

# 機能的知識外化ツール OntoGear を用いた失敗知識の構造化と活用

## Failure Knowledge Management Using OntoGear

高藤淳<sup>\*1</sup> 來村徳信<sup>\*1</sup> 溝口理一郎<sup>\*1</sup>

Sunao Takafuji<sup>\*1</sup>, Yoshinobu Kitamura<sup>\*1</sup>, and Riichiro Mizoguchi<sup>\*1</sup>

<sup>\*1</sup> 大阪大学 産業科学研究所

<sup>\*1</sup> The Institute of Scientific and Industrial Research, Osaka University

**Abstract:** While we have studied functional ontology and functional modeling framework for years, we developed an information system that enables users to describe function decomposition tree, which represents functional structure of an artifact, and successfully delivered it into industry. Recently, we renewed that system, which is called OntoGear, to adapt our new theory of ontology engineering with information contents centric architecture. OntoGear allows us not only to describe basic functional decomposition tree, but also to reveal a specific functional structure such as failure one. This paper discusses modeling of failure functional structure, and utilizing of it with using OntoGear.

## 1. はじめに

近年、ものづくり環境の複雑化・多様化に伴い失敗を適切にマネジメントすることが求められている。例えば、失敗学[1]では失敗に対して構造的な分析を加え、分類し、記録することで、失敗を肯定的に捉えて有効活用する取り組みを提案している。

失敗に関する社会的な影響度の大きい自動車や家電などのリコール・不具合情報は、すでに Web で公開されている[2, 3]他、前述の失敗学を礎とした包括的な失敗事例の収集や Web 公開も行われている[4]。

インターネットの利用という意味では、ICT (Information and Communication Technology) を活用していると言えるが、検索して参照することができるという水準であり、自然言語による記述の恣意性、属人的な記述の非一貫性等の問題から表層的な語彙を主体とするアクセスでは、不具合の未然予防等に寄与する十分な活用に至っていないのが現状である。

不具合情報の秘匿性から、企業内での失敗活用事例が開示されることはほとんど無いが、筆者らの研究における限られた接点においても、不具合を適切にマネジメントするニーズは少なくない。

そこには、単なる文書へのアクセス・参照という水準を超えて、知識の相互運用性を踏まえた高度な失敗知識活用の余地が残されていると考えられる。

筆者らは、人工物の機能に関する積年の研究[5]を踏まえ、不具合に関しても機能の視点で統一的にそ

の構造を顕在化することができる機能・不具合統合モデル[6]を提案している。また、人工物の機能的知識を外化するツール OntoGear を開発している[7]。これらの理論的・実践的基盤により、失敗知識を構造化して再利用するという、より高次の ICT 知識活用に貢献可能であると考えている。

本研究では、失敗知識の中でも人工物の不具合に関するものに限定し、機能オントロジーの視点から不具合の機能構造を効率的にモデリングする枠組みについて述べる。この枠組みには、テキストから不具合を表す機能構造を自動構築する試みを含む。また、OntoGear をベースとしたプロトタイプにより、具体的な活用手段を提示する。

## 2. OntoGear による不具合の機能モデリング

### 2.1 機能・不具合統合モデル

筆者らの機能オントロジーでは、機能を「振る舞いがあるコンテキストの下で意図したゴールを達成するために果たす役割(ロール)概念」である、と定義している。

例えば、ある 2 つの物質を擦り合わせることで熱を生成する。また、それらの一方が刃物状で硬く一方が柔らかいものである場合に、後者を切断することができる。この時、"擦る"という同一の振る舞い

が、熱を出す目的の場合は"熱を生成する"機能を果たし、切る目的の場合は"分割する"機能を果たす。

このように機能を捉えた上で、装置の入出力における対象物の状態変化に視点を固定し、「何を達成するか」を表す機能と「どのように達成するか」を表す方式を明確に分離することで、一貫した機能構造の表出化が可能となる。前記記述様式に基づいて全体装置から部分装置へと木構造の形式で記述したものを機能分解木と称する。機能分解木の記述には、機能の本質的な概念を体系化した機能概念オントロジーを用いる。

複数の機能分解木を記述した後、方式に着目して達成すべき機能を中心に方式を集約・分類し、汎用化・組織化することで再利用可能な方式知識ベースを構築することができる。

ここで、人工物がある目的を果たすことは、設計者が「意図した目的」(以下、意図内目的)ということを含意しているが、不具合を「意図しない目的」(以下、意図外目的)と見なすことで、同一の様式で機能構造の記述が可能となる。このように意図内及び意図外の機能構造を統合的に記述するモデルを機能・不具合統合モデルといい、このモデルに基づいて記述された意図外の分解木を不具合機能分解木という。

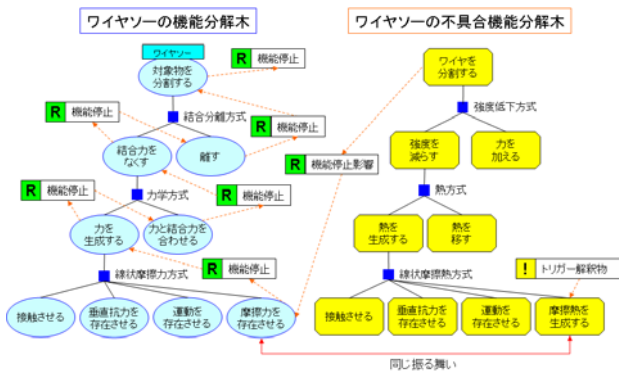


図 1. 機能・不具合統合モデルによる機能分解木

例えば、図 1 は、機能・不具合統合モデルに依るワイヤソーの機能分解木を表しており、図中左側が通常の機能分解木、右側が不具合機能分解木である。意図内の機能を正機能、意図外の機能を反機能(不具合機能も同義)ということにすると、本例では「摩擦力を存在させる」正機能に対して「摩擦熱を生成する」反機能が対峙する。機能分解木と不具合機能分解木の対置によってワイヤソーが対象物を切断する時に、摩擦熱を生成すること、そのためワイヤ自身を分割し、ワイヤソーの機能が停止することを明瞭に把握できる。

## 2.2 機能的知識外化ツール OntoGear

OntoGear は、機能的知識体系化の枠組みに基づいて人工物の機能構造を記述するツールである。本ツールは、株式会社ジャストシステムの XML アプリケーション構築プラットフォーム xfy[8]によって構築されており、目的に応じて XML (eXtensible Markup Language) の記述内容を変更し固有のスクリプト言語 xvcd で表現系に写像することで、柔軟に拡張することができる。

OntoGear は、前節で述べた機能・不具合統合モデルを含め、オントロジー工学の発展に準じた最新の機能モデリングに関する機能性を実現している。

また、方式知識を組織化・体系化するための方式知識エディタ (OntoGear WKE (Way Knowledge Editor)) を同梱しており、現在、試用評価版を公開している。

OntoGear は、xfy の plug-in アーキテクチャに準拠しており、Java 言語で任意の機能モジュールを作成して組み込むことにより任意の機能拡張が可能である。この拡張性に基づき、本研究では、不具合の未然予防を支援する機能モジュール FPoD (Failure Prevention on Demand) plug-in を組み込んで機能拡張を図ったプロトタイプを試作した。

## 3. 失敗知識の構造化に関する分析

本研究では、冒頭で述べたように失敗知識を人工物の不具合に関する機能的知識と位置付けている。従って、本研究における失敗知識の構造化は、不具合の機能構造を機能分解木の形式で表出化することと等価である。

不具合機能分解木は、機能・不具合統合モデルにおいて表出化の枠組みが提示されているが、表出化を容易にするための支援があると望ましい。

そこで、本章では、不具合報告書等の文献に記述されたテキスト情報から、表出化の支援に寄与する情報を得るため、不具合関連文献特有の語彙や記述上の特徴について分析を行う。

ここで、本稿の用語としてテキスト中で正機能を表す語彙を表層語彙、反機能を表す語彙を不具合表層語彙と称する。表層語彙に対して、機能概念オントロジーの概念ラベルを表す語彙を標準語彙という。

### 3.1 不具合表層語彙の分析と辞書構築

不具合特有の機能語彙を顕在化するために、一般的な技術文献(以下、正コーパス)と意図外の表現を多く含む不具合関連文献(以下、反コーパス)の 2 つを用意し、各々から機能概念を内包する語彙(動詞)を抽出して両者の語彙統計分布から不具合表現に固有性が高い語彙を弁別する実験を行った。

厳密には前記語彙定義に合致しない場合もあるが、

近似的な意味で正コーパスから抽出した語彙を正機能語彙、反コーパスから抽出した語彙を反機能語彙と称することにする。

正機能語彙については、表層・標準語彙の写像用辞書を構築する際に抽出したものを援用した[9]ため、既存研究における付随情報と結果のみ記載する。以下、実験における各実施段階の概要を記す。

### 3.1.1 文献解析用コーパスの収集・整備

企業内の不具合関連文書は一般に公開されるものではないため、Web 公開されている失敗知識データベースおよび自動車のリコール・不具合情報を収集・整備し、文献解析用のコーパスとして用いた。

失敗知識データベースに関しては、1,160 事例中(2008 年 8 月時点)ヒューマンエラー等の事例を除いた 415 事例、自動車のリコール・不具合情報に関しては、ランダムに選んだ 2,088 事例を用いた。

### 3.1.2 機能語彙の抽出

前記コーパスに対して形態素解析処理を行い、複合語句をまとめ上げることで機能語彙を抽出した。表 1 に正機能語彙と反機能語彙の抽出結果を示す。

	正コーパス	反コーパス
延べ語数(語)	28,922	6,613
異なり語数(語)	2,097	1,121

表 1. 正機能・反機能語彙抽出結果

上記結果に対して、自動詞・他動詞や表記揺れの統制等を行い、802 語の比較対照語彙群を得た。

### 3.1.3 語彙統計の定量化

文献中の語彙が、不具合表現語彙として用いられ易いかどうかを弁別するために、式(1)に定義される反機能固有指数を導入した。

- ・  $F_p$  : 語彙出現頻度 / 正機能文献中の全語彙数
- ・  $F_n$  : 語彙出現頻度 / 反機能文献中の全語彙数
- ・ 反機能固有指数 =  $F_n / F_p$  (1)

反機能固有指数は、その定義から 1.0 付近では正・反いずれにも用いられ、1.0 より大きいほど不具合の記述に用いられ易く、逆に 1.0 より小さいほど通常の機能記述に用いられ易い指標である。

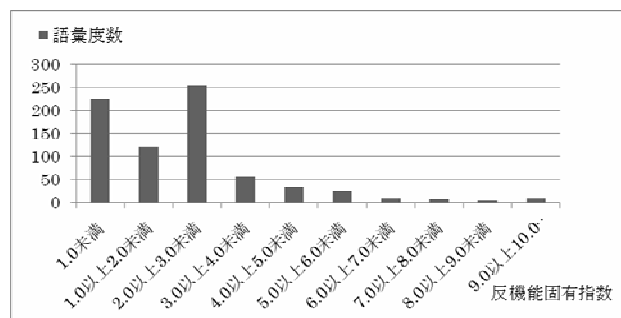


図 2. 反機能固有指数ヒストグラム

前節の比較対照語彙群に対して反機能固有指数を計算し、0~10 までの指数値に対して度数分布を取ると図 2 が得られる。

この結果から、反機能固有指数 3.0 付近で明確な正機能と反機能の境界が見出される。ここで、同指数 2.0 台の語彙について詳細な出現頻度を見ると、該当語彙の 91.7% は正機能語彙 1 語、反機能語彙 2 語の出現割合であった。従って、2.0 台の境界付近は、正・反に関する明瞭な弁別性を有しない領域と判断し、以下の区分を設けた。

- ・ 1.0 未満：正機能語彙として有意
- ・ 1.0 以上 3.0 未満：正・反機能のいずれにも使用
- ・ 3.0 以上：反機能語彙として有意

### 3.1.3 不具合表層・標準語彙辞書の構築

反機能固有指数が高い表層語彙は、不具合機能分解木を記述する際にも多用されることが予測されるため、これらの語彙を不具合表層・標準語彙辞書として体系化することは、不具合の機能モデリングの支援に寄与すると考えられる。

そこで、反機能固有指数が、3.0 以上の 151 語に関して、ドメイン依存性が高い、或いは特定の装置に固有性が高いと思われる語彙を除いた 102 語を選別し、標準語彙に割り当てた後、不具合表層語彙・標準語彙辞書を構築した。

## 3.2 技術文献における不具合構造の記述様相に関する分析

### 3.2.1 不具合の表出化に関する考察

製造業の従事者が、ある製品の不具合を表出化する場合、言語情報を主体として表現することが一般的である。設計時に信頼性解析手法の一つである FMEA (Failure Mode Effect and Analysis) を用いる場合は FMEA 記述シートの各項目欄に自然言語により記入する。或いは、保守フェーズで不具合報告書を作成する場合も、通常、自然言語で内容を記述する。

FTA (Fault Tree Analysis) のように論理的な木構造(FT 図)の形式で、事象間の論理構造を明示化する手法もあるが、その場合においても個々の事象の記述は自然言語で描写される。

不具合の表出化に関して、自然言語で記述する場合と何らかの構造化の手法を適用する場合とは、表出化された結果は異なる。しかし、いずれの様式であっても、不具合情報を伝える側と伝えられる側でそれが不具合であるとの共通認識を得ることができるということは、表出化結果の背後に共通する根源的な事態構造(以下、基底事態構造)の存在を仮定することができる。

逆に言えば、ある基底事態構造を目的や制約に準

じて表出化した結果が、不具合報告書、FMEA シート、FT 図などであると考えられる。

そのような基底事態構造は、人間の思考そのものに深く関与するため、一般的なモデルとして定式化することは難しい。しかし、人工物に関しては、筆者らの機能オントロジーが近似的なモデルとして援用可能であると考えられる。何故なら、任意の人工物に対して、その構成要素及び機能構造を必要な粒度に応じて、記述する人に大きく依存することなく、同一の様式で顕在化できることは、共通の理解構造を顕現していると思なせるからである。

実際、デザインレビューにおける異なる業務分野同士、或いは、技術者と弁理士のような異なる専門分野同士が、機能分解木を媒介にして容易に理解を共有できたという成功事例[10]は、このことを強く示唆するものと考えられる。

次節では、不具合関連文献の実際の記述から、共通の理解構造としての不具合機能分解木を導出する可能性を探るため、機能オントロジーにおける概念要素がどのように表出化されているかを分析する。

### 3.2.2 失敗知識ベースにおける不具合記述の分析

不具合の言語的記述において、中核的な概念要素である主体、対象物、機能、方式に限定した描写を元に記述の様相を検討する。

まず、「破裂」に関する不具合を表現する3つの例を挙げる。

■事例 A. 「加熱による酸素ガス容器の破裂・飛翔」より抜粋

この火がそばにあった酸素ガス容器を加熱したため容器が破裂し、……

■事例 B. 「油注入中にディーゼル油タンクが脆性破壊し油が流出」より抜粋

破裂したタンクはどれも溶接に欠陥があった。

■事例 C. 「日向に置いたブタジエンボンベの爆発・火災」より抜粋

モノマーより体積が増えることにより、配管を詰まらせたり、容器を破裂させることがある。

上記3例に関して、言語学的には次のような解釈が一般的である。

□解釈 A. 主語：容器、述語：破裂する

□解釈 B. 連体修飾節：破裂した、被修飾名詞：タンク

□解釈 C. 目的語：容器、述語：破裂させる(使役)

事例 A における機能発揮主体は、主語である容器ではなく、記述の有無は別として他に存在する。他の機能発揮主体が容器に作用して複数の部分に分かれた状態をもたらす。

事例 B では、言語学的に主部/述部の関係は同定されないが、タンクが複数の部分に分かれた状態を表

明している。

事例 C は、容器が対象物として扱われており、記述されていない主体を補完すれば、最も自然な機能表現である。

これら3事例は、言語学的な観点からは表現形態も要素区分も異なるが、機能構造の観点では、ある機能発揮主体が対象物に「分割する」機能を発揮して、全体性を完備した入力状態から複数の部分に分かれた出力状態への変化を生じた、という基本構造は同一である。

また、事例 A と C には、破裂の原因が記述されており、容器破裂の因果関係が見出せる。因果関係の表現は接続詞や文間照応など多様であり、個々の独立した事態構造を持つ事象間の関係である場合もあるが、本例では、加熱や体積増加という方式で機能が達成された、すなわち熱エネルギーや膨張現象を原理とする方式概念による分割機能の達成と見なすことができる。

## 4. 不具合方式知識の構造化・組織化

本章では、前章の考察と分析を受け、不具合の記述を含むテキストから不具合の最も基本的な機能構造(不具合方式知識)を自動構築する手法について議論し、計算機で共有・再利用が可能な不具合方式知識ベースを構築する。

### 4.1 不具合機能構造理解モデル

2.1 節において、不具合の機能構造は機能分解木として記述できることを述べた。また、3.2 節において技術文献における不具合記述が異なる表現様式であってもそこには共通の機能構造を内包していることを考察した。

不具合のテキスト記述から、異なる表現様式に共通して内在する同一の機能構造を顕在化し、機能分解木の形態に転化することができれば、人間が不具合というものを理解している共通の認識構造を顕在化させたことと等価であると考えられる。

本研究では、テキストから機能分解木を導出するモデルを不具合機能構造理解モデルと呼称し、以下、その具体的な内容について考察する。

本モデルでは、テキスト全体に対する言語学的な主部、述部の弁別や厳密な統語構造の同定は行わない。主体、対象物、機能等の機能オントロジーの中核的概念要素の同定及び概念要素間の関係を元に機能構造の顕在化を行う。人間の判断や人工物の静的な状態、性質の描写等は、対象外である。

3.2.2 節において、テキスト記述には機能分解木を構成するための概念要素が含まれていること、及び異なる言語的構造パターンが同一の機能構造型を有

することを確認した。

そこで、機能分解木を構成する概念要素、同一の機能構造類型と見なされる言語パターンの同定を行うボトムアップ処理と、機能分解木全体の構造に該当要素群を布置させるトップダウン処理を組み合わせることで、テキストから機能分解木の構造化を実現することが可能となる。

不具合機能構造理解モデルは、上記のような機能概念要素と要素間関係の同定を司るボトムアップ処理と該当要素を要素間関係に基づいて機能分解木の構造に配置するトップダウン処理を組み合わせ、任意の不具合記述を不具合機能分解木に統合するモデルである(図 3)。

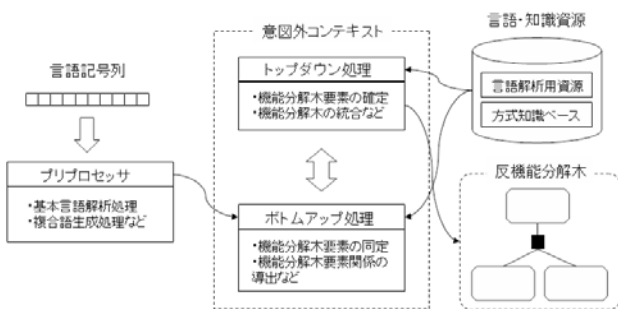


図 3. 不具合機能構造理解モデル

## 4.2 不具合方式知識の構造化支援枠組み

不具合機能構造理解モデルにより、不具合の機能構造を導出することが可能となるが、記述の省略や曖昧性等の要因で、完全な機能分解木の自動構築は困難である。

また、本モデルは、導出されたインスタンスとしての機能分解木を一般化して、階層的に組織化することも含んでいない。

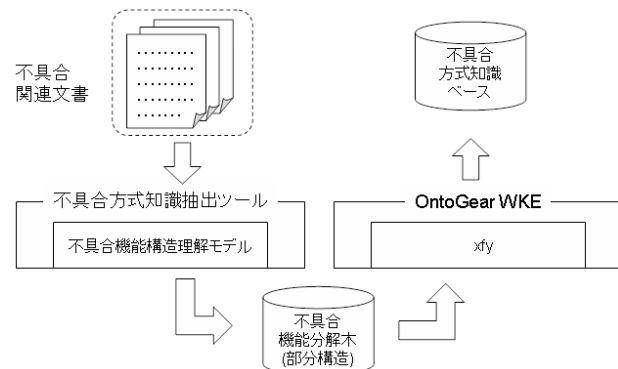


図 4. 不具合方式知識構築支援枠組み

従って、ユーザとの協調により、自動的に抽出されたインスタンス方式知識に対して補正・補完を行い、さらに汎用的な不具合方式知識として体系化する

る不具合方式知識構築支援枠組みを考案した(図 4)。

この枠組みでは、まず、方式知識抽出ツールが任意の不具合文書を入力として、不具合機能分解木(部分)を自動構築する。この不完備な不具合機能分解木に対して元の文献を参照しながら方式知識抽出ツール上で修正を行う。次に、修正後の不具合機能分解木を OntoGear WKE に読み込んで、一般化を図り、不具合方式知識としての体系化を行う。最終的な不具合木が不具合方式知識ベースに登録される。

## 4.3 不具合方式知識ベースの構築

機能・不具合統合モデルにおける不具合は、あくまでも意図しない目的を達成すると見なした機能の解釈の結果であり、不具合機能分解木から不具合のみの方式知識を組織化することは、一般的には適切ではない。

しかし、語彙のレベルで不具合の表現に固有性が高い機能語彙が見出されたように、不具合の機能達成方式として用いられやすい方式知識を便宜的に不具合方式知識と見なすことは、元となる統合モデルを損なうものではないと考えられる。

そして、不具合方式知識を体系化して方式知識ベースの形態で体系化しておくことは、不具合の機能モデリングに寄与するだけでなく、過去の不具合に関する知識を共有・再利用することに貢献すると考えられる。

そこで、前節で考察した枠組みに基づき、不具合方式知識構造化支援システムを構築して、具体的な不具合方式知識ベースの構築を行った。

失敗知識データベース 415 事例を対象に、本システムを適用して、共通の不具合方式知識を集約した結果、約 20 例の不具合方式知識を得た。抽出して組織化した例を図 5 に示す。

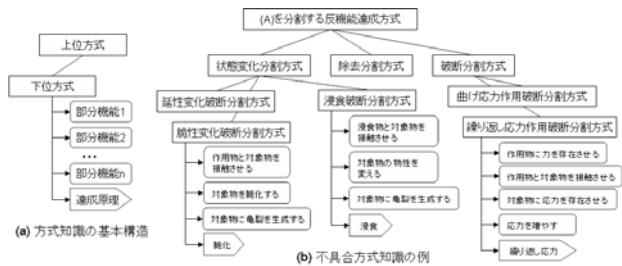


図 5. 不具合方式知識例

## 5. 構造化された失敗知識の活用

### 5.1 OntoGear による不具合機能分解木の記述とメタデータ化

OntoGear は、3.1 節において構築した不具合表層・標準語彙及び 4.3 節において構築した不具合方式知識ベースを備えている。このため、ユーザは不具合の機能モデリングを効率的に行うことができる。

OntoGear を用いて不具合機能分解木を記述する際、例えば、ユーザが「疲労破壊する」という表現を用いようとする、不具合表層・標準語彙辞書により機能概念が「分割する」であること、「応力作用方式」を含むこと、などの示唆がユーザに提示される。ユーザは、そのような OntoGear が提示するガイドに沿って、適切な機能概念の記述を行うことができる。

また、不具合方式知識ベースには、意図外の「分割する(破壊する)」機能を、「延性変化方式」「振動作用方式」「曲げ応力作用方式」などの様々な方式で達成する方式知識が格納されている。

このため、図 6 に示すように、例えば、ユーザが、「分割する」機能を設定する時に(図中①)方式検索を行うと、達成可能な方式が一覧されるため(図中②)、その中から任意の方式知識を選択し容易に部分機能列を展開することができる(図中③)。

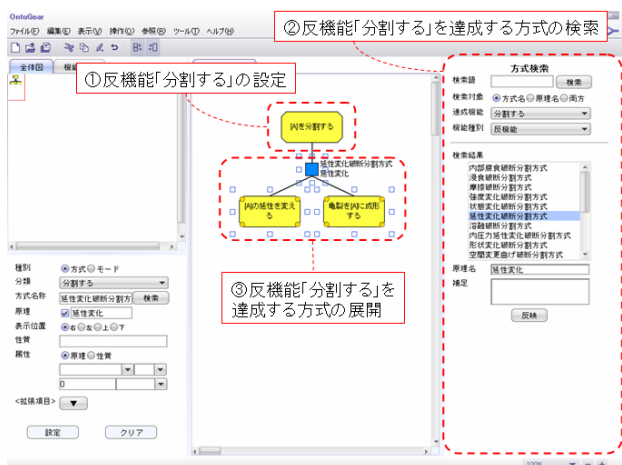


図 6. 不具合方式知識ベースの利用

不具合機能分解木の記述によって、不具合の機能構造を明確化できるが、記述された機能分解木は文書検索などへの応用が可能である。

OntoGear では、他所での応用のため不具合機能分解木を機能メタデータとして出力することができる。

機能メタデータは、Funnnotation (Functional annotation) 形式 [11] で表現され、RDF (Resource Description Framework) / XML ファイルとして出力される。出力例を図 7 に示す。

本機能メタデータの具体的な応用としては、以下のような SemanticWeb 的利用が考えられる。

- 1) 不具合報告書を参照しながら不具合機能分解木を記述する、

- 2) その記述内容を機能メタデータ形式で出力し、不具合報告書のメタデータとして関連付ける。
- 3) 検索時には、機能メタデータを Funnnotation Search System [11] など検索し文書へのリンクを表示する。

```
<?xml version="1.0"?>
<rdf RDF:xml base="http://www.hozo.jp/owlif/unnotation" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:funtnotation="http://www.hozo.jp/owlif/unnotation" xmlns:rdfs="http://www.w3.org/2000/01/19/rdf-schema#">
  <funtnotation:分割する rdf:about="">
    <funtnotation:input rdf:resource="#a_processing_object" />
    <funtnotation:output rdf:resource="#a_processing_object" />
    <funtnotation:agent rdf:resource="#a_processing_object" />
    <funtnotation:has_supplementary rdf:resource="#failure" />
    <funtnotation:selected_way>
      <funtnotation:frictional_way rdf:about="#延性変化破断分割方式" />
      <funtnotation:has_principle rdf:about="#延性変化" />
      <funtnotation:goal_function rdf:about="#分割する" />
      <funtnotation:method_function>
        <funtnotation:#変える rdf:about="#変える">
          <ontogear:id rdf:resource="" />
          <funtnotation:input rdf:resource="#a_processing_object" />
          <funtnotation:output rdf:resource="#a_processing_object" />
          <funtnotation:agent rdf:resource="#a_processing_object" />
          <funtnotation:has_supplementary rdf:resource="#failure" />
          <funtnotation:#変える>
        </funtnotation:#変える>
        <funtnotation:#成形する rdf:about="#成形する">
          <ontogear:id rdf:resource="" />
          <funtnotation:input rdf:resource="#a_processing_object" />
          <funtnotation:output rdf:resource="#a_processing_object" />
          <funtnotation:agent rdf:resource="#a_processing_object" />
          <funtnotation:has_supplementary rdf:resource="#failure" />
          <funtnotation:#成形する>
        </funtnotation:#成形する>
      </funtnotation:method_function>
    </funtnotation:selected_way>
  </funtnotation:分割する>
</rdf:RDF>
```

図 7. Funnnotation 出力例

## 5.2 OntoGear による不具合未然予防アプリ

### 一 一

設計においては、FMEA (Failure Mode Effect and Analysis), FTA (Fault Tree Analysis) 等の信頼性解析手法が同時並行的に実施される。また、実際に発生したトラブル・故障等は不具合報告書などで文書データベースに蓄積され、設計時に参照することも行われる。

しかし、上記のような別個の形態では、相互に関係があったとしても不具合の情報が離散し、連携性を保つことは難しい。結果として、必要な情報を適時的に集約することが困難になる。

一方、OntoGear は、機能モデリング時に関連情報を紐付けることができることに加え、不具合方式知識ベースに汎用的な不具合方式知識として登録する際に、参照した不具合文書などの情報を併せて記録することで関連性を保持することができる。

また、本来、機能・不具合統合モデルは、意図内・意図外に関わらず、同一の機能分解木で各々の機能構造を明示化するものであるが、両者には機能停止影響関係が存在する。そこで、OntoGear WKE では、方式知識ベースに方式知識を登録する際に、意図内の機能と意図外の機能の両方式知識間に意図内・意図外関係を付与する仕組みを設け、過去に明確な関係性があった事象構造を相互に参照できるようにしている。

これらの機能性を利用することで、ユーザが基本

機能の機能モデリングを行っている際に、OntoGearがユーザに対して予測される不具合への気付きを与えることが可能となる。また、その際に、実際に過去にどのような不具合が生じていたのかを即時的に確認することも可能となる。

本研究では、このような不具合未然予防に寄与する機能群を集約してFPoD (Failure Prevention on Demand) と称するプラグインにまとめ、プロトタイプシステムに搭載した。

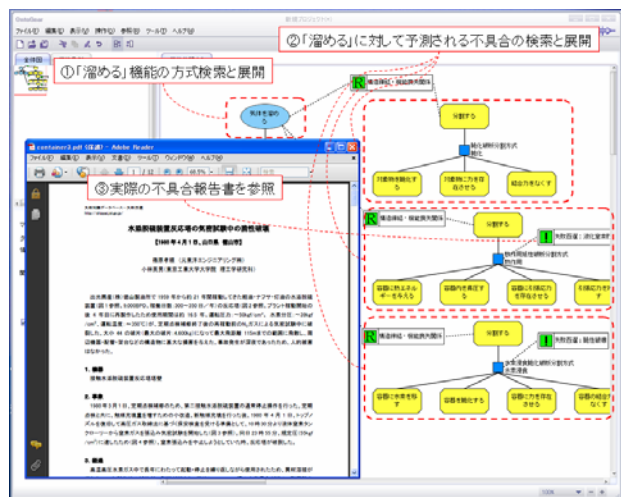


図 8. FPoD による不具合方式知識の活用例

図 8 は、容器が「溜める」機能を果たす際に、どのような不具合があるかを予測する例を示している。まず、「溜める」機能に関する方式知識を検索し、その中の一つを展開している(図中①)。

次に、「溜める」機能とその達成方式を基点として、失敗百選を題材に予め構築しておいた不具合方式知識ベースから、紐付けられている不具合方式知識を検索する。例では、可能性のある 3 つの不具合方式知識を選択し、各々を展開している(図中②)。

そして、展開された不具合方式知識のうちの一つに関して、関連付けられた元文書である失敗百選の PDF 文書を表示している(図中③)。

## 6. おわりに

本稿では、失敗知識、特に人工物の不具合に関する機能的知識の構造化と活用を効率的に行う枠組みとして、語彙表現レベルの支援枠組みと方式知識レベルの支援枠組みの 2 つについて考察した。考察に基づき、機能的知識外化ツール OntoGear を用いてプロトタイプを作成し、実際の機能性と効果を確認した。

語彙表現レベルの支援枠組みに関しては、不具合に固有の語彙を機能概念に写像する言語資源を構築

し、その言語資源を用いることで不具合機能分解木を記述する際の支援が行えることを示した。

また、方式知識レベルの支援枠組みに関しては、過去の不具合を参考にして予め不具合方式知識として構造化・登録しておくことで、概念設計時に、不具合の機能モデリングが容易になること、不具合の未然予防に貢献できること、を例示した。

今後は、不具合方式知識インスタンス自動生成の精度を高めるとともに、FPoD プラグインの洗練と言語・知識資源の充実を図ることで、現実の応用における有用性を高めていきたいと考えている。

## 参考文献

- [1] 畑村洋太郎: 続々・実際の設計 失敗に学ぶ, 日刊工業新聞社, (1996)
- [2] 国土交通省, 自動車のリコール・不具合情報, <http://www.mlit.go.jp/jidosha/carinf/rcl/index.html>
- [3] 家電製品 PL センター, 事例・社告等検索, <http://www.aeha.or.jp/plc/search/index.php>
- [4] 科学技術振興機構(JST), 失敗知識データベース, <http://shippai.jst.go.jp>
- [5] 來村徳信, 溝口理一郎: オントロジー工学に基づく機能的知識体系化の枠組み, 人工知能学会論文誌, Vol. 17, No. 1, pp.61-72, (2002)
- [6] 小路悠介, 來村徳信, 加藤義清, 筒井良夫, 溝口理一郎: 相互運用性を指向した機能・不具合知識の統合とその概念写像に基づく知識変換, 人工知能学会論文誌, Vol.22, No.1, pp. 78-92, (2007)
- [7] 高藤淳, 來村徳信, 溝口理一郎: オントロジー工学と XML 技術に基づく技術知識統合管理プラットフォームの構築, 人工知能学会論文誌, Vol.23, No.6, pp. 424-436, (2008)
- [8] 株式会社ジャストシステム, xfy, <http://www.xfy.com>
- [9] 高藤淳, 來村徳信, 溝口理一郎: オントロジー工学と自然言語処理技術による機能分解木記述支援の枠組み, 電子情報通信学会 NLC 研究会, (2007)
- [10] 溝口理一郎, 來村徳信, 布瀬雅義: オントロジー工学の成功事例 ～機能オントロジーに基づく生産技術知識の共有・再利用～, 人工知能学会研究会資料, SIG-KBS-A202-07, (2002)
- [11] Kitamura, Y., Washio, N., Koji, Y., Sasajima, M., Takafuji, S., Mizoguchi, R.: An Ontology-Based Annotation Framework for Representing the Functionality of Engineering Devices, In Proc. of ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME IDETC/CIE 2006), September 10-13, 2006, Philadelphia, Pennsylvania, USA, DETC2006-99131, (2006)