

タスク・ドメインロールに基づくオントロジー構築ガイドシステムの設計と開発

—石油精製プラントを例として—

Design and Development of a Guide System for Building an Ontology based on Task/Domain Role Concepts

—A Case Study on an Oil Refinery Plant—

石川 誠一

Seiichi ISHIKAWA

大阪大学産業科学研究所^{†1}

I.S.I.R., Osaka University

久保 成毅

Shigeki KUBO

(同 上) ^{†2}

古崎 晃司

Kouji KOZAKI

(同 上)

kozaki@ei.sanken.osaka-u.ac.jp, <http://www.ei.sanken.osaka-u.ac.jp/~kozaki/>

来村 徳信

Yoshinobu KITAMURA

(同 上)

kita@ei.sanken.osaka-u.ac.jp, <http://www.ei.sanken.osaka-u.ac.jp/~kita/>

溝口 理一郎

Riichiro MIZOGUCHI

(同 上)

miz@ei.sanken.osaka-u.ac.jp, <http://www.ei.sanken.osaka-u.ac.jp/members/miz/>

keywords: ontology, building ontologies, role, methodology

Summary

Recently, ontology is expected to contribute to sharing and reuse of knowledge, and a lot of research on ontology has been carried out. However, useful methodologies for building ontologies are not yet established in spite of the understanding of its necessity. The crucial point of building ontologies is to articulate concepts in the target field. If a concept is defined in a manner depending on a particular context, it is difficult to reuse it in other contexts. Therefore, it is important to distinguish context-dependent concepts from context-independent ones. The difficulty of the discrimination requires development of a methodology and a guide system for building ontologies. This paper proposes AFM (Activity-First Method), a methodology of building domain ontology depending on task analysis, and an ontology building guide system based on AFM. The guide system supports building ontologies by providing an ontology author with building steps, and managing intermediate results. In the first half of this paper, we take an ontology of oil refinery plant operation task as an example and discuss articulation of the role concepts. Role concepts represent conceptual categories of roles of matters in a particular context. This consideration serves as the foundation of this guide system. In the second half of this paper, we show how the guide system supports an ontology building process.

1. ま え が き

オントロジーは対象世界の「何を表現すべきか」という知識の内容の理解に貢献し、知識の共有・再利用性を高める [溝口 94]。この為、近年様々な研究が盛んになされている。オントロジーとは元来、哲学の分野で「存在に対する体系的な理論」を意味するが、知識処理の分野では、知識システムを構築する際に用いられる基本概念の体系的記述をいう。オントロジーは、その対象となっ

ている知識やシステムにおける概念の定義とそれらの間に成り立つ関係の定義から構成される。従って、オントロジーには、システム構築者の設計意図やそのシステムが持つ規約といったものが反映される。そのため、オントロジーに対する合意が形成できれば知識やシステムの再利用が容易となる。このようにして知識の内容そのものに関する問題を解消し、知識の共有・再利用に大きく貢献する。しかしながら、オントロジーを構築するための有用な一般的方法論などはまだ確立されているとは言えない。そのため、構築方法論の確立やそれに基づく構築支援環境の開発が望まれている。

^{†1} 現在、NTTコムウェア

^{†2} 現在、NTT 西日本

このような背景のもと筆者らは Activity-First Method (AFM) [Mizoguchi 95, Takaoka 96] というオントロジー構築方法を提唱した。しかしながら AFM に基づいた計算機支援システムは論文 [Mizoguchi 95, Takaoka 96] では開発されていない。そこで、筆者らは、AFM に基づいたオントロジー構築ガイドシステム概念工房の設計・開発を進めてきた。

本論文では AFM に基づいたオントロジー構築ガイドシステム概念工房の設計・開発について述べる。概念工房は、構築ステップを詳細に規定することでタスクオントロジー、ドメインオントロジーの構築過程を支援する。また、もう一つの特徴として本システムはロール概念に関する基礎的考察結果が反映されており、基本概念とロール概念との峻別を示唆する機能を持つ。

本論文ではまず、オントロジー構築方法に関する筆者らの考えを述べ、オントロジー構築過程においてロール概念を考慮しない場合の問題点を示し、本システムの基盤となるタスクロール・ドメインロールに関する基礎的考察を行う。続いてその考察に基づいて設計・開発した概念工房におけるオントロジー構築過程の支援機能について述べる。本研究では「石油精製プラントの復旧運転」を題材とし、この問題解決知識に関するオントロジー構築過程で AFM を適用し、ガイドシステムのプロトタイプを実装した。

本論文は、7つの章から構成され、2章では本研究の基本的な考え方について述べる。3章では、石油精製プラントの復旧運転問題を例にしてロール概念の基礎的考察を述べる。続いて4章で、構築方法 AFM に基づいたオントロジー構築ガイドシステム概念工房によるオントロジー構築の流れと支援機能を概説し、5章で概念工房の利用例を述べる。次に6章で他の関連研究との比較を行い、7章で全体の総括と今後の検討課題を述べる。

2. 概念工房の設計方針

汎用性のあるオントロジーを構築するには、様々な問題があり、データベースの立場からは情報の非均質 (Data Conflicts) などの問題が指摘されている [Goh 97]。オントロジー構築におけるこのような問題の中でも知識ベースの立場ではタスクとドメインの分離に関する問題、ロール概念に関する問題が大きな問題として議論されており、本研究はこれらの問題を主な解決対象とする。

オントロジー構築のポイントは、複雑な対象世界からの概念の切り出しであり、問題解決に必要な概念をオントロジーにいかに適切に定義するかが重要となる。本研究では、専門家へのインタビュー結果や運転マニュアルなどの自然言語で記述されたドキュメントなどの情報源から概念を抽出することでボトムアップにオントロジーを構築することを想定している。しかしながら、ドキュメントに記述されている語彙には表記揺れなどが多く見

られ、一貫性が欠如していることが多い。そのため、情報源から抽出した概念も一貫性がなくなりやすい。特に、3・1節で述べるように、情報源には状況に依存しない概念 (基本概念) と状況に依存する概念 (ロール概念) が混在して記述されている。ある状況 (コンテキスト) に特化した概念を定義すると、他の状況での再利用が困難になる。そのため、基本概念とロール概念を区別して扱い、タスクや目的依存性を除去した汎用性のある概念構造を構築することが重要である [溝口 99]。しかしながら、基本概念とロール概念を峻別し、一貫性を保ちながら概念を切り出して汎用性のある概念構造を構築することは容易ではない。また、状況独立とはいえ、不必要に広範囲で詳細な概念構造を構築することは望ましくない。これらのことを考慮しながらオントロジーを構築する必要があるため、その作業過程は煩雑かつ不明確になりやすく試行錯誤を伴う。そのため、何らかの作業指針や手順がオントロジー構築者 (オーサ) に与えられなければ、構築作業を一定の手順で進めていくことが困難となり、その結果、必要な概念の定義の欠落や、逆に不必要な概念の過剰な定義などの問題が生じやすくなる。そこで、このようなオーサの負担を軽減するための構築方法論や構築支援環境が必要と考えられる。

筆者らが提唱する AFM はこのようなオントロジー構築時の困難さを軽減することを目指して開発された構築方法である。

AFM の基本となる考え方は、筆者らが提唱してきたタスクオントロジーを用いた、タスクに依存する部分とタスクから独立した部分との切り分けである。オントロジーは問題解決 (タスク) の手順などに関するタスク知識を支えるタスクオントロジーと問題解決の対象 (ドメイン) に関するドメイン知識を支えるドメインオントロジーとに分けて記述される [溝口 94]。AFM の Activity はタスクの処理を表す概念 “タスクアクティビティ” を意味し、タスクオントロジーの中核をなす。つまり AFM とは「タスク分析を行い、タスクアクティビティを最初に抽出・整理し、それらに参照されるドメイン概念を整理していく構築方法」である。

多くの技術資料には何らかのタスクから見た対象世界に関する情報が記述されている。AFM はまずタスクの構造を抽出し、タスクから見たドメイン概念をタスクからある程度切り離して記述することで、タスクから見て必要な範囲の概念をドメイン知識として抽出することを可能にしている。そして、このドメイン知識とタスク知識との関連は、タスクコンテキストロールという概念を導入することによって保持している。

本論文で述べるオントロジー構築ガイドシステム概念工房では構築ステップを規定し、また構築過程における中間生成物を管理することで試行錯誤が必要となる構築作業を支援する。本システムは構築ステップを提供することで各ステップにおけるオーサの行うべき作業を限定

し、オーサの負担を軽減することができると考えられる。

本システムは、筆者らの所属する研究室で開発が進められているオントロジー構築・利用環境「法造」の一部として開発された [古崎 02a]。「法造」は概念工房とオントロジーエディタ、オントロジーサーバーから構成される。「法造」において概念工房は、オントロジー構築手順の提供や、オントロジーに定義すべき概念を切り出す際に注意すべき観点の提供を行うことで、オントロジー構築の上流工程の支援を行うシステムである。一方、オントロジーエディタは概念の詳細な意味定義の記述や整合性の検証など、オントロジー構築における下流工程を支援するシステムである。オントロジーエディタと概念工房はシステムの結合されており、オーサは概念工房で構築したオントロジーをオントロジーエディタで読み込むことで詳細な意味定義を記述することができる。

3. ロール概念に関する考察

3.1 オントロジー構築時における問題点

前章では、オントロジー構築方法に関する筆者らの基本的な考え方について述べた。本節では、まずコンテキストに応じて対象領域における概念の認識が変化する例を紹介し、それを考慮せずにオントロジーを構築した際に起こる問題点について述べる。例えば、ある高校に勤務している男性は、高校では「教師」、家庭では「父親」、夫婦間では「夫」というようにコンテキストの変化に応じて別の概念として認識される。しかし、彼はどのようなコンテキストにおいても「男性」であり、「人間」であることには変わらない。同様の例として自転車の「車輪」が「前輪」、「操舵輪」、「駆動輪」とコンテキストによって異なる概念として認識されるなどもある [古崎 02a]。このようにコンテキストに応じて概念の認識の変化が起こることは知識システムの対象世界に一般的に見られることである [Guarino 98]。

次に本研究で例題とした石油精製プラントの復旧運転タスクにおける例について述べる。例えば (図 1)、ポンプがドラムからナフサを抜き出す流量を制御するために「流量コントローラ (FC29)」に操作される制御弁は「流量制御弁」と呼ばれる。しかしこの制御弁は、ドラムの液レベルを制御するために「液レベルコントローラ (LC29)」に操作されると「液レベル制御弁」と呼ばれる。そのためオーサは「制御弁」の下位概念として機能の対象に関して定義が詳細化されている「流量制御弁」、「液レベル制御弁」を定義しがちである。ここで注意すべきことは、「流量制御弁」と「液レベル制御弁」は別の名前で呼ばれているが、図 1 のように液レベルコントローラ (LC29) と流量コントローラ (FC29) との 2 つのコントローラによって制御されている場合 (カスケード制御と呼ばれる) には、同一の部品であることである。液レベルは、ナフサの流量が変化した結果として変化する。そ

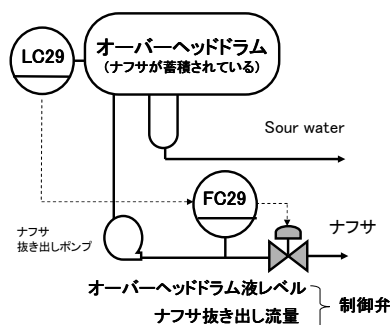


図 1 名前が変わるドメイン概念の例

のため液レベル制御弁はこのようなカスケード制御構造というコンテキストが決定した後、そのコンテキストに依存して定義される概念であるのに対して、流量制御弁はコンテキストに依存せずに定義することができる概念である。上記の例ではコンテキストに依存する概念「液レベル制御弁」とコンテキストに依存しない概念「流量制御弁」という等質でない概念が同じ is-a 階層に混在して記述されている。このように同じ is-a 階層に等質でない概念を混在させると、オーサは物理的に 2 つである流量制御弁に対して「流量制御弁」、「液レベル制御弁」という異なるインスタンスを 2 つ作ってしまうという問題が考えられる。

また同様の例として例えば石油精製プラントにおいてナフサは、その存在する場所に注目している場合は「塔頂成分」、その化学的性質に注目している場合は「軽沸成分」、蒸留塔において蒸留されたという、受けた機能について注目している場合は「留分」と呼ばれる。

このような問題が起こる原因はオーサが「人間」、「ナフサ」のような状況に依存せずに定義できる概念 (基本概念) と「教師」、「塔頂成分」のような状況に依存して定義される概念 (ロール概念) を混同しているためである。このような基本概念とロール概念を担った概念の混同による問題も論文 [古崎 02a, Guarino 98] で指摘されており、論文 [古崎 02a] では、ロール概念の定義内容などに関しての一般的な考察が報告されている。本研究の目標は、論文 [古崎 02a] における考察を踏まえて、ロール概念をタスクロール、ドメインロールという二つの概念に詳細化し、これらを決定する際に必要な概念を分類することで基本概念とロール概念の混同を避けるための指針を得ることである。この考察が概念工房の基盤となる。

3.2 ロール概念

ロール概念 [溝口 99] とは、あるものが特定のコンテキスト下でどのような「役割」を担うかを表す概念であり、対象への視点を反映する。ロール概念には概念間の関係や全体に依存するもの、タスク (問題解決知識) やドメイン (問題解決対象) に依存するものなどが知られている。例えば「徴候」や「原因」といったタスクの処理や

タスクのコンテキストに依存して決まる「役割」を表す概念がタスクロールであり、ドメイン概念がドメインの状況に応じて担う「役割」を表す概念がドメインロールである。次節から、このタスクロールとドメインロールについて基礎的考察を行う。

3.3 タスクロールの分類

タスクオントロジーを構成する主要な概念にタスクアクティビティとタスクロールがある。例えば石油精製プラントの復旧運転では、プラントの振る舞いを監視し、異常があればその原因を診断し、異常を復旧できるように装置を操作することで遂行される。従って、「(復旧方策を)決定する」、「実施する」といった問題解決の操作や処理を表す概念がタスクアクティビティである。タスクアクティビティはより粒度(グレインサイズ)の小さなタスクアクティビティの組み合わせで達成される(図2網かけ部分)。このようなアクティビティ間の達成関係を“is-achieved-by”関係と呼ぶ。このような視点からタスクアクティビティを整理したものが is-achieved-by 階層であり、タスクの構造を明示する。図2は is-achieved-by 階層の例を表している。本稿では、あるグレインサイズにおけるアクティビティの系列を1つの階層とみなし、図2に示す is-achieved-by 階層を左から第1階層、第2階層と呼ぶ。例えば先ほどの「予測する」の存在する階層は第4階層である。

タスクアクティビティは一般に処理対象や処理結果という入出力を持つ。この入出力となる対象の実体はドメインの対象物やその属性などであるが、問題解決の中で用いられる際には、問題解決過程での役割が与えられる。この役割を表す概念がタスクロールである。

タスクロールにはタスクコンテキストロールとアクティビティロールがある。タスクコンテキストロールはタスク全体の中での役割であるのに対し、アクティビティロールは実行されているタスクアクティビティから決まる役割である。図2の第3階層に「(異常から)原因推定する」という処理がある。この入力のアクティビティロールは「原因推定する」のみから決まり単に「推定対象」と言われるが、復旧運転タスクというタスクコンテキストにおいては「異常」という役割を持つので、タスクコンテキストロールは「異常」となる。このようにアクティビティロールはタスクコンテキストへの依存性が低く、タスクコンテキストから独立した形で定義できることに對して、タスクコンテキストロールはタスクのコンテキストと強く結合したロール概念である。

このように、タスクコンテキストロールは is-achieved-by 階層の上位-下位関係や順序関係などで決まり、タスクアクティビティの目的(タスクコンテキスト)に依存して定義される。そのためタスクコンテキストロールは、タスクアクティビティとの関連性から分類できる。本節では、is-achieved-by 階層の上位、下位を単に上位、下

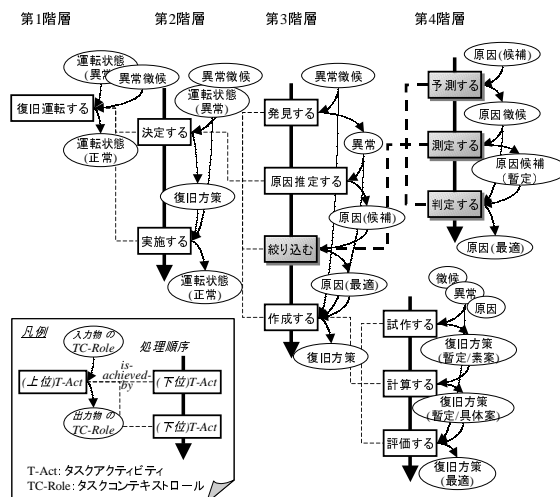


図2 is-achieved-by 階層とタスクコンテキストロール

位と呼び、以下にタスクコンテキストロールの分類について述べる。

- 最上位タスクアクティビティの入出力のタスクコンテキストロール： 最上位タスクアクティビティの入出力のタスクコンテキストロールはそのタスクアクティビティにしか依存しないため、ほぼそのタスクアクティビティのアクティビティロールと同じ定義となる。
- 上位タスクアクティビティの入出力と一致する物のタスクコンテキストロール： 下位タスクアクティビティは上位タスクアクティビティの内部処理を明示化したものと捉えることができる。そのため目的は共有されており、上位タスクアクティビティの入出力物と一致するものは、タスクコンテキストロールも一致する。第3階層の「発見する」の入力は、第2階層の「決定する」の入力と一致し、そのタスクコンテキストロールは「異常徴候」となる。
- 内部処理生成物のタスクコンテキストロール： 上位タスクアクティビティの入出力と一致しない物は、そのタスクアクティビティの内部処理で必要とされる物である。これに与えられるタスクコンテキストロールは次のような依存関係がある。例えば、図2第2階層のタスクアクティビティの粒度では、復旧運転タスクを「(方策を)決定する」、「(方策を)実施する」の2つのアクティビティで記述している。この2つのアクティビティにおいて「実施する」がタスクのコンテキストを決定する。そのため、タスクコンテキストロール「復旧方策」はタスクコンテキストを決定する「実施する」に依存して定義される。
- 状態ロール： 状態ロールは処理の進行に伴う状態の変化を表すロールである。復旧運転では「試作する」により「復旧方策暫定案」を作成し、「評価する」

を経て「復旧方策最適案」となる。この「暫定案/最適案」というタスクロールは、処理の過程で同じアイデンティティを持つ「復旧方策」の状態遷移 [林 98] を表している。「復旧方策」に伴う「暫定案/最適案」の様に、他のタスクロールを補助する役割であることが多い。このような補助的役割の持つロールを本論文では「復旧方策（暫定）」のように（ ）を用いて表す。

3.4 ドメインロール

専門知識の多くの部分を占めるものがドメイン知識である。タスク知識と切り離されたドメイン知識は利用目的からある程度独立した客観性の高い知識となる。タスクの実行の際には、知識システムはタスク知識を用いて問題解決を進め、必要に応じてドメインの知識を参照する。ドメイン概念はコンテキストにおいてその名前が変化する。

例えば、3.1 節で述べたように流量コントローラと液レベルコントローラによってカスケード制御されている制御弁（図 1）が「ナフサの流量を制御するという役割」を担うと、「ナフサ流量制御弁」と呼ばれる。また「オーバーヘッドドラム内のナフサの液レベルを制御するという役割」を担うと、この制御弁は「オーバーヘッドドラム液レベル制御弁」と呼ばれる。この「ナフサの流量を制御するという役割」などドメイン概念がその状況によって担う「役割」を表す概念がドメインロールである。このドメインロールは「ナフサ」、「流量」など「制御弁」以外の他のドメイン概念に依存して定義されることが多い。

3.5 ドメインロールを決定する際に必要な概念の分類

3.1 節で述べた問題点は、コンテキストに依存するロール概念とコンテキストに依存しない基本概念を混同することに起因する。

そこで本節では基本概念とロール概念を担った概念とを区別する指針を与えるために、ドメインロールを決定する際に必要なドメイン概念の分類を行う。今回、例とした石油精製プラントにおけるドメイン知識のトップレベルカテゴリーを、

- デバイス：プラントを構成する部品
 - 対象物：デバイスの処理対象
 - 属性：デバイスや対象物が持ちうる属性
 - ドメインアクティビティ：デバイスの挙動や機能を表す
 - 状態/性質語彙：対象物などの状態や性質を表す
- の 5 つとして約 300 個のドメイン概念を抽出した。そして、これらのトップレベルカテゴリーごとにドメインロールを決定する際に必要な概念を抽出し、合計 27 種類に分類した（図 3）。なお、図 3 にはドメインロールを決定する際に必要な概念の分類の一般性を示すため石油精製プラントに現れるドメイン概念以外も例として示して

いる。

ここで注意すべきことは、全ての人工物は果たすべき目的を持って作られることである。例えば「流量制御弁」は流体の流量を制御する目的で作られる。このように本来の果たすべき目的を表す名前を持つ人工物は、ドメインの状況によって名前が決定されたわけではない。よってこれらはドメインロールを担った概念ではなく、基本概念として扱われることが多い [溝口 99]。従って「流量制御弁」という名前は『機能の対象とする属性』から決定されていると考えられるが、この名前は本来の目的を表しているので「流量制御弁」は基本概念として扱うべきであるとする。

4. 概念工房によるオントロジー構築の流れ

4.1 構築方法 AFM の概要

オントロジー構築方法 AFM ではタスク分析を行い、問題解決で参照されるドメイン概念を絞り込んでいく。タスクアクティビティの対象となるドメイン概念が問題解決過程においてどのようなタスクロールを担っているかを明らかにし（3.3 節）、次にタスクロールと結びついたドメイン概念を中心に抽出し、ドメインロールを考慮して（3.5 節）、整理を行う。

本章では、石油精製プラントの復旧運転問題を例題に用いて AFM に基づいた構築ガイドシステム概念工房によるオントロジー構築過程を示す。この構築過程は大きく 4 つのフェーズからなり、さらに 12 個のステップに細分される（図 4）。本システムでは、各ステップでの作業を支援するためにそれぞれに支援ウィンドウが用意されている。システムがステップを提供することで、オーサにオントロジー構築作業の手順を示す。それと共に、システムが各ステップのオーサの作業過程を中間生成物として管理しており、最終出力のオントロジーの修正作業が容易になる。

AFM の大きな流れは以下ようになる（図 4）。

- (1) タスクユニット抽出フェーズ：ドキュメントから、処理を一つだけ含むタスクユニットを抽出する。
- (2) タスクアクティビティ組織化フェーズ：(1) で抽出したタスクユニットをもとにタスクアクティビティの入出力の定義と組織化（is-a 階層化）を行う。
- (3) タスク構造分析フェーズ：処理（タスクアクティビティ）の流れを分析し、処理の入出力の流れを明示することで、タスクコンテキストロールを定義する。
- (4) ドメイン概念整理フェーズ：(3) で抽出したタスクコンテキストロールと結びついたドメイン概念を整理する。このフェーズでは、ドメインロールを担っている概念からドメインロールを分解し、ドメイン概念の組織化（is-a 階層化）を行う。

実際には、これらの作業は一方向のみに進むものではなく、適宜繰り返行われる。以下に各ステップを詳し

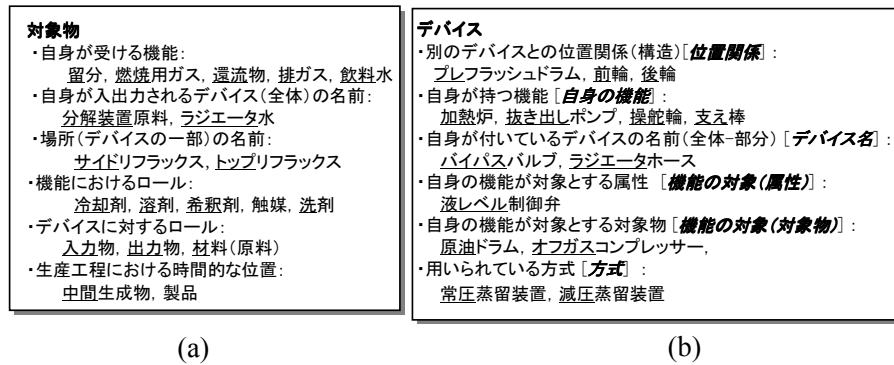


図 3 ドメインロールを決定する際に必要な概念の分類(一部)

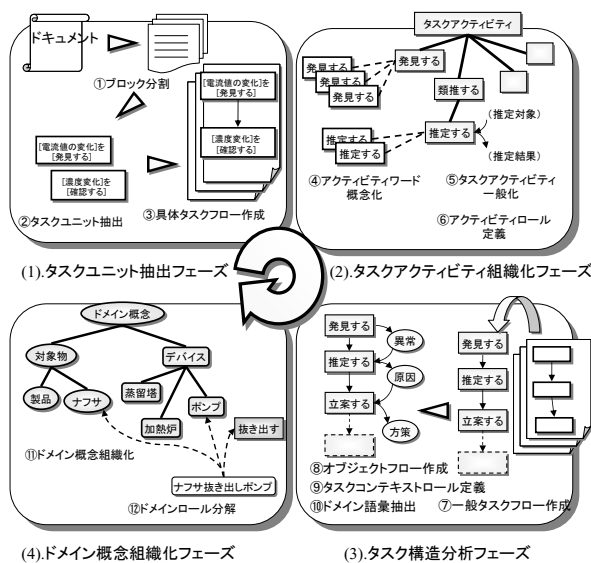


図 4 構築の流れ

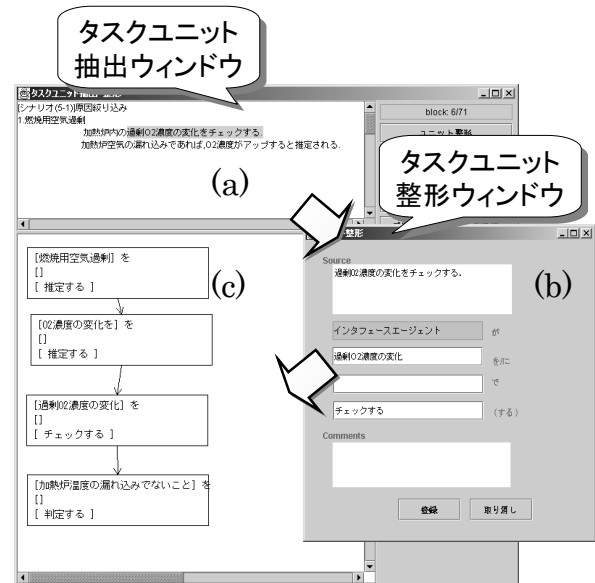


図 5 タスクユニット抽出画面例

く説明する．さらに各ステップでオーサが直面する問題を例示し、その解決策を示す．

4.2 タスクユニット抽出フェーズ(図4(1))

本フェーズでは、まず自然言語で記述されているドキュメントをブロック(後述)に分割する．次にそこから、タスクユニットを抽出する．タスクの中での処理一つ分をタスクユニットと呼ぶ．それらをフローチャート化する．これを具体タスクフローと呼ぶ．

§1 ブロック分割(図4①)

まず自然言語で記述されているドキュメントをオントロジーに必要な語彙(概念)を抜き出しやすい大きさ(ブロック)に区切ることが目的となる．オーサは1つのブロックが1つの問題解決、つまり問題の認識から解の生成までの一連の作業1回分を表すようにドキュメントを区切っていく(図5(a))．

§2 ユニット整形・抽出(図4②)

次に分割したブロックからタスクユニットを抽出していく．オーサに「主語がタスク実行者」となるように文

章を整形した形でタスクユニットを抽出させる．問題解決の各処理は「(問題解決実行者が)“対象”を“処理する”」という形式で記述できる．そこで各文章をこのような記述に改め、1つのタスクユニットが1つの問題解決処理を表すようにする．もとの文章から必要な部分を選択し、必要に応じて語彙を補ってタスクユニット整形ウィンドウの「対象」、「処理」の欄を埋めていく(図5(b))．このように抽出されたタスクユニットが具体タスクフローパネルに表示される(図5(c))．このように文章を整形することで、タスクとドメインのアクティビティの混在を防ぐ．

§3 具体タスクフロー作成(図4③)

前章で抽出したタスクユニットを実行順に並べてフローチャート化して具体タスクフローを作成する(図5(c))は具体タスクフロー作成結果を表している)．

タスク実行者が複数いる場合、あるタスク実行者を中心としたドキュメントでは、そのタスク実行者のアクティビティに対する他のタスク実行者のリアクションが省略

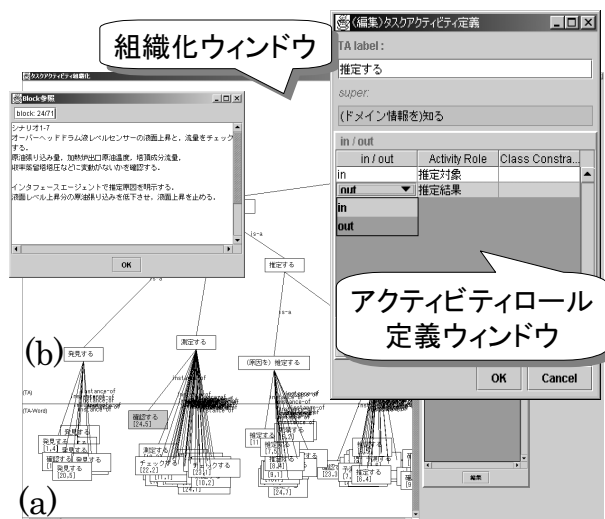


図 6 タスクアクティビティ組織化画面例

されることがある．具体タスクフローを作成し，タスクアクティビティの実行順序を明示することでドキュメントでは省略されている概念の抜けを防ぐ．

アクティビティには「次に～すると決める」，「～できるかどうか確かめる」，「～することを繰り返す」のようなアクティビティを制御する一種のメタなレベルのものがあがるが，これらと属性などを処理対象に持つアクティビティを同じように扱くと，整合性の高いオントロジーを構築する妨げとなる．そのためこの時点でオーサは具体タスクフローを制御しているユニットを注意して区別することが重要である．しかし現時点では，本システムはこの 2 種類の性質の異なるアクティビティを区別して管理する機能は提供していない．

4.3 タスクアクティビティ組織化フェーズ (図 4 (2))

次に 4.2 節で抽出したタスクユニットをもとにタスクアクティビティの定義と組織化を行う．

抽出されたタスクユニットの動詞部には，ドキュメントから抽出されたままの処理を表す語彙が記述されている．本稿では，これらをタスクアクティビティワードと呼ぶ．タスクアクティビティはこれらのタスクアクティビティワードを概念化した結果の概念を指す．タスクアクティビティワードとタスクアクティビティは図 6 のようなウィンドウで編集・表示される．

組織化ウィンドウ (図 6) は，オーサのタスクアクティビティワードの概念化およびタスクアクティビティの一般化・定義の過程を支援する．図 6 では，タスクアクティビティワードがワードパネル (図 6 (a)) の画面に表示されている．それらを概念化した結果のタスクアクティビティが概念パネル (図 6 (b)) に表示される．

§ 1 タスクアクティビティワード概念化 (図 4④)

ドキュメントから抽出したままのデータとしてのタスクアクティビティワードはまだ概念化されておらず，表記揺れ等がある．ドキュメント上のタスクアクティビティワードには同じラベルで記述されているが，異なった意味を持つ場合があったり，逆に異なるラベルで記述されているが同じ意味を持つ場合がある．そこでタスクアクティビティワードを概念化し，ラベルを統一する．

このようなラベルの統一作業では，オーサにとってタスクアクティビティワードがどのようなタスクコンテキストで使われていたかということが重要な情報となる．本システムは，各タスクアクティビティワードがどのブロックから抽出されたかという作業履歴を管理しており，オーサがタスクアクティビティワードを選択するとそれが記述されていたブロックを別のウィンドウでオーサに提示することで (図 6 左上)，各タスクアクティビティワードの用いられているタスクコンテキストの把握を支援する．以後のステップにおいても抽出した概念がどのようなタスクコンテキスト内で用いられていたかということはオーサにとって重要な情報であるが，現時点でこのように別ウィンドウを用いて抽出元の文章を提示する機能が実装されているのは，このステップのみである．

また，本システムではオーサが概念化する際に，対象とするタスクアクティビティワードを絞ることで労力の軽減を図る．

§ 2 タスクアクティビティ一般化 (図 4⑤)

次に § 1 で得られたタスクアクティビティの一般化を行う．下位概念が与えられていて，その概念間で共通の部分に注目し，異なる情報を捨象することで，上位概念を抽出する作業を一般化という．本システムでは，概念の is-a 階層をボトムアップに構築していく．

なお，本システムには is-a 階層構築を内容レベルで支援する機能は実装していない．

§ 3 アクティビティロール定義 (図 4⑥)

このステップでは，タスクアクティビティの入出力のアクティビティロールを定義する．アクティビティロール定義ウィンドウは入出力メニューを持つ．3.3 節で述べたようにタスクアクティビティは処理対象や処理結果という対象を入出力として持つ．

図 6 右側に，タスクアクティビティの入力 (in) と出力 (out) のアクティビティロールの定義を示す．

このようにアクティビティの入出力のアクティビティロールを記述することで，次のフェーズにおける作業であるアクティビティの入出力の流れを明示する作業が容易になる．

これらの各作業は直列に実行するのではなく，ある作業で不都合が出るたびに作業を繰り返すことで組織化を段階的に詳細化していく．

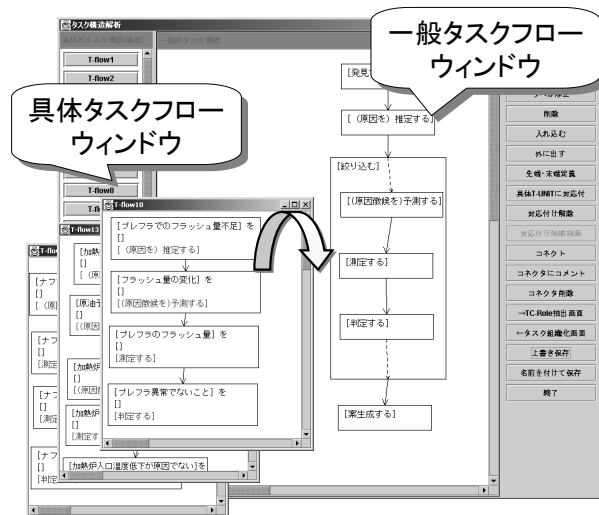


図 7 一般タスクフロー作成画面例

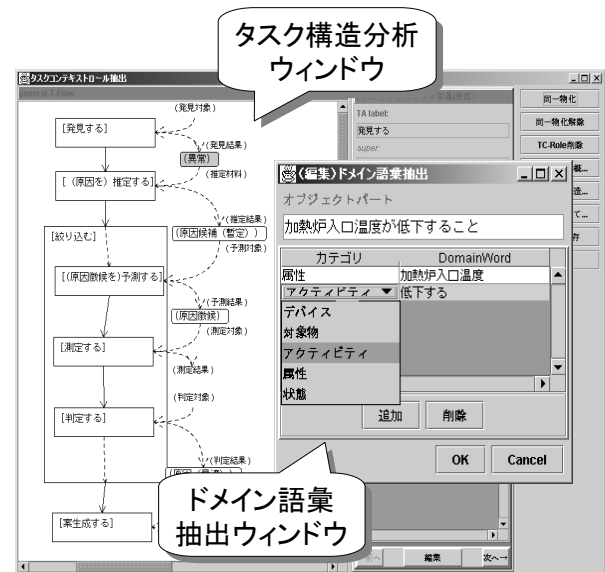


図 8 タスク構造分析画面例

4.4 タスク構造分析フェーズ (図 4 (3))

本フェーズでは、4.3 節で組織化したタスクアクティビティや、その入出力のアクティビティロールを用いて、具体タスクフローを一般化することで、タスクの構造分析を行う。そしてタスクアクティビティが参照するドメイン概念 (入出力オブジェクト) の流れを明らかにして、タスクコンテキストロールを抽出する。

なお、本研究で扱うタスク構造分析は、タスクから参照されるドメイン概念を絞り込むことが目的であり、タスクのプロセスを厳密に記述することが目的ではない。そのため、このステップでは、我々の研究室で開発された概念レベルプログラミング環境 CLEPE[瀬田 98] などの問題解決モデル記述環境で行う厳密なタスクプロセスの記述は行わない。

§ 1 一般タスクフロー作成 (図 4 (7))

ここでは、§ 3 で作成した複数の具体タスクフローを参照しながら、そのタスクの一般的構成 (操作の流れ) を明らかにし、オーサが一般的なタスクフローを作成する。この一般的なタスクフローを一般タスクフローと呼ぶ。図 7 に示すウィンドウでオーサの一般タスクフローの作成を支援する。具体タスクフローウィンドウには § 3 で作成した具体タスクフローが表示されている。ここで参照する具体タスクフローには、§ 1 においてタスクアクティビティワードのラベルを統一したタスクアクティビティが表示される。この具体タスクフローを参照してオーサは一般タスクフローを作成する。

一般タスクフローの各ユニット (一般タスクユニット) と各具体タスクフローのユニット (具体タスクユニット) との対応関係はシステムが保持しており、一般タスクユニットを選択すると、対応する具体タスクユニットの色が変わり、対応関係を明示する。

このように共通の処理の流れを持つ複数の具体タスクフローを参照しながら一般タスクフローを構築すること

で、具体タスクフロー構築時において、タスクがうまく分離できていない問題や、タスクアクティビティの抽出忘れの問題を発見することができる。

また、オーサは 3.3 節で述べたアクティビティの is-achieved-by 関係を図 7 右のような入れ子構造で記述する。 § 2 オブジェクトフロー作成 (図 4 (8))

次にオブジェクトフローの構築を行う。一般タスクフローを作成したのち、それを用いて、タスクアクティビティのある出力がどの入力から生じているかなど、タスクアクティビティの入出力間の関係を明示したものをオブジェクトフローと呼ぶ。 § 3 で定義された各処理 (タスクアクティビティ) の入出力と、 § 1 で構築したタスクアクティビティの順序は明らかになっている。この 2 つを元に、出力と入力のつながりを明示化していく。タスク構造分析ウィンドウ (図 8) には、一般タスクフローとそれを構成するタスクアクティビティの入出力のアクティビティロールが表示される。同一と分かった 2 つのアクティビティロールを 1 つのオブジェクトに同一化することで、オブジェクトの流れを表す。図 9 はオブジェクトフロー作成の過程を表している。

このようにオブジェクトフローを構築することで、定義したタスクアクティビティの出力のオブジェクトが次に出てくるタスクアクティビティに参照されていないなどの問題がないことを確認する。こうして問題解決に最低限必要なオブジェクトを絞り込んでいく。またオブジェクトの流れを明示化することで、次項での作業であるタスクコンテキストロールの定義が容易になる。

§ 3 タスクコンテキストロール定義 (図 4 (9))

作成したオブジェクトフローを用いてタスクコンテキストロールを定義していく (図 9 (b), (c))。

タスクコンテキストロールの定義は 3.3 節で述べた is-

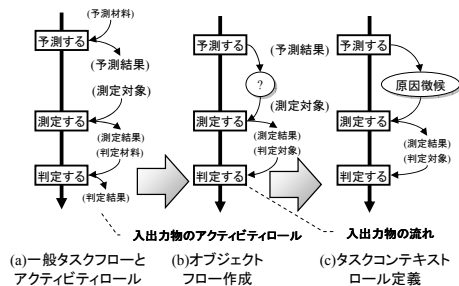


図 9 オブジェクトフローとタスクコンテキストロール定義

achieved-by 最上位のタスクアクティビティの入出力から始めて、順次下位のタスクアクティビティへのタスクコンテキストロールの定義作業を進める。タスクユニットの整形ステップにおいて (§ 2), タスクアクティビティの対象 (目的語) として抽出したドメイン語彙に対応するオブジェクトにこのようなタスクコンテキストロールを結びつけることで、タスク知識とドメイン知識との関連を保持する。また、タスクコンテキストロールを定義し、それと結びついたドメイン概念を抽出することで、問題解決に参照されているドメイン概念を絞り込んでいくことができる。

§ 4 ドメイン語彙抽出 (図 4⑩)

次にオーサは、前節までの手順に従って抽出したタスクコンテキストロールと結びついたドメインの語彙を抽出し、3.5 節で述べた 5 つのトップレベルカテゴリーのいずれかに分類する。この過程を図 8 右のドメイン語彙抽出ウィンドウで支援する。ドメイン語彙抽出ウィンドウでは、オーサがドメイン語彙を抽出する過程を支援するために 5 種類のトップレベルカテゴリーを提示するプルダウンメニューが表示され、これを参考にしてオーサはドメインの語彙を抽出する。

4.5 ドメイン概念整理フェーズ (図 4 (4))

§ 1 ドメイン概念組織化 (図 4⑪)

次にドメインロールの分解およびドメイン概念の組織化を行う。

前節で抽出したドメインの語彙は、ドキュメントに記述されているままの語彙である。本論文ではこれらのドメインの語彙をドメインワードと呼ぶ。ドメイン概念はこれらのドメインワードを概念化した結果の概念を指す。

組織化ウィンドウ (図 10) はドメイン概念の 5 つのトップレベルカテゴリーごとに用意されており、オーサのドメイン概念の組織化の過程を支援する。図 10 はデバイス用の組織化ウィンドウである。ドキュメントから抽出されたドメインワードは抽出時に指定された (図 8 右) カテゴリー用のドメインワードパネルに表示され (図 10 (b)), それらを概念化したドメイン概念がドメイン概念パネルに表示される (図 10 (a))。ドメイン概念組織化の手順はタスクアクティビティ組織化の手順と同様で

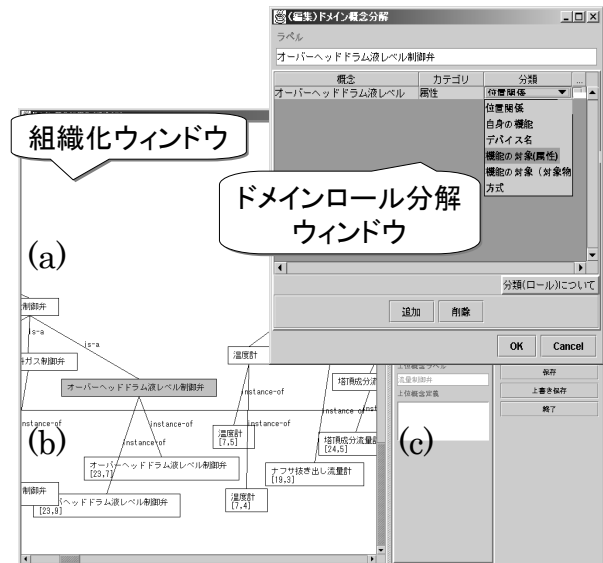


図 10 組織化ウィンドウ画面例 (デバイス用)

ある。ここで注意することは、図 10 に表示されたドメイン概念はドキュメントに記述されたままのかたちであり、基本概念とドメインロールを担った概念とが混在していることである (3.1 節)。そこでドメインロールを担った概念から、ドメインロールを分解する過程をドメインロール分解ウィンドウで支援する。

§ 2 ドメインロール分解 (図 4⑫)

ドメインロール分解ウィンドウ (図 10 右) を用いて、オーサは 3.5 節で述べたドメインロールを決定する際に必要な概念を参照することでドメインロールの分解を行う。

ドメインロール分解ウィンドウ (図 10 右上) は、ドメインロールを決定するために必要な概念をプルダウンメニューでオーサに提示する。図 10 はデバイス用のウィンドウを示しており、デバイスのドメインロールを決定するための概念が提示されている。分類メニューの項目は図 3 (b) と対応しており、例えば、メニュー項目「位置関係」は「別のデバイスとの位置関係」を表している。

オーサは提示された分類を参考にして、ドメインロールを担った概念からそのドメインロールを決定している概念を判定し、その決定する概念を分解していく。この作業をドメインロールの分解と呼ぶ。

これらの分解作業の結果はシステムが管理しており、分解された概念はその選択されたトップレベルカテゴリーの組織化ウィンドウの組織化パネルに表示される。

このようにドメインロールを分解する作業をオーサが行うことで 3.1 節で示した問題を解消する。

オーサがドメインロールの分類を参考にして「液レベル制御弁」を分解して「制御弁が機能する属性」である「液レベル」を得る。この過程でオーサは「液レベル制御弁」は「制御弁」が「自身の機能が対象とする属性」から決まるドメインロールを担った概念であることがわか

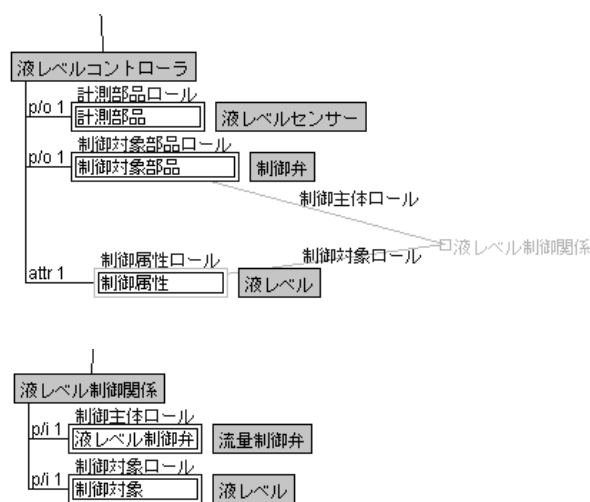


図 11 オントロジーエディタとの結合

るため、等質でない概念の混在を防ぐことができる。

4.6 オントロジーエディタとの結合

2章で述べたように概念工房は、概念の切り出しを支援するオントロジー構築の上流工程を支援するシステムであり、概念の詳細な定義を記述する支援は行わず、それは下流工程を支援するオントロジーエディタで記述される。概念工房で抽出する関連性を用いてオントロジーエディタ [古崎 02a] 上でドメインロールの詳細な定義が記述可能である。概念工房で構築したオントロジーはオントロジーエディタが提供する記述環境に半自動的に変換される。概念工房ではドメインロールを担った概念から、前節で述べた「ドメインロールを決定する際に必要な概念」を抽出し、それらの関連性を明示化することでドメインロールを記述する (§2)。オントロジーエディタではロール概念を厳密に定義する。しかし、概念工房とオントロジーエディタでは支援目的の違いから、それぞれ重視する情報が異なるため、ロール概念のすべてを自動的にオントロジーエディタへ変換するために必要な情報が足りない場合がある。そのためオントロジーエディタにおける記述形式が一意に同定できない場合があり、ドメインロールの定義に関しては一部、オントロジーエディタに変換されたオントロジーをユーザが手動で修正する必要がある。

3.4 節の「制御弁」の例をオントロジーエディタを用いてロール概念を用いて記述すると図 11 のようになる。このように記述することで、概念が担うロール概念が何から決定されているかを明示化することができ、また基本概念と『制御する』機能が対象とする属性から決定されているロール概念を担った概念との違いを捉えた記述が可能になる。

5. 概念工房の利用例

本章では概念工房の利用例について述べる。筆者らが所属する研究室では、旧通産省の支援のもと行われたヒューマンメディアプロジェクトの一環として石油精製プラントのオントロジーを構築した。詳細は別稿 [古崎 02b] で報告されている。本システムは、ヒューマンメディアプロジェクトでも検討されていたロール概念に関する考察をより詳細化し、そこから得られた知見に基づいて設計された。本研究では、このプラントオントロジーを作る元となった日石エンジニアリング株式会社から提供された A4 版 18 ページのプラント運転に関する技術資料を題材として、概念工房を用いてオントロジーの再構築を行った。その結果として、ドキュメントから 355 個のタスクユニットを整形し、36 個のタスクアクティビティで構成されるタスクオントロジーを再構築した。それらを用いて、5 個の一般タスクフローを抽出し、常圧蒸留塔回りの概念を中心に 356 個のドメイン概念を抽出し、そのうち 190 個を基本概念として定義したドメインオントロジーを構築した。このように本稿で提案したオントロジー構築方法と構築ガイドシステムを用いてプラントオントロジーを再構築した結果、従来のプラントオントロジーにおいて留分、軽沸成分、抜き出し流量など二十数箇所においてロール概念を担った概念と基本概念とを混同して定義していたことがわかった。例えば、従来のプラントオントロジーでは、常圧蒸留塔から出力される「留分」を基本概念として定義していた。しかし、留分はドメインロールを担った概念であり (3.5 節)、基本概念として定義するべきではない。このように、概念工房はオントロジーの専門家にとっても困難である基本概念とロール概念の峻別過程を支援できることが確認された。

このように概念工房を利用して構築したオントロジーはロール概念と基本概念の違いを捉えている。そのため、ヒューマンメディアプロジェクトにおけるプラント運転状況説明生成システムなどで状況に応じた説明文を生成する際に、概念工房を用いて再構築したプラントオントロジーに記述された概念に基づくことで、従来のプラントオントロジーよりコンテキストに応じた語彙を表現することがより適切に行うことが可能になる。

6. 関連研究との比較

本章では、国内外のオントロジー構築支援を目指した研究と本研究との比較を行う。

オントロジー構築を支援する計算機環境としては、概念工房のような概念の組織化を支援するシステムと、オントロジーエディタのようなオントロジーの記述環境に分かれる。

オントロジーにおける概念の組織化を支援するシステムとして、山口らは計算機可読辞書の 1 つである WordNet

を利用して領域オントロジーの構築を支援するシステム DODDLE の開発を進めている [山口 99]。DODDLE で構築するのは、ある領域の概念ラベルとそれらの包含関係からなる階層記述のみが存在するオントロジーである。また Infosleuth [Hwang 99] は、ドキュメントから自動的に概念階層を構築していくオントロジー構築システムである。Infosleuth は抽象度の高い seedword をオーサがシステムに入力し、自然言語処理や機械学習の技術を利用することで、自動的にドキュメントから概念を抽出して概念間の関連付けを行うシステムである。これらの自動でオントロジーを構築するシステムと異なり、概念工房はオントロジー構築を手動で行う。本システムでは自動でオントロジーを構築するシステムでは支援できない厳密な、タスクとドメインとの分離やロール概念に関する支援を行う。また KBSI (Knowledge Based Systems, Inc.) による IDEF5 は、企業の様々な業務活動のモデリングを意識したオントロジー開発方法論である [Knowledge Based Systems 94]。IDEF5 同様、企業活動モデリングのためのオントロジー構築方法論として、トロント大学で開発された TOVE 方法論 [Uschold 96] や、エジンバラ大学で開発されたエンタープライズオントロジーの経験に基づいた Methodology [Uschold 96]、また López らの METHONTOLOGY [López 99] などがある。これらの支援環境と比べて概念工房は、ロール概念に関する考察を基盤として設計されているため、ロール概念に対して詳細に支援することができる。そしてこれらのシステムが扱うオントロジーは主にドメインオントロジーであり、タスクオントロジーは含まれていない。AFM はタスク依存のドメインオントロジー構築方法であり、まずタスクオントロジーを構築してからドメインオントロジーを構築する。問題解決過程 (タスク構造) を明らかにすることで、それらに付随するドメイン概念が整理しやすくなる。一方、概念工房では、DODDLE の、電子化辞書のような既存のオントロジーを用いた構築方法はサポートしていない。その重要性は認識しているが、本研究では扱っていない。

またソフトウェア開発における対象領域分析の方法として UML (Unified Modeling Language) を用いた方法がある。UML はオブジェクト指向モデリングにおいて業務の分析などを行う際に用いるオブジェクト指向分析・設計の標準表記法である。UML では、ユースケース図やアクティビティ図などでタスク分析を行う。AFM の前半の作業であるタスク分析は UML で提案されているタスク分析と共通するところが多い。UML と本研究の違いは、本研究ではロール概念に関する考察に重点をおいていることである。そのため UML やそれを利用したツールに比べて、概念工房の優位性は特にドメインを人工物に限定して、ロール概念に関する考察を深めることで、タスクロールおよびドメインロールと基本概念との峻別に対する詳細な支援を行うことができる点にある。

また筆者らの研究室でもオントロジー記述環境の開発が進められている。概念レベルプログラミング環境 (CLEPE) はタスクオントロジーを記述し、それを用いて問題解決モデル (インスタンスモデル) を作成して概念レベルでモデルを実行する実行環境である [瀬田 98]。CLEPE は問題解決モデルの実行を主な目的としていて、モデル構築に利用するオントロジー構築における上流工程は支援しない。一方、概念工房はオントロジー構築の上流工程である概念の切り出し作業を支援するシステムである。またオントロジーエディタとの違いは、CLEPE はタスクオントロジーに特化した記述環境を持ち、モデルの概念レベルでの実行を目標にした記述・利用環境であるのに対してオントロジーエディタは汎用のオントロジー記述環境であり、モデルの実行を目標にしていないことである。

オントロジー記述環境に関してはいくつかの開発例がある。例えば、Swartout らのドメインオントロジーを構築するための Web ベースのツール Ontosaurus [Swartout 96]、スタンフォード大学の KSL が開発したオントロジーの共同構築・共有をはかる Ontolingua Server [Farquhar 96] や共有オントロジーの構築を目指す武田らの Designers amplifier [鷹合 98]、また我々の研究室で開発が進められているオントロジーエディタなどがある。概念工房とオントロジーエディタとの違いは 2 章で述べたとおりであるが、他の記述環境については本稿の範囲外であるので省略する。

7. む す び

本論文では、筆者らが開発を進めているオントロジー構築方法 AFM と、それに基づいたオントロジー構築ガイドシステム概念工房について述べた。本システムは Java 言語で実装されている。本システムはタスクオントロジーから構築していくことや、ロール概念についての考察に基づいた設計になっていることが特徴であり、このような構築支援環境は他のシステムに見られない。

概念工房は、以下の機能を提供することで、オントロジー構築過程を支援する。

- ロール概念と基本概念との区別の支援
- 語彙と概念の分離
- 12 ステップの導入による構築手順の提供
- 作業履歴の管理による試行錯誤処理の支援

これらの他にオントロジー構築ガイドシステムに望まれる機能としては、適切なガイドを適切なタイミングで提示する機能が考えられる。しかし、そのためには用意すべき事前知識が膨大であり、機能の実装は困難である。例えば、ドキュメント上で名詞化されて記述している動詞 (アクティビティ) をオーサがアクティビティと判断せずに名詞として抽出した際に、そのことをシステムが示唆することが望ましい。しかし動詞のほとんどは名詞

化することができるため、事前にシステムに用意しておくことは難しい。このようなオーサの誤った概念抽出に対するガイド機能を発火するための引き金として利用するために、オーサの間違いうるパターンを事前にシステムに用意しておくことは困難である。筆者らはドキュメントからオントロジーを構築する過程におけるこのようなオーサの間違いうるパターンを蓄積してきたが[久保99]、システムには実装していない。

今後の課題としては、他の多様なドキュメントを扱うことでルール概念の考察を深め、システムにガイド機能を実装する必要がある。§3で述べたように本システムはオントロジー構築における作業ステップを提供し、その作業は直線的ではなく作業を繰り返すことでより適切なオントロジーを構築することを支援する。そのためには、あるステップにおける不都合をそのステップ内で解決するうえで、そのステップ内で解決可能かどうか、または、どのステップに戻ればよいのかという基準の整理が重要である。現在、筆者らは、オーサが切り出した概念を編集することで他の概念への影響の伝播を分析することを支援するために必要となる、構築過程で現れる中間生成物間の依存性に関する検討を進めている[船本02]。また、本システムは既存のオントロジーを用いた支援を行っていない。そのため既存のオントロジーを用いた支援の枠組みについて検討することが今後の課題である。

謝 辞

本研究開発は、通産省産業科学技術研究開発制度として、新エネルギー・産業技術総合開発機構からヒューマンメディアの研究開発として(財)イメージ情報科学研究所が委託を受け進めている研究開発の一環で行った。また、本研究に関して有益な助言を頂いた大阪大学 池田満助教授、そして構築ガイドシステムの実装に協力して下さい(株)日本総合研究所に感謝致します。

◇ 参 考 文 献 ◇

- [Farquhar 96] Farquhar, A., Fikes, R., and Rice, J.: The Ontolingua Server: a Tool for Collaborative Ontology Construction, *Proceedings of the 10th Banff Knowledge Acquisition Workshop* (1996).
- [Goh 97] Goh, C. H.: Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources, *Phd*, MIT (1997).
- [Guarino 98] Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources, in *International Conference on Lexical Resources and Evaluation* (1998).
- [林 98] 林 他: 概念間関係に関するオントロジー的考察 ~ is-a, part-of, identity ~, 信学技報 AI98-40, pp. 1-8 (1998).
- [船本 02] 船本和宏: オントロジー構築ガイドシステムにおける処理履歴分析支援インタフェースの設計, 大阪大学工学部電子工学科特別研究論文 (2002).
- [Hwang 99] Hwang, C. H.: Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information, *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)* (1999).

- [Knowledge Based Systems 94] Knowledge Based Systems, I.: IDEF5 Method Report (1994), <http://www.idef.com/idef5.html>.
- [古崎 02a] 古崎, 来村, 池田, 溝口: 「ルール」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).
- [古崎 02b] 古崎, 来村, 佐野 他: 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. 407-419 (2002).
- [久保 99] 久保, 古崎, 来村, 溝口: オントロジー構築方法 AFM(Activity-First Method) の詳細化の試み, 人工知能学会全国大会論文集(第13回), pp. 114-117 (1999).
- [López 99] López, M. F., Gómez-Pérez, A., Sierra, J. P., and Sierra, A. P.: Building a Chemical Ontology Using Methodology and the Ontology Design Environment, *IEEE Intelligent Systems*, Vol. 14, No. 1, pp. 33-46 (1999).
- [溝口 94] 溝口理一郎: 知識の共有と再利用研究の現状と動向, 人工知能学会, Vol. 9, No. 9, pp. 3-9 (1994).
- [Mizoguchi 95] Mizoguchi, R., Ikeda, M., Seta, K., and Vanwelkenhuysen, J.: Ontology for Modeling the World from Problem Solving Perspectives, in *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing* (1995).
- [溝口 99] 溝口, 池田, 来村: オントロジー工学基礎論, 人工知能学会誌, Vol. 14, No. 6, pp. 1019-1032 (1999).
- [瀬田 98] 瀬田, 池田, 角所, 溝口: 問題解決オントロジーの構築 - スケジュールタスクオントロジーを例にして -, 人工知能学会誌, Vol. 13, No. 4, pp. 597-608 (1998).
- [Swartout 96] Swartout, B., Patil, R., Knight, K., and Russ, T.: Toward Distributed Use of Large-Scale Ontologies, *Proceedings of the 10th Banff Knowledge Acquisition Workshop* (1996).
- [鷹合 98] 鷹合, 武田, 西田: 協調設計作業を実現するための設計者支援環境, 電子情報通信学会論文誌, Vol. J81-D-I, No. 5, pp. 488-495 (1998).
- [Takaoka 96] Takaoka, Y. and Mizoguchi, R.: Identification of Ontologies to Reuse Knowledge for Substation Fault Recovery Support System, *Decision Support Systems* 18, pp. 3-21 (1996).
- [Uschold 96] Uschold, M.: Building Ontologies: Towards A Unified Methodology, *Proc. Expert Systems* 96 (1996).
- [山口 99] 山口 他: 計算機可読辞書を利用した領域オントロジー構築支援環境, 人工知能学会誌, Vol. 14, No. 6, pp. 1080-1087 (1999).

〔担当委員: 木下哲男〕

2001 年 9 月 20 日 受理

著 者 紹 介



石川 誠一

2002 年 3 月大阪大学大学院基礎工学研究科修士課程修了。同年 4 月エス・ティ・ティ・コムウェア株式会社入社、現在に至る。在学中は、タスク・ドメインルールに基づくオントロジー構築ガイドシステムの設計と開発に従事。



久保 成毅

2000 年 3 月大阪大学大学院工学研究科修士課程修了。同年 4 月西日本電信電話株式会社入社。法人営業本部にて、顧客組織における情報共有・知識共有のための情報システム・通信システムのコンサルティング、トータル・コーディネートに携わる。在学中は、オントロジー構築方法論および、構築ガイドシステムに関する研究に従事。



古崎 晃司(正会員)

1997 年大阪大学工学部電子工学科卒業。2002 年同大学院工学研究科博士課程修了。同年、化学工学会嘱託研究員、現在に至る。博士(工学)。現在、大阪大学産業科学研究科にて、ナノテクノロジーオントロジーの開発、オントロジー構築・利用環境の設計・開発に関する研究に従事。情報処理学会会員。



来村 徳信(正会員)

1991 年大阪大学基礎工学部情報工学科卒業。1993 年同大学院基礎工学研究科前期課程修了。同年、同大学産業技術研究所技官。1994 年同助手。現在に至る。博士(工学)。物理的システムに関するオントロジー工学的考察と、それに基づいたモデル化と推論に関する研究に従事。1996 年人工知能学会創立 10 周年記念論文賞受賞。情報処理学会会員。



溝口 理一郎(正会員)

1977 年同大学院基礎工学研究科博士課程修了。1978 年大阪大学産業科学研究所助手、1987 年同研究所助教授、1990 年同教授。現在に至る。工学博士。音声の認識・理解、エキスパートシステム、知的 CAI システム、オントロジー工学の研究に従事。1985 年 Pattern Recognition Society 論文賞、1988 年電子情報通信学会論文賞、1996 年人工知能学会創立 10 周年記念論文賞、1999 年 ICCE99 Best paper Award 受賞。電子情報通信学会、情報処理学会、教育システム情報学会、日本認知科学会、Intl. AI in Education(IAIED) Soc., AAAI, IEEE, APC of AACE 各会員。現在、IAIED Soc. 会長、及び APC of AACE の次期会長。