

学位論文 博士（工学）

ルール・オントロジーとドメイン・オントロジー
に基づく知識継承支援システムの開発と評価

2010年度

慶應義塾大学大学院理工学研究科

岡部 雅夫

目次

第 1 章 序論.....	1
1.1 背景.....	1
1.2 目的.....	1
1.3 本論文の構成.....	2
第 2 章 背景知識・関連研究.....	3
2.1 製造現場における二つの熟練.....	3
2.1.1 匠的熟練と知的熟練.....	3
2.1.2 知的熟練の重要性の増大.....	3
2.2 ナレッジマネジメント.....	4
2.2.1 ナレッジマネジメント.....	4
2.2.2 個人知から組織知への変換.....	4
2.2.3 暗黙知と形式知.....	5
2.2.4 知識変換.....	5
2.2.5 ナレッジマネジメント・ツール.....	7
2.2.6 ナレッジマネジメント・ツールの導入事例.....	8
2.2.7 ナレッジマネジメント推進上の課題.....	9
2.2.8 ナレッジマネジメントと情報技術.....	10
2.3 知識ベース.....	13
2.3.1 知識ベースについて.....	13
2.3.2 Prolog.....	15
2.3.3 エキスパートシステム.....	17
2.3.4 エキスパートシステムの現状.....	19
2.4 知識モデリング.....	20
2.5 オントロジー.....	21
2.5.1 オントロジーについて.....	21
2.5.2 オントロジーの分類.....	23
2.5.3 オントロジー記述言語.....	26
2.5.4 オントロジーの例.....	29
2.5.5 オントロジーの役割.....	31
2.5.6 オントロジー構築支援ツールとオントロジー・ポータル.....	32
2.5.7 オントロジーの応用.....	33
2.5.8 オントロジーとナレッジマネジメント.....	34
第 3 章 ルール・オントロジーとドメイン・オントロジーに基づく知識継承支援システム.....	

.....	37
3.1 スコープ	37
3.2 提案システムの概要	37
3.3 業務プロセス・フロー	39
3.4 ルール・オントロジー	40
3.5 ドメイン・オントロジー	41
3.6 ルールベース・システム	42
3.7 支援システム	42
3.7.1 GEN (General knowlEdge Navigator)	42
3.7.2 ルールベース・システム	45
3.8 初期構築手順	48
第4章 モデルケースー停止調整業務ー	51
4.1 概要	51
4.1.1 対象業務	51
4.1.2 検証の進め方	52
4.2 初期構築とその新人による内面化に関する検証	53
4.2.1 初期構築	53
4.2.1.1 業務の概要把握と業務プロセス・フロー構築	53
4.2.1.2 ルール・オントロジー構築	54
4.2.1.3 ドメイン・オントロジー構築	59
4.2.1.4 業務プロセス・フローとオントロジーの関係づけ	60
4.2.1.5 スケジューリング・システム構築	62
4.2.2 新人による内面化に関する評価	65
4.2.2.1 実験方法	65
4.2.2.2 実験結果	67
4.2.2.3 考察	68
4.2.2.4 GENにより独習した被験者によるオントロジーに対する評価	72
4.2.3 業務プロセス・フローの詳細化と作業オントロジーの再構築	72
4.2.3.1 業務プロセス・フローの詳細化	72
4.2.3.2 作業オントロジーの再構築	75
4.3 熟練者による持続的な表出化・連結化に関する検証	76
4.3.1 事業所の長の意思を反映させるためのルール・オントロジーの改善	76
4.3.1.1 正当性関係の抽象度の統一	77
4.3.1.2 最も強く正当性を説明するルールの明示	77
4.3.2 ドメイン・オントロジーからの日本語ルール生成と実行ルールへの変換	79
4.3.2.1 最も浅いルールの分析	80

4.3.2.2 ドメイン・オントロジーの拡充	84
4.3.2.3 日本語ルール生成と実行ルールへの変換	85
4.3.2.4 新たな実行ルールの追加作業の流れ	87
4.3.3 ルール・オントロジーを活用した事業所の長の意思の自動反映	90
4.3.4 熟練者による持続的な表出化・連結化に対する評価と考察	95
4.3.5 いっ水電力量の低減をより重視したスケジューリングの改善	97
第5章 結論	99
5.1 結論	99
5.2 今後の課題と展望	101
補遺Ⅰ	103
1. 平均学習時間の差の検定	103
2. 平均回答時間, 平均誤答数の差の検定	105
補遺Ⅱ	106
1. 単独作業ルール	106
2. 相互依存作業ルール	106
2.1 対称ルール	106
2.2 非対称ルール	106
参考文献	108
関連発表	111
謝辞	113

目次

図 1	本研究の目的を表す全体像.....	2
図 2	SECI モデル.....	6
図 3	属性制約による新たな概念の定義.....	15
図 4	エキスパートシステムの構造.....	17
図 5	is-a 関係の例.....	22
図 6	オントロジーにおける公理.....	23
図 7	John F. Sowa の Top-Level Categories	24
図 8	CyC の Upper Ontology.....	24
図 9	アップパー・オントロジーにおけるトップレベル・カテゴリー.....	25
図 10	Conceptual Graph の例 1.....	28
図 11	Conceptual Graph の例 2.....	28
図 12	PSL における “before” の公理化.....	30
図 13	“skos:broader” の特徴付け.....	30
図 14	オントロジー構築ツール Protégé クラス階層画面.....	33
図 15	MIT Process Handbook における Create to order の記述.....	35
図 16	提案システムの全体像.....	38
図 17	クラス構造の定義例.....	43
図 18	インスタンス表示例.....	44
図 19	業務プロセス・フロー，ドメイン・オントロジー，ルール・オントロジーの関 係.....	44
図 20	GEN システム構成（東京電力における試行での例）.....	45
図 21	ルール編集画面.....	46
図 22	日本語ルール生成と実行ルールへの変換のモジュール構成.....	47
図 23	実行ルールの適用除外アルゴリズム.....	48
図 24	停止調整業務の概要.....	52
図 25 A	総合制御所全体および停止調整業務の業務プロセス・フロー.....	54
図 26	ルール・オントロジーの視覚化方法.....	56
図 27	業務ルールの文字数の分布.....	56
図 28	ルール・オントロジーの一部.....	57
図 29	GEN に組み込まれたルール・オントロジー.....	58
図 30	設備オントロジー.....	59
図 31	作業オントロジー.....	60
図 32	発電所別スケジュール化作業一覧.....	61

図 33 発電所別スケジュール化作業とオントロジーなどとの関係.....	61
図 34 業務ルールを表出化プロセス	63
図 35 スケジューリング・システム入力画面	64
図 36 スケジューリング・システム結果表示画面	64
図 37 業務プロセスに関する知識の欠落による誤答	69
図 38 作業規模に関する知識の欠落による誤答	70
図 39 スケジューリングリファレンスモデル	74
図 40 業務プロセス・フローの詳細化	74
図 41 再構築した作業オントロジー	76
図 42 中間の業務ルールの追加	77
図 43 最も強く正当性を説明するルールの明示	78
図 44 水垢除去を最も強く justify する最も深いルール	79
図 45 最も浅いルールの類別化	81
図 46 「重ねない」のパターン	84
図 47 ルール・オントロジーとドメイン・オントロジーの関係の例	84
図 48 実行ルールへの変換のためのプルダウン・メニューによる指定	85
図 49 変換された Prolog の実行ルール	86
図 50 使用された対象ルールのテンプレート	86
図 51 「開始日、終了日を相互に重ねない」のテンプレート	86
図 52 スケジューリング・システム結果表示（2010 年度）	87
図 53 Prolog の実行ルールへの半自動変換機能	88
図 54 追加ルールが反映されたルール編集画面	89
図 55 追加ルールが反映されたスケジューリング結果	89
図 56 スケジューリング・システムによる 2011 年度停止スケジュール	91
図 57 原価低減を重視した停止スケジュール	92
図 58 スケジューリング結果の変化	93
図 59 いっ水電力量低減を重視するアルゴリズムの変更結果	98

表目次

表 1	GEN のシステム規模.....	43
表 2	最も深いルールの一覧.....	57
表 3	実験結果.....	67
表 4	原因別の誤答数.....	67
表 5	回答方法別実験結果.....	71
表 6	結論部の類別によるルールの分類.....	83
表 7	いっ水電力重視によるルールの適用除外率といっ水電力削減率.....	94
表 8	生成された日本語ルールと熟練者が修正した日本語ルール.....	95
表 9	条件部に実施時期, 作業日数を指定するルール数.....	95
表 10	グループ 1, 3 の学習時間の等母分散の検定.....	103
表 11	グループ 1, 3 の学習時間の母平均値の差の検定 (等母分散を仮定).....	104
表 12	グループ 1, 3 の学習時間の母平均値の差の検定 (等母分散を仮定せず).....	104
表 13	グループ 1, 3 の母平均の差の検定結果.....	105

第1章 序論

1.1 背景

ベテラン技術者の大量退職期を迎え、技術・技能の継承が大きな問題となって久しい。この問題は情報システムのサポートすべき大きな領域とされ、国家的なプロジェクト[1]も推進されたが、この問題は決して一過性の問題ではない。

これまで、我が国の技術・技能は、幅広く深い OJT により形成され、それが、製造業を中心とする我が国の現場の国際競争力の源泉であるとされてきた。仕事の段取りや手順を定める「マニュアル」があっても、現場ではそれを工夫して修正する。そして、その様々な工夫が暗黙知として蓄積される。そして、その暗黙知は、熟練者が新人と一体となって仕事することにより、暗黙知のまま新人へ継承されてきた[2, 3]。ところが、昨今の競争環境の激化に伴い、組織のスリム化が進み、新人が熟練者と一体となって仕事をする機会が得にくくなってきており、これまでのような OJT による技術・技能の継承が困難になりつつある。また、密な人間関係を前提にしたハイコンテクスト社会の終焉も、OJT による技術・技能の継承を困難にしている。

一方で、最近の企業ガバナンス強化の流れの中で、業務マニュアルの整備が進められているが、一部に、業務マニュアルに技術・技能まで盛り込もうとする動きもある。ただし、ガバナンスの強化と技術・技能の継承は同一の枠組みで扱うものではない。技術・技能の継承は、代々蓄積されてきたものを継承していくと同時に、経営環境の変化にも呼応して、より柔軟に経営層の意思を反映し、現場の工夫を折り込み、進化して行くべきものである。

1.2 目的

以上のような背景をもとに、本論文では、スリム化された組織においても、OJT に依存することなく、経営層の意思を反映し、現場の工夫を折り込みながら、技術・技能を組織的に蓄積し、継承していくための IT 支援を提案する。図 1 が本研究の目的を表す全体像である。

本提案システムは、「知的熟練」と言われる技術・技能を対象とし、そのため、知識の体系化・構造化を試行する。その意味で、本提案システムはオントロジーを活用するものである。

本提案システムは、東京電力株式会社 A 総合制御所で停止調整業務をモデルケースとして試験適用され、実際にオントロジーとルールベース・システムを構築し、評価実験を行うことで、その有効性を確認した。

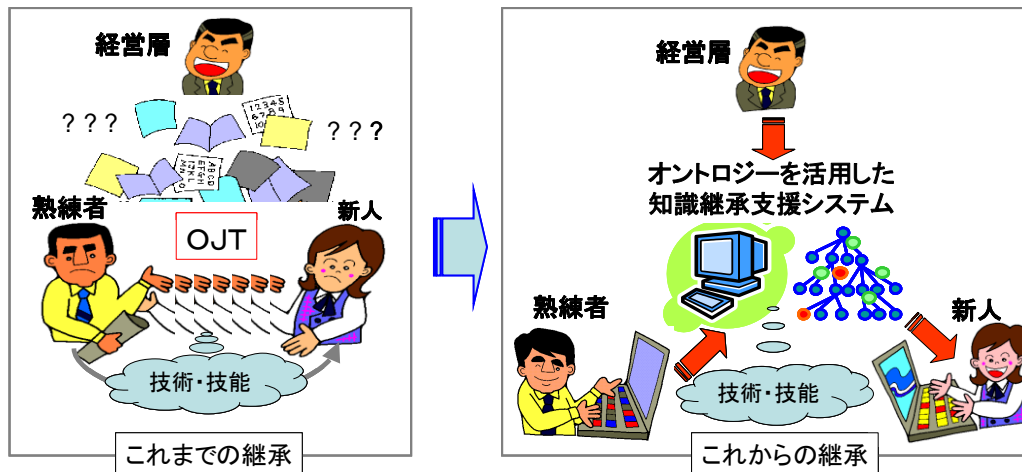


図 1 本研究の目的を表す全体像

1.3 本論文の構成

本論文の構成は次の通りである。

第1章 序論

本研究の背景及び目的を明らかにし、本研究の目指すところについて述べる。また、本論文の構成についても記す。

第2章 背景知識・関連研究

知的熟練、ナレッジマネジメント、ルールベース、オントロジーといった本研究の背景知識および関連研究に関して述べる。

第3章 ルール・オントロジーとドメイン・オントロジーに基づく知識継承支援システム

本研究の提案であるルール・オントロジーとドメイン・オントロジーに基づく知識継承支援システムについて述べる。

第4章 モデルケースー停止調整業務ー

本研究の提案であるオントロジーを活用した知識継承支援システムを東京電力のある現場事業所において停止調整業務という業務に試験的適用したモデルケースおよびそれに対する評価について述べる。

第5章 結論

本研究の結論および今後の課題と展望について述べる。

第2章 背景知識・関連研究

2.1 製造現場における二つの熟練

2.1.1 匠的熟練と知的熟練

もの造りを特徴づける熟練には、匠的熟練と知的熟練（問題発見・解決型熟練）の二つがある[4]。匠的熟練と知的熟練についてそれぞれ説明をする。

(1) 匠的熟練

匠的熟練は、道具・機械・装置などの限界を極限にまで突き詰めた形の加工・組立技能である。例えば、機械のすり合わせ面をキサゲと呼ばれる簡単な道具でミクロン単位（1000分の1ミリ）の精度に仕上げる技能、半導体露光装置用投影レンズの非球面を仕上げる研磨技能、旋盤でミクロン単位の精度を出す技能などがそれらにあたる。

(2) 知的熟練

知的熟練は、製品やそれらを生み出す生産プロセス自体の問題の発生を未然に察知して対処することや、発生した原因を迅速に究明・解決できる熟練技能である。この種の問題発見・解決作業は、製品やそれらを生み出す生産プロセスに関する統合的な知識と、それらを駆使しながら現象面から問題を発生させている真の原因を絞り込んでいく論理的推理能力がバランスよく備わっていることで、迅速かつ的確に実行できる。

2.1.2 知的熟練の重要性の増大

技術・技能の継承といった場合、上記の内、匠的熟練に焦点が当てられる場合が多い。そのため、技術・技能の継承を支援するための情報技術の活用も、バーチャルリアリティなどの情報技術の活用が提案される場合が多い[4他]。

しかしながら、製造業を中心とする我が国の現場の国際競争力の源泉は、実は、幅広く深いOJTにより初めて形成される知的熟練にあり、また、最近の高度に自動化・統合化された製造現場においては、「匠的熟練」がコンピュータによる数値制御に置き換えられつつある一方で、「知的熟練」の重要性はますます高まっている[5]。

知的熟練の重要性がより高まっている大きな要因の一つは、デジタル化を旨とする急速な技術革新の進展である。実際、知的熟練が必要となる作業は、コンピュータの最も不得意とする統合的な知識・ノウハウを駆使した高度かつ複雑な判断業務である。このような判断業務は、スキルやノウハウのデジタル化が高度になされればなされるほど、その希少性を増大させる。その結果、知的熟練の重要性が高まってきている。また、知的熟練の重

要性が増している他の要因としては、製品やそれを支える要素技術の高度化、複合化、微細化や、情報技術やネットワーク技術の発達に伴い情報共有が活発に行われているという状況が挙げられる。

このような現状を踏まえ、本研究では知的熟練の継承を支援する。

2.2 ナレッジマネジメント

2.2.1 ナレッジマネジメント

知的熟練の継承は、広い意味で、ナレッジマネジメントの一分野と考えることができる。ナレッジマネジメントは、企業などの組織において、その共有資産としての「知識」の発見や蓄積、共有、創造、活用などを行うプロセスを体系的な形でマネジメントすることである。ナレッジマネジメントの目的は、既存の知識を管理して共有・再利用の効率化に留まらず、企業の価値を高めるような新しい知識を創造し続けることにある。日本では、ナレッジマネジメントを支援する IT ツールの発展などにより、1990 年代後半から多くの企業において、様々な取り組みがなされてきた。ナレッジマネジメントを浸透させることによって、個人の能力の育成、組織全体の生産性の向上、意思決定のスピードの向上などが実現できるとされている。

以下の 2.2.2 ～2.2.4 において、1995 年に野中郁次郎他の「知識創造企業」[6]において提唱され、現在なお広く受け入れられている「SECI モデル」について説明をする。

2.2.2 個人知から組織知への変換

ナレッジマネジメントにおいて重要なのは、企業の価値を高めるような新しい知識を創造し続けることである。その知識創造の特徴の 1 つとして、「個人知から組織知への変換」が挙げられる。

熟練者など個人が持つ極めて主観的な洞察や勘は、分かりやすい形（言葉など）に変換して社内の人々と共有しない限り、企業にとっては価値を持たない。つまり、ナレッジマネジメントを実践するためには、社内の人々が作り出した知識を、組織全体で具現化する必要がある。企業において、新しい知識はいつも個人から始まり、その個人の知識が組織全体にとって大事な知識に変換される。優秀な研究者のひらめきが会社の新しい特許に結びついたりすることなどがその例である。つまり、組織それ自体では、個人の自発的行動とグループレベルでの相互作用がない限り、知識を創ることはできないのである。組織の中では、個人個人が対話・議論を通じ、当然視されてきたことに疑問を抱くことが、新しい角度から物事を理解するきっかけとなる。そういった相互作用が、個人が持つ知識（＝個人知）から組織の知識（＝組織知）への変換を促進する。

2.2.3 暗黙知と形式知

「知識」には暗黙知と形式知の2つがある。次に両者について述べる。

(1) 暗黙知

暗黙知は、「知識」のうち、勘や直感、個人的洞察、経験に基づくノウハウのことで、言語・数式・図表で表現できない主観的・身体的な知のことをいう。非常に個人的なもので形式化しにくいので、他人に伝達して共有することは難しい。

厳密に言うと、暗黙知は二つの側面を持っている。

一つは、技術的側面で、「ノウハウ」という言葉で捉えられる、はっきりとはこれだと示すことが難しい技能や技巧などが含まれる。例えば、長年の経験を持つ熟練職人は、指先に豊かな技能を蓄えている。しかし、職人が自分の持っている「知」の背景にあたる科学技術的原理をはっきり説明できないことが多い。

もう一つは、認知的側面である。これに含まれるのは、メンタル・モデル、思い、知覚などと呼ばれるもので、無意識に属し、表面に出ることはほとんどない。

(2) 形式知

形式知は、「知識」のうち、言葉や文章、数式、図表などによって表出することが可能な客観的・理性的な知のことをいう。したがって、厳密なデータ、科学方程式、明示化された手続き、普遍的原則などの形でたやすく伝達・共有することができる。

2.2.4 知識変換

さらに、暗黙知と形式知は完全に別々なものではなく、相互補完的なものである。人間の創造的活動において、両者は相互に作用し合い、互いに成り変わる。このような相互循環を「知識変換」と呼ぶ。知識はこの「知識変換」の社会的相互作用を通じて創造されるとされ、具体的には、「共同化」「表出化」「連結化」「内面化」という4つの知識変換モードを通じて、組織の知識が創造される。

次に、この4つの知識変換モードについて述べる。

(1) 共同化(Socialization)

共同化は、経験を共有することによって、メンタル・モデルや技能などの暗黙知を創造するプロセスである。人は言葉を使わずに、他人の持つ暗黙知を獲得することができる。修行中の弟子がその師から、言葉によらず、観察、模倣、練習によって技能を学ぶのはその一例である。OJTも基本的に同じ原理を使う。

(2) 表出化(Externalization)

表出化は、暗黙知を明確なコンセプトに表すプロセスである。表出化は、暗黙知が、コンセプト、仮説、モデルなどの形をとりながら、次第に形式知として明示的になっていくという点で、知識を創造するプロセスの真髄である。表出化は、対話すなわち共同思考によって引き起こされる。熟練技術者が持っている暗黙的な技術をマニュアルに落とし込むときなどが該当する。

(3) 連結化(Combination)

連結化は、コンセプトを組み合わせることで一つの知識体系を創り出すプロセスである。連結化は、異なった形式知を組み合わせることで新たな形式知を創り出す。私たちは、書類、会議、電話などを通じて、知識を交換しながら組み合わせる。コンピュータ上のデータベースなどのように既存の形式知を整理・分類して組みかえることによって新しい知識を生み出すこともできる。学校における教育・訓練などが、この連結化である。ビジネスにおいては、複数の部門で表出化された知識を組み合わせることで、新たな製品を開発するときなどが該当する。

(4) 内面化(Internalization)

内面化は、形式知を暗黙知へ変換するプロセスである。共同化、表出化、連結化を通じて得られた形式知を個人が行動、経験を通じて、洗練し、メンタルモデルや技能的ノウハウという形で新たな暗黙知として獲得するプロセスである。これは、頭で理解した知識を本当の意味で身につけるという面において重要なステップである。

図2はこれらの知識変換モードの関係を示す図で、4つの変換モードの頭文字をとって、「SECIモデル」と呼ばれる。

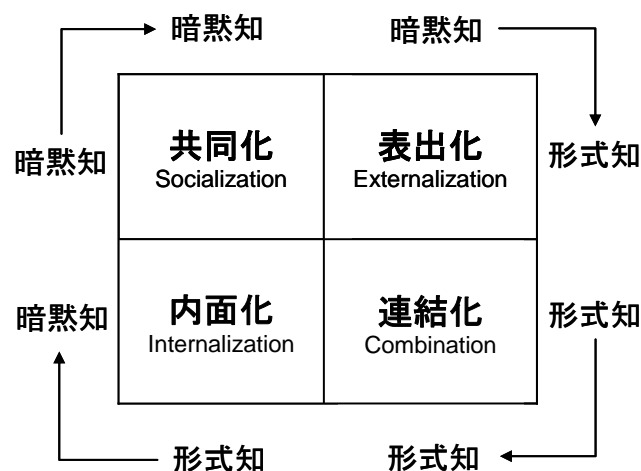


図2 SECIモデル

SECI モデルは、ナレッジマネジメントとして組織の知識創造のための知識変換のモデルであり、知識の共有に留まらず、組織としての知識の創造に踏み込んだ知識経営を提唱している。この4つの知識変換のモードは一巡して終わるものではなく、スパイラル的に繰り返され、その中で各知識変換モードにおいて、知識が増幅され創造されていく。また、これらの相互変換がなされるところを「場」と称し、このSECIモデルが有効に機能するために適切な「場」を設けることが重要である[7]。

本論文は知的熟練の次世代への継承の支援を提案するものであり、知的創造の支援を意図するものではない。従って、本論文の提案はSECIモデルに基づくものではないが、知的熟練の次世代への継承のために、熟練者は自らの知的熟練を表出化・連結化し、それを、新人が内面化することを支援するものである。熟練者は自らの知的熟練を新人が獲得できるようにするため、表出化することが必要であり、また、知的熟練においては、様々な局面に応じて知識を活用できるよう、体系化ないし構造化、すなわち、連結化も行うことも必要である。その上で、新人が表出化・連結化された知的熟練を内面化する。

業務知識・ノウハウの表出化・連結化の最も端的な例としては、体系化された業務マニュアルがあげられる。知的熟練において幅広く深いOJTが重視されるということは、即ち、知的熟練すべてを体系化されたマニュアルとして表出化・連結化することが困難であること、また、それがなされたとしても膨大なマニュアルをOJT抜きに内面化することの困難さを意味している。土方らは、2人の熟練者がSECIモデルに基づき暗黙知を効率的に表出化・連結化できる「場」をシステムにより提供することを提案している[8]。

本論文は、最低限の業務マニュアルを前提にしつつ、それを補うものとして、SECIモデルに基づき、熟練者による知的熟練の表出化・連結化に留まらず、新人による内面化も支援する「場」を提案するものである。

2.2.5 ナレッジマネジメント・ツール

SECIモデルが広く社会に受け入れられる中で、SECIモデルを支援するITツールとして多くのナレッジマネジメント・ツールが開発された。ただし、それらは、KnowledgeMeister[9]のように、ドキュメント管理を核とする場合が多い。一般的には、通常の電子ドキュメントに限定されずに、Webコンテンツ、業務システムの情報、メール情報など幅広い情報を知識データベースとして一括登録管理する機能を提供するとともに、必要な時に容易にアクセスできるように、全文検索などの機能が提供される。ただし、あくまで対象となる知識はすでにドキュメントなどに表出化されている知識が中心であり、一元的に管理し検索できるようにすることにより連結化をサポートするに止まる。

また、所謂企業情報ポータルが多く企業において導入され、コンテンツの充実が進む中

で、INSUITE [10]のように企業情報ポータル向けのスイート製品の中にも、ナレッジマネジメント向け機能を謳うものもある。その場合は、むしろ、電子会議室、Q&A システムのような組織横断的なコミュニケーション活性化による暗黙知の表出化・共有化支援機能が中心となるが、ただし、その場合も、ドキュメント、Web コンテンツ、業務システムなどの幅広い情報にアクセスできる機能が前提となっていて、目的に応じポータルに集約した情報以外は全文検索などでのアクセスが中心となる。

どちらにも共通な特徴としては、すでに表出化されている知識はそのままの形式で登録・管理し、また、電子会議室などにより新たに表出化された知識もそのまま登録される。連結化の支援という観点から見ると、一括管理されたものに対する全文検索などに限定され、暗黙知の体系だった表出化、表出化された知識の体系化・構造化という意味での連結化の支援の視点は乏しい。

これらのナレッジマネジメント・ツールは、知的熟練を必要とする業務に対しても有効であるが、それはあくまで熟練者に対してである。なぜなら、熟練者といえども、必要なすべての知識を記憶していることはあり得ず、過去の技術文書などの参照が必要となることはしばしばある。これらのツールは、熟練者が必要な技術文書などを発掘し、さらに、その文書などに対し新たな知見を付加していくには有効である。ただし、利用者が全文検索などにより必要な技術文書などを検索できるためには、必要な知識を特定できていて、また、検索した技術文書を自らの状況に当てはめて活用できる必要があり、そのためには利用者はすでにかかなりの知的熟練を獲得していることが前提となるからである。

そもそもこれまで OJT により継承されてきた知的熟練は、技術文書のような形で表出化されていないことが問題であり、知的熟練の熟練者から新人への継承という観点で見た場合、まず、知的熟練そのものを体系的に表出化することが必要である。また、不確実性や変化への対応が重視される知的熟練の継承には、単に直接的に必要な知識を提供するだけでなく、それを様々な状況において活用できるよう、関連する知識を含め、幅広く体系的に提供することが必要であり、知的熟練の表出化においては、単に技術文書といった形式ではなく、体系化・構造化することが重要である。その意味で、既存のナレッジマネジメント・ツールは知的熟練の継承という目的には十分とは言えない。

2.2.6 ナレッジマネジメント・ツールの導入事例

前節で述べたナレッジマネジメント・ツールの典型的な導入事例として、中部電力株式会社発電本部火力部におけるナレッジマネジメント支援システム導入を挙げる。このシステムは火力部門全体での運転保守にかかわるノウハウの蓄積と活用により、運転保守業務の効率化と信頼性向上をめざして導入されたものである。

このシステムでは、現場メンバーがパソコンで手軽に文書を検索できる「文書管理システム」と、現場の暗黙知を気づきや Q&A という形で形式知化し共有する「気づきメモシス

テム」から構成されており、前節にあげた全文検索と暗黙知の表出化・共有化を統合的にサポートするものである。

このシステムの導入により、次のような効果をもたらした。

- (1) 広く深く、かつ最新の知識を現場メンバーが共有・活用することで新技術、不慣れな業務も確実にこなすことが可能になった。
- (2) 組織や地理的距離を越えた部門全体でノウハウ共有・蓄積・活用することで、運転保守業務のスピードアップ、効率化が可能になった。
- (3) 技術文書に対する改善点や疑問点を提案・共有できる場を作ることで、リアルタイムに最新ノウハウを共有、技術文書の継続的な改善が可能になった。

ただし、これらは既存文書、気づきメモ、Q&A をそのままの形で提供するものであり、不確実性や変化への対応が求められる知的熟練のために、それらを様々な状況において活用できるよう体系化するという視点はなく、このシステムを活用する現場メンバーはすでにかかなりの知的熟練を獲得していることが求められる典型的な事例である。

2.2.7 ナレッジマネジメント推進上の課題

本節においては、「知識マネジメント」[11]に従ってナレッジマネジメント推進上の課題について述べる。

まず、ナレッジマネジメント導入前の課題としては、導入を推進する部署や担当者がいない、運用管理の手間やコストがかかる、経営トップ層が重要性を理解していないなど、経営トップの無理解と人的リソースの問題がある。

ナレッジマネジメントに取り組み、実際に成果を上げている企業には、スカンディア AFS 社 (Skandia Assurance and Financial Service) のように、トップマネジメント、企業組織において高い地位にいる者が強力なリーダーシップでナレッジマネジメントを推進しているという特徴がある。また、実際に業務を担当する人々、現場の人々の協力も重要となってくる。今まで形式化されてなかった知識を共有するために仕事量が増えてしまう人、自分で努力して得た知識を他人に提供するために作業しなくてはならないことに対し不合理な感情を抱いてしまう人などが出てくる可能性がある。その様な人々に対して、知識の形式化を奨励する報奨制度など、現場の人々自身がナレッジマネジメントにメリットを感じる、知識を提供することに価値観を見出すようにしていかないといけない。

次にナレッジマネジメントの導入後の課題としては、課題や目的をはっきりさせないままナレッジマネジメントを導入したり、企業内の現状を把握せず現状にあわないシステムを導入したりしたために、導入したナレッジマネジメントが十分に機能している場合があることが挙げられる。

特に業務経験の浅い人にとっては、

- ・業務を遂行する上で、いつ蓄積されたどの知識を活用すると有効なのか分からない
- ・必要な知識を見つけても、背景が分からず、活用方法が分からない
- ・過去の事例を見ても、その経緯が分からず、どのように業務を遂行してよいか分からない

という課題があり、業務経験を問わずに全員が活用できる仕組みを実現することが、ナレッジマネジメントの成功の鍵となっている。

これらの課題を克服していくためには、ユーザにとって操作が簡単で、必要な機能が厳選されたシステムであること、ナレッジマネジメントの機能が文書やデータを蓄積し検索するだけでなく、組織内でのコミュニケーションを生み、暗黙知の形式知化による新たな知識を創造し、業務と効率を向上させることが重要である。また、蓄積された知識に対して、なぜその知識が妥当かなどの理由やその知識が生まれた経緯や背景、どう使うと有効かなどの活用情報を関連付けることも重要である。現在のナレッジマネジメントは、従来の知識共有に留まらず、業務経験の浅い人でも知識を有効に活用できるという知識継承の実現を目指すものになってきている。

また、このような課題を解決する上で、目的を明確化するために、ある程度対象業務を絞り、その業務に特質に応じ、その目標とする成果とそのためのコストを明確にすることも重要である。例えば、Hansenらは、知識共有の二つの戦略として、形式知化しデータベース化することにより共有化を目指す「コード化」(Codification)と個人の対話により暗黙知を共有しようとする「個人化」(Personalization)を挙げ、対象業務に応じて、一方に重点を置き、一方を補完的に扱うべきであるとしている[12]。ただし、2つの戦略の補完関係が必ずしも明確ではなく、また、技術・技能の継承という観点から見ると、特に「コード化」に基づく形式知の共有と技術・技能の継承との間にはギャップがある。

2.2.8 ナレッジマネジメントと情報技術

以上のようなナレッジマネジメント・ツールの現状、ナレッジマネジメント推進上の課題を踏まえ、改めて、ナレッジマネジメントと情報技術の関わりについて考察する。

「知識マネジメント」[11]、「知的グループウェアによるナレッジマネジメント」[13]では、ナレッジマネジメントを支援する技術として次を挙げている。

(1) エキスパートシステム

1980年代に、専門家が持つ知識を引き出してルールとするエキスパートシステム(Expert System)が多数開発されたが、専門家から知識を抽出してルールを更新することが困難であり、現在では活用される領域は限られている。

(2) 制約ベースシステム

特定の領域に絞った知識を扱う組織では、エキスパートシステムの他に制約ベースシステム(Constrain-Based System)が有益である。多数のオブジェクト間の複雑な制約によって意思決定が行われる場合に適している。エキスパートシステムよりも修正が容易である。

(3) 事例ベース推論

事例ベース推論 (Case-Based Reasoning) は、ある問題に関する一連の事例から現在の問題に良く似た事例を検索し、知識を抽出する技術である。この技術は、企画、デザイン、法的推論などに応用できるとされていたが、広く受け入れられた商品はほとんどない。しかし、類似事例を検索することにより必要な知識を発見するまでの時間が比較的短いため、顧客サービスのヘルプデスクなど、即答が求められる場面に多く利用されている。

(4) 知識リポジトリ

知識リポジトリは、ナレッジマネジメントで最もよく用いられる技術である。多くの場合、文書の形で知識を蓄える。インターネット上の World Wide Web (WWW) も知識リポジトリの一種と言え、これらを用いたナレッジマネジメントの実践例も多い。しかし、リポジトリが巨大になると、提供されている知識から必要な知識を取り出すことが困難になり、情報の埋没化が起きる。そのため、文書処理技術や新しい情報共有支援技術が必要となった。

(5) データマイニング

膨大なデータから長期的な分析を行って、知識を抽出する方法としてデータマイニングがある。データマイニングは知識発見プロセスの最初の部分で、さまざまなデータ処理をしたうえで、機械学習や統計手法を使って意味あるパターンやルールを見出す。しかし、データの構造化や解釈などの高度な前処理が必要であり、人間の介入が不要になるわけではない。

また、データマイニングにおける問題は、法則を見つけ出すのはユーザである点である。データマイニングツールの役割は、一定のパターンやルールなどの仮説を導き出すことであり、そこから意味を見出し、その仮説を検証できるかどうかは、統計解析や業務に精通したユーザ自身のスキルにかかっている。データマイニングという手法は、マーケティングや医療データベースで利用され、多くの成功事例が報告されている。

(6) グループウェア

企業内 LAN を活用して情報共有やコミュニケーションの効率化をはかり、グループによる協調作業、さらに知的創造活動を支援するためのソフトウェアの総称である。

主な機能として、

-
- a) グループ内のメンバー間及び外部とのコミュニケーションを円滑化する電子メール
 - b) メンバー間で打ち合わせや議論を行なうための電子会議室
 - c) メンバー間のリアルタイムな打ち合わせに利用されるテレビ会議
 - d) グループ全体に広報を行なう電子掲示板
 - e) メンバー間で共有可能な予定表管理機能
 - f) アイデアやノウハウなどのデータベース化による共有
 - g) 複数のメンバーで回覧される文書を電子化し流通させるワークフロー管理

などがある。

これらの機能を個別にではなく、統合した形で提供するのがグループウェアの特徴である。例えば、会議へ参加してほしい人にメールを出し、その人が承認すればその人の予定表管理システムに自動的にそのスケジュールが反映されるなどといった機能がある。

グループウェアは、うまく活用すれば大幅な業務改善が図れるが、そのために仕事の手順を再構築する必要があり、業務改革の意識がないまま導入すると、あまり活用できないまま終わってしまう可能性がある。

Thomas H. Davenport らは、知識の創造や移転などに関して、情報技術の活用に関して、組織の文化的側面などを踏まえつつ、実践的に論じている[14]。その中で、複雑な暗黙知を形式化することには多くの労力を要すること、また、労力を費やして暗黙知を形式化しても直接対面による信頼関係の構築なしには共有がなされない場合があることなどを指摘しつつも、知識の共有のために形式化は必要であり、それを支える情報技術として構造化された明示的な知識の貯蔵庫が重要であるとしている。そして、そこからの確に知識を発掘する意味的検索を実現する上でソーラスの整備の重要性を指摘している。

また、エキスパートシステムが有効であるための条件として、

- (1) 特化されたテクニカルな問題領域であること
- (2) 知識の形式化が容易であること
- (3) 知識が安定していること

をあげている。

Weber らは、ナレッジマネジメントにおいて業務ルールを表出化することは、エキスパートシステムにおける知識獲得と同等であることを指摘し、ナレッジマネジメントにおけるエキスパートシステムを含む知識処理技術の活用を説いている[15]。

先の土方らのシステムにより提供される「場」[8]は、2人の熟練者の暗黙知に対しシステムが事例から帰納学習した知識を含めて不整合を提示することにより、整合の取れた知識の連結化を効率的に実現できることを示したものである。そのためにエキスパートシステムにおける知識獲得の手法が有効であるとして、2人の熟練者が暗黙知を if then 形式で表出化することが前提となっている。

本論文においても、技術・技能の継承のために知的熟練の表出化が求められているとい

う時代背景を前提に、ルールベース・システムを、熟練者に対する表出化の支援・動機付け、新人に対する内面化支援のためのものとしての活用を考える。

2.3 知識ベース

2.3.1 知識ベースについて

「知的知識ベースシステム」[16]によれば、知識ベースは、ナレッジマネジメントのための特殊なデータベースである。それは知識の検索を可能とし、知識を組織化し、知識をコンピュータ上に集合させたものである。

知識ベースとデータベースはいずれも情報処理と呼ばれるコンピュータ利用システム技術の中にあって、非常に近い関係にある技術である。それらはどちらも、コンピュータの記憶能力を積極的に利用する技術であること、物事を表現する形式が従来のコンピュータ技術で用いられてきたプログラム言語（あるいは手続き列によって物事を表現することから手続き型と呼ばれる言語）によるものとは全く異なる構造のものであることによる。

しかし、データを集めて蓄えたものをデータベースと呼ぶのに対して、知識ベースは、人間が知的に問題を解決している背景で重要な役割を果たしている知識を表現し、蓄えたものである。つまり、このような知識を一部でも形式化し、知識ベースとしてコンピュータで処理できるようになれば、コンピュータの問題解決能力は増大する。

知識ベースにおける知識は、自然法則やビジネス上の決めごと、生産技術や機器の制御方式などに関するノウハウなど、人間が経験的に、または教えられて知ったものを指す。また、この知識は対象のタスク領域に関する専門家から獲得していくが、その知識には、一般に成立する正しい知識（事実知識）と、常に成立するとは限らないが、専門家の経験上多くの場合に成り立つ知識（経験知識あるいはヒューリスティックス）の2つのタイプがある。

知識ベース構築にするにあたって、知識の表現の仕方も重要になる。これは、タスク領域や格納すべき知識のタイプを考慮して決めるが、代表的な知識表現としてはルール、フレーム、意味ネットワーク、述語論理表現などが挙げられる。次にこれらを説明する。

(1) ルール

知識を if A then B のような形式で記述する。A を条件部、B を結論部と呼び、A にはそのルールが適用される条件を、B にはこのルールを適用した後どうなるかを書く。特徴は、単純な表現形式であるため、コンピュータに詳しくない問題領域の専門家にとって理解しやすく、また個々のルールを独立に解釈できるので、ルールの追加、削除が容易である。

(2) フレーム

スロットの集合で構成され、そのスロットにはオブジェクト（対象）の特性を“属性－値”の組で格納する。特徴として、“属性－値”を連鎖させることにより、記述の階層的表現が可能である。また、各フレームには、スロットの値に関する詳細な制約などを指定できるだけでなく、スロット値そのものに手続きを記述でき、表現能力が高い。

(3) 意味ネットワーク

オブジェクトをノード、オブジェクト間の関係をアークとして、オブジェクト間の関係を表現する。特徴としては、直観性に優れ、知識を理解したり、書き加えたりする作業が簡単である。また、あるオブジェクトに関する知識が、対応するノードから順にネットワークをたどるだけで容易に得られる。

(4) 述語論理

真偽の定まる文(p, q, \dots)を基本単位にし、論理結合子($\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$)によって結合された複合文の真偽を与えるものが命題論理であるが、述語論理は命題論理の1つの拡張形で、オブジェクト、および、オブジェクトの特徴および複数のオブジェクト間の関係を表す述語を基本要素とし、それらを組み合わせた原子文を基本単位とする。例えば、オブジェクト $p(X)$ は、 X は p であることを表す原子文で、 $p(X, Y)$ は、 X と Y の p という関係にあることを表す原子文である。これらの原子文は、オブジェクトや述語を実世界のどのようなものに解釈するかに応じて、真偽が決まる。述語論理では、この原子文を論理結合子($\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$)によって結合した複合文を扱う。また、オブジェクトや述語は変数として扱うことも可能であり、それは、全称記号や存在記号により量化され、複合文となる。複合文の真偽も、それを構成する原子文の真偽に基づき決定される。この論理式は、先に述べた3つの知識表現とは違い、事実をノードの名前やオブジェクトの属性の値を表現するのではなく、事実の値を真か偽で表現し、推論システムを実現できるのが特徴である。特に1階述語論理の場合は、変数として扱えるものはオブジェクトに限定され、それにより恒真式や充足不能式が推論により判別される完全性が成り立つが、充足可能式までを含めた判別を行う決定可能性は成り立たない。そのため、例えば、**Prolog** においては、決定可能性を担保するために、ホーン節のみを扱うといった制限が入る。ただし、ホーン節の否定はホーン節にならないため、**Prolog** では論理的な意味での否定は扱えず、閉世界仮説の下で記述されていない文はすべて偽とされる。

(5) 記述論理

記述論理[17]は、1階述語論理において、任意の文に対し、与えられた文の集合から伴意されるか否かについての決定可能性を担保するために、述語を、概念を表す単項述語と概念間の有向関係（属性）を表す2項述語に限定したものである。記述論理の特徴は、ホー

ン節とは異なり，概念の否定を含めても，決定可能性が担保されることにある．また，記述論理は，概念とその概念に関する属性に関する制約により，新たな概念を直感的にも分かりやすく，また，柔軟に定義できる．哺乳類とその食料という属性から草食哺乳類という新たな概念を定義する模式的な例を図3に示す．そのため，セマンティック Web でのオントロジー記述言語である OWL においても採用されている．ただし，ルールなどの制約式の記述力は高くない．

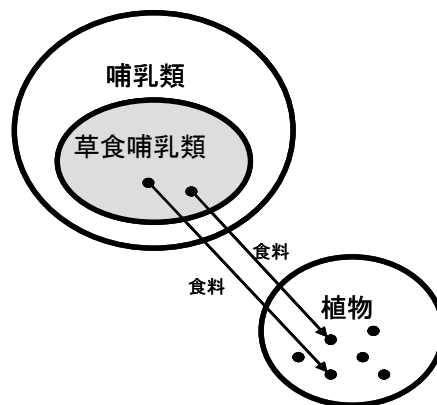


図3 属性制約による新たな概念の定義

知識ベースは，次に述べるコンピュータが解釈可能な知識ベースと，コンピュータは解釈出来ないが，人間が解釈可能な知識ベースの2種類に大きくは分けられる．

(1) コンピュータが解釈可能な知識ベース

コンピュータが解釈可能な形式で知識を格納する．通常，それら知識に対して自動推論を目的としている．知識は論理的に一貫した方法で規則という形式で記述される．個々の知識を論理積，論理和，論理包含，否定といった論理演算子を使って結合する．

(2) 人間が解釈可能な知識ベース

特に訓練などの目的で，人間が扱える形式で知識を集積したもの．特定の組織についての知識（トラブルシューティング，論文，白書，マニュアルなど）の集積である．その利点は，既知の解決法の中からその分野に不慣れな人が現在の問題の解決策を見つける手段を提供することである．

2.3.2 Prolog

Prolog (Programming in Logic の略) [18] は，非手続き型プログラミング言語の一つで，論理型言語に分類される．1972 年ごろにフランスの Alain Colmerauer によって考案された．プログラムは一階述語論理に基づいた文として記述され，問い合わせの文に対し，処

理系がそれらにパターンマッチング（ユニフィケーション）を施しながら、問い合わせの文を真とするオブジェクトを再帰的手続きによって探索する。人工知能における、トップ・ダウン式の問題解決と相性が良いために、人工知能研究とエキスパートシステムの実現のための主要言語として広く採用された。Prolog のもととなる演繹手法は導出と呼ばれ、自動定理証明の研究において Prolog 開発以前よりよく知られていた。Prolog は、導出において節を頭部が一つの命題からのみなるホーン節に限定したものととらえる事ができる。

Prolog のプログラムは事実・規則・質問の3つの要素から成る。

(1) **事実**：オブジェクトとその関係についていくつかの事実を宣言すること。

「taro likes hanako (太郎は花子を好きである)」という文を考えると、主語が「taro」、目的語が「hanako」、述語が「likes」になる。これは「taro」という事物と「hanako」という事物の関係を表現している。Prolog では、述語 likes に注目して、オブジェクト(taro, hanako)の関係を *likes(taro, hanako)* と表す。すべての関係とオブジェクトの名前は小文字で始め、最初に関係の名前が置かれ、次に括弧で囲まれた中にコンマで区切ってオブジェクトを書き、事実の最後には '.' (ピリオド)を置く。事実は必ずしも関係を表すものではなく、事物の性質を表すものでも構わない。Prolog では事実の集まりを「データベース (database)」と呼ぶ。

(2) **規則**：オブジェクトとその関係についての規則を定義すること。

「ある事実が、他のいくつかの事実に依存する」ことを表すために規則を用いる。

「太郎は花が好きな女性を好きだ」という規則は、

likes(taro, X) :- likes(X, flowers), female(X). と表現できる。

さらに「太郎は花と子供が好きな女性を好きだ」という規則もあれば、

likes(taro, X) :- likes(X, flowers), likes(X, children), female(X). と追加できる。

(3) **質問**：オブジェクトとその関係について質問すること。

質問は ?- という記号を前につけて表す。Prolog は質問に適合する事実があれば "yes" と回答し、なければ "no" と回答する。

Prolog は、もう一つの主要な AI 言語である Lisp と同様、リストを主要なデータ構造として用いるという重要な特徴がある。Lisp も Prolog も、Pascal のような汎用言語とは異なり、リストをデータ型としてサポートしている。そしてリストに対し、言語の機能として高レベルの操作機能を組み込みで備えている。それに比べ他の言語では、プログラマは、リストなどの構造を低レベルの構成要素から作り上げなければならない、また入力や出力のための手続きと同様に、メモリ管理のための手続きを書かねばならない。そのどれもが Lisp や Prolog では不要である。直接リストを表現する記法があり、プログラマがプログラム開

発においてデータ構造を調べることを容易にしている．これは，プロトタイプとなるシステムを構築するような場合に特に重要な特徴である．

Lisp と Prolog を比較すると，Lisp では計算に関する「どうやって」を規定しなければならないのに対して，Prolog では「何を」の側面，つまり「どうやって」を決定するための計算課題を規定するだけでよい．この方法が卓越しているのは，課題を達成するアルゴリズムの細部にわたって，プログラマがあれこれと思い煩う必要がないためである．その代わり，プログラマはその課題の正確な仕様規定だけに専念すればよい．

2.3.3 エキスパートシステム

エキスパートシステム[19, 20]は，1970 年代に人工知能の研究者によって開発され，1980 年代にわたって商業的に適用されたコンピュータプログラム的一种である．基本的に，特定の分野の問題についてルール群から構成されるプログラムであり，そのルールはその分野の専門家が提供する．問題の分析結果を提供するだけでなく，設計によっては利用者の行動を正しく導く指針を与えることもできる．

エキスパートシステムはユーザからの入力に対し，知識ベースに格納された専門家の知識を用いて，推論エンジンで推論を行った結果を出力する．ユーザから入力された，解くべき問題に関する知識や推論途中の経過は，作業領域に格納される．エキスパートシステムは，知識と推論が分離しているので，単に専門知識を追加することでシステムを変更することができる．従来の知識と推論部分が混ざり合っただけの一体となっている情報処理システムと違い，システムに変更を加える時に混ざりあった手順から必要な部分を見極めて改める必要がないのが特徴である．(図 4 参照)

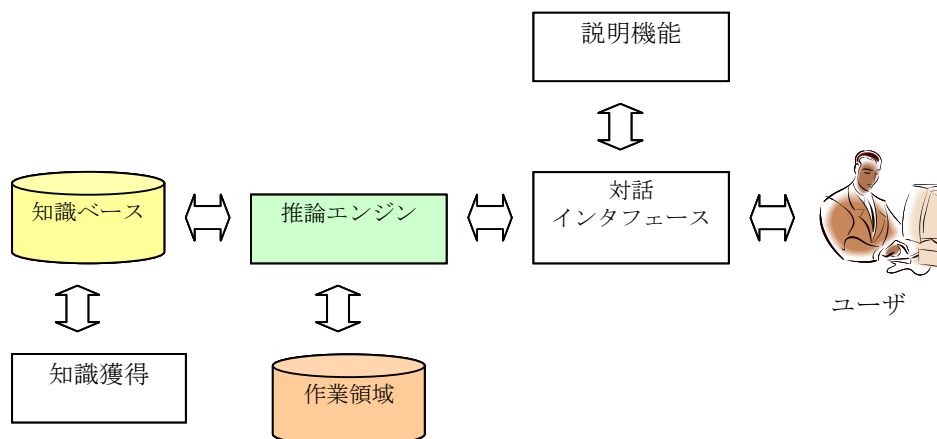


図 4 エキスパートシステムの構造

推論エンジンは、知識ベースから答えを導き出す仕組みである。エキスパートシステムの頭脳であり、知識ベース内の情報を元に推論を実行する方法を提供し、結論を導く。推論方法は知識ベース中の知識表現に依存して決定される。

推論エンジンには2つの主要な作業があり、1つは存在する事実とルールを調べ、可能ときには新しい推論結果を追加する作業と、もう1つは推論の順序を決定する制御の作業である。

1つ目の作業、推論方法としては三段論法における肯定式がある。この規則は、私たちが日常でよく使う考え方で、Aが真であるとわかっていて、if A then B というルールがあると、Bが真であるといえる。つまり、ルールの条件部が真であると、その結論部も真であるといえる。既知の事実とルールから新しい事実を引き出すことができるこの三段論法は、推論を行う簡単で直観的な方法である。

もう1つの作業、制御方法としては、ルールを適用する向きの違いにより分類され、前向き推論と後ろ向き推論がある。前向き推論は、作業領域であるワーキングメモリの初期状態（存在している事実）から、条件部が成り立つルールを実行し、結論部によりワーキングメモリを更新しながら目標とする状態を目指してルール実行を繰り返す方法である。後ろ向き推論は、反対に目標とする仮定した結論から出発して、それを結論部に持つルールを選択し、さらに、その条件部を結論部に持つルールを選択、というプロセスを繰り返す方法である。

エキスパートシステム構築においては、スパイラルモデル又はプロトタイプモデルと呼ばれる開発モデルが適用される。すべての仕様を明確にしたから設計に入るのではなく、わかる範囲の仕様で試作モデル（プロトタイプ）を短時間で作成し、その評価を通じて仕様を詳細化していくことを繰り返し、システムを開発していく。その手順を次に示す。

手順 1. 専門家へのインタビューなどを通じて、専門家の知識、推論方法を得る。

手順 2. 知識表現、推論方法を決定してプロトタイプシステムを構築する。

手順 3. 専門家にシステムを評価してもらい、知識ベースを修正する。

手順 1. では、専門家が暗黙的に持っている知識を引き出す必要があり、それにはある程度タスク領域の知識を持っていないと十分な知識は引き出せない。手順 3. では、専門家ではない人が知識を入れていったものに対して、間違った解釈で入力されていないか、また知識間の矛盾がなく整合性が取れているかについて、試験運用などを通じ、専門家の協力によりシステムの不備をチェックする。これらの手順 1.~3. を繰り返すことにより、要求される仕様に満足するシステムを作っていく。これらの作業、専門家から知識を得てコンピュータで利用可能な形式にする作業をする人をナレッジエンジニアと呼ぶ。

ここで、エキスパートシステムの例として MYCIN を挙げる。MYCIN は、専門家レベルの能力を持ち、利用者にその推論過程を説明する最初の大きなエキスパートシステムで

あった。Stanford 大学で 1970 年代初めに 5, 6 年の歳月をかけ, Bruce Buchanan と Edward H. Shortliffe によって開発された。MYCIN は, 伝染性の血液疾患を診断し, 詳細な推奨する治療を提供する。具体的には, 500 程度の規則からなる知識ベースを持ち, 推論エンジンは後ろ向き推論であり, 医者に対して, 単純な「はいいいえ」で答える質問や何らかの文章で答える質問をしていき, 最終的に可能性のある細菌名のリストとそれぞれの信頼度, なぜそう推論したかという理由, 推奨する抗生物質を示す。

しかし, MYCIN は実用化という点ではあまり成功しなかった。その理由を次に挙げる。

(1) データ入力環境

対話型のシステムであったため, 多くのデータを対話に従って手入力しなければいけなく, このプロセスには労力と時間はかかってしまった。

(2) コスト

難しい問題を解くために, 当時としては非常に高性能なコンピュータが必要であり, ハードウェア・コストが非常に高く, さらにこのソフトウェアの開発コストも非常に高かった。

(3) 正解率

複雑な医学知識をもれなく体系化することは難しく, MYCIN の診断結果の正答率は 65% と悪いわけではなかったが, その分野の専門医の平均正答率のおよそ 80%には及ばなかった[21]。

(4) 倫理や法律

コンピュータを医療に使って間違った診断を下した場合, 誰が責任を取るのかという問題があった。また, 人間の専門家が MYCIN を受け入れることへの抵抗もあった。

2.3.4 エキスパートシステムの現状

エキスパートシステムは 1980 年代の人工知能ブームをけん引し, 商用のシステムが多く開発されてきたが, 現在は限界が見え下火となっている。

その理由として, 知識獲得のボトルネックがある。タスク領域の専門家から問題解決に必要な知識を獲得するには, インタビューなどを行うナレッジエンジニア自身がその領域にある程度精通している必要がある。専門家自身は明示的・体系的に知識を整理できていくわけではないので, 問題解決に必要な知識を不足なく引き出せるかどうかは, ナレッジエンジニアの能力にかかってくる。さらに, 知識と一口に言っても, 浅い知識, 深い知識, 専門知識に一般常識までと様々なタイプがあるので, これらを取捨選択し, 体系的にシステムに組み込む必要もある。ナレッジエンジニアにとって, これらの作業はコストの高い

作業となってしまう。

また、そもそもコンピュータが解釈できる形で専門家自身が自分の知識を表現できるのかという問題もある。専門家自身が経験を通して得て獲得し、体系化できていない知識をすべて完全に表現することは困難である。ある程度表現できたとしても、知識が正確にはナレッジエンジニアには伝わらないことも考えられる。

もう1つの理由として、システムの保守・運用の問題がある。時が経つにつれて新たな知識を追加・修正する際に、知識ベースを修正する必要が出てきた場合に、開発当時のナレッジエンジニアがいないと知識ベースに含まれる単語の意味などを理解するのが非常に困難な作業となってしまうという課題がある。

一方で、比較的ルールが安定した業務に対しはエキスパートシステムの有効性が確認されて久しく、特に、スケジューリング、レイアウトなどの制約充足問題では、IBM社のILOG CP (Constraint Programming) などの商用ツールもあり、実務における活用事例も多い。また、エキスパートシステムの位置付けは、専門家に代わって問題解決することだけでなく、業務の属人性の排除、新人に対する教育などに広がってきている。そのため、ルールの説明性が求められるようになり、深い知識に基づくエンドユーザ向けの説明機能[22]などが重視されるようになってきている。

2.4 知識モデリング

一般的にエキスパートシステムでは知識の獲得と更新がボトルネックとなるが、その要因の一つとして、エキスパートシステムの開発が専門家からの知識の抽出も含め、実装方式に依存した方法によっていることがあげられる。そのため、所謂知識システムにおいても、通常の情報システムと同様な統合的開発方法論の整備が進められていて、その中の一つにCommonKADS [23]がある。CommonKADSでは、知識システムを単に専門家に代わって推論などを行うものとして狭く捉えることなく、組織の知的生産性を高めるためのものとして広く捉え、通常の情報システムの分析・設計と同様、課題設定、組織モデル、業務モデル（タスクモデル）といったハイレベルな分析から、実装方式とは独立なモデリングにより、ブレークダウンしていく。その中で核となるのが知識モデルであるが、知識モデルは実装方式とは独立の概念レベルのモデルであるが故に、ナレッジマネジメントに対しても知識の形式化・共有などの面で有効である。知識モデルでは、そこで実行されるタスクをタスク知識、そのタスクを実行する上で必要となる推論を推論知識としてモデル化するとともに、その推論において必要となる対象世界の静的構造を領域知識としてモデル化する。タスク知識、推論知識は、その再利用性・メンテナンス性を高めるために、診断、スケジューリングなどの知識システムの代表的なタスク毎に、テンプレート化されている。また、領域知識には、UMLのクラス図に相当するものとして領域スキーマがあるが、クラス間の制約などに関しては、領域スキーマとは別に知識ベースとして実装独立にモデル化

し、メンテナンス性・再利用性を高めているのが領域知識の特徴である。さらに、領域スキーマに関しては、推論知識への適用性を高めるためにより汎用性を意識して体系化したオントロジー（次節を参照のこと）に拡張される。

2.5 オントロジー

2.5.1 オントロジーについて

オントロジー[24]は、元々はギリシャ語で存在を意味する“ontos”と理論を意味する“logos”から17世紀初頭にドイツ哲学者により造られた「存在に関する体系的な理論（存在論）」を意味する哲学用語であり、世の中に存在するすべてのものを体系化することを目指したものであり、その起源はアリストテレスのカテゴリー論に遡る。

情報科学の分野でこのオントロジーという用語が初めて使われたのは、人工知能学者のPatrick Hayesによる1985年の論文「素朴物理学 – 流体のためのオントロジー」(Naïve Physics I: Ontology for Liquids) [25]が最初である。当時、エキスパートシステムは興隆を極めていたが、その一方、想定外の問題に対しては、エキスパートシステムが機能しないことが問題となっていて、そのような状況への対処として、Patrick Hayesは、推論システムの高度化よりも、むしろ、一般の人が世の中の理解のような汎用的な「常識」の整備が必要であることを提唱していて、それをオントロジーと呼び始めた。

その後、「概念化の明示的な規格 (an explicit specification of a conceptualization)」[26]というThomas Gruberの有名な定義がなされている。ここでいう概念化は、対象（世界）を基本となる概念とそれらの間の関係として捉えることを意味していて、この概念化は、具体的には、オントロジー記述言語として当時最もポピュラーであったKIF (2.5.3 (2)参照) による対象世界の記述と、KIFのモデル意味論による解釈を強く意識したものであった。

オントロジーの利用法として、オントロジーで規定された概念およびそれに対する語彙を、知識を表すための共通の語彙として利用するという形態がある。知識をコンピュータに格納する場合、知識の記述に用いられる語彙が統一されていないと、せっかく格納しても、共有し有効に活用することができない。そこで、語彙をオントロジーとしてあらかじめ定義しておき、知識記述の際に利用することで、知識の再利用性を向上させることが可能となる。つまり、オントロジーは、共通語彙を提供する体系化された辞書として有効である。

オントロジーは、次のものからなる。

- (1) 対象世界から切り出し基本的な概念の集合とそれに対する語彙
- (2) それらの概念間のis-a関係
- (3) それらの概念の意味を特徴付ける公理

これらの構成要素について説明する。

(1) 対象世界から切り出した基本的な概念の集合とそれに対する語彙

対象世界のオブジェクトを何らかの共通性に基づきまとめたものである概念の中から、オントロジーは、基本的なものを切り出し、それらに対し語彙を与える。オントロジーにおける概念はオブジェクト指向におけるクラスに近い要素であるが、クラスとは異なり状態や振る舞いを考慮しない。例えば研究室に在籍するメンバー一人ひとりがオブジェクトであるとすると、このオブジェクトを「研究室に在籍する」という共通性でまとめたものが概念となり、「研究室員」という語彙を与えることができる。

(2) それらの概念の is-a 関係

is-a 関係（もしくは kind-of 関係）は、日本語で言えば、汎化-特化関係といえる。言葉のとおり、「A is a B」といえる関係のことである。このとき、Bの方がより抽象的な概念となり、両者の上下関係として、Bが上位概念となり、Aが下位概念となる。より抽象的な概念が上位に位置し、より具体的な概念が下位概念に位置することになる。ただし、概念と実体のインスタンス関係も「A is a B」と言えてしまうので、注意が必要である。オブジェクト指向における継承関係とほぼ同様の意味づけであるが、何を継承するのかは場合による。is-a 関係の具体例を示したものが、以下の図 5 である。

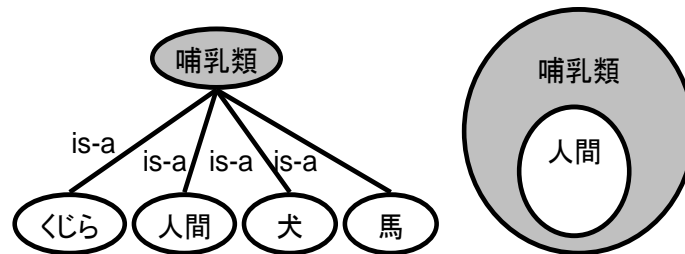


図 5 is-a 関係の例

例えば、「人間 is a 哺乳類」であり、人間と哺乳類の間には is-a 関係が成り立ち、哺乳類が上位概念、人間が下位概念である。概念を集合に付けられたラベルと考えると、哺乳類というラベルを振られたインスタンスの集合のほうが、人間というラベルを振られたインスタンス集合より大きい。つまり哺乳類 \supset 人間である。カバーしているインスタンスの集合が大きい方を小さい方の一般化といい、小さい方を大きい方の特殊化という。

(3) それらの概念の意味を特徴付ける公理

その他、概念間に意味的な制約を与える公理を設けて、その概念に対する解釈の規定することにより概念の意味をより強く特徴付ける場合がある。KIFで記述されたオントロジーであるPSL (Process Specification Language) での公理の例を以下の図6に示す。図6の公理では、アクティビティ (activity) およびサブアクティビティ (subactivity) が概念であり、この公理は「a1がa2のサブアクティビティであるならば、a1もa2もアクティビティである」という意味的な制約を規定している。これは人間の自然な理解からすれば極めて自明な内容であるが、コンピュータによる意味的处理を意識した場合、このような公理も必要になる。オントロジーにおいて、どこまで公理を導入するかは、コンピュータによる意味的处理をどこまで求めるかにより様々である。

6.6.1 Axiom 1

(forall (?a1 ?a2) (implies (subactivity ?a1 ?a2)
(and (activity ?a1) (activity ?a2))))

出所: ISO 18629-12 - Process specification language -Part 12: Outer core

図6 オントロジーにおける公理

2.5.2 オントロジーの分類

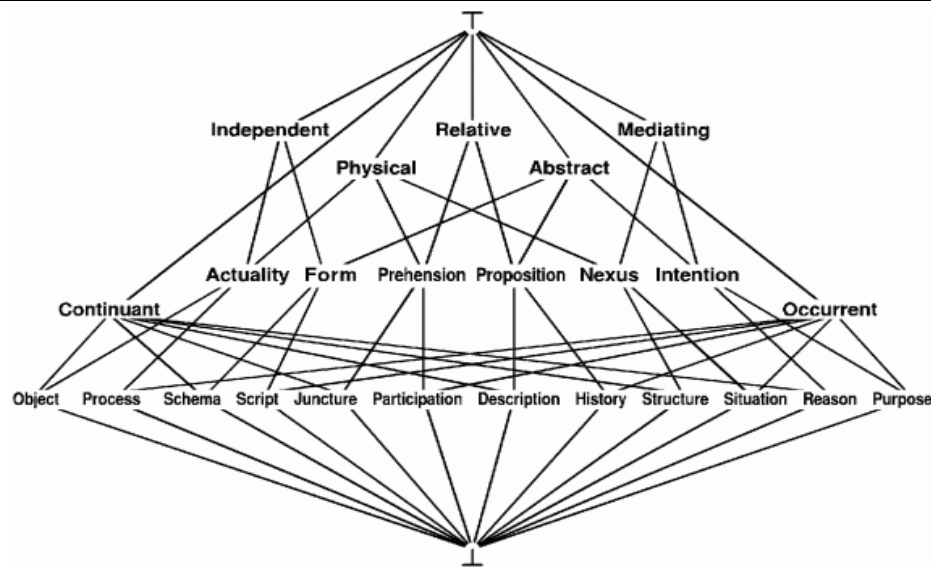
オントロジーは視点によっていくつかの種類に分類することができる。その主なものとして、アップパー・オントロジーとドメイン・オントロジー、ヘビーウェイト・オントロジーとライトウェイト・オントロジーを挙げる。ただし、これらの分類に関しては厳密な定義がある訳ではなく、捉え方に微妙な差がある場合がある。

2.5.2.1 アップパー・オントロジーとドメイン・オントロジー

(1) アップパー・オントロジー

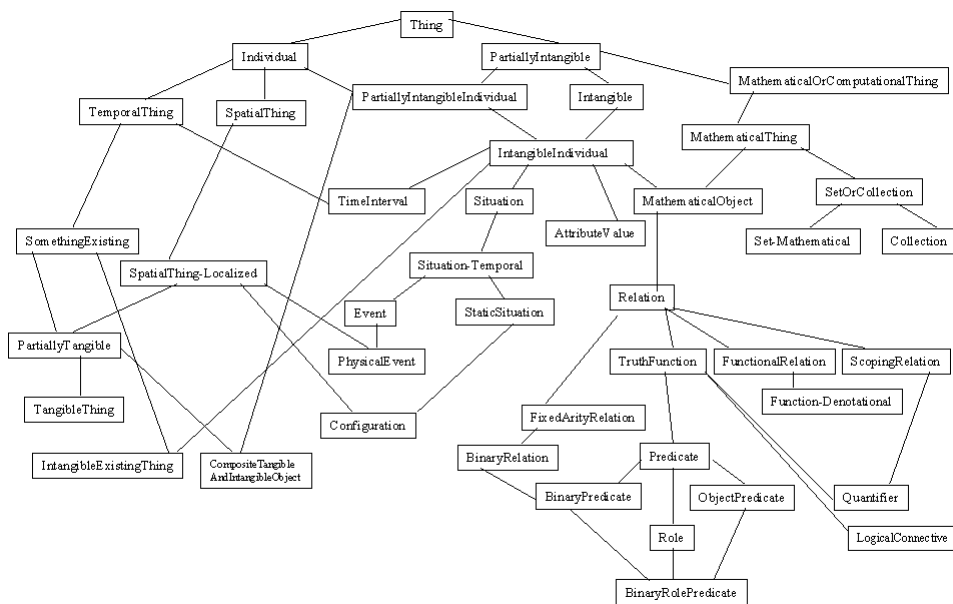
専門領域 (ドメイン) に依存しない汎用的なオントロジーで、世界の最も抽象度の高い汎用的なカテゴリーを含む。アリストテレスのカテゴリー論とも通じるものである。

SUMO (Suggested Upper Merged Ontology) [27], John F. Sowa の Top-Level Categories [28] (図7), CyC の Upper Ontology [29] (図8), BFO (Basic Formal Ontology) [30], YATO (Yet Another Top-level Ontology) [31] など、いくつかのものが提案されているが、哲学的論争も加わり、以下の図9に示すように、最上位のカテゴリーに関しても合意はされていない。



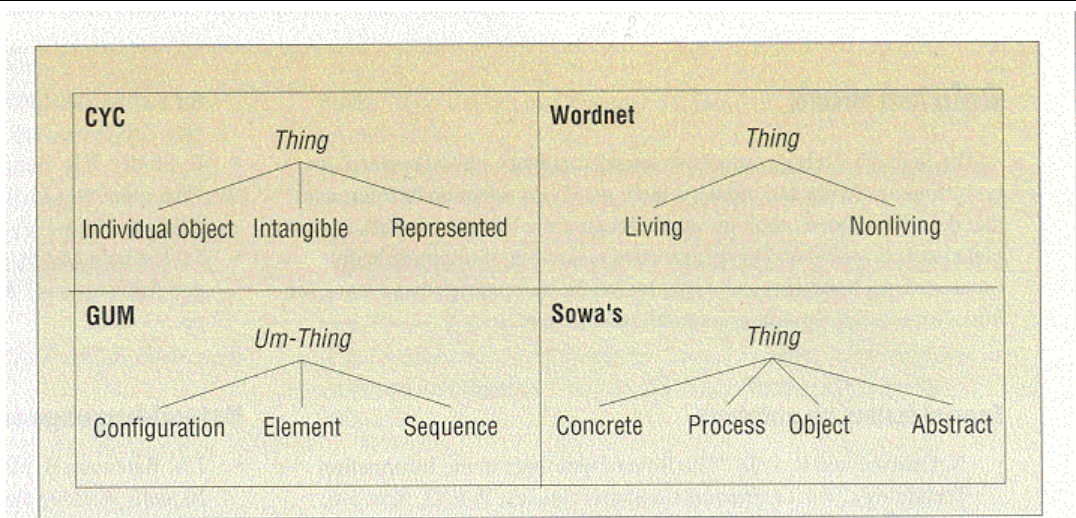
出所 : <http://www.jfsowa.com/ontology/toplevel.htm>

図 7 John F. Sowa の Top-Level Categories



出所 : <http://www.cyc.com/cycdoc/vocab/upperont-diagram.html>

図 8 CyC の Upper Ontology



注: GUM: Generalized Upper Model

出所: Chandrasekaran et al., What Are Ontologies, and Why Do We Need Them?
(IEEE INTELLIGENT SYSTEMS1999)

図 9 アッパー・オントロジーにおけるトップレベル・カテゴリー

(2) ドメイン・オントロジー

ドメイン・オントロジーは、プラント、ビジネス、法律など、特定の専門領域（ドメイン）についての概念を規定したものである。特定のドメインで用いられている専門用語を扱うため、必ずしも専門領域外の人が理解できる必要はないが、対象専門領域内の人が理解し、共用することができることが求められる。

本研究はある専門領域の知的熟練の継承を目的としているため、本研究で扱うオントロジーはドメイン・オントロジーである。ただし、その専門領域の新人でも無理なく理解できるものを目指している。

2.5.2.2 ヘビーウェイト・オントロジーとライトウェイト・オントロジー

(1) ヘビーウェイト・オントロジー

ヘビーウェイト・オントロジーは、コンピュータにより必要な推論ができるよう、意味的な制約の公理などにより、概念の特徴付けを強く行っているオントロジーである。ただし、必ずしも意味の完全な規定ができていない訳ではないことに留意する必要がある。

(2) ライトウェイト・オントロジー

ライトウェイト・オントロジーは、概念間の関係の規定は is-a 関係などを中心とし、コンピュータによる推論を意識した概念の強い特徴付けまでは行っていないオントロジーである。

本研究は人間・組織間の知的熟練の継承を目的としているため、本研究で扱うオントロジーはライトウェイト・オントロジーである。

2.5.3 オントロジー記述言語

オントロジーを記述するための言語としていくつかのものが存在する。記述しようとするオントロジーがコンピュータによる推論を意識してどこまで強い概念の特徴付けを行う必要があるかにより、適切なオントロジー記述言語を選択する必要がある。次に主なオントロジー記述言語の概要を述べる。

(1) RDF, RDF Schema, OWL [32]

所謂セマンティック Web で利用され、最も普及しているオントロジー記述言語である。Web 上でのオントロジーの記述を意識して、XML シンタックスを持つ。2004 年 2 月に第 1 版が W3C の勧告になり、OWL に関しては 2010 年 10 月に第 2 版が勧告された。

RDF は「主語(subject)－述語(predicate)－目的語(object)」の三つ組み（トリプル）で表現する枠組みを規定したものであり、RDFS は RDF の枠組みの上にクラスなどのスキーマ概念を加えたもの、OWL はさらに記述論理を記述する枠組みを加えたものである。それぞれについて説明する。

a) RDF

RDF は “Resource Description Framework” の略であり、情報を、「主語 (subject) － 述語 (predicate) － 目的語 (object)」の三つ組み（トリプル）で表現する枠組みを規定したものである。RDF では述語 (predicate) のことをプロパティとよぶ。RDF は、この三つ組み（トリプル）で表現していれば、原則としてその構文は自由である。ただし、適用範囲の広さなどから XML による表現が推奨されており、RDF の XML 構文は RDF/XML と呼ばれる。RDF のトリプルを RDF/XML で表現するためには、主語、述語、目的語をそれぞれ XML の要素として記述する。また、RDF では、すべてのものはリソースととされ、URI (Uniform Resource Identifiers) を用いて識別される。

b) RDF Schema

RDF Schema は、RDF において様々なリソースを記述するために必要な語彙を定義したものである。他の多くのプログラミング言語や知識表現手法と同じように、クラスによってリソースのグループを表し、また、あるクラスに属するリソースをそのクラスのインスタンスと呼ぶ。RDF Schema では、クラスの階層関係、プロパティの定義域、値域、階層関係を定義することでクラス、プロパティを表現する。

c) OWL

OWLは“Web Ontology Language”の略であり、ウェブのリソースを記述するためのオントロジー記述言語である。RDF Schemaでは、リソースを記述する語彙を比較的シンプルに定義したが、OWLではそこから一步踏み込んで、記述論理の枠組みに沿って定義する。それによって、語彙がより強く特徴付けられ、概念間の関係を精密に表現でき、記述論理に基づく推論が可能となる。

OWLには、目的によって使い分けることが可能な、OWL Lite, OWL DL, OWL Full という三つのサブ言語がある。

・ OWL Lite

クラス階層の構築と基本的な制約条件の記述が主目的である場合に使える。OWLそのものを規定している語彙の内、一部が利用できず、語彙の使い方が制限されているが、その分OWL Liteを処理するアプリケーションは比較的容易に実装が可能である。

・ OWL DL

OWL DLのDLはDescription Logic（記述論理）を表しており、OWLそのものを規定する語彙をすべて利用できる。ただし、記述論理に基づく推論の完全性と決定可能性を担保するために、クラスでありかつインスタンスであるリソースを認めないなど、OWLそのものを規定する語彙の使い方に制限がある。

・ OWL Full

OWL Fullは、OWLの語彙とRDFの柔軟性を最大限利用するための言語である。OWLのなかで最も表現力があり、OWL DLでは認められていなかったクラスでありかつインスタンスであるリソースも認められる。ただし、計算の完全性、決定可能性は保証されない。OWL Fullでは、すでにRDF Schemaなどで表現されている語彙をより詳細に記述したり相互に関係づけたりするために用いることができる。

以上のことから、三つのサブ言語の使い分けとしては、機能を絞り、すばやくコンパクトな開発を目指す場合はOWL Lite、記述論理に基づく推論エンジンを利用可能なオントロジーを構築するためにはOWL DL、クラスとインスタンスの区別を柔軟にするといった実用性を重視する場合はOWL Fullが有効であると考えられる。

(2) KIF (Knowledge Interchange Format) [33]

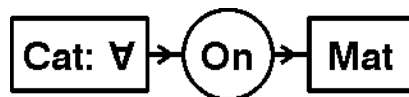
KIFは、必ずしもオントロジーに限らず、より一般的に知識表現の標準交換フォーマットとして開発された1980年代に開発された言語である。

オントロジーの有名な定義を与えている Thomas Gruber の論文[25]においても使用され、初期のオントロジーは KIF で記述されたものが多い。

KIF にはいくつかのバージョンがあるが、最も標準的なものは、等号を含む1階述語論理に定義機能を加えたもので、Lisp の S 式に倣ったシンタックスを持つ。KIF は、標準的には1階述語論理相当であり、そのため、完全性は保証されるが、決定可能性は保証されない。

(3) Conceptual Graph [34]

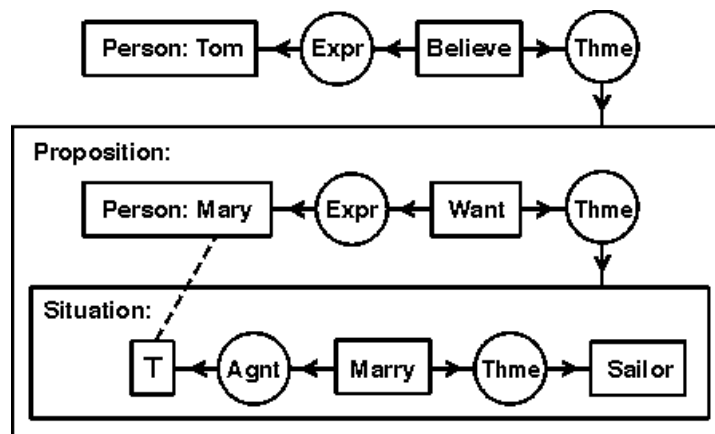
Conceptual Graph は John F. Sowa が提唱する論理学をベースにしたグラフ表記を含むノーテーションである。基本的には1階述語論理と同等で、例えば、“for all x if Cat(x) then exist y On(x, y) and Mat(y)” は図 10 のグラフとして記述される。



出所 : <http://www.jfsowa.com/cg/cgexampw.htm>

図 10 Conceptual Graph の例 1

このグラフ記法が、E/R モデルなどのデータモデル記法との類似していることもあり、実務家からの支持が強い。また、図 11 に示さるような階層的な表記により、「「Mary (T)は Sailor と結婚する(Marry)」ことを Mary は望んでいる(Want)」ということを Tom は信じている(Believe)」といった高階論理式の一部も表現可能である。



出所 : <http://www.jfsowa.com/cg/cgexampw.htm>

図 11 Conceptual Graph の例 2

(4) Common Logic (ISO/IEC 24707) [35]

1990 年代から米国を中心に KIF および Conceptual Graph の統合を中心にオントロジー言語ないし論理学に基づいた言語の標準化の動きがあった。その統合案が ISO/IEC の場に

提案され、2007年9月に国際標準になったものが ISO/IEC 24707 Common Logic である。

Common Logic は、KIF および Conceptual Graph の統合という当初のスコープを超え、より拡張された汎用的な仕様となっている。大きくは、次の3点が特徴である。

a) 解釈条件のみを規定

解釈条件を規定する抽象的な仕様で、その解釈条件を満たせばどのような構文も認められる。

b) Type free と Arity free

記号に関して、個体記号、述語記号、関数記号、変数のシンタックス上の区別はなく、解釈に依存する。また、述語記号、関数記号として解釈された場合に、その引数の数は自由である。

c) 2階論理のシンタックスを1階論理のセマンティクスで

記号は、個体記号、述語記号、関数記号、変数のいずれに解釈される場合も、Hilog [36]と同様、あくまで加算無限の解釈空間上で解釈される。それにより、解釈空間の濃度が連続になることにより発生する困難さが回避される。

2.5.4 オントロジーの例

(1) Cyc [37]

Cyc は Cycorp 社にて開発されている大規模商用オントロジーおよびその統合環境である。“常識”のデータベース化を目指したもので、Patrick Hayes が 1985 年に提唱した当初のオントロジーに非常に近いものである。

Cyc のオントロジーは、CycL という独自の言語で意味的に強く特徴付けられたヘビーウェイト・オントロジーであり、また、図 8 に示したようなアップパー・オントロジーを含むと同時に、広範な“常識”を含むことから、複数のドメイン・オントロジーの集合体とも見なせる。広範な“常識”を意味的に強く特徴付けているが故に、Cyc のオントロジーは、全体としての整合性は成り立たず、「マイクロ理論」と呼ばれる単位でのみ整合性が保証される。

(2) PSL [38]

ISO18629 PSL(Process Specification Language) は離散系プロセスを規定するオントロジーの国際標準である。製造業の離散系プロセスを管理・制御する情報システム間の相互運用性の促進を目的とするもので、そのために、各概念は KIF により厳密に定義されている。製造業の離散系プロセスというドメインを対象としたドメイン・オントロジーかつヘビーウェイト・オントロジーの典型である。

PSL における概念の公理化の例として、図 12 に “before” を挙げる。人間にとってはほとんど自明である “before” に関し、これだけの公理化を行っても、実は、これでは “before”

と“after”の区別が付いていないことはよく指摘されるところである。

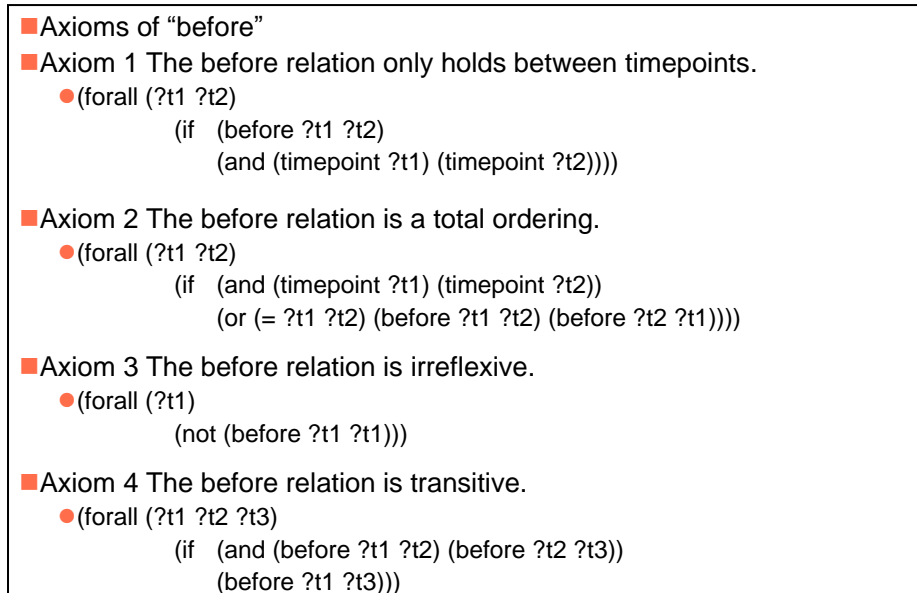


図 12 PSL における“before”の公理化

(3) SKOS [39]

SKOS (Simple Knowledge Organization System) はシソーラスなどの知識スキーマの記述を行うためのオントロジーである。SKOSは2009年8月にW3Cの勧告になっている。SKOSはOWL-DLで記述されているが、各概念の特徴付けは、の図13“skos:broader”の例に見るように、緩やかである。SKOSは知識スキーマをドメインとするドメイン・オントロジーで、ライトウェイト・オントロジーであると言える。

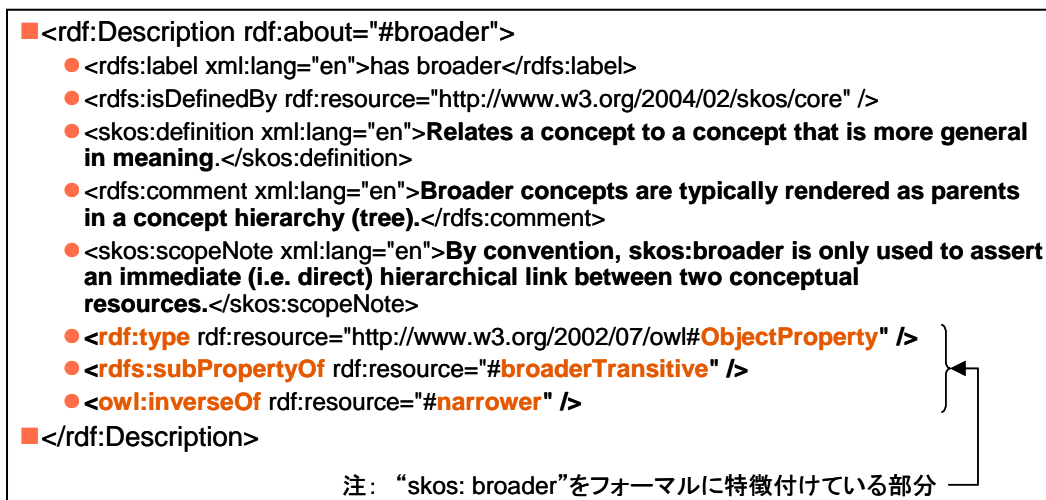


図 13 “skos:broader”の特徴付け

2.5.5 オントロジーの役割

次に、知識に基づく問題解決において、オントロジーが実際どのように役に立つのか、その効用について次に述べる。

(1) 合意を得る手段

オントロジーは知識そのものではなく、いわば対象世界の骨格を明示化したものである。よって、オントロジーはその対象世界における関係者の合意のもと作られるべきであり、逆に、合意を得る上でオントロジーを媒介とすることは有効である。この合意形成のプロセスは、まず、複数の人が合意できるものを探すという合意のもとに、参加者がオントロジーを出し合う。そして、それぞれのオントロジーを分析し、共通点・相違点などを抽出し、抽象度を変えながら合意できる部分を探していく。こうして、合意の成果物としてオントロジーを得る。このプロセスにおいて、他人と合意を得ることは簡単ではない。しかし、深い合意が得られたオントロジーは、概念階層が深くなり、詳細な概念構造に及ぶものが得られる。

(2) 暗黙知の明示化

オントロジーは対象世界を構成する概念に対し語彙を与え、相互の関係を記述することにより、概念の明示化を行うものである。暗黙知を明示化しようとする場合、まず、その暗黙知を構成する概念を明示化することが必要となる。その意味で、オントロジーは暗黙知の明示化する上で重要な役割を果たす。

(3) 再利用と共有

専門家の経験則などは主観的要素が強く、前提が暗黙的なことが多いため、それらの知識を共有したり、再利用したりすることは困難である。しかし、それらの知識を共有し再利用しようとする人々の間で、対象世界を構成する概念の合意の成果物としてのオントロジーが成立していれば、それらの知識をこれらの概念まで還元することにより、共有し再利用することが可能になる。

(4) コンピュータ上での知識の体系化

従来、ある対象世界において知識を体系化したとしてもそれは人間のための体系化であり、コンピュータが理解できるものではなかった。そこで、コンピュータが理解できるオントロジーを用いることで、人間による共有の枠を超え、コンピュータ上での処理が可能となり、体系化された知識を多岐にわたって利用することができるようになる。

(5) 標準化

オントロジーはその性質より、複数の人の間で共有される合意内容である。つまり、ここでは複数の人と合意形成をしており、オントロジーに含まれる概念は共通性が高く、さらには標準化の可能性を含んでいる。

(6) メタモデル的機能

オントロジー構築を通じて得られた、概念の体系や概念間の関係は、対象世界においてモデルを構築しようとする際に、その構築に必要な基本概念とガイドラインを提供する。

(7) 統合的効用

以上、述べてきたオントロジーの効用を眺めてみると、オントロジーがいかにより有用であるかが見てとれる。複数の人々の間の合意形成に役立ち、通常暗黙となっている基本的な概念を明示化し、それにより知識を構成する概念を共有し、また、再利用することができ、それらを標準化して体系化を行うと共に、必要なモデル構築を行う。このモデルは、透明度の高い、共有することができる規範的なモデルとなるのである。オントロジーとはこれらのことを実現する可能性を持っていて、ナレッジマネジメントや XML 文書タグ設計の基礎、およびそれらの有効利用に貢献することができる。

2.5.6 オントロジー構築支援ツールとオントロジー・ポータル

以上のようなオントロジーの役割を考えた場合、広範な知識の広範な共有・再利用を実現していくために、より多くのオントロジーを構築するとともに、それらが、より多くの人々の間で合意され、共有されていくことが重要である。そのために、オントロジー構築支援ツールやオントロジー・ポータルが重要となる。

オントロジー構築を支援するツールとして、最も有名なものに **Protégé** [40]がある。**Protégé**はStanford大学Medical Informaticsに所属するMark Musenらのグループによって開発されているJavaベースのオントロジー構築支援ツールである。オントロジー構築支援ツールとしては最も普及しており、世界中で利用されている。**Protégé**の主な特徴として、次の4つがある。

- ・記述モデルを独自に定義する機能や出力フォーマットを選択する機能
- ・オントロジーの出力形式を任意の形式言語にカスタマイズする機能
- ・インタフェースをカスタマイズする機能
- ・他の応用プログラムを組み込むための強力なプラグイン機能

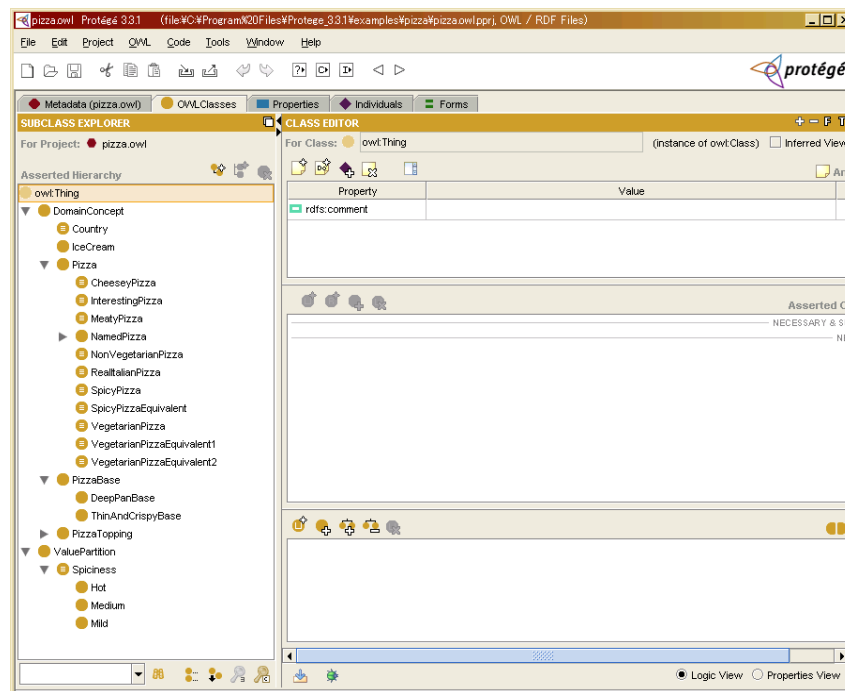


図 14 オントロジー構築ツール Protégé クラス階層画面

Mark Musen らは、協調分散作業でのオントロジーの効率的開発を目指していて、そのために、インターネット上で Web ブラウザによる協調分散作業でのオントロジー構築をサポートする Web Protégé もリリースされつつある。

さらに、Mark Musen らは、オントロジーの共有・再利用を促進するために、オントロジー・ポータルとして BioPortal [41] を構築している。現在、BioPortal では、生物医学関係のオントロジーを中心に、200 以上のオントロジーの 140 万以上の用語を参照することができる。

2.5.7 オントロジーの応用

次に、オントロジーの役割や性質を考慮したオントロジー応用のタイプ分類について、述べる。なお、このタイプ分類は溝口理一郎氏によるものである[24]。

タイプ1：共通語彙としてのオントロジー

知識の体系化の第一歩である、語彙の統一のためのオントロジーの応用。すでに、いくつかのドメインで開発されている。

タイプ2：情報アクセスのためのオントロジー

巨大な情報源（WWW など）から、すばやく必要な情報を入手できるようにするためのオントロジーの応用.

タイプ3：相互理解のための媒体としてのオントロジー

相互理解は次のいずれの間においても常に必要である.

a) 人間と人間

知識継承などといった知識を扱う問題の場合、人間同士の相互理解はとても重要な課題になる.

b) 人間とコンピュータ

情報検索などといった、人間とコンピュータが関わる問題の場合、検索するコンピュータが利用者の要求を正しく理解することは重要である. そこに、オントロジーの利用、オントロジー変換などによる意味解釈機能などが求められる.

c) コンピュータとコンピュータ

エージェント同士が通信する場合には、通信のプロトコルと語彙に関する合意が必要である.

タイプ4：規約としてのオントロジー

インスタンスは実世界のオブジェクトのモデルであり、オントロジーはインスタンスのモデルである. そのことから、オントロジーは実世界のオブジェクトのモデルに対するモデルという意味で、実世界のオブジェクトに対するメタモデルであり、実世界のオブジェクトに対する規約になっている.

タイプ5：知識の体系化の基礎としてのオントロジー

ものづくりをする企業においては、技術知識の共有に対する要求は極めて高い. 専門家が用いる専門用語の統一を図るためにオントロジーを用い、専門家にとって意味のある表現の形式と概念の体系化を行うことはナレッジマネジメントにとって有益である.

2.5.8 オントロジーとナレッジマネジメント

Thomas H. Davenport らが知識の貯蔵庫から意味的検索により的確に知識を発掘するためにシソーラスの整備の重要性を指摘していることは既に述べた（2.2.8 p.12 参照）が、昨今、ナレッジマネジメントにおいても、シソーラスを一步進めて、オントロジーの活用が

進んでいる。例えば、ある保険会社では、スキル情報の管理のために、スキル情報を記述する上で必要となる概念をオントロジー化し、社員のスキル情報記述の標準化・統制を行っている[42]。ただし、その目的はシソーラスと同様意味的検索の実現が主眼であり、スキル情報そのものの体系化という視点はない。

また、自動車ボディの製造工程におけるゆがみの発生を抑えるノウハウの再利用のためのある事例推論システムでは、その事例の記述に必要な概念がオントロジー化され、事例を記述するに当たっては、プルダウン・メニューよりオントロジーの語彙を選択して記述する[43]。その目的は、事例推論のために各事例の類似性の尺度を導入することと同時に、事例の記述の属人性を排除し、理解性・再利用性を向上させることにある。ただし、それは記述様式および語彙の標準化というレベルにとどまっている。

人間の理解性向上のための知識の体系化という観点からみると、オントロジーを構成するプリミティブは語彙が対応する概念とは限らない。MIT Process Handbook [44] は、自然言語によるビジネス・プロセスの記述であるが、その自然言語によるビジネス・プロセスの記述をプリミティブとし、それらの間に構成関係や汎化・特化関係などの意味的關係が定義され体系化されているが故に、オントロジーと呼ばれることがある。例えば、MIT Process Handbook における Create to Order (受注生産) の記述は図 15 エラー! 参照元が見つかりません。に示すように、自然言語での記述が中心である。



出所：http://process.mit.edu/Activity.asp?ID=2636

図 15 MIT Process Handbook における Create to order の記述

ただし、Parts of Create to Order に構成する子プロセスが示されているほか、View with Compass Explore をクリックすることにより、子プロセスを含む、次のような意味的關係が示される。

関連プロセス : Create computers to order{Dell}, Create automobiles to order {Volvo} 他

汎化 : Act <= Create <= ... <= Create with what customization? <= Create to order

特化 : Create to order <= Create technical products to order <= Create computers to order{Dell}

部品 : Design products and process, Buy, Make, Sell, Manage as a creator

MIT Process Handbook は、様々なビジネス・プロセスを体系化することにより、それらを容易に理解・比較検討できるようにし、それを基に改善された新たなビジネス・プロセスを創出することを意図している。

同様に、2.4 で述べた CommonKADS の知識モデルを、ナレッジマネジメントのために、人間の理解性向上のための知識の体系化という観点で見ると、オントロジーとして体系化されるべきプリミティブは、領域スキーマの要素だけでなく、知識ベース上のクラス間の制約なども含まれる。

本論文は知的熟練を体系的に表出化することによって新人への継承を支援しようとするものである。従って、本論文においても、人間の理解性向上のための知識の体系化という観点でオントロジーを幅広く捉え、知的熟練を比較的小さな粒度の業務ルールに分割するとともに、それらを含めオントロジーとして体系化することを提案する。

第3章 ルール・オントロジーとドメイン・オントロジーに基づく知識継承支援システム

本章では、これまでに述べた背景知識・関連研究を踏まえ、オントロジーを活用した知識継承支援システムを提案する。

3.1 スコープ

本提案を具体的に述べるに前に、まず、そのスコープを明確にする。

まず、対象とする組織は、知的熟練を必要とするあるまとまった責務を果たしている現場事業所である。大企業の場合であれば一現場事業所が該当し、中小企業であれば企業全体の場合もある。ただし、いずれにしても、そこには、当該事業所の長（事業所長、あるいは、経営企画・管理部門）があり、その方針の下で、知的熟練を要するいくつかの業務を遂行していて、組織のスリム化に伴い OJT による熟練者から新人への知的熟練の継承が困難になりつつあり、できる限り OJT によらない知的熟練の継承が求められている。

なお、ここで、知的熟練の継承の対象である新人とは、全くの新人を想定している訳ではない。業務に対する最低限の背景知識を持ち、業務マニュアルが整備されている場合であれば業務マニュアルに記載されていることは理解できるレベル、業務マニュアルが整備されていない場合であれば、標準的な業務に対する最低限の OJT は受けているレベルを想定している。現在問題となっているのは、全くの新人の教育ではなく、スリム化された組織において知的熟練を継承していくためにいかにして熟練者を育成していくかであるからである。

本研究は、そのようなレベルの新人が、知的熟練を獲得し、熟練者に育っていくことを支援することを目的とし、そのために、熟練者が自らの知的熟練を表出化・連結化し、それを新人が内面化するためのシステムの「場」を提供する。

なお、対象となる知的熟練は、代々蓄積されてきたものであり、比較的安定しているものの、経営環境などの変化に伴い、事業所の長の意思に追従して当然進化していく必要があり、熟練者による表出化・連結化は持続的なものであることが求められる。

3.2 提案システムの概要

上記のスコープを前提に、提案システムの全体像を図 16 に示す。

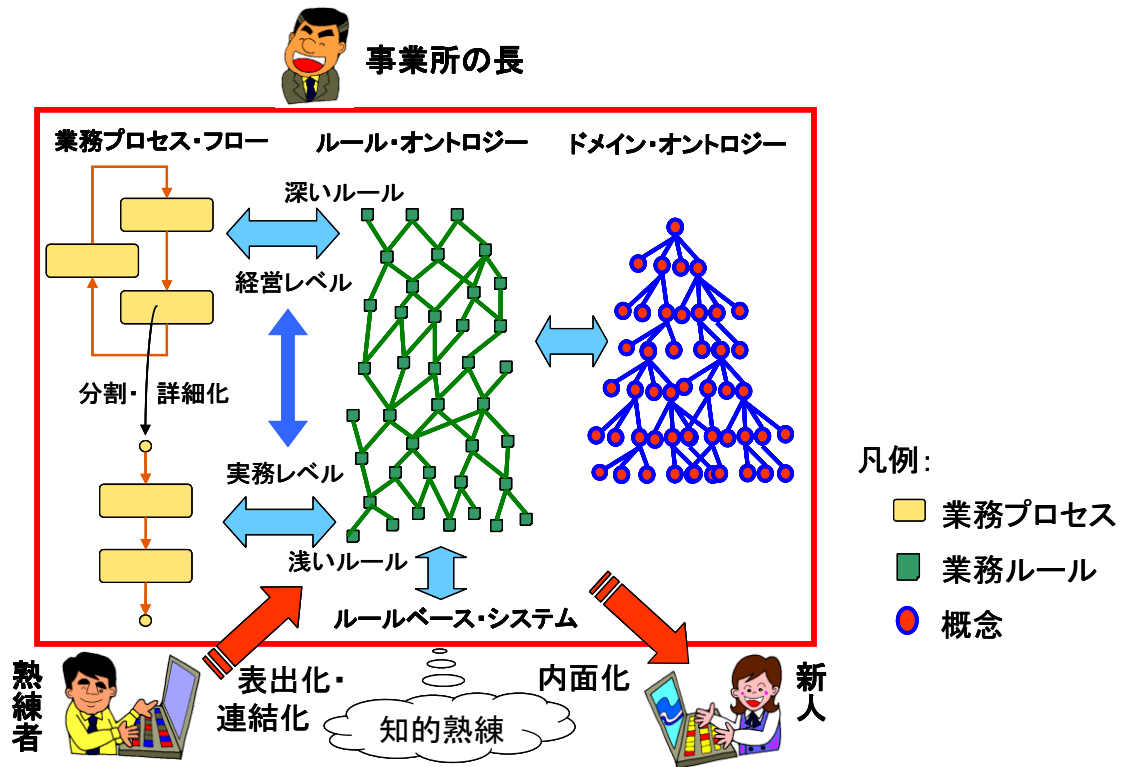


図 16 提案システムの全体像

標準的な業務プロセス・フローを前提に、熟練者は事業所の長の意思を反映しつつ、自らの知的熟練をルールベース・システムを活用しながら、ルール・オントロジー、ドメイン・オントロジーに表出化・連結化する。新人は、ルール・オントロジー、ドメイン・オントロジーに表出化・連結化された知的熟練を、やはりルールベース・システムを活用しながら、内面化する。

次節以降で詳しく説明する前に、その概略を説明する。図 16 左側の業務プロセス・フローは概略の業務フローを記述したものである。最上位の経営レベルの業務プロセス・フローは当該事業所の業務全体に関し、その長が全体を概観する上で必要になるレベルで記述したものであり、それをブレイクダウンしていった最下位ものが実務レベルの業務プロセス・フローである。

また、図 16 中央のルール・オントロジーが本提案の中核をなすものである。熟練者の暗黙知である知的熟練を表出化し、整理することにより、多くの場合、自然言語で記述できる比較的小さな粒度の単純な業務ルールに分割される。ルール・オントロジーは、それらの業務ルールをプリミティブとし、各業務ルール間に意味的關係を導入し、体系化したものである。経営レベルの「深いルール」(3.4 参照)は事業所の長の意思に従った業務の目標を表し、実務レベルの「浅いルール」(3.4 参照)は実務者が業務を遂行する上で直接使

われるルールとなる。個々の業務ルールが全体として体系化されることにより、新人も事業所の長の意思に沿って業務の全体像を理解する。また、熟練者も、経営環境などの変化に伴い、事業所の長の意思に従って、経営レベルの「深いルール」との関係から、更新すべき実務レベルの「浅いルール」を把握できるようになる。

図 16 右側のドメイン・オントロジーは、ルール・オントロジーはルール・オントロジーの各業務ルールを構成する概念を体系化し、それに対する語彙を標準化すると共に、新人も理解できるような説明を加えたものである。ドメイン・オントロジーにより、新人は各業務ルールの正確な意味を理解することができる。また、ドメイン・オントロジーは、熟練者が自らの知的熟練を業務ルールに表出化する際に、先行研究[41, 42]と同様にその記述を統制し、標準化するのにも利用され、また、それを次に述べるルールベース・システムの実行ルールへ変換するのにも利用される。

最後に、図 16 中央下のルールベース・システムは、新人がルール・オントロジー、ドメイン・オントロジーに表出化された知的熟練を内面化するために活用されると共に、熟練者が自らの知的熟練の業務ルールへの表出化に漏れがないことの確認にも利用される。そのためには、ルール・オントロジー上の日本語の「最も浅いルール」はルールベース上の実行ルールに容易に変換できる必要があるが、ルールベース・システムは、日本語の「最も浅いルール」の日本語記述がドメイン・オントロジーに統制されていることを前提に、「最も浅いルール」の日本語記述をルールベース上の実行ルールに変換する機能も保持する。

それぞれに関して、詳しく説明する。

3.3 業務プロセス・フロー

業務プロセス・フローは、事業所の長にとっても、あるいは、実務者にとっても、最も直感的に業務の概略を理解できるものである。事業所の長は当該事業所の業務全体を概観する上で、最上位の経営レベルの業務プロセス・フローは必要になるレベルで記述したものであり、CommonKADS における最もハイレベルな業務モデルに相当する。また、それをブレイクダウンしていった最下位の実務レベルの業務プロセス・フローは、業務マニュアル・レベルで記述したものであり、CommonKADS における知識モデルのタスク知識、推論知識に相当する。ここで、業務マニュアル・レベルとは、不確実性や変化に伴う差異を捨象して標準的な業務フローとして無理なく記述できるレベルのことである。新人は、この実務レベルの各業務プロセスに対し、何をする業務プロセスであるかは理解できるが、どうすればそれができるかは理解できていない。なお、業務プロセス・フローを作成するにおいては、業務マニュアルなどに業務プロセス・フローが整備されている場合は、それを積極的に参考にする。

3.4 ルール・オントロジー

熟練者の暗黙知である知的熟練は、表出化し、整理することにより、多くの場合、自然言語で記述できる比較的小さな粒度の単純な業務ルールに分割される。新人は次節のドメイン・オントロジーを参考にすることにより業務ルールを個々には理解できるが、それは必ずしも知的熟練全体の理解にはならない。ルール・オントロジーは、ドメイン・オントロジーを構成する個々の概念よりも粒度の大きい個々の業務ルールをプリミティブとし、それらの間に意味的関係を導入し、体系化することにより、新人が、個々の業務ルールの理解を超え、表出化された知的熟練全体を統合的に理解することを支援するものである。

発想としては、橋田の粗い粒度の知的コンテンツ [45] に近い。ただし、粗い粒度の知的コンテンツは、単文程度のテキストの理解が認知的負荷の小さいほとんど無意識な過程であるとして、単文程度のテキストをプリミティブとしているが、ここでは、業務的に意味をなし得るレベルの単純な業務ルールをプリミティブとしてとらえる。

具体的には、例えば、ある業務ルールがなぜ必要なのかを事業所の長の意思を踏まえ示したり、一般的に適用されるルールと特定の場面でのみ適用されるルールとの関係を示したりすることにより、新人が個々の状況に応じて、業務ルールを正しく適用して、実際に業務を遂行出来るようになることを支援する。

ルール・オントロジーの意味的関係には、次の4つがある。

1. 正当性関係 (justify)

あるルール A が別のルール B の正当性を説明する。このとき、相対的に、ルール A を「深いルール」、ルール B を「浅いルール」と呼ぶ。

2. 依存関係 (depend on)

あるルール A が適用されるためには、ルール B の適用が前提となる。

3. 特殊化関係 (specialized)

あるルール A は別のルール B の詳細化になっている。ルール A はルール B と矛盾しない。

4. 例外関係 (excepted)

あるルール A はあるルール B の例外を規定している。ルール A はルール B と矛盾する。

この中で、特に、正当性関係は、浅いルールをそのまま適用してはならない不確実性や変化が生じた場合に、その正当性を説明する深いルールを示すことより、新人が不確実性や変化をこなす能力を身につけることを支援する。正当性関係 (justify) における「最も深いルール」は、最上位の経営レベルの業務プロセス・フローにおける各業務プロセスに対

応し、その業務プロセスの当該事業所としての業務目標を表す。一方、「最も浅いルール」は、最下位の実務レベルの業務プロセス・フローの各業務プロセスと対応し、実務者がその業務を遂行する上で直接的に使うルールとなる。

また、正当性関係は経営環境などの変化に伴って業務目標が変化した場合に、変更すべき浅いルールを熟練者が認識する上でも活用される。実務で直接的に使われる浅いルールは一度確立されると、本来より深い業務目標を達成するための手段であるそのルール自体が目的化して必要な変更がなされないことは、現実にはよく見られる。ルール・オントロジーは、経営環境などが変化した場合に、当該事業所の業務目標である最も深いルールとの正当性関係により、更新すべき浅いルールを示唆する。ただし、一般的には、「最も深いルール」そのものは普遍性が高く、多くの場合、経営環境などの変化があっても、「最も深いルール」そのものが入れ替わるのではなく、既存の「最も深いルール」の重要度に変化が生じる。一方で、浅いルールと深いルールの関係は多くの場合多対多の関係になり、「最も深いルール」の重要度に変化が生じた場合に、それに追従すべき浅いルールの把握が必ずしも容易ではない。そこで、多対多の関係を維持しつつも、浅いルールから見て、その正当性を最も強く説明する深いルールを一つ選定することにした。それにより、各「最も浅いルール」から見て、その正当性を最も強く説明する「最も深いルール」が一つに定まるようになり、また、「最も深いルール」から見ても、それが正当性を説明している主な「最も浅いルール」が絞られ、把握が容易になる。

3.5 ドメイン・オントロジー

ドメイン・オントロジーは、知的熟練を構成する概念を体系化し、それに対する語彙を標準化したものである。ここで、知的熟練はルール・オントロジーの各業務ルールとして表出化されているので、ドメイン・オントロジーは業務ルールを構成する概念から構築される。

ドメイン・オントロジーの目的の一つは、新人が個々の業務ルールを理解することを支援することであり、そのため、ドメイン・オントロジーは、体系化した各概念に対し、標準化した語彙を与えるだけでなく、新人がその意味を理解できるよう、日本語による説明記述を含む。

また、ドメイン・オントロジーは、熟練者が自らの知的熟練を業務ルールに表出化する際に、先行研究[42, 43]と同様にその記述を統制し、標準化するのにも利用される。業務ルールの記述を統制し、標準化する目的は、新人による理解を容易にすることと共に、「最も浅いルール」に関しては、ルールベース上の実行ルールへの変換を可能にすることにある。これにより、ルール・オントロジー上の日本語記述による業務ルールにより、ルールベース・システム上の実行ルールの管理が可能になり、熟練者によるルールベース・システムの管理が可能になる。

3.6 ルールベース・システム

「最も浅いルール」は、CommonKADS における知識モデルの領域知識の知識ベースに相当するものであり、業務プロセスの性格によっては、ルールベース上の実行ルールに変換され、業務プロセスを半自動で実行するルールベース・システムを構成する。

ドメイン・オントロジー、ルール・オントロジーにより、新人にも理解しやすい形で知的熟練が表出化・連結化されたとしても、新人がそれを実務で適切に使用できるまでに内面化するには、実務に近い形での適用経験が必要であり、そのためにルールベース・システムが活用される。新人は自ら実行した業務結果とルールベース・システムのアウトプットを試行錯誤的に比較することを通じて、獲得した知的熟練の十分な内面化が可能になる。

また、ルールベース・システムは熟練者が自らの知的熟練をオントロジーに表出化・連結化することの支援にも活用される。OJT によって知的熟練を獲得してきた熟練者にとっては、それを表出化することは容易ではない。とりわけ、熟練者がルール・オントロジーにおけるプリミティブとしての業務ルールをもれなく認識することは容易ではない。彼らは、状況に応じて様々な業務ルールを使い分けているが、それを明示的には意識していないからである。そこで、熟練者はルールベース・システムを利用し、ルールベース・システムのアウトプットと自ら実行した業務結果とを比較することにより、業務ルールの抜け・不完全さなどをチェックする。

このルールベース・システムは、ルール・オントロジー上の「最も浅いルール」の記述がドメイン・オントロジーにより統制されていることを利用して、「最も浅いルール」をルールベース上の実行ルールに変換する機能を持ち、熟練者に代わって業務を半自動で実行することで熟練者に対しても直接的なメリットを与え、熟練者が自らの暗黙知をルール・オントロジーとして表出化することへの動機を与えることと合わせて、熟練者自らによるルールベースの持続的な管理を可能にする。

3.7 支援システム

3.7.1 GEN (General knowlEdge Navigator)

本提案を支援し、また、その有効性を検証するために、GEN (General knowlEdge Navigator) と呼ばれるナレッジ・マネジメントに特化したオントロジー・リポジトリの実験システムを開発した。これは、熟練者が知的熟練を表出化・連結化し、新人がそれを内面化する「場」を提供するものである。GEN の開発に当たっては、実験システムという性格を考慮し、プロトタイプの開発とそのスパイラル的改良に適したオブジェクト指向言語である Squeak (Smalltalk の一種) を採用し、また、オブジェクト指向データベース管理シ

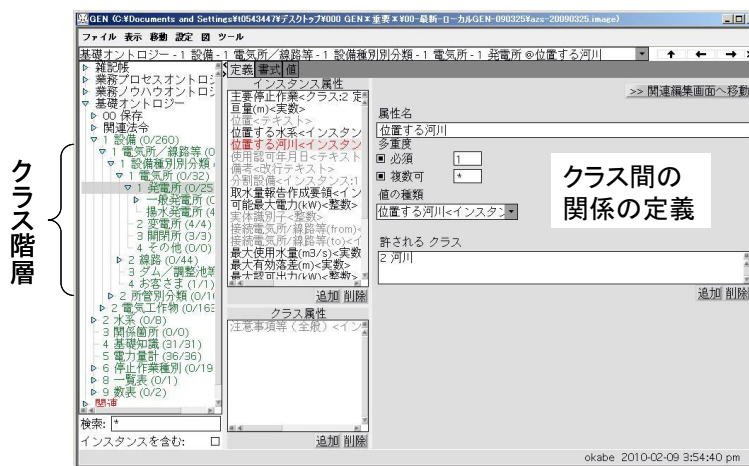
システムである omniBase を利用している. 現状のシステム規模をエラー! 参照元が見つかりません. に示す.

表 1 GEN のシステム規模

クラス数	541
メソッド数	7,328
ライン数	60,764

GEN はナレッジマネジメントに特化したオントロジー構築ツールであり, Protégé [40]と比較すると, よりエンドユーザ志向であり, ルール・オントロジーのようなテキスト情報の構造化に適している. GEN は, Protégé [40]とは異なりスロットという概念を明示的には用いず, すべてクラスに付随する属性として扱うなど, エンドユーザ自らが違和感なくオントロジーを構築できるよう配慮されている. クラスの構造を定義する画面例を図 17 に, また, インスタンスの表示画面例を図 18 に示す.

Protégé [40]にはない機能としては, GEN には業務プロセス・フローを記述する機能がある. これは UML のアクティビティ図に類似のものであるが, エンドユーザが無理なく利用できるよう, ノーテーションを簡素化する一方で, 自然言語で自由に補記できる領域を付加している. 各業務プロセス (アクティビティ) は, 必要に応じ, ドメイン・



オントロジーやルール・オントロジーとリンクされる (図 19 参照).

図 17 クラス構造の定義例

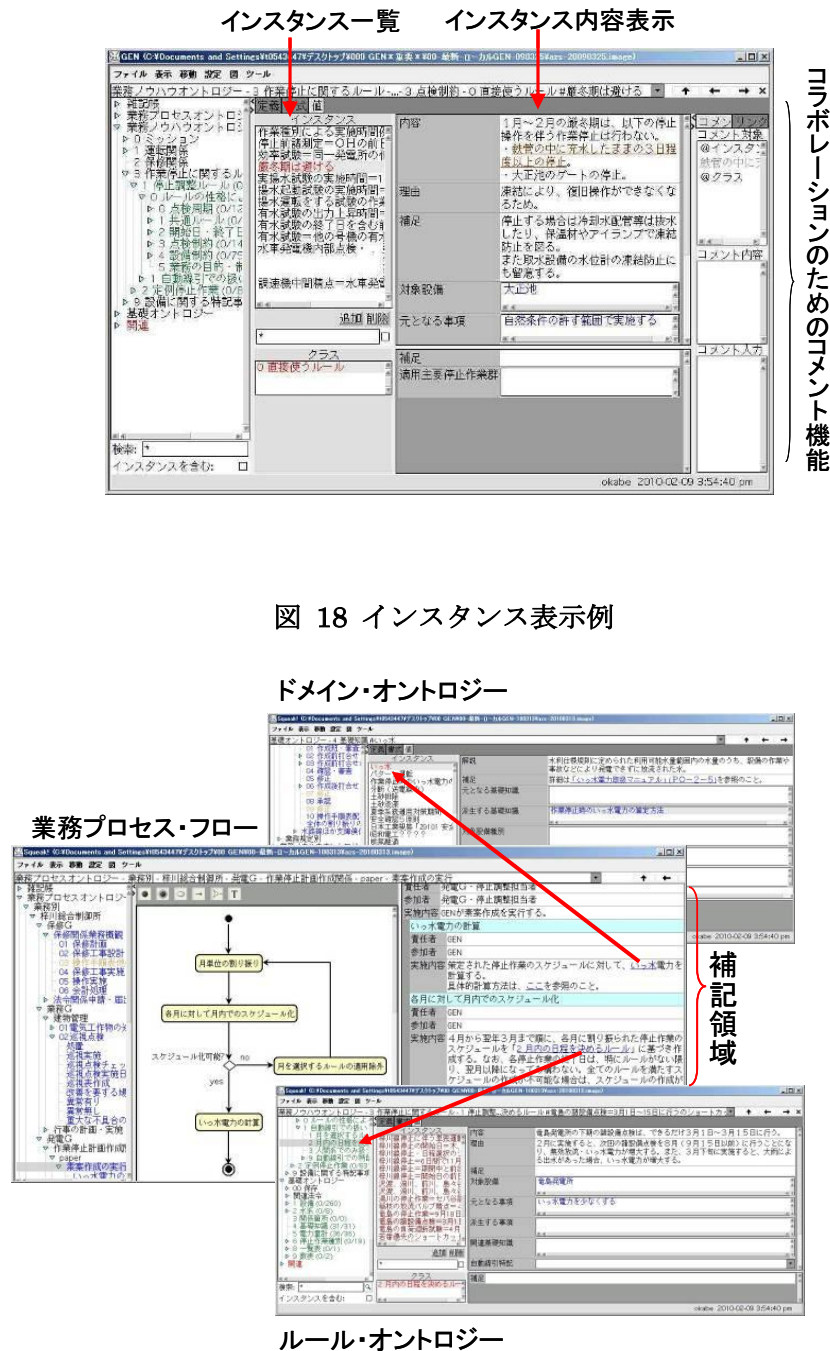


図 19 業務プロセス・フロー，ドメイン・オントロジー，ルール・オントロジーの関係

システム構成としては、クライアント側に Squeak の仮想マシンと GEN のプログラムを、サーバー側に omniBase を導入するクライアント・サーバー構成となっているが、omniBase 自身はプロセスを一切立ち上げないデータベース管理システムであり、クライアント側から見ると単に Windows ファイル共有プロトコル(SMB)での共有ファイルという位置づけになる。Squeak の仮想マシンが Windows, Mac OS X, Linux のいずれの OS にも対応し、Windows, Mac OS X, Linux のいずれも Windows ファイル共有プロトコル(SMB)をサポートしているため、GEN は実質的にはシステム環境に制約を受けることなく利用可能である。その特徴を活かし、次節に述べる東京電力における試行においては、社内標準の既存のファイル共有サーバー、業務用 Windows PC 上に GEN をインストールしている。そのシステム構成図を図 20 エラー! 参照元が見つかりません。に示す。

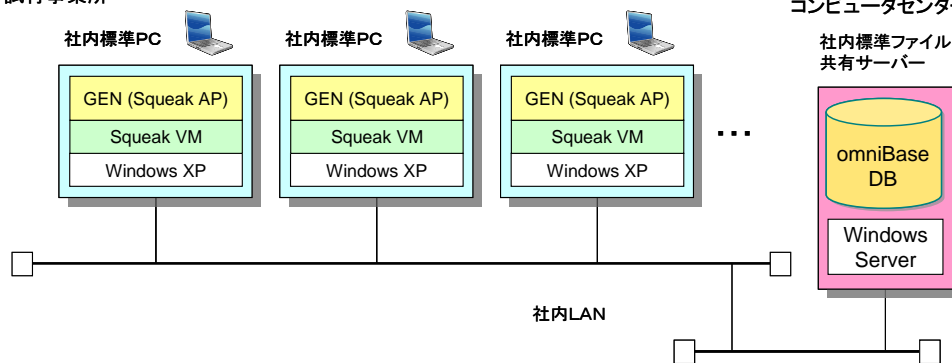
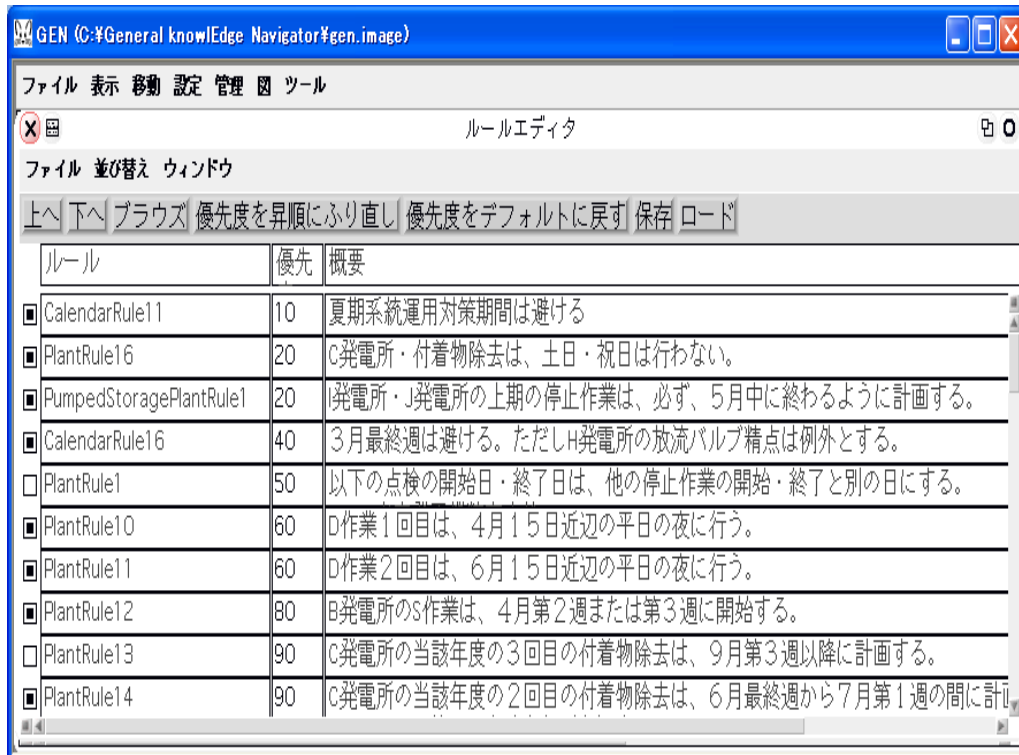


図 20 GEN システム構成 (東京電力における試行での例)

3.7.2 ルールベース・システム

3.7.2.1 ルールベース・システム本体

ルールベース・システムは、まず、Prolog により開発し、それを Squeak 上の有限領域制約充足問題用のライブラリである Backtalk[46]を利用して GEN に移植している。知的熟練は様々な状況に応じて知識を適切に選択することが求められる。逆に言うと、ルール・オントロジー上の「最も浅いルール」から変換されたルールベース上の実行ルールがすべての状況において整合であるとは限らず、状況によっては相互に矛盾しているような場合もある。ルールベース・システムがそのような状況にも対応できるようにするために、GEN 上のルールベース・システムは、図 21 に示すように、個々の状況に応じて、ルールを適用するか否かを左のチェックボックスで指定できるようにし、その上で、更に、各ルールに優先度を設定できるようにしている。この優先度とは、すべてのルールを満たす解が存在しない場合に、解を得るためにルールの適用を除外する順位を指定するものである。すな



わち、解が存在しない場合は、解が存在するようになるまで、優先度低いルールから順に適用を除外していく。

図 21 ルール編集画面

3.7.2.2 日本語ルールの実行ルールへの変換機能

「最も浅いルール」の日本語記述をドメイン・オントロジーにより統制することにより、「最も浅いルール」の日本語記述をルールベース上の実行ルールに変換する機能に関しては、現在、Java により Prolog によるルールベースを前提に開発されている。これらは、将来的には、GEN 上の Backtalk に移植する予定である。

システムの概略のモジュール構成を図 22 に示す。ここで、中核となるのは、ルール生成・変換モジュールである。ルール生成・変換モジュールは、ドメイン・オントロジーの語彙を選択することによりルールの条件部、結論部を構成し、そこから日本語ルールを生成する。また、ルール生成・変換モジュールは、粗粒度の日本語記述を細粒度の Prolog の実行ルールに変換するために、様々な条件部、結論部に対応した Prolog のルールテンプレートを保持している。ルール生成・変換モジュールは、条件部、結論部の指定に応じて、適当なルールテンプレートを選択し、そのルールテンプレートに条件部、結論部で選択したドメイン・オントロジーの語彙を埋め込むことにより、Prolog の実行ルールを生成する。なお、ルールテンプレートには、(年度, 月) と (年, 月) の変換、年月日とユリウス通日との変換などの共通に用いられるルールの呼び出しが含まれる。これらの共通に用いられる

ルールは、あらかじめ、ルールベース上に共通ルール・ライブラリとして定義してある。

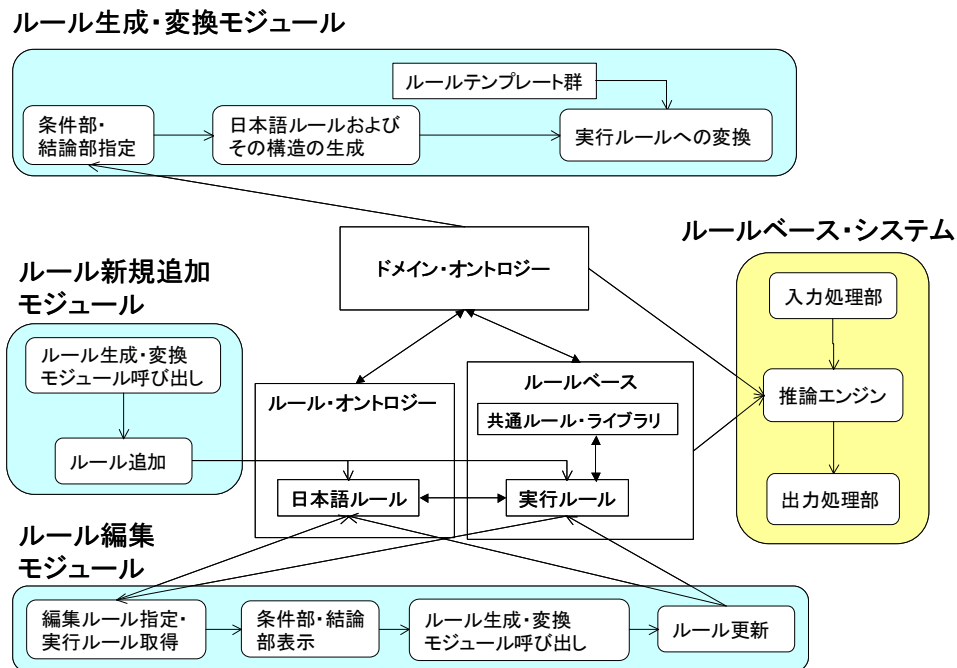


図 22 日本語ルール生成と実行ルールへの変換のモジュール構成

このルール生成・変換モジュールを用い、ルール新規追加モジュールは新たなルールの追加を行い、ルール編集モジュールは既存のルールの更新を行う。なお、どちらの場合も、生成された日本語によるルール記述に不自然さが残る場合も排除しきれないため、日本語によるルール記述に関しては後から修正できるようにしている。

3.7.2.3 ルール・オントロジーを活用した事業所の長の意思の反映機能

ルール・オントロジーの節で述べたように、ルール・オントロジーは、経営環境などの変化に伴い、事業所の長の意思に従い更新する必要がある「最も浅いルール」を、熟練者が把握するためにも利用される。また、ルール・オントロジーの節では、一般的には、「最も深いルール」そのものは普遍性が高く、多くの場合、経営環境などの変化があっても、「最も深いルール」そのものが入れ替わるのではなく、既存の「最も深いルール」の重要度に変化が生じることも述べた、そこで、これを一歩進めて、経営環境などの変化に対し「最も深いルール」の重要度を変化させるという事業所の長の意思が示された場合に、ルール・オントロジーを活用して「最も浅いルール」およびルールベース上の実行ルールを変化させ、ルールベース・システムが事業所の長の意思に沿った結果を半自動で出力できるようにする機能を提供する。「最も浅いルール」および実行ルールを変更するアルゴリズムはいくつか考えられるが、ルールベース上の実行ルールには優先度が与えられていて、解が存

在しない場合は、優先度の低い実行ルールから適用を除外して解を得るという方式に沿ったアルゴリズムとして、図 23 に示されるものがある。すなわち、選択した「最も深いルール」により正当性を説明されない「最も浅いルール」の中で、優先度の最も低いものを適用除外する。より自然には、「最も深いルール」により正当性を説明される「最も浅いルール」の優先度を高くすることも考えられるが、解が存在する場合に特定のルールの優先度を上げて結果は同一になるため、上記のアルゴリズムとしている。

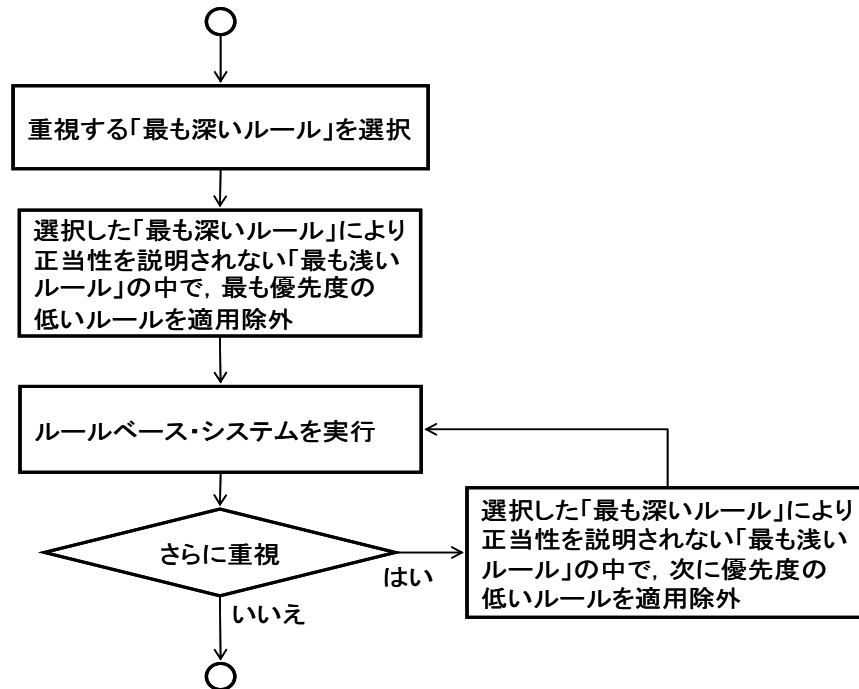


図 23 実行ルールの適用除外アルゴリズム

3.8 初期構築手順

本提案では、熟練者が知的熟練を自らオントロジーやルールベースに持続的に表出化・連結化することを前提としていて、既に述べたようにそのための支援機能を有している。ただし、初期構築においては、オントロジーやルールベース・システムを専門とするナレッジエンジニアが、熟練者と一体となって、業務プロセス・フロー、オントロジー、ルールベース・システムを構築することを想定している。次に初期構築の手順をまとめた。

Step 1. 知識獲得

まず、業務マニュアル、業務を行う時に参照する資料、業務のアウトプットである業務実施報告書などの分析をすると共に、事業所の長および実務家へのインタビューを行い、業務に必要な知識をできるだけ多く取り出していく。

Step 2. 業務プロセス・フロー構築

Step1 で獲得した知識をもとに、まず、事業所の長のレベルおよび実務者のレベルでの標準的な業務フローを業務プロセス・フローにまとめる。その際、業務マニュアルの業務プロセス・フローが記載されている場合には、それを積極的に利用する。その後、その業務プロセス・フローに沿って取り出した知識をもとに業務を模擬的に行い、熟練者が過去に行った業務と比較して、差異のあるところに関して、不足している知識がないか再び熟練者にインタビューをして確認する。

Step 3. ルール・オントロジー構築

取り出した知識を業務に直接使われる「最も浅いルール」とその理由を示す「深いルール」に分解する。それから、業務目標を含めてルール間の正当性関係を定義していく。正当性関係に意味の飛躍がある場合は、中間の業務ルールを追加する。また、正当性関係以外の意味の関係も定義する。この段階で、各業務プロセスとその業務プロセスがその抽象度に応じて必要とする業務ルールとの間のリンクを張る。

Step 4. ドメイン・オントロジー構築

ルールに含まれる概念を対象に、ドメイン・オントロジーを構築していく。また、この段階で、同じ概念に対して熟練者ごとに異なった記述をしている場合や、同じ記述で異なった概念を示している場合は標準化を行う。この段階で、各ルールとそのルールを構成しているルール・オントロジーの概念との間のリンクを張る。

Step 5. ルールベース・システム構築

業務に直接使われるルールを対象に、ドメイン・オントロジーを活用して、ルールベース・システムのルールを構築する。ルールベース・システムのアウトプットと熟練者の業務を比較して、差異を解消していくプロセスを繰り返すことで、オントロジーの洗練を行う。

Step 6. 全体の洗練

構築された業務プロセス・フロー、ルール・オントロジー、ドメイン・オントロジー全体を、改めて、事業所の長、実務者に、それぞれの視点からレビューしてもらい、Step 2 に戻り、その指摘事項に従い、必要な洗練を行う。Step 2 ～ Step 6 までを全体が整合し、安定するまで繰り返す。

第4章 モデルケースー停止調整業務ー

4.1 概要

本提案の有効性を確認するために、東京電力の水力発電関係のある現場事業所における「停止調整業務」と呼ばれる知的熟練を要する典型的な業務をモデルケースとして、GENを用い本提案を試験適用し、評価を行った。

4.1.1 対象業務

東京電力における水力発電は長い歴史を持ち、既に、高度に自動化・統合化されている。水力発電所はすべて無人で、水系毎に設置される総合制御所と呼ばれる現場事業所がその水系にあるすべての水力発電所を遠隔無人運転するとともに、保全も担っている。水力発電所は100年近い歴史を持つ古くて小規模なものから、最新式の大規模な揚水発電所まで様々であり、その運転および保全には多様な技術・技能が要求される。今回モデルケースとしたA総合制御所では、小規模な発電所から大規模な揚水発電所10箇所の発電所、22台の発電機の遠隔無人運転を行うとともに、設備メンテナンスの計画的実施やトラブルへの即応を少人数の体制で行っている。多様な発電所の運転・保全を行うため非常に広範な技術・技能を要求されるだけでなく、また、時代と共に、設備診断技術の進歩などにより、求められる技術・技能も変化していて、現場技術者は、少人数の体制の中で、皆、様々な局面に対する幅広い技術・技能を求められる。つまり、現場では今、技術・技能を組織的に蓄積し、活用することが求められている。

A総合制御所で行われている「停止調整業務」は、A総合制御所で管理する10発電所および送電線における点検や工事などのスケジュールを、次のようなさまざまな条件を考慮しつつ、「いっ水」と呼ばれる発電に利用されず放流される水の量が少なくなるように調整し、作成する技術事務業務である。

- 機器毎に定められている点検周期
- 需要に見合った発電量の確保
- 農業協同組合などの流域地域団体との協定
- 点検・工事量の平準化
- 天候・気象要因
- 構造上発生する機器間の停止依存関係 など。

具体的な条件の中身は様々であり、点検周期を除くほとんどのものが、長年の経験により暗黙知として形成されてきたもので、明示的に規定されていらず、それが、代々、OJT

により引き継がれてきている。また、条件の中には、構造上発生する機器間の停止依存関係など、必ず満たさなければならないものと、作業量の平準化など、望ましいが絶対的ではない条件が混在していて、すべての条件を完全に満たすスケジュールは必ずしも存在しない。そのような場合は、一部の条件を緩和してスケジュールを作成することになるが、その場合のどのような条件を緩和することが出来るかは状況に依存し、長年の経験により培われてきたものである。以上のように、この業務は、様々な状況・制約を幅広く考慮する必要がある「知的熟練」を要する業務であると同時に、その「知的熟練」が表出化されていない典型的な業務である。

本モデルケースでは特にその中で、発電所ごとの各発電機の点検についての年度レベルの粗い停止計画から、年間の日レベルの詳細なスケジュール（以降、停止スケジュール）を作成する業務を対象とする。現在、この業務は1人の熟練者によって行われているが、この熟練者は、5年を要して、この業務に必要な知識をOJTにより暗黙知のまま継承した。この事業所では、「停止調整業務」の次の後継者を育成するにあたり、習得期間をできるだけ短縮し効率よく知識を習得できる仕組みを必要としている。

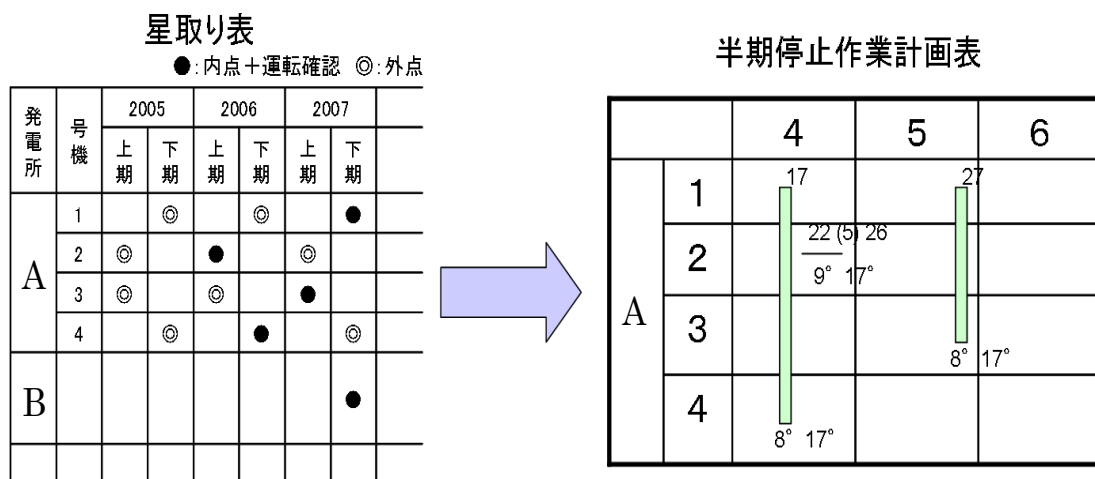


図 24 停止調整業務の概要

4.1.2 検証の進め方

検証すべきことは、本提案が熟練者から新人への「停止調整業務」に関する知的熟練の継承の支援に有効に機能することであるが、それは、大きくは、熟練者による知的熟練のオントロジーなどへ表出化・連結化、オントロジーなどに表出化・連結化された知的熟練の新人による内面化という2つのフェーズに分かれる。

さらに、熟練者による知的熟練をオントロジーなどへ表出化・連結化に関しては、初期の表出化・連結化と、その後の経営環境などの変化に伴い、事業所の長の意思に従った持

続的な表出化・連結化の2つのフェーズに分かれる。この内、初期の表出化・連結化に関しては、オントロジーやルールベース・システムを専門とするナレッジエンジニアが、業務の熟練者と一体となっていくことが可能であるが、その後の持続的に表出化・連結化に関しては熟練者自らが行っていく必要があり、そのための支援が重要である。本提案も、熟練者による表出化・連結化に関しては、後者を支援することを主眼としている。

そこで、今回のモデルケースによる検証においては、本提案が主として支援しようとしている内容に対応させて、次の二つのステップに分けて検証を行うこととした。

ステップ1：初期構築とその新人による内面化に関する検証

ステップ2：熟練者による持続的な表出化・連結化に関する検証

なお、ステップ1においては、熟練者とオントロジーやルールベース・システムを専門とするナレッジエンジニアが一体となって、初期の知的熟練の表出化・連結化が可能であることの確認も行った。

4.2 初期構築とその新人による内面化に関する検証

4.2.1 初期構築

4.2.1.1 業務の概要把握と業務プロセス・フロー構築

東京電力においては、全社大で業務マニュアルの整備が行われており、「停止調整業務」に関しても全社大の業務マニュアルが整備されていた。そこで、まず、この業務マニュアルから「停止調整業務」に関する情報を得ることとした。この業務マニュアルはあくまで全社大のものであり、A 総合制御所において具体的にどのようなすれば適切な「停止スケジュール」が作成されるかについての情報はなかったが、「停止調整業務」の目的、作成した「停止スケジュール」に対する承認行為の手順などの情報は得られた。その上で、A 総合制御所において、事業所長より「停止調整業務」を含む、A 総合制御所の業務概要・地域的特質などの説明を受けると共に、「停止調整業務」の熟練者より、A 総合制御所において、実際にどのようにして「停止スケジュール」が作成されるかの概要説明を受けた。

「停止調整業務」に関する業務プロセス・フローに関しては、「停止調整業務」の全社大の業務マニュアルに記載されているものから、A 総合制御所に該当する部分を抽出し、A 総合制御所の熟練者からの説明をもとに、承認行為などに関し補足を加えることにより、作成した。様々な状況・制約を幅広く考慮する必要がある業務の性格を考慮し、業務プロセス・フローの記述は、あくまで、作成された停止スケジュールに関する承認行為などに留め、停止スケジュールの作成に関しては、単に作成するということを記述するに留め、実

際にどのようにして「停止スケジュール」を作成するかに関しては記述しなかった。また、事業所長レベルの A 総合制御所全体の業務プロセス・フローに関しては、業務マニュアルは各業務毎に全社大で整備されているため、A 総合制御所全体の業務マニュアルといったものは存在せず、業務マニュアルから情報を得ることはできなかった。ただし、東京電力全社大で各組織の分掌業務をまとめた文書が存在したため、これをもとに、A 総合制御所の分掌業務を把握し、事業所長からの説明などをもとに、概略の業務プロセス・フローとしてまとめた。事業所長レベルの A 総合制御所全体の業務プロセス・フローとそこからブレイクダウンされた A 総合制御所の「停止調整業務」の業務プロセス・フローを図 25 に示す。なお、図 25 中の矢印は、A 総合制御所全体の業務プロセス・フローの中で A 総合制御所の「停止調整業務」の業務プロセス・フローにブレイクダウンされる業務プロセスを示している。

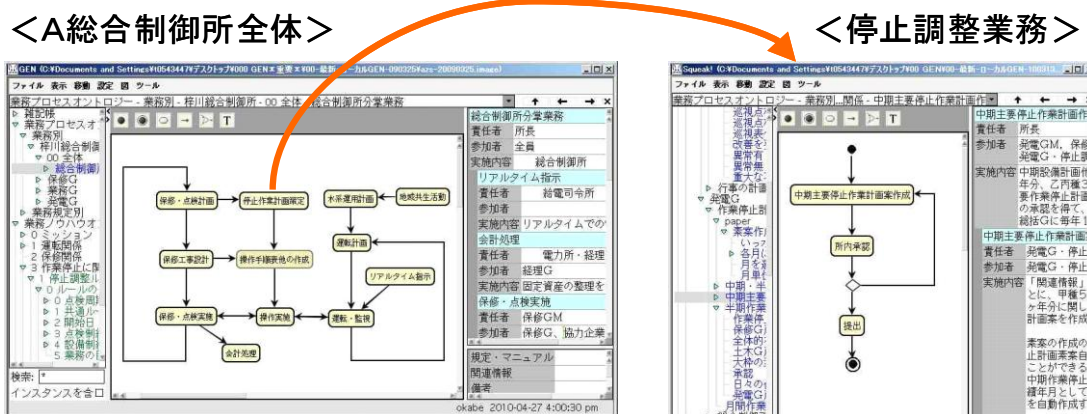


図 25 A 総合制御所全体および停止調整業務の業務プロセス・フロー

4.2.1.2 ルール・オントロジー構築

事業所長および熟練者から熟練者からの概要説明を業務プロセス・フローにまとめた後は、熟練者が過去に実際に作成した停止スケジュールを基に、それをどのような業務ルールに基づき、どのようにして作成したかについてヒアリングし、そこから得られた情報を業務ルールにまとめ、確認していった。

初期に得られた業務ルールは、例えば、次のようなものである。

- ・ G 発電所では、4～9 月は、停止作業時も、農業用水確保のため、大量の放流をしなければならぬため、できるだけ 4～9 月は停止作業を行わない

ただし、これには、停止スケジュールの作成に直接使われる「最も浅いルール」とその正当性を示す「深いルール」が混在しているため、それを「浅いルール」と「深いルール」に分離していった。

具体的には、「最も浅いルール」は、

「最も浅いルール」：G 発電所では、できるだけ 4～9 月は停止作業を行わない
となり、その正当性を示す「深いルール」は、

「深いルール」：G 発電所では、4～9 月は、停止作業時も、農業用水確保のため、大量の
放流をしなければならない。

となる。

さらに、この「深いルール」自体にも、その正当性も示すルールが混在しているため、
これも、次のように「浅いルール」とその正当性を示すより「深いルール」に分離していっ
た。

「浅いルール」：G 発電所では、4～9 月は、停止作業時も大量の放流をしなければならない。
い。

「深いルール」：地域との協定により、G 発電所は 4～9 月は農業用水確保のため、常に
大量の放流をしなければならない。

また、逆に、初期に把握された業務ルールには、「最も浅いルール」のみが記述され、そ
の正当性を示す「深いルール」が洗い出されていないものも多かった。そこで、「最も浅い
ルール」からその正当性を示す深いルールの書き出しを行った。
その例を次に示す。

「浅いルール」：作業日数 5 日の作業の開始曜日は月曜にする

「深いルール」：作業開始・終了日は平日にする

「浅いルール」：作業開始・終了日は平日にする

「深いルール」：作業者を確保する

「浅いルール」：内部点検の開始・終了日は他の停止作業は行わない

「深いルール」：作業者を確保する

上記のように、「浅いルール」「深いルール」の関係が階層化したり、また、異なる複数の
の「浅いルール」が同じ「深いルール」により正当性を説明されたりする場合が存在した。
そのため、ルール間の正当性関係の理解を容易にするために以下の図 26 エラー! 参照元が
見つかりません。ように視覚化を行いつつ整理を行った。図 26 エラー! 参照元が
見つかりません。はルール B がルール A の正当性を説明していることを表している。

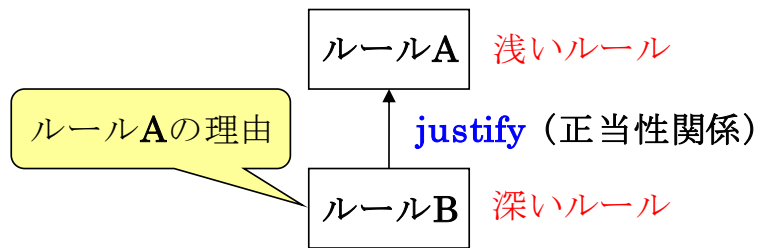


図 26 ルール・オントロジーの視覚化方法

このようにして、熟練者からヒアリングした内容を整理した結果、その内容は 134 個の単独で意味をなし得るレベルの単純な業務ルールに分解された。これらの業務ルールの大きさ（文字数）の分布を図 27 に示す。なお、本業務ルール記述は新人が理解できるためのものであるため、理解のための簡単な補足的な説明を含んでおり、文字数はそれを含んだ文字数である。

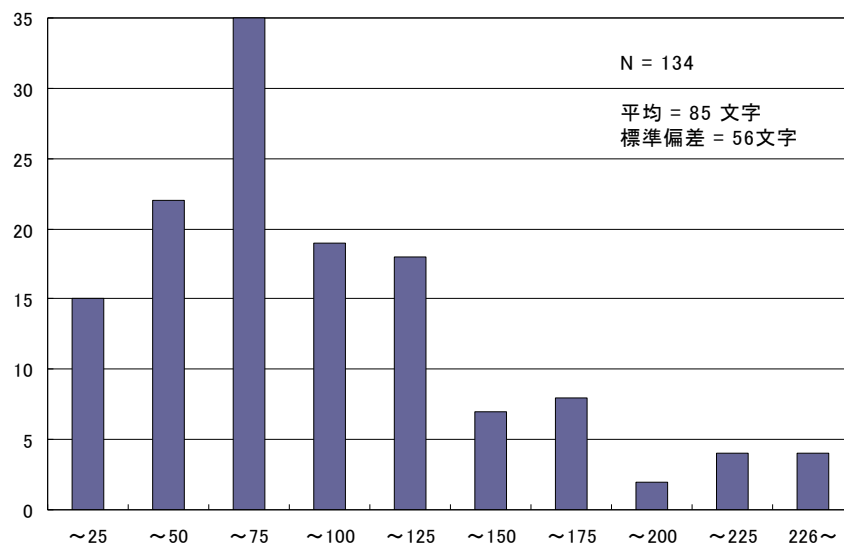


図 27 業務ルールの文字数の分布

134 個のルールのうち 90 個が業務の遂行に直接使われる最も浅いルールであった。また、当該事業所の業務目標と直接対応する最も深いルールは、「いっ水電力を少なくする」「地域社会と共生する」「電力供給信頼度を維持する」など、12 個になった。最も深いルールの一覧を表 2 エラー! 参照元が見つかりません。に示す。

表 2 最も深いルールの一覧

いつ水電力量を低減する。	設備信頼性を維持する。
電力品質を維持する。	機器保全を図る。
供給信頼度を損なわない。	自然条件の許す範囲で実施する。
地域と共生する。	作業体制を確保する。
人身安全を図る。	点検周期を遵守する。
設備効率性を維持する。	効率的に作業を実施する。

また、以下の図 28 に、業務ルール間の意味的関係を視覚化したルール・オントロジーの一部を示す。

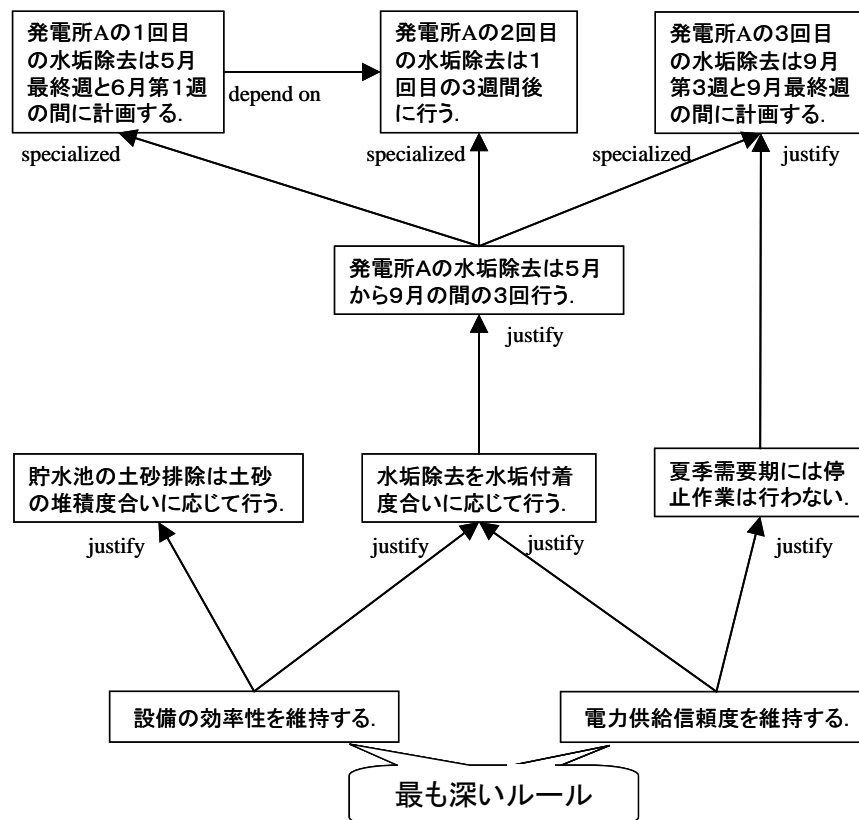


図 28 ルール・オントロジーの一部

例えば、「A 発電所の水垢除去は 5 月から 9 月の間の 3 回行う」というルールに対して、「水垢除去を水垢付着度合いに応じて行う」というルールが正当性を説明している。これは正当性関係 (justify) である。また、「A 発電所の 2 回目の水垢除去は 1 回目の 3 週間後に行う」というルールが適用されるためには、「A 発電所の 1 回目の水垢除去は 5 月最終週と

6 月第 1 週の間計画する」というルールの適用が前提となっている。これは依存関係 (depend on) である。さらに、「A 発電所の 2 回目の水垢除去は 1 回目の 3 週間後に行う」というルールは、「A 発電所の水垢除去は 5 月から 9 月の間に 3 回行う」というルールの詳細化になっていて、矛盾していない。これは特殊化 (specialized) である。その他に、図 28 では示されていないが、「B 発電所の負荷遮断試験は 4 月 5 日から 4 月 20 日に行う」というルールは、「B 発電所の作業は 9 月 18 日から 3 月 15 日に行う」というルールの例外を規定していて、矛盾している。これは例外 (excepted) である。

このようなルール間の関係を導入することにより、「A 発電所の水垢除去」は、単に設備の効率性のためだけではなく、水垢は設備の停止の原因にもなるため、電力供給信頼度維持の関係からも必要であることが、「業務目標」との関係より理解できる。また、電力消費が大きい夏季には電力供給信頼度を維持するために停止作業を行わないことも勘案して、水垢除去の具体的なスケジュールが決められていることなどが新人にも理解できる。

構築したルール・オントロジーを GEN に組み込んだ。図 29 に GEN に組み込まれたルール・オントロジーを示す。GEN では、元となる事項には、選択されているルールの正当性を説明するルールが記述され、派生する事項には、選択されているルールが正当性を説明するルールが記述される。

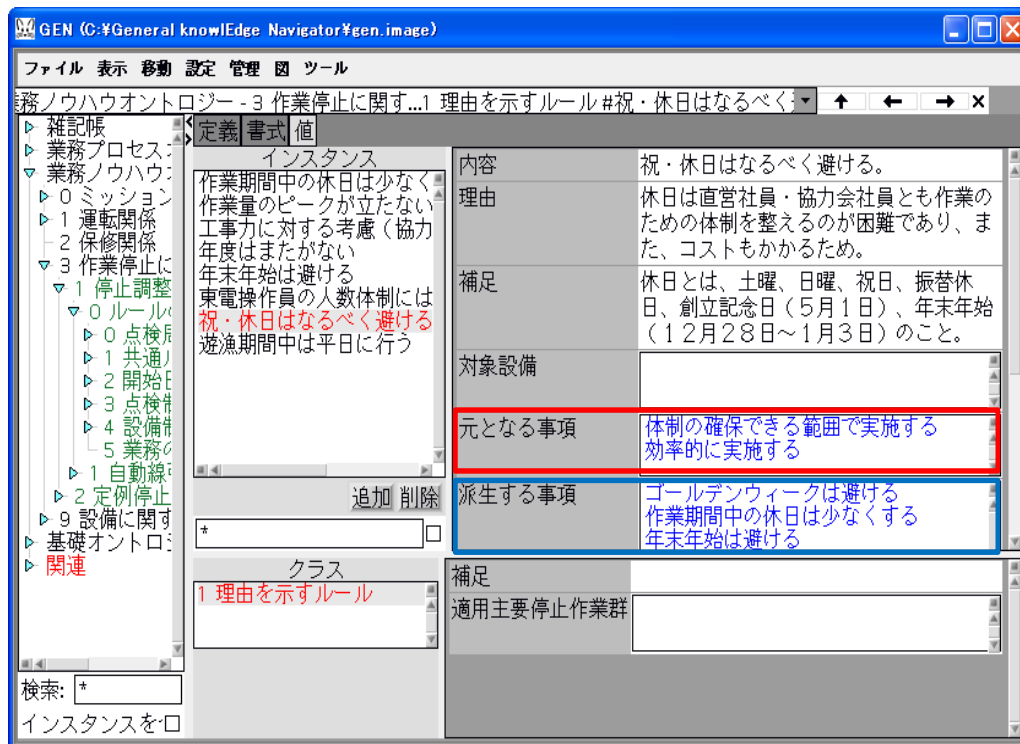


図 29 GEN に組み込まれたルール・オントロジー

4.2.1.3 ドメイン・オントロジー構築

日本語で記述された業務ルールに含まれる概念を対象に，ドメイン・オントロジーを構築した．構築したオントロジーのクラス数は 55 個，インスタンス数は 292 個であった．今回対象とするドメインは，A 総合制御所で行われている設備に関する停止作業のスケジュールを作成する業務であるため，ドメイン・オントロジーは主には設備に関するもの（以下，設備オントロジー）と作業に関するもの（以下，作業オントロジー）となった．その他に，業務ルールを新人が理解する上で必要となる専門的概念も登録した（以下，専門概念オントロジー）．

設備，作業とも，東京電力の水力発電部門において幅広く使われている概念体系が確立していたため，設備オントロジー，作業オントロジーは，それらを概念階層として取り入れた．

まず，設備オントロジーを図 30 に示す．楕円がクラス，四角がインスタンスである．

作業のスケジュール化は発電所（インスタンス）単位に行われるが，その作業をスケジュール化するための業務ルールは多くの場合，その発電所には直接に依存せず，その発電所の属するクラスに依存する．そのため，停止スケジュールの作成において，発電所（インスタンス）からその属するクラスを求めることが頻繁に必要となり，新人が停止スケジュールを作成する場合，そのために新人は設備オントロジーを頻繁に参照することになる．

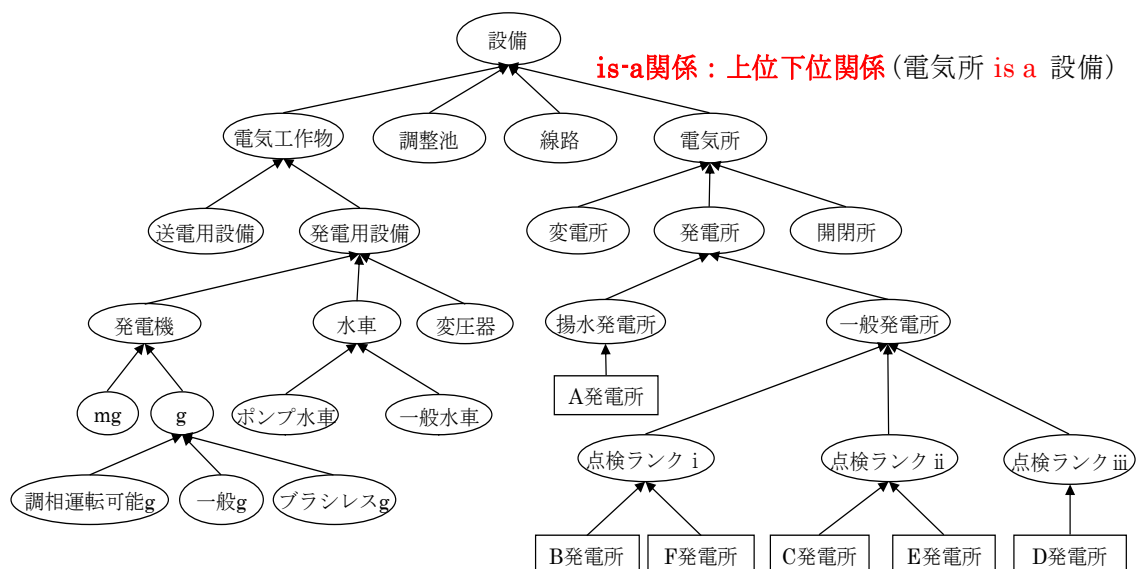


図 30 設備オントロジー

次に作業オントロジーを

図 31 に示す。これは作業の内容によって分類したものである。スケジュール化される作業はインスタンスであるが、それは、発電所、時期などにより特定され、作業の指定はあくまでクラスレベルである。また、そのスケジュール化のための業務ルールも、作業に関してはあくまでクラスに依存する。そのため、実際の停止スケジュールの作成においては、業務ルールに直接的に記述されている作業に関する情報のみで十分な場合が多く、作業オントロジーを参照する必要がある場合は多くない。ただし、すべての業務ルールを満たす作業スケジュールが存在しないような場合など、作業スケジュールの作成が困難な場合には、作業オントロジーを参照し、作業の重要度などを理解することにより、一部の業務ルールを緩和して、実行可能な作業スケジュールを作成することが可能になる。

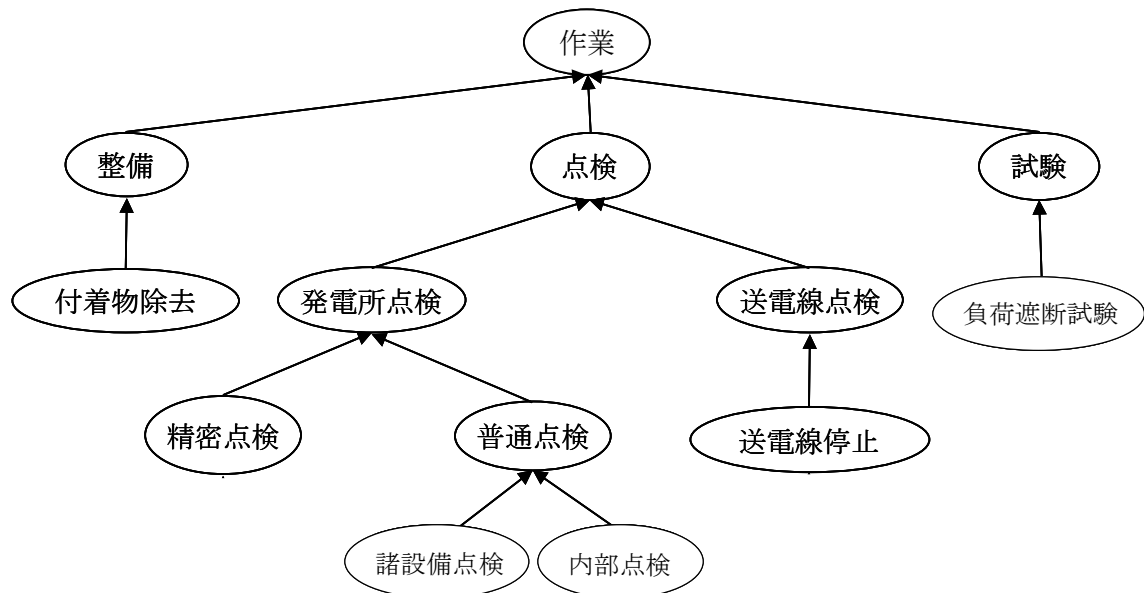


図 31 作業オントロジー

4.2.1.4 業務プロセス・フローとオントロジーの関係づけ

新人は、業務プロセス・フローにより「停止スケジュール」を作成するということは理解できるが、どのようにすれば「停止スケジュール」が作成できるのかは理解できない。そこで、各オントロジーを利用して実際に停止スケジュールの作成が出来るよう、業務プロセス・フローと各オントロジーのプリミティブを結ぶものとして、図 32 エラー! 参照元が見つかりません。に示すように、発電所別スケジュール化作業の一覧を設けた。これは、各発電所の各スケジュール化作業に対し、設備オントロジー、作業オントロジーにリンクをとると同時に、その作業のスケジュール化に適用するルールに対し、ルール・オントロジーにリンクをとっている。これにより、新人は業務プロセスから発電所別スケジュール

化作業を参照し、各作業のスケジュール化に必要となるオントロジーを参照しながら、「停止スケジュール」を作成することが可能になる。「発電所別スケジュール化作業」と業務プロセス・フロー、各オントロジーの関係を図 33 に示す。

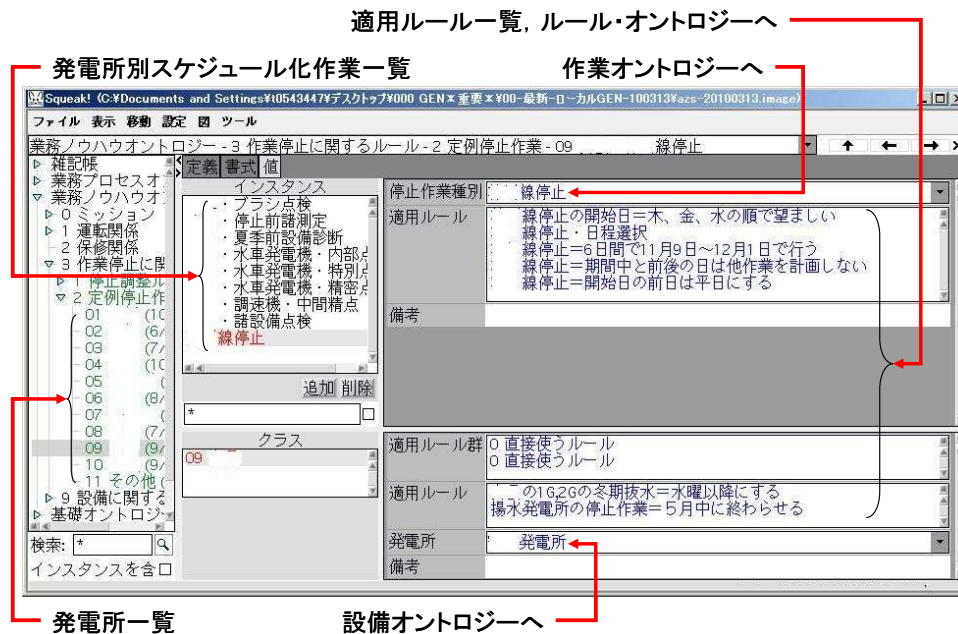


図 32 発電所別スケジュール化作業一覧

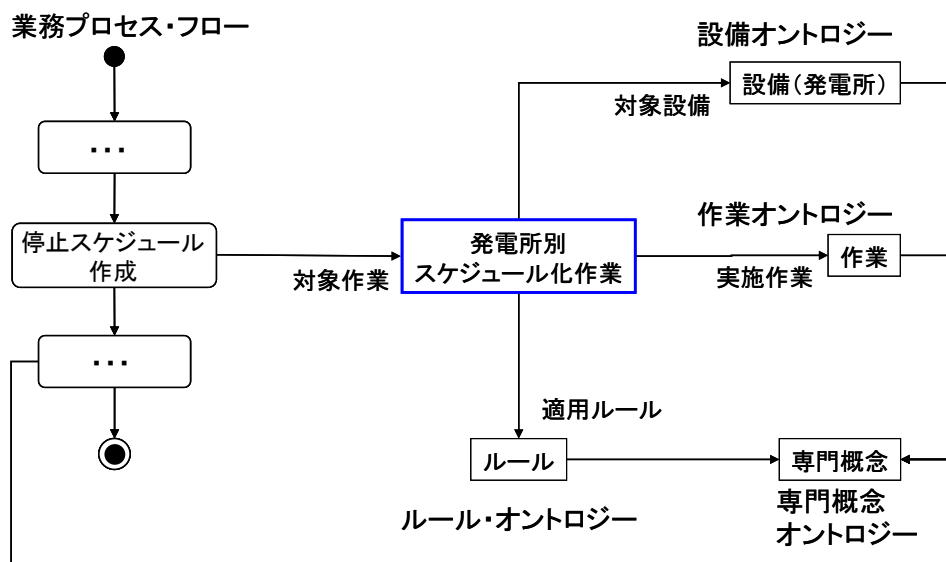


図 33 発電所別スケジュール化作業とオントロジーなどとの関係

4.2.1.5 スケジューリング・システム構築

本モデルケースにおける本提案のルールベース・システムとして、停止スケジュールを作成するスケジューリング・システムを構築した。スケジューリング・システムの構築にあたっては、最初に 90 個の「最も浅いルール」を Prolog 上の実行ルールに変換し Prolog を用いて開発し、それを、Squeak の有制限約充足ライブラリである Backtalk を利用して、GEN 上の移植した。

知識には、一般に成立する正しい知識（事実知識）と、常に成立するとは限らないが、熟練者の経験上多くの場合に成り立つ知識（経験知識あるいはヒューリスティックス）があるが、知的熟練から洗い出された「最も浅いルール」の多くは経験知識である。一方、事実知識に関しては、例えば、「停止調整業務上の休日とは土曜日、日曜日や祝日である」、「作業開始日、作業終了日と作業日数との関係」などの基本的なものは、業務ルールとしては洗い出されていなかった。人間が停止スケジュールを作成する場合は、これらの基本的な事実知識は「常識」であり、業務ルールとしては洗い出されている必要はないが、ルールベース・システムは、これらの基本的な事実知識も必要となるため、暗黙の内に前提としていた基本的な事実知識も洗い出し、Prolog 上のルールベースとして作成した。

また、今回対象としている停止調整業務は、状況によってはルールが相互に矛盾する場合もあり、必ずしもすべてのルールを満たす解が存在するとは限らず、状況に応じて、一部のルールの適用を除外する必要がある。単純化した 1 例を挙げると、

ルール 1：作業 T の開始日は水曜日が望ましい。

ルール 2：作業 T の開始日は 3 月 1 日が望ましい。

という 2 つのルールがあった場合、停止スケジュールを作成する年度において、3 月 1 日が水曜日でなかった場合、ルール 1 とルール 2 のどちらのルールを適用除外するかという問題が発生する、現実の問題においては、このような状況がより複雑な形で多数発生した。

当初は、熟練者へのインタビューにより、個々のルールが相互に矛盾する個々の状況に応じて、適用除外するルールを決定していったが、可能性のあるすべての状況を洗い出し、それぞれの状況に対して適用除外するルールを決定することは、組み合わせの数が増大になり多くの労力を要するとともに、そのメンテナンスを考えると現実的ではなかった。そのため、熟練者へのインタビュー結果を考慮しながら、各ルールに統一的に優先度（優先度を表す整数値）を与え、解が存在しない場合は、優先度の低いものから適用を除外していくこととした。これにより、どんな状況においても、優先度の低いルールから適用を除外していくことにより、常に、何らかの解は得られることとなった。

熟練者へのインタビュー結果を考慮し、各ルールに統一的に優先度を与えた後、過去に熟練者が作成した 9 年分の停止スケジュールを利用し、スケジューリング・システムのルールベースを改善していった。具体的には、まず、熟練者が作成した 9 年分の停止スケジュールと同じ条件でスケジューリング・システムにて停止スケジュールを作成し、熟練者が作

成した停止スケジュールを比較し、差異を洗い出した。その差異について、再び、熟練者へのインタビューを行い、必要に応じてルールの追加・修正、更新されたルールの Prolog ルールベースへの変換、また、ルール優先度の変更などをして、再度、スケジューリング・システムにて停止スケジュールを作成し、結果を比較していった。これを以下の図 34 に示されるように繰り返すことによって、スケジューリング・システムのルールベースを洗練させていった。

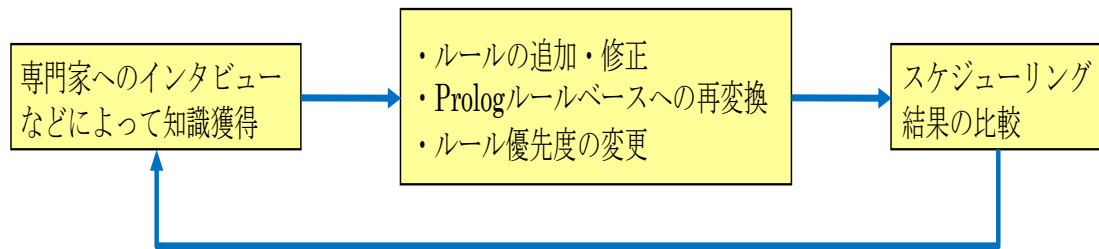


図 34 業務ルールの表出化プロセス

その過程において、各ルールの優先度の修正だけでなく、多くのルールの追加や修正が発生した。追加されたルールは、単純な事実知識や、あるいは、例えば、祝日が土曜日となった場合の扱いなど、経験知識であっても単純なものが多かった。これらは、熟練者には自明であるため、熟練者は表出化を漏らしがちであったが、初心者には必要な業務ルールである。また、ルール・オントロジー上の業務ルールは自然言語で記述されているが故の曖昧さを含んでいて、Prolog のルールベースに変換する際に、誤って明確化されてしまっていたものも存在した。例えば、作業日数1日の作業に関しては、その日を開始日かつ終了日と見なすか否かが曖昧であったが、最終的には、開始日、終了日という概念は作業日数が1日の作業に関しても適用され、その日が開始日かつ終了日になるという解釈に明確化された。

このような過程を経て、Prolog のルールベース・システムの洗練が行われ、結果として、ルールの優先度は15段階に落ち着いた。この洗練されたルールベースを GEN に移植した。GEN では、図 21 (3.7.2.1 参照) に示したように、各ルールの優先度を設定することが可能である。また、左のチェックボックスでルールを適用するかしないかを指定もできる。

GEN のルールベース・システムの利用法について述べる。まず、利用者は図 35 で示される入力表に必要な項目を設定する。点検を行う作業にチェックをつけ、チェックをつけた作業に対して、作業日数を入力する。また、必要に応じて、前回点検日、作業時期（上期/下期）、作業開始月、作業開始日の指定などを行う。



図 35 スケジューリング・システム入力画面

入力表を作成し、適用ルールを決定した後、停止スケジュールの作成を実行すると図 36 のように結果がグラフ表示される。それぞれの発電所ごとに、作業スケジュールが示され、日程を選択するとより詳しい情報を見ることができる。また、スケジューリングを行った際に除外したルールも画面上部の除外ルールという項目を選択することにより確認できるようになっている。

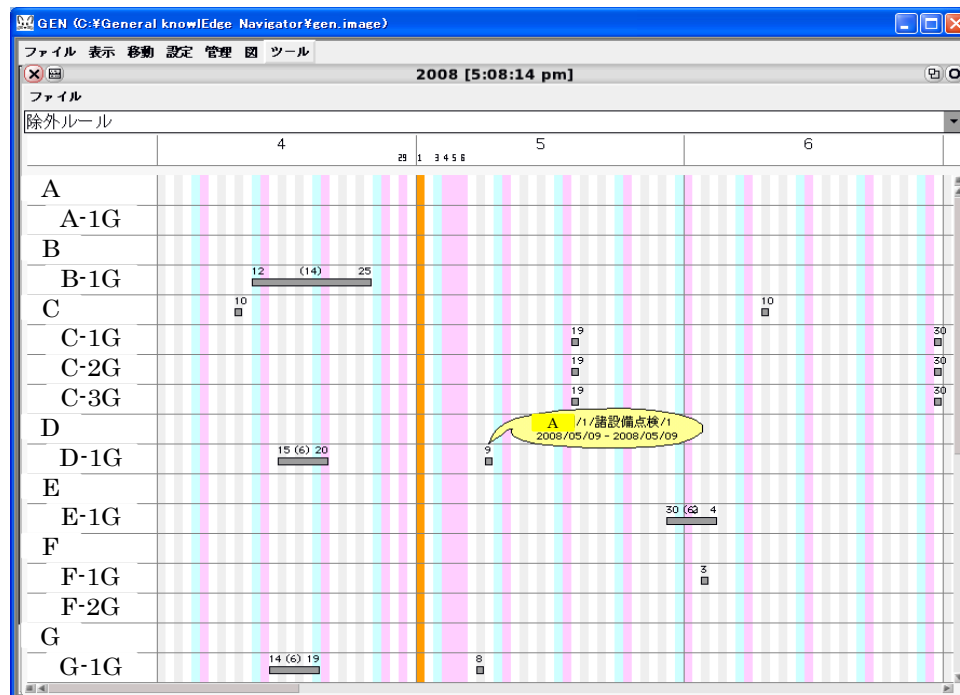


図 36 スケジューリング・システム結果表示画面

4.2.2 新人による内面化に関する評価

4.2.2.1 実験方法

上記により GEN 上に初期構築されたオントロジーとルールベース・システムが、新人による停止調整業務に関する知的熟練の内面化にどのように有効に機能するかを評価するために実験を行った。本来であれば本提案が対象としている新人（3.1 参照）を複数集め、それを本提案により内面化するグループと従来通り OJT により内面化する2つのグループに分け評価実験を行うべきであるが、A 総合制御所の組織は既に非常にスリム化されていて、次の理由により、このようなグループを構成し、評価実験を行うのは困難であった。

- (1) 評価実験を行う時点では対象となる新人が存在しなかったこと。
- (2) 対象となる新人が集められたとしても、その新人に本評価実験のために OJT を行うことが困難であること。

そこで、本評価事実験では、本研究プロジェクトのメンバー5人を被験者として評価を行うこととした。被験者は、知的熟練の獲得方法の観点から、次の3つのグループに分けられた。

グループ 1. 熟練者から直接獲得

このグループの被験者は、プロジェクト開始時からのメンバーで、プロジェクト開始時は停止調整業務に対する知識は保持していなかったが、熟練者と一体となってオントロジーを構築する過程で、熟練者からの講義や熟練者へのインタビューを通して停止調整業務に関する知的熟練を獲得した。このグループの被験者は2名である。（被験者 A, B）

グループ 2. オントロジーをもとに学習

このグループの被験者は、プロジェクトに途中から参加し、ある程度体系化されたオントロジーをもとに、主として、ルールベース・システム開発を通じて、停止調整業務に関する知的熟練を獲得した。このグループの被験者は1名である。（被験者 C）

グループ 3. GEN による独習

このグループの被験者は、停止調整業務に対する知識のない状態で、提案システムを組み込んだ GEN を用い独習により、停止調整業務に関する知的熟練を獲得した。このグループの被験者は2名である。（被験者 D, E）

ここで、グループ 3 の被験者は本提案により知的熟練を獲得する新人に相当し、グルー

プ1の被験者は従来通り OJT により知的熟練を獲得する新人に相当する。グループ3をグループ1と比較することにより本提案の有効性が確認できる。そのために、まず、グループ3の被験者を、本提案が対象とする新人と同等のレベルにする必要があるため、評価実験は次のステップで行った。

(1) 概要説明

まず、グループ3の GEN を用い独習を行う被験者 D, E に対して、熟練者が停止調整業務の概要について4時間説明した。内容は次のとおりである。

- ・ A 総合制御所の位置付け・規模など電力所について
- ・ 業務の目的や種類など停止調整業務について
- ・ 東京電力の内部組織について

以上の内容は、A 総合制御所に配属された新人が事前に持つ、もしくは勤務内で自然に獲得されると思われる知識である。停止調整業務を行うにあたり被験者に新人と同程度の前提知識をつけるために行った。また、この4時間という時間は被験者 A, B, C がプロジェクト参加の初期に熟練者から受けた概要説明の時間と同じ時間となるように決めた。

また、これとは別に GEN の利用方法などに関し次のことも説明した。

- ・ ルール・オントロジー・ドメイン・オントロジー・ルールベース・システムについて
- ・ 大まかな学習の流れなど学習方法について
- ・ オントロジーの概念階層の表示方法など GEN の使い方について

(2) GEN を用いて独習

被験者 D, E は GEN を用いて停止調整業務に関する知識を独習した。学習時間の制限は特に定めず、被験者自らが十分に学習したと判断した時点で学習を終了した。なお、被験者 D, E には、被験者 A, B, C の初期構築を通じての学習時間は開示していない。

(3) ルールベース・システムを利用した知識の定着

被験者 D, E は、GEN に組み込まれているオントロジーを用いながら、4年分の練習問題を解き、自身が作成したスケジューリング結果とルールベース・システムの結果を照らし合わせ、間違っただ所について知識の抜けや勘違いがなかったか GEN を利用して確認し、学習を深めた。なお、この時間は上記の(2)の学習時間に含まれている。

(4) 評価問題解答

被験者5人全員が実際に停止スケジュール(2年分、計130問)を、ルールベース・システムを用いずにハンドシミュレーションによって解いた。作成する停止スケジュールは、予め、熟練者により停止スケジュールを作成済みのものであるが、可能なスケジュールは必ずしも一通りとは限らないため、正解／不正解は熟練者により個別に判断した。

4.2.2.2 実験結果

評価実験の結果を以下の表 3 に示す。なお、誤答に関しては、「記述の欠如」、「記述の曖昧さ」、「ケアレスミス」という 3 つの原因別に分類した。（各原因別の誤答の詳細については、次節を参照のこと。）また、それをグループ別に集約したものを表 4 に示す。

グループ 3 の GEN により独習した被験者は、グループ 1 の熟練者から直接知的熟練を獲得した被験者と比較して、およそ 1/4 の学習時間（GEN による独習者：平均 47 時間、熟練者から直接獲得した被験者：平均 199 時間）で、ほぼ同じ正答率（GEN による独習者：平均 88%，熟練者から直接獲得した被験者：平均 91%）を挙げ、提案システムを組み込んだ GEN を用いて学習することで、熟練者から継承するよりも短い時間で、同程度の正確さで業務をこなすことができるようになることが限られた被験者においてではあるが確認され、提案システムの有効性が支持された。（グループ 1, 3 間の差の有意性の統計的検定に関しては補遺 I を参照のこと。）

表 3 実験結果

グループ		1		2	3	
被験者		A	B	C	D	E
学習方法	知識の獲得	熟練者から直接獲得		オントロジーによる学習	GENによる独習	GENによる独習
	内在化	オントロジーの構築	ハンドシミュレーション	ルールベースシステムの構築	GENによる独習	GENによる独習
学習時間(時間)		198	200	138	55	39
回答時の参照情報		GEN	個人資料	個人資料	GEN	GEN
回答時間(分)		330	255	255	320	250
正答率(%)		90	92	96	88	88
誤答数(／130問)		13	10	5	15	16
原因別 内訳	記述の欠如	0	0	0	3	5
	記述の曖昧さ	5	5	1	5	3
	ケアレスミス	8	5	4	7	8

表 4 原因別の誤答数

グループ		1	2	3
被験者		A, B	C	D, E
学習方法	知識の獲得	熟練者から直接獲得	オントロジーによる学習	GENによる独習
学習時間(時間)		199	138	47
平均回答時間(分)		293	255	285
平均正答率(%)		91	96	88
平均誤答数(／130問)		11.5	5.0	15.5
原因別 内訳	記述欠如	0.0	0.0	4.0
	記述の曖昧さ	5.0	1.0	4.0
	ケアレスミス	6.5	4.0	7.5

また、グループ2のオントロジーにより学習した被験者も、GENによる独習ではないが、グループ1の熟練者から直接的熟練を獲得した被験者のおよそ2/3の学習時間（オントロジーにより学習した被験者：138時間、熟練者から直接獲得した被験者：平均199時間）で、熟練者から直接的熟練を獲得した被験者を上回る正答率（オントロジーにより学習した被験者：96%、熟練者から直接獲得した被験者：平均91%）を挙げた。このことは、GENというツールとは独立に、オントロジーによる体系的な学習の有効性を示している。なお、オントロジーにより学習した被験者が、熟練者から直接的熟練を獲得した被験者を上回る正答率を得たことに関しては、次節の考察の中で、改めて、分析する

4.2.2.3 考察

GENにより独習したグループ3はグループ1との比較において平均誤答数に有意な差は認められなかったが、グループ3に特徴的な誤答がある可能性があり、それを分析することにより本提案のさらなる改善をすべく、誤答の原因分析を行った。誤答の原因は、「記述欠如」、「記述の曖昧さ」、「ケアレスミス」の3つに分類された。「記述の欠如」とは、回答時の参照情報にスケジューリングに必要な知識が記述されていなかった場合である。「記述の曖昧さ」とは、回答時の参照情報に記載されたルールの記述が曖昧であったために本来の意味とは異なる意味でとらえられてしまった場合である。「ケアレスミス」とは、ルールの適用を漏らした場合などである。

特徴的な点としては、グループ3のGENにより独習した被験者には、他のグループにはない誤答として「記述の欠如」による誤答（表4 原因別内訳 記述の欠如）が平均4.0件あったことが挙げられる。具体的にはGEN上のオントロジーへの記述が欠如していた点は次の2点である。

(1) 業務プロセスに関する知識の欠落

すべてのルールを満たす解が存在しない場合、ルール優先度の低いルールの適用を除外して改めてスケジュールを作成することになるが、グループ3のGENにより独習した被験者の1名は適用を除外すべきでないルールを除外し、結果として作成されたスケジュールが誤っている部分があった。具体的には、

図37に示されるように、B送電線停止は11月中旬にスケジュールされるべきものであったが、A発電所の諸設備点検他多くの作業との関係で11月末にスケジュールされている。A発電所の諸設備点検他多くの作業、B送電線停止とも11月中旬に実施すべきものであるが、一方、体制上の制約からこれら作業を同時に実施はできないという制約もあり、これらの作業のうち一部は11月中旬に実施するという業務ルールの適用を除外する必要がある。通常、熟練者は、作業のスケジュール順を意識していて、B送電線停止は、いつ水への影響

[illegible]

ラン ク		月																												
		日	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
		曜日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火					
	発電所	号機																												
揚水	A	1																												
		2																												
		3																												
		4																												
B送電線																														

(2) 作業に関する包含関係の知識の欠落

69

あるが、精密な点検は簡易な点検を兼ねるということが明示的に記述されていなかった。このような作業間の包含関係は、作業オントロジーに記述されるべき内容であったが、多少でも業務のバックグラウンドを持つ者には常識であることから記述はされず、スケジュール・システムの構築においては、ルール・オントロジー上の業務ルールには表れない事実知識として追加されていた。そのため、スケジュール・システム上は正しく処理されるが、その「精密な点検は簡易な点検を兼ねる」という知識はグループ3のGENにより独習した被験者には意識されることはなく、グループ3のGENにより独習した被験者は精密な点検のすぐ後に、改めて簡易な点検をスケジュールするという初歩的な過ちを犯した（図38参照）。今回のオントロジーは、東京電力の現場業務への適用という観点から作成されたが、業務のバックグラウンドを全く持たない新人に対しては必ずしも十分とは言えない面が明らかになった。

グループ3の被験者の作成した停止スケジュール

		月	4月																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
ラン ク		日	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	発電所	号機	曜日	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	休	木																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
揚水	A	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						

正しい停止スケジュール

		4月																															
ラン ク	発電所	月	日	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		曜日	号機	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	休	木
揚水	A	1																															
		2																															
		3																															
		4																															
		5																															
		6																															

図 38 作業規模に関する知識の欠落による誤答

また、「記述の曖昧さ」に起因する誤答（表4 原因別内訳 記述の曖昧さ）は、グループ1の熟練者から直接知的熟練を獲得した被験者は平均5.0件であるのに対し、グループ3のGENにより独習した被験者は平均4.0件で大きな差はなかったが、グループ2のオント

ロジーにより学習した被験者は1.0件であった。この誤答は、日本語での業務ルールの記述に曖昧性があり、本来とは異なる意味で解釈されてしまったことに起因したものである。グループ2のオントロジーにより学習した被験者はルールベース・システムの開発を担当し、実行可能なルールベースを構築の中で、自然言語で記述されたルールの曖昧さを排除する必要がある、それを通じて、日本語での業務ルールの本来の解釈を確認し、獲得していたために、この誤答が少なかったものと考えられる。

ケアレスミスに関しては、グループ1の熟練者から直接知的熟練を獲得した被験者は平均6.5件であるのに対し、グループ3のGENにより独習した被験者は平均7.5件で大きな差はなかった（表4 原因別内訳 ケアレスミス）が、グループ2のオントロジーにより学習した被験者を含め、回答時の参照情報をGENから得たグループ（被験者A, D, E）とGENではなく紙による個人資料を参照して回答したグループ（被験者B, C）と分けると、表5の通り、ケアレスミスの件数は前者が7.7件に対し後者は4.5件となり明確な差が生じ、また、回答時間に関しても前者が300分であるのに対し後者では255分と、GENを参照して回答した場合、個人資料を使用した場合に比較して、ケアレスミス、回答時間とも増大する結果となった。

表5 回答方法別実験結果

回答時の参照情報	GEN	個人資料
被験者	A, D, E	B, C
平均回答時間(分)	300	255
平均ケアレスミス率(%)	5.9	3.5
平均ケアレスミス数(／130問)	7.7	4.5

その理由としては、

- ・現状のGENは実験システムであり、ユーザインタフェースの作り込みが十分ではなく、特に、情報の一覧性に欠けること
- ・この規模の情報（業務ルール：134個、内、最も浅いルール：90個）では、紙資料の方がコンピュータ・ディスプレイよりも一覧性に優れること
- ・万人向けのGENの記述は、各個人のバックグラウンドを反映した個人資料に比べると、各個人からみた可読性は劣ること

などが挙げられる。逆に、GENのこのような弱点にも拘わらず、グループ3のGENにより独習した被験者が短時間の学習でグループ1の被験者と有意な差のない正答率を挙げていることは、本提案の有効性を補強するものとも考えられる。

4.2.2.4 GENにより独習した被験者によるオントロジーに対する評価

ここでは、グループ3のGENにより独習した被験者による、構築したオントロジーに対する評価について述べる。

(1) ドメイン・オントロジー

今回は主に設備オントロジーと作業オントロジーが利用された。作業オントロジーは、作業の性質や特性を知ることができたため、作業の全体像を把握するのに有用であったが、作業の内容や規模などの情報が欠落していたため、前項で挙げたような誤答が生じた。設備オントロジーは、停止作業の対象である設備とその上位下位関係を体系的に学習することができ、スケジューリングの実行理解に役立った。発電所固有の分類が存在し、分類によってルールが変わるケースがあったので、その部分の理解を助ける働きを持っていた。だが、直接スケジューリングに関係しない部分が多く含まれているため、必要な箇所を調べるのが困難だったという指摘があった。

(2) ルール・オントロジー

ルール・オントロジーは、ルールを適用した理由を知りたい場合に、最も浅いルールからその正当性を示すルールをたどっていくことで、ルールとその理由まで理解できた。しかし、理由や最も深いルールである業務目標ごとにルールのグループ化がなされていれば、よりルールの全体像の理解につながるのではないかという指摘があった。

4.2.3 業務プロセス・フローの詳細化と作業オントロジーの再構築

以上の評価に基づき、業務プロセス・フローの詳細化と作業オントロジーの再構築を行った。

4.2.3.1 業務プロセス・フローの詳細化

実験結果から、GENには作業のスケジュール順などの業務プロセスに関する知識が不足していることが明らかになった。しかし、他GENにより速習したグループ3以外の被験者は、スケジューリングをする際は作業ごとにスケジュールを決定していき、その順番もある程度決まっているという知識を熟練者から得ていて、それを用いてスケジューリングしていた。

そこで、その知識をもとに、作業のスケジューリング順がどのように決められているかを分析した結果、作業のスケジューリング順に関し次の3つのルールが導かれた。

(1) 作業の規模順

作業日数が長いもの、作業対象となる設備が多いもの、また作業対象となる設備の規模が大きいものなど作業の規模が大きくなるものほど優先的にスケジューリングしている。理由としては、電力供給上重要であることや、いっ水の量に対して影響が大きいことが挙げられる。また、作業員の確保などを考え、作業をなるべく重ねないようにスケジューリングをしていくために、作業日数が長い作業や、多くの設備で行う作業や設備の規模が大きい作業を優先してスケジューリングすることで、候補日程がなくなり、もう一度スケジューリングをやり直すことがないようにしている。

(2) 作業の実施時期順

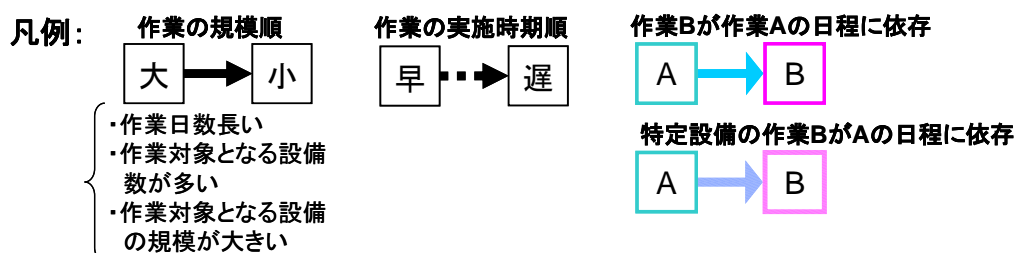
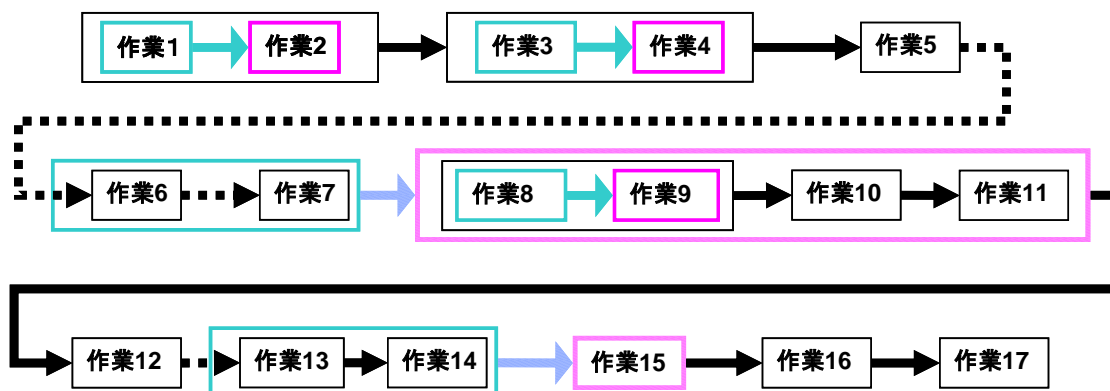
個別の事情により実施時期が指定されている作業は他の作業よりも優先されていて、さらに、その中でもより実施時期の早いものほど優先的にスケジューリングしている。

(3) 作業日程に依存関係による順番

作業 A の日程が決まってからでないと作業 B の日程が決められない場合は、作業 A を作業 B より優先的にスケジューリングすることが必要である。この中には、すべての作業 B の日程が作業 A の日程に依存する訳ではなく、特定設備の作業 B の日程だけが作業 A の日程に依存する場合もある。

これらの作業のスケジューリング順に関する 3 つのルールに従って、すべての作業に関してスケジュール順を整理したところ、以下の**エラー! 参照元が見つかりません**。に示すように 17 種類の作業間のスケジュール順となった。凡例に示す通り、**エラー! 参照元が見つかりません**。で黒の矢印が上記の「(1) 作業の規模順」に基づく順番を表し、黒の波線の矢印は上記の「(2) 作業の実施時期順」に基づく順番を表す。また、青緑の矢印は、上記「(3) 作業日程に依存関係による順番」に基づく順番で日程が依存する作業が特定の設備に限定されないものを表し、青の波線の矢印は、上記「(3) 作業日程に依存関係による順番」に基づく順番で日程が依存する作業が特定の設備に限定されるものを表す。これは、業務プロセスと同時に、その業務プロセスとなる理由も理解できるモデルになっている。今回は作業を 17 種類に分類し、それらの間に順序を定めたが、状況が変わったとしても、3 つの視点を考慮することで臨機応変に順番を組みなおすことが可能である。

この検討結果に基づいて、図 40 のように業務プロセス・フローをもう 1 段詳細化した。**4.2.1.4 エラー! 参照元が見つかりません**。において、発電所別スケジュール化作業の一覧を通じて、業務プロセスと各オントロジーを結び付けていることはすでに述べたが、評価実験時の業務プロセス・フローでは、「中期主要停止作業計画」というプロセスに発電所別スケジュール化作業すべてが繋がり、それを通じて「最も浅いルール」すべてを参照することになり、実質的にはあまり意味を持たなかったが、改善後の詳細化された業務プロセス・フローでは、各業務プロセスが発電所別スケジュール化作業を通じて参照する「最も



浅いルール」が限定されるため、新人にとっては、停止スケジュールの作成手順が格段に分かりやすくなった。

図 39 スケジューリングリファレンスモデル

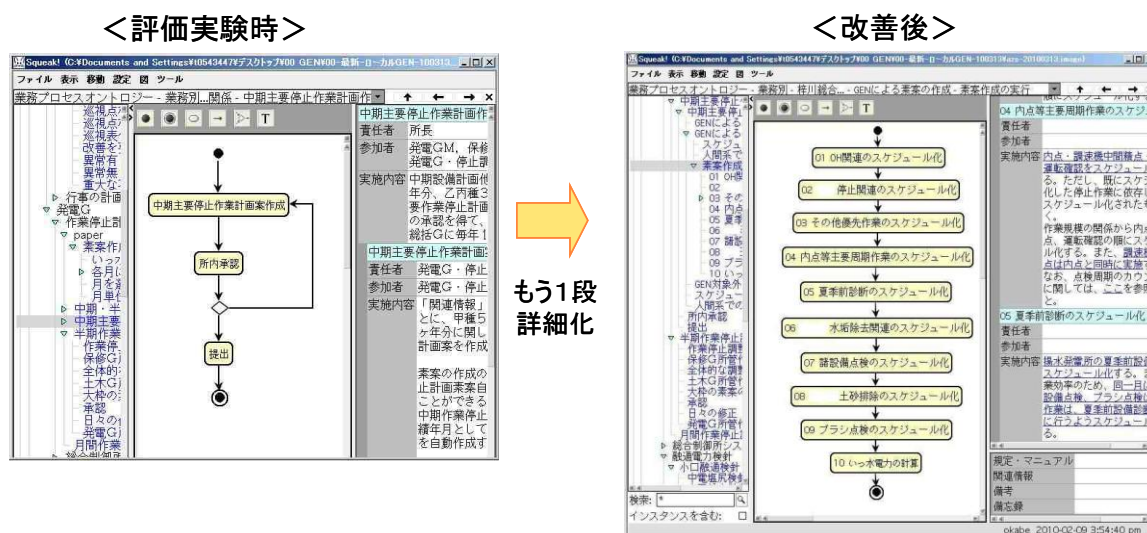


図 40 業務プロセス・フローの詳細化

一方で、このような手順情報を含めることは、そのメンテナンス負荷の増大も危惧され

る。そこで、この詳細化した業務プロセス・フローを、熟練者に評価してもらったところ、次のような肯定的な評価を得た。

- ー 熟練者自身は、初期のスケジュールを作成する場合このような手順は意識していない。
- ー ただし、新人がスケジュールを作成する上では参考になる。
- ー この手順は、いっ水電力への影響の大きいものを優先的にスケジュール化するようになっていて、スケジュール対象作業に大きな変更がない限り、汎用的で安定している。
- ー また、スケジュール対象作業に大きな変更が生じた場合も、いっ水電力に大きな影響を与えるものを優先するという基本的考えに則れば、更新することは困難ではない。

4.2.3.2 作業オントロジーの再構築

実験結果とグループ3のGENにより独習した被験者によるオントロジーの評価から、構築した作業オントロジーには作業の包含関係に関する情報が欠落していることが明らかになった。しかし、他GENにより速習したグループ3以外の被験者は熟練者から作業の包含関係に関する知識も獲得していた。そこで、再び熟練者へインタビューすることによって、作業の包含関係に関する情報を整理し、作業オントロジーを再構築した。これまで作業オントロジーは上位下位関係(is-a)で構築されていたが、この上位下位関係(is-a)はあくまで作業の一般的な分類体系に従っていた。構築し直した作業オントロジーにおいては、上位下位関係(is-a)に対して明確な意味を持たせた。すなわち、作業をどのような作業内容を持つかにより特徴付けられるクラスとして認識する。すると、下位の作業は上位を作業の特化であるから、上位の作業の特徴付ける作業内容を引き継ぐことになる。このような考えに従って再構築した作業オントロジーの一部を図 41 エラー! 参照元が見つかりません。に示す。例えば、ブラシ点検は発電機のブラシの点検であり、諸設備点検は発電機のブラシを含む諸設備に関する点検である。諸設備点検の点検作業は、ブラシ点検の点検作業を含むから、諸設備点検はブラシ点検のis-aになる。これを業務的に言えば、諸設備点検はブラシ点検を兼ねることを意味する。これによって、新人は作業間の包含関係を把握できるようになり、今回の実験で起こったような間違いを防ぐことができる。

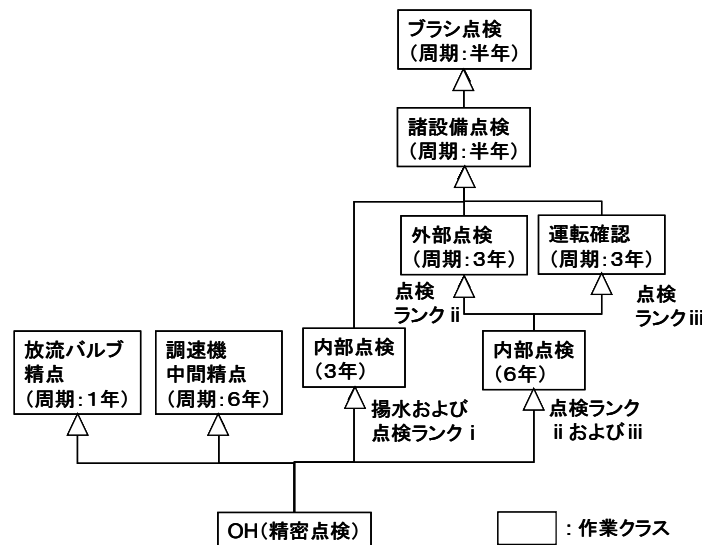


図 41 再構築した作業オントロジー

4.3 熟練者による持続的な表出化・連結化に関する検証

4.2 において、いくつかの改善すべき点が明らかになったものの、構築されたオントロジーやルールベース・システムが停止調整業務の知的熟練の新人による内面化に関して有効であることが確認された。次のステップとして、この構築されたオントロジーやルールベース・システムが、経営環境の変化に伴い、事業所の長の意思に従って、熟練者により持続的に更新されていくための検討とその検証を行う。

4.3.1 事業所の長の意思を反映させるためのルール・オントロジーの

改善

熟練者が経営環境などの変化に応じて、更新すべき業務ルールを把握するという観点から見た場合、最も深いルールそのものは普遍性が高く、多くの場合、経営環境などの変化により業務目標が変化するとしても、最も深いルールそのものが入れ替わるのではなく、既存の最も深いルールの重要度に変化が生じることが想定された。一方で、浅いルールと深いルールの関係は、必ずしも抽象度が統一されてなく、また、多くの場合多対多の関係になっていて、最も深いルールの重要度に変化が生じた場合に、それに追随すべき浅いルールの把握が必ずしも容易ではなかった。

そこで、まず業務ルールを追加・整理して正当性関係の抽象度を揃えること、最も強く

正当性を説明する業務ルールを明示することの2つを方針でルール・オントロジーを改善した。

4.3.1.1 正当性関係の抽象度の統一

業務の遂行に直接使われる「最も浅いルール」は、その正当性を説明する「深いルール」を順に辿ることにより業務目標を表す「最も深いルール」と関連づいている。ただし、ルール・オントロジーのプリミティブとして洗いだされた業務ルールの抽象度が統一されていなかったため、「浅いルール」と「深いルール」の間の正当性関係に意味的な飛躍があり、その関係の理解が新人には困難なものもあった。

そこで、業務ルール間の正当性関係に意味的な飛躍がある場合には、例えば、以下の図42のように、その「浅いルール」の正当性を説明し、また、「その深いルール」から正当性を説明される中間の業務ルールの追加を行った。

特に、「最も深いルール」が直接的に「最も浅いルール」の正当性を説明している場合には、その中間に、その「最も深いルール」によりその正当性を説明され、また、その「最も浅いルール」の正当性を説明する「深いルール」を最低一つ設けるようにした。

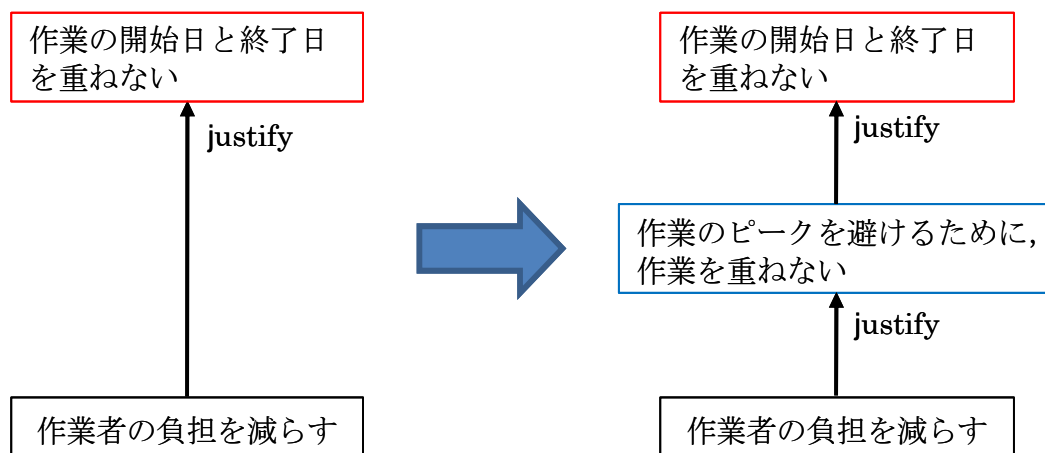


図 42 中間の業務ルールの追加

4.3.1.2 最も強く正当性を説明するルールの明示

一般にある業務ルールは、その目的・意義などが単一ではないため、複数の「深いルール」により正当性を説明される。ただし、その正当性の説明度合いは様々であり、そのことが、正当性関係の理解を難しくしていた。そこで、ある業務ルールに対し、その正当性を説明する「深いルール」が複数の存在する場合には、その中で、最も強く正当性を説明している「深いルール」を一つ明示することにした。

例えば、以下の図 43 に示されるように、「祝・休日をなるべく避ける」というルールは、「効率的に行う」というルールと「体制を確保する」という二つの「深いルール」により正当性を説明されている。ここで、熟練者へのインタビューを通じて、「体制を確保する」というルールがより強く正当性を説明していることが明らかになったため、最も強く正当性を説明している「体制を確保する」というルールとの正当性関係を実線、「効率的に行う」というルールとの正当性関係を点線で表現し、最も強く正当性を説明する「深いルール」を識別できるようにした。

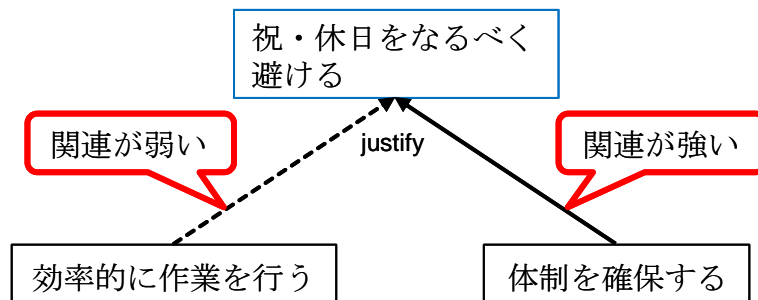


図 43 最も強く正当性を説明するルールの明示

それにより、各「最も浅いルール」から見て、それを最も強く正当性を説明する「最も深いルール」が一つに定まるようになり、また、「最も深いルール」から見ても、それが正当性を説明している主な「最も浅いルール」が絞られ、把握が容易になった。例えば、4.2.1.2 の図 28 の水垢除去関係のルール・オントロジーに関しては、図 44 エラー! 参照元が見つかりません。に示されるように「水垢除去を水垢付着度合いに応じて行う」を最も強く正当化しているルールは、「設備の効率性を維持する」ではなく、「電力供給信頼度を維持する」である。これは、水垢付着は、単に設備の効率性を低下させるだけでなく、設備の停止の原因ともなり、電力供給信頼度にも影響を及ぼすことを示している。

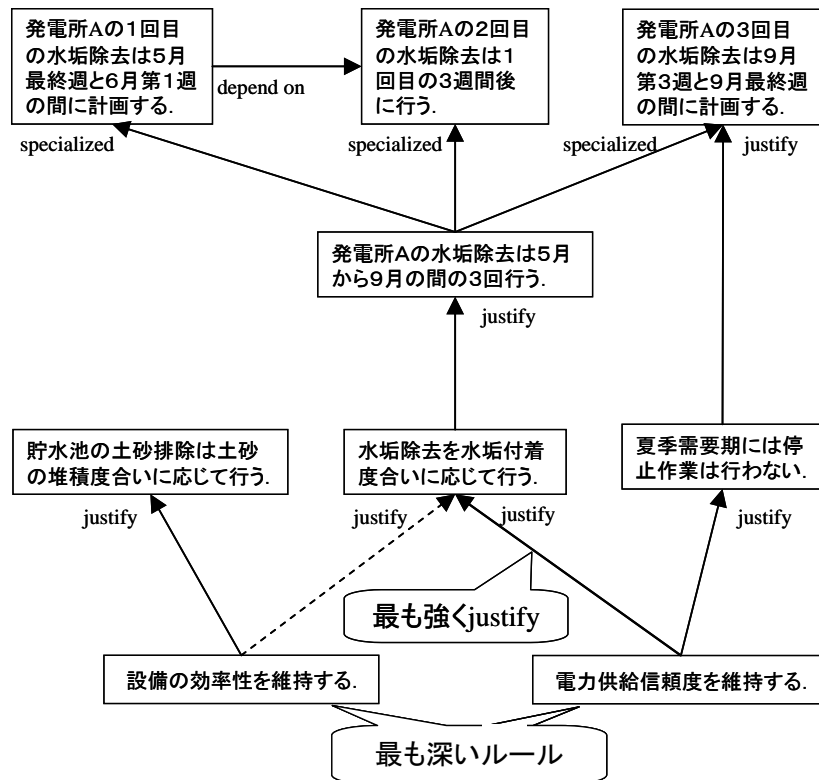


図 44 水垢除去を最も強く justify する最も深いルール

4.3.2 ドメイン・オントロジーからの日本語ルール生成と実行ルールへの変換

以上のルール・オントロジーの改善により、熟練者が、経営環境の変化に伴い、事業所の長の意思に従い、更新すべき「浅いルール」を把握することが容易になったが、更新すべき「浅いルール」を把握できたとしても、熟練者が、それを、対応するルールベース上の実行ルールを含めて、適切に更新していくことは、必ずしも容易ではない。そこで、ルールベース・システムに含まれる語彙についてドメイン・オントロジーで定義することによって、日本語で記述された「最も浅いルール」から、ドメイン・オントロジーを参照して半自動的にルールベース上の実行ルールを生成することで、ルールベース・システムのメンテナンスを支援することを考える。

4.3.2.1 最も浅いルール of 分析

まず、「最も浅いルール」90 個全体に関して、ルールベース上の実行ルールと一体に管理できるよう、詳細な分析をおこなった。

「最も浅いルール」の中には、点検周期に関するルールやいつ水電力をできるだけ少なくするための可能解の探索順に関するルールも含まれている。従前から点検周期は点検周期表により管理されおり、また、可能解の索順に関しては各河川の過去 30 年間平均の月別河川水量表に基づき決定されていた。どちらのルールもこれらの表に基づき管理するのが最も管理しやすい形態と考えられたため、GEN 上でも表に基づき管理し、そこから実行ルールを生成することとした。

その上で、これらのルールを除いた「最も浅いルール」72 個を、熟練者自らがルールベース上の実行ルールと一体に無理なく維持・管理できるようにするために、類別化した。まず、ルールを単独作業ルールと相互依存作業ルールに分類した。例えば、「A 発電所の B 作業は 12 月に実施する」というルール（注：本論文で例示するルールはすべて説明用に簡略化したものであり、実際のものとは異なる。）は A 発電所の B 作業に関する単独作業ルールである。また、例えば、「（電力需要の高い）夏季は（電力供給信頼度を維持するために）停止作業は行わない」といったルールも、夏季に実施されうるあらゆる作業が対象となるが、それらの個々の作業に対し独立に適用されるルールであるから、単独作業ルールである。一方、「A 発電所の B 作業と C 作業は同時には実施しない」といったルールは、A 発電所の B 作業と C 作業の相互依存の制約を規定している。このようなものは、相互依存作業ルールである。また、相互依存作業ルールには、この例のように対象作業の相互依存性に対称性が成り立つ場合と、「A 発電所の B 作業は C 作業の終了後に実施する」といった対象作業の相互依存性に対称性が成り立たない場合がある。前者を対称ルール、後者を非対称ルールと呼ぶことにする。（単独作業ルール、相互依存ルール・対象ルール、非対称ルールの正確な定義に関しては補遺 II を参照のこと。）

その結果、「最も浅いルール」90 個は以下の図 45 エラー! 参照元が見つかりません。に示すように類別化された。なお、対称ルール、非対称ルールとも、可能性としては、3 つ以上の作業間の多項関係の制約となる可能性もあったが、実際には 2 項関係以上のものはなかった。理論上は多項関係があり得ても実際には 2 項関係までで十分であることが実務上は多いが、この場合も同様であった。

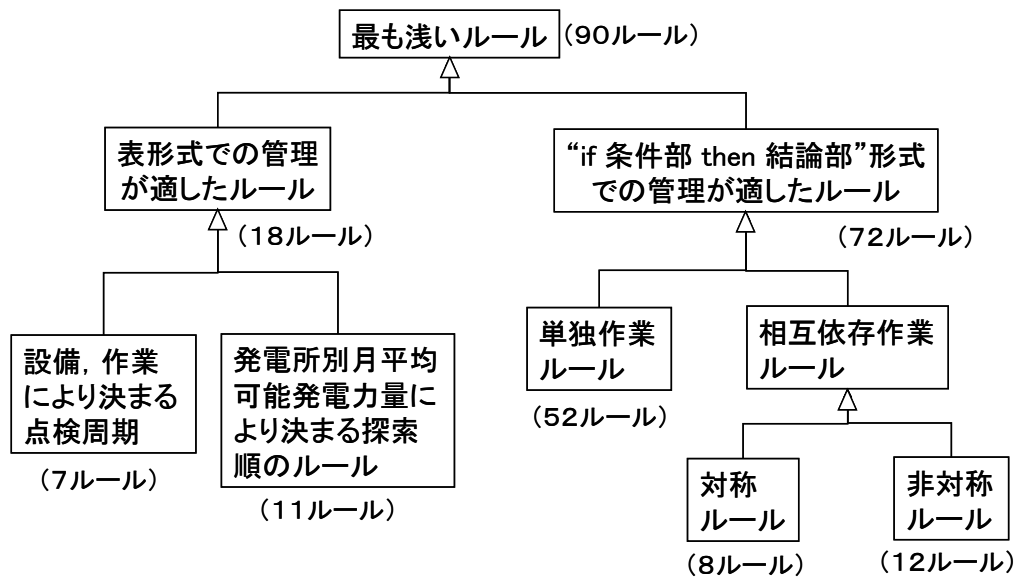


図 45 最も浅いルールの類別化

以上の分類を踏まえ，“if 条件部 then 結論部”形式での管理が適した 72 個のルールを自動的に実行ルールに変換できるよう更にパターン化した。ここで、結論部は設備と作業を指定しただけでは特定されず、その設備のその作業の実施時期、作業日数にも依存する場合がある。そのため、条件部では、設備、作業に加えて、実施時期、作業日数も指定できるようにした。

条件部は、次の各項の or 指定の and 解釈により指定される。

項 1：設備の指定

発電所，発電機 （注：発電所によっては複数の発電機を有する。その場合，発電機も指定することにより設備を識別する。）

項 2：作業の指定

作業，回 （注：作業によっては年に同一設備に対して複数回実施するものがある。その場合，1 回，2 回等，何回目の作業であるかにより識別する。）

項 3：実施時期の指定

上期／下期，月，旬，日の指定 ないし 期間指定

項 4：作業日数の指定

具体的日数の指定 ないし $m+7n$ ($m=1, \dots, 6, n=0$ または任意の自然数)による指定
 （注： $m+7n$ は曜日との関係を意識したもの）

ただし，相互依存作業ルールの場合は，相互依存関係の及ぶ範囲を指定することが必要になる場合がある。対称ルールにおいては，例えば，「A 発電所の各発電機に対する B 作業

が同一月に実施される場合は、（作業効率の観点から）連続して行う」というルールにおいて、連続して行うという制約の対象となる作業は、A 発電所の各発電機の B 作業であり、かつ、実施時期は何月でも構わないが同一月にあるものである。そこで、各指定項に対し、例えば、次のように同一であることを指定できるようにした。

項 1：設備の指定

発電所=A 発電所

項 2：作業の指定

作業=B 作業

項 3：実施時期の指定

月=同一

また、相互依存ルールの内、非対称ルールに関しては、複数の対象作業の集合を指定する必要が生じるが、前述のようにルールは現実には 2 項関係までであるので、対象作業の指定は 2 つまで可能とし、それらの間の相互依存関係の及ぶ範囲は対象ルールの場合と同様にして指定することにした。例えば、「A 発電所の B 作業は同一月に C 発電所の D 作業がある場合は、C 発電所の D 作業の後に行う」の場合は、次のようになる。

対象作業 1

項 1：設備の指定

発電所=A 発電所

項 2：作業の指定

作業=B 作業

項 3：実施時期の指定

月=同一

対象作業 2

項 1：設備の指定

発電所=C 発電所

項 2：作業の指定

作業=D 作業

項 3：実施時期の指定

月=同一

以上のように条件部を類別化した上で、結論部の類別化を行った。結論部は、大きく次の 7 種類に類別された。

- (1) 期間指定
 - (2) 除外期間指定
 - (3) 曜日指定
 - (4) 重ねない
 - (5) 同時に行う
 - (6) 連続して行う
 - (7) その他
-

最初の3つ「期間指定」「除外期間指定」「曜日指定」は「単独作業ルール」に関する結論部である。ここで、「期間指定」「除外期間指定」共に、開始日に関するもの（例：「4月上旬までに開始する」）、終了日に関するもの（例：「5月末までに終了する」）、全期間に関するもの（例：「4月15日～20日の間に実施する」）など、いくつかのパターンが存在する。「曜日指定」は、作業日数に応じて、作業工程上の作業輻輳日との兼ね合いから、開始曜日を定めるものである。次の3つ「重ねない」「同時に行う」「連続して行う」は「相互依存作業ルール」に関するものである。ここで、「重ねない」は作業日程を重ねないことを意味しているが、重ねない期間に関し「全期間を重ねない」、（作業が特に輻輳する日である）「開始日・終了日は重ねない」など、やはりいくつかのパターンが存在し、対称性が成り立つ場合と成り立たない場合が存在する。また、「同時に行う」に関しては、作業期間の短い作業を他の作業期間の長い作業の実施中に実施することを意味し常に非対称であり、「連続して行う」に関しては対称、非対称の双方がある。最後の「その他」は個別的な制約であって、他のルールにおいて再利用される可能性が少ないものをここに分類した。この結論部の類別に従って、72個の最も浅いルールを分類したのが表6 エラー! 参照元が見つかりません。である。

		単独作業 ルール	相互依存作業ルール		計
			対称ルール	非対称ルール	
結論部	期間指定	22			22
	除外期間指定	12			12
	曜日指定	17			17
	重ねない		6	4	10
	同時に行う			7	7
	連続して行う		1	1	2
	その他	1	1		2
	計	52	8	12	72

表6 結論部の類別によるルールの分類

さらに、この結論部の類別に基づき、「その他」を除いた各類別のパターンを整理した。「重ねない」のパターンを図46に示す。

4.3.2.2 ドメイン・オントロジーの拡充

ルール・オントロジー

最も深いルール

いっ水電力を少なくする

最も浅いルール

A発電所のB作業とC発電所のD作業は相互に重ねない。

いっ水電力

いっ水

使用

設備オントロジー

A発電所

C発電所

B作業

D作業

作業オントロジー

〜と〜を相互に重ねない

特化

汎化

重ねない

ドメイン・オントロジー

84

4.3.2.3 日本語ルールの生成と実行ルールへの変換

以上を基に、プルダウン・メニューからドメイン・オントロジーの語彙を選択し、日本語ルールを生成し、さらにそれを実行ルールに変換する。

一例として、結論部が「開始日，終了日を相互に重ねない」である対称ルールの場合を示す。なお，本実装に関しては，Prolog のルールベースに対して Java を用いて付加的な機能を追加することにより実現した。その有効性が確認されたため，現在，GEN のルールベース・システムに移植中である。

図 48 がプルダウン・メニューからのドメイン・オントロジーを選択する画面であり，そこから生成される日本語ルールは「上期に実施する作業日数が(3+7n)日の A 発電所の内部点検は開始日，終了日を相互に重ねない」になる。また，そこから生成される Prolog の実行ルールは図 49 のようになる。この実行ルールの生成にあたっては，図 50 に示す対称ルールのテンプレートの一つおよび図 51 に示す「開始日，終了日を相互に重ねない」に対応するテンプレートを使用している。また，条件部を満たす作業を判別するうえで，上期／下期の判定を行う *jiki* など，予めルールベース上に定義されている共通ルール・ライブラリを使用している。

なお，個別的な制約であり表 6 で「その他」に分類された結論部に関しては，その結論部そのものに対応する用語をドメイン・オントロジーに個別に定義しプルダウン・メニューから選択できるようにするとともに，それに対応するルールテンプレートを個別に用意している。

以上の結果，少なくとも，現状の 72 個の「最も浅いルール」に関しては，プルダウン・メニューからの選択により日本語ルールの生成および実行ルールへの変換ができるようになった。また，新たなルールに関しても，多くの場合，既存のパターンを組み合わせることにより日本語ルールが生成され，また，実行ルールへも変換されるものと期待される。

ルール作成

ホームに戻る

条件部

- 設備: [プルダウン]
- 作業件名: 内部点検
- 作業日数: 3+7n日
- 実施時期: 上期

結論部

動詞: 重ねない 分類1: 作業日指定ルール 分類2: ~を重ねない

next step >>

ルール作成

条件部

- 設備: [プルダウン]
- 作業件名: 内部点検
- 作業日数: 3+7n日
- 実施時期: 上期

結論部

開始日、終了日 [プルダウン] を重ねない

next step >>

図 48 実行ルールへの変換のためのプルダウン・メニューによる指定

```

rule87(ResultStock,a発電所,内部点検,3,上期,StartDate,EndDate):-!,
    rule87_2(ResultStock,a発電所,内部点検,3,上期,StartDate,EndDate).

rule87(____):- true.

rule87_2(ResultStock,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    rule87_3(ResultStock,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).

rule87_2(____):- skip_rule87,true.

rule87_3([],____):- true.

rule87_3([[Setubi2,G2,Sagyo2,OtherStartDate,OtherEndDate]]R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    member(Setubi2,[Setubi]),
    member(Sagyo2,[Sagyo]),
    Length_Class2 is (OtherEndDate-OtherStartDate+1)%7,
    member(Length_Class2,[Length_Class]),
    julian_henkan(OtherStartDate,[_,Month2,_]),
    jiki(0,Month2,Jiki2),
    member(Jiki2,[Jiki]),
    !,
    StartDate =/= OtherStartDate,
    StartDate =/= OtherEndDate,
    EndDate =/= OtherStartDate,
    EndDate =/= OtherEndDate,
    rule87_3(R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).

rule87_3([_R],Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    rule87_3(R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).

```

図 49 変換された Prolog の実行ルール

```

rule#xx(ResultStock,#setubi,#sagyo,#length_class,#jiki,StartDate,EndDate):-!,
    rule#xx_2(ResultStock,#setubi,#sagyo,#length_class,#jiki,StartDate,EndDate).

rule#xx(____):- true.

rule#xx_2(ResultStock,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    #xx_3(ResultStock,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).

rule#xx_2(____):- skip_rule#xx, true.

rule#xx_3([],____):- true.

rule#xx_3([[Setubi2,G2,Sagyo2,OtherStartDate,OtherEndDate]]R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    member(Setubi2,[Setubi]),
    member(Sagyo2,[Sagyo]),
    Length_Class2 is (OtherEndDate-OtherStartDate+1)%7,
    member(Length_Class2,[Length_Class]),
    julian_henkan(OtherStartDate,[_,Month2,_]),
    jiki(0,Month2,Jiki2),
    member(Jiki2,[Jiki]),
    !,
    #結論部テンプレート,
    rulexx_3(R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).
rule#xx_3([_R],Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate):-
    rule#xx_3(R,Setubi,Sagyo,Length_Class,Jiki,StartDate,EndDate).

```

注: #で始まる文字列が置換される文字列

共通ルール・ライブラリの利用

図 50 使用された対象ルールのテンプレート

StartDate =/= OtherStartDate, StartDate =/= OtherEndDate, EndDate =/= OtherStartDate, EndDate =/= OtherEndDate

図 51 「開始日, 終了日を相互に重ねない」のテンプレート

4.3.2.4 新たな実行ルールを追加作業の流れ

実際に新たな実行ルールを追加する作業の流れを、画面を示して説明する。

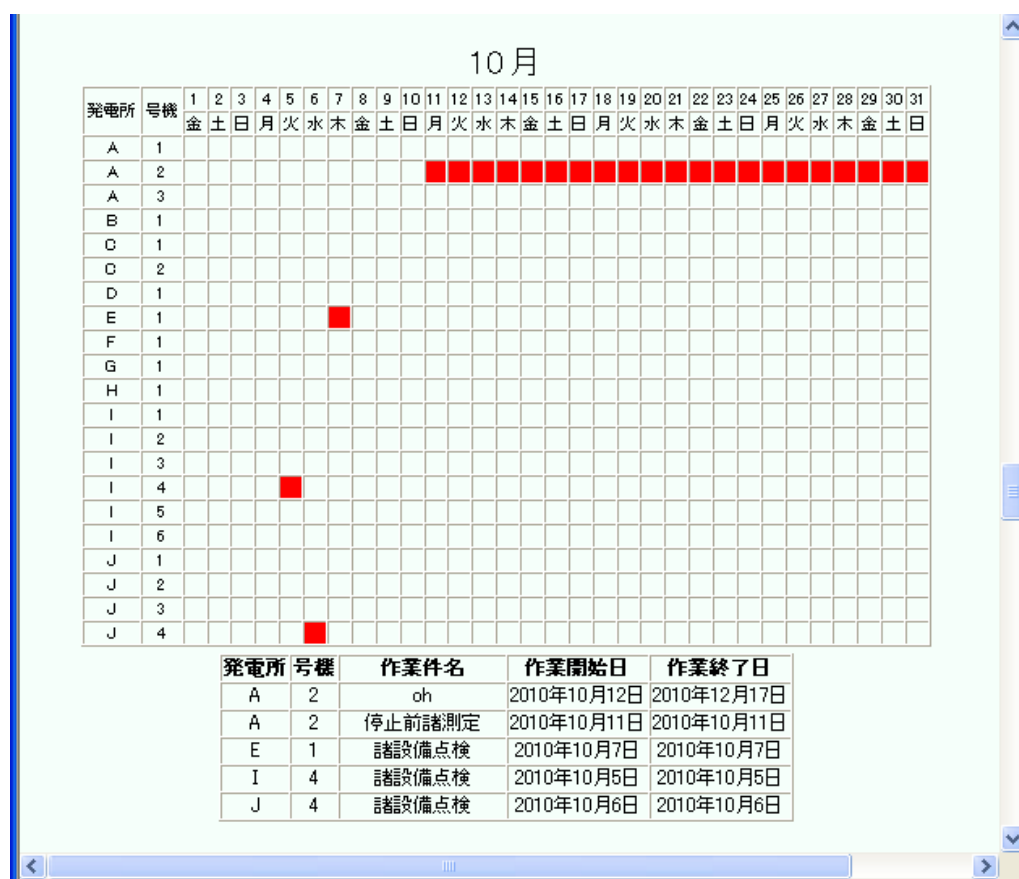


図 52 スケジューリング・システム結果表示（2010 年度）

図 52 エラー! 参照元が見つかりません。は、スケジューリング・システムで作成した 2010 年度の停止スケジュールである。ここに、一例として、「揚水発電所で 10 月に実施する作業は 10 日以降に行う」という単独作業ルールを追加することを考える。なお、揚水発電所は I 発電所と J 発電所であり、このルールの追加により、10 月 5 日にスケジュールされている I 発電所 4 号機の諸設備点検と 10 月 6 日にスケジュールされている J 発電所 4 号機の諸設備点検の作業日程が 10 月 10 日以降に変わることが想定される。

The diagram illustrates the semi-automatic conversion process for creating an execution rule in Prolog. It consists of three sequential screens connected by red arrows.

Screen 1 (Top Left): Titled "ルール作成" (Rule Creation). It contains a "ホームに戻る" (Return Home) link. Under "条件部" (Condition Section), there are four items: "設備" (Equipment) set to "揚水発電所" (Pumped Storage Power Station), "作業件名" (Job Name) set to "指定なし" (Not Specified), "作業日数" (Job Days) set to "指定なし" (Not Specified), and "実施時期" (Implementation Period) set to "10月" (October). Under "結論部" (Conclusion Section), "動詞" (Verb) is "行う" (Perform), "分類 1" (Category 1) is "期間指定ルール" (Period Specified Rule), and "分類 2" (Category 2) is "○月△日以降に行う" (Perform on or after ○ month △ day). A "next step >>" button is at the bottom.

Screen 2 (Top Right): Also titled "ルール作成". It shows the same "条件部" (Condition Section) as Screen 1. The "結論部" (Conclusion Section) now has "動詞" (Verb) set to "行う" (Perform) and "分類 2" (Category 2) set to "10月10日以降に行う" (Perform on or after 10 month 10 day). A "next step >>" button is at the bottom.

Screen 3 (Bottom): Also titled "ルール作成". It shows the "ルール記述" (Rule Description) as "揚水発電所で10月に実施する作業は10日以降に行う" (Perform work at pumped storage power station on or after 10 days in October). It also has "優先度" (Priority) set to "10" and "対応する業務目標" (Corresponding Business Goal) set to "地域共生" (Community Coexistence). A "登録" (Register) button is at the bottom.

図 53 Prolog の実行ルールへの半自動変換機能

図 53 エラー! 参照元が見つかりません。はこのルールを追加する手順を示している。まず、左上の画面が表示され、ユーザは条件部と結論部を候補の中から選択する。条件部は、プルダウンにて、設備として揚水発電所、実施時期として 10 月と選択し対象作業を指定する。結論部は期間指定の 1 つとして「○月△日以降に行う」を選択し、次のステップに進むと、右上の画面が表示される。ここでは、結論部の「以降に行う」に対し、具体的な日付 10 月 10 日を入力する。そして、下の登録画面になる。登録画面には生成された日本語のルール記述が表示されていて、優先度を入力し、対応する業務目標を選択する。なお、生成された日本語のルール記述は、必要に応じて編集できる。最後に登録ボタンを押すと、実行ルールが生成される。図 54 エラー! 参照元が見つかりません。は、新しく追加された実行ルールが反映されたルール編集画面である。

<input checked="" type="checkbox"/>	74	終了日＝金曜日を避ける	130
<input checked="" type="checkbox"/>	75	開始日＝金曜日を避ける	130
<input checked="" type="checkbox"/>	76	開始日＝月曜日を避ける	120
<input checked="" type="checkbox"/>	77	開始日＝土日を避ける	80
<input checked="" type="checkbox"/>	78	終了日＝土日を避ける	110
<input checked="" type="checkbox"/>	79	揚水発電所で10月に実施する作業は10日以降に行う	10

保存名:

図 54 追加ルールが反映されたルール編集画面

このルール編集画面に登録されることにより、「終了日＝金曜日を避ける」などの既存のルールと同様に、適用を除外することや優先度を変更することができるようになる。

図 55 エラー! 参照元が見つかりません。は, 追加したルールの優先度を最も高くして, 2010 年度の停止スケジュールに対し再スケジューリングをした 10 月の結果である。

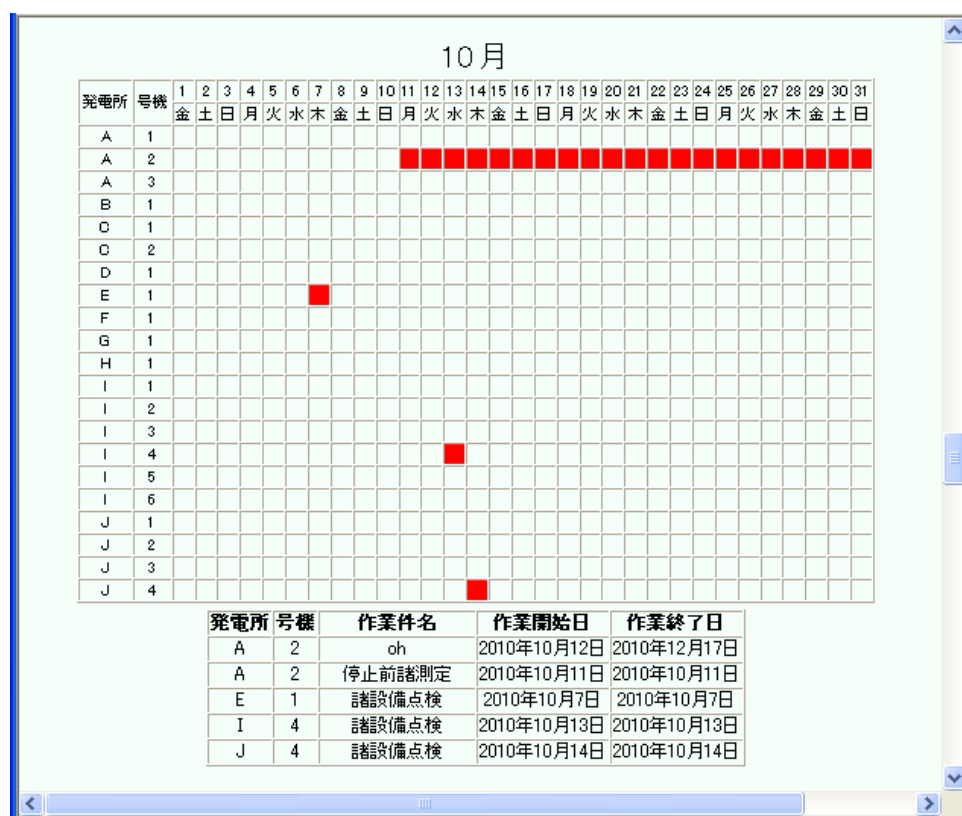


図 55 追加ルールが反映されたスケジューリング結果

ルールが追加された結果, 図 52 エラー! 参照元が見つかりません。で 10 月 5 日にスケジューリングされていた I 発電所 4 号機の諸設備点検は 10 月 13 日に, 10 月 6 日にスケジューリングされていた J 発電所 4 号機の諸設備点検は 10 月 14 日に変更されている。

このように、ドメイン・オントロジーを参照して、日本語ルールから半自動的に Prolog の実行ルールを作成できるようになったことで、熟練者自身によるルールベースの更新が可能になった。新たなルールが必要になった時も、熟練者はまず日本語でルール・オントロジーを構築し、そこから Prolog の実行ルールを生成して、ルールの洗練を行うというプロセスが可能であり、スケジューリング・システムのメンテナンス性は大きく向上した。

4.3.3 ルール・オントロジーを活用した事業所の長の意思の自動反映

ルール・オントロジーにおける「浅いルール」と「深いルール」を関係づける正当性関係は、新人が業務目標を踏まえた上でルール全体を理解することを支援すると共に、経営環境などの変化に伴い、事業所の長の意思として重視する業務目標に変化が生じた場合に、熟練者が変更すべき「浅いルール」を把握することを支援することを意図していることは既に述べた。ただし、これはあくまで、熟練者自らが「深いルール」と「浅いルール」の関係を辿ることを前提としていた。本節では、3.7.2.3 で述べた「**ルール・オントロジーを活用した事業所の長の意思の反映機能**」の活用の一例として、事業所の長の意思として、「**いつ水電力量を低減する**」という業務目標をより重視することが示された場合に、それに沿って停止スケジュールを再作成することを考える。

2011 年度の実際の停止スケジュールをサンプルケースとして、スケジューリング・システムにより一旦停止スケジュールが作成された後に、事業所の長の意思として「**いつ水電力量を低減する**」という業務目標をより重視することが示されたと仮定して、停止スケジュールの再作成する流れを、画面を示しながら説明する。

図 56 がスケジューリング・システムにより作成された初期の 2011 年度停止スケジュールである。図 56 の日程表上**エラー! 参照元が見つかりません。**、発電所毎に作業がスケジュール化された日にちが赤で示されている。その作業件名などの詳細な情報は、下部の表に表示される。



図 56 スケジューリング・システムによる 2011 年度停止スケジュール

ここで、「いっ水電力量を低減する」という業務目標をより重視するという事業所の長の意思が示され、停止スケジュールを再作成する必要が生じたとする。その場合、上部の「業務目標を変えて実行」の下「いっ水をより重視する」の右にある「実行」ボタンを押下することにより、最も深いルール「いっ水電力量を低減する」により正当性を説明されない実行ルールの中で、優先度の最も低い実行ルールの適用が除外され、再びスケジューリングが実行され、図 57 に示すように、「いっ水電力量を低減する」をより重視した停止スケジュールが従来の停止スケジュールと対比されて出力される。図 57 では、いっ水電力量は 6,046kwh 削減されている（削減率 0.05%）。

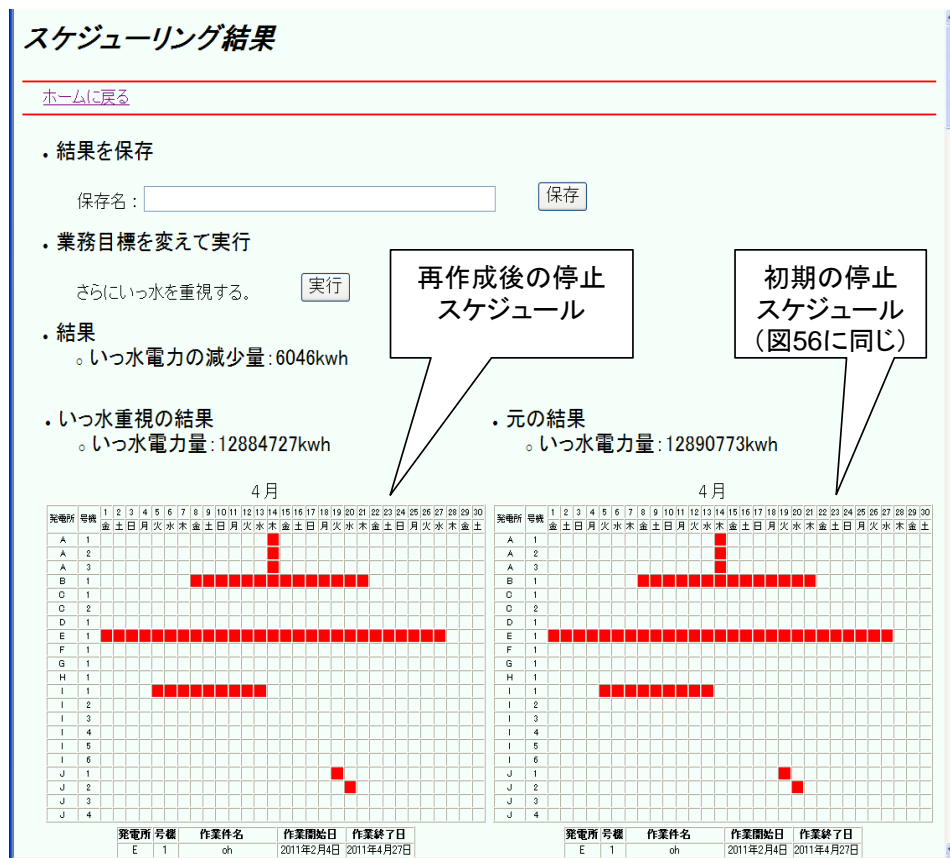


図 57 原価低減を重視した停止スケジュール

さらにいっ水を重視したい場合は、「さらにいっ水を重視する」の右の「実行」ボタンを押下することにより、「いっ水電力量を低減する」により正当性を説明されない実行ルールの中で、次に優先度の低い実行ルールも適用が除外され、再びスケジューリングが実行される。その結果を図 58 に示す。青色および黄色で示された部分が日程が変更された停止作業を表す。相対的に規模が大きくいっ水電力量に大きく影響する G 発電所の停止作業が、いっ水電力が少なくなるよう、水量の少ない月の初めに移動された結果、いっ水電力量は 208,625kwh 削減された（削減率 1.6%）。

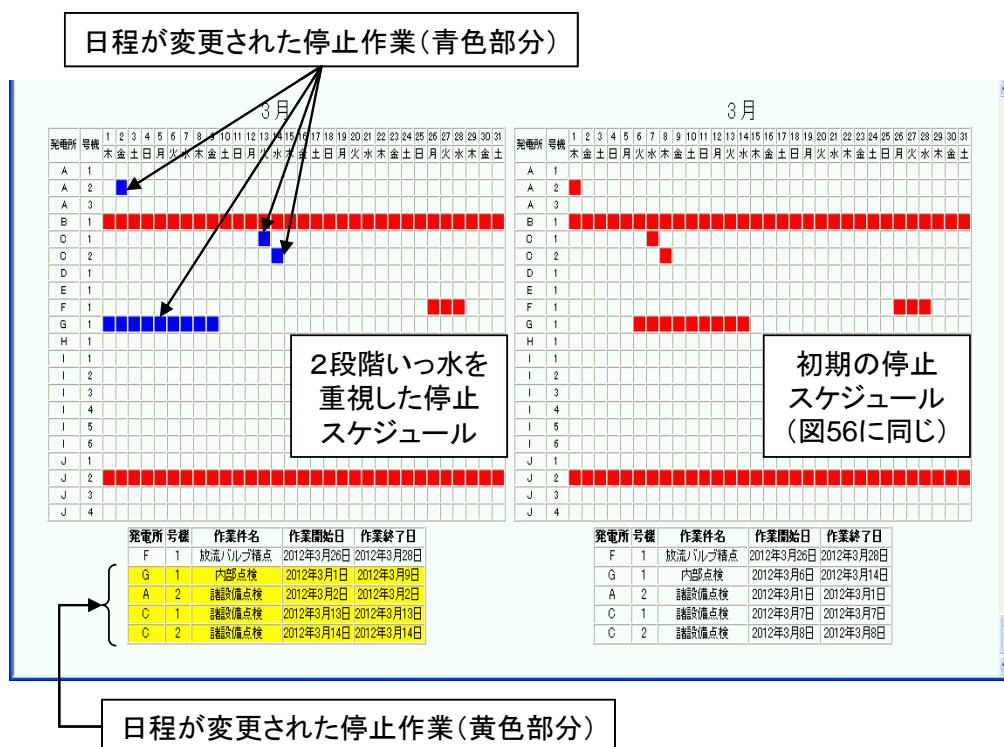


図 58 スケジューリング結果の変化

なお、2009 年度、2010 年度、2011 年度の実際の停止スケジュールをサンプルケースとして、「いっ水電力量を低減する」をより重視するレベルに応じた、ルール適用除外率、いっ水の削減率の推移を表 7 にまとめた。例えば、優先度 12 の行は、優先度 13, 14, 15 の「いっ水電力を少なくする」により正当性を説明されないルール計 7 個が適用除外され、ルールの適用除外率は $7/72=9.7\%$ であることを示していて、その場合のいっ水電力の削減率は、それぞれ、1.62%, 0.29%, 0.02% になる。

いっ水電力の削減率は、年度により大きな差があるが、これは、各年度の初期の停止スケジュールにおいて既に発生している適用除外ルールの数に影響される。例えば、2010 年度では、もともとスケジュールがタイトであり、初期の停止スケジュールにおいても既にルールの適用除外が多く発生していたため、改めて、「いっ水電力を少なくする」により正当性を説明されない優先度の低いルールを除外しても、それらのルールの多くは、実際には初期のスケジュール作成段階においても既に適用が除外されていたため、実行可能解空間が広がらず、いっ水電力はほとんど削減されない。一方、2011 年度では、初期のスケジュール作成において適用除外されているルールが少ない。そのため、「いっ水電力量を低減する」により正当性を説明されない優先度の低いルールを除外すると、その中から、いっ水電力がすくなくなるよう探索をし直すため、相対的に、いっ水電力の削減率が大きくなる。また、すべての年度において、優先度 11 (即ち、優先度 12 以降を適用除外)、優先度 12 (優先

度13以降を適用除外)において、いつ水電力の削減率の変動が相対的に大きい、これは優先度12, 13ルールが状況によって実行可能解空間をいつ水電力を大きくする方向に制約するものであるためである。各年度の初期の停止スケジュールの作成において、優先度12, 13のルールがどの程度適用除外されているかの差が、その段階での年度のいつ水電力の削減率の大きな差を生んでいると同時に、全体的ないつ水電力削減率にも大きな影響を及ぼしている。

表7 いつ水電力重視によるルールの適用除外率といつ水電力削減率

優先度	計	if then 形式の「最も浅いルール」			いつ水電力削減率(%)		
		「いつ水電力を少なくする」により正当性を説明		適用除外率(%)	2009年度	2010年度	2011年度
		されない	される				
1	26	22	4	56.9%	—	—	—
2	5	5		50.0%	—	—	—
3	2	2		47.2%	-0.68%	-0.13%	-1.83%
4	4	4		41.7%	-0.66%	-0.12%	-1.77%
5	4	3	1	37.5%	-0.66%	-0.12%	-1.77%
6	2	2		34.7%	-0.66%	-0.12%	-1.77%
7	10	10		20.8%	-0.66%	-0.12%	-1.77%
8	5	4	1	15.3%	-0.66%	-0.12%	-1.77%
9	1	1		13.9%	-0.66%	-0.11%	-1.66%
10	1	1		12.5%	-0.66%	-0.11%	-1.66%
11	1	1		11.1%	-0.66%	-0.11%	-1.66%
12	1	1		9.7%	-0.29%	-0.02%	-1.62%
13	2	2		6.9%	0.00%	-0.01%	-0.05%
14	4	4		1.4%	0.00%	0.00%	0.00%
15	4	1	3	0.0%	0.00%	0.00%	0.00%
計	72	63	9				

このように、もともとの停止スケジュールによって、いつ水電力の更なる削減の余地は異なる。それは、さらにいつ水電力量を削減するという経営の意思とは独立に、現場レベルにおいても、実行可能な停止スケジュールを作成しなければならないという必要性によって同様のことが行われているからである。すなわち、経営レベルの意思と現場レベルの必要性そのものは異なっているが、それが実際に最も浅いルールの変更に反映された場合には同様の効果を生むことがある。そして、経営レベルの意思が示された場合に、それが最も浅いルールに及ぼす変化が、現場レベルの必要性により先取りされていた場合は、経営レベルの意思の実現は困難を伴う場合がある。

これはあくまで一例であるが、ルール・オントロロジーとルールベース・システムを連携させることで、これまで結びつきが必ずしも十分ではなかった経営の意思と現場の考えを結ぶことの支援ができることが示された。

4.3.4 熟練者による持続的な表出化・連結化に対する評価と考察

まず、熟練者がプルダウン・メニューからの日本語ルール生成と実行ルールへの変換機能により、将来的に必要となるであろうルールを生成できるかどうかを評価した。具体的には、熟練者2名に、その操作方法を説明した後、状況が変化した場合を想定して、自由に新たなルールを生成してもらった。生成してもらったルールは、おおよそ10個である。その結果、熟練者が生成しようとするルールに「ルール生成・変換モジュール」が対応していない場合はなく、かつ、操作上も、熟練者はプルダウン・メニューからの選択により問題なくルールを生成できることが確認できた。生成された日本語ルール記述に関しても、通常の日本語と全く同等とまでは言えないが、誤解を生むような記述となることはなかった。生成された日本語ルール記述とそれを熟練者が修正した例を表8に示す。

表8 生成された日本語ルールと熟練者が修正した日本語ルール

生成された日本語ルール記述	熟練者が修正した日本語ルール記述
10月に実施するA発電所のすべての作業は、開始日は10月10日以降とする。	10月に実施するA発電所のすべての作業は、10日以降に開始する。
上期に実施するA発電所の内部点検は、開始日、終了日を相互に重ねない。	上期に実施するA発電所の内部点検の開始日と終了日を相互に重ねない。
同一月に実施するB発電所1号機のC作業と同一月に実施するD発電所2号機のE作業は、前者の開始日と後者の全期間は重ねない。	B発電所1号機のC作業の開始日は、同一月に実施するD発電所2号機のE作業とは重ねない。

なお、4.3.2.1 に述べたように条件部にて設備、作業に加えて、実施時期、作業日数を指定できるようにしている。一方、生成されるルールは基本的には設備に対する作業に関するルールであるため、条件部で設備、作業に加えて実施時期、作業日数を指定することに対し、熟練者が違和感を抱く可能性があった。特に、実施時期に関しては、結論部が「期間指定」「除外期間指定」(4.3.2.1 表6を参照のこと)の場合は、実施時期が条件部とともに結論部にも現れる。この条件部に対する熟練者による評価は、この実施時期、作業日数を含めた条件部の指定は熟練者のこれまでの思考と合っているとのことであった。実際、既存の72個のif then形式で管理されるルールの中で、条件部で実施時期、作業日数を指定しているルールは表9にあるように、19個ある。また、実施時期指定の10個の内、7個は、結論部にて更に絞り込んだ実施時期を指定するものであった。

表9 条件部に実施時期、作業日数を指定するルール数

条件部	ルール数	内、結論部にて実施時期を制約
実施時期指定有り	10	7
作業日数指定有り	9	0
全数	72	-

注：条件部にて実施時期、作業日数の両方を指定するルールはない。

全体としてプルダウン・メニューからの日本語ルールの生成と実行ルールへの変換機能に対する評価は高く、熟練者からは、このようにプルダウン・メニューから日本語ルールの生成と実行ルールへの変換ができれば、自分たちでスケジューリング・システムを維持していけるという評価を得た。

次にスケジューリング・システム全体に対する熟練者の評価を具体的に挙げる。

- (1) A総合制御所が管理する全21台の発電機の大筋の年間作業計画を短時間に作成できる。
- (2) 「他発電機の作業計画との重複を避ける.」、「休日には作業を入れない.」などのルールに対して、人間が作成する場合には発生しがちなミスが生じない。
- (3) 祝日が土曜日となった場合の扱いなど、単純ではあるが、熟練者といえども、ミスや適用漏れを起こしやすいルールがあるが、それらのルールを確実に適用してくれる。
- (4) 作成後の検討結果から、修正が容易にできる。一部を変更するために、全体構成に影響が出る場合、人間がやり直すと多大な作業量とミスの危険性が生じる。
- (5) 計画作成と同時に、いっ水電力量を正確に算出できるため、違う作業計画ケースを作成し、生産性の検討比較が可能。人間が計算する場合は、かなりの時間が掛かりミスもある。
- (6) 計算させるためには正確に入力することが必要であり、事前の準備の段階で正確な情報を集めることにつながる。

熟練者のこのようなスケジューリング・システムに対する評価は、熟練者に対しスケジューリング・システムを維持していくことへの動機を与え、上記のプルダウン・メニューからの日本語ルールの生成と実行ルールへの変換機能に対する評価と合わせて、スケジューリング・システムは維持されていくものと考えられる。

ただし、将来的には、今回用意したパターンに収まらないルールが発生する可能性は否定できない。例えば、相互依存作業ルールの条件部に関しては、2項関係に収まらない多項関係が必要になるかもしれないし、また、結論部に関しても、現在の「その他」以外にも個別に対応しなければならないものが発生する可能性もある。このような可能性に対する対応としては、条件部に関しては極力現在の枠組みを維持し、それを超える場合は結論部側での個別に対応することを想定している。既存の枠組みでは記述できない条件 A を持つルール(if A then B) が必要になった場合は、A を既存の枠組みで記述できる条件 A' と記述できない条件 E により $A \equiv A' \cap E$ と分割し、(if (A' \cap E) then B) と (if A' then (if E then B))が同値であることを利用して、条件部は A' として既存の枠組みに抑え、結論部に (if E then B) に個別対応することを基本方針とする。例えば、「A 発電所の B 作業は計画年度が閏年の場合は、2 月に実施する」というルールが必要になったとする。この場合、条件部の既存の枠組みでは「計画年度が閏年である」という条件は記述できない。従って、「計画年度が閏年である」という条件部は結論部に折り込み、「計画年度が閏年であれば、2 月

に実施する」という結論部を個別対応により作成する。これにより、既存の枠組みでは対応できないルールが発生した場合でも、個別対応は結論部に限定される。結論部の個別対応を熟練者に委ねるのは無理があるにしても、その発生頻度を考えると、システム部門による対応が十分に可能であると考えられる。

また、いつ水電力低減をより重視した停止スケジュール再作成機能に関しても、熟練者から肯定的な評価を得た。その理由としては、これまでも、停止スケジュールを作成した結果、いつ水電力が想定よりも多くなってしまった場合等において、いつ水電力を少なくするために停止スケジュールの再作成がよく行われていたが、再作成しても必ずしも期待したようないつ水電力の削減が得られないことも多く、それが業務量増大につながっていたことがあげられる。

3つのサンプルケースによる検証により、再作成によるいつ水電力の削減率に大きな開きがあることが明らかになったが、この再作成機能により迅速に停止スケジュールを再作成し、かつ、いつ水電力の削減率も明らかにできる。このことは、熟練者のいつ水電力低減をより重視した停止スケジュール再作成に対する業務量軽減に貢献するだけでなく、事業所の長のレベルと実務者のレベルを結び付けるという点でも重要な意味を持つ。すなわち、この機能により、事業所の長は「いつ水電力をより少なくする」という意思を示した後、実務レベルにおいてそれにどの程度応えられるかを直ちに知ることができ、必要であれば別の方策をとることの意思決定が迅速に行えるようになる。例えば、個々の停止作業は既定とした上での「停止スケジュール」の再作成では所定の削減が出来ないことが明らかになれば、事業所の長は、個々の停止作業の工程に踏み込んで、個々の停止作業の作業日数の短縮に取り組むという次の方策を示すこともできる。

4.3.5 いつ水電力量の低減をより重視したスケジューリングの改善

ただし、いつ水電力量の低減をより重視したスケジューリングに関して、本来影響を受けるべきでない作業に関しても、日程が変化してしまっている部分があった。具体的には、ダムの貯水能力により通常はいつ水電力が発生しない揚水発電所の作業に関しても、いつ水電力量の低減をより重視した場合に、日程が変更されてしまっていた。現在のルール・オントロジーは、その情報を持っていないため、こういった間違いが起こってしまう。そこで、ドメイン・オントロジーに、いつ水の低減に影響ある発電所は一般発電所であることを記述し、いつ水の低減を重視する場合に、ドメイン・オントロジーを参照して、一般発電所の作業のみ、「いつ水電力を少なくする」により正当性を説明されない実行ルールの中から優先度の最も低いものが除外して再スケジューリングを行い、それ以外ならば、元の実行ルールのままでスケジューリングするようにアルゴリズムを変更した。次に、2010年度の問題に対して、この新しいアルゴリズムで再スケジューリングした結果と以前のアルゴリズムの結果を示す。

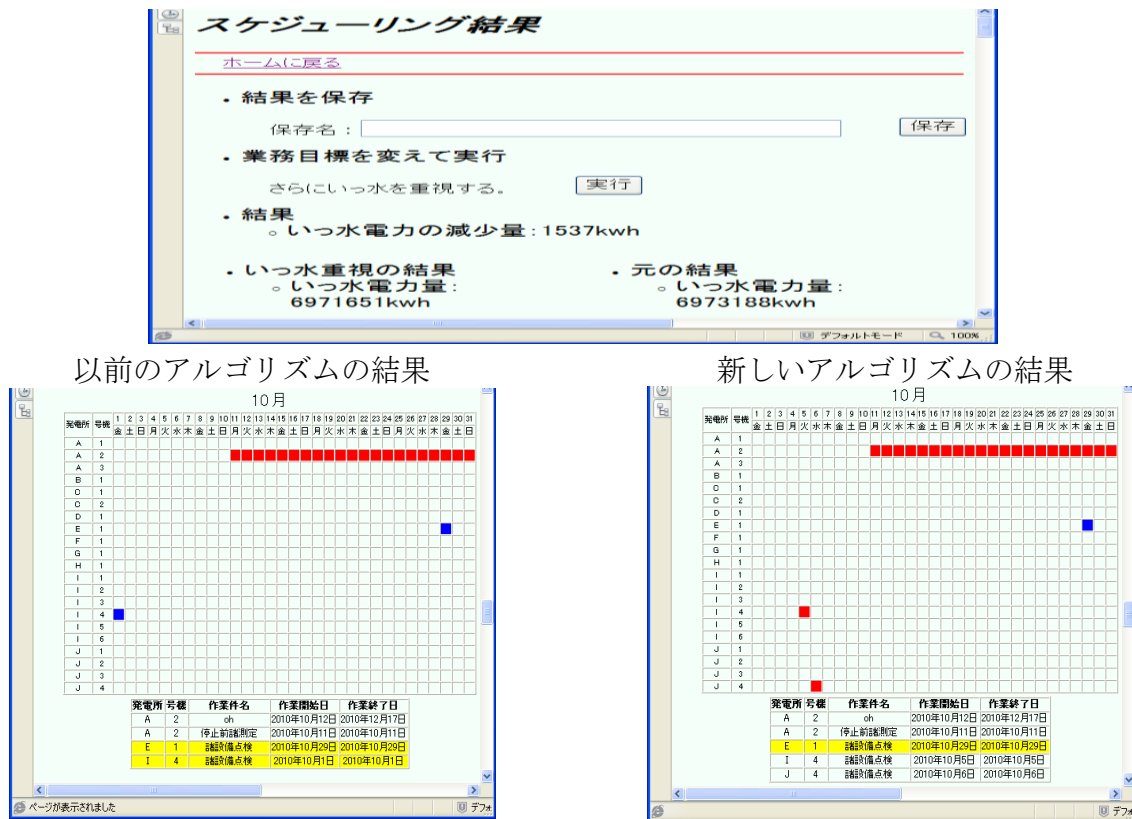


図 59 いっ水電力量低減を重視するアルゴリズムの変更結果

図 59 エラー! 参照元が見つかりません。において、一般発電所である E 発電所の諸設備点検に関しては、アルゴリズムを変更しても、再スケジューリング結果は変わらず、削減されるいっ水電力量は変化していない。一方、揚水発電所である I 発電所と J 発電所の結果を見ると、以前のアルゴリズムでは再スケジューリングにより日程が変更されていたが、新しいアルゴリズムでは再スケジューリングによっても日程は変更されていない。これによって、新しいアルゴリズムでは、いっ水電力低減を重視することによって影響を受ける作業を特定することで、いっ水電力の減少量はそのまま、スケジューリング結果の変更を必要最小限にとどめ、より適切な高いスケジューリング結果を提示することができるようになった。

第5章 結論

本章では、モデルケースに対する評価および改善を踏まえ、本研究の結論を述べる。また、本研究の今後の課題および展望についても述べる。

5.1 結論

本提案は、あるまとまった責務を果たしている事業所を対象に、長年に渡って OJT により暗黙知のまま継承されてきた技術・技能が組織のスリム化により OJT による継承が困難になりつつあることを踏まえ、新たな持続的な継承の枠組みを提案するものである。対象とする技術・技能は、長年に渡って継承されてきたものであり、短期的に見れば比較的安定しているが、経営環境などの変化に伴い、事業所の長の意思に従い進化していくものである。また、匠的熟練がコンピュータによる数値制御に置き換えられつつある中、本提案が対象としている技術・技能は知的熟練である。

本提案では、当該事業所の業務全体を、その事業所の長が必要とするレベルで業務プロセス・フローとして記述したうえで、それを実務者向けに標準的な業務として無理なく記述できるレベルまでブレイクダウンする。各業務プロセスにおいて必要となる比較的小さな粒度の業務ルールは正当性関係などによりルール・オントロジーとして体系化される。ここで、正当性関係における「最も深いルール」は事業所の長が認識する業務プロセスと対応し、その業務目標を表すものとなる。また、「最も浅いルール」は、ブレイクダウンされた業務プロセスに対応し、実務者が具体的に業務を遂行する上で用いられるとともに、ルールベース上の実行ルールに変換され、ルールベース・システムを構成する。さらに、業務ルールで使用される概念はドメイン・オントロジーとして体系化される。新人は、業務プロセス・フローから、ルール・オントロジー、ドメイン・オントロジーを参照して知的熟練を獲得することができる。また、熟練者は、経営環境などの変化に対しても、業務目標を表す最も深いルールとの関係によりルール・オントロジー全体に対する更新の示唆を受け、ドメイン・オントロジーを活用して、各業務ルールに対し必要な更新を持続的に行うことができる。ルールベース・システムは知的熟練を新人が内面化する上でも、熟練者が漏れなく表出化する上でも活用され、また、熟練者に対し知的熟練の表出化に対する動機を与える。

本提案は、モデルケースとして、東京電力の水力発電関係の現場業務である「停止調整業務」に試験適用され、その有効性が確認された。「停止調整業務」は、設備に関する技術的知識、地域との関係、自然条件など、幅広い知識を必要とし、それらの知識を様々な状況に応じて適切に組み合わせて使用する必要があるという意味で知的熟練を必要とする業務であり、また、その知識が表出化されておらず OJT により継承されていたという点において、本提案が対象とするひとつの典型的な業務である。この典型的な業務において本提案

の有効性が確認されたことは、同様の知的熟練を要する業務に対しても、新人による内面化、および、熟練者による経営環境等の変化に対する事業所の長の意思に従った持続的な表出化・連結化に関し一定の見通しを得られたものと考えられる。

本提案では、熟練者の暗黙知である知的熟練は、表出化し、整理することにより、多くの場合、自然言語で記述できる比較的小さな粒度の単純な業務ルールに分割されることを前提としている。今回のモデルケースでは、スケジュール作成という業務の性格もあり、実際に小さな粒度の単純な業務ルールに分割された。

また、スケジュール作成という業務の性格により、最も浅いルールは、ルールベース・システムとの親和性が高く、比較的容易に、スケジューリング・システムとして実装され、新人による内面化支援に大きな役割を果たした。また、最も浅いルールが類型化されたことにより、ドメイン・オントロジーからの語彙の選択による日本語ルールの生成、および、その実行ルールの変換が実現され、熟練者によりスケジューリング・システムが持続的にメンテナンスされうるようになった。

経営環境等の変化に対する事業所の長の意思の反映の支援という観点から見ると、「最も深いルール」そのものは普遍性が高く、経営環境等の変化に対する事業所の長の意思は、既存の「最も深いルール」の重要度の変更によって示されることが想定できたことが重要である。

以上のような「停止調整業務」の性質を踏まえると、本提案は、以下のような性格を持つ知的熟練を要する業務一般に対し、有効であると考えられる。

- (1) 業務が成熟・安定していて、経営環境等の変化に対する事業所の長の意思は、既存の「最も深いルール」の重要度の変更によって示されうると想定されること。
- (2) 知的熟練が自然言語で記述できる比較的小さな粒度の単純な業務ルールに分割されること。
- (3) 特に、業務を遂行する上で直接必要となる「最も浅いルール」に関しては類型化されること。

このような性質を満たす典型的な業務としては、今回適用した水力発電設備に関する停止調整業務に限らず、設備の点検等の作業の年間計画レベルでのスケジューリング業務が挙げられる。東京電力は、水力発電設備以外にも、様々な発電設備、流通設備等の設備を有しており、設備に対する点検等の作業のスケジューリング業務は数多く存在するが、それらの業務に対して本提案が有効であると考えられる。

また、このような作業のスケジューリング業務以外においても、所謂制約充足問題で、その制約が長年の経験に基づき形成されている業務は、その制約を構成する「最も浅いルール」が非常に複雑で多岐にわたる場合を除いては、(1) (2) (3) が満足される可能性が高く、本提案が有効である可能性が高い。

ただし、今回のモデルケースでは十分に確認できていない点もある。まず、今回のモデルケースでは、オントロジー等などの初期構築に関しては、プロジェクト・メンバーが熟練者にインタビューを行いつつ、一体となって実施したため、知的熟練の表出化・連結化の業務量・コストに関しては問題となることはなかったが、今後、一般的に適用していくにあたっては、知的熟練の表出化・連結化の業務量・コストをどう考えるかも重要になってくる。また、今回のモデルケースでは、特定の業務での試行であったため、ルール・オントロジー、ドメイン・オントロジーにより、新人が知的熟練を他の業務にも幅広く活用できるよう内面化しえたかどうかは十分に確認できていない。経営環境の急変などにより、知的熟練そのものが一時に抜本的な見直しを求められるような状況においても、熟練者が表出化・連結化された知的熟練を更新することが可能かどうか、必ずしも定かではない。

なお、本提案では、ルールベース・システムを使った試行錯誤が、新人に対しても熟練者に対しても重要な役割を果たしたが、今回試行した業務は、状況に依存して様々なルールの緩和が必要になるという特質があったものの、スケジューリングというルールベース・システムの適用しやすい業務であったことが幸いした面も否定できない。ルールベース・システムが適さない業務の場合には、ルールベース・システムに代わるものを考える必要もあるだろう。

以上、いくつかの不確実な点、制約、課題はあるものの、本論文の提案は、これまで表出化されておらず継承を OJT に依存してきた知的熟練に対し、熟練者による持続的な表出化・連結化を前提にした OJT に依存しない継承の枠組みとして有効であると考えられる。

5.2 今後の課題と展望

本提案に対する評価をより確実にしていくためには、様々な業務ドメインにおいて、適用実績を積み重ねるとともに、必要な改善を積み重ねていくことが第一である。その際、一番の重要なことは、熟練者が知的熟練の表出化・連結化を、経営環境などの変化に対応し、事業所の長の意思に従いながら、いかにして持続的に実施していけるかである。その上で、業務の特質による、知的熟練の表出化・連結化コストおよびその安定性が大きなファクターになる。

そのため、「停止調整業務」に関して、まずは、東京電力全社大で適用し、定着させることが、実績を積み重ねる第一歩と考えている。「停止調整業務」は、A 総合制御所での試験適用において、表出化・連結化コストはそれなりにかかるものの、その知的熟練が比較的安定していて、また、スケジューリング・システムによる直接的なメリットの享受も受けやすいことが明らかになったからである。

また、**2.2.7 ナレッジマネジメント推進上の課題** において述べたように、ナレッジマネ

ジメントが成功するための大きな要因にトップのリーダーシップがある。本提案に即して言えば、事業所の長の理解とリーダーシップが必須となる。各事業所における日常の取り組みが中心となる本提案は、組織のスリム化に苦しむ各事業所にとっては新たな業務負荷の発生であり、日業業務に付加する形で実施されるが故の負担感がある。ただし、東京電力に限らず、多くの企業で、知的熟練を含む技術・技能の維持・向上・継承のために、技術技能研修、技術技能競技大会などの様々な取り組みが行われている。各事業所における本提案の取り組みに対するコストは、全社的に行われる技術技能研修、技術技能競技大会などの様々な取り組みに対するコストと公正に比較されるべきであろう。その上で、各事業所における本提案の実施にあたっては、その取り組みに対して、その長の理解とリーダーシップが重要となる。

補遺 I

4.2.2.2 実験結果 に関し，グループ 1 とグループ 3 に有意な差があるかどうかの統計的検定を行う．なお，被験者数が非常に少ないため，正規分布を仮定する本検定はあくまで参考的なものである．

1. 平均学習時間の差の検定

学習時間に関し，グループ 1, 3 間の母平均に有意な差があるかどうかを検定するために，まず，グループ 1 とグループ 3 の学習時間の分散に有意な差があるかどうかを F-検定（両側検定）により検定する．

帰無仮説：グループ 1 とグループ 3 の学習時間の母分散が等しい．

対立仮説：グループ 1 とグループ 3 の学習時間の母分散が等しくない．

表 10 グループ 1, 3 の学習時間の等母分散の検定

	グループ1	グループ3
標本平均	199.0	47.0
標本分散(不偏分散)	2	128
標本数(被験者数)	2	2
f値	0.01563	
自由度	1	1
$P(F \leq f \text{ or } F \geq 1/f)$	0.1583	

表 10 に示されるとおり分散比は 0.01563 であり，グループ 1, 3 の学習時間の分散が等しいという帰無仮説のもとで，これは自由度 1,1 の F 分布に従い，自由度 1,1 の F 分布の値が 0.0625 以下または $1/0.0625$ 以上となる確率は 0.1583 である．従って，有意水準 0.05 の下でこの帰無仮説は棄却されない．

よって，グループ 1, 3 の学習時間の分散が等しいという前提の下で，グループ 3 の学習時間の母平均がグループ 1 よりも有意に短いかどうかを t-検定（両側検定）により検定する．

帰無仮説：グループ 1 とグループ 3 の学習時間の母平均が等しい．

対立仮説：グループ 1 とグループ 3 の学習時間の母平均が等しくない．

表 11 グループ 1, 3 の学習時間の母平均値の差の検定（等母分散を仮定）

	グループ1	グループ3
標本平均	199.0	47.0
標本分散(不偏分散)	2	128
標本数(被験者数)	2	2
t値	18.85	
自由度(小数点以下切り捨て)	1	
$P(T \leq - t \text{ or } T \geq t)$	0.0337	

表 11 に示されるとおり t 値は 18.85 であり, グループ 1, 3 の学習時間の母平均が等しいという帰無仮説のもとで, これは自由度 2 の t 分布に従い, 自由度 2 の t 分布の値が -18.85 以下または 18.85 以上となる確率は 0.0028 である. 従って, 有意水準 0.05 の下でこの帰無仮説は棄却される. 従って, グループ 3 の学習時間の母平均は, グループ 1 の学習時間の母平均より少ない.

なお, グループ 1, 3 の学習時間の母分散が等しいという帰無仮説は棄却されなかったが, これは必ずしもグループ 1, 3 の学習時間の母分散が等しいことを意味する訳ではなく, また, グループ 1, 3 の被験者数が共に 2 と非常に少ないため, グループ 1, 3 の学習時間の母分散の差が大きい可能性もある. そこで, グループ 1, 3 の学習時間の母分散が等しいことを仮定せずに, ウェルチの近似を用いて, より保守的な検定も行うことにする.

表 12 グループ 1, 3 の学習時間の母平均値の差の検定（等母分散を仮定せず）

	グループ1	グループ3
標本平均	199.0	47.0
標本分散(不偏分散)	2	128
標本数(被験者数)	2	2
プールされた分散	65	
t値	18.85	
自由度	2	
$P(T \leq - t \text{ or } T \geq t)$	0.0028	

表 12 に示されるとおり t 値は 18.85 であり, グループ 1, 3 の学習時間の母平均が等しいという帰無仮説のもとで, これは近似的に自由度 1 の t 分布に従い, 自由度 1 の t 分布の値が -18.85 以下または 18.85 以上となる確率は 0.0337 である. 従って, より保守的なウェルチの近似を用いた検定においても, 有意水準 0.05 の下でこの帰無仮説は棄却される.

なお, ウェルチの近似の場合の近似する t 分布の自由度 v は, グループ i ($i=1, 3$) の標本数 (被験者数) が n で等しいことを利用すると,
 $v = (S_1 + S_2)^2 / (S_1^2 / (n_1 - 1) + S_2^2 / (n_2 - 1))$ で, $S_1=2, S_2=32, n=2$ を代入すると, $v=1.125$ であり, 保守的な検定をするために, 小数点以下を切り捨てて, 自由度 $v=1$ とした.

また, グループ 1, グループ 3 の標本数が等しいため, ウェルチの近似の場合も, t 値は, 次の通りとなり, 通常の t-検定の場合の t 値に等しくなる.

$$t = (M_1 - M_3) / ((S_1 + S_2) / n)^{1/2}$$

ただし、 M_i 、 S_i は、それぞれグループ i ($i=1, 3$) の標本平均、標本分散（不偏分散）を表し、 n はグループ i ($i=1, 3$) の標本数（被験者数）を表す（すなわち、 $n = 2$ ）。

2. 平均回答時間、平均誤答数の差の検定

同様に、回答時間、誤答数に関し、グループ 1, 3 間の母平均の差の検定を行った。なお、誤答数に関しては、その値域が 0～130（設問数）に限定されるが、値域に制限のない正規分布を仮定して検定することは保守的な検定となるため、そのまま、通常の前平均値の差の検定を行っている。この誤答数の差の検定は、正答率ないし誤答率を値域が[0, 1]であることを考慮せずに通常の前平均値の差の検定により検定したものに等しくなる。

その結果を平均学習時間の差の検定を含め、表 13 に示す通りであり、グループ 1, 3 間において、いずれも有意水準 0.05 とすると、平均学習時間に関しては有意な差が認められたが、回答時間、誤答数（あるいは、正答率）に関しては、有意な差は認められなかった。

表 13 グループ 1, 3 の母平均の差の検定結果

検定対象		学習時間		回答時間		誤答数	
グループ		1	3	1	3	1	3
等母分散 の検定	標本平均	199.0	47.0	292.5	285.0	11.5	15.5
	標本分散(不偏分散)	2	128	2812.5	2450.0	4.5	0.5
	標本数(被験者数)	2	2	2	2	2	2
	f値	0.01563		1.148		9.000	
	自由度	1	1	1	1	1	1
等母平均 の検定	$P(F \leq f \text{ or } F \geq 1/f)$	0.1583		0.9561		0.4097	
	帰無仮説(有意水準0.05)	棄却せず		棄却せず		棄却せず	
	t値	18.85		0.15		-2.53	
	等母分散を 仮定	自由度		2		2	
	等母分散を 仮定せず (ウェルチ近似)	自由度(小数点以下切り捨て)		1		1	
		$P(T \leq - t \text{ or } T \geq t)$		0.0028		0.8972	
		帰無仮説(有意水準0.05)		棄却		棄却せず	
		$P(T \leq - t \text{ or } T \geq t)$		0.0337		0.9076	
		帰無仮説(有意水準0.05)		棄却		棄却せず	

補遺 II

4.3.2.1 最も浅いルールの分析 における制約部と結論部の解釈に基づくルールの分類の正確な定義を記す．記述は，制約部に規定される作業集合をソートとして扱い，通常の述語論理の記法によった．なお，本定義はプロジェクト・メンバー内で正確な定義を共有するために利用したものであり，エンドユーザが必要とするものではない．

1. 単独作業ルール

作業を表す変数を x ，制約部に規定される作業集合を S ，結論部に規定される制約を表す単項述語を C として，次で表されるルール．

$$\forall x/S \ C(x)$$

2. 相互依存作業ルール

2.1 対称ルール

以下の i) ii) のいずれかで表されるルール．

i) 制約部に規定される作業集合が単一の場合

作業を表す変数を x, y ，制約部に規定される作業集合を S ，結論部に規定される制約を表す 2 項述語を C として，

$$\forall x/S \ \forall y/S \ (if \ not \ (x=y) \ then \ C(x, y))$$

ii) 制約部に規定される作業集合が複数の場合

作業を表す変数を x, y ，制約部に規定される作業集合を S_i ，結論部に規定される制約を表す 2 項述語を C として，

$$\forall i/I \ (\forall x/S_i \ \forall y/S_i \ (if \ not \ (x=y) \ then \ C(x, y)))$$

ただし，作業集合 S_i 合の添え字 i は月等を表すもので，そのソートを I (i が月の場合であれば，月を表す添え字の集合, i.e. 1～12) とする．また，作業集合の添え字が i, j 等複数になることも認める．

2.2 非対称ルール

以下の i) ii) のいずれかで表されるルール．

i) 制約部に規定される作業集合の組が一つの場合

作業を表す変数を x, y ，制約部に規定される作業集合を T, U ，結論部に規定される制約を表す 2 項述語を C として，

$$\forall x/T \ \forall y/U \ (if \ not \ (x=y) \ then \ C(x, y))$$

ただし、 T と U は同一の集合であってはならない。

ii) 制約部で規定される作業集合の組が複数の場合

作業を表す変数を x, y , 制約部で規定される作業集合を T_i, U_i , 結論部で規定される制約を表す 2 項述語を C として,

$$\forall i / I (\forall x / T_i \forall y / U_i \text{ (if not } (x=y) \text{ then } C(x, y)))$$

ただし、作業集合 T_i, U_i の添え字 i は月等を表すもので、そのソートを I (i が月の場合であれば、月を表す添え字の集合, i.e. 1~12) とする。また、作業集合の添え字が i, j 等複数になることも認める。

また、 T_i と U_i は同一の集合であってはならない。

補足

対称ルールにおいては、その定義より、制約を表す述語 C は対称 (i.e. $\forall x / S_i \forall y / S_i C(x, y) = C(y, x)$ (真理値として)) となる。

一方、非対称ルールにおいては、制約を表す述語 C に対して非対称性 (i.e. $\exists x / T_i \exists y / U_i \text{ not } (C(x, y) = C(y, x))$ (真理値として)) は要求しない。

従って、対称ルールと非対称ルールは必ずしも排他関係にはなく、

例えば、対称ルール

$$\forall x / S \forall y / S \text{ (if not } (x=y) \text{ then } C(x, y)) \text{ ないし}$$

$$\forall i / I (\forall x / S_i \forall y / S_i \text{ (if not } (x=y) \text{ then } C(x, y)))$$

において、作業集合が $S=\{a, b\}$ ないし $S_i=\{a_i, b_i\}$ と 2 つの要素からなる場合、

$T=\{a\}, U=\{b\}$ ないし $T_i=\{a_i\}, U_i=\{b_i\}$ とすることにより、

非対称ルールとして

$$\forall x / T \forall y / U \text{ (if not } (x=y) \text{ then } C(x, y)) \text{ ないし}$$

$$\forall i / I (\forall x / T_i \forall y / U_i \text{ (if not } (x=y) \text{ then } C(x, y)))$$

と記述もできる。

このことは、実際、対称ルールとして対象作業が各項の or 指定の and 解釈の単一作業集合として指定できない場合 (例えば、A 発電所の B 作業と C 発電所の D 作業 など) に、非対称ルールとして規定するために用いられている。

参考文献

- [1] 森下泰, “デジタル・マイスター・プロジェクト”, 精密工学会誌, Vol. 67 巻, 5 号, pp. 709-710, 2001.
 - [2] 小池和夫, “仕事の経済学 第3版”, 東洋経済新報社, 2005.
 - [3] 小池和夫, “日本企業の人材育成 不確実性に対処するノウハウ”, 中公新書 1373, 1997.
 - [4] 綿貫啓一, “バーチャルリアリティ技術による匠の技の伝承と人材育成”, 精密工学会誌 Vol.72 No.1, pp.46-51, 2006.
 - [5] 中馬宏之, “もの造り現場における問題発見・解決型熟練”, 日本労働研究雑誌, 510 号 特別号, 日本労働研究機構, 2002.
 - [6] I. Nonaka and H. Takeuchi, “The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation”, Oxford University Press, 1995 (梅本勝博訳, “知的創造企業”, 東洋経済新報社, 1996).
 - [7] I. Nonaka and N. Konno, “The Concept of 'ba': Building a foundation for knowledge creation”, California Management Review, Vol. 40, No. 3, pp. 40-54, Spring, 1998.
 - [8] Y. Hijikata, T. Takenaka, Y. Kusumura, and S. Nishida, “Interactive knowledge externalization and combination for SECI model”, Proceedings of the 4th international conference on Knowledge capture, Whistler BC Canada, pp. 151-158, 2007.
 - [9] KnowledgeMeister, <http://www.toshiba-sol.co.jp/pro/km2/>, 2010 年 8 月 12 日.
 - [10] INSUITE, <http://www.insuite.jp/>, 2010 年 8 月 12 日.
 - [11] 大澤幸生: 「知識マネジメント」, オーム社, 2003.
 - [12] M. T. Hansen, N. Nohria, and T. Tirney, “What's Your Strategy for Managing Knowledge?”, Harvard Business Review March-April pp.106-116, 1999.
 - [13] 国藤進, “知的グループウェアによるナレッジマネジメント”, 日科技連出版社, 2001.
 - [14] T. H. Davenport and L. Prusak, “Working Knowledge, How Organizations Manage What They Know”, Harvard Business School Press, 1998. (梅本勝博訳, “ワーキング・ナレッジ 「知」を活かす経営”, 生産性出版, 2000).
 - [15] R. Weber, and R. Kaplan, “Knowledge-based knowledge management”, In Innovations in Knowledge Engineering, International Series on Advanced Intelligence. Vol.4, pp.151-172, 2003.
 - [16] W.J.ブラック, 白井英俊, 益田誠也訳, “知的知識ベースシステム入門”, 啓学出版, 1989.
 - [17] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. Patel-Schneider, “The Description Logic Handbook: Theory, Implementation and Applications”, Cambridge University Press, 2004.
-

-
- [18] 大須賀節雄“データベースと知識ベース”, オーム社, 1989.
- [19] 薦田憲久, 大川剛直, 安信千津子“エキスパートシステムの設計と開発”, 情報系教科書シリーズ第 21 巻, 昭晃堂, 1997.
- [20] P.ハーモン, D.キング, 諏訪基監訳, “エキスパートシステムズ”, サイエンス社, 1986.
- [21] MYCIN Clinical Decision Support System,
<http://neamh.cns.uni.edu/MedInfo/mycin.html>, 2010 年 8 月 12 日.
- [22] 山口高平, 溝口理一郎, 田岡直樹, 小高浩, 野村康雄, 角所収, “深い知識に基づく一般ユーザー向けの説明機能”, 電子情報通信学会論文誌 D, Vol. J70-D No.11, pp. 2083-2088, 1987.
- [23] G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W. V. Velde, and B. Wielinga, “Knowledge Engineering and Management The CommonKADS Methodology”, The MIT Press, 1999.
- [24] 溝口理一郎, “オントロジー工学”, オーム社, 2005.
- [25] P. J. Hayes, “Naïve Physics I: Ontology for Liquids”, Formal Theories of the Commonsense World, in Jerry R. Hobbs, Robert C. Moore, eds, Ablex Publishing Corporation, pp.71-107, 1985.
- [26] T. R. Gruber, “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, Vol.5, Issue 2, pp.199-220, 1993.
- [27] SUMO (Suggested Upper Merged Ontology), <http://www.ontologyportal.com>
- [28] Cyc Upper Ontology, <http://www.cyc.com/cycdoc/vocab/upperont-diagram.html>, 2010 年 8 月 12 日.
- [29] John F. Sowa’s Top-Level Categories, <http://www.jfsowa.com/ontology/toplevel.htm>, 2010 年 8 月 12 日.
- [30] BFO (Basic Formal Ontology), <http://www.ifomis.org/bfo>, 2010 年 8 月 12 日.
- [31] YATO (Yet Another Top-level Ontology), http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm, 2010 年 8 月 12 日.
- [32] 神崎正英, “セマンティック・ウェブのための RDF/OWL 入門”, 森北出版株式会社, 2005.
- [33] Knowledge Interchange Format, <http://logic.stanford.edu/kif/dpans.html>, 2010 年 8 月 12 日.
- [34] J. F. Sowa, “Conceptual Structures: Information Processing in Mind and Machine”, Addison-Wesley 1984.
- [35] ISO/IEC 24707:2007 Information technology - Common Logic (CL): a framework for a family of logic-based languages, [http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007(E).zip), 2010 年 8 月 12 日.
- [36] W. Chen, M. Kifer, D. S. Warren, “HiLog: A foundation for higher-order logic
-

-
- programming”, The Journal of Logic Programming, Vol. 15, Issue 3, PP. 187-230, 1993.
- [37] Cyc, <http://cyc.com/cyc/technology/whatisccyc>, 2010 年 8 月 12 日.
- [38] ISO 18629 Industrial automation systems and integration - Process specification language
- [39] SKOS, <http://www.w3.org/2004/02/skos/>, 2010 年 8 月 12 日.
- [40] Protégé, <http://protege.stanford.edu/>, 2010 年 8 月 12 日.
- [41] BioPortal, <http://bioportal.bioontology.org/>, 2010 年 8 月 12 日.
- [42] T. Lau and Y. Sure, “Introducing Ontology-based Skill Management at a large Insurance Company”, Proceedings of the Modellierung 2002, Tutzing, Germany, 2002.
- [43] A. P. Morgan, J. A. Cafeo, D. I. Gibbons, R. M. Lesperance, G. H. Sengir, and A. M. Simon, “CBR for Dimensional Management in a Manufacturing Plant”, Lecture Notes in Artificial Intelligence 2080. Springer Verlag, New York, pp 597-610, 2001.
- [44] T. W. Malone, K. Crowston, and G. A. Herman, “Organizing Business Knowledge: The MIT Process Handbook”, MIT press, 2003.
- [45] 橋田浩一, “オントロジーと制約に基づくセマンティック・プラットフォーム”, 人工知能学会誌, Vol. 21, No.6, pp. 712-717, 2006.
- [46] P. Roy, F. Pachet, “Reifying Constraint Satisfaction in Smalltalk”, Journal of Object-Oriented Programming, Vol. 10, No. 4, pp. 43-51, 1997.
-

関連発表

- [1] 岡部雅夫, “オントロジーによる技術・技能の組織的蓄積のサポートについて”, 情報システム学会 第2回全国大会・研究発表大会, 2006.
 - [2] 吉岡亜紀子, 大平昌弘, 飯島正, 山口高平, 山崎浩志, 柳澤雅彦, 岡部雅夫, “オントロジーによる知識継承とスケジューリングの支援”, 第21回人工知能学会全国大会, 1B-5, 2007.
 - [3] 岩間貴史, 立花浩, 山崎浩志, 岡部雅夫, 黒川利明, 小林圭堂, 加藤美穂, 吉岡亜紀子, 山口高平, “業務知識の組織的蓄積・活用を支援するオントロジーに対する一考察”, 情報システム学会第3回研究発表大会, D1-1, 2007.12.
 - [4] 小林圭堂, 加藤美穂, 吉岡亜紀子, 山口高平, “知識継承における外在化支援のための業務ルールとドメイン・オントロジーの構築”, 第22回人工知能学会全国大会, 3D3-2, 2008.
 - [5] A. Yoshioka, K. Kobayashi, T. Yamaguchi, M. Okabe, M. Yanagisawa and H. Yamazaki, “Developing Ontologies and a Rule-based System for Organizational Knowledge Transfer”, PAKM2008 Joint Workshop on Knowledge Management for Service Innovation and Knowledge Networks: Discovering network structure and patterns using Social Network Analysis, pp.1-14, 2008.
 - [6] M. Okabe, M. Yanagisawa, H. Yamazaki, K. Kobayashi, A. Yoshioka and T. Yamaguchi, “Organizational Handing Down of Intelligence Skill Using Ontologies and a Rule-Based System”, 7th International Conference on Practical Aspects of Knowledge Management, LNAI5345, pp.207-218, Springer, 2008.
 - [7] M. Okabe, M. Yanagisawa, H. Yamazaki, K. Kobayashi, A. Yoshioka and T. Yamaguchi, “Ontologies that Support Organizational Knowledge Transfer of Intelligence Skill”, The 2nd Interdisciplinary Ontology Conference (InterOntology09), pp.147-159, 2009.
 - [8] 小林圭堂, 竹田百合恵, 吉岡亜紀子, 岡部雅夫, 柳澤雅彦, 山崎浩志, 山口高平, “オントロジーとルールベース・システム構築に基づく知識継承支援”, 第23回人工知能学会全国大会, 3G2-1, 2009.
 - [9] K. Kobayashi, A. Yoshioka, M. Okabe, M. Yanagisawa, H. Yamazaki and T. Yamaguchi, “How much well does Organizational Knowledge Transfer work with Domain and Rule Ontologies?”, 3rd Conference on Knowledge Science, Engineering and Management (KSEM2009), LNAI5914, pp.382-393, 2009.
 - [10] 岡部雅夫, 小林圭堂, 石川達也, 山口高平, “業務知識の組織的継承へのオントロジーおよび知識モデリングの活用”, 情報システム学会 第5回全国大会・研究発表大会, 2009.
 - [11] 岡部雅夫, 小林圭堂, 石川達也, 山口高平, “業務知識の組織的継承へのオントロジー
-

および知識モデリングの活用”，情報システム学会，第5回全国大会・研究発表大会，2009.

- [12] M. Okabe, A. Yoshioka, K. Kobayashi and T. Yamaguchi, “Organizational Knowledge Transfer Using Ontologies and a Rule-Based System”, IEICE Transactions on Information and Systems, Special Section on Knowledge-Based Software Engineering, Vol. E93-D, No.4, pp. 763-773, 2010.
- [13] 石川達也, 小林圭堂, 岡部雅夫, 山口高平, “オントロジーとルールベース・システムを利用した知識変換支援”, 人工知能学会 第24回全国大会, 2010.
- [14] Tatsuya Ishikawa, Keido Kobayashi, Masao Okabe, Takahira Yamaguchi, "Support for Externalization of Intelligence Skill Using Ontology and Rule-based System", Proceedings of the 9th Joint Conference on Knowledge-based Software Engineering (JCKBSE'10), Kaunas University of Technology, pp.145-159, 2010.
- [15] 岡部 雅夫, 小林 圭堂, 石川 達也, 飯島正, 山口 高平, "オントロジーを活用した知的熟練の継承支援", 情報システム学会誌, 2010 (採録決定).

謝辞

本論文を執筆するにあたり、多くの方々から多大なるご指導およびご助言を賜りました。とりわけ、くじけそうになる自分を常に暖かく励まし頂き、丁寧な助言およびご指導を頂くと共に、学会発表などの貴重な機会を与えて頂きました山口高平先生には、心から御礼を申し上げます。

本論文の副査を快諾していただき、執筆にあたり、有益なご助言およびご指導を頂いた、山本喜一先生、櫻井彰人先生、鈴木秀男先生に厚く御礼申し上げます。

本研究を進めるにあたって、折に触れ、適切なご指導およびご助言を頂いた飯島正先生、森田武史先生に心から感謝致します。

本論文は、山口研究室の大平昌弘さん、吉岡亜紀子さん、加藤美穂さん、小林圭堂さん、竹田百合恵さん、石川達也さんの皆様との研究プロジェクトの成果でもあります。皆様との貴重な意見交換により、本論文をまとめることができました。深く感謝いたします。

最後に、多忙な業務の中、貴重な時間を割いていただき、業務の熟練者の立場から様々な情報を提供してくださると共に、技術・技能の継承の実現のために様々なアイデアも出していただいた東京電力株式会社の柳澤雅彦さん、上條員利さん、山崎浩志さん、立花浩さんをはじめとする多くの皆様に厚く御礼申し上げます。