ドメインモデルに基づいて技術文書中の矛盾を検出する方法

山田隆弘[1]
[1]CONOCIMISTA
tyamada99@yahoo.co.jp

概要

本稿では，技術文書のように厳密に事実を伝えるべき文書中の矛盾を検出する方法について議論する．文書中の矛盾を放置したままにすると重要な損害がもたらされる場合は多い．例えば，システムの開発において，文書中の矛盾に気付かずにシステムを製造してしまうと，テスト時に不具合が発生し，製造をやり直すことになる．本稿では，技術文書中の矛盾を検出する基本的な方法を提案する．特に，システム開発において実際に発生しやすい矛盾に焦点を当てる．

1　はじめに

自然言語処理技術の用途の一つは，自然言語で書かれた文章について推論を行うことである．自然言語処理技術を利用した推論としては，いくつかの種類のものが考えられるが，本稿では，一つの文書中（あるいは複数の文書間）の矛盾の検出を取り上げる．その理由は，文書中の矛盾の検出は，実用的に重要な課題であるにもかかわらず，実践的な研究があまり行われていないからである．

文書中の矛盾を放置したままにすると重要な損害がもたらされる場合は多い．例えば，筆者が長年関わってきた人工衛星の開発においては，一つの人工衛星のために多くの人が分担して大量の技術文書（要求仕様書や設計仕様書等）を作成する．これらの文書中に矛盾がないことを人工衛星の製造の前に確認するのであるが[1]，文書中の矛盾に気付かずに人工衛星を製造してしまうと，人工衛星のテスト時に不具合が発生し，製造をやり直すことになる．人工衛星以外でも，複雑なシステムの開発においては，同様なことが発生するであろう．

ところで，実際の技術文書は矛盾だらけというわけではなく，矛盾はそれほど多くはない．しかし，矛盾がなかったとしても文書中に矛盾がないことを確認する必要はある．文書中に矛盾がないことの確認を行うためには，矛盾の検出と同じ手順を実行する必要がある．従って，あらゆる技術文書に対して，矛盾の検出を実行する必要があるわけである．

本稿では，技術文書のように厳密に事実を伝えるべき文書中の矛盾を検出する方法を論じる．特に，システム開発において実際に発生しやすい矛盾（特に，異なる担当者が執筆した別々の記述の間の矛盾）に焦点を当てることにする．

なお，本稿の内容は，筆者の過去の研究[2]の内容を詳細化しつつ発展させたものである．

2　技術文書中の矛盾の検出

2.1　矛盾検出に関する先行研究

自然言語処理技術を使用して文書中の矛盾を検出するための代表的な先行研究としては[3]-[10]等がある．矛盾とは「二つの命題が同時には成立しない」ことであるが，上記の先行研究の全てにおいて，反対あるいは対照的な意味を有する二つの命題（あるいは文）の検出が主目的となっている．検出のための言語資源としては，WordNet [11]や VerbOcean [12]における反義語の規定が使用されている．例えば，「大きい」と「小さい」は反義語であるから，「A は大きい」と「A は小さい」は矛盾しているとして判断している．

しかし，矛盾とは「同時には成立しない」ことであるから，必ずしも意味が反対でなくても矛盾していることはあり得る．例えば，「太郎は歩いている」と「太郎は走っている」は矛盾している．なぜならば，「歩く」と「走る」を同時に行うことは不可能であるからである．また，「太郎は名古屋に滞在している」と「太郎は大阪に滞在している」とは矛盾しているが，「太郎は札幌に滞在している」と「太郎は北海道に滞在している」とは矛盾していない．これらの矛盾を検出するためには，WordNet や VerbOcean よりも詳細な知識が必要となる．

2.2 技術文書中の矛盾

技術文書中の矛盾として典型的な例は，以下のようなものである．

(1)「監視カメラ 01 はタイプ S のカメラである」

(2)「監視カメラ 01 はタイプ T のカメラである」

ここで，タイプ S のカメラとタイプ T のカメラとは別のものであり， (1)と(2)は矛盾している．また，

(3)「監視カメラ 01 はセンサである」

(4)「監視カメラ 01 はカメラである」

の二つの文は矛盾していない．なぜならば，カメラはセンサの一種であるからである．

上記のような矛盾の検出，あるいは，矛盾していないことの確認を行うためには，この技術文書が対象としているシステムで使用される装置や技術に関する詳細な情報が必要である．このような情報は，特定の分野あるいはシステム毎にドメインモデルとして規定する必要がある．

筆者は，ドメインモデルに基づいて矛盾検出のための辞書あるいはオントロジーを構築する方法を過去に提案したが[2]，以下ではその研究をさらに詳細化し，ドメインモデルに基づいて技術文書中の矛盾を検出する方法について論じる．

3 ドメインモデル

3.1 ドメインモデルの定義

ドメインモデルとは，特定の分野で使用される基本概念を一つのモデルとして定義したものである．本稿ではオブジェクト指向モデリング[13]の手法を用いてドメインモデルを構築する．ドメインモデルの主要な構成要素を以下に示す．以下で，オブジェクトとは，個々の物を属性値の集合として表現した仮想的な物である．

- クラス（共通の特徴を有するオブジェクトの集合）
- 属性（オブジェクトの特徴を表すパラメータ）
- 属性値（属性の取る値）
- 属性値型（属性の取り得る値の集合）
- 関係（複数のオブジェクト間に成立する関係）
- 関係の多重度（一つのオブジェクトが他のいくつのオブジェクトと同一の関係を持てるか）

属性値型には，整数，実数，文字列，列挙型などがある．列挙型とは，いくつかの離散的な値（文字列で表される）の集合として規定され，それらのうちの一つが属性値として取られることを示す．ここでは，属性値型が列挙型の場合，列挙されている個々の値は互いに排他的であるとする．

複数のクラスの間には一般化および特殊化の関係を規定することができるが，これについては，実例を使って 3.2 節で説明する．

ドメインモデルは， Unified Modeling Language (UML) [14]のクラス図を用いても定義できる．オブジェクト指向モデリングや UML の詳細は[13]や[14]を参照して頂きたい．

ところで，特定のドメインモデルに基づいて技術文書中の矛盾の検出を行う場合，その技術文書も同じドメインモデルに準拠して書かれている必要がある．これは，矛盾の検出だけでなく，文書の曖昧性をなくすためにも有用である．筆者は，ドメインモデルに基づいて技術文書を作成する方法についても研究を行なっているが[15]，残された課題もあり（第 5 節の最後の部分参照），今後も研究を続ける予定である．

3.2 ドメインモデルの実例

簡単なドメインモデルの例を UML [14]のクラス図として図 1 に示す．この図では，例えば，システムというクラス（オブジェクトの集合）があり，そのクラスに属するオブジェクトは，システム ID という属性を持ち，その属性値は文字列であることが表わされている．さらに，システムクラスのオブジェクトは，コンポーネントクラスの 1 つあるいは複数のオブジェクトとの間に関係を持つことができる．システムクラスのオブジェクトである X とコンポーネントクラスのオブジェクトである Y がこの関係を有する場合，「Y は X の構成要素である」「X は Y の親システムである」というように表現される．

プロセッサとセンサの二つのクラスは，コンポーネントクラスを特殊化したものであり，逆に後者は前者を一般化したものである．これは，プロセッサやセンサはコンポーネントの一種であることを意味する．また，プロセッサとセンサは互いに排他的であるとする．プロセッサクラスとセンサクラスのオブジェクトは，そのクラスの属性とその上位のコンポーネントクラスの属性の両方を持つ．

カメラクラスのカメラ種別という属性の属性値型であるカメラ種別名は列挙型であり，カメラ種別の属性値として，タイプ S，タイプ T，タイプ U の三つのうちのいずれかを取るものとする．

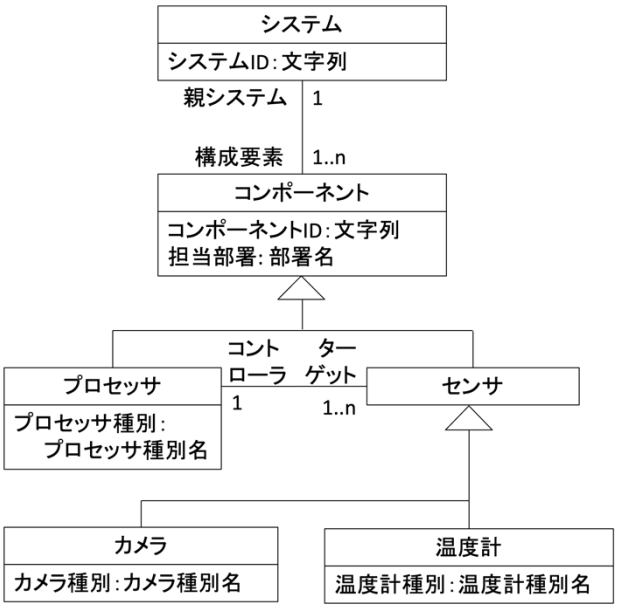

図 1　ドメインモデルの実例

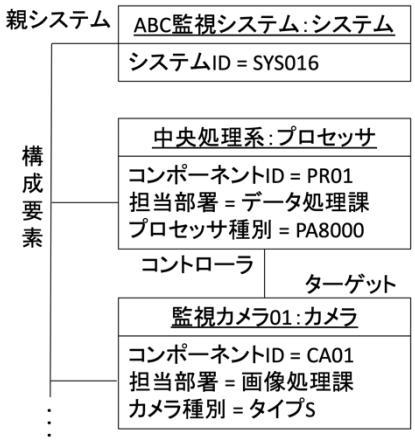

図 2　オブジェクトモデルの実例

4　エンティティの同定

Kalouli 等[7]が指摘しているように，「A は B である」と「A は C である」が矛盾しているかどうかを決定するためには，双方の文で A が同一のエンティティ（人や物）を指し示していることを確認する必要がある．これは，[8]や[10]が扱っているような病気や薬品の場合は問題にならないが（病気や薬品はエンティティではなく概念であるので），類似した個物を多数扱う可能性のある技術文書においては，重要な課題となる．

この課題を解決するためには，第 3 節で示したドメインモデル（クラスのモデル）に基づいて，該当文書で使用されるエンティティをオブジェクトモデルとして示し，該当文書に現れる各エンティティをオブジェクトモデル中のオブジェクトに紐づければよい．

図 1 のドメインモデルに準拠したオブジェクトモデルの実例を UML [14]のオブジェクト図として図 2 に示す．この図は，例えば，ABC 監視システムというシステムは，システムクラスのオブジェクトであり，システム ID の値は SYS016 であることを示している．また，ABC 監視システムは，中央処理系（プロセッサクラスのオブジェクト），監視カメラ 01（カメラクラスのオブジェクト）等を構成要素として持つことも示している．この図は，例えば，「本システムは ABC 監視システムと呼ばれる．・・・」という仕様書中の記述より作成してもよいし，このオベジェクト図を仕様書の一部として作成してもよい．

技術文書中で何らかのオブジェクトに言及するときは，オブジェクトモデルに現れる名前や ID を使用すれば，エンティティの同定が確実に行われることになる．このように同定されたオブジェクトについて(1)と(2)のように相反する属性値が設定されていれば，矛盾であると判定できる．

エンティティの同定と同様にイベントの同定も必要になる．例えば，「太郎はホテルに泊まった」と「太郎は旅館に泊まった」は，これだけでは矛盾かどうか分からない（太郎が別々の日にホテルと旅館に泊まる可能性があるから）．しかし，「太郎は昨夜はホテルに泊まった」と「太郎は昨夜は旅館に泊まった」は矛盾している．

技術文書においては,「何がいつどこで何をする」という情報が扱われることは（計画書類以外では）あまりなく，重要なのは「どのような条件で何が何をするか」である．条件についてもドメインモデル（クラスモデル）とオブジェクトモデルに準拠して記述すれば，同定することが可能になる．このように同定された条件とエンティティについて，相反する行為（例えば，排他的な機能の実行）が設定されていれば，矛盾であると判定できる．

5　語彙の同定

これは必ずしも技術文書に限った話ではないが，実質的には同一のことが異なる語彙を用いて表現される場合がある．例えば，

(5)「監視カメラ 01 が撮像を行うには，中央処理系は監視カメラ 01 に信号 X を送る」

(6)「監視カメラ 01 が撮像を行うには，監視カメラ 01 は中央処理系から信号 Y を受け取る」

の二つの文は矛盾している．このような矛盾を検出するためには，語彙の意味を同定する必要がある．

語彙の意味の同定は，文書で使われる語彙の意味を基本的な語彙の組み合わせとして規定すれば可能となる．上の例でいえば，「B に X を送る」の事後条件（文を実行した後の状態）は「B が X を受け取る」であると規定すればよい．この場合，「受け取る」が基本的な語彙であり，「送る」の意味を「受け取る」との関係によって規定していることになる．

(5)と(6)の場合,「監視カメラ 01 に信号 X を送る」を実行すると「監視カメラ 01 が信号 X を受け取る」という状態が出現するという知識を利用すれば，(5)と(6)の間の矛盾が検出できる．

文書で使われる語彙を基本語彙に対応させた一種の辞書もドメインモデルの一部として作成すべきである．筆者は，文書で使われる語彙の意味を基本的な語彙を用いて規定する方法についても検討を行っているが[15]，基本的な語彙を選定する方法等についていくつかの課題が残されている．

5　複合的な場合

最後に，実際のシステム開発で起こりがちな複合的な矛盾を取り上げる．

図 2 のオブジェクトモデルにおいて，監視カメラ 01 は撮像した画像を中央処理系に送り，中央処理系は受け取った画像を保存することになっている．このようなシステムの開発において，以下のような事態が起こり得る．すなわち，監視カメラ 01 の担当者は，取得した画像をそのままの形で中央処理系に送れば，中央処理系がそれを圧縮した上で保存してくれると考えている．ところが，中央処理系の担当者は，圧縮された画像が監視カメラ 01 より送られてくるので，受け取ったデータをそのまま保存すればよいと考えている．

上記の問題は，データ圧縮の機能をどのコンポーネントが実施すべきかについての矛盾として解釈することもできるし，監視カメラ 01 から中央処理系に画像データを受け渡すインタフェースのデータ形式に関する矛盾として解釈することもできる．この矛盾を文書に基づいて検出することが可能であろうか．

この矛盾を機能の実現方法の矛盾として検出することも可能であるが，本稿で論じてきた方法を適用しやすいのは，インタフェースの矛盾としての検出である．すなわち，ドメインモデルにおいて画像データというクラスを定義し，そのクラスは画像データ形式という属性を有することにする．画像データ形式の属性値としては，列挙型の画像データ形式名を定義し，非圧縮画像データと圧縮済み画像データの二つの値を取るものする．すると，監視カメラ 01 の担当者と中央処理系の担当者の理解は，それぞれ

(7)「監視カメラ 01 は非圧縮画像データを中央処理系に送る」

(8)「中央処理系は監視カメラ 01 から圧縮済み画像データを受け取る」

と表現できる．(7)と(8)に第 4 節の語彙の同定法を適用すれば，これらの間の矛盾が検出できる．

各々の担当者が(7)と(8)のように文書を記述するかどうかという問題もあるが，文書の矛盾を検出する前に，そもそも技術文書はドメインモデルに準拠して書かれるべきである．これは，文書から曖昧性を取り除くためにも必要である．

6　おわりに

本稿では，ドメインモデルに基づいて技術文書中の矛盾を検出する方法について論じてきた．本稿では，技術文書がドメインモデルに準拠して書かれていることを前提としている．この点において，本研究は，そのような前提の成立しない既存の文書中の矛盾の検出を目指している先行研究[3]-[10]とは異なっている．しかし，技術文書は，そもそも誰が読んでも同一の解釈に至るように書くべきであり，そのためにも統一的なドメインモデルに準拠して書く必要がある．

本稿で述べたのは，矛盾を検出するための基本的な方法のみであり，実際の技術文書に適用するためには本稿の内容をさらに詳細化する必要がある．また，矛盾を検出するシステムを構築し，実際の技術文書に適用することも今後の課題である．なお，技術文書を計算機に蓄積する方法としては知識グラフが最適であると筆者は考えており，それに関する研究は別途実施しているところである[16].

参考文献

[1] 山田隆弘, “バックキャスティングによる技術開発のすすめ”, ISAS ニュース, JAXA 宇宙科学研究所, No. 488, p. 8, 2021.

[2] 山田隆弘, “常識推論を支援するための辞書（あるいはオントロジー）の構築方法,” 言語処理学会第 29 回年次大会, 2023.

[3] Sanda Harabagiu, Andrew Hickl and Finley Lacatusu, “Negation, Contrast and Contradiction in Text Processing,” *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, pp. 755-762, 2006.

[4] Marie-Catherine de Marneffe, Anna N. Rafferty and Christopher D. Manning, “Finding Contradictions in Text,” *Proceedings of ACL-08: HLT*, 2008.

[5] Alan Ritter, Doug Downey, Stephen Soderland and Oren Etzioni, “It’s a Contradiction—No, it’s Not: A Case Study using Functional Relations,” *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 11–20, 2008.

[6] Valentina Dragos, “Detection of contradictions by relation matching and uncertainty assessment,” *21st International Conference in Knowledge Based and Intelligent Information and Engineering Systems (KES 2017)*, 2017.

[7] Aikaterini-Lida Kalouli, Livy Real and Valeria de Paiva, “Correcting Contradictions,” *Proceedings of the Computing Natural Language Inference Workshop*, 2017.

[8] Noha S. Tawfik and Marco R. Spruit, “Automated Contradiction Detection in Biomedical Literature,” *Proceedings of the 7th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp.138-148, 2018.

[9] 外園康智, 長谷川貴博, 渡邉知樹, 馬目華奈, 築有紀子, 谷中瞳, 田中リベカ, Pascual Martínez-Gómez, 峯島宏次, 戸次大介, “意味解析システム ccg2lambda による金融ドキュメント処理,” 第 32 回人工知能学会全国大会, 2018.

[10] Graciela Rosemblat, Marcelo Fiszman, Dongwook Shin and Halil Kilicoglu, “Towards a characterization of apparent contradictions in the biomedical literature using context analysis,” *Journal of Biomedical Informatics*, 98, 103275, 2019.

[11] Christiane Fellbaum, *WordNet: an electronic lexical database*, MIT Press, 1998.

[12] Timothy Chklovski and Patrick Pantel, “VERBOCEAN: Mining the Web for Fine-Grained Semantic Verb Relations,” *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 33–40, 2004.

[13] Michael Blaha and James Rumbaugh, *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice Hall, 2005.

[14] Grady Booch, James Rumbaugh and Ivar Jacobson, *The Unified Modeling Language User Guide*, Second Edition, Addison-Wesley, 2005.

[15] 山田隆弘, “公理としてのドメインモデルに基づくユースケースの記述方法,” 情報処理学会研究報告，Vol. 2024-SE-217, No. 6, 2024.

[16] 山田隆弘, “様々な技術文書を表現する知識グラフの構築方法,” 第 38 回人工知能学会全国大会, 2024.