

論証の枠組みに基づく設計意図の獲得

Design Rationale Capture Based on Argumentation Framework

加藤義清*1 堀浩一*2
Yoshikiyo Kato Koichi Hori

*1宇宙航空研究開発機構
Japan Aerospace Exploration Agency

*2東京大学先端科学技術研究センター
Research Center for Advanced Science and Technology, University of Tokyo

Although the importance of recording design rationale has been recognized, it has not been widely accepted in practice. Major reasons hindering the acceptance of such practice are 1) the cost associated with capturing design rationale, 2) failure to perceive the direct benefits of describing design rationale, and 3) the difficulty of utilizing captured design rationale. This study aims to overcome the capture bottleneck of design rationale by employing computable design rationale representation. In this paper, we propose a computational model of design rationale based on a formal argumentation framework, report an experiment of describing a design rationale in NHL knowledge representation language and applying the argument generation functionality of the legal reasoning system New HELIC-II, and present a prospect of a system for design rationale capture based on the proposed framework.

1. まえがき

大規模・複雑システムの設計・製造・運用を円滑に行うためには、関係者の間でシステムに関する知識が十分に共有されている必要がある。システムに関する重要な知識の一つとして、設計意図 (design rationale) が挙げられる。設計意図は設計者間で互いの設計を良く理解する為に重要であることは当然であるが、設計以降のフェーズにおいても関係者がシステムを理解するのに役立つ。

設計意図が重要な設計知識の一部であることが認識され、様々な研究がなされてきたが [Moran 96], 現場において設計意図を記録し、活用する活動が必ずしも広く普及していないのが現状である。その原因としては、1) 設計意図を記述するコストが高いこと (コスト), 2) 記述をしても即時的な効果が得られないこと (短期的効果), 3) 記録として残しても有効な活用が難しいこと (長期的効果) 等が挙げられる。特に、1) に関して、設計意図の記述のコストの高さについていくつも報告がなされている [Conklin 91, Buckingham Shum 94, Shipman 99]. これは、エキスパートシステムにおける知識獲得のボトルネックや、知識マネジメントにおけるキャプチャーボトルネック (capture bottleneck)[Motta 00] 等と同種の問題である。

本研究は、計算可能な設計意図表現による短期的効果の向上を狙い、それに基づく設計意図獲得支援の枠組みの構築を目指している。本稿では、論争の枠組みおよびシステム工学におけるシステム設計評価の考え方にに基づき、設計意図表現の枠組みである Argumentative Design Rationale Framework (ADRF) を提案し、その計算可能性を確認するための設計意図記述実験について述べた後、提案する枠組みに基づく設計意図獲得支援システムへの展望を示す。

2. 論証の枠組み

本節では本研究が対象とする論証の枠組みの概要を示す。本節の内容は [Sartor 93] に基づいたものであり、後述の実験で使われる法的推論システム New HELIC-II も同じ枠組みを採

用している。

まず、ルールは次の形式で記述される。

$$\mathbf{n} : p_0 \leftarrow p_1 \wedge \dots \wedge p_n \quad (1)$$

ここで、 \mathbf{n} はルール名、 p_i はリテラルである。リテラルは q あるいは $\neg q$ の式で表され、 q はアトム、 \neg は論理否定 (classical negation) を表す。否定リテラルは前件、後件のいずれにも現れ得る。

ルール名 \mathbf{n} は $\mathbf{r}(X_1, \dots, X_n)$ に対するラベルであり、 \mathbf{r} は関数記号、 X_1, \dots, X_n はルール中の変数を示す。変数 X_1, \dots, X_n に項 t_1, \dots, t_n を代入することにより、ルールの全てのインスタンスについての名前 $\mathbf{r}(t_1, \dots, t_n)$ が得られる。

ルール間には順序が定義され、 $\mathbf{n}_2 \succ \mathbf{n}_1$ はルール \mathbf{n}_2 がルール \mathbf{n}_1 に優先することを表す。関係 \succ は推移的かつ反対称的で、厳密な半順序関係である。

論証は結論を導き出すルールの連鎖として定義される。以下、反論、直接論破の定義を与え、その上で論破された論証 (defeated argument), 妥当な論証 (plausible argument), 正当化論証 (justifying argument) を定義する。

定義 1 ある論証 A の部分論証 $A_i \subseteq A$ の結論 q に対して、結論が \bar{q} (\bar{q} は q の補リテラルである) であるような論証 B が存在する時、 B を A への**反論**という。

定義 2 ある論証 A について、その結論を q 、結論 q を導くルールを $\mathbf{r}_1 \in A$ とする。その時、結論が \bar{q} である論証 B が存在し、 \bar{q} を導くルール $\mathbf{r}_2 \in B$ について、 $\mathbf{r}_2 \succ \mathbf{r}_1$ が成り立ち、かつ B の部分論証 $B_i \subseteq B$ が全て正当化されているとき、 B は A を**直接論破**するという。

定義 3 ある論証 A の部分論証 $A_i \subseteq A$ を直接論破する論証 B が存在する時、 A は**論破された論証**という。また、その時 B は A を**論破する反論**であるという。

定義 4 ある論証 A について、 A を論破する反論が存在しない時、 A は**妥当な論証**であるという。

定義 5 ある論証 A について、 A に対する妥当な反論が存在しない時、 A は**正当化論証**であるという。

連絡先: 加藤義清, 宇宙航空研究開発機構, 〒305-8505 茨城県つくば市千現 2-1-1, Tel: 029-868-2799, Fax: 029-868-2987, E-mail: kato.yoshikiyo@jaxa.jp

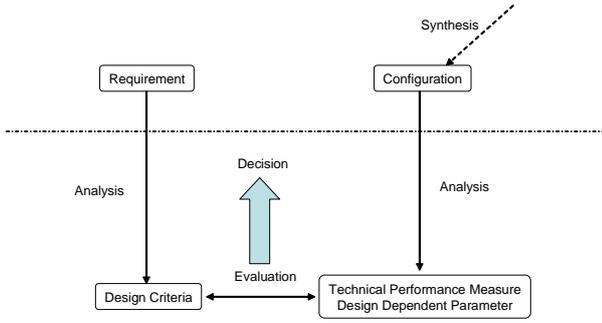


図 1: 設計意図表現の要素とその関係

3. 設計意図表現の枠組み

3.1 システム設計評価

本研究では設計意図を表現する前提として、合成 (Synthesis)、分析 (Analysis)、評価 (Evaluation) のサイクルからなる設計プロセスを考える。このサイクルの中で、設計に関する重要な意思決定がなされるのは評価の段階においてである。そこで、システム工学におけるシステム設計評価 (System Design Evaluation) の考え方 ([Blanchard 98] の Section 2.5 を参照) を参考にして、設計意図表現の構成要素検討した。

ここで特に着目したのが設計基準 (Design Criteria)、技術的パフォーマンス測定 (Technical Performance Measure (TPM)) および設計依存パラメータ (Design Dependent Parameter (DPP)) である。設計基準は要求分析の結果として得られ、設計結果に対する評価の指針を与えるものである。一方 TPM や DPP は設計基準に基づいて設計を評価する際、システムに関する測定可能な項目を与える。評価の際には設計基準と解析や試験など様々な方法により取得された TPM や DPP を照らし合わせ、意思決定を行う。

図 1 にシステム設計評価の概要を示す。システムの設計 (ここでは「構成 (configuration)」と呼ぶ) は合成により与えられる。設計基準は要求分析を通して明らかになり、設定される。また、解析等により各構成案の TPM が求められ、与えられる。ここまで得られた設計基準と TPM に基づきシステム設計に対する評価が行われ、次のフェーズに関する決定がなされる。

3.2 Argumentative Design Rationale Framework

Argumentative Design Rationale Framework (ADRF) は前節で述べたシステム設計評価に基づく設計過程のモデルを論争の枠組みを用いて表現したものである。

Argumentative design rationale framework \mathcal{F} は以下で表される組である。

$$\mathcal{F} = \langle O, E, M, V, DC, J, C \rangle \quad (2)$$

ここで、 O , M , DC , および C はリテラルの集合であり、 E , V , および J はルール集合である。各 $o_i \in O$ は設計案を表す。 M は TPM m_j を表すリテラルの集合であり、 $m_j(o_i, X)$ は設計案 o_i の TPM m_j についての値 X を与える。 $DC = \{dc_j\}$ は設計基準を表すリテラルの集合であり、その要素 $dc_j(o_i, X)$ は設計案 o_i の設計基準 dc_j についての評価値 X を与える。 C は $\{pro(X), \neg pro(X)\}$ からなるリテラルの集合であり、設計案に対する賛成あるいは反対の主張を表す。

E はある設計案が、ある TPM において特定の値をとることの証拠 (evidence) を表すルール集合である。証拠ルールは以下のように記述される。

$$e_i : m_j(o_k, x_l) \quad (3)$$

設計案 o_k が TPM m_j について、値 x_l を取ることを示す証拠 e_i が存在することを意味している。

V はある設計案が、ある設計基準において特定の値をとることを示した評価 (evaluation) ルール集合である。

$$v_i : dc_j(o_k, x_l) \leftarrow f_i(M, O) \quad (4)$$

ここで f は設計案 O についての TPM を含む式である。評価ルールの形式は設計案の設計基準における評価値は、その TPM 値に従属することを表している。

最後に、 J は設計案の評価値に基づいてなされる設計案についての判断 (judgement) ルール集合である。判断ルールは以下の形で表される。

$$j_i : c_j(o_k, x_l) \leftarrow g_i(DC, O) \quad (5)$$

ここで、 g は設計案 O についての設計基準の評価値 DC を含む式である。

4. 設計意図記述実験

前節で述べた設計意図表現の枠組みが形式的に記述可能であることを確かめる目的で法的推論システム New HELIC-II の知識表現言語 NHL により設計意図を記述し、New HELIC-II の論証生成機能による論証生成実験を行った。

4.1 NHL による設計意図の記述

New HELIC-II で扱うための知識は知識表現言語 NHL により記述されなければならない [Nitta 94]。NHL は項に型継承を導入した LOGIN[Ait-Kaci 86] を失敗による否定 (NAF)、論駁推論、事例ベース推論などを扱えるように拡張したものである。New HELIC-II は 3 種類の知識ベースにより構成される。

本節ではスタートラッカの設計を例題に、NHL による設計意図の記述の例を述べる。スタートラッカとは人工衛星の姿勢を決定するための装置であり、宇宙空間を撮像して画面に映った星像のパターンと自分の持っている恒星カタログとを照合することにより、センサの視野が天球のどちらの方向を向いているかを特定し、その情報を元に人工衛星の姿勢を決定する。

設計意図記述の例題としてスタートラッカの視野角の決定問題を取り上げる。スタートラッカがその方向を特定するためには、カタログに乗っている恒星がある程度数以上 (例えば 3 個以上) その視野に同時に捉えられている必要があり、その為には一定数以上の視野角が必要となる。天球上で星の分布は一律ではなく、視野角が小さいと星がまばらな領域では視野の同定^{*1}が出来なくなる。ここで全天のうち視野同定が可能な領域の割合を「視野同定被覆率」と呼ぶことにして、視野同定の要求に関する TPM とする。

一方、使用する撮像素子の画素数が一定だとすると、視野角が大きくなればなるほど画素あたりの角度が大きくなり、姿勢決定精度が悪くなる。つまり、精度の観点からは視野角が小さ

*1 視野が天球上のどの位置にあるかを同定すること。

```

&define_RBR_rule.
{
% 判断ルール
decision_1:: select(object=X/stt)
  <- satisfy(agent=X, object=coverage_requirement),
  satisfy(agent=X, object=accuracy_requirement).
decision_2:: select(object=X/stt)
  <- satisfy(agent=X, object=coverage_requirement).
decision_3:: select(object=X/stt)
  <- satisfy(agent=X, object=accuracy_requirement).
decision_4:: -select(object=X/stt)
  <- not satisfy(agent=X, object=coverage_requirement).
decision_5:: -select(object=X/stt)
  <- not satisfy(agent=X, object=accuracy_requirement).

% 評価ルール
design_criteria_1:: satisfy(agent=X, object=coverage_requirement)
  <- derive(agent=X/configuration,
  object=Y/stt_tpm[coverage=>[98..100]]).

design_criteria_2:: satisfy(agent=X, object=accuracy_requirement)
  <- derive(agent=X/configuration,
  object=Y/stt_tpm[resolution=>[80..9999]]).

% 証拠ルール
analysis_1:: derive(agent=c1,
  object=stt_tpm[coverage=>80, resolution=>200]).
analysis_2:: derive(agent=c2,
  object=stt_tpm[coverage=>90, resolution=>100]).
analysis_3:: derive(agent=c3,
  object=stt_tpm[coverage=>99, resolution=>67]).
analysis_4:: derive(agent=c4,
  object=stt_tpm[coverage=>98, resolution=>80]).
}

```

図 2: 設計意図を記述したルール定義.

いほうが良いことになる*2。ここで、視野角あたりの画素数を TPM とする。

このように、視野角というパラメータについて、視野同定被覆率と視野角あたりの画素数という 2 つの TPM はトレードオフの関係となっている。以上で述べた視野角の決定の問題について NHL により設計意図を記述したものを図 2 に示す。

図 2 では、評価の結果としての判断を表す判断ルールが 5 つ、要求を具体的な TPM と結びつける設計基準を表す設計基準ルールが 2 つ、各パラメータを持つ構成についての TPM を与えるファクト*3として証拠ルールが 4 つ定義されている。

design_criteria.1 は **coverage** 属性が 98 から 100 の範囲に入っている場合に **coverage_requirement** が満たされることを示している。**design_criteria.2** は **resolution** 属性が 80 以上の場合*4 **coverage_requirement** が満たされることを示している。

analysis.1 から **analysis.4** では、**coverage_requirement** のみを満たす構成が 1 つ、**accuracy_requirement** のみを満たす構成が 2 つ、両方の要求を満たす構成が 1 つとなるように TPM が与えられている。

*2 実際には姿勢決定精度には様々な要因が関係していて一概に述べることが出来ないが、ここでは話を簡単にするために視野角が小さいほど姿勢決定精度が良くなることにする。

*3 例えば解析によって得られることを想定している。

*4 上限の 9999 は便宜的に与えたもの。

4.2 論証生成による設計意図の評価

前節で示した NHL による設計意図記述に対して、New HELIC-II の論証生成機能を利用して論証の生成実験を行った。**select(object=X)** をゴールに設定して、使用する判断基準を「なし」「**weigh.coverage**」「**weigh.accuracy**」の 3 つ場合で実行した。その結果を表 n 1 に示す。いずれの判断基準においても 6 つの論証が生成された。そのうち、**X=c4** となる論証は 3 つ生成され、それら全てがいずれの判断基準においても正当化論証となっている。**c4** は二つの要求を満たし、**select(object=c4)** への反論となる、各要求を満たさない場合のルール **decision.4** および **decision.5** を論破しているためである。**c1~c3** については、片方の要求しか満たさないため、判断基準が無い場合には妥当な論証に留まり、判断基準が与えられた場合に優先される方の要求を満たす構成を選択する結論の論証が正当化論証となっている。

5. 設計意図獲得支援システム

本節では、本研究で提案する設計意図表現の枠組みに基づく、設計意図獲得支援システムの構想について述べる。1 節でも述べたように、本研究の狙いは計算可能な設計意図表現を導入することにより、設計者に対して設計意図を記録することによる短期的効果を実現し、設計意図の獲得を促進することである。そこで、先に述べた計算可能な設計意図表現を用いて設計意図獲得支援システムを構築し、そのシステムの評価を通して本研究の最終的な評価とする予定である。

現在の構想ではプロジェクトメモリー管理システムを基盤に設計意図獲得支援システムを実現することを計画している。プロジェクトメモリー管理システムとは具体的には設計会議の議事録、設計検討の資料、解析に関する資料、試験の計画書やデータ、報告書などで、実際の設計開発作業の中で成果物や、副産物として出てくる資料やデータを管理するシステムである。

設計意図獲得支援機能として以下のものを検討している。

論証構造としての設計過程の記録・表示 設計の過程を設計意図表現に従う形でシステムに入力してもらうことにより、論証構造として設計過程を獲得する。獲得された設計過程は論証構造として可視化することにより、設計がどこまで進んだか、検討が足りない箇所はないかどう、設計者に対して設計過程を見通しよくする効果が期待される。

設計意図表現によるプロジェクトメモリーの索引付け 各種資料をシステムに登録する際に、設計意図表現に基づいた索引付けを行うことにより、論証構造として資料を整理して提示が可能となり、資料の検索性が高まるとともに、論証構造が実際の検討資料等と対応していることにより、論証構造自体の理解を助けることにもなることが期待される。

論駁推論による設計過程の評価 獲得された設計過程の記録に対して、論駁推論を適用することにより設計に矛盾が無いかが、検討が不足しているところが無いかをチェックし、設計者に注意を促す。

6. むすび

本稿では、システム工学におけるシステム設計評価の考え方に基づいた設計意図表現についての検討をし、更に知識表現言語 NHL による設計意図の記述および法的推論システム New

| X (トッブルール) | 判断基準なし | weigh_coverage | weigh_accuracy |
|-----------------|--------|----------------|----------------|
| c1 (decision.3) | P | D | J |
| c2 (decision.3) | P | D | J |
| c3 (decision.2) | P | J | D |
| c4 (decision.1) | J | J | J |
| c4 (decision.2) | J | J | J |
| c4 (decision.3) | J | J | J |

表 1: `select(object=X)` をゴールにした時に、判断基準毎に生成される論証とその評価。P は妥当な論証、D は論破された論証、J は正当化論証であることをそれぞれ示している。

HELIC-II の論証生成機能による設計意図記述に基づく論証生成実験を行った。また、提案する枠組みに基づく設計意図獲得支援システムの構想を述べた。

実験の結果、計算可能な設計意図表現について十分に実現の可能性があることが明らかとなった。しかし、計算可能になったことの代償として従来のセミフォーマルな設計意図表現よりも記述コストはかなり高いものとなった。今後、設計意図獲得支援を実現するに当たっては、論証に基づく設計意図表現に基礎付けられた形でより記述コストの低い、設計意図入力インタフェースの検討が必要である。

謝辞

本研究の一部は、文部科学省科学研究費補助金若手研究(B)No.16700158 による。

参考文献

- [Ait-Kaci 86] Ait-Kaci, H. and Nasr, R.: LOGIN: A Logic Programming Language with Built-in Inheritance, *Journal of Logic Programming*, Vol. 3, pp. 185–215 (1986)
- [Blanchard 98] Blanchard, B. S. and Fabrycky, W. J.: *Systems Engineering and Analysis*, Prentice Hall, Upper Saddle River, New Jersey, 3rd edition (1998)
- [Buckingham Shum 94] Buckingham Shum, S. and Hammond, N.: Argumentation-based design rationale – what use at what cost, *International Journal of Human-Computer Studies*, Vol. 40, No. 4, pp. 603–652 (1994)
- [Conklin 91] Conklin, E. J. and Yakemovic, K. B.: A Process-Oriented Approach to Design Rationale, *Human-Computer Interaction*, Vol. 6, pp. 357–391 (1991)
- [Moran 96] Moran, T. P. and Carroll, J. M. eds.: *Design Rationale – Concepts, Techniques, and Use*, Computers, Cognition, and Work, Lawrence Erlbaum Associates (1996)
- [Motta 00] Motta, E., Buckingham Shum, S., and Domingue, J.: Ontology-driven document enrichment: principles, tools and applications, *International Journal of Human Computer Studies*, Vol. 52, pp. 1071–1109 (2000)
- [Nitta 94] Nitta, K., Shibasaki, M., Sakata, T., Yamaji, T., Ohsaki, H., Tojo, S., Kokubo, I., and Suzuki, T.: Knowledge Representation of New HELIC-II, Technical Report

TM-1301, Institute for New Generation Computer Technology (ICOT) (1994)

- [Sartor 93] Sartor, G.: A Simple Computational Model for Nonmonotonic and Adversarial Legal Reasoning, in *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pp. 192–201 (1993)
- [Shipman 99] Shipman, F. M. and Marshall, C. C.: Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems, *Computer-Supported Cooperative Work*, Vol. 8, No. 4, pp. 333–352 (1999)