7A−8

知識ベースを取り入れた制約指向フロアプランニングシステムの設計*

市原尚久† 本多一賀‡ 大和田勇人† 溝口文雄†

東京理科大学 理工学部 東京ガス(株)

1 はじめに

住宅設計の専門家が間取りの配置を考えるとき，その経験，知識及び試行錯誤により完成されていく事が多い．この間取りの配置問題を考えるフロアプランニング問題は要求される制約条件を基に探索的に間取り配置を考えることを目的とした一種のレイアウト問題であるが，特定のアルゴリズムを持たず，複雑性を含むものである．フロアプランニング問題を解決する試みとして制約指向フロアプランニングシステム [1] があるが，これは実際的な応用には至っていない。本システムではこの制約指向フロアプランニングシステムの実用化への拡張を試みるために建築設計の専門知識に基づく知識ベースを導入した。

2 制約指向アプローチ

間取り設計は通常,「LDK，水まわりの他に南向き和室と隣接した子供部屋，寝室が欲しい」などの要求する条件を基に考えていくが，このような条件を制約条件として考え，部屋の配置位置をそれらを満たす解とすれば，フロアプランニング問題は制約充足問題として考えられる．

本システムでは次のような制約集合を制約条件として定義した。

必然制約 on-floor 各部屋はフロア内に含まれるという制約．

必然制約 size 各部屋は縦幅，横幅の最小値，最大値を持つという制約．

必然制約 non-overlap 各部屋どうしは重ならないという制約．

方位制約 location 東西南北のいずれかに配置される部屋の制約．

隣接関係制約 adjacent 隣り合わせになる2部屋の制約．

ここで，必然制約とは必ず与えなければならない制約を意味する。

要求する制約条件はこの5つの制約の組み合わせにより記述可能である．例えば,「南向きの寝室（8畳）が欲しい」という制約は次のような述語で記述される．

$$plan(Bedroom, Otherrooms) : -on\_floor(Bedroom),$$

$$size(Bedroom, 3640, 3640, 3640, 3640),$$

$$non\_overlap(Bedroom, Otherrooms),$$

$$location(Bedroom, south).$$

フロアプランニング問題はこれらの制約集合で記述されるので，その解はこれらの制約を満たす変数集合であり，間取り案に相当する．

従って，次のような手順で間取り生成が行われる．

手順1 ユーザが制約条件を入力する．

手順2 与えられた制約条件の制約充足（プランニング）を行う．

手順3 求めた解を間取り案として出力する．

*Incorporating knowledge base into constraint-oriented floorplanning system
†Science Univ. of Tokyo
‡Tokyo Gas Co.,Ltd.

3 知識ベースの導入

前節で述べたような制約充足のみで適当な間取りを完成させる事は非常に困難である．その原因は与える制約に問題があるが，主として以下に述べる3点から言える．

1. プランニングから得られる間取りが非現実的である．(非実用的問題)
2. 与える制約が少ないために，解が具体化されない.(under-costraint問題)
3. 制約を与えすぎてプランニングに失敗する (over-constraint問題)

非実用的問題は，間取り知識による制約の組み合わせ制限などにより解決が可能であり，これによりシステムが実用性を持つことになる．under-constraintの問題は制約の補充により，over-constraintの問題は制約の緩和により解決が可能である．これによりシステムは柔軟性を持つことになる。

4 知識ベースの定義

本システムでは以下のような3種類の知識ベースを導入した．

1. 情報型知識ベース KB-Im
2. デフォルト型知識ベース KB-Df
3. 推論型知識ベース KB-In

4.1 情報型知識ベース

定義 KB-Im

目的 与える制約の組み合わせの制限，ユーザ入力の簡易化

機能 それぞれの制約情報を持つ部屋，空間のタイプをデータベース化し，入力のサポートを行う。

種類 LDK，水まわり，和室，階段，玄関

例えば，LDKなどのコンビネーション空間は部屋間の隣接関係制約などの組み合わせにより記述可能であるが，その組み合わせ方は理論的に何通りも存在する．しかし，実際の間取りでは非現実的な組み合わせも多く存在する．例えば，LD+Kという組み合せが存在しても，LK+Dという組み合わせは存在しない．入力インターフェイスにおいてLDKを選択する際，この組み合わせは用意されていないので，非現実的な組み合せは回避できる．つまり，ユーザはシステムがプリミティブに用意したタイプのものから選択できることになるのである．

例えばユーザがLDKについて「LD+Kタイプの L型」を選択すると次のような関係の制約が導出される．

$$location(Dining, corner)$$

$$adjacent(Living, Dining, any)$$

$$adjacent(Kitchen, Dining, any)$$

さらに，形状，大きさなどがほぼ決まっている空間（階段，玄関，和室，押入など）についてもその大きさの情報を予め知識ベースが持っている．

例えば，「和室６畳＋１畳（押入）」という入力により，次のような大きさ，関係の制約が導出される．

$$size(Japanese, 2730, 2730, 3640, 3640)$$

$$size(Closet, 1820, 1820, 910, 910)$$

$$adjacent(Japanese, Closet, any)$$

4.2 デフォルト型知識ベース

定義 KB-Df

目的 under-costraint,over-constraint の問題の回避

機能 制約の補充及び緩和

種類 方位，隣接関係，大きさ

このデフォルト型知識ベース (KB-Df) はユーザが指定しなかった制約や，矛盾を起こすような制約の代わりをするものであり，「... は... が好ましい」という知識を優先度付きでシステムに保存してある．

システムはユーザの与えなかった制約で，それに関する知識ベースを持つ場合にそれを取り出してプランニング（配置）に使っていく．（制約補充機能）

又，ユーザが与える制約が配置不可能であったり，矛盾を含むものであることがあり，プランニングに失敗した場合にシステムはユーザの与えた制約を削除して，このデフォルト型知識ベースを取り出して再びプランニングを行う．（制約緩和機能）

例えば，ユーザが「子供部屋を南向き」としたがプランニングに失敗したとき，「南向きが無理なら東向きにする」という知識から，次のような制約の緩和が行われる．

$$location(Childs, south) \Rightarrow location(Childs, east)$$

4.3 推論型知識ベース

定義 KB-In

目的 under-constraint,over-constraint の問題の回避，間取りの拡張

機能 制約の補充，緩和，オプション追加

種類 部屋名，大きさ，オプション

推論型知識ベース (KB-In) は必要な部屋，オプション (窓，ドアなどの大きさ，位置) の推論を行うものである．これはある条件が成り立つときに知識ベースが働く仕組みになっている．すなわち，条件等を必要としないデフォルト型知識ベースと比較して柔軟な推論構造に基づいている．

例えば，フロアの大きさに対して与えた部屋が少ない場合に，必要な部屋とその大きさを推論する事ができる．

オプションに関するデフォルト知識ベースは間取りの配置が完成した後に自動的に部屋に窓，出窓，ドアを適当な位置に適当なオプションをつけ加えるものである．この知識ベースは部屋の大きさ，位置などを考えて配置するオプション，位置，向きを考えてそれを間取りに追加する機能を持っている．

	入力した制約条件	導出される制約情報	知識ベース
(1)	1階フロアは 6370mm x 8190mm	size(Floor,7280,7280,9100,9100)	
(2)	2階フロアは 6370mm x 4560mm	size(Floor,7280,7280,8190,8190)	
(3)	LDKは LD+Kタイプ L型	size(Living,2730,4550,2730,4550)	KB-In
		size(Dining,2730,3640,2730,3640)	KB-In
		adjacent(Dining,Kitchen,any)	KB-Im
		adjacent(Dining,Living,any)	KB-Im
		location(Dining,corner)	KB-Im
(4)	キッチンはカウンター型	size(Kitchen,2730,3640,2730,3640)	KB-Im
(5)	水まわりは全調換型	size(Bathroom,2730,2730,1830,1830)	KB-Df
		size(Toilet,910,1820,910,1820)	KB-Df
		size(Utility,1820,1820,1820,1820)	KB-Df
		adjacent(Bathroom,Utility,any)	KB-Im

図 1: 知識ベースにより導出された制約

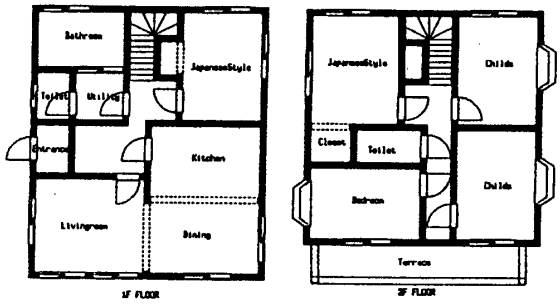

図 2: 間取り案

5 システムの拡張

先ほど示した手順に知識ベースを導入し，次のような新たな手順のシステムに拡張する．

手順 1 ユーザが KB-Im を利用して制約条件を入力する．

手順 2 KB-Df,KB-In により制約の補充，緩和を行う．

手順 3 与えられた制約の制約充足（プランニング）を行う．

手順 4 失敗したら手順 2 へ，成功したら手順 5 へ．

手順 5 KB-In のオプション追加により完全な間取りを完成させる．

手順 6 完成した間取り案を出力する．

6 実行例

8 つのユーザ入力を行った結果，27 個の制約情報が導出された．その結果（一部）を図 1 に示す．この実行例ではユーザの入力に対して KB-Im,KB-Df,KB-In がそれぞれ使われている．

さらに，この 27 個の制約によるプランニングを行い，その結果に窓やドアが追加され，図 2 の様な間取り案が出力される．

7 おわりに

本論文では制約指向フロアプランニングシステムにおける知識ベースの導入について述べた．フロアプランニング問題を制約充足問題として扱ったが，非現実的，非効率的問題が生じた．これらを解決するために，本システムでは建築設計の専門家の知識を知識ベースとして保存することで，システムの実用化，柔軟性が実現された．

参考文献

[1] 溝口，大和田，本多，制約指向フロアプランニングシステムの設計，情報処理学会第 45 回全国大会