

宇宙航空研究開発機構研究開発報告

JAXA Research and Development Report

宇宙機的设计・運用へのオントロジー工学の適用に関する研究

加藤 義清, 筒井 良夫

2005年9月

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

宇宙航空研究開発機構研究開発報告
JAXA Research and Development Report

宇宙機的设计・運用へのオントロジー工学の
適用に関する研究

A Study on Application of Ontology Engineering to Design and Operation
of Spacecrafts

加藤 義清、筒井 良夫

Yoshikiyo KATO, Yoshio TSUTSUI

総合技術研究本部 情報技術開発共同センター

Information Technology Center

Institute of Space Technology and Aeronautics

2005 年 9 月

September 2005

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

目 次

1. はじめに.....	1
2. オントロジーに基づく知識獲得・共有の枠組.....	3
2.1 タスク実施者.....	3
2.2 タスク支援アプリケーション.....	3
2.3 情報モデル層.....	3
2.4 オントロジー層.....	4
3. 不具合知識表現のためのオントロジー.....	4
3.1 機能オントロジー.....	5
3.2 構成・振舞いオントロジー.....	5
3.2.1 基礎概念.....	5
3.2.2 構成・振舞いモデルのメタモデル.....	6
3.2.3 部品概念.....	6
3.3 不具合オントロジー.....	8
3.3.1 基礎概念.....	9
3.3.2 不具合知識表現のメタモデル.....	10
3.3.3 不具合モード.....	11
4. 不具合知識獲得支援ツール.....	12
4.1 ツールによる不具合知識獲得のシナリオ.....	12
4.1.1 オントロジーの定義.....	13
4.1.2 機能モデルの記述.....	13
4.1.3 部品の定義・利用.....	13
4.1.4 構成・振舞いモデルの記述.....	14
4.1.5 不具合モデルの記述.....	14
4.2 構成・振舞いモデル作成支援ツール.....	14
4.2.1 システム構成の定義.....	14
4.2.2 部品の定義.....	15
4.2.3 振舞いの定義.....	15
4.2.4 機能モデルとの関連付け.....	15
4.3 不具合モデル作成支援ツール.....	15
4.4 ツールにより記述されたモデル例.....	16
5. オントロジーに基づくアプリケーションの枠組み.....	17
5.1 データ層.....	17
5.2 オントロジー層.....	17
5.2.1 オントロジー操作・管理機能.....	19
5.2.2 インスタンス操作・管理機能.....	19
5.2.3 汎用データ構造に対する操作機能.....	19
5.2.4 オントロジー.....	19

5.3	モデル依存層	20
5.4	汎用データ構造操作機能	20
5.5	オントロジーマッピング機能	21
6.	評価	21
6.1	他の不具合知識表現手段との比較	22
6.2	モデルの表現力	22
6.3	モデルおよびオントロジー上の推論	23
6.3.1	モデル独立な推論	23
6.3.2	モデル依存の推論	23
6.4	開発されたオントロジーの規模	23
6.5	オントロジー記述環境の能力	23
6.6	オントロジー制約	24
6.6.1	選択オプションの提示	24
6.6.2	図形オブジェクトの指定・選択の制御	24
6.6.3	入力語彙チェック	24
6.7	オントロジー変更のアプリケーションへの影響	24
6.8	知識共有支援システム（ツール）の能力	25
6.8.1	パフォーマンス	25
6.8.2	システムフレームワーク	25
6.8.3	アプリケーション機能	25
6.9	評価のまとめ	26
7.	関連研究	27
7.1	不具合知識の獲得・共有・利用に関する研究	27
7.2	オントロジー・マッピング機能	27
8.	おわりに	28
	謝辞	29
	参考文献	29
	付録 「評価・分析」－知識共有の視点－	31

宇宙機の設計・運用へのオントロジー工学の適用に関する研究

加藤義清, 筒井良夫 (宇宙航空研究開発機構)

A Study on Application of Ontology Engineering to Design and Operation of Spacecrafts

Yoshikiyo Kato and Yoshio Tsutsui (Japan Aerospace Exploration Agency)

Abstract

Space systems are typical large complex systems. Knowledge sharing among stakeholders becomes crucial for a sound and efficient development or operation of a space system, as the scale and complexity of such a system grow. There are many activities involved in assuring safety or reliability of a system, and many software tools have been developed to support each of specific activities. However, such tools do not support coordination among different tasks, especially at the knowledge level.

This report presents a framework for capturing fault knowledge based on ontology engineering. The purpose of the framework is to facilitate externalization of fault knowledge, which is to be shared among human or software tools. Two tools for supporting construction of ontology-based models which have been built are described. We also explain the application framework for building ontology-based software we adopted in developing those tools, and address the problem of accommodating the changes of ontology in ontology-based software by ontology mapping.

1. はじめに

宇宙ステーション, 人工衛星, ロケット等に代表される宇宙システムは典型的な大規模複雑システムである。その開発・運用には大勢の様々な分野の専門家が携わるが, システムの規模が大きくなりその複雑さが増すほど, システムの開発・運用を円滑かつ着実に進めるには関係者間でシステムに関しての知識共有が重要となってくる。しかしながら, その重要性は認識され, 宇宙航空研究開発機構 (以下, JAXA) でも不具合情報データベースの構築等の取組みはされているものの, 体系だった知識共有の枠組みは不十分というのが現状である。一方, 宇宙システムの開発過程ではシステムの安全性・信頼性を確保するために様々な活動がなされ, それらの活動を助けるための支援ツールも開発・利用されている。各活動で用いられる知識には共通部分も多いと予想されるが, 現状では活動間で十分な知識連携が出来ているとは言えない。

知識共有の基盤を与えるものとして, 近年注目されているのがオントロジー工学である。溝口らはオントロジー工学の目的を人間と計算機双方が理解できる共有／再利用が可能な実世界の対象のモデル構築の基盤を与えることとしている[12]。オントロジーとは, モデル

化対象の基本概念とその体系を明示化したものである。この明示化された基本概念の体系がオントロジーという形で与えられることにより、それを用いて記述された知識の共有・再利用が可能となる。

そこで、本研究はオントロジー工学に基づく宇宙システム開発のための知識共有の枠組を提案し、その有用性を実証することを目指した。宇宙システムの開発・運用において共有されるべき知識は多岐に渡るが、本研究では特に不具合に関する知識に焦点を当てた。それは、システムの安全性・信頼性を確保する上で不具合を識別し、対処することが重要であるのにも関わらず、個々の不具合というのは個別具体的であり、従来の方法では体系化された知識として整理することが難しかったためである。

本研究では不具合に関する知識を表現するのに、以下に述べる二つのアプローチを採用した。一つ目は、補助機能の存在理由としての不具合に着目するものである。補助機能とは人工物がその主要な機能を発揮するのに直接は寄与しないが、その品質や効率を改善する目的で付与されるものである。改良設計を行う場合には、このような補助機能が改良対象となることが多く、重要な設計知識の一つである。そこで本研究では、来村らによる機能的な設計知識を体系的に明示化する枠組み[17][18]を発展させ、補助機能とともに、その存在理由である不具合を不具合原因発生木として記述する枠組みの研究に取り組んだ。このアプローチによる研究は、大阪大学産業科学研究所溝口研究室との共同研究により実施された。溝口研究室は機能オントロジーに基づく不具合知識表現の理論的な検討、不具合知識から FMEA 表現への知識変換方式の開発を、JAXA は実例に基づいて記述された知識、および知識変換により得られる FMEA シートの妥当性についての検証をそれぞれ担当した。

一方、二つ目のアプローチでは FMEA (Failure Modes and Effects Analysis) や FTA (Fault Tree Analysis) 等の信頼性解析手法[15]で扱われる不具合を対象とした。信頼性解析手法によって不具合の具体的な表現方法は異なるものの、同じシステムを対象とする以上、不具合について何らかの共通的な知識が存在すると考える。そこで、本アプローチでは JAXA における不具合に関する語彙に基づいて不具合オントロジーを構築し、解析手法に依存しない形で不具合に関する知識を記述するための枠組みを提案する。本稿では本研究の範囲のうち、後者について述べる。前者の詳細については別稿[14][21]を参照されたい。

以下、本報告の構成を述べる。まず 2 章で本研究の提案するオントロジーに基づく知識共有・知識連携の枠組みについて説明する。次に 3 章で、開発対象システムを表現する各モデルを基礎付けるオントロジーについて議論する。4 章では 2 章で述べた枠組み、および 3 章で述べたオントロジーに基づいて開発された不具合知識獲得支援の為のツールについて概説する。5 章でツールを用いて構築した人工衛星電源サブシステムの各種モデルについて解説し、本研究のアプローチの有効性を議論した後、6 章で本研究の評価について述べる。7 章で関連研究に触れた上で、8 章で本報告を締めくくる。

2. オントロジーに基づく知識獲得・共有の枠組

宇宙開発分野の知識の多くは、共有の基礎となる概念を正確に規定し、それらを用いて高次の概念を明確な意味論のもとで組み立てて表現することで、共有できると考えられる。これを前提として本研究が採用している、オントロジー技術と計算機を用いた知識獲得・共有の枠組みを図 1 に示す。この枠組みには 4 つの要素、すなわち、1) 解析タスクを実施する人間、2) 各タスクを支援する計算機アプリケーション、3) アプリケーションがアクセスする情報モデル、および 4) 情報モデルを構成する概念を供給するオントロジー、が含まれている。なお、本研究における主要な共有対象は宇宙システムの潜在的または顕在化した不具合に関する知識であるが、この知識が強く依存する対象システムの機能的知識、およびシステムを構成する要素と各構成要素の振舞いに関する知識（構成・振舞い知識）を含めて記述の対象としている。

2.1 タスク実施者

本研究では、宇宙開発・運用関連のタスクのうち、FMEA, FTA, 不具合診断など、宇宙機の高信頼化に直接係わるタスクに焦点を当てており、タスク実施者が保有する不具合に関する豊富な知識・経験が獲得・共有の源泉である。不具合知識の他の重要な源泉としてこれまでに蓄積されデータベース化された不具合記録があるが、これらについても、本研究で提案している表現に変換することで、本枠組みで捉えることができる。本枠組みにおけるタスク実施者のもう 1 つの重要な役割は、知識共有に関する他の関係者とともにオントロジーの内容について理解し合意することである。

2.2 タスク支援アプリケーション

各タスク(FMEA 等) を支援するアプリケーションは、タスクを特徴づける固有の知識に基づいてタスクの効率的な実施を支援する一方、解析者との相互作用の過程で解析者の知識・経験に基づく判断を捕捉し、また、解析結果を保持する役割を果たす。

ここで重要になるのが、アプリケーションがアクセスする情報モデルが適切に階層化され、各タスクやアプリケーションに固有な部分と独立な部分に分離されていることである。これにより、アプリケーションの手続きも階層的に構成でき、手続き間の相互運用性が実現可能になる。なお、今回開発したアプリケーションはタスク固有のものではなく、構成・振舞いモデルと不具合モデルの記述を支援するものであるが、これらは FMEA など特定の解析手法向けの支援ツールを開発する場合に容易に利用できる。アプリケーションの詳細については 4 章で述べる。

2.3 情報モデル層

情報モデル層には、タスクやドメインに依存しない共通的な基礎概念のみによって構成されたモデル、タスク固有の概念を含んだモデル、さらに、タスクを支援するアプリケー

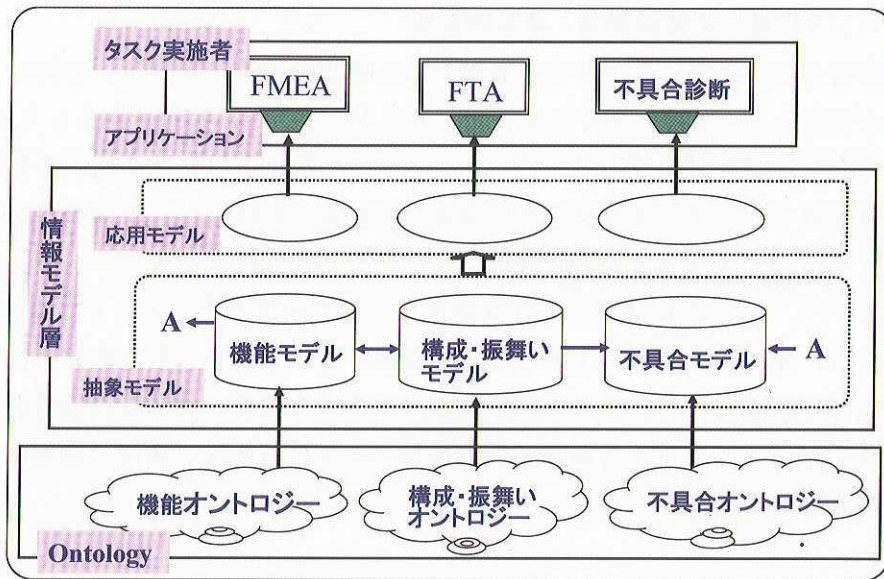


図 1 知識獲得・共有の枠組み。

ション固有の情報（例えば、GUI におけるオブジェクトの表示位置情報等）を含んだモデルなどが存在する。これらはドメイン、タスク、アプリケーションに対する依存の程度やそのモデルに付与される属性の多少（抽象性）による階層をなす。これらの様子を図 1 では「抽象モデル」と「応用モデル」に二分して表現しているが、実際には中間に位置するモデルも多く存在する。図中、応用モデルがタスクへの依存性の強い情報モデルである。一方、タスク独立な抽象モデル層に位置づけられている機能モデルは具体的な対象系の機能を表現したものであり、同じく構成・振舞いモデルは対象系の構成と振舞いを記述したものである。さらに、抽象層の 3 つ目の不具合モデルは各解析の結果や過去の不具合知識を表現する共通のモデルとして利用される。この共通性が、タスク間、あるいは他プロジェクトとの不具合知識の共有と流通を促進すると考えられる。

2.4 オントロジー層

オントロジーは情報モデルを記述する際の基本概念とその構造を規定するメタモデルを提供する。この層には、is-a 関係にある多くの概念群が存在し、それらが情報モデルのデータ構造を制約づけている。このことによって、そうでなければお互いに「孤立」しがちな情報モデル層の各モデルの構造間の関係が明示化される。これが本研究におけるオントロジーの重要な役割である。本研究では、共有する知識の種類に対応して、機能オントロジー、構成・振舞いオントロジー、不具合オントロジーという 3 つのオントロジーを開発した。各オントロジーの詳細については次章で述べる。

3. 不具合知識表現のためのオントロジー

本研究では、不具合知識の獲得と共有を主要な目標としているが、その記述と理解にと

って不可欠な、対象システムの機能的知識および構成・振舞い知識についてもオントロジー化の対象とした。これらは、設計段階で決定されるものであるが、信頼性解析の主要な源泉情報となる。

本研究で開発されたオントロジーは工学ドメインオントロジー[20]の一例であり、信頼性解析や不具合解析を支援する計算機システムにおいて、具体的な宇宙機的设计モデルである機能モデルや構成・振舞いモデル、および、そこで起き得る不具合、原因、影響を捉えた不具合モデルに対して、その構造に関する規約（メタモデル）を与える。また、概念の定義や構造に関するユーザからの問い合わせに対する出力を通じて、計算機と人間の知識共有を基礎づけるものとなる。さらに、基礎的な概念群を定義することにより、関係者のシステムに関する理解や捉え方に含まれる恣意性[20]の排除が期待できる。以下、オントロジー記述環境法造[11]のエディタを用いて実装した機能、構成・振舞いおよび不具合の各オントロジーを概観する。

本稿では詳しく述べないが、上述のオントロジーの他に、宇宙システムの分類、ロケットおよび衛星の一般的な要素の階層（サブシステムやコンポーネント）、宇宙機の運用に係わる地上設備、およびそれらの設備が配置される施設、設備についての分類を、オントロジーとして記述した。

3.1 機能オントロジー

システムの不具合の多くはシステムの機能が不全に陥ることにより認識される。そのような意味では、不具合に関する知識を記述するに当たって、システムの機能に関する知識は、不具合知識を記述したり、理解したりする際の重要な前提知識の一つであると言える。機能オントロジーは対象システムの機能的側面を記述・理解するための基礎概念群を提供する。本研究における機能オントロジーは来村らによる「機能的知識・オントロジー体系」[18]に基づいている。

3.2 構成・振舞いオントロジー

本研究では、対象システムの要素（サブシステム、コンポーネント、部品など）の構成と各要素の振舞いを記述したものを対象システムの「構成・振舞いモデル（C&Bモデル）」と呼んでいる。構成・振舞いオントロジーは、C&Bモデルに含まれるべき基礎的な概念群とC&Bモデルの構造定義を与える。後者は、必要最小限の属性を持つ抽象的なものから応用目的に応じた属性を追加したものまで、抽象レベルの異なる幾つかのモデルから成り、C&Bモデルに対してメタモデルを供給している。

3.2.1 基礎概念

構成・振舞いオントロジーでは、1) 要素の「振舞い」、2) 要素の振舞いの中で、要素が作用する対象となる「オブジェクト」（電気、熱、力など）、3) オブジェクトの伝播媒体である論理概念としての「接続」、4) オブジェクトの「出入り口」として各要素に所属する「ポ

ート」の4つを対象系の構成と振舞いを表現する重要特性として同定し、相互の関係を含めて定義した。

3.2.2 構成・振舞いモデルのメタモデル

「対象系」の最も抽象的な概念定義を図 2 に示す。この定義では、対象系がある外部特性を持ち、また同じ外部特性を持つ子孫の要素を持てることを規定しているが、外部特性については何も規定しておらず非常に抽象的な定義になっている。ここで、オントロジー定義の記法に触れておくと、図中 a/o は attribute-of を p/o は part-of を、これらの右隣の数字や*は繰り返し数(cardinality)を表す。属性は is-a 関係にあるクラス間で常に継承されるが、属性スロットの表示は、そのクラスで新たに追加された場合はその説明（ロール名）が青色で、上位概念から継承された場合は継承パスとともに赤色で表示される。後者は、属性の制約クラスを上位のそのサブクラスに変える等、上位での属性定義を変更する場合の override 操作によって生じる。オントロジー定義に関する記法の詳細については[11]を参照のこと。

C&B モデルは、3.2.1 で述べた4つの特性に加えて、5) その系が実現している機能を特定する「Function」を属性として与えた「対象系 withCB」として概念化した(図 3)。これらの属性は図 2 の外部特性の部分に相当する。但し、図には示してはいないが、「対象系」との間に「構造対象系(1)」というクラスが定義されており、対象系 withCB はその特殊化として導入されている。さらに、対象系 withCB のサブクラスとして、今回の研究目的にとって重要な不具合などの安全・信頼性・保全性知識（SRM 知識）を表す属性を追加した「対象系 withCBplusSRM 知識」の概念を導入した(図 4)。今回開発したアプリケーションは C&B モデルを後者のインスタンスとして生成する。この SRM 知識の属性追加によって、後述の「不具合モードを随伴する部品」の概念を対象系 withCB の下位概念として位置づけることができる。

このように、基本的な概念の定義から出発して、目的に応じた属性の追加によるサブクラスの生成を行うことで、オントロジー層に抽象レベルの異なる幾つかの層が出現し、これらがアプリケーションを通じてユーザが生成する情報モデルのスキーマとそのメタモデルを提供することになる。

3.2.3 部品概念

システムの開発では様々な部品が使用される。部品は通常、その開発に先立って存在する。当該システムの設計時に新たに設計されるものもあるが、この場合を含めて、再利用性が部品の本質的な特性である。これにより開発効率の向上だけでなく、部品自身とこれが組み込まれるシステムの信頼性向上への寄与が期待できる。このことを反映するため、前述の「対象系 withCBplusSRM 知識」のサブクラスとして「部品 with SRM 知識」を定義した。その定義部を図 5 に示す。

部品は構造を持つ1つのシステムであり、ダイオードなど比較的単純な構造の部品もあれば、SoC (System on Chip)など複雑な構造を持つ部品もある。したがって部品のライブラリは概念的な階層構造(部品カテゴリの is-a 木)として表現される。「部品」という用語はクラスとインスタンスのいずれをも指しうるが、本研究では、部品ライブラリをクラスと捉え、その中から選択されて、具体的な対象システム(例えば、人工衛星等)の C&B モデルの中に埋め込まれたときに、インスタンス化されると見なす。

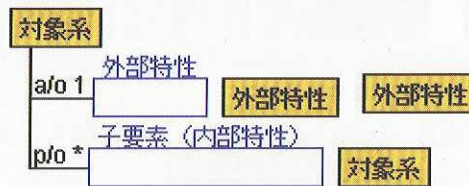


図2 概念「対象系」の定義.

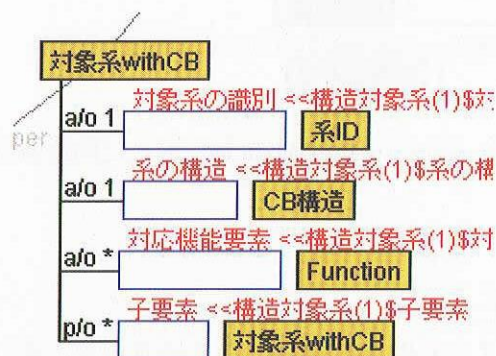


図3 概念「対象系 withCB」の定義.

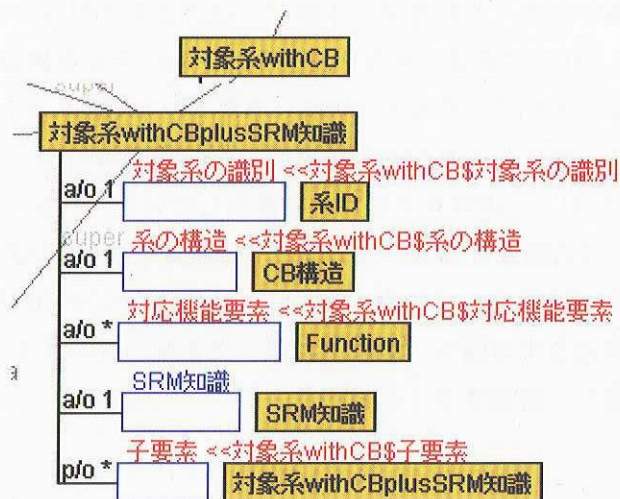


図4 概念「対象系 withCB plusSRM 知識」の定義.

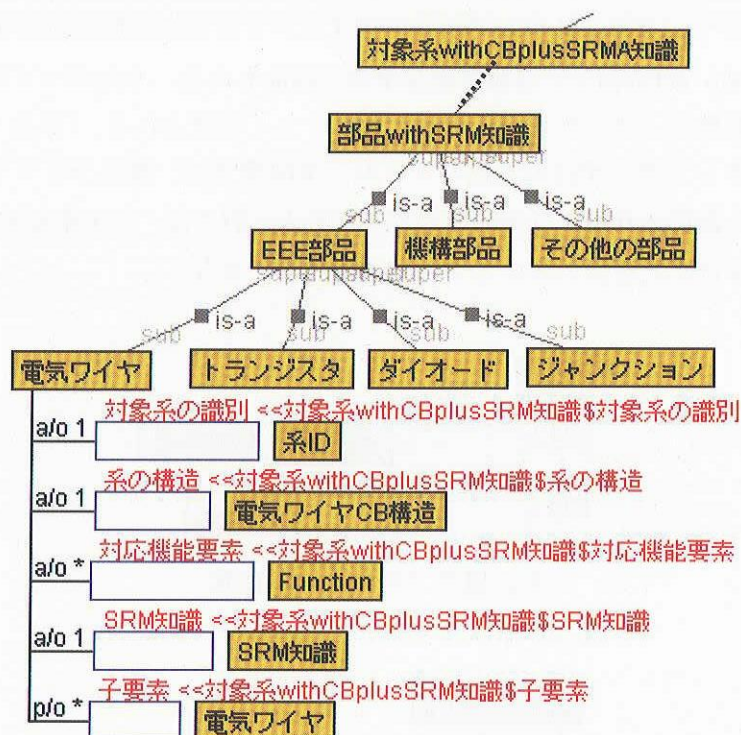


図 5 概念「部品 withSRM 知識」および関係概念の定義.

3.3 不具合オントロジー

実機で発生する望ましくない状態や事象である不具合を、それと何らかの因果関係にある幾つかの具体的な状態・事象とともにネットワーク表現したものを「不具合ネット」と呼ぶ。1つの不具合ネットは、原因や伝播影響が十分詳細に記述されていれば、完結性の高い1つの不具合知識となる。不具合オントロジーは、この不具合ネットを表現するための基礎的な概念群と、これらで構成された不具合ネットのモデルを供給している。後者は、不具合ネットに対するメタモデルであり「状態・事象ネット(SE ネット)」と呼ぶ。

なお、人工システムにおける不具合知識の基盤となる概念群については、故障オントロジー[16]によって扱われており、本研究でもこれを採用している。一方、厳しい信頼性が求められる宇宙開発分野では信頼性管理等が精力的に実施されており、JAXA 標準文書でも不具合に関する主要な概念が規定されている。従って、オントロジーの構築に当たってはこれら既存の定義との適合化を十分図り、最終的に残った乖離はオントロジー間のマッピング機能(5.5 節参照)を拡張して対処することとした。

3.3.1 基礎概念

状態・事象, 事象時間, 不具合, 不具合の原因と影響, 影響伝播における因果関係等, 不具合に関する基礎的な概念を同定し定義を与えた. 定義内容は基本的に[16]で与えられた定義に従っている. 以下, 本稿のほかの箇所で言及される重要概念について説明する.

状態と事象

ある系における状態(State)と事象(Event) は不具合の概念を構成する最も基本的な概念である. まず, 状態は1つまたはそれ以上の論理的または物理的パラメータと値の組である. 事象は, 1) 原因状態, 2) 原因状態から何らかの因果関係により生じる結果状態, 3) 事象の発生箇所, 4) 原因状態が生じてから結果状態が発生するまでの時間(推移時間)という 4つの属性で定義される.

不具合

不具合とは, ある系における望ましくない状態や事象の発現である. 故障は機能(ある目的のもとで対象系に求められている振舞い)の喪失であり不具合の典型である. 宇宙システム分野では不具合を「システムの1つ以上の特性が要求と合致しない又は異常な物品の状態」と定義されており, 故障に加えて, 「異常」, 「故障」, 「偏差」, 「欠陥」, 「不足」及び「機能不良」が不具合の下位概念とされる. 一方, 設計書の誤りや設計者のミスは「誤り」, 「フォールト」と呼ばれ, 不具合とは区別されている.

本研究では, 不具合知識をその原因や影響を含めて複数の事象・状態の間の伝播と捉え, まず, 誤りやフォールトを含めた広い概念を「不具合(0)」として捉え, 次いで, 人工物に限定したものを不具合(1)とした.

状態遷移, 事象伝播の因果性

1つの状態の遷移, 事象における原因状態から結果状態への影響伝播, あるいは事象から事象への影響伝播には何らかの(多くの場合物理的な機序による)因果性が関与する. 以下は, この因果性を原因が生じた場合の結果事象の発生尤度の観点から分類したものである.

- ◆ 必要(Necessary)
- ◆ 十分(Sufficient)
- ◆ 促進(Promotive)
- ◆ 抑制(Resistive)
- ◆ 禁止(Inhibitive)

実際に起きる不具合は, 他の不具合の発生をマスクする可能性があり, 有効な不具合処置を同定するには, 単に生じた不具合やその原因・影響だけでなく, 起きえた不具合やそれが起きなかった理由を含めて, 不具合解析で明らかにすることが重要である. この観点か

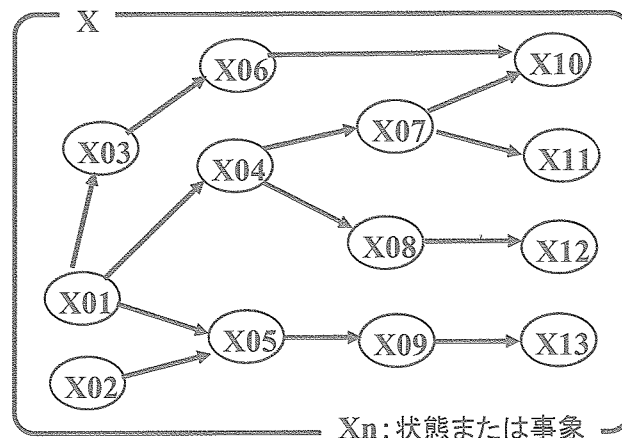


図6 SE ネットのイメージ.

ら，上述のような因果性を記述する能力を不具合モデルに持たせることとした．

状態・事象の発生位置

状態や事象の発生位置の概念は，実際に起きた不具合の位置，原因や影響の伝播先の位置，取られる対策の実施位置など，不具合知識の記述の中で頻繁に出現する．このような発生位置の概念は，組織，施設，設備，対象系の階層名(衛星やロケットのサブシステムやコンポーネントのジェネリックな名前)など，対象系の設計に独立な上位レベルのクラス名と実際の設計で与えられる個別の要素名で構成される．

3.3.2 不具合知識表現のメタモデル

不具合の具体的な知識表現である不具合モデルの構造（メタモデル）を，「不具合」の上位概念である「状態・事象」をノードとするネットワーク，「SE ネット」によって規定した．SE ネットのイメージを図6に示す．

この図で，ネットの全体 X は，1つの不具合知識のまとまりに対応し，同時に，これを知識の一部とする上位のネットワーク（知識）の1ノードと見なされる．図中の各ノードは X の子ノードである．また各有向線分（アーク）は両端のノードの間の何らかの因果性による影響関係を表している．

SE ネットの定義を図7に示す．以下，図6のノード X_{ij} にこの定義を適用した場合について説明する．本定義は C&B モデルの定義と同様，自己言及的になっていて，「 X_{ij} はあるクラス（概念）で制約された自分の特性を持ち，それと同じクラスの特性を持つ子孫を持つことができる」と読める．うち「特性」部分として，自分の識別（ノード ID= X_{ij} ），自分が表している状態や事象（不具合を含む）の内容記述（SE 記述），自分の親の識別（ X ），自分と他の状態・事象との影響伝播関係の記述（ノード構造）などが指定されている．ま

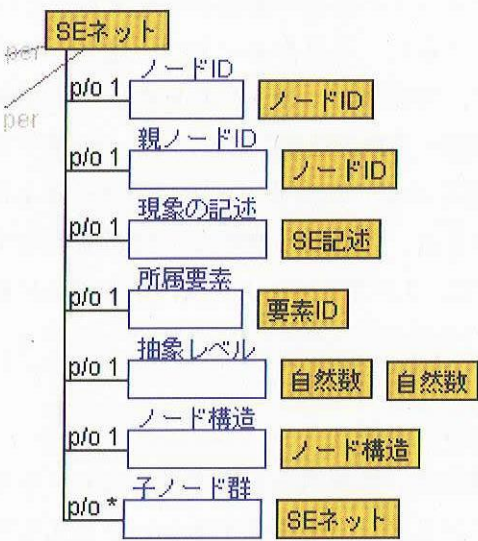


図 7 概念「SE ネット」の定義.

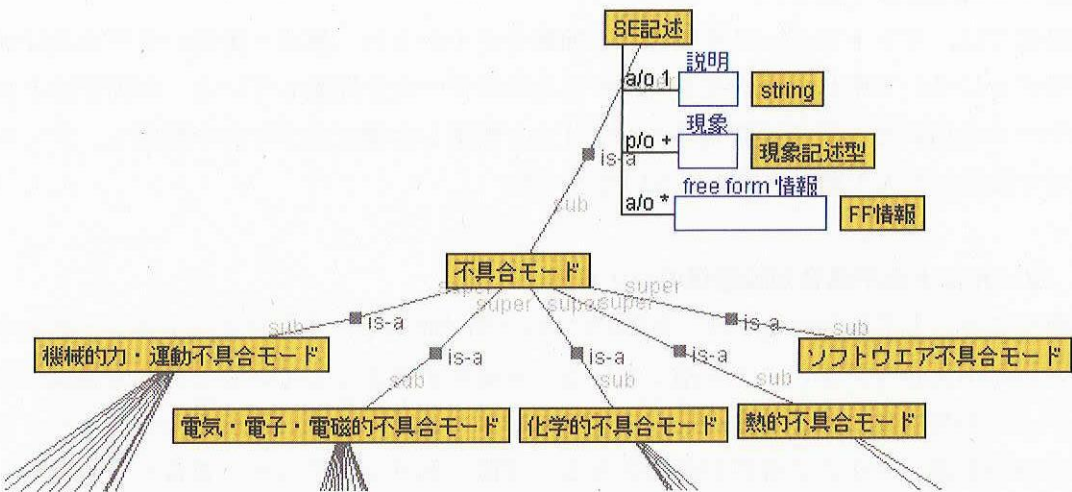


図 8 不具合モードの分類.

た、子孫部分（子ノード群）は、その原因や結果・影響などその不具合を説明する内容（ネットワーク）に対応する。（これは図 6 では表示されていない。）

3.3.3 不具合モード

不具合モードとは発生する可能性のある不具合を予め類型化したものであり、JAXA では「故障・異常モード」と呼ばれる。電気配線に用いられているワイヤや接続部の「Open」では、瞬時に破断する場合やそのハンダによる接合面が徐々に遊離する場合など、実際の現象は多様であるが、これを「断線」として類型化したものが不具合モードである。複雑・

大規模で多数の要素を抱えるシステムで、可能な不具合を漏れなく把握することは大きなコストを要するが、解析に先立って不具合の類型が存在することは、労力の削減と、解析の完全性の向上に寄与する。さらに、不具合モードは、過去の不具合経験・知見を含む不具合知識のレポジトリとして、当該設計に過去の知見を生かす役割を担うことになる。本研究では、不具合モードを「状態・事象記述(SE 記述)」(図 7)のサブクラスとして定義した。図 8 に不具合モードの分類例を示す。不具合モードは、ツールを用いて不具合モデルを作成する際にそのリストが表示され、そこから選択して記述中のモデルにインスタンスとして設定することができる。また、3.2 で述べたように、予め部品に割り当てておくこともできる。

4. 不具合知識獲得支援ツール

オントロジーにより基礎概念及び情報構造が規定されたモデルを構築するに当たって、オントロジーエディタ上でインスタンスを生成するという方法も可能ではあるが、オントロジーエディタはオントロジーに関して汎用的な操作を提供するものであり、領域の専門家にモデルを書いてもらう場合に熟知性 (familiarity)、学習性 (learnability) あるいは効率性の観点から現実的ではない。

本研究では、オントロジーに規定される抽象モデルのうち、構成・振舞いモデル及び不具合モデルについてそれらの構築を支援するためのツールを開発している。本節ではこれらのツールを用いた不具合知識獲得のシナリオを提示した後に各ツールを説明し、ツールによって記述した人工衛星モデルについて述べる。

4.1 ツールによる不具合知識獲得のシナリオ

本研究における不具合知識とは、具体的には3.3 節で説明した不具合オントロジーにより規定される不具合モデルのことを指している。本研究が対象としている不具合知識は宇宙システムの開発段階における安全性・信頼性を確保する為の活動に関わるものであり、具体的な開発対象システムの存在が前提となる。実際、あるシステムの不具合知識を記述しようとするれば、どの機能が影響を受けるのか、不具合と認識される事象がシステムのどこで発生し得るのか等、開発対象システムの機能や構成に言及する必要がある。そこで、本研究では、不具合知識を獲得する前提として、開発対象システムに関する知識が明示化され、共有されていることが重要だと考え、不具合モデルに先立って機能モデルや構成・振舞いモデルを記述するシナリオを想定する。

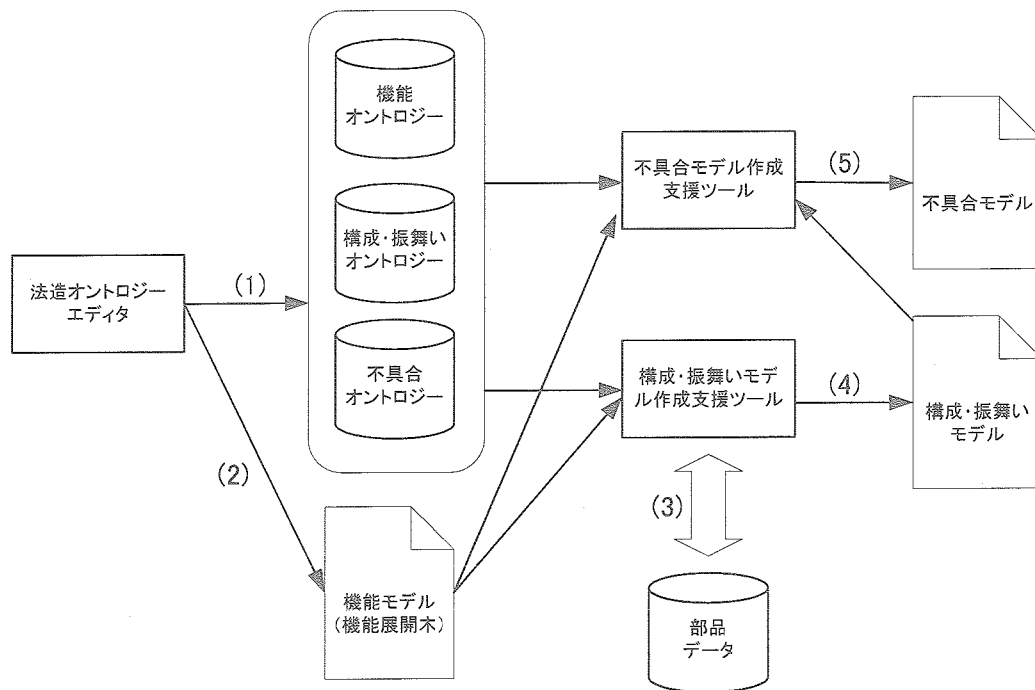


図 9 支援ツールによる不具合知識獲得のシナリオ.

図 9 に本研究におけるツールを利用した不具合知識獲得のシナリオの概要を示す。以下、その内容を、順を追って説明する。

4.1.1 オントロジーの定義

抽象モデル層の各モデルを基礎付けるオントロジーを定義する。これは法造オントロジーエディタ[11]を用いて行う。一度定義されたオントロジーは不変ではなく、必要に応じて、更新・拡充されることも想定している。特に、部品、不具合モード、状態・事象の発生位置等はオントロジーの他の部分に比べて追加・更新が頻繁であることが予想される。オントロジーの変更の問題については後述する。

4.1.2 機能モデルの記述

機能モデルを法造オントロジーエディタのモデル作成機能を利用して記述する。機能オントロジーで定義された機能概念や方式概念をインスタンス化し、インスタンス間を関係付けることによって機能モデルの記述を行う。本研究では機能モデルを記述するための支援ツールは開発していないが、将来的には機能的知識体系化の方法論を実装した SOFAST [19] 等のようなソフトウェアとの連携も考えられる。

4.1.3 部品の定義・利用

構成・振舞いモデル作成支援ツールを用いて、構成・振舞いモデルで利用する部品の定

義を行う。記述対象の中に繰り返し現れるような要素を部品として定義しておくことにより、構成・振舞いモデルの記述コストを軽減することが出来る。また、不具合モードの設定など、部品に固有の知識を定義し、不具合モデル記述の際に利用する。

4.1.4 構成・振舞いモデルの記述

構成・振舞いモデル作成支援ツールを用いて、「要素」、「ポート」、「接続」、「環境」など構成・振舞いオントロジーで定義される概念で開発対象システムの構成・振舞いモデルを記述する。この際、図 9 の(3)で定義した部品なども利用できる。

4.1.5 不具合モデルの記述

不具合モデル作成支援ツールを用いて、これまでに記述した機能モデルや構成・振舞いモデルを参照しつつ、システムに起こりうる、あるいは起きた不具合について、3.3 節で述べた不具合オントロジーの枠組みに則った形で不具合モデルを記述する。

4.2 構成・振舞いモデル作成支援ツール

構成・振舞いモデル作成支援ツール(以下、CBModeler)は3.2 節で述べた構成・振舞いオントロジーの定義に則った形で構成・振舞いモデルの記述を支援するためのアプリケーションである。

2 章でも述べたように、知識共有の前提として知識を記述する基礎となる概念が正確に規定され共有されている必要がある。本研究における知識獲得の直接の対象は不具合知識であるが、その知識は開発対象としているシステムの機能・構成・振舞いに関する知識とは独立したものではありえず、それらの知識を含む形で記述されなければならない。CBModeler は不具合知識を記述する基礎となる開発対象システムの構成・振舞いモデルの記述を支援する。

以下、CBModeler の主要な機能について述べる。

4.2.1 システム構成の定義

3.2 節で述べたように、構成・振舞いモデルにおけるシステム構成の基本要素は「要素」である。要素は階層的に構成されるとともに、同階層に存在する要素同士については「ポート」を介して接続関係により結び付けられる。

CBModeler では、ある要素について、その子要素、兄弟要素、及び環境要素を作成できる。図 10 に示した CBModeler の画面中央に表示されているのが構成・振舞いモデルで、SAT というラベルの付いた一番大きな箱が編集対象の要素を示している。SAT の中表示されている箱は SAT の子要素であり、SAT の左側に表示されている SUN, EARTH, GSE, GME は環境要素である。この図では表示されていないが、もし編集対象の要素が親要素または兄弟要素を持つ場合にはそれぞれに対応する箱が環境要素と同様に、編集対象要素の箱の

外側に描画される。ユーザは各種要素の追加，削除，移動，大きさの変更を自由に行うことができる。

要素の辺に表示される数字を伴った小さな四角形は，その要素が持つポートを表している。要素の間で接続関係を結ぶには必ずポートを介する必要がある。

各要素をダブルクリックすることにより，その要素が編集対象となり画面が切り替わる。また，画面左側パネルの上側のツリービューは要素の階層構造を表しており，ツリー上の要素をクリックすることにより，その要素を編集対象にして画面を切り替えることも可能である。

4.2.2 部品の定義

システムの中で繰り返し現れるような要素に関しては部品として定義して再利用である。図 10 に示した画面左上のメニューの下にある「部品情報」というタブをクリックすることにより部品定義の画面に切り替わる。部品定義の操作は基本的に(1)で述べたシステム構成の定義と同じである。

4.2.3 振舞いの定義

各要素について振舞いを定義することが可能であり，振舞い定義のダイアログボックスに必要な情報を入力することにより設定する。

4.2.4 機能モデルとの関連付け

システムの各要素がどのような機能を果たしているのかというのは重要な情報の一つである。CBModeler では機能モデルを読込んで要素との関連付けを行うことが出来る。図 10 に示した画面左下のツリービューに読込んだ機能モデルが表示される。要素に関連付けられた機能についてはアイコンの色が変わり，関連付けがなされていることを示し，漏れなく関連付けることを助ける。また，ツリービューで機能をクリックすることにより，関連付けられた要素がハイライト表示され，どの要素に関連付けられているのかが直ちに分かるようになっている。

4.3 不具合モデル作成支援ツール

不具合モデル作成支援ツール(以下，FModeler)は 3.3 で述べた不具合オントロジーの定義に則った形で不具合モデルの記述を支援するためのアプリケーションである。FModeler は図 9 の(2)で作成された機能モデルと，(4)で作成された構成・振舞いモデルとを読み込み，ユーザがそれらのモデルで記述された機能やシステムの構成要素を参照する形で不具合モデルの記述することを支援する。

不具合モデルの記述は，FModeler に読込まれた機能モデルあるいは構成・振舞いモデルを「舞台」として事象・状態の設定や，影響伝播などを記述していく。FModeler の画面を

図 11 に示す。この図では構成・振舞いモデル上で事象や影響伝播を記述している。機能モデルや構成・振舞いモデル上で不具合ネットを記述することにより、システムのどの機能／要素でどのような不具合が起こったのかを理解する助けとなる。

3.3 節でも述べているように不具合ネットの各ノードは子ノードを持つことが出来、階層的なネットワークを形成する。FModeler では、表示されている不具合ネットのノードうち、選択された n 個のノード集合 $X=\{x_1, \dots, x_n\}$ を一つのノード x' に置き換え、 X に含まれるノードを全て x' の子ノードにする操作（抽象化）や、あるノードに対して子ノードを設定していく操作（詳細化）等が可能である。

また、不具合ネットは有向グラフであるので、あるノードからあるノードへの到達可能性の検査や最短パスの探索など機能も提供し、ある事象の影響範囲を調べる、二つの事象の因果関係を調べる、などといった用途に用いることが出来る。

4.4 ツールにより記述されたモデル例

本節では、人工衛星を例として CBModeler および FModeler により記述した各モデルについて解説を行う。典型的な人工衛星はロケットにより打ち上げられて地球を周回する軌道に投入され、観測、通信、放送、測位など与えられたミッションをその寿命が尽きるまで果たす。人工衛星のシステムは大きくミッションを達成するのに直接寄与する系（ミッション系）とそれ以外の多くの人工衛星に共通に必要な機能を果たす機器（バス系）とに分けることが出来る。例えば、地球観測衛星では、地球観測が目的であるから地球を観測するためのセンサ、取得したデータを記録するためのデータレコーダ、データを地上に送信するための高速回線用通信機器等がミッション系に含まれる。一方、太陽電池から供給される電力を各機器に分配したり、人工衛星の姿勢を制御したりする為の機器はバス系に含まれる。

図 10 はある人工衛星の構成・振舞いモデルを示している。この図はシステム構成のトップレベルを表しており、人工衛星システム全体に相当する要素 SAT の子要素として各サブシステムが定義されており、環境要素として太陽や地球が定義されている。

次に図 11 に図 10 の構成・振舞いモデルを読み込み、その上で記述した不具合モデルを示す¹。この不具合モデルでは、電源系における「シャントダイオードのバイアス不良（不十分）」という事象に端を発して、「電力分配器出力不足」という事象に至る。その影響が推進系での「推力異常」や、姿勢制御系での「制御異常」という形で現れ、結果として「SAT 太陽捕捉不十分」という事態に至る。通常、人工衛星は十分な電力を確保するために、常に太陽電池パドルが太陽の方向を向くように姿勢を制御している。しかし、十分な推進力が得られなかったり、姿勢制御系に異常が起きたりすることにより、人工衛星が太陽電池パドルを太陽の方向に向けられないと発生電力の低下を招いてしまう。つまり、ここで「太

¹本節で例として取り上げた人工衛星および不具合はあくまでも仮想的なものであり、実際の人工衛星ではこのような不具合が起きないように対策されている。

陽捕捉不十分」という事象は太陽電池パドルでの「発電量不足」に繋がる。この事象は再び電源系での「バス電圧異常低下」という事象を引き起こしている。ここでループを形成していることから、スパイラル的に状況が悪くなっていくことが予想され、どこかで連鎖を断ち切るための対策が必要だということが認識される。

このようにして、CBModeler および FModeler は不具合知識を外在化し共有するための基盤を提供するものである。

5. オントロジーに基づくアプリケーションの枠組み

本研究の目的には、不具合知識の獲得に加えて、知識共有を促すためオントロジーに基づいて動作するアプリケーション間の知識連携の実現が含まれる。本研究では、アプリケーション間でのオントロジーに基づく相互運用を実現する為の枠組みを考案し、これに基づいて各種ツールを実装した。本研究での経験が、オントロジーに基づくアプリケーションを構築する上での一つの知見になると考え、本節では本研究で採用したアプリケーションの枠組みについて述べる。

図 12 に本研究でのアプリケーションの枠組みの概要を示す。ここに示した枠組みは各モジュールの再利用性や保守性を考慮して層構造となっている。アプリケーション間での連携はこの枠組みによって提供される API を介してオントロジーやオントロジーに基づいて記述されたモデルにアクセスすることによって実現される²。以下、各層の実現する機能について述べた後、オントロジー層の機能のうち、汎用データ構造に対する操作機能及びオントロジーマッピング機能について詳しく述べる。

5.1 データ層

本研究ではオントロジー及び一部のモデルの記述に法造オントロジーエディタ[11]を採用している。法造オントロジーエディタで記述されたオントロジー及びモデルのデータは XML 形式で保存される³ため、それらを利用するアプリケーションは XML 形式のデータを読込む機能が必要となる。データ層では XML 形式のデータを操作するための機能を提供する。

5.2 オントロジー層

オントロジー層では概念、スロット、インスタンス等といったオントロジーを定義する要素に対する操作の為の機能を提供する。オントロジー層で提供される機能のうち、以下に挙げる 4 つの主要な機能について簡単に説明する。

²本稿で述べる API は古崎らによる法造 API[10]とは異なるものである。

³法造の最新版(<http://www.hozo.jp/>)では、RDF(S), OWL 形式へのエクスポート機能を実装している。

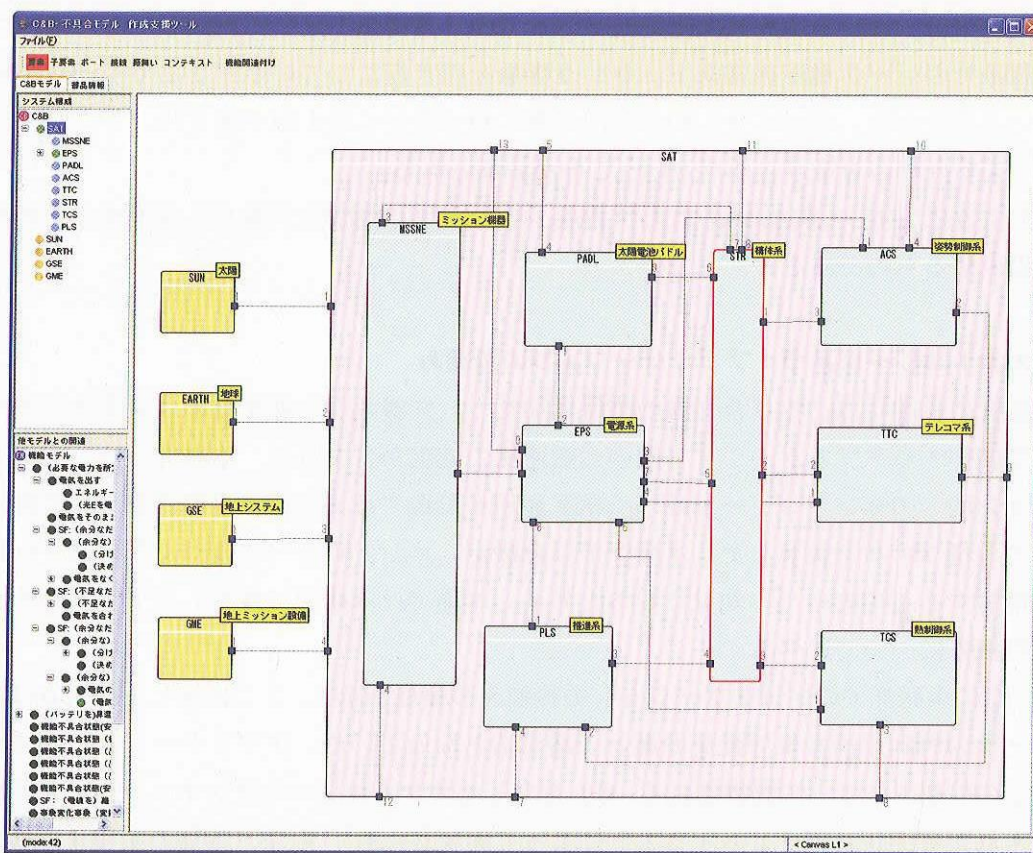


図 10 CBModeler の画面.

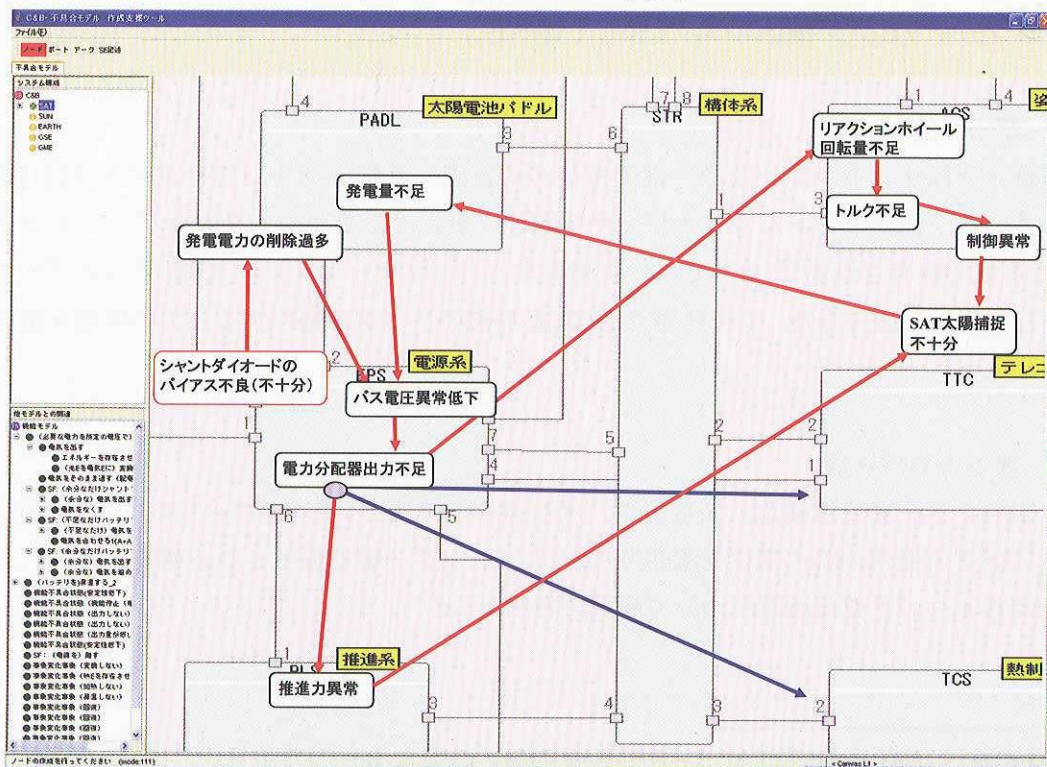


図 11 FModeler の画面.

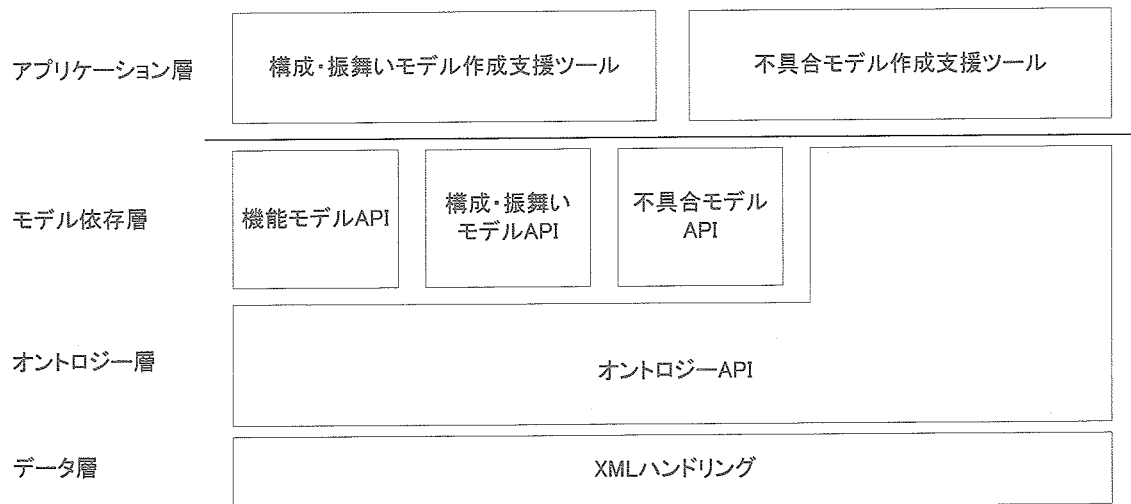


図 12 オントロジーに基づくアプリケーションの枠組み。



図 13 構成・振舞いモデルの例。

5.2.1 オントロジー操作・管理機能

オントロジー管理機能はXMLファイルから読込んだオントロジーデータをオブジェクトとして管理し、オントロジーに対する問合せや概念のインスタンス化等の機能を提供するものである。

5.2.2 インスタンス操作・管理機能

インスタンス管理機能はインスタンスデータをオブジェクトとして管理し、インスタンスに対する問合せ機能や、XMLファイルの読込・保存を提供するものである。

5.2.3 汎用データ構造に対する操作機能

汎用データ構造に対する操作機能はオントロジーの定義に現れる汎用データ構造への操作に対して共通のインタフェースを提供するものである。本機能に関しては5.4で詳しく述べる。

5.2.4 オントロジー

オントロジーマッピング機能はオントロジーの変更に際してアプリケーションが柔軟に

対応できるように設けられた機能である。本機能については5.5 節で詳しく述べる。

5.3 モデル依存層

モデル依存層では抽象モデルを操作する機能を提供する。抽象モデル層の各モデルに対応する形で機能モデル、構成・振舞いモデル、不具合モデル操作用の API が用意されている。

各抽象モデルの基本概念や情報構造はオントロジーで定義されているので、オントロジーAPI を直接利用することにより抽象モデルの構築・操作は可能である。しかし、抽象モデルに対する操作を考える場合、オントロジーの定義上の概念やスロットを直接操作するのは煩雑になるため、モデル依存層の提供する API では抽象モデルの基本概念に対する操作を行うインタフェースを提供している。

図 13 に示すような構成・振舞いモデルの作成過程を例に考えてみる。この図では 2 つの要素、E1 と E2 がそれぞれポートを一つずつ有していて、それらを介してお互いが接続されている。このように「要素」が「ポート」を持つことが出来ること、「要素」同士は「ポート」を介して接続されること等についてはオントロジーで規定されている。一方で、構成・振舞いモデル作成支援ツールのようなアプリケーションでは、ユーザが操作可能な GUI オブジェクトとしての「要素」や「ポート」が実装されており、それらのオブジェクトに対する操作をオントロジーやインスタンスに対する操作に変換する必要がある。そのような抽象モデルに対する操作をインタフェースとしてまとめたのがモデル依存層で提供される API 群である。

5.4 汎用データ構造操作機能

オントロジーやオントロジーの規約に従って構築されたモデルは様々な構造を有するが、具体的な表現方法が異なっても同じ構造のデータとしてみなすことが可能な場合がある。例えば、オントロジーにおける概念の is-a 階層は木構造をなしているが、2 で示される構成・振舞いオントロジーの定義によって構築される構成・振舞いモデルにおいて、「対象系」インスタンスは「子要素（内部特性）」という名前のスロットで子要素であるインスタンスと関係付けられ、これもまた木構造をなす。ユーティリティ機能ではオントロジーやモデルにおける木構造の 4 種類の表現に対応して、木構造に対する一般的な操作に対する統一されたインタフェースを提供する。

木構造データ用機能は対応付けられた処理対象のデータを木構造とみなして操作する。処理対象のデータとして、以下の 4 種のデータを扱うが、将来的にこれ以外のデータを扱えるようにする場合、対応が簡単となるような設計としている。

インスタンススロット型

インスタンスを木構造のノードとみなし、ノード間の親子関係がスロットによって与えら

れるデータ。

インスタンス-関係型

インスタンスを木構造のノードとみなし、ノード間の親子関係が関係概念のインスタンスによって与えられるデータ。

is-a 階層型

オントロジー定義における概念を木構造のノードとみなし、ノード間の親子関係が is-a 関係で与えられるデータ。

概念-スロット型

オントロジー定義における概念を木構造のノードとし、ノード間の親子関係が part-of あるいは attribute-of スロットのクラス制約により与えられるデータ。

木構造の他に不具合モデルにおける不具合ネットに対応した有向グラフに対する操作機能も実装した。

5.5 オントロジーマッピング機能

モデル依存層の API の動作はオントロジーの定義に依存しているため、元のオントロジーの定義が変更されると影響を受ける。あらゆる変更に対応することは不可能であるが、概念の名称の変更や、ある程度の構造の変更等、変更が一定の範囲内ならば対応することは可能である。そのようなオントロジーの軽微な変更に対応するために実装されたのがオントロジーマッピング機能である。

オントロジーに基づくアプリケーションである以上、その上で取り扱われるモデルはオントロジーの定義に則ったものでなければならない。もし、あるオントロジーを前提としてアプリケーションが開発された場合、オントロジーに変更が生じた場合にアプリケーションにも改修が必要となってしまう。しかし多くの場合、アプリケーションの提供する機能はオントロジーの定義に全面的に依存しているわけではなく、アプリケーションで操作対象となるオブジェクトとオントロジーで定義される概念との対応関係さえ把握できれば、オントロジーの変更に際してアプリケーションの改修無しでも（変更後の）オントロジーに則った形で機能を提供し続けることが可能となる。

オントロジーマッピング機能とはアプリケーションで操作対象となるオブジェクトと、オントロジーで定義された概念との対応関係についての情報（マッピング情報）を利用することで、オントロジーの定義に変更があった場合でも、アプリケーションを修正すること無しにマッピング情報の修正だけでの対応を可能とするものである。

6. 評価

知識共有に対する本研究アプローチの有効性を示唆する事項を、これまでの成果物であるモデル、オントロジー、知識共有支援システム（ツール）とその構築過程に対する精察

と分析に基づいて整理した結果を以下に示す。現段階ではツールの完成度が不十分で第3者による試使用などを通じた本格的な評価が困難なため、今回は以下に示す分析をもって評価に代える。

6.1 他の不具合知識表現手段との比較

知識表現手段の例として 1) 自然言語文を中心とするもの（信頼性技術情報など）、2) テーブル形式による表現（不具合情報システムなど）をとりあげ、これらと本研究が提案する表現方式とを比較した。比較の結果を付録の表に示す。なお、図表などの補助情報はこれらのいずれでも添付のかたちで閲覧可能であり、比較の対象から除外した。

自然言語文を中心とした表現の場合、比較的に人間が自由に記述出来る分、表現された知識の品質は書き手に委ねられてしまう。また、表現された知識を計算機で処理しようとする場合には、自然言語処理が必要となり、検索においても精度や再現性に限界がある。

テーブル形式による表現の場合、人間に記述すべき項目を提示することにより、ある程度の品質を確保することが可能になる。但し、各項目の内容については自然言語文を中心とした表現と同様の問題がつきまとう。

FTA や FMEA など、信頼性解析手法に基づく表現の場合は、知識を記述する人間がその手法について知っている必要があるが、手法について熟知しているものであれば、上述の2つよりもより迅速に的確な知識を表現することが可能となる。

オントロジーに基づくアプローチは、オントロジーという形で知識を表現するための概念を明示化してユーザに提供する。明示化のプロセスにはコストが必要となるが、一度明示化され、ユーザの間でその意味について合意が取れれば、その後、知識を表現していく際に、より厳密な意味に基づいた正確な知識の記述が可能となり、知識の利用価値が高いものとなる。

オントロジーに基づくアプローチは上述の表現を置き換えるものではなく、むしろ計算機上で知識を表現するにあたって計算機処理がしやすいような枠組みを与えるものである。本研究では、事象・状態ネットワークという不具合知識の表現形態を提案し、ツールとして実装して、不具合知識が十分に表現可能であることを確認した。今後の課題としては、オントロジーに基づくことにより得られるより厳密な検索能力、知識表現間の相互運用性（モデル変換等）について実証していくことが挙げられる。

6.2 モデルの表現力

モデルの表現力（表現すべき知識を本研究が提案しているモデルがどの程度表現できるか）は重要な評価点である。表現力について現段階では十分な検証を行っていないが、その重要な要因である必要語彙の網羅性（オントロジーがどの程度網羅しているか）については、まず、構成・振舞いオントロジーに関してはほぼ十分と考えられる。また不具合オントロジーに関しても、JAXA 標準文書の調査と十分な検討を経て不具合知識に含まれる基

礎概念の抽出・定義を行っているため、現時点でもその網羅度は高いと考えられる。（具体的な数字については 6.4 項参照）

なお、オントロジーが不十分な場合、現在のオントロジーの追加・変更が必要となるが、これに伴うアプリケーションの変更等の影響については、6.7 節を参照のこと。

6.3 モデルおよびオントロジー上の推論

オントロジーアプローチの有用性の重要な一部は記述されたオントロジーやそのインスタンスが計算機によって理解可能な点にある。これはオントロジーの記述が厳密な意味論にもとづいて記述されることによるが、その有用性の実証は、以下に示すような推論機能の実現を通じて行うことになる。現時点でこれらの機能は実装していない。

6.3.1 モデル独立な推論

概念 A の下位概念は何か、概念 B と同じ親を持つ概念は何かなど、記述されたオントロジーの内容に関する問合わせのうちあるクラスのものは、記述された内容によらず、オントロジー言語の意味論のみに基づいて回答可能である。このような機能はオントロジーで定義された概念の人間による理解やリマインド、次に述べるモデル依存の推論にプリミティブを提供するなど、明らかに知識共有を支援する。

6.3.2 モデル依存の推論

記述したオントロジーやモデルに人が応用上の意味を厳密に与えることにより、モデルに関する有益な推論が可能になる。本研究における記述済みの C&B モデルや SE ネットワークに演繹やアブダクションのアルゴリズムを適用することにより、記述されたモデルが明示的に表していない知識の抽出が可能となる。

6.4 開発されたオントロジーの規模

開発済みのオントロジー含まれる概念（クラス）数は約 1,000 個であり、それらには合計約 400 個の属性が定義されている。また is-a 階層は最も深い場合で 7 である。不具合知識として出現する概念を十分カバーしたとはいえないが、前述のように不具合に関する基礎概念はほぼ網羅されている。また、属性を多く定義しなければならないような「複雑な」概念の定義はほぼ出尽くしていて、残りは is-a 階層だけを規定し属性を特に与えない「語彙体系」的なものが中心と考えられる。但し、これらは多数にのぼり語彙の切り出しのためのツールを要し、また、同義語、類義語、多義語の整理のように応用上重要で且つ専門家の判断を要するものが含まれている。

6.5 オントロジー記述環境の能力

エディタを用いてオントロジーを記述する際、下例のような関係を直接記述できること

が実用上は望ましいと感じられた。但しこれらの幾つかはオントロジーエディタ設計の視点からは、ユーザ定義（それ用の関係概念を別途定義する）で十分であるとの論もある。

- ①同義語 ②類義語 ③多義語 ④類別 ⑤複数の視点からの類別（分類）
- ⑥インスタンスの直接記述 ⑦多重継承

6.6 オントロジー制約

ここで言うオントロジー制約とは、ユーザによるモデルの記述が、オントロジー定義に適合するかたちでなされるように強制（制約）することを言い、下例のような方式によるツールの能力として実現される。

6.6.1 選択オプションの提示

ある概念およびその下位クラスからの選択を強制する方式。is-a 木（選択肢）の表示とそこからのクリック選択などが典型であり、ユーザにとっても便利な方法である。今回の実装でも各種属性の指定などに使用している。

6.6.2 図形オブジェクトの指定・選択の制御

図形オブジェクトを、GUI を通じて指定・記述する機能の制御にオントロジーの定義を適切に反映することにより、これらのオブジェクトに対応する概念の制約強制を実装することになる。今回の実装では、C&B モデルや不具合モデルの GUI を通じた記述の制御メカニズムにより、1 ポートには内部と外部、それぞれ1つだけの接続が許されるなどの制約が強制されている。

6.6.3 入力語彙チェック

入力されたテキストを構成する語彙の一部または全部がオントロジーで定義されているかどうかをシステムがチェックする方式。この場合、オントロジー制約の強度（受け取った語彙のうちどの範囲をチェックの対象にするか）が重要な考慮点となる。制約強度を強めることは一般に、より深い機械推論を可能にするなど、オントロジー研究の立場からは魅力的な課題であるが、オントロジーのカバレッジが十分でない段階で制約強度を強めると書くべきことが書けないという問題が生じる。

現時点のツール（知識共有支援システム）は、オントロジー制約が比較的緩やかになるよう実装されている。今後は、同義語や類義語の定義を含めオントロジーのカバレッジを広げながら制約を増強するというアプローチが妥当と考えられる。

6.7 オントロジー変更のアプリケーションへの影響

オントロジーの網羅性に関りを持つ重要な論点として、オントロジーに追加・変更がなされたときアプリケーションをどの程度変更しなければならないか、すなわち、オントロ

ジー変更のアプリケーションへのインパクトが挙げられる。

ドメインを限定してもこれに関して人が持つ知識は一般に膨大である。その中から限られた時間内に記述すべきものを取捨選択し記述することになるが、オントロジーは時間の経過とともに追加・変更されるべきものの前提に立つほうが現実的である。一般論としてオントロジーの「全体性の中の部分性」という特性により、オントロジー自体の追加・変更は比較的容易であると言われている。

一方、ある時点で記述されているオントロジーに基づいて開発されるアプリケーションソフトの変更についてはそのインパクトをどう吸収するかが問題となり、「オントロジーマッピング」などオントロジーの研究課題となっている。

本システムでもオントロジーマッピングの考え方を反映したインパクト吸収のための機能を実装している。

6.8 知識共有支援システム（ツール）の能力

冒頭でも述べたように、実装した知識共有支援システム（ツール）の機能は限られており、現時点では試使用を通じた知識共有の実証を行う段階にはない。以下、これまでに認識されている事項や知見について述べる。

6.8.1 パフォーマンス

作成した C&B モデルの読み込み時に、XML ファイルの処理でメモリをかなりの量（500MB 程度）消費すること、かなりの時間（数分）を要することが判明している。これらは今後改善する必要がある。

6.8.2 システムフレームワーク

今回試作したシステムは、1) 知識共有支援を目的とし、2) 共有対象（ドメイン）が不具合知識であり、また、3) システムが扱う情報の構造等がオントロジーによって供給されるという特徴を持つ。このことから自然に導かれる本システムのフレームワークを図 14 に示す。知識共有を支援するアプリケーション機能は多様であるためそれらの「脱着性」を促進するプラットフォーム化が実用化のためには望ましいが、これはフレームワークに沿った今後の機能追加によって達成可能である。

6.8.3 アプリケーション機能

実装済みのアプリケーション機能と今後の実装が望ましい機能を図 15 に示す。図中⇒印のものが実装済みである。また「共通機能」部は、多くが「知識共有支援」というシステムの目的による特徴を反映しているが、KBMS 部（フレームワーク参照）に属す機能の支援を受けて実装されると想定される。

6.9 評価のまとめ

オントロジーの概念定義能力およびその計算機理解可能性を活用する本研究のアプローチは、JAXA における不具合知識の共有に有効である可能性が高いと判断する。但し、その有効性の実証にはオントロジーのさらなる充実と支援システム機能の強化が必要である。また、この支援システムの機能強化では、オントロジーに基づく知識ベースの管理機能とこれを利用したアプリケーションの充実を図るべきである。

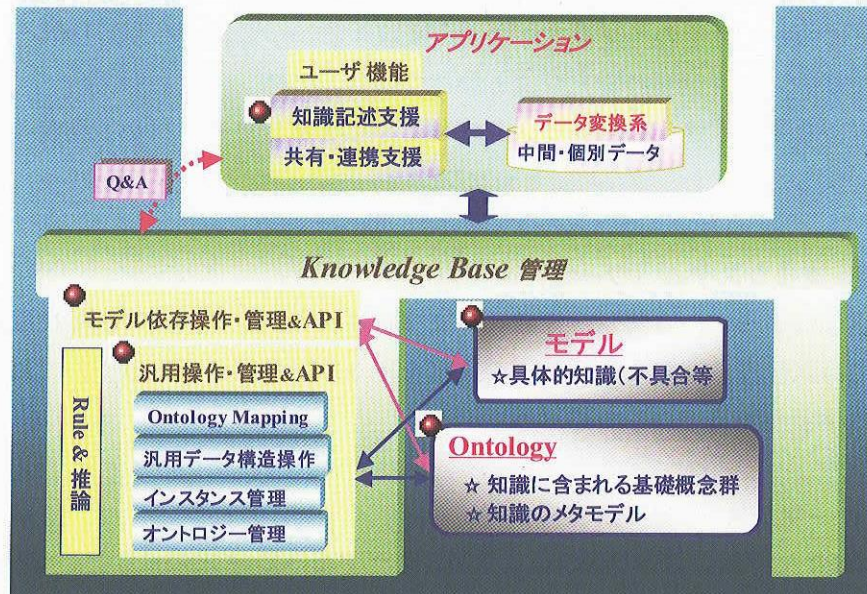


図 14 知識共有システムのフレームワーク

■ 共通機能

- 不具合モデル(SEネット)の記述・表示
- 構成・振舞いモデルの記述・表示
- 故障モード定義・参照・利用
- 部品定義・参照・利用
- SE ネット上の推論機能 未実装
- 構成・振舞いモデル上の推論機能
- 検索機能(同義語、多義語、類義語対応)
- 粒度(抽象度)制御(概念の上下、モデルの親子)
- 概念定義参照・Q&A機能

■ 個別機能(未実装)

- | | |
|---------------------|--------------|
| ■ FMEA支援関連 | ■ FTA支援関連 |
| □ 機能重要度設定・参照 | □ ハザード設定・参照 |
| ■ FMEA チャート生成 | □ ハザードレポート生成 |
| ■ FMEA/ FTA クロスチェック | |

図 15 知識共有システムのアプリケーション機能

7. 関連研究

7.1 不具合知識の獲得・共有・利用に関する研究

FTA や FMEA に代表される信頼性解析手法[15]においては、不具合や故障に関する様々なモデルが用いられている。しかし、それらの手法は解析の枠組みを与えるものの、解析モデルにおける「故障」や「不具合」の概念についての明確な規定が与えられておらず、文献によってその定義がゆれることを Kmenta らも指摘している[5]。そのことに起因して、考慮されるべき不具合が網羅されずに見落とされがちであった。本研究では故障オントロジー[16]の考え方を拡張した不具合オントロジーを開発し、不具合とは何かを明示化することにより、この問題に対処している。

FMEA の欠点を改善するために、Kmenta らは Advanced FMEA(AFMEA)という手法を提案している[5]。AFMEA では機能、状態変数、物理実体を関連付けるような振舞いモデル(behavior model)を導入する。振舞いモデルの中では不具合を明示的に定義することができ、その原因や影響について評価するための枠組みを与える。しかし、AFMEA で不具合は機能の定義の一部である事後条件の否定という形で与えられるため、表現できる不具合が限られている。

Hata らは人工物の機能について、構成要素と、その間を流れる機能の流れ(functional stream)によるネットワーク構造として表現し、その構造を解析することにより故障モードを同定する手法を提案している[2]。しかし、Hata らの手法では機能喪失を不具合とみなしており、本研究で対象とする不具合の一部しか表現されない。

モデルベース診断の分野では製品の振舞いモデルに基づいた故障推論が行われている[1]。モデルベース診断では故障を「予測された正常な振舞いと観測された異常な値の不一致」として定義しており、論理的にはありえるが、物理的にはありえない故障を推論する可能性がある。信頼性の確保という観点からすると、正常の振舞いからの偏差という形で不具合を陰に扱うだけでは不十分であり、不具合モードを類型化・認識して対策を立てることが重要である。

7.2 オントロジー・マッピング機能

Heflin らはオントロジー記述言語 SHOE について、一階述語論理への写像により意味論を与え、オントロジーから導出される理論の論理的帰結を比較することにより、オントロジーの変更について論じている[3]。Heflin らはインスタンスについては所与として扱っており、オントロジーに則ったモデルの構築方法には言及しておらず、本研究におけるマッピング機能の扱う問題の対象が異なる。

Stojanovic らはオントロジーの変更について、再構成設計問題と捉えて、要求が与えられると要求を満たすような変更を自動的に求める方法について提案している[8]。Stojanovic らはオントロジーの保守を直接の対象としており、オントロジーに基づいて記述されたモデル

については言及していない。

Noy らはオントロジーの変更管理について、異なる二つのバージョンのオントロジーについて、ヒューリスティックスを用いつつ変更箇所を自動的に検出し、更に二つのオントロジーの間の対応関係を求める方法について提案している[6]。この方法を適用により変更前と変更後のオントロジー間の変更箇所を自動検出し、マッピング情報を自動的に生成・利用できる可能性がある。

Klein らは、オントロジーのバージョン管理の考え方について論じるとともに、二つのオントロジーを構造的に比較する手法について述べている[4]。オントロジーの構造比較の手法については、オントロジーのバージョン間のマッピングを記述するのに労力を軽減するのに役立つことが予想される。

Oliver らは医学分野における専門用語の分類をオントロジーにとらえて、オントロジーの変更について医学分野で用いられているいくつかの専門用語分類体系といくつかの知識表現言語について比較を行い、新しい表現体系を提案している[7]。Oliver らは専門用語というカテゴリー的な知識を対象としており、インスタンスについては取り扱わないと宣言している。

砂川らはオントロジーの分散開発におけるオントロジーの変更の問題に関して、オントロジーのモジュール間の依存関係を管理し、オントロジー全体として整合性を保つ為の仕組みについて提案している[9][13]。砂川らの手法はオントロジーの整合性を保つことを目的としているのに対して、本研究でのオントロジーマッピング機能はオントロジーに基づくアプリケーションに対して、データ構造を規定するオントロジーとアプリケーションが処理の上で想定するオントロジーとの対応関係を与えるものである。これらは相補的なものであると考える。

8. おわりに

本稿では、オントロジー工学を応用した不具合知識獲得の枠組みについて述べた。高い安全性・信頼性が求められる宇宙開発分野において不具合知識を共有し、宇宙システムの開発・運用を着実に進めることは非常に重要なテーマである。本稿では知識獲得までのシナリオを提示したが、その後には当然共有のシナリオが来るべきである。獲得された知識の効果的な共有・再利用の実現については今後の課題となる。

また、本研究の中で各種ツールを開発する過程で考案したオントロジーに基づくアプリケーション開発の枠組みについて紹介した。オントロジー工学の目指す知識の共有・再利用を実現する上で、オントロジーに基づくアプリケーションの開発は必要となってくるが、オントロジーのバージョン管理など解決されなければならない問題はまだ存在する。オントロジーによって表現された知識を効果的に活用するためにはオントロジーに基づくアプリケーションを開発するための基盤を整備し、広く普及させることが肝要である。オント

ロジー・アウェア (ontology-aware) なアプリケーションが増え、知識レベルでのアプリケーションの連携が一般的になることを期待する。

謝辞

本研究の一部は大阪大学産業科学研究所溝口研究室との共同研究により実施された。また、研究を進めるに当たってオントロジー工学の視点より広く助言を頂いた。ここに、溝口研究室の溝口理一郎教授、來村徳信助教授、小路悠介氏に対して深い感謝の意を表するものである。

人工衛星電源系サブシステムのモデルを記述するに当たって、宇宙利用推進本部 OICETS プロジェクトチーム、および総合技術研究本部エレクトロニクス技術グループに協力を頂いた。

参考文献

- [1] De Kleer, J. and Williams, B. C.: Diagnosing Multiple Faults, *Artificial Intelligence*, Vol. 24, pp. 347—410 (1987)
- [2] Hata, T., Kobayashi, N., Kimura, F., and Suzuki, H.: Representation of Functional Relations among Parts and its Application to Product Failure Reasoning, in *Proceedings of the 2000 International CIRP Design Seminar on Design with Manufacturing: Intelligent Design Concepts, Methods and Algorithms* (2000)
- [3] Heflin, J. and Hendler, J.: Dynamic Ontologies on the Web, in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 443—449 (2000)
- [4] Klein, M., Fensel, D., Kiryakov, A., and Ognyanov, D.: Ontology Versioning and Change Detection on the Web, in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, Vol. 2473 of *Lecture Notes In Computer Science*, pp. 197—212 (2002)
- [5] Kmenta, S., Fitch, P., and Ishii, K.: Advanced Failure Modes and Effects Analysis of Complex Processes, in *Proceedings of the ASME Design for Manufacturing Conference*, DETC99/DFM-8939 (1999)
- [6] Noy, N. F. and Musen, M. A.: Ontology Versioning as an Element of an Ontology-Management Framework, *IEEE Intelligent Systems*, Vol. 19, No. 4, pp. 6—13 (2004)
- [7] Oliver, D. E., Shahar, Y., Shortliffe, E. H., and Musen, M. A.: Representation of Change in Controlled Medical Terminologies, *Artificial Intelligence in Medicine*, Vol. 15, No. 1, pp. 53—76 (1999)
- [8] Stojanovic, L., Maedche, A., Stojanovic, N., and Studer, R.: Ontology Evolution as Reconfiguration-Design Problem Solving, in *Proceedings of the Second International Conference on Knowledge Capture (K-CAP'03)*, pp. 162—171 (2003)

- [9] Sunagawa, E., Kozaki, K., Kitamura, Y., and Mizoguchi, R.: An Environment for Distributed Ontology Development Based on Dependency Management, in *Proceedings of the Second International Semantic Web Conference (ISWC2003)*, pp. 453—468, Sanibel Island, FL, USA (2003)
- [10] 古崎 晃司, 來村 徳信, 佐野 年伸, 本松 慎一郎, 石川 誠一, 溝口 理一郎: オントロジー構築・利用環境「法造」の開発と利用—実規模プラントのオントロジーを例として, 人工知能学会論文誌, Vol. 17, No. 4, pp. 404—419 (2002)
- [11] 古崎 晃司, 來村 徳信, 池田 満, 溝口 理一郎: 「ロール」および「関係」に関する基礎的考察に基づくオントロジー記述環境の開発, 人工知能学会論文誌, Vol. 17, No. 3, pp. 196—208 (2002)
- [12] 溝口 理一郎, 池田 満: オントロジー工学序説—内容指向研究の基盤技術と理論の確立を目指して—, 人工知能学会誌, Vol. 12, No. 4, pp. 559—569 (1997)
- [13] 砂川 英一, 古崎 晃司, 來村 徳信, 溝口 理一郎: 「法造」におけるオントロジー分散開発, 人工知能学会第 3 回セマンティックウェブとオントロジー研究会研究会資料, SIG-SWO-A301-02 (2003)
- [14] 小路 悠介, 來村 徳信, 溝口 理一郎: 機能モデルにおける補助機能の分類とその設計意図の明示化について, 第 17 回人工知能学会全国大会, 1E1-05 (2003)
- [15] 鈴木 順二郎, 牧野 鉄治, 石坂 茂樹: FMEA・FTA 実施法, 日科技連出版社 (1982)
- [16] 來村 徳信, 溝口 理一郎: 故障オントロジー: 概念抽出とその組織化, 人工知能学会誌, Vol. 14, No. 5, pp. 68—77 (1999)
- [17] 來村 徳信, 笠井 俊信, 吉川 真理子, 高橋 賢, 古崎 晃司, 溝口 理一郎: オントロジーに基づく機能的知識の体系的記述とその機能構造設計支援における利用, 人工知能学会論文誌, Vol. 17, No. 1, pp. 73—84 (2002)
- [18] 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)
- [19] 來村 徳信, 小路 悠介, 高橋 知伸, 吉川 真理子, 柏瀬 雅一, 布瀬 雅義, 溝口 理一郎: 機能的設計知識記述・共有の枠組みとその実用展開, 第 17 回人工知能学会全国大会, 1E1-04 (2003)
- [20] 來村 徳信: 工学ドメインオントロジー, 人工知能学会誌, Vol. 19, No. 2, pp. 179—186 (2004)
- [21] 大阪大学, 宇宙航空研究開発機構: 2004 年度「衛星設計・運用支援へのオントロジー技術適用の研究」成果報告書 (2005)

付録 「評価・分析」－知識共有の視点－

知識獲得		文章	不具合情報システム	FMEA	FTA	オントロジーに基づく知識共有の枠組み
定義への適合性保証		書き手に委ねられる。	項目を選択させることにより、入力時に制約を与えることが可能。	書き手に委ねられる。	書き手に委ねられる。	知識記述に用いられる概念はオントロジーで定義されており、定義に沿った正確な知識の記述が可能。
記述・獲得の容易性 (コスト)		誰でも書くことが出来る。コストは記述の内容に依存する。	いくつかの項目については予め定義された選択肢から選ぶことができ、入力力が簡略化されている。	解析手法に従って記述すること、で、比較的容易に記述できる。	解析手法に従って記述すること、で、比較的容易に記述できる。	形式に則ってモデルを記述しなればならぬためコストは高い、ツールの支援により負担の軽減は可能。
記述性 (Expressiveness)		言葉で表現できるものなら何でも記述可能である。但し、論理的な関係などを厳密に書くのには向いていない。	不具合情報システムが想定している範囲で知識を記述することが可能。	FMEA 解析が対象とする項目 (品目、故障モード、影響等) については記述可能。	FTA 木で表現される事象間の因果関係で表現できるものは記述可能。	枠組みで提供するモデルで表現可能なものについては記述可能。記述不可の事柄についてもオントロジーの修正や拡張により対応できる可能性がある。
知識源の範囲						
知識源となる人		文章が書ける人	不具合情報システムに情報の登録が許されている人	解析手法を知っている人	解析手法を知っている人	オントロジーに合意が出来て、知識の記述方法を理解した人。
知識源の媒体		紙、電子ファイル等	データベース	FMEA シート	FTA 木	オントロジーに基づけられた各種モデル
知識の内容		言葉で表現できるもの	不具合情報システムが対象とする知識	FMEA 解析で取り扱われる知識 (システムの構成、機能、不具合等に関する知識)	FTA 解析で取り扱われる知識 (システムで発生しうる事象や振舞い、およびそれらの間の因果関係に関する知識)	開発対象システムの機能、構成、振舞い、不具合に関する知識。
記述内容の品質		書き手により左右される。	テンプレートによりある程度品質を保つことが可能。	FMEA シートに沿って記述することによりある程度品質を保つことが可能。	FTA 木という形式で記述することにより、ある程度品質を保つことが可能。	形式的なモデルのために、品質は保証される。ツールによる一貫性チェック機能により品質向上に貢献可能。
知識の共有・流通						
概念の正確な定義		用語集などにより概念を定義することが可能。	テンプレートの項目や、各種分類の項目について予め定義されている。	シートに現れる概念 (故障モード、推定原因) について厳密な定義はされていない。	事象の論理関係については定義されるが、事象が何であるか、その内容については定義を与えない。	知識の記述に用いられる概念はオントロジーにより明示的に定義されている。
記録 (知識) へのアクセス性		電子媒体であれば、全文検索等が可能。	属性やキーワードによる検索が可能。	電子媒体に記述されているキーワード検索、属性検索が可能。	電子媒体に記述されているキーワード検索が可能。	形式的なモデルで記述されているため、オントロジーに対する汎用的な問い合わせ技術を利用可能。
理解性						
意図、文脈、前提知識の捕捉		書き手が明示する必要あり。	不具合情報システムの提供するテンプレートに含まれている。	解析手法が前提となっており、書き手と読み手で共有されていることが前提。	解析手法が前提となっており、書き手と読み手で共有されていることが前提。	オントロジーに合意すること、で、共有されることを前提としている。
粒度 (抽象度) の選択・制御		特に制御、制約は無い。	テンプレートの提供する項目に制約される。	品目や故障モードの選び方により制御可能。	特に制御、制約はない。FTA 木の深さに合わせて制御することは可能。	粒度を制御するための階層的な表現が可能となっている。

宇宙航空研究開発機構研究開発報告 JAXA-RR-05-002

発行日 2005年9月30日
編集・発行 独立行政法人 宇宙航空研究開発機構
〒182-8522
東京都調布市深大寺東町七丁目4番地1
TEL 0422-40-3000 (代表)
印刷所 株式会社 ビー・シー・シー・
東京都港区浜松町2-4-1

©2005 JAXA

※本書(誌)の一部または全部を著作権法の定める範囲を超え、無断で複写、複製、転載、テープ化およびファイル化することを禁じます。

※本書(誌)からの複写、転載等を希望される場合は、下記にご連絡ください。

※本書(誌)中、本文については再生紙を使用しております。

<本資料に関するお問い合わせ先>

独立行政法人 宇宙航空研究開発機構 情報化推進部 宇宙航空文献資料センター



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