

# 企業における業務知識をシステム開発に活用するための 知識ベースシステムの検討

## A study on a Knowledge Base System to utilize business knowledge the enterprise for system development

亀田 栄一†  
Eiichi Kameda

高木 正則‡  
Masanori Takagi

望月 雅光††  
Masamitsu Mochizuki

勅使河原 可海†  
Yoshimi Teshigawara

† 創価大学工学研究科

‡ 創価大学工学部

†† 創価大学経営学部

† Graduate school of engineering, Soka University

‡ Department of engineering, Soka University

†† Department of Business Administration, Soka University

### 要旨

システム開発において、過去に得られたノウハウが組織に蓄積されることは難しい。また、ナレッジマネジメントシステムなどのツールは、システムによって膨大な情報は蓄積されていても、知識の整理がなされていないため、有効な知識の共有や継承を行うことは困難である。本研究においては、システム開発の支援システムを、知識ベースシステムとして構築することを目指している。本稿では、知識ベースシステムの開発および保守に関する体制、知識ベースシステムの有効性について考察し、システム開発に関する指示を意図したとおりに相手に伝えることを支援する知識ベース型の意味伝達支援システムを提案する。

### 1. はじめに

システム開発に携わる企業において、システム開発プロジェクトを円滑に進められるかどうかは、当事者の経験に依るところが大きい。同じ企業において過去に行われた経験などの知識は、ノウハウとして組織に蓄積されることは難しく、属人的なものになってしまう。また、以前より、ナレッジマネジメントシステムなどのツールを導入して、知識の共有や継承を図る試みがなされているが、その多くがドキュメント管理システムや企業情報ポータルであり、システムによって膨大な情報は蓄積されていても、知識の整理がなされていないため、有効な知識の共有や継承を行うことは難しい。

本研究においては、システム開発を円滑に進めるための支援システムを、知識ベースシステムとして構築することを目指している。本稿ではまず、知識ベースシステムの開発および保守に関するあるべき体制について考察する。次に、知識ベースシステム自体の有効性について検討し、さらにシステム開発に関する指示を意図したとおりに相手に伝えることを支援する知識ベース型の意味伝達支援システムを提案する。

### 2. 研究の背景と目的

開発現場には、システム開発手法や業務知識などの貴重な知識が多く存在する。しかし、現場の人間にとって、何が価値ある知識であるかがあいまいなため、その知識を有効に活用することは難しい。一方、知識工学の分野においては、これまで知識表現手法の研究、その活用について研究がなされてきている。しかし研究の立場からは、実際の開発現場で蓄積されるシステム開発や業務に関する知識、さらには現場が抱えている問題などに触れることは難しい。

本研究においては、現場におけるシステム開発などのノウハウ、およびそこで得たデータを元にするとともに、知識工学における知識表現手法に基づき、知識の体系的整理を試みている。また知識表現のためのツールを活用することにより、知識ベースシステムの開発を行い、それを開発現場において活用することを目指している。研究の背景のイメージを図1に示す。

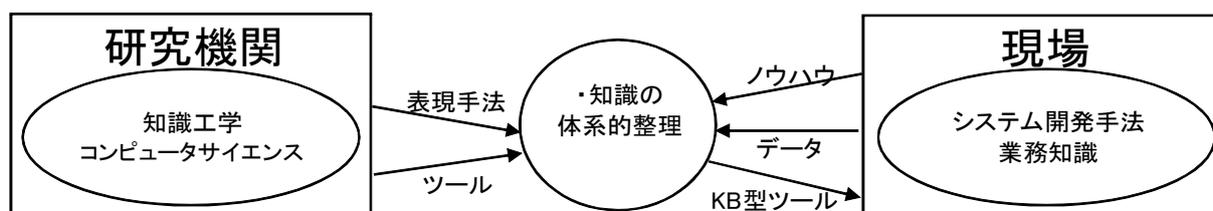


図1 研究の背景

また、開発した知識ベースシステムが価値を持ち続けるために一番重要なことは、いかに知識をメンテナンスしていくかということである。そのためには、知識のメンテナンスを日々の現場での活動の中で行い、実行可能な知識として企業に蓄積されていく仕組みが必要である。本研究においては、業務知識のメンテナンスを現場で行うことを前提としており、価値ある知識が現場で蓄積されることを目指している。

### 3. 知識ベースシステムの有効性

かつて人工知能の分野においては、エキスパートシステムのような業務知識集積型のシステムにより、業務をサポートする試みがなされたが、業務知識自体のメンテナンスにコストがかかりすぎるという問題を抱えていた。これに対し、ILOG社(2008年7月にIBMに買収)のビジネスルール管理システム(BRMS)[2]は、これまでプログラムソースに埋め込まれていた業務知識に関わるビジネスルールをソースから切り離すことにより、ビジネスルールのメンテナンス性を向上させることを試みた。このような知識ベースシステムにより、これまで時間と共に陳腐化していたルールシステムの価値を、保ち続けていくことが可能となる。本研究においては、システム開発に関する知識を分類・整理し、それぞれの知識の特性にあわせたメンテナンス方法を提示することにより、運用可能な知識ベースシステムの提案を行う。

### 4. 知識ベースシステムの提案

ここでは説明のため、メールにおける指示内容を補完するためのシステムを例に取りながら、知識表現のための要件、システム概要について述べる。

#### 4.1. 知識表現の要件

知識表現の枠組みが必要とする要件は次のとおりである。

- (1) **知識表現の範囲**：システム開発の業務に関する知識が表現できる必要がある。例えば、システム開発手法に関する知識、対象業界に関する知識、IT技術に関する知識、会社内におけるルール、マシン名・ネットワーク機器名などのインフラ情報、人事情報などが存在する。
- (2) **業務ルールの表現**：上述したように、複雑な業務ルールが存在するため、それを体系的に整理でき、それを直接的に記述できる表現手法が必要である。
- (3) **動的に変化するリソースの管理**：動的に変化するプロジェクトに関わるリソースの状態が常に管理され、知識ベースシステムから利用できる形になっている必要がある。
- (4) **辞書の管理**：開発案件に関連する用語、その会社独自の用語などが管理され、知識ベースシステムから利用できる形になっている必要がある。

#### 4.2. 知識プログラミング

本研究では、業務ルールとして、「必要情報補完ルール」「妥当性確認ルール」の2つを扱う。「必要情報補完ルール」とは、あいまいな指示に対して本来必要な情報を補完するためのルールであり、勘違いや忘れによる実施ミスを防ぐ。「妥当性確認ルール」とは、その指示自体が妥当であるかどうかをチェックするためのルールであり、「スキルの妥当性」と「権限の妥当性」が含まれる。すべてのルールは、IF-THEN型の推論規則で記述し、条件部と推論結果部に分けることができ、以下の式が成り立つ。

$$C_1 \vee C_2 \vee C_3 \vee \dots \vee C_n \rightarrow A \quad (C_1 \sim C_n : \text{条件部} \quad A : \text{推論結果部})$$

条件部は、「変数 比較演算子 値」の形で表現される。推論結果部は、文字列で表現される。文字列に

変数を含めることが可能である。また、「妥当性確認ルール」の推論結果部は省略することが可能とし、その場合、ルールシステムは、条件部が成り立たない限り、「指示は妥当ではありません」との文字列を返す。競合解消については、最も新しく追加されたルールを優先して実行する。「必要情報補充ルール」および「妥当性確認ルール」の例を図2に示す。必要情報補充ルールの例の場合、指示内容が“ユーザ登録”であり、かつサーバーが“SV001”であり、かつ登録対象が社員マスタに含まれない場合、「社員以外のユーザを登録することはできません。」という文字列を返す。

必要情報補充ルールの例	妥当性確認ルールの例
<b>条件部</b> 指示内容 == ユーザ登録 サーバー == SV001 登録対象 ≠ 社員マスタ <b>推論結果部</b> 社員以外のユーザを登録することはできません。	<b>条件部</b> 指示内容 == ユーザ登録 サーバー == SV001 ユーザ登録権限 ≠ 受信者の権限 <b>推論結果部</b> 受信者のマネージャーに確認を行って下さい。

図2 ルールの例

また、業務ルール適用において使用されるリソースに関する情報を、社内データベースとして保持しておく必要がある。さらに、そのメールの文面等から特定のクラスの用語を抽出するための用語辞書、および類義語に対応するためのシソーラスを用意する。

### 4.3. システムの試作と実験

上述の知識表現の要件に従い、インターフェースを「メール」に限定したシステムを試作した。画面遷移等のメールシステム部分においては、開発ツールとして、Rubix 2.0 IDE (なうデータ研究所) を用いた。当ツールは画面遷移のコントロールがしやすく、業務の流れと密接に関わる画面遷移が変更になった時の対応が簡便であるというメリットがある。また、ルールシステム部分の開発ツールとしては、DSP 5.3 (なうデータ研究所) を用いた。当ツールは、プラント機器[3]、病院情報システム[4]の設計など、多くの知識ベースシステムにおいて用いられた実績がある。それぞれのツールの開発画面イメージを図3および図4に示す。

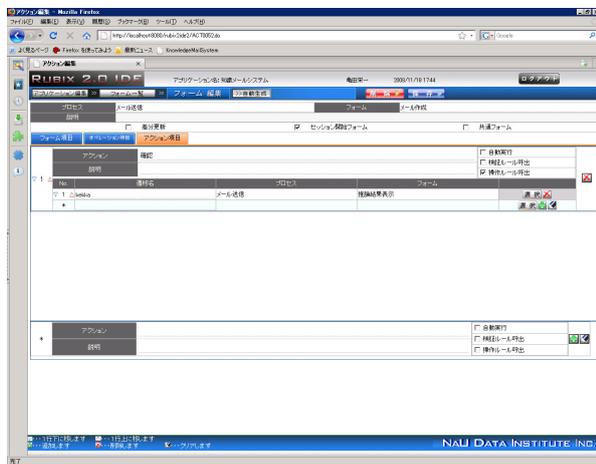


図3 Rubix 2.0 IDE における開発イメージ

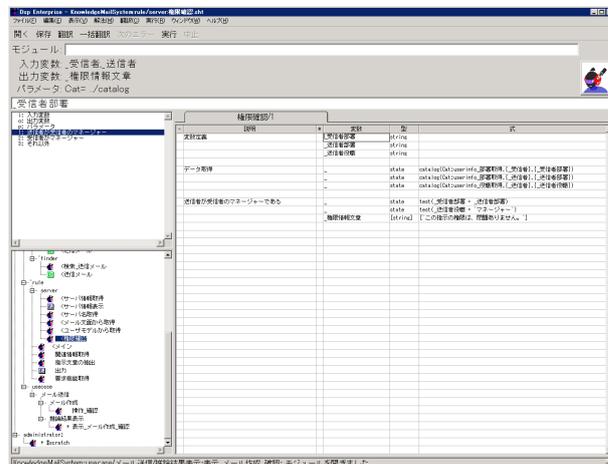


図4 DSP 5.3 における開発イメージ

試作システムの概要を図5に示す。メンテナンス画面(①)には、業務ルールを記述するためのルールエディタ(②)、および用語辞書・シソーラスを記述するための辞書エディタ(③)が含まれる。

送信者が指示を送信(④)すると、メールの文面から、特定の語彙が含まれる文章を指示文章と見なして抽出(⑤)する。抽出された指示文章に対して、指示内容やメール文面の用語から、必要情報補充ルール、および妥当性確認ルールを適用し(⑥・⑦)、その推論結果を送信者のメール送信画面に表示(⑧)し、送信者が確認した上で、メールを送信する(⑨)。

上述のシステム概念に基づいて、知識ベースメールシステム「Knowledge Mail System」を試作し、本手法の有効性を確認する。本システムの機能は、次の2つである。(1)メールの本文を解析し、権限の確認、対象になるシステムの情報等、あいまいな部分を自動的に付与する機能。(2)人やものの状態

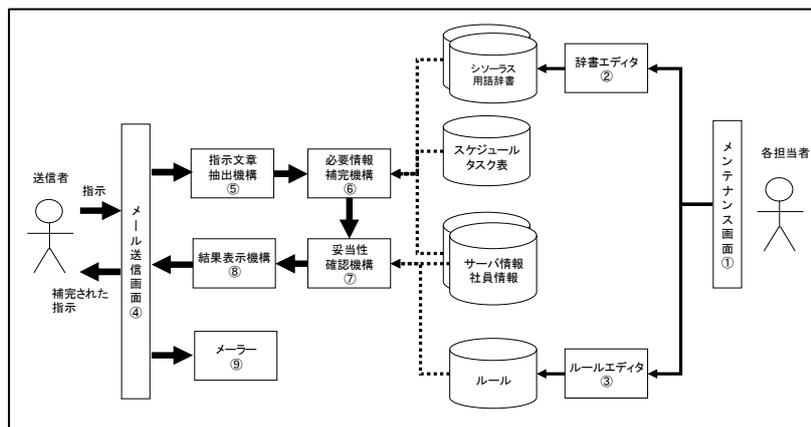


図5 システム構成

や権限などから、指示の妥当性をチェックする機能。

本システムを用いて、「帳票サーバにフォーム F001 を設定して下さい」という指示を行った場合に補足される推論結果を図6に示す。



図6 実験結果

この推論結果に基づき、最初の指示は次のように補完される。「このサーバの所轄部署は、管理部です。このサーバは、帳票サーバです。このサーバの稼働時間は、月～金の8:00～22:00です・・・この指示の権限には、問題があります。Aさんのマネージャーに確認を行って下さい。」

このことにより、指示の送信者がその指示の妥当性を確認することができる。また、指示の受信者が指示の補足情報を同時に受け取ることができる。これらのことより、指示の伝達ミスを少なくすることが可能となる。

## 5. まとめと今後の課題

知識表現の要件を整理し、限られたメールのやり取りを想定して、知識ベースを構築した。また、試作システムを用いて、その有効性を確認できた。

今後は、すべての業務を分析して、実運用に耐えられる知識ベースを構築する必要がある。また、実際の開発プロジェクトでの運用実験を行う必要がある。

## 参考文献

- [1] ウルシステムズ株式会社, 間違いだらけのシステム開発, 翔泳社, 2006.
- [2] <http://www.ilog.com/products/businessrules/>, ILOG 社 ビジネスルール管理システム(BRMS).
- [3] 山口秀行,長澤勲,梅田政信,望月雅光,章志華,プラント機器の基本設計支援のための知識表現モデル, 情報処理学会論文誌,Vol.41,No.11,2000.
- [4] 長瀬啓介,高田彰,五十嵐徹也,大内隆信,網野貴文,大野国弘,Java2 Enterprise Edition を用いた推論エンジンを有する病院情報システムの開発,医療情報学,Vol.23,2003.