

要求空間と機能空間の可視化によるソフトウェア設計の試み

角 康之 小川 竜太 堀 浩一 大須賀 節雄

東京大学 先端科学技術研究センター

〒153 東京都目黒区駒場4-6-1 Phone: 03-3481-4486

Email: sumi@ai.rcast.u-tokyo.ac.jp

ソフトウェア開発における要求獲得を支援し、得られた要求と過去に開発されたソフトウェアの機能の関連性を可視化する方法について述べる。ユーザの要求を表現するものとして、それを構成するアイデアや概念に相当するテキストを、そして過去の開発事例を表現するものとして、それを構成する機能について書かれたテキストをオブジェクトとし、それらの構造を距離空間に可視化する手法をとった。こうすることによって、ユーザ自身の要求空間や、それに関連する機能空間の構造を認識することが容易になり、ソフトウェア開発における創造的な思考を支援することが可能になった。

Software Designing with Visualization of Requirement Space and Function Space

Yasuyuki SUMI Ryuta OGAWA Koichi HORI Setsuo OHSUGA

Research Center for Advanced Science and Technology, The University of Tokyo

4-6-1 Komaba, Meguro-ku, Tokyo 153, Japan Phone: +81-3-3481-4486

Email: sumi@ai.rcast.u-tokyo.ac.jp

The paper will discuss a method to support requirements acquisition in software development by visualizing relationship between the requirements acquired and functions in existing software. In order to visualize structures of requirement space and function space, we put text-objects corresponding to idea or concept for user's problem and function of existing software into metric spaces, respectively. This makes a user possible to understand his/her requirement space and function spaces related to it, and support his/her creative thinking in software development.

1 はじめに

本稿では、ソフトウェア開発における要求獲得を支援し、得られた要求と過去に開発されたソフトウェアの機能の関連性を可視化する方法について述べる。

ソフトウェア開発とは、問題を持っているユーザの心の中の世界と、それを解決するために利用されるコンピュータの世界という二つの異なった世界を仲介 (mediation) する作業であると、考えられる [1]。ユーザは、ソフトウェアに対する要求について、問題が存在する対象世界を構成する言葉を用いて思考する。その一方で、コンピュータの世界は、機能を表示する言葉で語られる。そこで、ユーザの要求を表示するものとして、それを構成するアイデアや概念に相当するテキストを、そして過去の開発事例を表示するものとしては、それを構成する機能について書かれたテキストをオブジェクトとし、それらの構造を距離空間に可視化する手法をとった。こうすることによって、ユーザ自身の要求の構造や、それに関連する事例の構造を認識することが容易になり、ソフトウェア開発における創造的な思考を支援することが可能になった。また、要求空間と事例空間を擦り合わせるにより、要求から機能への橋渡しを支援することが可能になった。

以下では、まず2章でソフトウェア開発における上流工程の現状と、本研究の目的を述べる。次に3章で、思考支援システム CAT1 を利用して、ユーザの要求空間とそれに関連する事例空間を可視化し、要求獲得を支援する方法を述べる。4章では、要求空間や事例空間において使われる言葉の共起関係を可視化する方法を導入し、問題対象世界の知識や、ソフトウェア内の機能の共有と再利用について考える。最後に5章で、要求空間と事例空間を擦り合わせることによって、要求-機能間の関連性を可視化し、創発的なソフトウェア設計を支援する方法について述べる。

2 ソフトウェア開発における上流工程の現状と、本研究の目的

現在、ほとんどのビジネスソフトウェアの開発に採用されているウォーターフォールモデルでは、要求定義工程の成果物の善し悪しがそれ以降の工程に大きく影響するため、その重要性和ともにその難しさが広く認識されている [2]。要求定義工程の困

難さの原因としては、

1. ユーザ自身が、自分の持つ問題や要求を認識していないこと
2. 顧客 (ユーザ) と SE (System Engineer) の背景知識が異なることによるコミュニケーションギャップ
3. 要求仕様から設計仕様への変換の難しさ

が考えられる。本研究の目的は、これらの問題を克服するための上流 CASE (Computer Aided Software Engineering) ツールを構築することである。

3. は要求定義工程と言うよりもむしろ設計工程における問題であると思われるが、この問題の克服はソフトウェア開発における最も重要な課題であり、そもそも要求定義工程と設計工程の間にはっきりとした境界を引いてしまうことが無意味であると考え、1. および2. の問題と切り離さずに、本研究の課題として位置付けた。ソフトウェアの設計とは、「まず要求定義がなされ、その後に設計仕様への翻訳がなされる」というような単なるひと流れの作業ではなく、心の世界とコンピュータの世界のそれぞれで自己組織化された構造同士を互いに擦り合わせるような作業であると考えている。

ウォーターフォールモデルでは、要求定義工程では要求仕様書、設計工程では設計仕様書、といったように各工程ごとに成果物を生成する。しかし、各工程間での連絡がこれらの成果物のみでしかなされないのだとしたら、それは各工程間に余計な解釈の壁を設けたことになり、上流工程の意図と下流工程における成果物との行き違いを引き起こしかねない。したがって我々は、あえて要求定義工程と設計工程を切り離さずに、ユーザの要求空間と、コンピュータで実際にできることを表現している事例空間の擦り合わせを行い、要求の分析が完了したらそれと同時に、その副産物として設計モデル (情報モデル、状態モデル、プロセスモデル...) をも得られるようなパラダイムを構築すべきであると考えている。これは、近年盛んに行なわれているプロトタイプングの研究の動機と同じものである [3]。

本研究では、上述のソフトウェア開発上流工程における問題を解決するために、思考空間を可視化することによってユーザの思考活動を支援することを試みる。ここでは、要求仕様やソフトウェアといった成果物を、知識やアイデアといった情報の断

片の集合が構造を織りなしたものととらえ、それらの構造を可視化して思考活動やコミュニケーションの媒体とする方針をとる。また、その際、過去の成果物を事例として積極的に利用することを考える。

以下に、各問題を解決するための本研究の方針を示しておく。

要求獲得支援 ユーザ(顧客)が自分の問題や要求を認識するのを助けるために、対象世界の分節作業を支援する。

対象世界を表現する言葉の分析の支援 ユーザの要求の世界を記述するのに用いられる概念素と、コンピュータの世界を記述するのに用いられる機能のマイクロフィーチャーの分析を支援する。

創発的なソフトウェア設計の支援 ユーザが構築した要求空間と、過去に開発されたソフトウェアの事例空間を擦り合わせることによって、ユーザの要求を満たす新たな機能空間の創発を支援する。

本稿では、以下の章でこれらについて順に述べ、現在までに行ってきたことを報告する。

3 要求空間と事例空間の可視化による要求獲得支援

3.1 要求獲得工程におけるユーザの発想支援

これまででも多くのCASEに関する研究が行われてきたが、これらの多くは、少なくとも要求者であるユーザの心のなかでは開発対象となるソフトウェアへの要求が明確な形で存在しているということの大前提としてきた。しかし実際は、ユーザの心の中でさえ要求はあいまいであることが多く、ときとしてソフトウェア開発の進行中にも要求の形式や内容が変化してしまうことがある。

ソフトウェア開発に限らず、設計型問題解決を支援するということは人間の創造を支援することに他ならない。我々は、「創造とは、心の世界を通して実世界内に新しい分節を見つけるプロセスを前提として持つ」と考えている。ここで分節(articulation)とは、対象世界から構成要素を切り出し、それらが創り出す様々な構造の切口を見つけ出す作業、またはそれによって見つかった新しい視点のこ

とを言う。上述の問題を解消するには、ユーザの問題が存在する対象世界を分節する作業を支援する必要がある。

「自分の問題も把握していないうちから、ソフトウェアを作ろうと考えることはあり得ない」とか、「ソフトウェア開発に、新しい発想なんて必要ない」という意見もあるかもしれない。組織や業務が静的に確立した世界における問題を解決しようとする際には、確かに開発対象となるソフトウェアへの要求は比較的簡単に獲得・分析できるであろう。しかし、組織や業務自体が新規なものであったり、研究活動におけるソフトウェア開発においては、ユーザの問題が存在する対象世界がまだ分節化されておらず、動的な構造化を必要とすることが多い。また、自分の問題をコンピュータを利用して解決しようと考えるとき、対象世界を構成する言葉のみで問題を認識するだけでは十分ではなく、実際にそれがコンピュータの世界ではどのような構造を持つのかを即座に確認しながら要求や問題を認識する必要がある。そのような理由から、我々は要求空間と機能空間を可視化し、要求獲得工程におけるユーザの発想を支援する方法を提案する。

3.2 思考支援システムによる要求獲得支援

筆者らは以前から、専門家からの知識獲得、文章生成、文献検索などにおける分節作業を支援するためのツールとして、分節支援システムAA1(Articulation Assistant, version 1)[4]と思考支援システムCAT1(Computer Aided Thinking, version 1)[5, 6]の開発を試みてきた。これらは、ユーザが思考対象としている世界の様々な分節を可視化するために距離空間を利用している。つまり、ユーザの心の中の思考空間を間接的にコンピュータディスプレイ上に投影表示することを試みたわけである。

ここでは、今回ユーザの要求獲得を支援するために利用したCAT1について説明する。図1は、CAT1を利用しながら、ユーザが自分自身の問題を認識し、それをソフトウェアへの要求空間として構造化していく様子を示したものである。

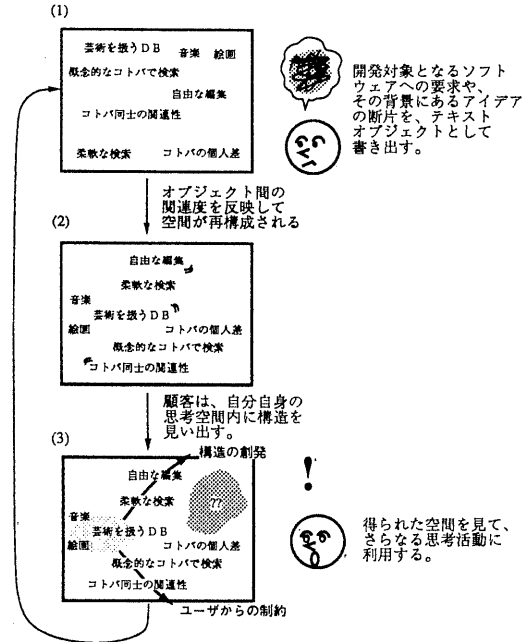


図1 ユーザがCAT1を利用しながら、自分自身の要求を認識、構造化していく過程。

ユーザは、開発対象となるソフトウェアに関するアイデアや知識、要求の断片として、テキストオブジェクトを作成し、それらをディスプレイ上の2次元空間へ配置する。テキストオブジェクトとは、ディスプレイ上の仮想的なカードのようなもので、各テキストオブジェクトには自由に複数のキーワードを宣言できる。CAT1は、これらのキーワードの共有度をオブジェクト同士の関連度とみなし、関連度が高いオブジェクト同士ほど距離が近くなるように空間を自動的に再構成する。この際、多次元尺度法と呼ばれる統計手法を利用している。ユーザは再構成された空間配置のなかに新しい構造を見出し、それをさらなる思考活動に利用するのである。この作業を繰り返すことによって、徐々にユーザの心の中にあるアイデアや知識が明らかなるものとしてユーザ自身に認知されていくのである。

図2は、筆者の一人が「芸術を扱う感性データベース」の開発を想定し、実際にCAT1を利用して要求空間を構築してみた様子である。

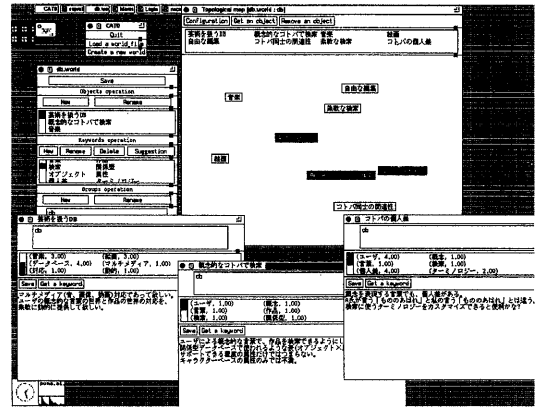


図2 CAT1の利用例。中央上部にあるウィンドウがユーザの思考空間を投影した距離空間に対応し、テキストオブジェクトがアイコンとして空間配置されている。それらをクリックすると、文章本体とキーワードのリストを持った仮想的なカードが開かれ、それらはいつでも自由に変更できる。それらの変更は動的にオブジェクト空間の空間配置に反映される。

3.3 事例空間の可視化

要求定義を行なう際に、ただ図1のような作業を続けているだけでは、作業にいきづまってしまうことが度々ある。そのようなとき、人間は例を見て作業を進めたり、類推を働かせたりする。これまでソフトウェア開発の現場においては、SEが過去の事例を顧客に見せて、それらとの類似点や相違点を議論することにより、顧客の要求を分析することが行われてきた。事例とは、実際に何らかの要求を獲得・分析し、さらにそれをコンピュータの世界で実現したものであるから、その成果物そのものだけでなく、その開発プロセスも、新たなソフトウェア開発に利用されるべきである。

事例の再利用と言うと、ソースコードそのものや、それを表現するデータフローダイアグラムなどを再利用すると思われるがちであるが、それらはユーザの要求を表現する言葉の世界とはあまりにもギャップが大きい。そこで、ここでは、開発されたソフトウェアが顧客先に納入される際に添付されるユーザ向けのドキュメントを、再利用される知識の媒体として利用する。また、研究開発の現場の場合には、ソフトウェアを開発した研究者自身がそのソフトウェア自身について書いたメモや、それを開発するにあたって書いた研究メモを利用する。

事例を参考にするとき、ソフトウェアを構成する機能モジュールやそれに関連するアイデアを表す文章の単なる集合を見せられても、それは類推の対象とはなりづらい。そこで、事例に関するドキュメントの断片をテキストオブジェクトとし、CAT1で各事例ごとの構造を可視化することを試みた。

つまり我々は、過去の事例を知識ベース化し、それを図3に示すような形でユーザの要求定義工程における発想の支援に利用することを考えている。

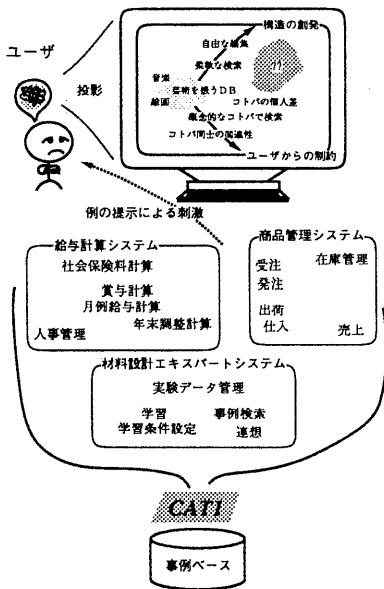


図3 過去の事例を知識ベース化し、それらの構造をCAT1で可視化することによって、ユーザからの要求獲得の刺激とする。

現在は、要求に関連する事例空間の検索をユーザ自身もしくはSEが行う必要があるが、照合プロセスをエキスパートシステム化することが将来の課題である。

4 対象世界における言葉の分析

4.1 言葉同士の共起関係の可視化

我々は、AA1やCAT1の派生システムとして、CSS(Communication Support System)と呼ぶシステムを新たに開発し、実験に利用している。これは名前が示す通り、本来は、人間同士のコミュニケーションを支援することを意図して開発された

システムである。今回は、これをソフトウェア設計における、文脈の可視化に利用する。

背景知識が異なる人間は、おのずと言葉の使い方も異なってくる。正確に言うと、ある分野に精通した人にとっては思考活動に不可欠である言葉が、他人にとってはまったく知らない言葉であることがある。また、表層的な表現としては同じである言葉を共有する場合でも、分野が異なると、その言葉が意味する概念や、他の言葉との関連性が異なることが度々ある。

ここで、個人個人がおかれている状況や、属している専門分野などをまとめて、文脈と呼ぶことにする。コミュニケーションに必要な言葉や、それらの関わり合いは、文脈に依存する。ソフトウェア開発においても、それぞれが異なる背景知識を持つために生じる、顧客とSEのコミュニケーションギャップが深刻な問題になっている。また、言葉の世界だけに限らず、例えば、ソフトウェアの部品化が進んだ場合を想定した場合、それらの部品を組み合わせることによって生じる機能(振舞い)は、組み合わせの数だけ存在する。このことは、部品の粒度、もしくは抽象度の高低に関わらず、どのようなレベルにおいても存在する問題であると考えられる。したがって、ソフトウェア開発のどの工程においても、複数の言葉や部品が組み合わせられて、ある概念や機能が生じるわけであり、開発に携わる者はその構造を理解し、さらなる思考に利用する必要がある。そこで、我々は特に言葉のレベルに着目し、対象世界の文脈を可視化することを試みた。

対象世界の文脈を定量化するには、言葉の共起関係を利用するのが便利であると考えられる。CSSは、双対尺度法と呼ばれる統計手法を用いて、複数のテキストオブジェクトと、それらを構成するマイクロフィーチャー(構成要素または属性)としてのキーワードの、所属・共有関係を2次元平面上に可視化する。図4は、CSSが出力した「芸術を扱う感性データベース」に関する要求空間である。

CSSが出力する空間では、あるテキストオブジェクトで宣言されているキーワードは、そのテキストオブジェクトの近くに配置される。したがって、複数のテキストオブジェクトで宣言されているキーワードはそれらのテキストオブジェクトの間に配置されることになる。逆に、テキストオブジェクトは、それを構成する複数のキーワードの間に配置される。双対尺度法は、これらの局所的な制約を満

たすために、空間全体の歪みがなるべく小さくなるように、空間を構成する。したがって、例えば「階層的、体系化、類似性、文脈」のように、共起関係(テキストオブジェクトによる被宣言状況)が似通っているキーワード群はクラスタを作り、それらは「コトバ同士の関連性」と名付けられたテキストオブジェクトに宣言されていることがわかる。また、例えば「柔軟な検索」と「自由な編集」のように、近くに配置されたテキストオブジェクト同士はキーワードの共有度が大きいことを表している。

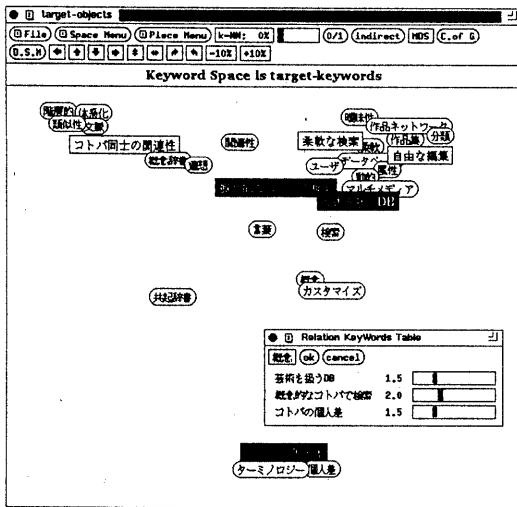


図4 CSSが出力した「芸術を扱う感性データベース」の要求空間。長方形の枠で囲まれたアイコンがテキストオブジェクトを表し、楕円の枠で囲まれたものがキーワードを表す。画面上では、「概念」と名付けられたキーワードが宣言されているテキストオブジェクトが、反転表示されている。

以上で説明したように、CSSは統計処理によって、言葉の共起関係を可視化したものに過ぎないが、我々はCSSによって得られる空間構造全体が、一つの文脈を表現していると考えられる。また、これは言葉同士の連想関係を可視化するものであると考える。

4.2 知識の共有と再利用に向けて — 概念素の整備 —

知識ベース技術を有効にするための一つのアプローチとして、専門知識を表現するための概念素

(語彙)を体系化することの必要性がさげられている[7]。概念素には、その知識が属する分野の文脈や、そのときの利用目的などが反映されている。したがって、知識の共有や再利用を実現するには、知識を表現するための概念素を整備すると同時に、それらの組み合わせによって創り出される文脈を表現し得る手段が必要なのである。

本研究がとったアプローチは、ソフトウェア開発が対象とする様々なドメインを表現するための概念素を、整備するための一手段としても役に立つと思われる。図5に、過去に開発されたソフトウェアの事例として、「青果物卸売市場のための販売管理システム」について書かれたドキュメントを利用して構成された事例空間を示す。

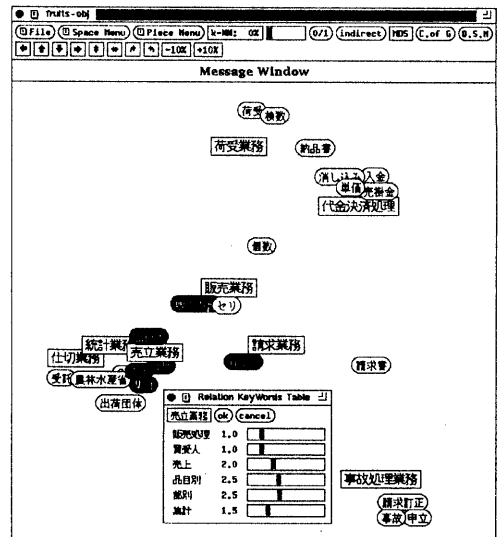


図5 CSSが出力した「青果物卸売市場のための販売管理システム」の事例空間。画面上では、テキストオブジェクト「売上業務」が宣言しているキーワードが、反転表示されている。

図5を見ると、販売業務一般のドメインに特有の言葉がキーワードとして顕現化されることがわかる。これが、対象世界を表現するための概念素となり得ると考えている。具体的に見てみると、大きく分けて、{受託、集計、荷受、販売処理、...}といったタスク(業務)を表現する概念素と、{納品書、請求書、農林水産省、買受人、...}といったドメインに依存した概念素に分類されることに気付く。このようなツールを利用して、対象世界の言葉

を分析することは、ソフトウェア開発におけるドメイン分析のための、一つのアプローチとなり得ると考えている。

また、このようなツールを利用することによって、文脈を伴った機能検索が可能になると期待される。現在でも、全文曖昧検索システムを利用し、ドキュメントを媒介として、ソフトウェア部品を検索・再利用する試みがなされている。しかし、要求を満たす機能を検索するとき、それらが他の要素とは独立に、一対一に対応することは稀であり、実際は、互いに複数の要素の組み合わせがそれぞれの構造を持ちながら、対応し合う。そのような理由から、次章では、要求空間と事例空間(すなわち機能空間)の擦り合わせを試みる。

5 要求空間と事例空間の擦り合わせによる、要求-機能間の構造の創発

ここで、要求と機能の擦り合わせを実現するために、図6に示すような方法で、要求空間と事例空間を擦り合わせる。要求空間は、CAT1やCSSを利用してユーザが構築したものである。事例空間は、図3のとときと同様、事例ベースに蓄えられた事例である。そして、対象世界の概念素辞書が成熟していると仮定し、互いの空間で利用されているキーワードを架け橋として、互いの空間にただよっているオブジェクト集合を一つの空間内に混在させるのである。こうすることによって、ユーザの要求と、過去の事例において実際に実現された機能の関連性が、視覚的に確認できるようになる。

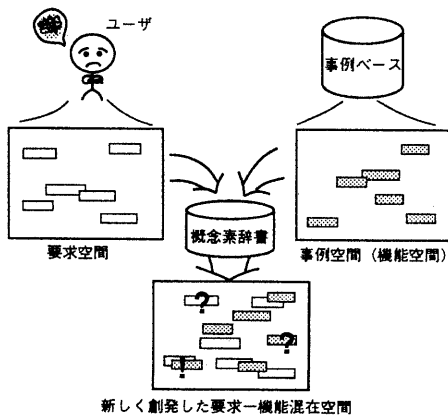


図6 要求空間と事例空間を擦り合わせることによって、要求と機能が混在した新しい空間が再構成される。

今回、実験する際には、要求空間内のオブジェクトを構成しているキーワード集合と、自分のキーワード集合が多く重なっているような事例空間をあらかじめ与え、単純にそれらのキーワードを媒介として、要求空間内のオブジェクト集合と事例空間内のオブジェクト集合を、一つの空間に空間配置してみた。今回、事例としては、CAT1とその派生システム二つについての機能に関するドキュメントを、混在させて利用した。図7に、CSSが出力した事例空間を示す。

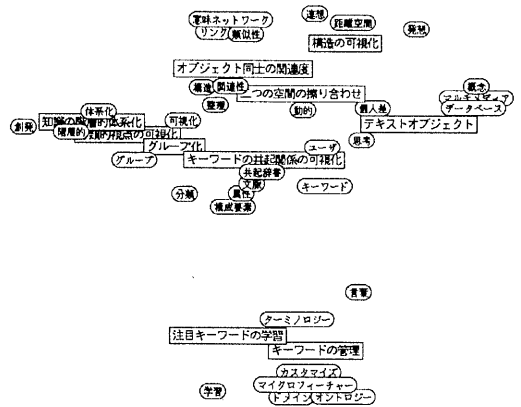


図7 既に開発されている三つの思考支援システムに関する機能空間。

図7を図4と比較しながら見ると、要求空間と事例空間において、{言葉、概念、個人差、関連性、...}といったキーワードを共有していることがわかる。しかし、それらのキーワードと他のキーワードの間に生じている構造は、それぞれの空間においてかなり異なっている。要求空間(図4)においては、まだ具体的な機能としてこれらのキーワードが分析されておらず、他の抽象的な言葉と共に使われているに過ぎない。それに対して、事例空間(図7)では、これらのキーワードは、コンピュータの世界における具体的な機能を表すキーワードと共起している。

そこで、上に示したような、要求空間と事例空間において共有しているキーワードを橋渡しにして、要求空間の中にただよっている要求と、事例空間の中に存在する機能の関連性を顕現化してみる。図4と図7中のオブジェクトを一つの空間に投げ込み、再構成された空間構造を、図8に示す。

