in Proceedings of HCI'94 "People and Computers IX" The University of Glasgow, 23-26 August 1994), G. Cockton, S.W. Draper, G.R.S. Weir (Eds.), Cambridge University Press, 1994, pp. 125-143.

A Comparison of Placement Strategies for Effective Visual Design

Jean Vanderdonckt

Institut d'Informatique, Facultés Universitaires Notre-Dame de la Paix, Rue Grandgagnage, 21, B-5000 Namur, Belgium.
Tel : +32 (0)81-72.49.75 - Fax : +32 (0)81-72.49.67
Telex : 59.222 FacNamB
Email : jvanderdonckt@info.fundp.ac.be

Missiri Ouedraogo

Ministère de la Fonction Publique et de la Modernisation de l'Administration, 03 BP 7006, Ouagadougou 03, Burkina Faso.
Tel: +226 - 30.22.86

Banta Ygueitengar

c/o D. Ngarboldoum, Meridian Biad, B.P. 87 N'djamena, Republic of Tchad.
Tel: +235 - 51.43.14 - Fax: +235 - 51.23.45 - Telex: 5228 Afr Bk

The development of graphical user interfaces for interactive applications is subject to a series of well-known problems which could be relevant of the domain of visual design. This typically includes the problem of placing aesthetically interaction objects (IO) according to principles applied in placement strategies. This paper first reviews the problem of IO placement and shows the rationale for the most significant placement strategies found today. It then tries to compare six such strategies along several dimensions and mathematical relationships with respect to three points of view : the designer's point of view, the human factors expert's point of view, and the user's point of view.

Keywords: Arrangement, Dimensioning, Interaction Object, Localisation, Interaction Object Placement, Placement Strategy, Visual Design.

1. Introduction

The problem of placement concerns the spatial position of interaction objects (IO) such as edit box, radio button, list box,... with respect to visual elements in a screen layout. Effective placement can be achieved through a particular *placement strategy* involving a certain amount of visual techniques such as proximity, alignment, separate reference, centring, and conformity. For instance, Galitz suggests a placement strategy where IOs should be placed according to their nature : "All elements on a screen should be located in a unique and consistent position. These elements are : title, screen identifier, screen body (including caption, data, section headings, completion aids, prompting), status or instructional messages, error messages, command field or area" (Galitz,1992). In this paper, the placement is defined as the description of a composite IO (e.g. a form container widget) for displaying IOs to be placed

relatively to one another. Composite IOs generally allow IOs to resize themselves as the composite resizes. IOs can be placed in relative positions either by percentage or by absolute distances. Every IO is univocally determined on its top, bottom, left and right sides.

1.1. How to Define a Placement Strategy?

On one side, there is a need for knowing how to present data, graphics, images in general (e.g. on paper). This knowledge is generally expressed in terms of geographical, physical or spatial principles such as adjacency (Galitz, 1992), proximity (Galitz, 1992; Kim, 1990), similarity (Kim, 1990), proportion (Kim, 1990; Marcus, 1992), format (Galitz, 1992; Lauer, 1990, Marcus, 1990), page layout grid (Kim, 1990, Marcus, 1992; Tarlin, 1990; Tufte, 1983), symmetry (Galitz, 1992; Lauer, 1990), ordering (Lauer, 1990),... Basics design principles are found extensively in (Galitz, 1992; Lauer, 1990; Marcus, 1992; Tufte, 1983). A summary of visual techniques that could be applied is reported in (Vanderdonckt, 1994).

On the other side, it is necessary to interpret these principles (which are independent of the domain of user interfaces) into rules or guidelines to be followed. It is not so easy to determine a realistic set of rules from these principles in order to work with properly (Kim, 1990). However, it could be done by translating some of these principles into a comprehensive placement strategy. In order to characterise this translation, quantifiable dimensions and relationships are needed.

1.2. Dimensions and Relationships in Placement Strategies

The placement is decomposed into three parts :

1. the *localisation* is interested by logically positioning IOs in the container. It covers position, alignment, justification. Localisation could be mainly achieved through
   - consistency, e.g. the position of IO should be compatible with the users' conventions, should be consistent in format;
   - sequentiality, e.g. most frequently used IOs should be located first;
   - screen image, e.g. IOs should be equally located in all four quadrants of the container.
2. the *dimensioning* goes in for the uniformisation and standardisation of IO dimensions. It deals for instance with the length of abbreviations, the maximum number of characters per label, the length of an edit box, the item number in a list box, the harmonisation between length and height for a dialog box.
3. the *arrangement* takes into account the IO orientation and constraints related to the logical ordering of IOs. Arrangement should be logical as much as possible (e.g. by preference, by consensus, by physical property, by data flow), should emphasize visual cues, should care about aesthetics and should reduce ocular movements and screen density.

About 300 guidelines related to these dimensions have been gathered in a *corpus ergonomicus* (Bodart, 1994a). It is also sound to define mathematical relationships to characterise the practicability, the workability and the applicability of visual principles into the three above dimensions. These relationships will be helpful to compare different placement strategies. In the following list of relationships, the abbreviation "iff" denotes "if and only if":

- *Left justification*: two IOs are left justified iff their left abscissa are identical.
- *Right justification*: two IOs are right justified iff their right abscissa are identical.
- *Upper justification*: two IOs are upper justified iff their left ordinates are identical.
- *Bottom justification*: two IOs are bottom justified iff their right ordinates are identical.

- *Horizontal centring*: two IOs are horizontally centred iff the ordinates of their centres are identical.
- *Vertical centring*: two IOs are vertically centred iff the abscissa of their centres are identical.
- *Horizontal uniformity*: two IOs are horizontally uniformised iff their lengths are identical.
- *Vertical uniformity*: two IOs are vertically uniformised iff their heights are identical.
- *Horizontal equilibrium*: three IOs are horizontally equilibrated iff the horizontal inter-object distances are identical.
- *Vertical equilibrium*: three IOs are vertically equilibrated iff the vertical inter-object distances are identical.
- *Diagonal equilibrium*: three IOs are diagonally equilibrated iff their centres are distributed on a same line.
- *Proportional equilibrium*: three IOs are proportionally equilibrated iff they are uniform and equilibrated either vertically or horizontally.
- *Total equilibrium*: three IOs are totally equilibrated iff they are equilibrated either vertically or horizontally, and uniform vertically and horizontally.

For example, the dialog box reproduced in fig. 1 shows four rows of IOs that are equilibrated. Labels are left justified, vertically uniform and vertically equilibrated, therefore proportionally equilibrated. Drop-down list boxes are right and left justified, vertically and horizontally uniform, vertically equilibrated, therefore totally equilibrated. Pushbuttons (Ok), (Cancel), and (Help) are totally equilibrated in order to avoid different probabilities of selection which namely depends on the pushbutton dimensions. However, these pushbuttons are not vertically centred.

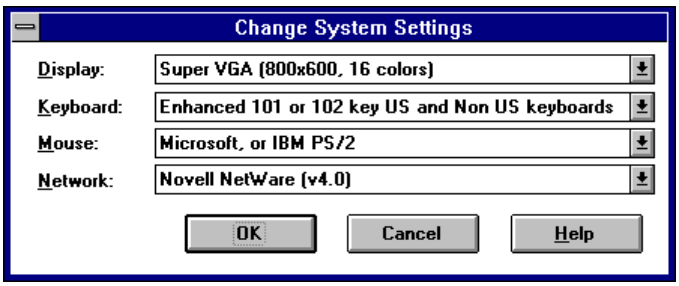

Figure 1. A dialog box with proportionally equilibrated labels and total equilibrated drop-down list boxes

The characterisation of these relationships is not really new since the eleven first relationships are provided as spatial manual arrangement options by most structured graphics editors and graphical user interface builders. But what is more interesting is to try to incorporate these relationships in placement strategies as a way of quantifiable measurement.

2. Manual Placement Strategies

Placement strategies that should be manually applied are now considered in order to highlight the differences existing between such manual strategies and systematic strategies that could be included in computer-aided or automated tools for placing IOs.

Manual placement strategies are performed when a designer is free to place IOs in the composite IO. This situation arises with interface builders (e.g. interface toolkit editors) where IOs could be created, placed and modified according to the designer's point of view.

Such strategies also include placement by demonstration : Peridot (Myers, 1988) attempts to establish left, right, upper or bottom justification of IOs by using global rules defined by demonstration. Druid (Singh, 1990) and Excel's Dialog Editor (Microsoft, 1992) both place IOs according to the implied current design situation. If the designer creates a first label, it is placed automatically with some margins. If an edit box should be added, it will be automatically placed on the right of the label with some extra space, assuming that the previously entered label identifies the edit box.

When the designer creates another label after the previous one, the tool guesses that the new label and the previous one should be left justified. The problem with this technique is that, if the designer fails to follow a predefined traditional sequence, a bad IO placement may result. For instance, if the designer creates a group box, an edit box and a label, the last label will be placed beneath the edit box (fig.2). Moreover, these tools may require a lot of physical manipulations rather than logical ones. But, once these manipulations have been recorded, they can be replayed as many times as necessary. Gilt (Hashimoto, 1992) solves the manipulation problem by introducing a *graphical tab*, which is an absolute placement position, and a gr*aphical style* incorporating property and placement attributes. Redefining an existing graphical style modifies all IOs placed with that style.

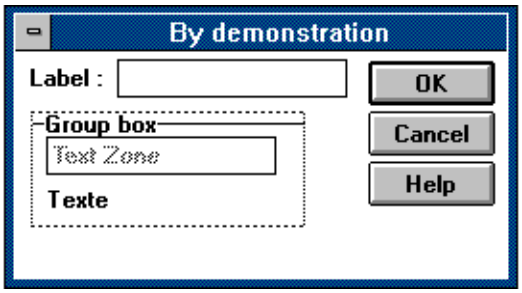

Figure 2. Placement by demonstration

3. Computer-Aided and Automated Placement Strategies

As we can see, demonstrational strategies really become operational once all necessary visual techniques have been exemplified, thus requiring significant manipulation at the beginning. It stands to reason that the price of genericity may be considered here as the cost of this demonstration task. Thus, placement strategies providing a higher level of assistance are to be investigated.

Model-based tools, like Humanoïd (Szekely, 1993), are design tools where user interfaces are described by giving a model of their presentation (including the placement of IOs) and their behaviour. Humanoïd provides a modelling language allowing designers to express how IOs should be manipulated. In particular, the "Presentation" slot of the templates defines how IOs have to be laid out according to their nature. The three dimensions and all the mathematical relationships can therefore be theoretically preserved if they are included in the model. The biggest advantage of this approach is that the placement strategy is completely described and formalised in the model. Modifying the placement strategy in the model automatically regenerates the new placement of IOs, allowing the designer to work at a design level rather than at a physical level. This is why Humanoïd could be considered as a computer-aided placement strategy. If such tools say with which mean the placement of IOs can be formalised, specified, they do not necessarily say which mathematical relationships and which graphical principles should be so formalised. Of course, not all problems can be equally settled at the same time.

Tarlin (Tarlin, 1990) recalls us that modern word processors support user-defined documents through

templates called *style sheets* which is a variant of underlying layout grid for structured documents. These templates specify various aspects of a document page such as its size, section header appearance, body text appearance, and the number of columns on the page. Why do we not reuse style sheets for placing IOs in dialog boxes? This is what Tarlin did when he completely visually redesigned "Print" and "Options" dialog boxes for X-Windows (fig. 3). Dotted lines represent the vertical and horizontal lines for defining a layout grid.

The layout grid technique has been widely applied in computer-aided and automated placement strategies, but with different extents and interpretations. Such strategies are the one-column strategy applied in UIDE (de Baar, 1992), the one or two-column strategy in GENIUS (Janssen, 1993), the balanced two-column strategy in TRIDENT (Bodart, 1994b), the shape strategy in DON (Kim, 1993), the Layout Appropriateness (Sears, 1993), and the Right/Bottom strategy in TRIDENT (Bodart, 1994b).

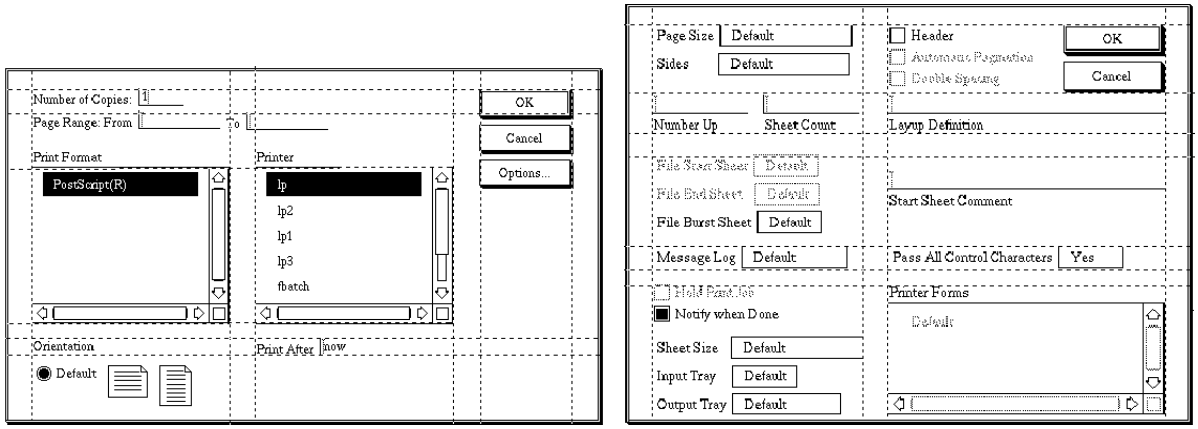

Figure 3. Layout grids of "Print" and "Options" dialog boxes

3.1. Purposes and Limits of the Comparison

Rather than analysing all currently available tools for placing IOs, the comparison will focus on the six mentioned placement strategies according to three points of view and an example to see how they can be performed.

Other placement strategies that could be investigated so far include Scope (Beshers, 1989), UofA* (Singh, 1991), Jade (Vander Zanden, 1990) and ITS (Wiecha, 1990).

3.1.1. Points of Comparison

Each placement strategy will be first described by the strategy type (column-oriented, shape-oriented, frequency-oriented, continuity-oriented), a summary of the main ideas of the strategy, its aim, the underlying assumptions and rationales if relevant.

The designer's point of view includes the adequacy of the generated user interface with respect to placement dimensions and mathematical relationships identified in sub-section 1.2, the automation level (computer-aided or full automatic), the flexibility (capability to adapt the placement on the fly), the tailorability (ability to tailor a given placement according to a personal conventional scheme), and the level of assistance and control provided by the strategy and/or the tool implementing the strategy.

The human factor expert's point of view includes the adequacy of the generated user interface with respect to visual techniques. These visual techniques namely encompass

- screen density: how unused blank spaces are optimised;
- fragmentation: how IOs are embedded, overlapping;
- regularity: how regular is the layout grid;
- consistency: how consistent are the placed IO across multiples cases.

The user's point of view is characterised by a subjective satisfaction, the visual appealing and the compatibility with the user's level of experience.

3.1.2. Example of a Placement Problem

Each strategy will be applied on a small example taken in the field of a hospital admission application. The hierarchy of predefined IOs is depicted in fig. 4 with the appropriate information ordering to be respected. DBX denotes a dialog box, GBX denotes a group box, EBX denotes an edit box, RBX denotes a radio box, SLB denotes a scrollable list box, MSG denotes a message area and PBT denotes a pushbutton.

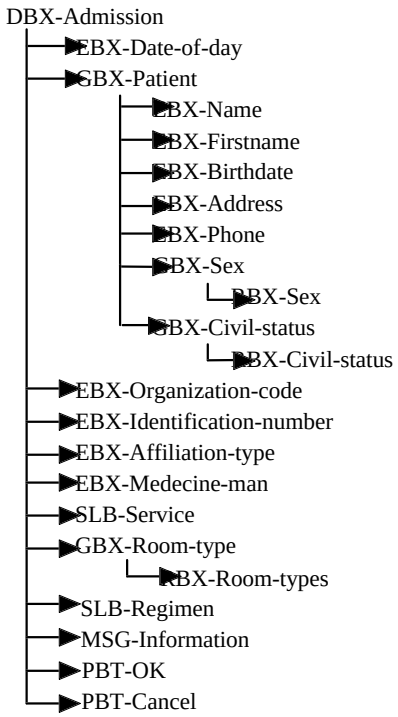

Figure 4. Hierarchy of the Interaction objects to be placed in the example

3.2. Placement in UIDE

UIDE (de Baar, 1992) includes rules for mapping data objects belonging to the application design to IO belonging to the user interface design. Two main steps are emphasised : the selection of IO and their layout. UIDE's rules automatically lay out selected IOs in a parent window in order to create consistent layout which are compliant with a set of guidelines. These embedded guidelines constitute the first incursion of visual principles for producing a layout. A virtual grid is also used to insure proper IO

justification, separation and adequate window dimensions. No particular assumption is made for applying this strategy. The placement strategy mainly rests on a visual design with six principles :

1. vertical placement of IOs one after another in a row,
2. right justification of labels,
3. left justification of IOs,
4. regular space between labels and prompts, between prompts and IOs,
5. centring of standard push buttons,
6. normalised margins for all the four edges.

From the designer's point of view, since the placement strategy is straightforward, the generation of dialog boxes is very easy and rapid (fig. 5). From the human factors expert's point of view, the localisation and the dimensioning of IOs in a dialog box are very simple and the arrangement is more or less non existent since all IOs are vertically placed on the same level, therefore avoiding any visual structure, grouping, and continuity. However, left and right justification, proportional equilibrium implies a very clean, easy to read and to follow layout which is intrinsically kept consistent. If the number of IOs within a dialog box increases, the sizes of the composite IO may become huge or too large to be managed into one separate composite IO. A lot of screen space is left unused. From the user's point of view, the best and cleanest placement does not necessarily mean the easiest to use.

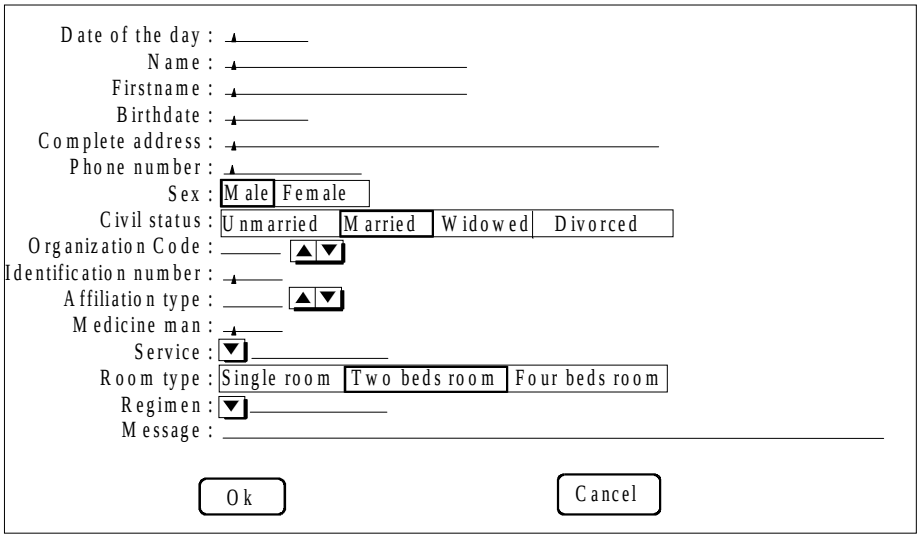

Figure 5. The example resulting from UIDE

3.3. Placement in GENIUS

GENIUS (Janssen, 1993) works similarly to UIDE except that

1. the placement is governed by values input in property sheets (e.g. the data type, range, and condition values);
2. the placement is determined first for each IO (or IO group like a radio button with four items) separately, and, second, group by group;
3. the IO sequencing follows the ordering of attributes specified in entities and relationships.

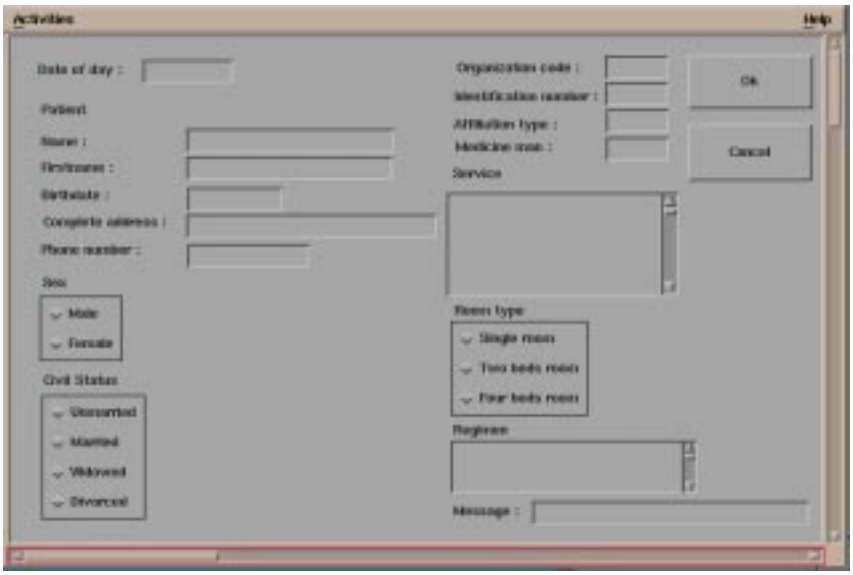

Figure 6. The example resulting from GENIUS

No particular assumption is made for applying this placement strategy. It supports five principles :

1. vertical placement of IOs into one or two columns,
2. left justification of labels,
3. left justification of IOs,
4. vertical uniformity and equilibrium between IOs and groups of IOs,
5. total equilibration of the pushbuttons.

Once again, from the designer's point of view, the resulting placement is fairly straightforward since all the IOs flow into one or two vertical columns. The rationale behind is that reading and IO scanning are faster if they are vertical rather than horizontal. The message is processed like every other IO, not like a message. Flexibility is not supported, but the placement is provided instantly (fig. 6).

From the humans factors expert's point of view, the spaces between IOs in each column are constant. Therefore, the composite IO is very vertical. Thus, a lot of blank space is used and visual groups (e.g. the Patient group) are not visually isolated.

This system seems to work efficiently for generating dialog boxes with a moderate number of IOs (typically, not more than 20). What will happen if the number of IOs increases? First, this can potentially lead to a vertically-arranged dialog box where placement is purely sequential, that is no appropriate placement according to the user's task. Second, the dialog box could grow like an awful stack of IOs.

3.4. Placement with Two-Column Strategy

The rationale behind the two-column strategy is to equally distribute IOs into two columns, but with different vertical equilibration (Bodart, 1994a). This strategy is therefore a bit more general than GENIUS's strategy. In fact, proportional and total equilibrations are considered as the most complete and aesthetic spatial relationships. These relationships are also the most difficult to apply for the two columns since different types of IOs are involved. The summarised steps of the two-column strategy (Bodart, 1994a) are the following :

1. the characterisation of the IO sequence: the contents of all IOs are identified (identification and descriptive labels, prompt and field);
2. the fixation of visual parameters: recommended standard space between two lines or rows, between label and prompt, prompt and field, font height;
3. the computation of standard dimensions of the IOs: all IOs are first pushed in a vertical stack whose dimensions are calculated recursively as the sum of the heights of individual IOs;
4. the computation of the dimensions of the two columns: the above stack is bipartitioned by finding regular proportions between two balanced columns; IOs are distributed in each column with respective dimensions and vertical uniformity/equilibrium if possible;
5. the computation of the internal proportion: after adding title, message area and separators if necessary, the current proportion is computed;
6. the adding of standard and custom pushbuttons, drawn buttons, and icons (if any). The goal is to choose whether these buttons will be arranged in column on the right or in a row at the bottom of the composite IO. These buttons are automatically dimensioned with total equilibrium;
7. the selection of appropriate margins: margins result from an optimisation problem based on the internal proportion which is forced to converge to one of the proportion recommended by Marcus (Marcus, 1992).

The layout grid to be reached by the two-column strategy looks like the one represented in fig. 7.

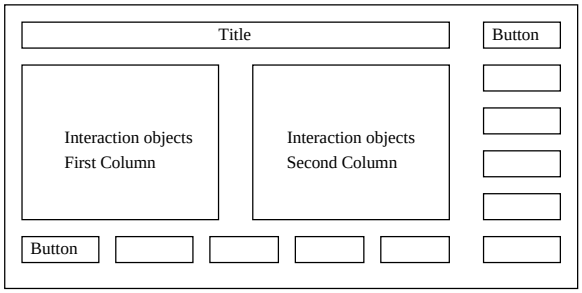

Figure 7. The layout grid in the two-column strategy

This two-column based strategy produced the dialog box of fig. 8. This strategy is now analysed :

- *flexibility* : the strategy provides a tree with physically arranged IOs and supports the information ordering since the IO ordering reflects the information ordering. This strategy is rigid since it proposes a single version of the future dialog box according to a constant layout;
- *button placement* : the strategy always places buttons either horizontally at the bottom of the dialog box or vertically on the right of the dialog box. Though always consistent and visually isolated, these placements require a lot of space;
- *column placement* : the strategy attempts to share out IOs equally in the two columns to preserve balance and repartition. Thus, if the number of IOs is high, a visually appealing dialog box might result; if the number of IOs is very low, a flat unappealing dialog box might be viewed to the point that horizontality becomes irrelevant;
- *proportion* : the strategy uses a convergence to a recommended proportion. This could resize global dimensions if the current dialog box has bad dimensions, but this could potentially lead to an increase of the global dimensions. But the dimensions are large regardless the unused greyed regions showed in fig. 8.
- *justification* : the strategy automatically establishes all possible justifications increasing the feeling of unity;

- *grouping* : the strategy lays out IOs in each group recursively, thus preserving the grouping principle and the initial IO sequence.

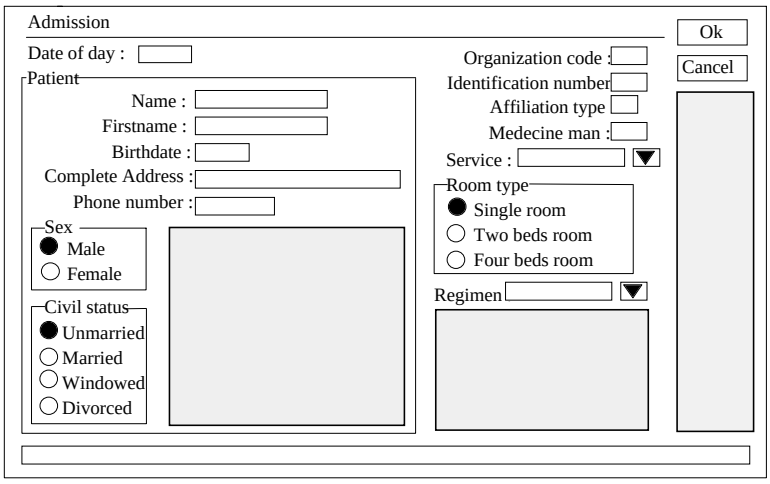

Figure 8. The example resulting from the two-column strategy

On the other hand, this two-column strategy does not necessarily generalise particularly well to either the one-column or the multi-column case (as in a newspaper). This approach produces consistent layout but does not consistently produce visually appealing layouts. The two-column strategy lacks both generality and reusability in other contexts. This strategy have been so defined to conform to a predefined grid so that changing the number of columns, as recommended by Tarlin (Tarlin, 1990), modifying the title or pushbutton location is not supported. The last goal for this strategy was to build a narrow, specific tool for form-filling applications, and therefore more sophisticated than a large general tool where precision and consistency could be endangered.

3.5. Placement in DON

DON is more devoted to the layout generation and evaluation (Kim, 1990, 1993). All selected IOs subject to a dialog box are collected in a pool. At each step, the dimensions of IOs are examined two by two. IOs are laid out according to the longer, smaller or similar dimensions leading to groups of progressively decreasing sizes.

We now analyse both strategies on different point of views (fig. 9) :

- *flexibility* : the strategy provides a tree with physically arranged IOs, but could generate an inconsistent IO ordering since the information ordering and/or structure is not taken into account. DON is not necessarily rigid since it suggests numerous placement versions;
- *forms and dimensions analysis* : only DON deals with IO dimensions. This generates a more aesthetic interface by minimising the surface of remainder areas, but logically related IOs can be separated if their dimensions are not similar;
- *overflow solution* : in the two-column strategy, this problem is supposed to be solved in the step of IO selection. In DON however, overflow solutions are performed only if necessary during the placement. If IO dimensions and margins cannot be reduced enough to accommodate the IOs, DON is forced to go back to the previous step of selecting IOs;

- *button placement* : DON's strategy is the most parametrisable and flexible. Moreover, buttons are placed wherever remainder areas are left;
- *column placement* : DON generates a placement where IOs are sorted by decreasing dimensions.

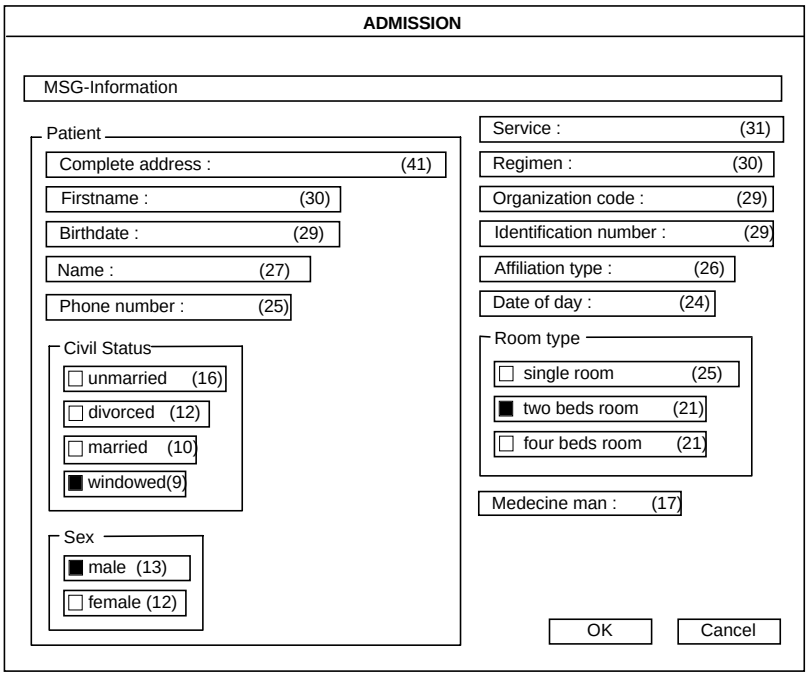

Figure 9. The example resulting from DON

On one hand, the designer benefits from improved layouts with minimised blank spaces and multiple possible layouts to choose from ; on the other hand, the designer looses the information structure since the generation is only shape-based. DON takes the visual IO organisation much more into account : balance, symmetry, space usage and density, size limits and ratio are immediately supported and evaluated. DON automatically computes some aesthetic measures. It gives the designer complete feedback for deciding if a possible layout receives good scores for these measures. Combining automatic generation and evaluation of layouts have been proved very useful for providing the designer relevant assistance (Kim, 1993). One of the great feature of DON is that the designer can choose a particular placement at his/her convenience if necessary and act with the editor at his/her convenience.

As we saw, DON's strategy is very "visual design" oriented, since only the shapes and dimensions of the IOs are taken into account for the placement. The limits of such a visual strategy could be highlighted : one can wonder to what extent (i.e. beyond conventional form-filling interaction styles) it makes sense to optimise the placement of IOs without considering the conversation on the user interface. For example, the conversation states, coding techniques, activation and deactivation of IO by the user or the interactive application, dynamic transitions between IOs could be considered. We will see in sub-section 3.6 how dynamic task aspects such as the transitions between IOs may influence the arrangement of IOs.

3.6. Placement with Layout Appropriateness

Sears developed a Layout Appropriateness (LA) method that can take advantage of a simple task description rather than a complete one (Sears, 1993). This metric is able to evaluate a layout, given different parameters detailing how users perform their tasks in the layout. These parameters integrate : the set of IOs, the sequences of performed actions, and their frequency. The LA metric is able to serve as

basic measure for computer-aided searching of a best layout, called *LA-optimal*. LA metric remains the only one in its category. We will see now how the sequences of performed actions and their frequency may have an effect upon the placement.

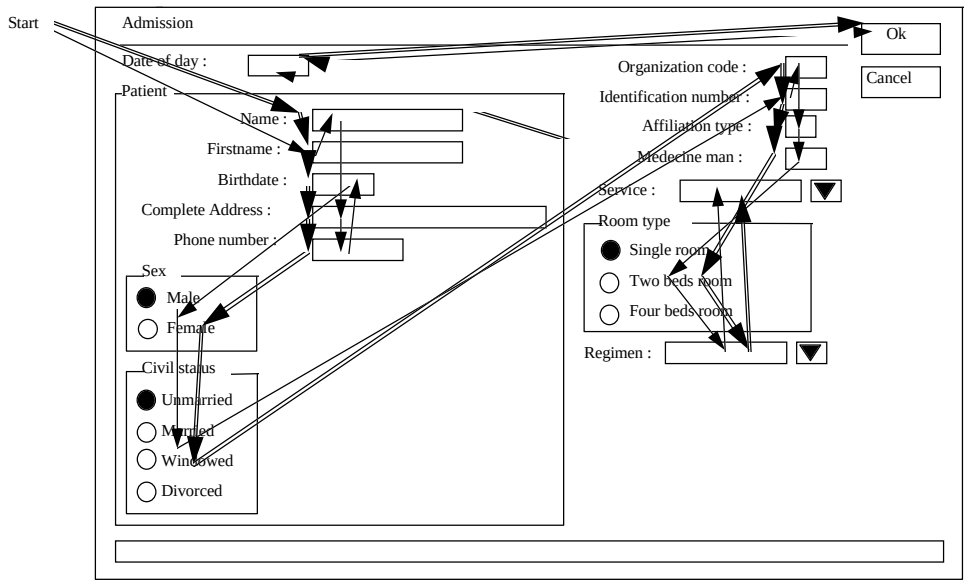

Figure 10. The two most frequent IO path on the two-column placement

The two most frequent sequence of actions have been represented on the result of the two-column placement strategy (fig. 10). The single solid line represents the most frequent sequence of actions among the IOs. The double solid line represents the second most frequent sequence of actions. These sequences are not optimal because the user first types the name and first name of the patient, then the other pieces of information. Then, the patient's id. is provided because it is an identifier, but the IO named "Identification number" is not the first of the second column, and so on. The user finishes the form by providing the date of day when the patient information are no longer necessary. Then the (Ok) pushbutton is pressed. We therefore conclude that the IO ordering in the columns does not match the most frequent sequence of actions since the user has to come back several times. The LA strategy tries to overcome this situation. Fig. 11 shows a LA-optimal layout for our problem. Several modifications have been introduced :

- the most frequent accessed IOs are placed first;
- the radio boxes (i.e. Sex, Civil status, Room type) have been re-arranged horizontally to reduce distance between the items;
- the "Female" item is placed before the "Male" item, assuming that 52% of the population consists of females and 48% of males;
- two new semantic groups have been identified due to the sequence of actions (i.e. Organisation and Medical care);
- the "Date of day" edit box and the message area are put at the end of the layout since they are the least frequent IOs.

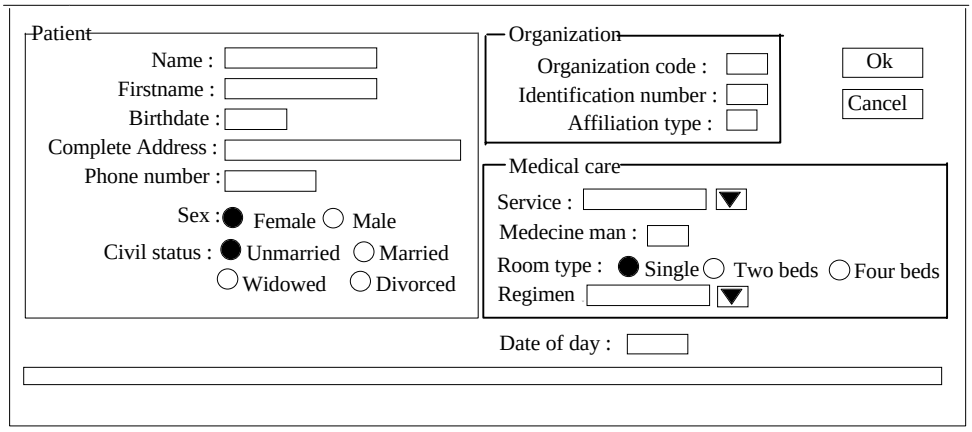

Figure 11. The example resulting from the Layout Appropriateness

Applying the LA-algorithm automatically leads to a placement in which the IO scanning is minimal. This is the strongest point of the strategy, but it precludes that the IO ordering by frequency is significant from the user's point of view. If the semantic structure of the information is not equal to the most frequent sequences of actions performed on these information items, the user may get lost, especially if these sequences of actions change from one user population to another. Otherwise, the user often prefers compact layouts such as in fig. 11. LA is useful for other purposes :

- LA allows any size of IO, even though this seems to rely on the LA-optimal searching technique;
- the evaluation of appropriate dimensions for a particular IO (e.g. the visibility of a list box, the preference for a drop-down object) : for instance, if the IO is frequently used and has small dimensions, a larger target could be suggested. Similarly, an IO that is rarely used could be made smaller to save screen space;
- the evaluation of a same layout manipulated by different populations of users, and not just one which is the most frequent;
- the evaluation of a paper layout without requiring much user resources, programming efforts, and costly task descriptions.

3.7. Placement with Right/Bottom Strategy

If the semantic structure of the information is not equal to the most frequent sequences of actions performed on these information items (the assumption for LA strategy is no longer valid) and if the semantic structure consists of a large amount of information items, another strategy that could be applied is the Right/Bottom strategy (Bodart, 1994b). The main hypothesis of this strategy is to provide an initial tree of IOs to be laid out. The IOs are then arranged by placing the next IO at the right or at the bottom of the previous one. The algorithm is summarised as following :

if the total length does not exceed the limit
then place S_{i+1} with horizontal sequencing
three cases are to be considered
1. height (S_i) = height (S_{i+1})

apply proportional uniformisation

2. height (S_i) > height (S_{i+1})
   if S_{i+1} = edit box then
      if S_i = list box or edit box
         then apply bottom justification
         else apply upper justification
3. height (S_i) < height (S_{i+1})
   if available space is sufficient
      then apply bottom justification
      else maximise upper justification

else place S_{i+1} with vertical sequencing.

Because each step consists of two alternatives (right or bottom), the space of all placement alternatives becomes a complete state-space which is a binary tree. A *state* is a set of placement conditions that describe the system at a specified point during the processing. At each state, dynamic heuristics are applied depending on the sequence and type of IOs. These heuristics are based on the mathematical relationships described in section 1.2. For example, apply horizontal equilibrium among horizontally placed IOs, apply vertical equilibrium among vertical groups, apply proportional or total equilibrium for any group of radio boxes. Finding a good placement then consists of browsing the tree, examining right-bottom alternative placements at each state, applying placement heuristics based on mathematical relationships, and cutting branches which lead to unpleasant placements. Fig. 12 depicts the screen dump of an example resulting from the Right/Bottom strategy.

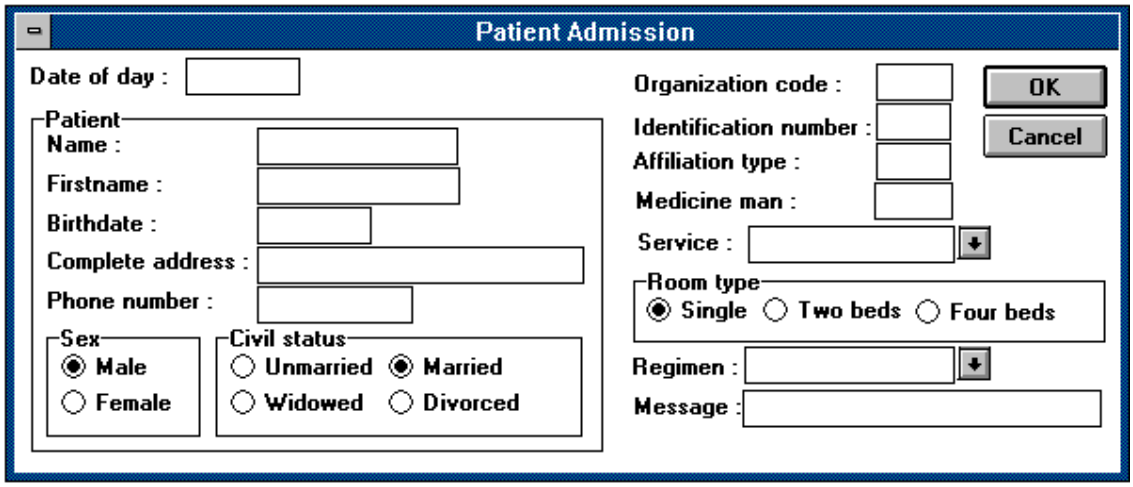

Figure 12. The example resulting from the Right/Bottom strategy

The Right/Bottom strategy allows the designer to dynamically follow the layout of the composite IO (e.g. the dialog box). A computer-aided tool for displaying the different steps for this is welcome. Moreover, multiple solutions are offered to the designer, from which a final layout is kept. One of the possible solutions is the "vertical and left-pushed" layout that may be obtained with UIDE and GENIUS. Therefore, their strategy can be considered as a particular case of the Right/Bottom strategy. On the other hand, the Right/Bottom minimises unused blank spaces by condensing the placement of IOs. This is a good point if the screen density does not become huge. In order to control this density, additional dynamic heuristics should be added. Of course, new heuristics could always be supplemented, but this requires an important job resulting from intensive testing with examples. The cost for obtaining this expertise is significant and not negligible.

The Right/Bottom strategy is quite the opposite of the two-column strategy : the first is only driven by user-definable heuristics, whereas the former is unmodifiable. This strategy seems consequently to be more flexible. In particular, the heuristic of establishing vertical justification across vertical groups of IOs and horizontal justification across horizontal groups of IOs is sound and generally applicable. Taken alone, however, these relationships are not enough to produce a quality layout simply by applying them locally throughout the different steps. Heuristics for local re-arrangement are to be performed.

The Right/Bottom strategy is grounded on a strong assumption which could be a hard part : the IO sequence is already characterised. After that, the placement problem becomes fairly straightforward. The ideal strategy should make more use of the application semantics to help solve this problem without manual intervention by the designer. If one asks how would real users or experts rate these visual configurations, one could answer that experts users often prefer large composite IOs with a plenty of IOs rather than multiple smaller composite IOs. They do not care too much about the screen density and the fragmentation of the result, provided that the composite IO includes all necessary IOs to accomplish the required interactive task.

4. Final Discussion, Conclusions and Future Work

We are now able to summarise the most salient features of the six compared strategies.

Characteristic	UIDE	GENIUS	Two-column	DON	LA	R/B
Automation level	Full automatic	Full automatic	Full automatic	Computer-aided	Full-automatic	Computer-aided
Orientation	One column	Two columns	Two balan-ced columns	Shape	IO usage frequency	Visual continuity
Assumptions	None	Ordering of attributes in entities and relationships	Sequence of IOs is given	There is no particular IO sequence	IO usage frequency is given	Tree of IOs is given
Rationale	Stack appearance	Two-columns appearance	Book appearance	IOs are sorted by decreasing sizes	IOs are laid out by frequency	IOs are laid out to reduce unused spaces
Amount of IOs	small	moderate	moderate	large	very large	very large
Solutions	unique	one or two	unique	multiple	unique	multiple
Flexibility	minimal	moderate	minimal	high	none	maximal
Fragmentation	minimal	minimal	minimal	low	moderate	maximal

˝

New placement strategies investigated seem quite appropriate, though the evaluation of the placement strategies is grounded more on subjective evaluation and heuristic claims that on actual experimentation

with users. For instance, task completion time, error rate have not been measured. But really aesthetic specifications in the placement problem is a very subtle skill that cannot easily be replicated trough automated strategies

Therefore, computer-aided visual design could be a replacement solution :

- for including explicitly human-computer guidelines without experts;
- for guaranteeing consistency between the choices made during the design;
- for providing advice-giving information to the designer;
- for assessing style guide or standard compliance, if necessary;
- for automatically building interaction objects of the user interface;
- for reducing costs of visual design by decreasing decision time;
- for allowing iterative design and rapid prototyping;
- for working at a higher level of control than simply arranging IOs.

However, computer-aided visual design seems not to be the universal panacea because

- there is far too much material for visual principles if one desires to include that in a fully automatic system. Thus, it is virtually impossible to include them all;
- our knowledge of visual design can be theoretical, experimental, or just common sense, even if common sense is sometimes wrong. Despite that we already dispose of much rules, several questions and trade-offs are still relevant of the research domain. The resulting strategy should consequently be incomplete itself;
- most of the guidelines can be easily incorporated into an automatic builder or an expert system. The rest of the guidelines cannot be incorporated because
  - they are either too general or too specific;
  - they are difficult to apply formally without human involvement;
  - they can require a great amount of calculations to be verified and the result will not probably be better;
  - the time for implementing the guideline is greater than the gain we can reasonably hope.

Though automatic tools can bring substantive benefits, they still need to be extended, completed, human assisted, and validated in the different steps for bests results.

Acknowledgments

The authors would like to thank four HCI'94 referees for helpful comments on earlier versions of this manuscript.

References

de Baar, D., Foley, J.D., Mullet, K.E. (May 1992), "Coupling Application Design and User Interface Design", *Proc. of CHI'92*, ACM Press, pp. 259-266.

Beshers, C.M., Feiner, S. (Nov 1989), "Scope: Automated Generation of Graphical Interfaces", *Proc. of UIST'89*, ACM Press, New York, pp. 76-85.

Bodart, F., Vanderdonckt, J. (1994), "Guide ergonomique des interfaces homme-machine", Presses Universitaires de Namur, Namur.

Bodart, F., Hennebert, A.-M., Leheureux, J.-M., Vanderdonckt, J. (Jun 1994), "Towards a Dynamic Strategy for Computer-Aided Visual Placement", *Proc. of AVI'94*, ACM Press, New York.

Galitz, W.O. (1992), "User-Interface Screen", Q.E.D. Information Sciences, Wellesley.

Hashimoto, O., Myers, B.A. (Nov 1992), "Graphical Styles for Building User Interfaces by Demonstration", *Proc. of UIST'92*, ACM Press, pp. 117-124.

Janssen, C., Weisbecker, A., Ziegler, J., (Apr 1993), "Generating User Interfaces from Data Models and Dialogue Net Specifications", *Proc. of INTERCHI'93*, ACM Press, pp. 418-423.

Kim, W.C., Foley, J.D. (Oct 1990), "DON: User Interface Presentation Design Assistant", *Proc. of UIST'90*, ACM Press, pp. 10-20.

Kim, W.C., Foley, J.D. (Apr 1993), "Providing High-level Control and Expert Assistance in the User Interface Presentation Design", *Proc. of INTERCHI'93*, ACM Press, pp. 430-437.

Lauer, D. (1990), "Design Basics", Holt, Rinehart, and Winston, Inc., Fort Worth.

Marcus, A. (1992), "Graphic Design for Electronic Documents and User Interfaces", ACM Press, New York.

Microsoft Excel (1992), "User's Guide 2, Worksheet Analysis, Exchanging Data, Customizing, Automating", p. 263.

Myers, B.A. (1988), "Creating User Interfaces by Demonstration", Academic Press, London.

Sears, A. (July 1993), "Layout Appropriateness: A Metric for Evaluating User Interface Widget Layout", *IEEE Transactions on Software Engineering* 19(7), pp. 707-719.

Singh, G., Kok, C.H., Ngan, T.Y. (Oct 1990), "Druid: A System for Demonstrational Rapid User Interface Development", *Proc. of UIST'90*, ACM Press, pp. 167-177.

Singh, G., Green, M. (July 1991), "Automating the Lexical and Syntactic Design of Graphical User Interfaces: The UofA* UIMS", *ACM Transactions on Graphics* 10(3), pp. 213-254.

Szekely, P., Luo, P., Neches, R. (Apr 1993), "Beyond Interface Builders: Model-Based Interface Tools", *Proc. of INTERCHI'93*, ACM Press, pp. 383-390.

Tarlin, E. (Apr 1990), "Direct Manipulation Design Studio", CHI'90 Tutorial #6 Notes.

Tufte, E.R. (1983), "The Visual Display of Quantitative Information", Graphics Press, Cheshire.

Vanderdonckt, J., Gillo, X. (Jun 1994), "Visual Techniques for Traditional and Multimedia Layouts", *Proc. of AVI'94*, ACM Press, New York.

Vander Zanden, B., Myers, B.A. (Apr 1990), "Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces", *Proc. of CHI'90*, ACM Press, New York, pp. 27-34.

Wiecha, C., Bennett, W., Boies, S., Gould, J. (Jul 1990), "ITS: A Tool for Rapidly Developing Interactive Applications", *ACM Transaction on Information Systems* 8(3), pp. 204-236.