

エキスパートシステムを用いた勘定科目の仕訳

杉村 挙[†] 大森 健児[‡]

[†] 法政大学大学院情報科学研究科 〒184-8584 東京都小金井市梶野町 3-7-2

E-mail: [†] i04t0008@k.hosei.ac.jp, [‡] ohmori@k.hosei.ac.jp

あらまし エキスパートシステムを用いて、入力された文章から勘定科目の仕訳を行うシステムを提案する。エキスパートシステムは Java Prolog、意味ネットワークを利用する。文章を分割し、意味ネットワークと照らし合わせて勘定科目を取得する。科目が貸方、借方どちらに属するかを助詞、述部の関係から決定する。

キーワード エキスパートシステム, Java Prolog, 意味ネットワーク

Journalization of account title using an expert system

Aguru Sugimura[†] Kenji Ohmori[‡]

[†] Graduate School of Computer and Information Sciences, Hosei University

3-7-2 Kajino-cho, Koganei-shi, Tokyo, Japan

E-mail: [†] i04t0008@k.hosei.ac.jp, [‡] ohmori@k.hosei.ac.jp

Abstract This research proposes the development of the system that carries on journalization of account title using an expert system. The expert system uses the semantic network and Java Prolog. The system divides sentences, compares with the semantic network, and gets account headings. After that, the subject is decided between a debtor or a creditor by relationship of particles and predicates.

Keyword semantic network, expert system

1. はじめに

日本版 SOX 法の施行が始まろうとしているが、これに伴って、企業での内部統制の基幹となっている会計システムのあり方が問われている。伝票を集めて経理課システムでの一括処理から、会計的な行動が発生するスポットでの現場入力へと、対応を求められている。会計のシステムは、仕訳伝票に代表されるように一定の専門的知識を必要とするが、現場入力となったときは、会計の知識を有する者が処理するわけではなく、販売や購買、

あるいは、商品の生産を担当するものが入力することとなる。このため、日常的な言葉での入力システムが強く求められている。

会計システムは、資産の増加を示す借方と資産の減少を示す貸方との間で現金、商品、人件費、消耗品などの変化に伴う企業活動が生じたときに、それを記述することで、資産の変化、生じた利益(損益)、現金の流れなどを示せるようにしたものである。借方と貸方は資産の種類ごとに、現金、預金、売掛金、買掛金、販売原価

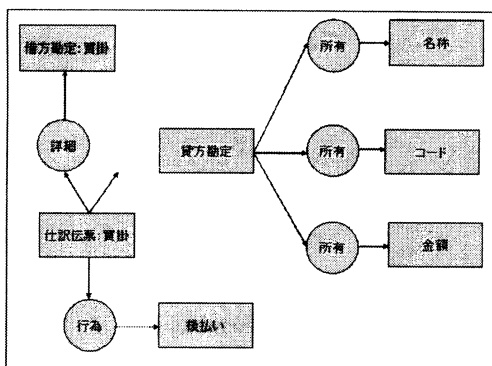


図 1 仕訳伝票

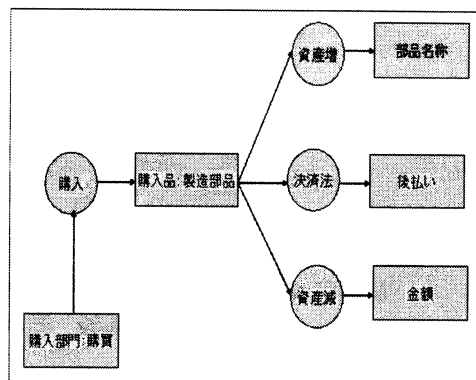


図 2 購入部門

などの勘定科目に分類されている。勘定科目は、商法による一定のルールがあるものの裁量で許されている部分が多く、企業ごとに異なっている。また、多くの場合、それぞれの勘定科目は、コード化されているが、これは基本的には企業ごとに異なる。このため、SOX 法施行後の会計システムでは、会計的な行動を日常的な言葉で入力できるようにするとともに、それぞれの企業の勘定科目の体系にあわせてこの入力を柔軟に変換できるシステムが必要とされている。

今回、日常的な言葉での会計的な行動を入力するシステムの開発を行ったのでこれについて説明を行う。

2. 概念グラフ

会計システムでは、日常的な会計的な行動を企業で定めた勘定科目あるいは勘定科目コードに変換する必要があるが、これは会計的な知識を必要とする。会計の知識は、幸いなことに、使われている単語も限定されているだけでなく、きわめて論理的に構成されているため、コンピュータの中で利用することが可能であると考えられる。

ここでは、会計の知識を表わすために、概念グラフ(Conceptual Graph)を用いる。概念グラフは、John Sowa(1986)がまとめ

たものであるが、その起源は 19 世紀の Charles Peirce である。概念グラフは、概念を四角で表わし、概念同士の関係を円で表し、それらを結んだものである。John Sowa は概念グラフが一階述語論理と同値であること、即ち、概念グラフで表現されるものは一階述語論理で表わされることと、また、一階述語論理で表わされるものは概念グラフで表わされることを示した。

会計的な知識をコンピュータの中で利用する場合には、知識を再構成する必要があるため、その開発は試行錯誤的にならざるを得ず、開発途中のまとまってきた知識の体系を分かりやすく表現する必要があるが、概念グラフはこれを行うのに極めて適した知識である[1]-[3]。

今回開発したシステムでは、会計的な知識を概念グラフで表現し、その知識をベースにして、日常的な言葉で示された会計的な行為を、殆どの会計システムでの入力の基本となっている仕訳伝票に変換する。

3. システムの概要

現場での入力は、例えば、購買部であれば、「2006年1月14日に、製品Aを作るために、部品製造会社Bから、単価400円の部品Cを100個購入し、

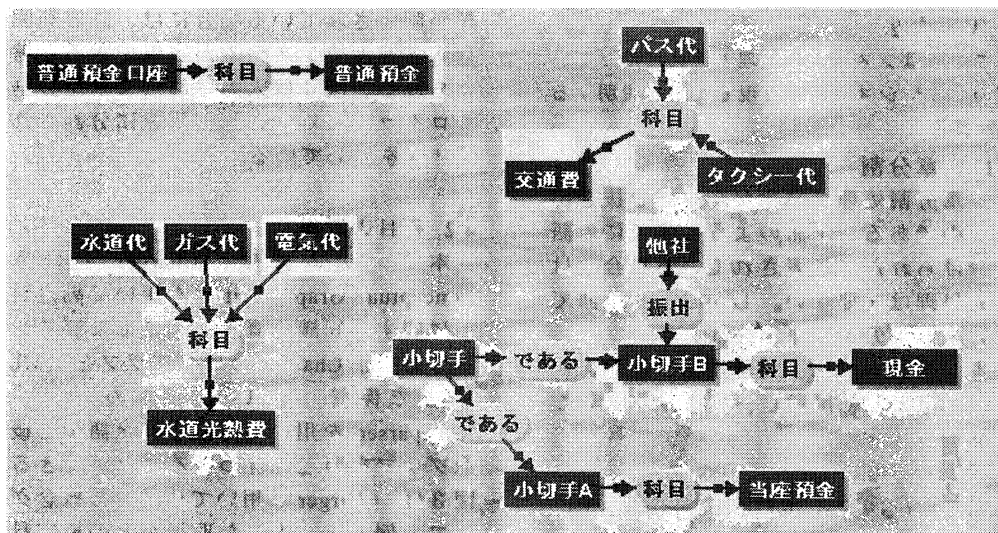


図 3 Conceptual Graph

2ヵ月後に支払うこととした」となる。この取引は、会計システムでの仕訳伝票では、借方は（仕入）40000円、貸方は（買掛金）40000円と表される。

概念グラフでは、図1，2に示すように、買掛金と購買部の購入については示すことができる。この図では、仕訳伝票と勘定科目と購入部門の知識が記述されている。勘定科目については一般的な知識、仕訳伝票と購入部門については、それぞれ買掛と購買のインスタンスに関する知識の記述である。現場からの入力、ここで表されている知識を基にして、仕訳伝票を作ることになる。

図で説明したような概念グラフをそのまま用いることはシステムの開発を難しくするので、今回開発したシステムでは、論理的構造と限定的な語彙を特徴とする会計の体系を利用しただけでなく、概念間の関係が日本語では多くの場合動詞により表されることを利用した。

システムの概要は次のようになっている。開発したシステムはWebベースである。クライアントはそれぞれ現場でFirefoxなどのブラウザを用いて出張、物品の購入、商品の販売などの入力を行う。この入力はいんターネットを経由してサ

ーバとなっている本システムへと送られてくる。サーバ側のシステムは、J2EEにPrologの機能を付加したものとなっている。J2EEは主要な3つの部分で構成され、1つはブラウザとのやり取りを行うWeb、1つはデータベース、残りの1つはWebとデータベースの間に介して、トランザクションの処理を行うEJBで構成されている。本システムでは、この入力を自動的に仕訳けるが、仕訳はいくつかの作業を通して行われる。

最初に行う作業は文章の分割と形態素解析である。会計で用いられる語彙を基に作られた辞書を利用して、文章をその構成要素に分解する。

次に、この分解された構成要素と、概念グラフとのパターンマッチングを行う。このパターンマッチングは概念グラフを用いての勘定科目の取得、助詞を用いての勘定科目の借方、あるいは貸方の決定、金額の確定を行う。パターンマッチングが済むと貸方と借方の勘定科目とその金額が得られるので、それをデータベースに記憶する。

4. 実現方法

ここではシステムの処理の流れに沿って、どのようにシステムを実現したかを説明する。

4.1. 文章分割

文章分割処理は文章から単語を抜き出すものである。英語のようにすでに単語に分けられた形で表されている場合にはこの処理は必要ない。しかし、日本語では必要不可欠な処理である。

本システムでは、会計用語、助詞、記号を辞書として用意し、最長一致検索に

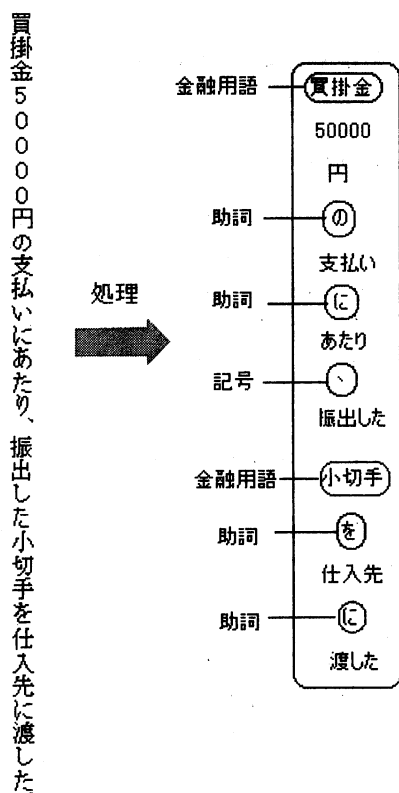


図 4文章分割一例

基づいて文章の分割を行う。最長一致検索は、日本語入力プログラムに多く用いられている手法だが、検索に用いる語彙の種類が多く、長い文章の場合には処理時間が長くなる。しかし、本システムのように入力される文章の分野、語彙の種

類が限定されている場合には、許容範囲の応答速度でシステムが実現できると考えられる。図4は入力された文章が分割プログラムで処理され、語彙に分割された状態を表している。

4.2. 科目の取得

本システムでは、Charger という Conceptual Graph Editor を用いて勘定科目に関する知識を概念グラフでモデリングした[4]。Charger は概念グラフを XML 形式で保管している。このため、XMLparser を用いて分割された語彙と概念グラフを照らし合わせることができる。図3は、Charger を用いて作成した概念グラフの例で、発生した取引がどの勘定科目に分類されるかを表している。ここでの概念グラフは、概念を次の3種類の概念関係で結んでいる。

- ① 包含関係を表した「である」
- ② 勘定科目の関係を表した「科目」
- ③ 主語と述語の関係を表した概念関係

分割した語彙から勘定科目を取得するには、分割した語彙と一致する概念から「である」で結ばれた概念を辿っていき、「科目」で結ばれている概念を探しだせばよい。ただし、図3で表した小切手の関係のように一つの概念から二つ以上の概念関係へリンクがされていることがある。この例は、他社が振出した小切手は現金に分類され、それ以外の小切手は当座預金に分類されることを示している。この時の処理は、条件的な概念関係を持ち、その関係概念を満たすものを優先的に探索する。

4.3. パターンマッチング

勘定科目が貸方、借方どちらに属するかを決定する場合、勘定科目は、資本、資産、負債、収益、費用の5つに分類される。分類された科目の増減を判断し、表1に示した仕訳ルールに従って貸方、借方を決定する[5]。本システムでは、科

目情報に付随する助詞、述部を用いてパターンマッチングを行うことで、その科目の増減を推測し、以下のように判断する。

- 「から」が付随する科目、金額情報は貸方科目に属する。
- 「に」「へ」が付随する科目、金額情報は借方科目に属する。
- 「を」「が」「で」が付随する科目、金額情報は上記二つの助詞の関係、および述部との兼ね合いによって科目が決定される。
- 「として」が付随する情報は「を」、「が」の情報と置換的な関係にある。
- 「の」は「を」、「が」が付随する情報を修飾する。
- 「と」は同格の助詞なので、その後に出てくる助詞により判別する。

表 1仕訳ルール

	借方	貸方
資産	資産の増加	資産の減少
負債	負債の減少	負債の増加
資本	資本の減少	資本の増加
収益	収益の取消し	収益の発生
費用	費用の発生	費用の取消し

5. 環境

システム開発においてプログラミング言語は、Prolog と JAVA を用いた。Prolog は文字処理において優れた能力を発揮する利点と、グラフィック環境を前提にした API が無い、日本語の入力に対応した処理系がほとんどないことが欠点である。今回のシステム開発では、文字処理や推論を Prolog で記述し、ユーザーインターフェースやデータベース処理部分は EJB、JSP を用いて JAVA で記述する。Prolog で開発したソースコードは最終的に PrologCafe と呼ばれる JAVA Prolog を用いて JAVA に変換するので、最終的にはすべて JAVA によって構成される[6]。そのため、Prolog の持つ利点を活かし、欠点

を補ったプログラムを作成することができる。また、JAVA の特徴であるオペレーティングシステムや CPU の種類に依存せずに実装することができる。

サーバは JBOSS を用いた。JBOSS はアプリケーションサーバの機能だけでなく、データベースサーバ、Web サーバとしての機能も保有しているので、今回の開発ではそれを利用することとした。Web サーバの機能は Struts、データベースサーバの機能は Hypersonic によって提供される。また、EntityBean の永続性管理をコンテナが行う CMP 方式を利用し、データアクセス機能を実現する。図 5 は、今回開発したシステム構成図である。

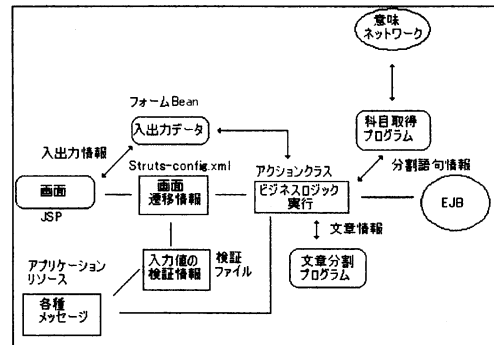


図 5システム構成図

6. 結果・考察

図 6 はシステムに仕訳の一例を入力したものである。入力した文章は「練馬営業所から電気代 5250 円、交通費 10500 円、買掛金 4200 円の支払い報告を受け、同額を他社が振出した小切手で補充した。」というものである。この仕訳は、貸方科目が現金 19950 円、借方科目が買掛金 4200 円、交通費 10500 円、水道光熱費 5250 円となる。図 7 は、図 6 で入力された文章をシステムで処理し、結果を出力したもので、正しい仕訳を行っている。

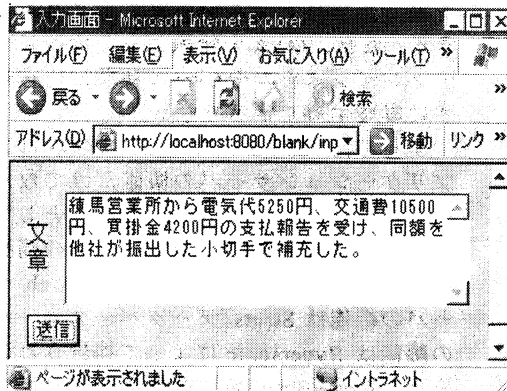


図 6 事象入力

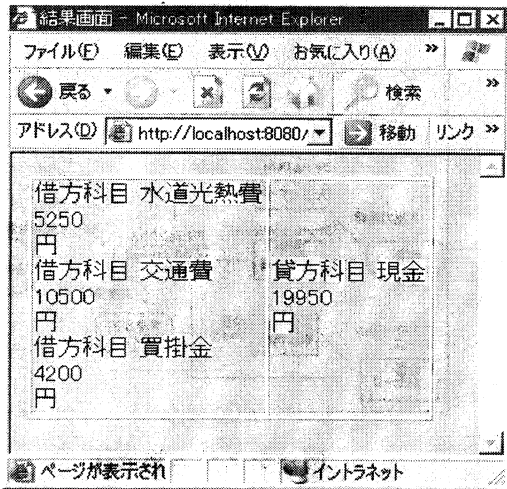


図 7 出力結果

図 6、7 に示した例以外にも 20 ほどの例を入力し、システムの評価を行った。結果としては、正確な仕訳が行えたのはそのうちの 7 割ほどであった。図 4 の例とは違い 2 文かそれ以上の事象は、本システムによる判断が難しい。また、金額の流れが明確でない例も仕訳が行えない場合が多かった。仕訳における間違いは致命的である。このため、本システムを実用的にするにはシステムの精度を上げることが不可欠となる。

今後の課題としては、上記の問題点の改善である。2 文以上に渡る場合の処理やシステムの精度をさらに高くするには、

指示代名詞などの係り受けや文章の意味を解析すること、格解析のように構文を解析することが必須となる。

文 献

- [1] 笠原要, 稲子希望, 加藤恒昭, “テキストデータを用いた類義語の自動作成”, 人工知能学会論文誌, vol.18, no.4, pp221-232, 2000
- [2] J. F. Sowa. Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/ Cole, Pacific Grove, CA, 2000.
- [3] <http://www.jfsowa.com/krbook/>
- [4] <http://www.cs.uah.edu/~delugach/CharGer/body.html>
- [5] 陣川公平, “勘定科目の処理がすぐに行える事典”, 日本実業出版社, 2003.
- [6] 神戸大学学術情報基盤センター:Java Prolog「Prolog Café」, <http://kaminari.istc.kobe-u.ac.jp/PrologCafe/>