

解説



仕様獲得と知識獲得

1. 仕様獲得 vs. 知識獲得†

折原良平†† 荒木大†† 西村一彦††

1. はじめに

ソフトウェアシステムの開発における要求分析・定義の重要性が指摘されてから久しい。SADT¹⁵⁾、SREM¹⁹⁾、プロトタイプングパラダイム²⁵⁾など、さまざまな技法やツールが提案されてきたにも関わらず、いまだに決定版と言えるものは存在しない。

こうした中、エキスパートシステムの普及にともなう知識獲得ボトルネック¹⁶⁾の認識に合わせて、要求分析・定義の困難さを知識獲得ボトルネックとして捉える見方が現れた^{20), 27)}。従来の要求分析・定義では、ユーザの要求を明確な仕様書として定義するという、いわば「はじめに要求ありき」の立場であったのに対し、新しい見方では、そもそもユーザから要求を抽出することが困難であるとの認識に立っている。このような立場の研究においては、要求定義あるいは要求分析という言葉に代わって、仕様獲得または要求獲得という言葉が用いられることが多い^{13), 26)}。

仕様獲得*の研究においては、ユーザから要求を抽出することが知識獲得の1インスタンスであることは暗黙のうちに仮定されており、仕様獲得がどのような特徴をもった知識獲得のインスタンスであるのかまでは語られないのが普通である。しかし、知識獲得としての仕様獲得を明確に特徴付けることなしには、知識獲得手法を効果的に仕様獲得に適用することはできないであろう。

逆に、エキスパートシステムも一つのソフトウ

ェアシステムである以上、知識獲得という工程は広い意味での仕様獲得の一部であるという見方もできる。知識獲得研究の成果が、知識獲得工程のごく一部を支援するに留まっている現在、広い視野から知識獲得を眺めなおすことには意義がある。

本稿では、獲得される知識の種類という観点から、仕様獲得を知識獲得の一種として特徴付けることを試みる。特徴を際立たせるため、エキスパートシステム開発における知識獲得との対比も行う。また、仕様獲得の視点から知識獲得をみることで、エキスパートシステム開発の上流工程への計算機支援の可能性を探る。

2. 仕様獲得、知識獲得とは

最初に、本稿における仕様獲得、知識獲得が何を指すかを明らかにしよう。

エキスパートシステム（以下 ES）の開発における知識獲得とは、大きく分けて、知識の抽出、知識の変換、知識の構造化・洗練化を意味する。知識の抽出とは、領域専門家のもつ問題解決のための知識を第三者に理解可能な言語あるいは図表に形式化する作業を意味する。この作業は、知識エンジニア（KE: Knowledge Engineer）という第三者によってインタビューや問題解決過程の観察といった形態で行われることが多いが、適当な表現形式が与えられれば領域専門家自身で行うことも可能である。知識の変換とは、抽出した知識を知識ベースとしての実装形式に変換することを意味する。知識抽出段階での表現形式がそのまま実装形式であることが理想である。知識の構造化・洗練化とは、知識相互の関連性を明確化する、知識ベースの不備な部分を補うなどの操作により、知識ベースを質的に整備する作業を意味する。

一方、知識獲得という言葉は狭く捉えて、専門

† Requirements Acquisition vs. Knowledge Acquisition by Ryohei ORIHARA (Systems and software engineering lab., Toshiba corp.), Dai ARAKI (Systems and software engineering lab., Toshiba corp.) and Kazuhiko NISHIMURA (Systems and software engineering lab., Toshiba corp.).

†† (株)東芝システム・ソフトウェア技術研究所

* 本稿におけるこの言葉の意味はすぐ後に述べるが、それまでは「いわゆる要求獲得または仕様獲得」のことを仕様獲得と呼ぶことにする。

家からの知識の抽出のみを指す立場もある。本稿では、この狭義の場合に従う。

ところが、仕様獲得に対しては、このような定説がまだない。現状では、図-1 に示すシステム構築²⁾の、計画から設計の段階におけるさまざまな知識の抽出や表現の変換をすべてこの言葉で表しているようである。具体的には、

1. 問題設定あるいは要求仕様設定：ソフトウェアシステムの導入によって解決しようとする問題記述の獲得、あるいは、その裏返しとしての、ユーザ自身の問題領域の言葉で記述したソフトウェアシステムの仕様の獲得*
2. 設計仕様への変換：要求仕様を、設計者の言葉で言い換えたものの獲得。

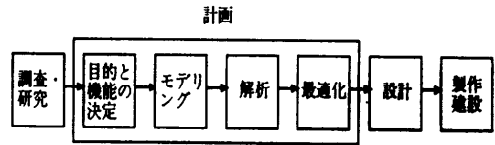


図-1 システム工学の一般的なアプローチ

の二つがこの名前で呼ばれている。本稿では、前者を要求仕様獲得、後者を設計仕様獲得と呼ぶ**。

図-2(a)は、ES 開発を、KE、領域専門家***、ユーザ、領域専門家の果たすタスク****、作られる ES***** という 5 者から眺めたものである。KE の対話相手は領域専門家であり、その果たしているタスクは、作るべき ES と同じである。一

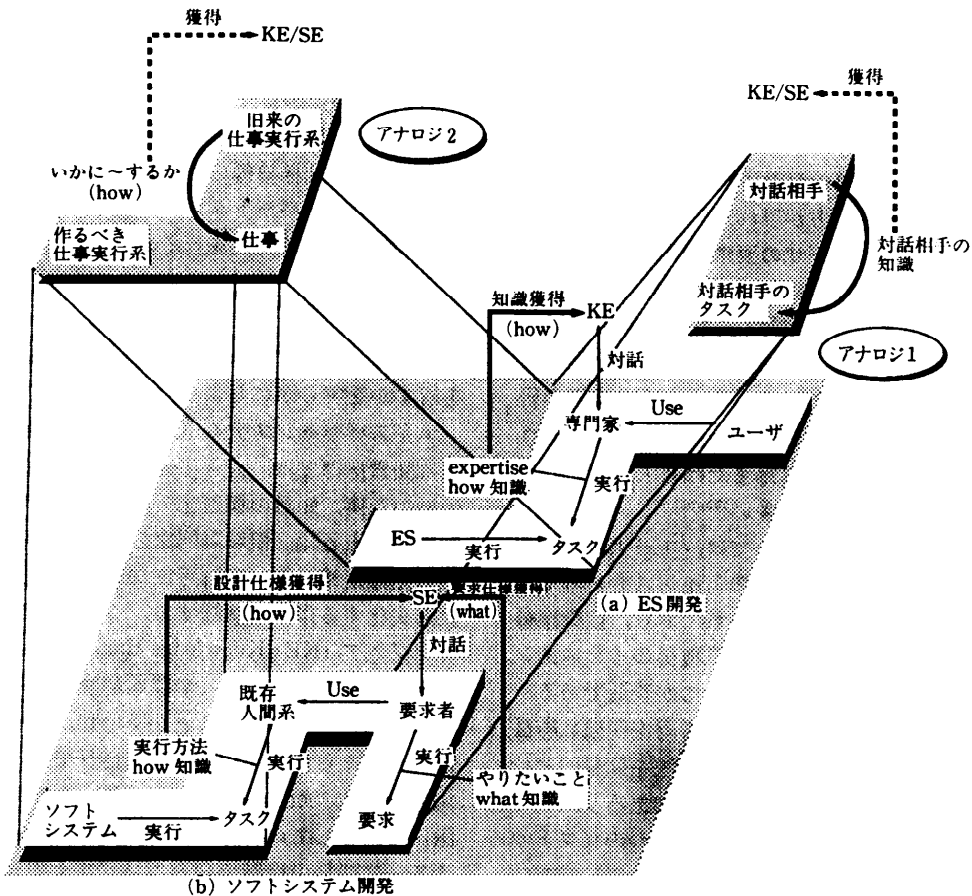


図-2 仕様獲得と知識獲得のアナログ

*これを、たとえば requirements specification と behavioral specification というように分けている例⁽¹⁾もある。

** 混乱を避けるため、あえて一般的でない言葉を用いた。既存の言葉を用いれば、前者を要求獲得、後者を仕様獲得と呼ぶべきところか。

*** 図中では“専門家”。

**** 図中では“タスク”。

***** 図中では“ES”。

方、図-2(b)は、一般ソフトウェアシステム開発を、SE、要求者、要求者の果たすタスク^{*}、既存人間系、既存人間系の果たすタスク^{**}、作られるシステム^{***}という6者から眺めたものである。SEの対話相手は要求者であり、その果たしているタスクは、「このソフトウェアシステムは…のようなタスクを果たさねばならない」と述べることであり。つまり、要求者の果たしているタスクは、作るべきソフトウェアシステムそのものの果たすタスクとは異なる。これらのことから、ES開発とソフトウェアシステム開発の間には次の2通りのアナログが考えられる。

1. 要求仕様獲得と知識獲得のアナログ

図-2(b)のSEと要求者間の対話を、要求者の果たすタスクに関する知識の獲得と考えると、要求仕様獲得と知識獲得のアナログが検出される。なぜなら、要求仕様獲得は、要求者のもっている要求者に関する知識(要求仕様)の、要求者からSEへの移行であり、知識獲得は、専門家のもっている専門家に関する知識(expertise)の、専門家からKEへの移行だからである。要求者は、自らの要求に関する専門家であると考えられるから、この意味で、要求仕様獲得は知識獲得の一種であると言える。診断型/計画型などの個々の領域での知識獲得のための技術を要求仕様獲得に応用するために必要となるのは、このアナログに基づく類推^{****}である。

2. 設計仕様獲得と知識獲得のアナログ

図-2(b)の“対話”を、既存人間系の果たすタスクに関する知識の獲得と考えると、設計仕様獲得と知識獲得のアナログが検出される^{*****}。設計仕様は設計者の言葉で記述されるとはいえ、個々の部分は当然問題領域に依存しており、また、そこでは一般に要求仕様獲得のときよりは詳細な記述が求められるので、設計仕様獲得においては、SEは作ろうとするソフトウェアシステムの設計仕様のうち、問題領域依存の部分を獲得す

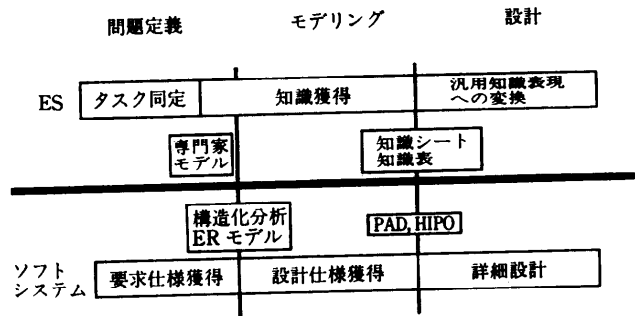


図-3 ESとソフトウェアシステムの構築プロセス

る必要がある。一方、知識獲得は、作ろうとするシステムの内部仕様^{*}のうち、問題領域非依存としてあらかじめ用意しておくことのできない部分をKEが獲得する作業である。したがって、この両者にもアナログがみてとれる。この場合には、ESは一つのソフトウェアシステムであるので、知識獲得は設計仕様獲得の一種であると言える。知識獲得のための技術を個々のソフトウェアシステムにおける設計仕様獲得に応用するためには、こちらのアナログに基づく類推^{**}(すなわち一般化)が必要である。

最後に、システム構築全体における位置付けという観点から、要求仕様獲得、設計仕様獲得、知識獲得を眺めてみよう。

システム構築全体における要求仕様獲得、設計仕様獲得および知識獲得の位置付けを表したのが図-3である。ここでは、問題定義、モデリング、設計というフェーズ分けは、時間の流れとは無関係に、単にプロセスの切り分けを表すものとする。

したがって、本稿における知識獲得とはおおむねモデリングのフェーズに属するが、ES開発ではシステム構築が螺旋型に進むので、問題定義的要素から完全に独立ではありえない^{***}。

一方、要求仕様獲得は明らかに問題定義フェーズであり、設計仕様獲得はモデリングのフェーズである。古くから要求定義で用いられている言葉を使うなら、要求仕様獲得が扱うのはwhat型の知識であり、設計仕様獲得が扱うのはhow型の知識であると言える。

* 図中では“要求”。
 ** 図中では“タスク”。
 *** 図中では“ソフトウェアシステム”。
 **** 類推の言葉で言えば、ソース領域がたとえば診断/計画型ES、ターゲット領域が要求者を構築するモデルと言える。
 ***** 一般ソフトウェアシステム開発においては、既存人間系に関する知識は、要求者から間接的に与えられることが多い。

* 次章の言葉を用いるなら、「具体化された専門家モデル」。
 ** 類推の言葉で言えば、ソース領域がES、ターゲット領域が、そのとき自動化しようとしている業務に対応するソフトウェアシステムという場合である。
 *** 初期KB構築時など。

図-4に、本稿で用いた用語と、ソフトウェア工学、知識工学、システム工学において一般的と思われる用語の対応表を示す。

ソフトウェア工学	知識工学	システム工学	本稿
要求定義	問題の設定	問題定義	要求仕様獲得
概念設計	知識の抽出	モデリング	設計仕様獲得/知識獲得

図-4 用語の対応

3. 仕様獲得と知識獲得の違い

3.1 トップダウン獲得 vs. ボトムアップ獲得

知識獲得においては、現状システムのタスク分析・整理が進められてきた結果、generic task⁴⁾のような形式でタスククラスごとでの問題解決方式の定式化と問題解決知識の分析が行われてきた。これらの成果は特定のタスクを対象を特化したエキスパートシステム構築ツール、すなわちドメインシユル³¹⁾を産み、エキスパートシステム開発の生産性向上に大きく寄与している。

知識獲得技術の研究においては、この考えをさらに進め、知識獲得のさまざまな過程、すなわち、知識の抽出、知識の変換、知識の構造化・洗練化のおおのこの段階に対する支援ツールを、特定のタスク向けに定義した専門家モデルをベースに用意しようとしている。たとえば MOLE⁹⁾では、ドメインシユル MOLE-p は Cover-and-Differentiate という問題解決手法を専門家モデルとしてもっており推論エンジンが用意されている。専門家モデルを具体化する作業は、「診断事象を説明できる仮説として何があるか？」という、診断問題の空間を Cover するための知識と、「何がそれらの仮説を差別化するか？」という Differentiate するための知識を獲得することである。知識獲得支援ツール MOLE-ka は、この2種類の知識を最小限・漏れなく収集するためのインタビュー戦略を備えている。このような枠組みで構築された知識獲得支援ツールは、KE を必要とせず、nonprogrammer である専門家自身だけでエキスパートシステムを開発することをも可能としている¹⁾。

このように、知識獲得においては、問題定義のフェーズにおいて、専門家モデルのライブラリから、専門家が行っているタスクがどれに当たるかをまず選択し、実際の知識獲得はこのモデルを基にトップダウンに進められることが理想的である。このような環境の実現例としては Spark, Burn and Fire Fighter⁹⁾がある。

一方、要求仕様獲得においては、このような頼

るべき枠組、すなわち generic task に相当する強力なものが存在しない。オブジェクト指向モデル⁵⁾など、きわめて抽象的な枠組は存在するが、適用の自由度が高過ぎて、オブジェクトの切り分けの難しさなどの問題が発生する。代わりに、問題を整理するためにしばしば KJ 法などの発想技法が用いられる^{10), 11)}。また、既存人間系に追いつくことが目標である ES 開発とは異なり、一般のソフトウェアシステムでは、実現しようとするシステムに対応する実世界の系は必ずしも存在しないし、存在した場合でも、実現しようとするシステムはある意味で実世界の系以上の能力を要求される。したがって、現状システムの分析に対する努力が怠られがちなので、設計仕様獲得でもこのようなガイドが存在することは少ない。獲得作業はボトムアップに進められるのが普通である。

3.2 獲得者と被獲得者のコミュニケーション

エキスパートシステム開発におけるシステム計画は、作ろうとする ES が診断・計画などのいくつかのタスクのどれとどれによって実現できるか、あるいは、今相手にしている領域専門家の、どのタスクによって実現できる部分を開発対象として選択するかを決めることである。各タスクに対しては、専門家モデルが付属している*。これは作るべき ES の内部仕様のうち、問題領域に依存しない部分であり、なんらかの計算機システムの言葉で記述されるべき部分である。専門家から問題領域に依存する部分だけを獲得すればよいということは、専門家は問題領域の言葉だけで思考することができることを可能とする。これが専門家の負担を軽くし、適切なモデル作りに貢献している。つまり、専門家モデルは、システム計画において、KE が必要以上に問題領域に関わらなくても良いようにし、モデリングにおいて、ES の設計仕様の計算機領域を領域専門家から隠蔽しているのである。

これに対し、要求仕様獲得は、既存のタスクへ

*現状では、これが完全に整備されているのは診断などに限られており、他のタスクでの整備が待たれる。

のあてはめではない。SE が要求者からの獲得に当たって頼れるような、問題領域に依存しない、かつタスクに特化された表現形式は今のところ存在しない。さらに、オブジェクト指向などの枠組は実現寄りであるため、要求者にその上で考えてもらうのは無理がある。そこで、通常 SE が問題領域を学ぶことを要求される。これは、SEにとって大きな負担となる。

また、設計仕様獲得においては、問題領域依存の知識を埋め込むベースとなる設計仕様は、非常に実現寄りであり、要求者に理解できないものが多い^{*}。この作業では、要求者は計算機システムの言葉で思考することを強いられる。したがって、獲得された仕様はユーザにとって理解しにくいものであり、その確認のためには、プロトタイプングによって疑似的に問題領域の言葉へと変換することが必要となる。

このように、知識獲得における専門家モデルに当たる抽象度をもった設計仕様のモデルが仕様獲得において存在しないことは、要求仕様獲得と設計仕様獲得の両方で SE～要求者間のコミュニケーションギャップを生み出している。

なお、知識獲得および仕様獲得における獲得者と被獲得者のコミュニケーションに関する問題点は、本特集の別稿^{17), 18), 25)}に詳しい。

4. 仕様獲得と知識獲得の技術的相関

知識獲得の技術を仕様獲得にどう生かすか、またその逆については、本特集の別稿^{29), 30)}に詳細な事例とともに解説されている。ここでは、要求仕様獲得と設計仕様獲得の違いに注目した上で、両者の技術的相関を概観するにとどめる。

4.1 設計仕様獲得が知識獲得に学ぶべきもの

2. で述べたように、知識獲得の知見を設計仕様獲得に生かすには、両者とも作るべきシステムの問題領域依存の部分を獲得するものだというアナロジに従う必要がある。このアナロジによって、知識獲得におけるタスクごとの専門家モデルの存在からの類推を行うと、一般のソフトウェアシステムの開発においても、タスクごとに、問題領域に依存しない設計仕様の概形を用意すればよいということになる。

このように考えると、現在 EDP 分野などで実

用化されている 4GL²⁴⁾ を利用してのソフトウェアシステムの設計仕様獲得は、知識獲得にきわめて近いといえる。4GL は、現状分析が十分に行われてきた数少ない分野において、その成果を結晶化したものである。

今後、他の分野においても知識獲得ライクなトップダウンな設計仕様獲得を行うためには、現状分析の結果を系統立てて残していき、問題領域の言葉で書かれた仕様を獲得することが必要であると言える^{*}。

4.2 要求仕様獲得が知識獲得に学ぶべきもの

2. で述べたように、知識獲得の知見を要求仕様獲得に生かすには、両者とも対話相手のタスクに関する知識を獲得するものだというアナロジに従う必要がある。すると、インタビュー²⁷⁾ や、発想支援^{12), 20), 21), 23)}などの初期 KB 構築時に用いられる対話手法の有効性が明らかになる。

このアナロジによって、知識獲得におけるタスクごとの専門家モデルの存在からの類推を行うと、「要求する」というタスクを分析して、問題領域に依存しない要求仕様の概形を用意すればよいということになる。しかし、人間のおかれた環境と、その人間の要求することを抽象的にまとめるこの作業は、心理学や医学を初めとして多くの知見を要するものとなるであろう。

4.3 知識獲得が仕様獲得に学ぶべきもの

エキスパートシステムがソフトウェアシステムである以上、設計仕様獲得や要求仕様獲得の知見は、そのまま知識獲得に生かせるはずである。こうした見地から、構造化分析・設計法 (SA・SD) の考え方がエキスパートシステム開発に役立つのではないかという見通しがある³⁰⁾。

また、知識獲得は若干問題定義の要素を含んでいる (図-3)。たとえば、診断型 ES の知識獲得においては、診断されるべき兆候の集合は知識獲得によって得られるが、これは問題定義フェーズに属するものである。したがって、要求仕様獲得技術に見るべきものができれば、知識獲得はそれを使うことができるであろう^{**}。

さらに、知識獲得の指す範囲を少し広げて、知識ベースの管理・検証までも対象とすれば、そこではプログラムの検証技術が生かせるであろう。

^{*} @NIX²⁸⁾ や RA²⁹⁾ はこれを行っている。

^{**} 発想支援は、この事例とも考えられる。

^{*} 典型的には、プログラムそのものに近い。

5. おわりに

いわゆる仕様獲得を要求仕様獲得と設計仕様獲得の二つに分け、それぞれについて知識獲得との相関および相違点について考察した。知識獲得は、システム構築のフェーズからすると設計仕様獲得に当たり、タスクごとの専門家モデルというアイデアは設計仕様獲得に直接応用可能である。しかし、知識獲得においてこれが整備できたのは、ES という狭い範囲を対象にしていたからであり、一般ソフトウェアシステムの開発においてこれを整備していくには大変な努力を要するであろう。

要求仕様獲得は、知識獲得とは別のフェーズであるが、対話相手のタスクに関する知識の獲得という共通点があり、やはり知識獲得の知見を適用できる。しかし、現時点では、「要求」というタスクの定式化は難しく、タスクごとの専門家モデルというアイデアを適用することは困難と言わざるをえない。現時点で適用可能な技術としては対話手法がある。むしろ、なんらかの形で要求仕様獲得技術が進歩したとき、その知見を ES 構築の計画フェーズに応用することが期待される。SA/SD の適用³⁰⁾は、その曙光となりうる可能性がある。

参考文献

- 1) Araki, D. and Kojima, K.: KASE Project Toward Effective Diagnosis System Developments, In *Proc. 6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop* (1991).
- 2) 浅居喜代治: 現代システム工学の基礎, オーム社 (1979).
- 3) Barstow, D.: Domain-specific automatic programming, *IEEE Trans. on Soft. Eng.*, Vol. SE-11, No. 11, pp. 1321-1336 (1985).
- 4) Chandrasekaran, B.: Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, *IEEE Expert*, Vol. 1, No. 3, pp. 28-30 (1986).
- 5) Cord, P. and Yourdon, E.: *Object-oriented analysis*, Prentice-Hall, Second edition (1991).
- 6) Eshelman, L.: MOLE: A Knowledge-Acquisition Tool for Cover-and-Differentiate Systems, In Marcus, S. editor, *Automating Knowledge Acquisition for Expert Systems*, pp. 37-80, Kluwer Academic Publishers (1988).
- 7) 川口敦生他: インタビューシステムのためのシェル, SIS の開発, 第 1 回人工知能学会大会論文集, pp. 343-346 (1987).
- 8) 木下哲男: 要求知識の獲得に関する一検討, 第 3 回人工知能学会大会論文集, pp. 715-718 (1989).
- 9) Klinker, G. et al.: Usable and reusable programming constructs, *Knowledge Acquisition*, Vol. 3, pp. 117-135 (1991).
- 10) 小林重信: 知識システム技術の現状と将来, 計測と制御, Vol. 27, No. 10, pp. 859-868 (1988).
- 11) 國藤 進: 知識獲得と学習研究の新しい流れ, 人工知能学会誌, Vol. 3, No. 6, pp. 741-747 (1988).
- 12) 國藤 進他: グループ知識獲得システム GRAPE 構想, 第 15 回システム・シンポジウム予稿集, pp. 47-52 (1989).
- 13) 久野禎子他: 仕様獲得支援 システム—オブジェクトモデルによる仕様獲得—, 第 4 回人工知能学会大会論文集, pp. 657-660 (1990).
- 14) Lowry, M. and Duran, R.: Knowledge-based Software Engineering, In Barr, A. et al., editor, *The Handbook of Artificial Intelligence Volume IV*, chapter XX, pp. 241-322, Addison Wesley (1989).
- 15) Marca, D. A. and McGowan, C. L.: *SADT™ Structured Analysis and Design Technique*, McGraw-Hill, Inc. (1988).
- 16) 溝口理一郎, 角所 収: 知識獲得支援システム, 人工知能学会誌, Vol. 3, No. 6, pp. 732-740 (1988).
- 17) 溝口徹夫: SE の要求仕様獲得の実際—要求仕様決定の要因と課題, 情報処理, Vol. 33, No. 6, pp. 617-619 (1992).
- 18) 尾方 説: 知識エンジニアにおける知識獲得の実際, 情報処理, Vol. 33, No. 6, pp. 620-622 (1992).
- 19) 御前宗平: TRW 社の SREM, bit 臨時増刊ソフトウェア・プロダクト工学, Vol. 13, No. 5, pp. 568-584 (1981).
- 20) 折原良平, 今野 宏: 発想支援システム「知恵の泉™」の試作, 日本ソフトウェア科学会第 8 回大会論文集, pp. 5-8 (1991).
- 21) Orihara, R. et al.: On Paraphrasing-based Analogical Reasoning—as a theoretical base of the abduction support system, In *Proc. of ALT '90*, pp. 134-148 (1990).
- 22) Rich, R. et al.: Toward a Requirements Apprenticeship, In *Proc. of 4th International Workshop on Software Specification and Design*, pp. 79-86 (1987).
- 23) 篠原靖志: 未整理な情報からの知識ベース構築, 情報処理学会第 34 回 (昭和 62 年前期) 全国大会 2 K-4, pp. 1465-1466 (1987).
- 24) Sommerville, I.: *Requirement Validation and Prototyping*, chapter 6, pp. 109-122, Addison Wesley, Third edition (1989).
- 25) 鈴木他: SE の仕様獲得の実際, 情報処理, Vol. 33, No. 6, pp. 612-616 (1992).
- 26) 玉井哲雄: 要求定義とプロトタイピングの現状, プロトタイピングと要求定義シンポジウム論文集, pp. 103-110 (1986).
- 27) 土田賢省他: 仕様獲得実験システム, 情報処理学会第 39 回 (平成元年後期) 全国大会 2 S-1, pp.

1515-1516 (1989).

- 28) 上原邦昭: 要求獲得の知的支援, 人工知能学会誌, Vol. 6, No. 2, pp. 170-172 (1991).
- 29) 上原邦昭: ソフトウェア設計における仕様獲得支援, 情報処理, Vol. 33, No. 6, pp. 631-640 (1992).
- 30) 山口高平: 知識獲得支援技術はソフトウェア開発方法論に適用可能か?, 情報処理, Vol. 33, No. 6, pp. 623-630 (1992).
- 31) 人工知能学会(編): 人工知能ハンドブック, chapter VII-1, pp. 570-592, オーム社 (1990).

(平成3年11月8日受付)



折原 良平 (正会員)

1963年生. 1986年筑波大学第三学群情報学類卒業. 1988年同大学院工学研究科電子・情報工学専攻博士前期課程修了. 同年(株)東芝入社.

現在, 同社システム・ソフトウェア技術研究所に勤務. 発想支援技術の研究, 類推の研究に従事. 人工知能学会, 日本ソフトウェア科学会各会員.



荒木 大 (正会員)

1962年生. 1986年大阪大学通信工学科卒業. 1988年同大学院工学研究科博士前期課程修了. 同年(株)東芝入社. 以来, システム・ソフトウェア技術研究所に勤務し, 知識獲得支援技術の研究開発, 帰納学習ツールの研究開発に従事. 人工知能学会, IEEE 各会員.



西村 一彦 (正会員)

1963年生. 1986年東京理科大学理工学部経営工学科卒業. 1988年同大学理工学研究科経営工学専攻修士課程修了. 同年(株)東芝入社. 現

在, 同社システム・ソフトウェア技術研究所に勤務. システム仕様化支援技術の研究に従事. 人工知能学会, 日本ソフトウェア科学会各会員.

