

アブダクションに基づく設計者支援環境の基本構想*

下村 芳樹 *1, 吉岡 真治 *2
武田 英明 *3, 富山 哲男 *4

Fundamental Plot for a Designer Support Environment based on Abduction

Yoshiki SHIMOMURA, Masaharu YOSHIOKA, Hideaki TAKEDA
and Tetsuo TOMIYAMA

This paper describes a new design support system that supports conceptual or creative design by dynamically integrating knowledge in different design domains. We argue that abduction for integrating theories can be a basic principle to formalize such design processes. Based on this principle, we propose Universal Abduction Studio, a design environment in which designers combine different theories to arrive at better design. In this new approach to computational support of conceptual design, the system should offer various types of abductive reasoning from which designers can select an interesting design method. We also discuss technologies to implement UAS and in this paper we propose to use analogical reasoning as abductive reasoning to discover relationships between knowledge from different sources. We demonstrate that a prototype system can discover a new idea in a design example taken from a real design activity.

Key Words: Computer Aided Design, Design, Design Engineering, Abduction, Creative Design

1. 緒 論

人工物に対する要求の高度化・複雑化に伴い、設計者に対する負担は一層大きなものとなりつつある。近年のネットワーク技術の発展と浸透により、多種多様な知識が計算機上に蓄えられるようになったが、それらの知識を設計において効果的に利用するためには知識管理の枠組みが必要であることが指摘されている^①。富山らは概念設計や創作的設計を支援する計算機システムの開発を目的として、設計作業において重要な役割を果たしている設計知識および知識操作に関する研究を行っている^②。そこでは設計者による設計を、個々の設計分野ごとに領域化された固有の知識群（領域知識）を用いて設計解を生み出す場合と、異なる領域で行われている設計の事例あるいは当初無関係と思われていた領域知識から新たな知見を得て設計解を生み出す場合に分類している。また、既存の設計知識の組合せのみでは得られないような創作的な設計解を生

み出す設計においては、後者の場合のように、ある問題を解くために一見無関係に思われるような様々な領域知識を統合し再構成することが重要であり、この知識拡張においてはアブダクション (Abduction) と呼ばれる推論形態がトリガーとして働くことを指摘している。異領域の知識を統合するために知識を互いに関連付ける方法は複数存在すると考えられるが、本研究では知識の関連付けに関する複数の手法を総合的に使用する環境を提供することにより、設計者による創作的設計を支援する知識統合環境の概念を提案する。

2. 設計における知識操作

本研究では設計者の設計活動を計算機により統合的に支援することを目標としている。そこでまず本章では前提となる設計活動のモデルについて説明する。

設計は大局的には人工物に対する機能的な要求仕様からその人工物を創造するに足る属性的な仕様を決める過程である^③。一般に属性から機能への写像関係は一對多関係であると言われる。すなわち、所与の属性群からその属性的性質を有する人工物の機能を推定することは演繹的操作で可能だが、その逆は困難である。設計はこの逆の操作に相当するため、単純な演繹的操作のみでは実現ができない。このため、設計過程のモデル化において非演繹的操作を用いることが考えられている^{④⑤}。武田らもアブダクションと演繹に基づく

*原稿受付 XXXX年XX月XX日。

*1 正員, 東京大学 (〒277-8568 千葉県柏市柏の葉5-1-5)

E-mail: simomura@race.utokyo.ac.jp

*2 正員, 北海道大学 (〒060-0814 北海道札幌市北区北14条西9丁目)

*3 正員, 国立情報学研究所 (〒101-8430 東京都千代田区一ツ橋2-1-2)

*4 正員, デルフト工科大学 (Mekelweg 2, 2628 CD Delft, The Netherlands)

設計過程を提唱しており⁶⁷⁾、本研究はこの設計過程モデルを基本的に踏襲する。

2.1. 設計におけるアブダクション

本研究の目的は、設計における創造の過程を表現し、これに対する支援を計算機の利用によって実現することである。武田らは、論理に基づいた設計過程の形式化により、計算可能な設計過程モデルを提案している⁶⁸⁾。まず、設計を次のように定義する。

$$Ds \cup Ko \models P \dots\dots\dots(1)$$

Ds は設計対象の記述、 Ko は対象に関する知識、 P は設計対象の性質の記述である。この式は、設計対象の記述と対象に関する知識から設計対象の性質の記述を論理的に導出できることを示している。ここで、設計とは設計対象の性質の一部を要求仕様として、対象に関する知識を用いて、その性質を持つ対象の記述を得ることである。この過程は論理的には演繹過程ではなく、アブダクション過程である。

このモデルでは、創造的設計の中心過程、すなわち新しいモノやアイデアを創造する過程では、アブダクションが重要な役割を果たす。C. S. Peirce によって提案されたアブダクションは、定理群から公理を発見するための論理的過程である⁶⁹⁾。アブダクションとは演繹の逆過程とする解釈も存在するが、それでは演繹と等価な操作になることが知られており、Peirce が主張するアブダクションの性質である新しい事項の発見が実現できない。このため、アブダクションとは知識を拡張するような推論であるという解釈を用いる⁷⁰⁾。この場合、新しい仮説発見が可能になる。

現実問題として大量の知識を取り扱わねばならない設計のような問題領域において、実現可能な知識の拡張とは、独立に存在する知識群を必要に応じて統合しつつ用いることである。アブダクションはこの知識統合を誘引するトリガーとして働くと考えられる。

以上のように考えると、設計におけるアブダクションとは、ある設計課題を満たす設計解が既存の知識のみからは導出不可能であるような場合に、既存の知識を用いて新たな統合された知識を形成することにより、設計解を発見するプロセスであると考えられる。ここで言う知識の統合は、領域に対して単に知識をつけ加えるだけでなく、知識の再構成や修正操作を含む。まず、いま利用する設計知識 $Ko_1 \dots Ko_n$ とする。統合された知識を Ko はその和、 $Ko_1 \cup \dots \cup Ko_n$ とするのは不十分である。このままでは知識相互の関連性がなく、知識を拡張したことになる。知識をお互いに

関連付けるために、異なる知識中の概念を同一視するような操作が必要となる。これを実現するような、 Ko_1, Ko_2, \dots, Ko_n に関して妥当な置換 ϕ を用意すると、

$$Ko' = Ko_1 \phi \cup \dots \cup Ko_n \phi \dots\dots\dots(2)$$

となる。これが統合された知識である。

アブダクションはこのように統合された知識を用いて設計解を提案可能であるかどうかを判定し解を導出するが、変換 ϕ はアブダクションそのものからは導出されない。変換は個々の知識の内容に基づく知識間の関係から導き出されるものである。

2.2. 知識間の関連性の発見

前節で述べたように、設計においては多様な視点・領域からの知識が存在し、それらの知識が統合され利用されることで創造的な設計が実現されると考えられる。そしてこのためには多様な設計知識間の関連性を知る必要があるが、この関連性には多様なものが考えられる。設計で用いられる知識の多くは物理世界の法則に由来するため、この物理法則のレベルで統合するという方法がある。これが富山らの提唱するメタモデルである⁷¹⁾。この方法は強力であるが、物理法則と密接に結びついた領域の知識に限られる。

別の方法としては様々な知識を統合するオントロジーを利用する方法がある⁷²⁾。オントロジーにより、知識を統合的に管理することができるが、オントロジーの構築は多大な労力を必要とし、現実的には概念がよく整理された領域に限られる。

上記二つの方法より広範な適用範囲のある方法として、本研究ではアナロジー (Analogy) に基づく推論⁽²⁾⁽³⁾の利用を試みる。アナロジーは、先の二つの方法のように概念の持つ意味そのものを扱うのではなく、関係が作る構造のみを利用して概念を関係づける。このため、意味的には間違っている推論をする可能性がある反面、意味的な関係がわからない場合においても適用可能である。本研究ではアナロジーをこのような意外性のある関係を発見する手法と位置づけ、既存の方法では見出せない関係発見の手法として利用する。

3. Universal Abduction Studio (UAS)

3.1. UAS の基本構想

本研究は、個別方法論に基づく設計支援ではなく、プリミティブな知識操作の支援を行うための基本的枠組みを提案するものであり、これを Universal Abduction Studio (以下、UAS と略記) と名づける。

また、設計の自動化を目的とするのではなく、設計者とシステムの動的なインタラクションによる設計課題の解決を支援する協調的な枠組みを目指す。UAS は言わば工房であり、設計者がそこに用意される多様なツール群を利用し、それらを自由に組み合わせて用いることによって自身の思考を拡張し、設計課題を解決する。ここでの協調とは、設計者が設計支援システムの支援内容に様々なレベルで介入し、システムの動作を確認しながら設計を進めることを意味する。

図 1 に UAS の概念図を示す。UAS は、ワークスペース、知識ベース群、知識統合モジュール群からなる。設計者は解決を必要とする設計課題、あるいは暫定的な設計案をワークスペースに配置し、その全体や部分に対して知識ベースに格納されている知識を適用しながら課題の解決を目指す。様々な知識を一つの設計に適用するためには知識の統合が必要となるが、このとき知識統合モジュール群から適切な知識統合モジュールが選択的に使用される。このようにして設計の各場面で必要な知識を、そこでの適切な知識に変換して適用することで UAS における設計は進行する。

3.2 設計知識の表現

設計においては、多様な設計対象および設計過程の知識を扱う必要がある。設計対象の表現において、単一の対象モデリング方法だけでは人工物を適切に表現することは困難であり、目的に応じた多様なモデリング方法の利用が必要であることが指摘されている。

設計対象の知識に比べ、設計過程の知識はより不完全である。現状存在する設計過程の知識と呼ばれるものは、個別分野の教科書的设计方法や事例的设计方法がほとんどである。すなわち対象とする分野や使用されるモデリング手法と混在して記述されることが多い。

このような断片的な設計に関する知識を取り扱うために、本研究では**設計知識文書**という概念を導入する。設計知識文書とは、設計に関する情報の断片であり、その中には設計対象と設計過程のそれぞれに関する知識が含まれる。また、人間向きの記述とコンピュータ向きの記述の両者の表現を含みうる。一つの設計知識文書は必ずしも上述のすべての要素を持つ必要はなく、ある一部の要素を含むだけでもよい。

このほか、モデリング方法そのものに関する知識や異なるモデリング方法間関係を記述するオントロジー⁴⁴⁾を知識として用意する。設計知識文書はオントロジーと背景にあるモデリング方法とともに解釈されることで他の文書と組み合わせた処理が可能になる。

3.3 アナロジーに基づく知識統合

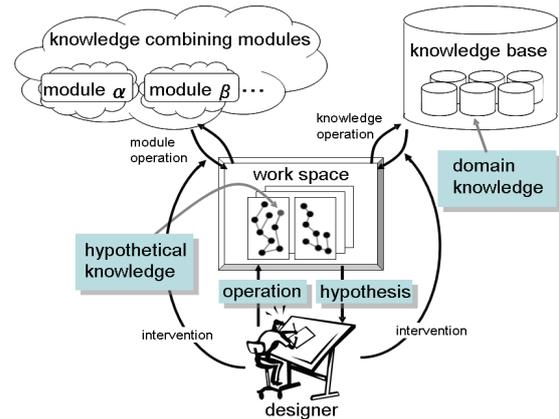


Fig. 1: Basic Concept of the UAS

アナロジーは、人の柔軟な思考を支える有力なメカニズムの一つであると考えられている。アナロジーによる発想とは、ある分野（**ターゲット領域**）における未知のアイデアを、その分野とは異なる別分野（**ベース領域**）との類似性に基づき、ベース領域の要素をターゲット領域に転写することで獲得するものである⁴⁵⁾。すなわち、このアナロジーの過程はこれまでに述べてきた創造的设计のための知識統合のメカニズムとして用いることが可能である。そこで本研究では、アナロジーによる知識統合を行う上で必要である知識間の類似性判定に、各領域に属する概念間の対応関係を用いることを考える。この時、領域知識の統合とは、各領域における概念間の対応関係に基づいた知識の変換により実現するものとする。

3.1 節にて、UAS は複数の知識統合モジュールを有することを述べた。この各知識統合モジュールが行う知識操作の違いが、UAS におけるアナロジーの種類に対応する。この時、知識統合モジュールごとに類似性判定の基準は異なるので、その基準に対応する知識表現形式が必要となる。例えば、知識の位相構造上の類似性を利用したアナロジーを行うには知識はグラフ構造に、ルールの類似性を利用したアナロジーを行うには知識はルール形式に表現されなければならない。

さらに、実際に記述された知識に対してこのアナロジーを適用するためには、先に述べたように記述に対して様々なモデルに基づく知識の同定を行う必要がある。以上のことを考慮すると、UAS における知識利用のイメージは図 2 のようになる。図 2 において、底面に位置する楕円がある知識表現の形式を表す。すなわち一つの楕円は、ある知識表現形式が指定された状態を表す。次に、知識表現形式の上存在する複数の壁が、記述と実体を結びつける個々のモデルを表し、個々の知識はそれぞれのモデル上に存在している。

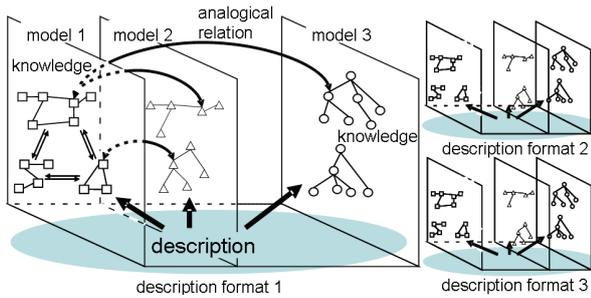


Fig. 2: Conceptual Image of Universal Knowledge Integration

UAS の各知識統合モジュールは、一つの知識表現形式上の知識群を対象とするが、その中にはさらに同じモデル上の知識の統合と、異なるモデルに跨る知識統合の二つが考えられる。異なるモデルに属する知識間では、知識を構成する概念、表現に用いられる語彙が異なり、また同じ語彙であっても意味が異なる場合があり得る。この場合、有効な知識を得るためには、語彙レベルの類似性を利用するなど、様々な類似性を利用することが必要である。

3.4. UAS アプローチにおける設計支援手順

図 3 に UAS において仮定する設計支援の手順を示す。先に述べたように、UAS は設計の自動化を実現するものではなく、設計の各ステージで設計者の選択や意思決定を促すことにより設計の進行を支援する。

(1) 知識統合モジュールの選定

対象としている設計において、どのように知識を獲得したいのかに応じて、適切と思われる知識統合モジュールを設計者が選択する。

(2) 類似知識の選定

選択された知識統合モジュールは、知識ベース群から、ワークスペースに配置されている設計対象に関する知識や設計課題に類似すると思われる知識群を探索し、設計者に対して適用可能な知識群候補として提案する。設計者は、提案された知識群候補の中から使用したいものを選定する。この時、設計対象と同じモデルに基づく知識群を選択することにより知識間に類似性を発見しやすく、信憑性の高い知識を得やすくなる反面、創造性の高い知識が得られる可能性は下がる。他方、設計対象と異なるモデルに基づく知識群を選択すれば、得られる仮説の信憑性は下がるが、創造的な知識の獲得を期待できる。すなわち、設計対象の属するモデルと、比較対象の属するモデルの概念的な距離の大小が、得られる知識の信憑性と創造性に影響する。

(3) 異なる領域概念の対応関係の生成

知識統合モジュールは、ワークスペース上に置かれた設計対象に関する知識と、(2)で選定された知識群との間にアナロジーに基づく対応関係を生成する。

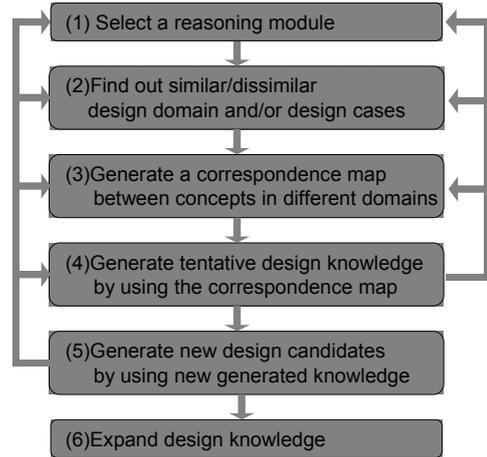


Fig. 3: Design Flow in the UAS

(4) 対応関係による仮説的知識の生成

さらに知識統合モジュールは、ステップ(3)で作成した対応関係に基づいて、仮説的知識を提案する。仮説的知識とは、(2)で選定した知識の一部を、(3)で作成した対応関係に基づいて、設計対象へと転写したものである。この段階で有効な仮説的知識が得られなかった場合は、ステップ(1)～(3)の各段階へ戻り、そこに示される設計作業を繰り返す。

(5) 設計解候補の生成

ステップ(4)で生成された仮説的知識を用いて、設計者は設計解の候補を生成する。ステップ(4)で生成された仮説が最終的に新たな知識として採用されるか否か、すなわち UAS でいう知識統合が成立するか否かは、本ステップで生成される設計解候補に対する評価の結果によって決まる。この評価により設計解候補が利用可能であると判断された場合は、その時点で設計を終了することができる。設計解候補が設計者の意図する要求を満たさない場合、あるいは、他の設計解候補の導出を継続する場合は、設計者はステップ(1)～(4)の各段階へ戻り、再び設計作業を繰り返す。

(6) 知識の拡張

(5)で設計解候補が設計解として利用可能であると判断された時点で、このとき用いられた仮説的知識は有効な新知識と成りうると判断され、知識ベースへ追加される。結果として、他の領域知識との部分的な統合が行われ、設計対象に関する知識が拡張されるが、追加された知識の妥当性はその後の複数の設計によって検証される必要がある。

4. プロトタイプ UAS

著者らはこれまでの議論に基づき、UAS を構成する一つの知識統合モジュールとして、知識の構造に基づく類推機構を Windows2000 環境下の Java 言語により試作した⁶⁶。本プロトタイプは UAS で提案する全

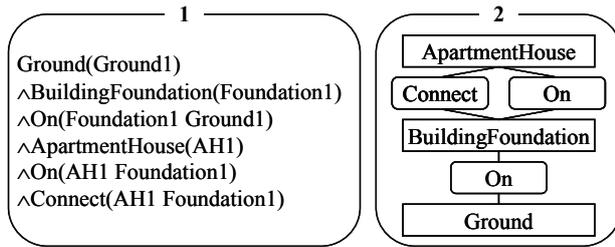


Fig. 4: An Example of Graphical Representation

ての概念、機能を実現したものではないが、知識表現形式として一階述語論理を利用し、グラフ構造で表現した知識間の関係に対して構造的アナロジーを利用するという知識統合支援を試行した。従って、複数の異なる関連付け方法を目的に応じて適切に選択する機構に関しても、このプロトタイプは対象としていない。

4.1. グラフ構造による知識表現

先にも述べたように、本知識統合モジュールは、グラフ構造で知識を表現するための知識表現形式として述語論理を採用している。述語論理は形式言語の一種であり、計算機が利用可能な知識の表現のための言語である⁴⁷⁾。述語論理は基本構成要素として対象と述語を持つ。対象は文中に現れる主語や目的語に対応し、述語は動詞や形容詞などに対応する。「自動車の車体の色が白い」という時、対象とする自動車を自動車 1 と表現すると、述語論理では「自動車(自動車 1) ^ 車体の色(白, 自動車 1)」と表現する。また、述語は一つの対象の性質を述べるだけでなく、二つ以上の対象間の関係を述べるにも用いられる。例えば「列車 1 が貨車 2 を引いている」時、この関係は述語論理では「引いている(列車 1, 貨車 2)」と表現する。

以上の述語論理の記法に従い、設計記述を述語論理で具体的に表現する手法について述べる。まず設計対象の実体の宣言を、引数をひとつ持つ一階述語により表現する。例えば、「あるアパート」という実体を記述するためには、述語 *ApartmentHouse* に対し、そのインスタンスとなる実体 *AH1* を引数に与えて *ApartmentHouse(AH1)* と記述する。これは、*AH1* という実体は *ApartmentHouse* であるということの意味する。次に、対象間の関係、対象の属性など二種以上の要素の関係を記述するために、二項以上の引数を取る述語を利用する。例えば、「アパートの上に建築基礎がある」という関係は、

On(AH1, Foundation1) ^ ApartmentHouse(AH1) ^ BuildingFoundation(Foundation1)

と記述する。さらに、「地面の上に建築基礎があり、アパートが建築基礎と接続されている」といった例は、図 4-1 のように記述可能である。各述語をノード、二

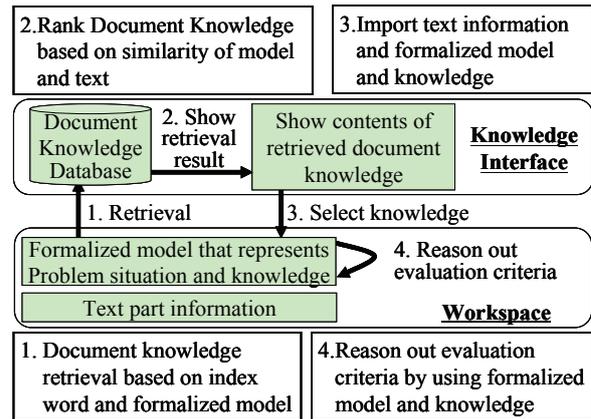


Fig.5: Conceptual Structure of DMaPS⁴⁸⁾

項以上の述語の引数の関係をアークとみなすことにより、グラフ構造として表現できる (図 4-2)。

4.2. プロトタイプ UAS のアルゴリズム

本プロトタイプでは、述語論理によって記述した要素および要素間の接続関係によって表現される知識間の類似性を判定する具体的なアルゴリズムとして、吉岡らの DMaPS (Documentation Management system for Problem Solving)⁴⁸⁾の類似性判定方法を採用した。DMaPS は、既存の文書を参照しながら新しい文書を作成し、それを用いて問題を解決する過程を支援するシステムである。DMaPS の構成を図 5 に示す。DMaPS には主に、ユーザが直面している問題領域やそこで決められた情報を表現するワークスペースと、その状況に応じて、役立つと考えられる知識を文書知識データベースから検索し、提供する部分から構成される。DMaPSにおける作業は次の手順で行われる。

- (1) 文書知識の検索
- (2) 文書知識のランキング
- (3) 文書知識の選択
- (4) 形式的知識の適用

DMaPS では上記(2)の文書知識のランキングにおいて、ユーザが保持する問題を表現する論理式を検索キーとして検索キーを含む文書を探し出し、語の文書中での頻度 (tf: 文書中での語の重要性) と検索語を含む文書数の逆数 (idf: 語の偏在性) を利用した *tf · idf* 法⁴⁹⁾に基づくスコアリングを行う。次に、ユーザが保持する問題が一階述語論理で表現されている場合には、モデルの類似度を使った検索を行う。この類似度は、問題領域を表すモデルにおける項と検索対象の文書知識のモデルにおける項の間に、ある写像関係を設定した場合に、その写像関係に基づいて設定される論理式の対応関係の数により判定する。この類似度の計算手順を図 6 に示すアパートと工場の建築基礎設計に関する文書知識のモデルの例を用いて説明する。

(2a) ユーザの問題を表すモデル中で、その問題に最も関係すると考えられるモデル中の論理式（着目部分：Target）と比較対象となる文書知識のTargetの論理式を比較する。この時、二つの論理式の対応する項に注目し、順番に写像関係を設定する。図6の例では、*Foundation1*と*Foundation2*という2つの項の間の写像関係を設定する。

(2b) ユーザが保持する問題を表現するモデルと文書知識のモデルから、写像関係を満たす項について、出現順序が対応し、かつ、同述語を持つ論理式の組を探索する。この時、組となる二つの論理式の対応する項に注目し順番に写像関係を設定する。図6の例では、*On(AH1 Foundation1)*、*On(Foundation1 Ground1)*に対し、*On(Factory2 Foundation2)*、*On(Foundation2 Ground2)*が見つかり、各々から、*AH1*と*Factory2*、*Ground1*と*Ground2*の間に写像関係を設定する。

(2c) 手順2bを満たす候補が複数存在する場合には、複数の対応関係を並列に処理する。また、各々の対応関係の生成では、2bの手続きによって、新たな項の写像関係が見つからなくなるまで、項の写像関係を生成する。また、各項の写像関係には、何ステップ目の手順2bによって発見されたものであるかを付加情報として記載する。図6の例では、*Ground1*と*Ground2*の対応関係から、*SandStrata1*と*SandStrata2*が2ステップ目の対応関係として得られる。

上記の方法により計算した写像関係に基づき、文書知識のモデルを変換し、対応するモデルの論理式の数をTargetに対する関連性に強弱を反映した形で数え、類似度とする。具体的にはTargetに含まれる論理式に対応関係が存在する場合には貢献度を5とし、Targetに含まれていない論理式については、2bで最も少ないステップ数で見つかった項を持つ論理式ほど、Targetとの関連性が強いと考え、そのステップ数(n)に基づく貢献度 $1 / (n+1)$ を各々の論理式の関連性の指標とする。類似性の指標としては、類似する論理式の個数という絶対値による指標と、類似する論理式の内、対応がつく論理式とつかない論理式を相対的に評価する指標を用いる。また、複数の項の写像関係が考える場合には、最も評価の良いものを提示する。

図6の例では、下線部の論理式が写像により対応付けられ、Targetに対応する論理式が存在するので+5。Targetに含まれる項(*Foundation1*)を含む残りの二つの論理式は、全て対応がとれているので+ $(1 * 2)$ 、Targetから1ステップの関係で対応関係が導かれた項(*AH1*、*Ground1*)を含む三つの論理式（既に対応関係

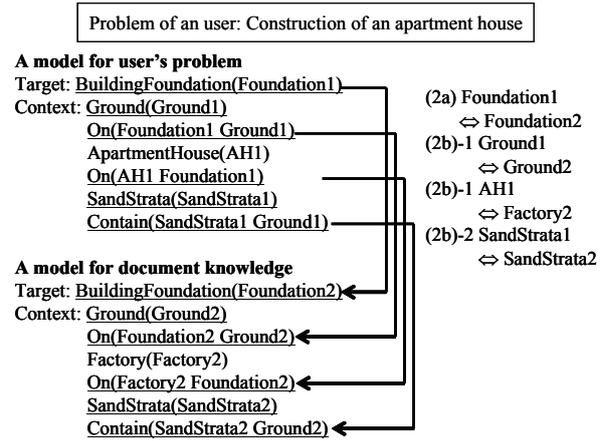


Fig. 6 : An Example of Structure Mapping

を数えた *Foundation1* を含む論理式は除く)は、(*Factory AH1*)という論理式以外の二つの式の対応がとれるため+ $(1 / 2 * 2)$ 、Targetから2ステップの関係で対応関係が導かれた項(*SandStrata1*)を含む一つの論理式（既に対応関係を数えた論理式を除く）についても対応がとれたことになるので+ $1 / 3 * 1$ 。従って、この絶対的指標による類似性の計算値は以下となる。

$$5 + 1 * 2 + (1 / 2) * 2 + (1 / 3) * 1 = 8.33 \dots\dots\dots (3)$$

同様に、絶対的指標の最大値は、完全に対応がとれた場合の以下となる。

$$5 + 1 * 2 + (1 / 2) * 3 + (1 / 3) * 1 = 8.83 \dots\dots\dots (4)$$

よって相対的指標では、絶対的指標をその最大値で割り算した以下となる。

$$8.33 / 8.83 = 0.94 \quad (5)$$

4.3. 知識統合モジュールの実行例

4.1節で述べた知識表現形式に従い表現した設計知識に対して、4.2節の(2)文書知識のランキングの手法を用いることにより、異なる知識を構成する要素間の対応関係を発見し、仮説的知識を生成する試行実験を行った。図7は本モジュールの実行画面の一部で、自動車用ホイールキャップの補助ホルダガイドリブの設計手順書とホルダガイドリブの位置決定の手順書という二つの異なる設計手順書における類似性を判定し、知識間の対応付けを行った様子を示す。図右側がベース領域（補助ホルダガイドリブ位置決定手順書）、左側がターゲット領域（ホルダガイドリブ位置決定手順書）の知識である。本例では、ハンパ頂点径を決定する際、割れに関する剛性を考慮に入れる必要性がベース領域の知識として与えられていた。

この二つの知識に対し、ベース領域とターゲット領域を設定後、設計者はまずベース領域に対して順位付け計算用の基準要素となるノードを選択し（4.2 節の(2a)）、システムはこのノードを含む論理式（部分グラフ）に対応するターゲット領域上の部分グラフを全て発見する（4.2 節の(2b)）。続いてシステムは 4.2 節の(2c)の論理式の比較手法を用いて導出した部分グラフに得点による順位付けを行い、得点の高い順にマッチした部分グラフを設計者に提示する。図 7 では以上の結果として、剛性（割れ）と意匠剛性、および、距離とハンブ頂点径がリンクで結ばれている。設計者は、このリンクの構造を解釈することにより仮説的知識を得る。この場合いくつかの解釈が考えられる。例えば、ハンブ頂点径を決定する際には、剛性（割れ）だけではなく意匠剛性も考慮すべきである、あるいは剛性（割れ）と意匠剛性は類似の概念であるという解釈が可能である。

一方、同じくベース領域におけるキャップ中心と風穴相関線間の「結ぶ」という述語、ターゲット領域における風穴相関線と意匠線間の「結ぶ」という述語間の対応関係も生成された。設計者はこのリンクの構造を解釈した結果、意匠線を決する際には、風穴相関線のみではなくキャップ中心位置も同時に考慮すべきであるという仮説知識を得、その後、本仮説は設計知識として妥当なものであるという結論に至った。

このように本例では、設計者の解釈の結果得られた複数の仮説的知識がホルダガイドリブ位置を決定する上で妥当なものであると評価された結果、構造的類似性に基づく推論を用いて複数の知識を統合し、新しい発見を得られる可能性を示すことができた。一般に、設計現場における知識は個々の設計者により離散的に個別領域として蓄積されることが常であり、一方その共有や統合を日常の設計業務と並行して進めることは容易でない。しかしながらそれらの設計知識間を例えば類似性を鍵として意識的に統合することにより、個々の設計課題に対して新たな設計知識を得ることが可能であることを本例は示している。すなわち既知の知識だけでは導出が困難であるような設計解、すなわち創造性の高い設計解を得ることが期待できる。

4.4. プロトタイプに見る課題

本章で述べたモジュールによる知識統合支援の試行を行った結果、以下の課題が明らかになった。

(1) プロトタイプシステムの実装レベル

今回実装したプロトタイプシステムは、UAS の中心的操作である仮説的知識生成のみにシステムの範囲

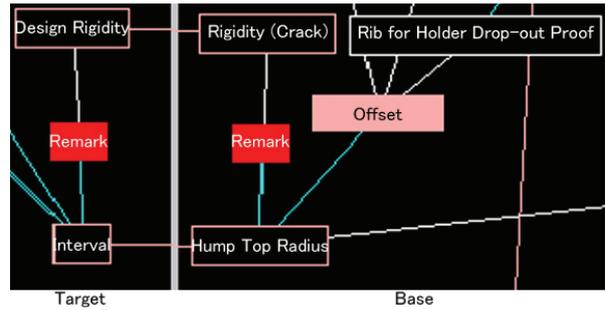


Fig. 7: Structure based Mapping among Domain Knowledge

を限定している。図 1 における多様な操作を実現するワークスペースやその他の知識結合モジュールなどは実装していない。これらの実装には設計課題の計算機上における表現と操作に対する形式化が必要である。

また 3.4 節に述べた設計支援手順のうち (4) までが実装されており、残る (5) と (6) については未実装である。(4) においても、設計者による解釈をもって最終的な仮説的知識を生成している。今後これらについても、追加実装を行うことが必要である。

(2) 語彙間の柔軟な対応付け

本知識統合モジュールでは、類似性の判定時に、ノードにあたる語彙の完全一致を求めている。このため、アナロジーの適用される範囲が狭く、異領域知識利用の観点で柔軟性に乏しい。今後、語彙間の類似性を柔軟に判定する手法等の開発により、この問題は解決可能であると考えられる。

(3) 語彙の意味的な解釈を行う類似性判定

また、単に語彙の表現レベルで一致では間違った類似性判定が起こる可能性がある。信憑性の高い類似性判定を行うためには、語彙の意味的な解釈を行い、語彙の意味に基づいた類似性判定が必要である。

(4) 異なるモデルに属する知識間の類似性判定

3.2 節で述べたように様々なモデルに基づく設計知識が存在していると考えられる。そして、異なるモデル上に存在する設計知識間の類似性を柔軟に判定することにより、有効かつ新規性の高い知識を獲得することにつながると考えられる。これに対して、本論文で述べた知識統合モジュールが実現しているのは、同じモデル上の知識の統合のみである。今後、異なるモデル上に存在する設計知識に対する拡張も必要である。

(5) 知識の組み合わせ爆発への対応

3.3 節で述べたように、UAS では、様々な知識を統合することにより、新規性の高い知識の生成を目指している。そのため、単純に、知識統合を繰り返していく場合に、組み合わせ爆発が発生する可能性がある。また、UAS で作成する知識は、あくまでも仮説的知識であり、その妥当性を確認せずに、更なる仮説生成

を行うことの有効性に疑問が残る。よって、UAS では、知識統合を単純に繰り返すのではなく、知識の妥当性検証のプロセスを適宜組み合わせながら問題解決に当たることが必要である。

5. 結論と展望

創造的設計を支援する枠組みとしての Universal Abduction Studio の概念を提案し、その知識統合モジュールの一つの例として、構造的類似性に基づく推論機構のプロトタイプ例を紹介した。さらに、当該プロトタイプシステムを用いて実際に知識統合支援を行った結果、明らかになった課題を整理し、今後の設計者支援環境開発の指針を示した。次期プロトタイプの開発方針は、オントロジーの概念を導入し、語彙の意味的解釈に基づくより信憑性の高い類似性判定を実現することである。今後、継続的なシステム構築と例題への適用を通じて、提案手法の有効性を検証する。

謝辞

本研究のシステム開発に協力して頂いた坂井宏充氏（現 NTT ドコモ）、野間口大氏（現大阪大学）に感謝の意を表す。また、本研究の一部は、文部科学省科学研究費、基盤研究(B)(1)、一般、14380170、「創造的アブダクションによる設計支援の研究—Universal Abduction Studio の開発—」の補助を受けた。

文献

- (1) Dieng, R., Knowledge Management and the Internet, *IEEE Intelligent Systems*, (2000), 14-17.
- (2) Tomiyama, T., *Proceeding of the Symposium on Science of Synthesis* (in Japanese), (2001), 7-60, Nihon Gakujyutu Shinkoukai Mirai Kaitaku Gakujyutu Kenkyu Suisinn Kikou.
- (3) Yoshikawa, H., The Preface of General Design Theory - An Axiomatic Approach for the General Design Theory - (in Japanese), *Seimitsu Kikai (Journal of the Japan Society for Precision Engineering)*, Vol. 45, No. 8, (1979), 906-912.
- (4) Roozenburg, N. F. M. and Eekels, J., *Product Design: Fundamentals and Methods*, (1995), John Wiley & Sons, Chichester.
- (5) Gero, J. S. and Kazakov, V., Using analogy to extend the behaviour state space in creative design, in Gero, J. S. and Maher, M. L. (eds.), *Computational Models of Creative Design IV, Key Centre of Design Computing and Cognition*, (1999), 113-143, University of Sydney.
- (6) Hayashi, K., et al., Analysis of Design Process and Logical Formalization (the 3rd Report) (in Japanese),

- Proceeding of the Regular Spring Meeting of the Japan Society for Precision Engineering*, (1989), 7-8.
- (7) Takeda, H., et al., A Computational Model of Design Process and Design Simulation (in Japanese), *Journal of the Japan Society for Artificial Intelligence*, Vol. 7, No. 5, (1992), 877-887
- (8) Yonemori, Y., *Peirce's Semiotics* (in Japanese), (1981), Keisoshobo.
- (9) Aliseda, A., Abduction as Epistemic Change: A Peircean Model in Artificial Intelligence, in Flach, P. and Kakas, A. (eds.), *Abductive and Inductive Reasoning, Applied Logic Series*, (2000), Kluwer Academic Press.
- (10) Tomiyama, T., From general design theory to knowledge-intensive engineering, *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing (AIEDAM)*, Vol.8, No. 4, (1994), 319-333.
- (11) Gruber, T. R., *A Translation Approach to Portable Ontology Specification*, Vol. 5, Issue 2, (1993), 199-220.
- (12) Bhatta, S., et al., Innovation in analogical design: A model-based approach, In Gero, J., ed., *Artificial Intelligence in Design*, (1994), 57-74.
- (13) Bhatta, S. and Goel, A., From Design Experiences to Generic Mechanisms: Model-Based Learning in Analogical Design, *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, special issue on machine learning in design, Vol.10, (1996), 131-136.
- (14) Sekiya, T., et al., Realization of an Integrated Design Support Environment using Ontology, *Journal of the Japan Society for Artificial Intelligence*, Vol. 14, No. 6, (1999), 1051-1060.
- (15) Special Issue on Analogical Reasoning, *Journal of the Japan Society for Artificial Intelligence*, Vol. 17, No.1, (2002).
- (16) Sakai, H., et al., H., Development of Universal Abduction Studio (the 2nd report) (in Japanese), *Proceeding of the Regular Spring Meeting of the Japan Society for Precision Engineering*, (2003), 23.
- (17) Furukawa, K., et al., *Inductive Logic Programming* (in Japanese), (2001), Kyoritsu Shuppan.
- (18) Yoshioka, M., et al., Research for Knowledge Management fusing Document Information and Formalized Knowledge (the 1st report) (in Japanese), *Proceedings of the Regular Meeting of the Japan Society for Artificial Intelligence*, (2002), CD-ROM.
- (19) Baeza-Yates, R. and Ribero-Neto, B., *Modern Information Retrieval*, (1999), Addison-Wesley.