

平成 16 年 10 月 31 日

1 注意

このノートは講義で説明するアウトラインや内容を簡潔に記すものである。慎重に定義すべき概念を大雑把に説明してあるので注意すること。

2 Lecture 2. Design, Description, and Knowledge

2.1 記述制作としての設計

2.1.1 人工物の構成

人工物を構成する部分や部品を**設計要素**（design element）とよぶ。設計要素は他の設計要素の部分や部品になりうる。設計要素のうち他の設計要素によって構成されないもの、すなわち設計要素の最小単位を**基本設計要素**（design atom）とよぶ。設計に用いられる記号は設計要素に対応する。すべての設計要素の集合を U_E によって示す。設計要素を配置したものを人工物（artifact）とよぶ。人工物の集合を U_A によって示す。設計要素は様々な特徴（property）をもつ。形態素は設計要素の特徴のひとつである。寸法、色、材質、それが何であるかなども設計要素の特徴である。また、配置されている設計要素は位置や姿勢等の特徴をもつ。設計要素 e が特徴 p をもつことを $e \in P_p$ によって示す。設計要素を配置することによって設計要素間の位置関係や機構的な関係が具体化される。例えば、動線、力や熱を伝える経路などが機構的な関係である。 U_E の二つの要素 e_1, e_2 が関係 r をもつことを $\langle e_1, e_2 \rangle \in R_r$ によって示す。ここで、 $R_r \subseteq U_E \times U_E$ である。

人工物はそれを構成するために配置されている全設計要素、設計要素の全特徴、設計要素間の全関係によって特徴づけられる。人工物のこのような特徴を人工物の**結構**（structure）とよぶ。人工物の形態的構成やある性能を実現する機構は人工物の結構の部分である。人工物の結構 a において配置されるすべての設計要素の集合が $U'_E (\subseteq U_E)$ であり、これらの設計要素が特徴 $p_i (i \in \{1, 2, \dots, k\})$ をもち、これらの設計要素間に関係 $r_i (i \in \{1, 2, \dots, l\})$ があることを 1 式によって表わす。ただし、 $P'_{p,i} \subseteq U'_E$, $R'_{r,i} \subseteq U'_E \times U'_E$ である。

$$a = \langle U'_E, P'_{p,1}, \dots, P'_{p,k}, R'_{r,1}, \dots, R'_{r,l} \rangle \quad (1)$$

結構の可能な形式のすべての集合を U_a によって示す。

2.1.2 人工物の記述

設計図は人工物の構成を特定の表記システム（notation system）によって描写する。人工物の描写において繰り返し現れる有意味な記号の最小単位を**設計詞**（design word）とよぶ。記号が有意味であるとはそれが何らかの設計要素を示しているということである。すべての設計詞の集合を**設計アルファベット**とよび、 Σ_D によって示す。設計図は Σ_D の要素である設計詞を人工物の構成が投射される空間（通常の場合、平面）に配置したものである。設計詞を配置することによってできる記号の集まりを**設計詞配置**（formation）とよぶ。ここで、設計詞の配置の仕方は配置される空間によってのみ制限され、それ以外の制約はないものとする。すべての設計詞配置の集合を Σ_D^* によって示す。

すべての設計詞配置が必ずしも人工物の構成を描写するわけではない。設計詞配置のうち人工物の結構（ U_a の要素）を指示対象とする設計詞配置を**設計描写**（design description）とよぶ。設計描写の集合を S_D によって示す。 S_D は人工物の結構の可能な形式の描写のすべてからなる集合である。設計詞配置のうち S_D に属さないものの集合を S_{diff} によって示す。 S_{diff} に属する設計詞配置に対応する結構（ U_a の要素）は存在しない。 S_{diff} の要素を人工物として物理的あるいは論理的に成立しえないものを指示しようとする設計詞配置とみなす。 S_D , Σ_D^* , S_{diff} の間には下式の関係がある。

$$S_D \cup S_{diff} = \Sigma_D^* \quad (2)$$

$$S_D \cap S_{diff} = \emptyset \quad (3)$$

2.1.3 設計解

人工物は使用される環境において新たな環境を創出する。ここでは「環境」という語を極めて広い意味で用いている。自然環境、建築環境、社会環境を含む意味を持つ語としても用いているし、人工物に対して人工物以外の事物や現象の総体、人工物の使用者に対してそれ以外の事物や現象の総体を示す語としても用いている。新たな環境を創出し、その環境と良好な関係を保つことが、人工物の実現の前から、人工物に期待されている。人工物を設計（design）することは期待される環境（人工物と良好な関係をもつことを含む）を実際に提供しうる人

工物の結構を具現化する有目的 (purposive) な行為である。与えられた目的に合う合目的 (purposeful) な人工物の設計描写を**設計解** (design solution) とよぶ。設計解の集合を S_{Ds} によって示す。 S_{Ds} の要素は人工物の目的によって異なる。設計解, 設計記述, 設計詞配置の関係を 4 式及び図 1 に示す。設計は設計解となる設計描写を制作する行為である。

$$S_{Ds} \subseteq S_D \subseteq \Sigma_D^* \quad (4)$$

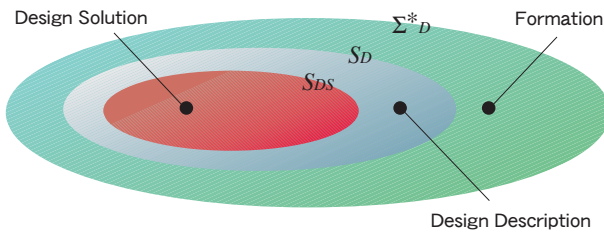


図 1 設計詞配置, 設計描写, 設計解の包含関係

2.1.4 環境描写による設計解の目的の表現

設計描写が目的に適う設計解のひとつ (S_{Ds} の要素) であることはその設計描写が指示する結構をもつ人工物が期待される環境を提供しうることを検証することによって確認される。人工物の結構を a , 人工物が使用される環境を s_o , 人工物によって創出される環境を s_e とする。 s_o にある a が s_e を創出することを $\langle a, s_o, s_e \rangle$ によって示す。環境がとりうるすべての状態の集合を U_S によって表わす。 $s_o, s_e \in U_S$ である。期待される環境は唯一つであるとは限らない。人工物が提供することを期待される環境を U_S の部分集合 $U_{S,ex}$ によって表わす。

環境のありようを示す記述を**環境描写** (environment description) とよぶ。すべての環境描写の集合を $\mathcal{L}_S$ とする。 $\mathcal{L}_S$ の任意の要素 (環境描写) は U_S の要素 (環境のありよう) に対応するものとする。設計描写 δ_a がその結構を示す人工物が環境記述 δ_{so} が示す環境で使用される場合に環境描写 δ_{se} によって示される環境が創出されることを 5 式によって表現する。

$$\delta_{se} = f_a(\delta_a)(\delta_{so}) \quad (5)$$

人工物の設計の目的を「環境描写 δ_{so} が示す環境における使用によって $\Delta_{S,ex}$ によって示される環境が創出されること」とする。ここで, $\Delta_{S,ex}$ は $U_{S,ex}$ を指示する。設計描写 δ_a が設計解となる条件 (十分条件) は $\delta_{se} = f_a(\delta_a)(\delta_{so})$ なる δ_{se} が $\Delta_{S,ex}$ を含意することである。このとき, $\delta_a \in S_{Ds}$ である。

2.2 設計と知

2.2.1 問題解決としての設計

人工物の設計は Σ_D^* の要素のうち目的に適うという特徴をもつ要素の集合 S_{Ds} に属する設計描写を制作する行為である。一般に, ある要素の集合 P からある特徴をもつ P の部分集合 S の要素を見つけることを問題解決 (problem solving) という。設計は問題解決の形式によって表現可能である。問題解決のプロセスは解生成 (generation) と解検証 (verification) の二つのサブプロセスからなる。設計における解生成プロセスは設計記述の**生成** (synthesis) であり, 解検証プロセスは生成された設計描写が S_{Ds} に属する (目的に適う) か否かを判定するプロセスである。4 式の関係により, 設計描写ではない設計詞配置は設計解になり得ない。設計の解検証プロセスはさらに二つのサブプロセス, **分析** (analysis) と**評価** (evaluation) からなる。設計描写が指示する人工物が想定される環境において使用される場合に創出される環境を予測するプロセスが分析プロセスである。創出されると予測された環境が期待されている環境であるか否かを判定するプロセスが評価プロセスである。評価結果が是の場合, 設計描写は設計解となる。評価結果が非の場合, 新たな設計記述が生成され, 検証される。評価結果を是とする設計描写 (設計解) が得られるまで生成と検証が繰り返される。環境描写 δ_{so} が示す環境における使用によって $\Delta_{S,ex}$ によって示される環境を創出することを目的とするとする。解生成プロセスにおいて設計描写 δ_a が生成され, 分析 $\delta_{se} = f_a(\delta_a)(\delta_{so})$ によって δ_a が δ_{so} において使用された場合に δ_{se} が創出されると予測され, 評価によって δ_{se} が $\Delta_{S,ex}$ を含意することが確かめられれば, δ_a は設計解である。

しかし, 設計の問題は輪郭の明確な問題 (well-defined problem) ではなく, 輪郭が明確でない問題 (ill-defined problem) である。 δ_{so} と $\Delta_{S,ex}$ が明確に定義されていれば, 設計は明確な問題となり, 設計記述が S_{Ds} に属するか否かの判定は上に示す方法によって可能である。しかし, $\Delta_{S,ex}$ は設計開始時には漠然 (vague) としており, 設計の進行に伴って推敲され, 問題の輪郭が明確になっていく。 $\Delta_{S,ex}$ を変更することによって問題を修正するプロセスを**再定義** (reformulation) とよぶ。設計描写と問題の輪郭との間にはトランザクショナル (transactional) な関係がある。再定義は設計描写 (または設計詞配置, 以下同じ) に対応してなされる。設計描写が詳細になるにつれて問題が明確になり, 問題が明確になるにつれて設計解となりうる設計描写の集合のサイズが小さくなるとともに設計描写の詳細が決定可能になる。

2.2.2 形式の採用としての設計

設計詞配置が持つべき形式を制限することによって生成可能なすべての設計描写の集合のサイズを小さくすることが可能である。設計詞配置の生成において S_D に属するもののみを生成する形式を採用することにより、 S_{diff} に属するナンセンスな設計詞配置の生成と検証（設計にとっては無駄と考えられるプロセス）を省くことができる。また、 $\mathcal{L}_D \subset S_D$ なる $\mathcal{L}_D$ の要素のみを生成する形式をもつ表記システムの採用によって設計解を探す空間が小さくなる。 $\mathcal{L}'_D \subset S_{Ds}$ なる $\mathcal{L}'_D$ に属する設計詞配置のみを生成する形式をもつ表記システムを採用した場合、その形式に従って生成された任意の設計詞配置は設計解となる。 $\mathcal{L}_D$ や $\mathcal{L}'_D$ を**設計言語**（design language）とよぶ。

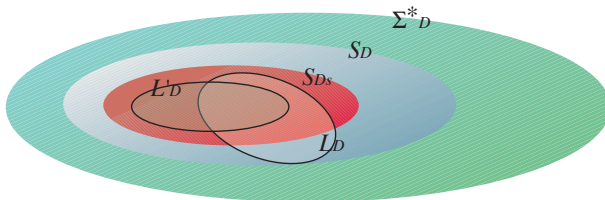


図 2 設計解と設計言語の関係

2.2.3 設計における知と合理性

問題解決としての設計及び形式の採用としての設計に基づいて設計において使用される知について考察する。設計には次の知が重要な役割を演じる。

設計解知 設計解の集合 S_{Ds} の要素であるか否かを直接判定するための知。

問題定義知 期待される環境 $\Delta_{S,ex}$ を明確に定義するための知。

分析知 設計描写と環境描写から環境描写を導くための知（ f_a に関する知）。

評価知 f_a によって導かれた環境描写が $\Delta_{S,ex}$ を含意することを確認するための知。

再定義知 期待される環境 $\Delta_{S,ex}$ を設計の履歴を考慮して再定義するための知。

構成知 設計描写の制作するための操作や操作方針に関する知。換言すれば、設計問題の解法や手順に関する知。

形式知 設計詞配置の可能な形式を制約する知。

設計解知をもっているということは問題の解を既に知っているということである。設計問題に対して直接に設計解が完全にわかれば設計する必要はない。設計問題が既に解かれ、得られた解がわかっている場合、新たな設計解を探すことも可能であるし、過去の設計解を現行の設計問題の設計解として用いることも可能である。

問題定義知が重要であることは建築学における活動や活動成果が示している。よい建物の定義、快適な建築環境の物理学的な数量による表現、安全な建築を担保するための性能指標など、建築学の様々な分野が問題定義のための知を提供している。これらは各分野の教義に基づくものである。

分析知の獲得についても、問題定義知と同様に、建築学が寄与している。設計描写（例えば、設計図書）から建築環境やそのために消費されるエネルギー、建設コスト、 $LCCO_2$ 、モーメント図などを導く方法は建築研究にによって精緻化あるいは（使いやすいという意味で）簡便化されつつある。

評価においては条件が満たされているか否かを判定される。計画、意匠、環境、構造、経済性などのすべての条件が満足される場合には、論理的な判定の仕方を知っていることが評価知をもっていることになる。各分野の教義に基づくそれぞれの条件をすべて満足することが論理的に不可能である場合、設計知には論理とは質が異なる知が必要となる。条件の取捨選択や軽重の塩梅の考慮など各教義をメタレベルで扱うための知が必要となる。

再定義知は問題定義知と共通する知を含む。現行の設計描写や設計描写が制作されて来た履歴から人工物が提供することを期待される環境の $\Delta_{S,ex}$ を導きだすための知である。建築デザインは、先に触れたように、輪郭の明確な問題ではない。設計解が得られるか否かは問題の再定義が適切になされるか否かに左右される。ここでの議論からは逸脱するが、デザインにおける創造的な発想には問題の再定義と設計描写の生成のトラザクションが寄与していると考えられる。

設計者は設計描写を闇雲に生成するわけではない。輪郭が明確であるか不明確であるかに関わらず、設計解であろうと設計者が考える設計描写を生成する方向で設計を進める。例えば、住宅を設計すべき問題において、発電所の設計描写を解として生成する設計者はいないと考えられる。設計描写を生成する操作の方向を定めることに寄与する知を構成知とよぶ。構成知がない場合、建築デザインは生成操作を任意に適用して設計描写を生成して検証する、単純な試行錯誤となってしまう。設計操作は何らかの知に基づいて恣意的に適用される。

現行の設計案（設計記述）が設計解ではないとき、設計描写のどこをどのように修正するか決定には構成知や分析知が寄与する。設計描写に基づいて環境描写を得る分析が演繹である場合、期待される環境描写と予測される環境描写の差異から設計描写を変更する方針を定めるには、分析の逆のプロセス（ f_a による演繹

を逆に適用したアブダクション)を可能にする分析知と構成知が用いられる。

形式知は生成を許す設計詞配置を限定する形式についての知である。必ず設計解となる設計言語を生成する形式が定義できれば、その形式に従って設計を進めれば何らかの設計解が得られることは明らかである。生成される設計詞配置が必ずしも設計解でなくとも、設計解を得るために吟味する設計詞配置の数(設計解の探索空間のサイズ)を減らすことができる。建築計画的観点による空間の隣接関係の制約や建築各分野が示す仕様に関する制約に関する知は形式知の一種である。この種の形式知は設計解の探索空間を小さくすると同時に設計の自由度を下げる(論理的には同じことだが、設計者にとっての印象は異なる)と言われる場合がある。その理由を踏襲することなく継承される形式なども形式知の一種である。一方、形態文法や図式文法のようにある形態のスタイルや空間構成の様式を明確に定義する形式知もある。設計詞配置を生成する文法を定義することにより、設計解の探索空間を小さくするものである。

これらの知が(論理的な意味で)適切に用いられている場合に設計は合理的である。

2.3 人工知能と設計

人工知能が設計において担う役割について、これまでの議論を踏まえて考察する。人工知能の基本アルゴリズムは探索(深さ優先探索と幅優先探索)である。解の候補を生成し、その候補が解でない場合には、解に近づくように候補を修正し、解であることが検証されるまで、生成と検証を繰り返す。この探索空間を減少させるために用いられるのが適用対象に応じて経験的に定められるヒューリスティクスである。また、探索の結果として得られた解が実際に解であることを保証するのは適切に定義された評価関数である。評価関数は分析知に対応づけられる。渡辺誠が都営地下鉄大江戸線飯田橋駅の設計において用いた手法は探索アルゴリズムに相当するものである。

2.3.1 遺伝的戦略を用いる設計

遺伝的戦略(genetic strategy)を用いた設計を例に、設計と人工知能アルゴリズムとの対応付けを行う。

遺伝的戦略を用いた設計では建物の結構が遺伝子としてコード化される。遺伝子コードは、通常、バイナリーコードの列(二進数)によって表わされる。遺伝子コードの各桁に結構のどの部分を割り当てるかが遺伝子コードによって扱える結構の可能な形式を定める。すなわち、コード化によって生成可能な設計詞配置が形式的に限定される(桁数が非常に大きい二進数を用いれば限定されない場合もありうる)。

各遺伝子の評価値が評価関数を用いて求められる。このプロセスは f_a を用いた分析に対応する。すなわち、ここで分析知が用いられる。遺伝的戦略の遺伝子は評価値の大小に応じて淘汰される。換言すれば、残すべき遺伝子が評価値の大小に応じて選択される。遺伝的戦略を用いた設計では淘汰と選択のプロセスに評価知が埋め込まれることになる。

選択された遺伝子は次世代の遺伝子を生成するために交配及び突然変異をされる。これは設計途上の設計詞配置を設計解に近づけるために修正する操作の一種である。すなわち、遺伝的戦略を用いた設計では交配と突然変異に構成知が埋め込まれている。

一般的な遺伝的戦略を用いた設計では問題の再定義はなされない。評価関数あるいは評価関数の構成が予め定められ、進化計算の途中でこれらを変更することはできない。再定義を可能にするために、設計詞配置と評価関数とを共進化させる方法が試みられている。

2.3.2 カオスやフラクタル構造を用いる設計

複雑系の数理に基づくパターンを設計詞配置がもつべき形式として与える設計手法がある。これらは複雑系の数理を形式知として設計に適用するものと考えられる。

3 参考文献

1. Alexander, C.: 1964, Notes on the Synthesis of Form, Harvard University Press; 形の合成に関するノート(稲葉武司訳), 鹿島出版会, 1978.
2. Akin, Ö.: *Psychology of Architectural Design*, Pion, 1986.
3. Rave, J., Wenger, E.: *Design Thinking*, The MIT Press, 1987; デザインの思考過程(奥山健二訳), 鹿島出版会, 1990.