

システム設計における要求モデル構築支援

Supporting the Modeling of Requirements in System Design

角 康之*¹ 堀 浩一*² 大須賀 節雄*³
 Yasuyuki Sumi Koichi Hori Setsuo Ohsuga

- * 1 (株) エイ・ティ・アール 知能映像通信研究所
 ATR Media Integration & Communications Research Laboratories, Kyoto 619-0288, Japan.
- * 2 東京大学先端科学技術研究センター
 Research Center for Advanced Science and Technology, The University of Tokyo, Tokyo 153-8904, Japan.
- * 3 早稲田大学理工学部
 School of Science and Engineering, Waseda University, Tokyo 169-8555, Japan.

1998年4月3日 受理

Keywords: system design, concept formation, requirement model, requirement acquisition, case base, automated modeling.

Summary

This paper describes a system for supporting the construction of requirement models, which are initial computable models representing users' requirements in system design. The proposed system principally consists of two components, a tool for aiding the formation of requirement concepts by visualizing a user's thought space, and a knowledge-based system which automatically assembles the ascertained requirement concepts into a requirement model. The system extracts reusable components of a requirement model, corresponding to the users' abstract requirement concept, from a store of similar past cases. The components are then automatically arranged using heuristic reasoning. By using the system, users can make their requirement concepts more mature, and simultaneously get computable requirement models as by-products.

1. ま え が き

本論文は、システム、特にソフトウェアを系の中心に持つシステムを設計する際の要求モデル構築を支援する方法論を提案するものである*¹。

今日、我々は様々な問題解決や日々の作業にコンピュータを利用するが、このことは、その作業対象となる世界をモデリングし、その一部をコンピュータの世界で処理可能な形式に変換し、その処理をコンピュータに委託することにほかならない。したがって、コンピュータを利用したシステムの設計を考えると、コンピュータ上で動作するソフトウェアの開発についてのみ切り出して議論するのでは不十分であり、本来の問題解

決や作業が存在する対象世界のモデリングを含んだ広い視野で議論する必要がある。従来の知識処理研究は、システム設計という観点で見たとき、コンピュータ内のモデル表現やモデル操作については十分な議論を行ない成果をあげてきた。しかしそれらは、設計に必要な知識はあらかじめ整備されコンピュータ内に表現されていることを前提としているため、未整備な知識や要求をコンピュータ内のモデルとして表現する、といった設計の最上流行程は完全に人間任せであった。そこで我々は、ユーザの心的世界に近い未整備で不完全な知識を扱える発想支援系をユーザと知識処理系の間の仲介として利用することを考える [堀 97]。

ソフトウェア工学の現状を見ながら、もう少し具体的な話をしよう。これまで、自動プログラミングなどの知的ソフトウェア開発支援に関する多くの研究がなされており、成果をあげてきた [Lowry 91, Rich 86]。そのため、ソフトウェア開発において人間が担う仕事の比

*¹ 本論文では、コンピュータ上でプログラムされ動作する部分を「ソフトウェア」と呼び、ソフトウェアおよびそれを利用するユーザや状況を含めた体系全体を「システム」と呼び分ける。

重は、コンピュータで問題をどうやって解くか (How) を記述する作業から、コンピュータに何をやらせたいか (What) を記述する作業に移行している [Borgida 85]. 一方、多くの分野でコンピュータを用いたシステム統合が進んでいる現代社会においては、コンピュータの専門家のみでなく、システムのユーザとなる適用分野の専門家を中心としてシステムの要求定義作業がなされることが望まれる。しかし、コンピュータの専門家ではないユーザの要求を獲得し、それをシステムとして実現するための設計仕様として記述するのは困難な作業であり、要求定義のための知的支援環境の発展が求められている [内平 92, White 91].

堀と大須賀は、ソフトウェア設計におけるユーザの要求分節支援の重要性を指摘した [Hori 92]. 彼らは、ソフトウェア設計を二つの自己組織する世界、すなわち、ユーザやソフトウェア開発者のアイデアによって形成される心的世界と、そのアイデアが計算可能なものとして形式化されるコンピュータの世界の間のすり合わせと考えることを提案している。この二つの世界の組織化はコンカレントになされる。したがって、ソフトウェア設計を包括的に支援するには、ユーザの心的世界における要求概念の分節支援と、そして、その要求概念をコンピュータの世界で処理できる形式に構造化する作業の支援の、両者をコンカレントに進めることを促進する環境を提供することが不可欠であると考える。本論文はそのような枠組を提案し、試作したシステムを紹介するものである。

本論文で我々は、発想支援系と知識処理系を結合した、要求モデル構築支援システムを提案する (2章). 発想支援系は、ユーザの心的世界に近い表現を持つ概念空間を表示し操作可能にするものであり、要求概念の形成や分析を助ける (3章). 知識処理系は、事例知識を利用して、ユーザの要求概念をコンピュータが解釈可能な要求モデルに変換する作業を支援する (4章). 本論文では、提案システムを新規システム設計と既存システム仕様の変更・管理に適用した例を紹介し、提案した方法論の評価を行なう (5章).

2. 提案システムの概観

図1に、本論文で提案する要求モデル構築支援システムの構成を示す。想定されるユーザは、例えば、情報工学の研究者であり、研究のテーマやアプローチを模索しつつ、それを具現化するシステムを試作しようとしている状況であるとする。そのような場合、ユーザが具体的なシステム要求を前もって持っていないか、

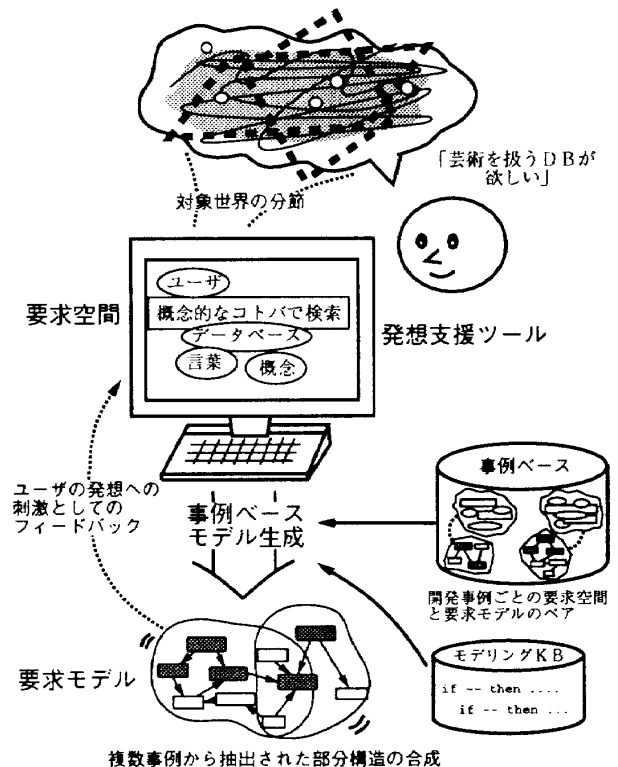


図1 要求モデル構築支援システムの構成

ユーザの問題対象世界が未分節なことが多い。そこで、ユーザは発想支援ツールを利用し、システム要求の断片をメモとして書き出し、自分の要求空間を構築する。発想支援ツールは要求空間を可視化し、ユーザの要求獲得や要求分析を助ける。提案システムは、ユーザの要求空間構築に応じて、過去の開発事例や汎用的なモデリング知識を用いて、構築中の要求空間に対応する要求モデルのプロトタイプを生成し、ユーザに提示する。ユーザは、提示された要求モデルに修正を加えたり関連する過去の事例を分析することで、自分の要求概念をより構造化された要求モデルとして認識し分析することが可能となる。さらにユーザは、発想支援ツール上で要求空間の更新作業を続けることができ、その結果は要求モデルの更新に反映される。以上の作業を繰り返すことにより、ユーザは自分の要求概念を成熟させると同時に、その副産物としてコンピュータ内で表現・操作可能な要求モデルを獲得することができる。

また、最終成果物として生成される要求空間と要求モデルは対で事例ベースに蓄えられ、次回以降のシステム利用の際に事例知識として利用される。つまり、提案システムの事例ベースは利用するユーザやグループの利用状況や作業ドメインを反映しながら成長していく。システム開発の下流工程のみでなく上流工程においても過去の成果を知識として利用するには、ドメインに依存した分析やモデリングが必要であることが主

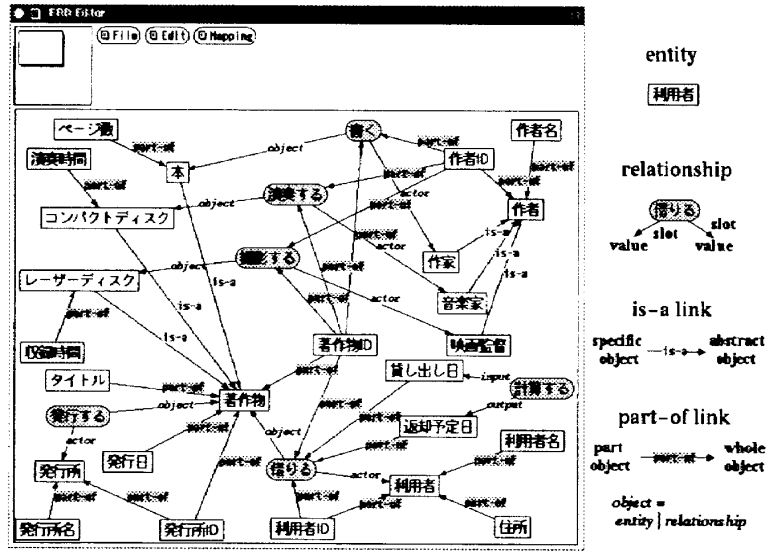


図2 要求モデルの例

張されているが [Arrango 91, 田村 94], 本研究で提案する方法論はこの課題に対する一つの回答であると考えられる。

以下に、提案システムの成果物となる要求空間と要求モデルについて簡単に説明する。要求空間とは、ユーザの要求概念やそれに関連するアイデアの関係を可視化する二次元空間である。ユーザが入力した要求やアイデアの断片としてのメモと、それらのキーワードをアイコンとし、統計的手法によりそれらの関係を可視化したものである。主にドメイン依存かつ抽象的な表現によって構成される傾向がある。要求空間は、抽象的な概念の世界と、より構造化されたコンピュータの世界の間の仲介として利用される。

要求モデルとは、ユーザの要求を下流工程（設計、実装、評価、管理）に伝えるための初期モデルである。図2に、図書館の情報管理システムを題材にして構築した要求モデルの例を示す。要求モデルは、ソフトウェアとして実現される範囲に限らず、ユーザの要求や問題が存在する対象世界をモデル化したものであり、実体を表すオブジェクト（図中の長方形アイコン）とそれらの間の関連を表すオブジェクト（図中の長円形アイコン）が構成するグラフとして表現される。関連オブジェクトは対象世界内に存在する動作を表すことが多く、図2の例にあるように、「計算する」、「登録する」といったソフトウェアに要求される機能を表すこともあれば、「借りる」、「書く」といったソフトウェア開発に直接は関わらないものもある。関連オブジェクトはフレーム構造を持ち、複数の属性を有し、実体をその属性値としてとる。実体間のリンクの特殊形として、抽象-具体構造を表す is-a リンクと、全体-部分構造を表

す part-of リンクを用意した。

3. 発想支援ツールによる要求空間の形成支援

本論文で提案するシステムは、[角 96]で提案された CSS と呼ばれる発想支援ツールを要求空間構築支援に利用する。以下、本論文に関連する範囲で CSS を簡単に紹介する。CSS は、ユーザが入力したテキストオブジェクトと呼ばれる電子化されたメモのアイコンを二次元空間中に自動配置することで、ユーザが形成中の概念空間を可視化する。ユーザは CSS を利用する際、テキストオブジェクトごとに複数のキーワードを指定し、その重要度を主観的に与えることが求められる。CSS は、このデータを用いてテキストオブジェクトを多変量データとして解釈し、ユーザのリクエストに応じて最新データに統計処理を施し、空間構造を再計算する。具体的には、データの固有値計算を行ない、固有値が上位である二つの固有ベクトルで空間の二軸を規定し、テキストオブジェクトとキーワードのアイコンを二次元空間中に自動配置する。直感的に説明すると、空間中では、より多くのキーワードを共有するテキストオブジェクトは近くに配置され、それらのキーワードはその近くに集まる。

図3は、新規ソフトウェア設計のアイデア形成のために、CSS を利用している際のコンピュータ画面例である。CSS は UNIX ワークステーションの X Window System 上で実装され、マルチウィンドウのグラフィカルユーザインタフェースを提供する。図中のメインウィンドウが、要求空間を表示している。空間中の長方形

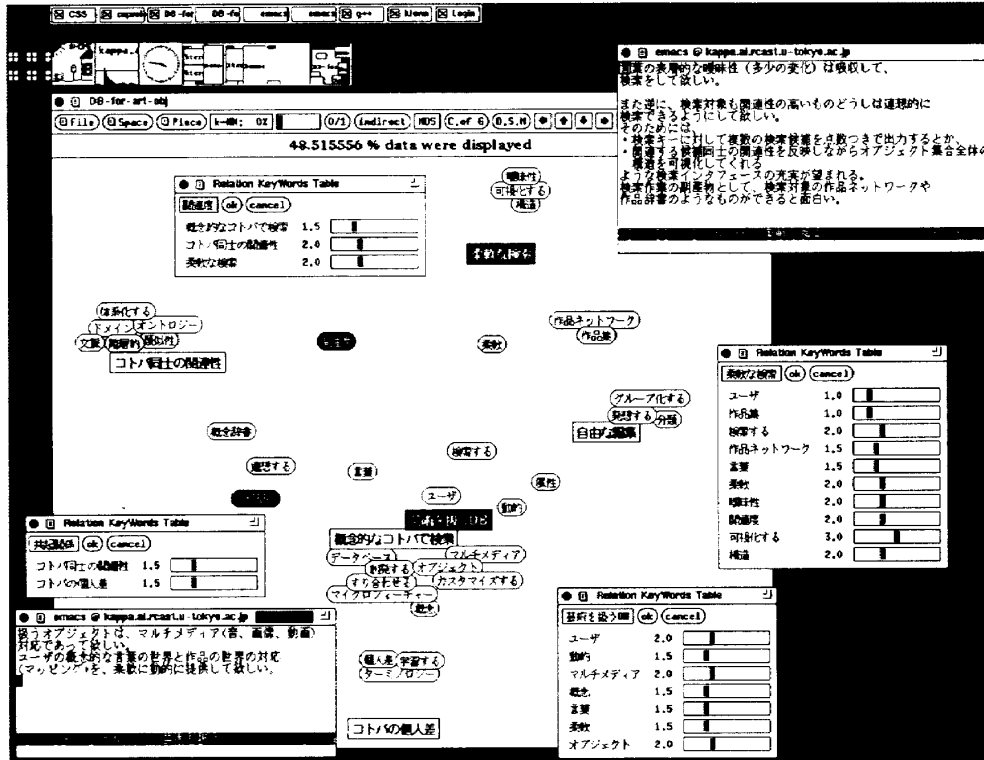


図3 発想支援ツールを用いた要求空間形成の例

のアイコンが、要求概念の書かれたテキストオブジェクトを表し、長円形のアイコンがそれらのキーワードを表す。

CSSは、テキストオブジェクト中のテキストの内容や長さなどを制限しないので、ユーザは、システム開発の上流行程で書きためられたドキュメントや、関連すると思われる研究メモなどをテキストオブジェクトとして自由に利用することができる。CSSを利用することで、ユーザは全体的な見通しを見失わずに自分のアイデアや要求を再考することが可能になる。そして、CSSが提示した空間中に生じるテキストオブジェクトやキーワードのクラスタを要求のモジュールとして認識し、さらにそれらの位相的な関連を把握することが容易になる。

ユーザは作業成果を蓄積して新しい思考作業に再利用することができるので、既存システムの概念構造を可視化するツールとしてもCSSは有効である。例えば、新規のシステム設計の要求空間構築の際に作業に行き詰まったときは、過去の要求空間から関連のあるものをたたき台として作業を始めることが効果的である。また、CSSを用いて、構築中の要求空間と関連事例の要求空間をすり合わせることで、構築中の要求空間中の曖昧で抽象的な要求と、事例要求空間中の具体的な要求モジュールの関連性を可視化することも可能である[角94]。

4. 要求概念のモデリング

4.1 知識処理アプローチによるモデリングの自動化

CSSを用いて認識された要求概念をシステム設計に利用するには、コンピュータ上で表現・操作可能なものにし、より構造化された要求モデルへ変換する必要がある。なぜなら、要求空間はユーザの頭の中にある要求概念の一部を外在化したものであり、ユーザが要求概念を認識・分析することを助けるものではあるが、要求概念そのものはユーザの頭の中にある。したがって、その要求概念をその後の行程である設計、実装、評価につなげるには、コンピュータ内で操作可能なより形式的表現を持つ要求モデルへ変換しなければならない。

要求概念を満たす要求モデルの構築は、1) 初期モデルの生成、2) モデルの解析と評価、3) モデルの修正、といった作業を繰り返す一種のモデル変換であり、人間が設計型の問題解決をする際の普遍的なパラダイムに属する[Ohsuga90]。新規にソフトウェア設計を行なう場合、認識された要求概念と一対一に対応する要求モデルが過去の事例から見つかることは稀であるので、様々な知識を用いながら試行錯誤で徐々に要求概念を満たすモデルへと接近する以外に方法はない。

本研究では、要求空間中に現れた要求概念から要求モデルへの変換を半自動化することを目標とする。上

で述べたように、要求概念そのものはユーザの頭の中にあるので、システムが提示するモデルがユーザの要求概念を満たすかどうかの判断はユーザ自身が行なうよりほかない。そこでここでは、初期モデル生成タスクと、モデル修正のためのモデル操作タスクを自動化した。そうすることで、ユーザは自分の要求空間に対応した仮の要求モデルを自動的に得ることができ、このことは要求概念の評価を容易にし、CSS上での再考を促進するので、要求空間がより成熟したものに作る作業を助けると考えられる。

我々は、上記のモデリングに関わるタスクの自動化を進めるために、知識処理アプローチを採用し、過去の事例の知識、すなわち、過去のシステム開発において構築された要求空間と要求モデルを利用する（図1参照）。また、要求モデルのプロトタイプ、評価、修正に利用される汎用的なタスクプログラムを記述し、モデリング知識ベースに蓄えた。プログラムとして記述したタスクは、モデルのプロトタイプ、複数事例をまたぐモデル中の要素間の関連の発見、特定の視点に基づいたモデルの再構成、モデルの意味的制約充足の評価などである[角 95]。本論文では、この中のモデルのプロトタイプについて説明する。

これらのタスクプログラムの記述には、東京大学で開発されてきた知識処理システム KAUS [Ohsuga 90] を利用する。KAUS は知識表現と推論のための宣言的プログラミング言語であり、構造情報の記述・操作に優れている。本研究では以下の用途で KAUS を利用する。

- プロトタイプ中のモデルと事例ベース中のモデルの記述、管理、操作
- モデリングタスクのプログラミングと利用
- モデリングのための推論エンジン

ユーザの要求空間に対応するモデルをプロトタイプするために、以下のタスクの自動化を試みた。

- ユーザが要求空間中に見いだした要求概念モジュールに対応する要求モデル部品を、事例の要求モデルから抽出する。
- 複数の事例から抽出された要求モデル部品を合成して、ユーザの要求空間全体に対応した要求モデルをプロトタイプする。

次節以降でこれらのタスクの自動化について説明する。

4・2 事例からの要求モデル部分抽出

本節では、ユーザの要求空間中に現れる要求概念モジュールに対応する要求モデル部品を、事例の要求モデルから抽出する手法を説明する。要求空間中の要求

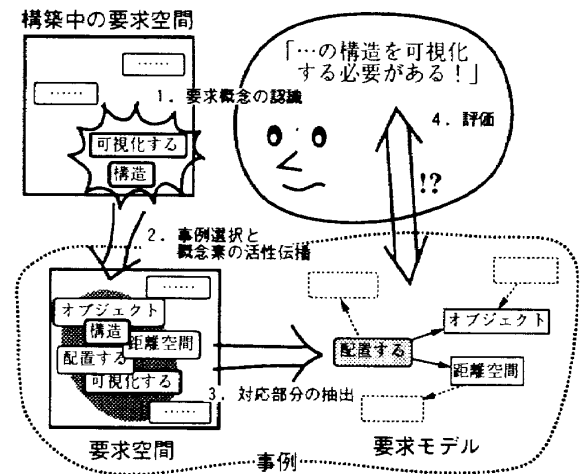


図4 事例からの要求モデル部分抽出の概念図

概念素（キーワード）の表現は要求モデル中のオブジェクト（モデル中の実体や関連）のそれに比べて抽象度が高いため、要求空間中の要求概念モジュールに対応するモデル部品を事例の要求モデル中から直接見つけることは困難である。そこで、事例の要求モデルと対で保存されている要求空間を仲介として利用する。つまり、まず要求空間同士で要求概念素のマッチングをとって事例選択を行ない、それからそれと対をなす要求モデル中から対応するモデル部品を抽出する。

以下に、要求モデル部分抽出の手順を、例をあげながら示す（図4参照）。

- (1) ユーザが要求空間中の二つのキーワード「可視化する」と「構造」が近くに配置されたことに注目し、「…の構造を可視化する必要がある」という要求概念を認識したとする。そこで、ユーザはその要求概念に対応する要求モデル部品を事例から探し出そうと考え、二つのキーワードを指定する。
- (2) システムはキーワード「可視化する」と「構造」を要求空間内に含む事例を選択し、その要求空間中において、それらのキーワードの周辺に位置する他のキーワードに活性を伝播させる。図4の例では、「オブジェクト」、「距離空間」、「配置する」といったキーワードに活性が伝播している。
- (3) 対をなす要求モデル内に、これらの活性したキーワードと表現が一致する実体/関連オブジェクトが見つかった場合、それらのオブジェクトを含む部分構造を抽出する。
- (4) ユーザは、事例から抽出された要求モデル部品が、ユーザの意図した要求概念と関連しているか評価する。

要求概念を表すキーワードを入力とし、それに対応する要求モデル部品を複数の事例から抽出するタスク

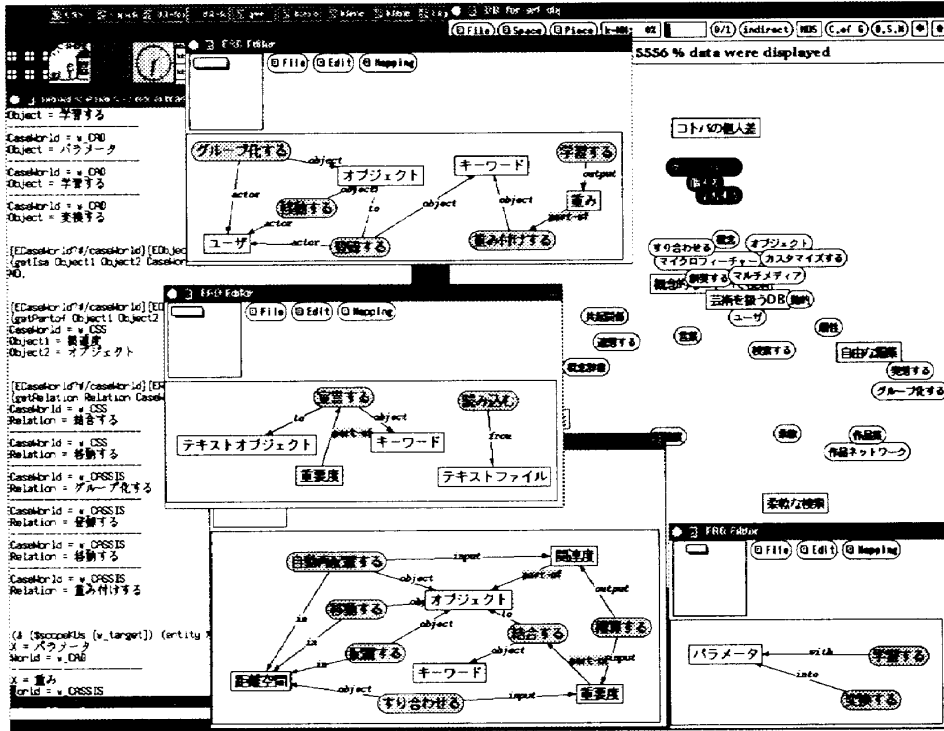


図5 複数事例からの要求モデル部分抽出の例

を KAUS プログラムとして実装した。実装に際しては、活性伝播の閾値を可変にしたり、事例選択の戦略を2種類用意して選べるようにし、思考過程に応じた使い分けを検討した [角 95]。

図5は、要求モデル部分抽出を実行している際のコンピュータ画面例である。画面の右上にあるのが、CSSを利用してユーザが構築中の要求空間である。この例ではユーザは、空間中にクラスタを形成しているキーワード集合「ターミノロジー」、「個人差」、「学習する」に注目し、この要求概念モジュールに対応する要求モデル部品を事例から抽出している。画面左にあるのが KAUS を実行しているコマンドウィンドウであり、KAUS による推論の様子を見ることができる。推論結果は KAUS で記述されたモデルとして出力され、グラフィカルな構造ブラウザで表示され、ユーザに提示される。この例では四つの事例から要求モデル部品が抽出され、表示されている。

本システムを利用することで、ユーザは抽象的な表現による要求概念から、具体的な要求モデル部品を得ることが可能になる。そのため、ユーザはシステムから提示されたものの中から自分の要求概念を満たすものを選択し、それを修正して、要求モデル構築に利用することができる。提示されたものが直接再利用可能でないとしても、これらはユーザの要求の更なる分析のためのたたき台として有効であると考えられる。

4・3 要求モデルの合成

更なる自動化を進めるために、複数事例から抽出された要求モデル部品を合成して、ユーザの要求空間全体に対応する仮モデルを自動生成するタスクを実装した。前節で説明した要求モデル部分抽出タスクは、単独事例から部品を抽出するものであったのに対し、ここで説明するタスクは複数事例から抽出した部品を合成するものである。したがって、前者は事例内の文脈の保存を保証するが、後者は生成される仮の要求モデル全体としての文脈の一貫性を保証しない。このことは、複数文脈の衝突、つまり、新たな構造の創発を引き起こす可能性を持つ。

一般的に合成問題においては、構造衝突への対処が重要な課題となる。ここでは、異なる事例から抽出された部品同士を合成する際の矛盾を解消する必要がある。例えば、同一名の関連オブジェクトが別々の事例から抽出され、かつ、それらが異なるフレーム構造を持っていた場合、どちらかを選ぶ必要がある。このような現象は図5の例でも観察され、例えば関連オブジェクト「移動する」が二つの事例から抽出され、それらのフレーム構造は異なっている。本研究では、例えば、「問題となっている関連オブジェクトの各候補のフレームの属性を比較し、属性値となるオブジェクトと同じ表現を持つキーワードがより多く要求空間上に見つかった候補を採用する」といったヒューリスティクスをモ

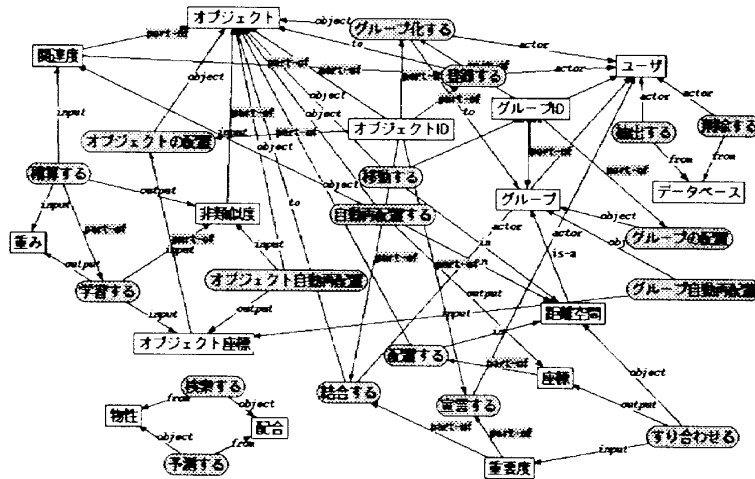


図6 自動生成された要求モデルのプロトタイプのプロトタイプ

デリング知識の一種として記述し、それらを利用して構造衝突対処を自動化した。

図6は、図3に示した要求空間を入力として自動生成された要求モデルである。これは、複数事例からユーザーの要求空間に対応すると推論された要求モデル部品を抽出し、それらを合成したものを、構造ブラウザで可視化したものである。このように、ユーザーはCSSを利用して要求空間を構築すると、それに対応する要求モデルのプロトタイプを直ちに得ることができる。このことは、要求獲得、分析、モデリングをコンカレントに進めることを可能にする。

5. 実験と評価

5・1 新規システム設計における試用実験

筆者の一人がユーザーとなり、新しい研究課題を想定して、アイデア外在化からシステム設計のための要求モデルのプロトタイプまで、一貫した作業の試用評価を行なった。

まず、ユーザーは漠然と「芸術を扱う感性データベースシステム」の開発を研究課題として想定した。そして、断片的な研究アイデアやシステムへの要求仕様のメモを書きため、それらを六つにまとめてテキストオブジェクトとし、CSSを用いて要求空間の外在化、構造化を行なった。そのスナップショットが図3である。

次に、要求空間中に現れた要求概念に相当する要求モデルを得るため、4・2節で示した手法を用いて、事例から要求モデルの部分抽出と評価を行なった(図5参照)。ユーザーは、抽出された要求モデル部品だけでなく、それぞれの要求空間を見ることができる。このことは、各部品がどのような文脈で形成されたのかを理解し、自分の要求を再考する上で有効であった。ま

た、複数事例間の横断的な関連に気づき、類推的な思考が促進された。なおこの実験では、事例ベースに蓄積する既存事例として10の事例を集め要求仕様ドキュメントを利用して要求空間を用意した。また、既存モデルを用いたモデリングの動作を確認するために、特にその中の四つについては筆者らが要求モデルを作成して、事例ベース中に格納した。

さらに、4・3節で説明した手法を用いて、複数事例の知識を用いた要求モデルのプロトタイプへの動作実験をした。図6がシステムの出力である。これは、ユーザーに要求全体の矛盾や言葉の定義の非一貫性を気付かせ、その結果、CSSを用いた要求空間の再構築を促した。また、要求空間の中で、そのプロトタイプされた要求モデルに反映されていない部分は、既存事例に見られない、その研究独自のアイデアである可能性があることをユーザーに認識させる効果があった。

提案システムは、単に、要求モデルだけを蓄積した事例ベースを用意し、単純なキーワード検索で関連事例を抽出するものではない。ユーザーの心的世界に属する抽象的な表現とコンピュータの世界に属する具体的な表現を併せ持つ要求空間が、要求モデルとユーザーの間の仲介となっているため、ユーザーは事例システム全体の文脈を認識しながら、事例から抽出された要求モデル部品を再利用することができるのである。そして、複数事例の文脈の衝突が新しいアイデアの発見を促進し、さらなる要求空間の成長を助ける。このようなサイクルを繰り返すことで要求空間や要求モデルに外在化された言葉は、ドメイン依存のオントロジーとしてグループの貴重な財産となる、と考える。

5・2 既存のシステム仕様の変更と管理への適用

グループによるシステム開発および管理において、

コンピュータ上で操作可能な要求モデルが果たす役割を知るために、次の実験を行なった。筆者らが所属した研究室内では World-Wide Web (WWW) 上で情報サービスを行っており、当時新たに、複数ユーザが利用可能なホームページ作成サービスプログラムを開発することになった。この事例の特徴は、グループによるシステム管理がなされていたこと、WWW という発展途上の技術を利用するためにシステム仕様に度々変更がなされたこと、ネットワークにオープンなシステムのためシステム動作の予測や全体像の把握が難しかったことである。

我々はまず、システム設計者の要求仕様ドキュメントを元に、要求モデルを作成した。この設計者は熟練したプログラマだったので、要求概念そのものは既に明らかであり、直接要求モデルを構築することは困難ではなかった。

要求モデルを作成した結果、元々の仕様ドキュメントと要求モデルを併用することにより、システム開発に関わるグループの間での要求仕様の相互理解が深まった。例えば、我々は要求モデル内の part-of リンクと is-a リンクを含む部分だけを切り出すことでソフトウェアのデータ構造を可視化したデータモデルを再構成したり、要求モデルの関連オブジェクト周辺の情報のみを切り出すことでシステムの機能モデルを再構成する汎用プログラムを作った。これらを援用することで、モデルの詳細について議論することが容易になり、ドキュメントのときには気付かなかったシステム仕様の不具合に気付くという効果があった。

気付いた不具合に対する対処としての要求仕様の修正も、KAUS をモデル操作インタフェースとして用いて行なわれた。そうすることで、例えば、システム内で不要になった機能の削除を行なった際に、それに伴って不要となるデータ構造があったのを紙の仕様書を見ていたときは見落としていたが、コンピュータ上で KAUS を介してモデル操作したときに気付かされ、体系的に削除処理を行なうことができた。これらは、KAUS をインタフェースとして体系的に要求モデルを操作・管理していることで実現された効果であると考ええる。

単独のプロジェクトの場合だけを考えると、要求空間や要求モデルを作るのは確かにわずらわしい作業である。しかし、要求仕様作成中の人的な誤りを削減し、個人では管理不可能な大規模システムの構築とグループによる管理を可能にするためには、システム開発の上流行程におけるコンピュータ導入は必然である。コンピュータ導入を進めることで、今まで見捨てられてきた上流過程での作業履歴やノウハウがコンピュータ

内に蓄積されることになり、それは、類似システムの開発やシステムの修正を含めた大きなサイクルを考えると利益が大きいと考える。

6. む す び

本論文では、発想支援技術と知識処理技術の統合による、システム設計における要求モデル構築支援について述べた。提案システムは、ソフトウェア開発や研究活動において副産物として生ずる非形式なドキュメントをシステムの設計や管理に活用することを意図している。これらのドキュメントは、これまでは体系的に再利用されてきたとは言えないが、個人やグループにおける創造的な生産性の向上を考える際には、体系的な活用手段が強く求められ、本研究はその一アプローチを示すことができたと考ええる。

本論文で提案した方法論は、ソフトウェア開発における人間とコンピュータの間のインタフェースを人間側に引き寄せることを意図したものである。ユーザの要求がコンピュータ上で操作可能なモデルとして一旦構造化されれば、それを実行可能なプログラムに半自動的に変換するための研究は、これまでも盛んに行なわれている（例えば、[Li 93]）。したがって、我々が提案したシステムとこれらを結合することで、エンドユーザが自分で要求空間を構築するとソフトウェアのプロトタイプが自動生成されることが可能になると期待される。その実現のためには、プロトタイプされた要求モデルの評価と修正を完全に自動化するか、もしくは、エンドユーザによる要求モデルの解釈をさらに容易にする手段の提供を検討することが重要であり、今後の課題と考えている。

本論文で提案したアプローチは、類推や事例ベース推論のような、経験に基づいた知識処理パラダイムに属する。これらの手法は一般的に、既知の問題と関連づけることで未知の問題を認識し、明らかでなかった部分を明示化するものである。我々が提案したシステムでもそのような効果が観察される。すなわち、ユーザは抽象的な要求空間からより具体的な表現である要求モデルを事例から抽出することができ、そのことで自分の要求をより明示的に認識することができる。

我々は、経験に基づく知識を有効活用するには、問題解決の結果だけでなくそのプロセスで得られた知識を陽に扱うことが重要であると考ええる。本論文で提案した方法論が、要求モデリングの最終成果物である要求モデルのみでなく、記号化が進む前の要求やアイデアの概念構造を残している要求空間を知識として重用

するのは、そのような理由からである。

謝 辞

本研究は主に1995年までに東京大学においてなされたものである。CSSの開発には小川竜太氏と杉本雅則氏、KAUSの利用には山内平行氏と矢野新一郎氏、ソフトウェア要求仕様の事例収集には芹澤正夫氏と内田裕之氏に協力頂いた。ここに感謝の意を表す。

◇ 参 考 文 献 ◇

- [Arrango 91] Arrango, G. and Prieto-Diaz, R.: Introduction and overview: Domain analysis concepts and research directions, In Prieto-Diaz, R. and Arrango, G. eds., *Domain Analysis and Software Systems Modeling*, pp. 9-26, IEEE Computer Society Press (1991).
- [Borgida 85] Borgida, A., Greenspan, S. and Mylopoulos, J.: Knowledge representation as the basis for requirements specifications, *IEEE Computer*, Vol. 18, No. 4, pp. 82-90 (1985).
- [Hori 92] Hori, K. and Ohsuga, S.: Computer aided thinking for software development, *Proc. 2nd Pacific Rim Int'l Conf. on Artificial Intelligence*, pp. 203-208 (1992).
- [堀 97] 堀浩一: システム統合のためのAIへむけて—発想支援系と知識処理系の結合の提案—, *人工知能学会誌*, Vol. 12, No. 2, pp. 258-265 (1997).
- [Li 93] Li, C. and Ohsuga, S.: A meta knowledge structure for program development support, *Proc. 5th Int'l. Conf. on Software Engineering and Knowledge Engineering* (1993).
- [Lowry 91] Lowry, M. R. and McCartney, R. D. eds.: *Automating Software Design*, AAAI Press/MIT Press (1991).
- [Ohsuga 90] Ohsuga, S.: Framework of knowledge-based systems - Multiple meta-level architecture for representing problems and problem-solving processes, *Knowledge-Based Systems*, Vol. 3, No. 4, pp. 204-214 (1990).
- [Rich 86] Rich, C. and Waters, R. C. eds.: *Readings in Artificial Intelligence and Software Engineering*, Morgan Kaufmann (1986).
- [角 94] 角康之, 堀浩一, 大須賀節雄: 要求空間を可視化することによるソフトウェアの要求モデル構築支援, 竹内彰一(編),

インタラクティブシステムとソフトウェアII(ソフトウェア科学会 WISS'94), pp. 233-242, 近代科学社(1994).

[角 95] 角康之: ソフトウェアの要求モデル構築支援に関する研究—要求獲得における発想支援と要求モデルの構造化支援—, 博士論文, 東京大学大学院 工学系研究科 情報工学専攻 (1995).

[角 96] 角康之, 小川竜太, 堀浩一, 大須賀節雄, 間瀬健二: 思考空間の可視化によるコミュニケーション支援手法, *電子情報通信学会論文誌*, Vol. J79-A, No. 2, pp. 251-260 (1996).

[田村 94] 田村恭久, 伊藤潔, 杵嶋修三: ドメイン分析・モデリング技術の現状と課題, *情報処理*, Vol. 35, No. 10, pp. 952-961 (1994).

[内平 92] 内平直志(編): 特集「仕様獲得と知識獲得—ソフトウェアシステムの視点から—」, *情報処理*, Vol. 33, No. 6, pp. 604-640 (1992).

[White 91] White, D. A.: The knowledge-based software assistant: A program summary, *Proc. 6th Annual Knowledge-Based Software Engineering Conference*, pp. 2-6, IEEE Computer Society Press (1991).

(担当委員: 國藤 進)

著 者 紹 介

角 康之(正会員)は, 前掲(Vol.14, No.1, p.40)参照。



堀 浩一(正会員)

1979年東京大学工学部電子工学科卒業。1984年同大学院博士課程修了。同年, 国文学研究資料館助手, 1986年同助教。1988年東京大学助教, 1997年同教授, 現在に至る。工学博士。この間, 1989年9月~1990年1月仏国コンピユエニユ大学客員助教。現在, 先端学際工学専攻所属。航空宇宙工学専攻と情報工学専攻を兼担。人工知能を中心とした情報処理システムの基礎から応用にわたる広範囲の研究・教育に従事。最近の個人的な興味を中心は創造活動支援システム。電子情報通信学会, 情報処理学会, 日本ソフトウェア科学会, 日本認知科学会, IEEE, ACM各会員。 <hori@ai.rcast.u-tokyo.ac.jp>



大須賀 節雄(名誉会員)

1957年東京大学工学部航空工学科卒業。同年, 富士精密工業(株)入社。1961年東京大学航空研究所助手, 1966年工学博士。1967年東京大学宇宙航空研究所助教, 1981年東京大学工学部教授。1995年早稲田大学理工学部教授, 現在に至る。人工知能, 特に知識処理, ヒューマンインタフェース, CAD等の研究に従事。情報処理学会, 電子情報通信学会, 航空宇宙学会, 日本ソフトウェア科学会, AAAI各会員。 <ohsuga@ohsuga.info.waseda.ac.jp>