipulations. The same analysis framework (set of views) can be applied to different cases or case-variants. Consequently, the consistency in methodology used is maintained across cases (Owen 1993).

Finally, dynamic-graphics techniques fit well in a view-based DSS. Both approaches can be integrated in a single system. Then, in the specification phase, the user is able to specify graphic views and link them to system views. In the application phase, the user would interact with graphic views of different perspectives (views) of the system. Highlighting objects in one view would cause the highlighting of the same objects in all linked views. This integrated approach would result in a highly flexible, interactive and graphic environment for planning.

Acknowledgment

This research is supported by the Technology Foundation (STW).

References

- AMER, S., SLIZIUZAS, R. and SUN, Y., 1993, Spatial decision support systems for urban planning and management. In *Proceedings of the Fourth European Conference on Geographical Information Systems, EGIS '93* (Utrecht: EGIS Foundation) pp. 116–124.
- ANTHONISSE, J. M., LENSTRA, J. K., and SAVELSBERGH, M. W. P., 1988, Behind the screen: DSS from an OR point of view. *Decision Support Systems*, **4**, 413–419.
- ARENTZE, T. A., BORGERS, A. W. J., and TIMMERMANS, H. J. P., 1993a, Geographical information systems and the evaluation of the location of facilities. In *Proceedings of the Fourth European Conference on Geographical Information Systems, EGIS '93* (Utrecht: EGIS Foundation), pp. 962–972.
- ARENTZE, T. A., BORGERS, A. W. J. and TIMMERMANS, H. J. P., 1993b, GIS and the evaluation of the performance of facility networks: the content and organization of input and output data. In *Proceedings of the Third International Conference on Computers in Urban Planning and Urban Management*, (Atlanta, Georgia: Georgia Institute of Technology), pp. 27–50.
- ARENTZE, T. A., BORGERS, A. W. J., and TIMMERMANS, H. J. P., 1993c, Integrating GIS into the planning process. *Paper presented at the GIS and Spatial Analysis Meeting*, ESF-GB data programme Amsterdam, The Netherlands 1993.
- ARMSTRONG, M. P., and DENSHAM, P. J., 1990, Database organization strategies for spatial decision support systems. *International Journal of Geographical Information Systems*, **4**, 3–20.
- ARMSTRONG, M. P., DE, S., DENSHAM, P. J., LOLONIS, P., RUSHTON, G., and TEWARI, V. K., 1990, A knowledge-based approach for supporting locational decision making. *Environment and Planning B: Planning and Design*, **17**, 341–364.
- ARMSTRONG, M. P., RUSHTON, G., HOENEY, R., DALZIEL, B. T., LOLONIS, P., DE, S. and DENSHAM, P. J., 1991, Decision support for regionalization: a spatial decision support system for regionalizing service delivery systems. *Computer, Environment and Urban Systems*, **15**, 37–53.
- BESHES, C., and FEINER, S., 1993, Auto-visual: Rule-based design of interactive multivariate visualizations. *I.E.E.E. Computer Graphics & Applications*, **12**, 41–49.
- BREUKER, J. A., and WIELINGA, B. J., 1989, Models of expertise in knowledge acquisition. In *Topics in Expert System design* edited by G. Guida and C. Tasso (New York: North-Holland), pp. 265–295.
- CLARKE, M., 1990, Geographical information systems and model based analysis: towards effective decision support systems. In *GIS for Urban and Regional Planning* edited by H.J. Scholten, and J.C.H. Stillwell (Dordrecht: Kluwer Academic Publishers), pp. 165–175.
- CLEMENTINI, E., DI FELICE, P., and TORLONE, G., 1993, Experimenting the usability of an O-ODBMS for the representation of topology in GISs. In *Proceedings of the Fourth European Conference on Geographical Information Systems, (EGIS '93)* (Utrecht: EGIS Foundation), pp. 1421–1430.
- CLEVELAND, W. S., and MCGILL, M. E., (editors), 1988, *Dynamic Graphics for Statistics* (California: Wadsworth & Brooks/Cole).
- DAVIES, R. L., 1984, *Retail and Commercial Planning* (London: Croom Helm).
- DENSHAM, P. J., 1991, Spatial decision support systems. In *Geographical Information Systems: Principles* edited by D.J. Maguire, M.F. Goodchild and D.W. Rhind (New York: John Wiley & Sons), pp. 403–412.
- DENSHAM, P. J., and ARMSTRONG, M. P., 1993, Supporting visual interactive locational analysis using multiple abstracted topological structures. In *Proceedings of AutoCarto 11* (American Congress on Surveying and Mapping, Bethesda, MD), pp. 12–22.
- DROR, M., SHOVAL, P., and YELLIN, A., 1991, Multiobjective linear programming: Another DSS. *Decision Support Systems*, **7**, 221–232.
- FEDRA, K., and REITSMA, R. F., 1990, Decision support and geographical information systems.

- In *Geographic Information Systems for Urban and Regional Planning* edited by H.J. Scholten and J.C.H. Stillwell (Dordrecht: Kluwer), pp. 177–188.
- GROTHE, M., and SCHOLTEN, H. J., 1992, Modelling catchment areas, towards the development of spatial decision support systems for facility location planning. In *Proceedings of the Third European Conference on Geographical Information Systems*, EGIS '92 (Utrecht: EGIS Foundation), pp. 978–987.
- HAAR, TER, P. W., 1993, Dynamic analysis for dynamic problems. In *Proceedings of the Fourth European Conference on Geographical Information Systems*, (EGIS '93) (Utrecht: EGIS Foundation), pp. 746–753.
- HASLETT, J., WILLS, G., and UNWIN, A., 1990, Spider—an interactive statistical tool for the analysis of spatially distributed data. *International Journal of Geographical Information Systems*, **4**, 285–96.
- HEIJDEN VAN DER, R. E. C. M., TIMMERMANS, H. J. P., and BORGERS, A. W. J., 1984, Measuring and evaluating the impacts of retail plans: A unified multifaceted approach. In *Planologische Diskussiebijdragen* (Delft: Delftsche Uitgevers Maatschappij), pp. 315–330.
- HURRIION, R. D., 1986, Visual interactive modelling. *European Journal of Operational Research*, **23**, 281–287.
- KEEN, P. G. W., 1983, Interactive computer systems for managers: a modest proposal. In *Decision Support Systems* edited by W.C. House (New York: Petrocelli), pp. 167–188.
- KOHSAKA, H., 1993, A monitoring and locational decision support system for retail activity. *Environment and Planning A*, **25**, 197–211.
- LIN, C., HUANG, C., and CHEN, A., (1993) A GIS-based decision support system for emergency service delivery. In *Proceedings of the Third International Conference on Computers in Urban Planning and Urban Management*, (Atlanta Georgia Institute of Technology), pp. 325–337.
- OWEN, P. K., 1993, Dynamic function triggers in an on-line topology environment. In *Proceedings of the Fourth European Conference on Geographical Information Systems* (Utrecht: EGIS Foundation), pp. 1249–1256.
- PARSAYE, K., and CHIGNELL, H., 1988, *Expert systems for experts* (New York: John Wiley & Sons).
- PRACHT, W. E., 1990, Model visualization: Graphical support for DSS problem structuring and knowledge organization. *Decision Support Systems*, **6**, 13–27.
- SPRAGUE, R. H., 1980, A framework for the development of decision support systems. *Management Information Sciences Quarterly*, **4**, 1–26.
- WESSELS, C., 1993, Object-orientation and GIS. In *Proceedings of the Fourth European Conference on Geographical Information Systems*, (EGIS '93) (Utrecht: EGIS Foundation, Faculty of Geographical Sciences), pp. 1462–1471.
- WRIGLEY, N., (editor), 1988, *Store choice, store location and market analysis*. (London: Routledge).