SCHEDAE INFORMATICAE
VOLUME 15 2006

Fuzzy Interpretation of Layout Hypergraphs [1]

ANDRZEJ LACHWA, EWA GRABSKA, GRAZYNA SLUSARCZYK
Jagiellonian University, Faculty of Physics, Division of Design and Computer Graphics
Reymonta 4, 30-059 Kraków, Poland
e-mail: {lachwa, gslusarc}@uj.edu.pl, grabska@cyf-kr.edu.pl

Abstract: This paper presents a new approach to design floor-layouts with the use of computer. This approach is based on a special type of data structure in the form of a hierarchical hypergraph. This data structure describes the whole dass of designs with fuzzy interpretation.

1. Introduction

This paper presents our new approach to design dwelling-house floor-layouts with the use of computer. This approach is based on a data structure being the internal representation of designs. This approach is a continuation of our research related to applying hypergraphs to diagrammatic design [2, 5]. Our method emerged while considering a set of view drawings of a given house. Starting from a given drawing describing the initial design requirements we create its representation in the form of a special type attributed hierarchical hypergraph which corresponds to a class of drawings. The hierarchical organization of this representation enables the designer to consider floor-layouts on different levels of detail.

Our main goal is to develop the automatic interpretation of drawings presenting designer's ideas, which are created with the use of a graphical editor, in the form of

[1] The research is supported by State Committee for Scientific Research (in Poland) grant no. 5 T07F 002 25.

hypergraphs being the internal design representations. This paper presents preliminary results of our research.

We propose fuzzy values (see [4]) for the attributes of hypergraph elements representing drawing components, like walls and rooms. Such a hypergraph representation leads to a fuzzy interpretation which determines a class of view drawings for this hypergraph. The specified fuzzy interpretation allows one to generate instances of a class of view drawings corresponding to the hypergraph representation. As in each drawing instance all its components have exact values of their features (like room areas, length of walls, angles between walls) the correspondence degree of the instance drawing to the hypergraph representation can be computed by comparing these crisp values with fuzzy values of hypergraph attributes.

Choosing drawing instances which satisfy the design goals and requirements in the best way consists in fuzzy evaluation of drawing instances, which is based on their degree of correspondence to the hypergraph representation. In our approach we assume that the evaluation of drawing instances functionality and aesthetics is left to the designer.

2. How to get the internal representation

The idea of the proposed data structure emerged from the analysis of technical plans. Given a technical floor projection of a dwelling-house, first the corresponding drawing on the chosen level of detail is obtained by removing everything but some walls with openings (Fig. 1). Drawings obtained from a given technical view can represent the division of the layout interior into regions on different levels of detail. For example, the drawing on the most detailed level can be obtained but leaving all walls while the one on the lowest level of detail by leaving only external walls.

The drawing on each level is composed of disjoint and adjacent closed regions. Each region has a boundary composed of the segments of walls. The regions of the drawing are represented by graph hyperedges labelled by the region names. Then the walls of the drawing are divided into segments and labelled. These segments are represented as hypergraph nodes. Hyperedges are connected with nodes representing segments which bound corresponding regions (Fig. 2). Each connection between two segments is represented by a hypergraph arc (Fig. 3).

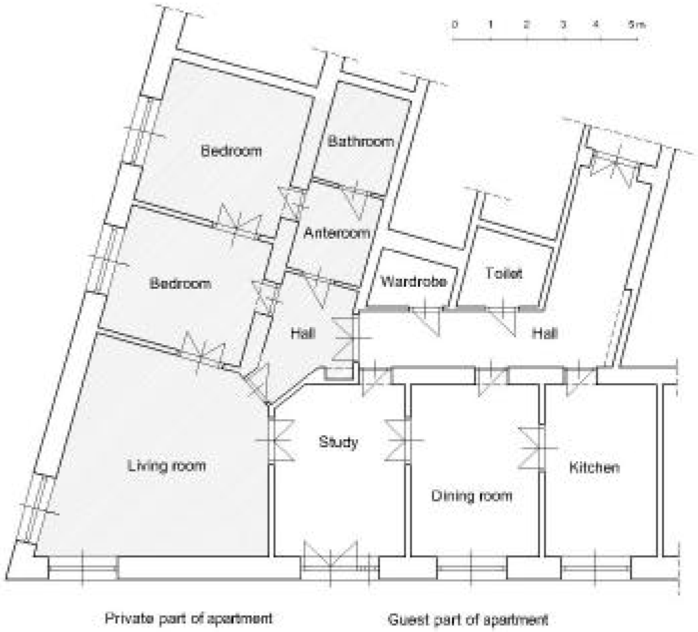

Fig. 1. An example of a floor-layout drawing of an apartment

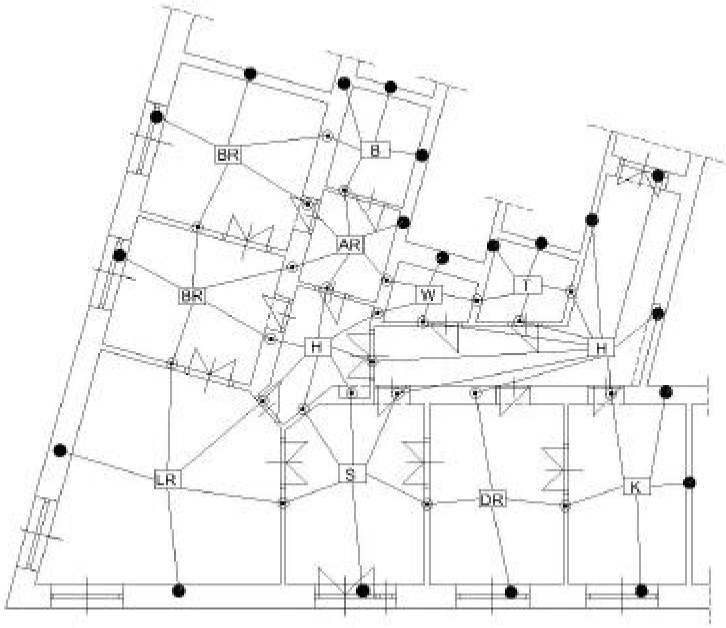

Fig. 2. The floor-layout from Fig. 1 with a hypergraph

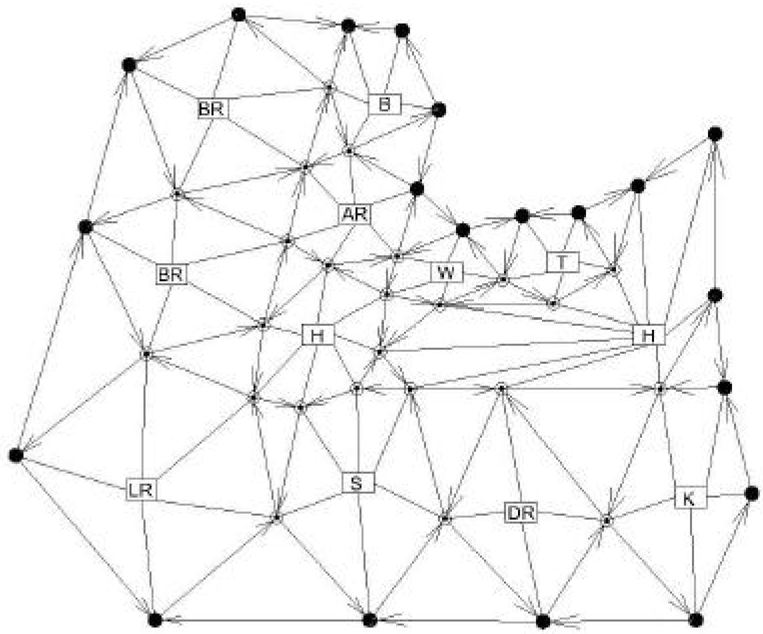

Fig. 3. A simple layout hypergraph representation of the drawing from Fig. 1

The same floor projection as the one presented in Fig. 1 can be represented on the lower level of detail as a space composed of the guest part and the private part, and on the lowest level as the undivided space of the whole apartment. The hierarchical simple layout hypergraph representing the two levels is presented in Fig. 4.

It should be noticed that several wall segments can form single segments on the lower level of hierarchy. Therefore we use hierarchical nodes in our representation (Fig. 4).

The characteristic features of the drawing are represented by attributes assigned to the corresponding elements of the hypergraph. Each hyperedge has attributes which characterize regions, e.g. shape, area and exposure. Each node has attributes which characterize wall segments, e.g. length, type and thickness. The hypergraph arc is attributed by the *angle* of the connection between wall segments. In our approach we assume that angles are oriented and their values are less or equal to 180^o (Fig. 5).

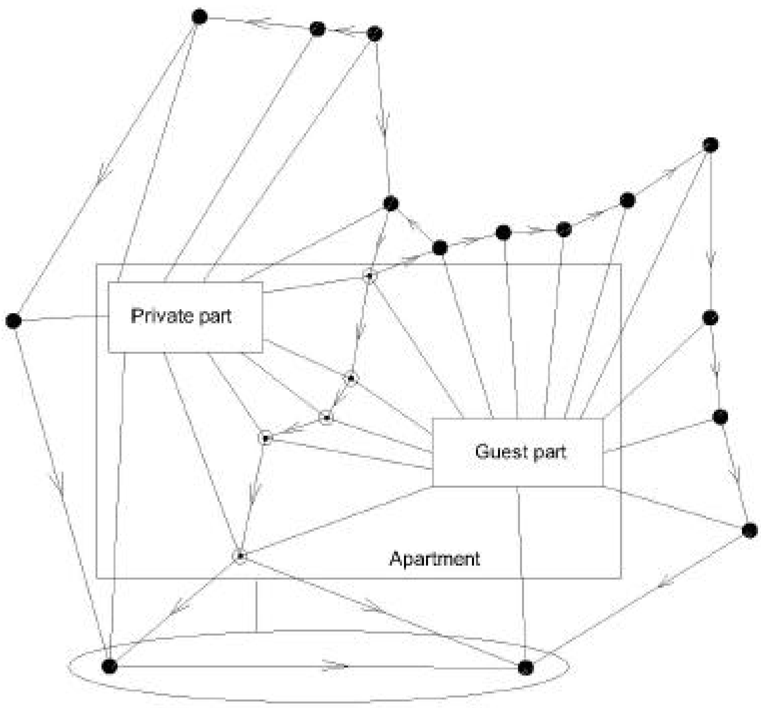

Fig. 4. A hierarchical simple layout hypergraph representation of the apartment from Fig. 1

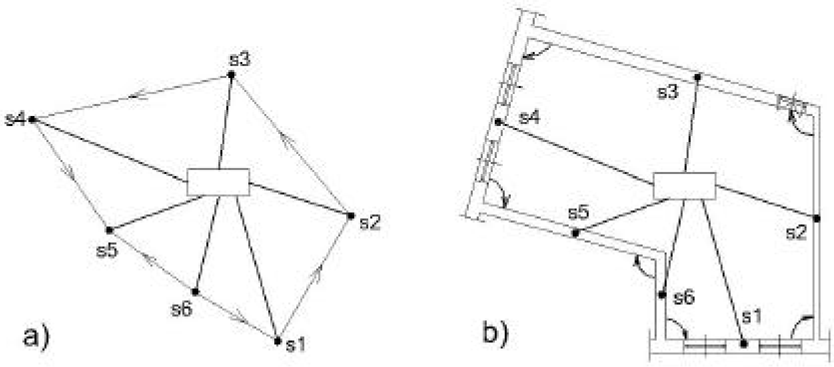

Fig. 5. A hypergraph representation of angles (a) between drawing segments (b)

3. Hierarchical simple layout hypergraphs

In [2] directed and hierarchical hyperedge-labelled hypergraphs called **hierarchical layout hypergraphs** were proposed to represent floor-layouts. In this paper the hypergraph definition is restricted to the simpler version which will be called a **hierarchical simple layout hypergraph** (HSL-hypergraph).

The hyperedges of the HSL-hypergraph are non-directed and represent drawing regions. By a **drawing region** we understand bounded and connected space which corresponds to a room or a group of rooms. A hyperedge which is hierarchical represents a group of regions as one region on the lower level of detail. Subregions of a given region can be represented in the form of another HSL-hypergraph nested in the hyperedge corresponding to this region.

Hypergraph nodes represent wall segments. By a **wall segment** we understand a wall fragment which can be connected with other wall segments only on its ends. Moreover it should be straight, maximal, uniform and either common for two neighbouring rooms (regions) or have to separate a room (region) from the outside. To each hyperedge representing a region a sequence of nodes, which represent walls bounding this region, is assigned. Two hypergraph nodes can be connected by a directed arc which represents the adjoin relation between segments corresponding to these nodes.

Hyperedges of the HSL-hypergraph are labelled by names of the corresponding regions of the drawing. These labels give semantic information about the designed layout. Hypergraph nodes and arcs are not labelled.

Since our representation is hierarchical each hypergraph can be treated as a hyperegde on the lower level of detail. Therefore for each hypergraph a sequence of its external nodes is determined. The length of this sequence specifies the type of a hypergraph.

Let $[i]$ denote the interval $\{1...i\}$ for $i \geq 0$ (with $[0]=\emptyset$). Let Σ be a fixed alphabet of hypergraph labels.

DEFINITION 3.1. *A* ***hierarchical simple layout hypergraph*** *over* Σ *is a system* $H=(E_H, V_H, R_H, t_H, lb_H, ext_H, ch_H, nd_H)$*, where:*

1. E_H *is a finite set of hyperedges representing drawing regions,*
2. V_H *is a finite set of nodes representing wall segments,*
3. $R_H \subseteq V_H \times V_H$ *is a finite set of directed arcs representing the adjoin relation between couples of segments,*
4. $t_H: E_H \rightarrow V_H^*$ *is a mapping assigning a sequence of nodes to each hyperedge,*
5. $lb_H : E_H \rightarrow \Sigma$ *is a hyperedge labelling function,*
6. $ext_H: [n] \rightarrow V_H^*$ *is a mapping specifying a sequence of external nodes,*
7. $ch_H: E_H \rightarrow P(A)$ *is a child nesting function, where* $A = V_H \cup E_H \cup R_H$ *is called a set of hypergraph atoms, and the following conditions are satisfied:*
   - $\forall a \in A \ \forall e_1, e_2 \in E_H \ a \in ch(e_1) \wedge a \in ch(e_2) \Rightarrow e_1=e_2$, *i.e., one atom cannot be nested in two different hyperedges,*
   - $\forall e \in E_H \ \ e \notin ch^+(e)$*, where* $ch^+(e)$ *denotes all descendants of a given hyperedge e, i.e., a hyperedge cannot be its own child,*

8. $nd_H: V_H \rightarrow P(V_H)$ *is a function specifying nodes representing wall subsegments for hypergraph nodes representing whole segments.*

Apart from labels semantic information can be assigned to the hypergraph elements in the form of attributes. Therefore a notion of an attributed hierarchical simple layout hypergraph is introduced.

Let *AT* be a set of hypergraph atoms attributes.

DEFINITION 3.2. *An* ***attributed hierarchical simple layout hyper-graph*** *is a system* $AH = (H, att_H)$, *where:*

1. $H = (E_H, V_H, R_H, t_H, lb_H, ext_H, ch_H)$ *is a HSL-hypergraph,*
2. $att_H: V_H \cup E_H \cup R_H \rightarrow P(AT)$ *is a function assigning sets of attributes to nodes, hyperedges and arcs, respectively.*

4. Fuzzy interpretation

In this section we shall define fuzzy interpretation of hypergraphs and fuzzy evaluation of potential design layouts.

In order to visualise an attributed hypergraph on the specified level first the designer has to determine the recurrence level of the child nesting function for each hyperedge and then assign values to hypergraph attributes.

Let *A* and *AT* be sets of hypergraph atoms and atom attributes, respectively. Let *D* be a set of fuzzy and linguistic values of atom attributes.

DEFINITION 4.1. *An* ***interpretative hierarchical simple layout hyper-graph*** *is a system* $IH = (AH, Val)$, *where:*

1. $AH = (H, att_H)$ *is an attributed HSL-hypergraph,*
2. *Val is a family of partial functions assigning values to attributes of the hypergraph atoms in such a way that* $\forall a \in AT\ val_a:(A, a) \rightarrow D$.

The interpretation of the HSL-hypergraph defined above is obtained by assigning drawing components to hypergraph atoms. These components can take different sets of crisp values corresponding to hypergraphs attributes. On the basis of each of these sets we obtain a set of drawings corresponding to the single floor-layout project.

Denote by *F* and *S* a family of drawing sets and a set of drawings, respectively. Let *R(IH)* be a family of interpretative HSL-hypergraphs.

DEFINITION 4.2. *By the* ***hypergraph interpretation*** *we understand a function* $int: R(IH) \rightarrow P(F)$, *where* $P(F)$ *is a set of subsets of F.*

Different sets of drawings $S \in int\ (IH)$ match the hypergraph represen-tation in various degrees. The *correspondence degree* of the set of drawings is computed as the weighted average of membership degrees of respective exact values of drawing component attributes to fuzzy values of hypergraph atom attributes.

For a given interpretative HSL-hypergraph *IH* the interpretation $int(IH) \subset F$ is a *space of potential design solutions*. If we assign a correspondence degree to each element $S \in int\ (IH)$ and treat it as a membership function value $\mu(S)$ then we obtain a *fuzzy space of solutions*.

DEFINITION 4.3. *By the* ***hypergraph fuzzy interpretation*** *we under-stand a function* $fint: R(IH) \rightarrow Fuzzy(F)$, *where* $Fuzzy(F)$ *is a family of fuzzy subsets on the universum* F.

5. Conclusions

There exist different approaches to design floor-layouts with the use of computer [1, 3]. In this paper we proposed a data structure in the form of an interpretative HSL hypergraph which enables us to gather and extract information about designs on different levels of details. Moreover, we extended the concept of a hypergraph interpretation to a fuzzy one. Due to the fuzzy interpretation we are able to evaluate the correspondence of solutions to the project specification.

6. References

[1] Borkowski A., Grabska E. and Szuba, J.; *Visualisation of Graphs in ArchiCAD*, Lecture Notes in Computer Science, Vol. 1774, 1999, pp. 241–246.

[2] Grabska E., Grzesiak-Kopec K., Lembas J., Lachwa A., Slusarczyk G.; *Hypergraphs in Diagrammatic Design*, in: K. Wojciechowski et al., (ed.), *Computer Vision*, Springer, 2006, pp. 111–117.

[3] Heisserman J., *Generative Geometric Design*, IEEE Computer Graphics and Applications, March 1994, 1994, pp. 37–45.

[4] Lachwa A.; *Fuzzy world of sets, numbers, relations, facts, rules and decisions* (in Polish), AOW EXIT, 2001.

[5] Minas M.; *Concepts and realization of a diagram editor generator based on hypergraph transformation*, Science of Computer Programming, 44, 2002, pp. 157–180.

Received January 1, 2006