

# プロダクトラインの要求仕様を統合する要求分析モデルの提案

ジリエ陽子<sup>†</sup> 本田耕三<sup>†</sup> 中川博之<sup>†</sup> 田原康之<sup>†</sup> 大須賀昭彦<sup>†</sup>

本論文では、プロダクトラインの要求仕様を統合する要求分析モデルを提案する。テレビやカーナビゲーションなど様々なバリエーションのある製品は、多機能で類似の仕様のソフトウェアを短期間で並行して開発する。このような開発ではプロダクトライン開発が有効であり、どのような製品系列とするのか、どのようにソフトウェアを資産化して再利用するのが重要である。しかし、実際の開発現場においては、そもそも要求仕様が煩雑化して把握しにくく、ソフトウェアの資産化や再利用が困難でプロダクトライン開発は適用しにくい状況にある。本研究では特に、ゴール指向要求分析手法 KAOS をベースに、フィーチャモデルの機能もサポートできるようにプロダクトラインの要求仕様を統合し、情報可視化手法を用いて、複数製品の要求仕様の可視性を向上させる。本研究ではカーナビゲーションシステムの開発を想定した評価実験も実施し、提案手法を用いることにより、仕様変更時の影響範囲の確認や要求分析モデルのメンテナンスしやすさについて要求仕様の煩雑化が改善されていることを確認した。

## Requirements Analysis Models to Integrate Requirements Specifications for Software Product Lines

YOKO GIRIER<sup>†</sup> KOZO HONDA<sup>†</sup> HIROYUKI NAKAGAWA<sup>†</sup>  
YASUYUKI TAHARA<sup>†</sup> AKIHIKO OHSUGA<sup>†</sup>

This paper proposes requirements analysis models to integrate requirements specifications for software product lines (SPLs). Products with different variations like television, car navigation system, etc. are developed in parallel in a short time as software with a lot of functions and similar specifications. SPLs are effective for such development. It is important for SPLs to design series of products, make the software an asset and reuse. However, in the first place requirements specifications are very complicated and difficult to understand on the field. It is hard to make software an asset and reuse, therefore SPLs aren't easy to apply. In this research, particularly based on goal-oriented RE methodology KAOS, we integrate requirements specifications for SPLs, use information visualization methods and improve visibility of requirements specifications for multiple products with supporting functions of feature models. Also in this research we did experiment assuming development of car navigation system. We verified complexity of requirements specifications is improved in terms of the impact of requirements specifications change and maintenance of requirements analysis models.

### 1. はじめに

機能に様々なバリエーションがある携帯、シリーズの製品ラインナップがあるテレビ、顧客ごとに若干異なる仕様のカーナビゲーションシステム（以降、カーナビと記述）といった組み込み機器、データ管理業務系アプリケーションや発電所向け監視制御システムなど、実に様々な製品において類似の仕様でバリエーションのある多機能なソフトウェアが開発されている。

図 1 はカーナビの開発体制について表示したものである。カーナビは大まかに地図表示、探索、誘導、検索機能を備えている。A社、B社、C社、…、はカーナビの顧客、すなわち自動車会社である。顧客ごとに配置された機種担当チームが顧客とやり取りし、仕様をまとめて開発チームとの認識擦り合わせを行う。開発チームは地図表示、探索、誘導というように機能ごとに配置され、例えば地図表示開発チームはA社、B社、C社、…、すべての顧客向けの地図表示機能開発を担当する。短い開発期間で複数の顧客向けに多機能なソフトウェアを並行して開発し、頻繁な要求の変更を課されることも多い。このように類似の仕様で

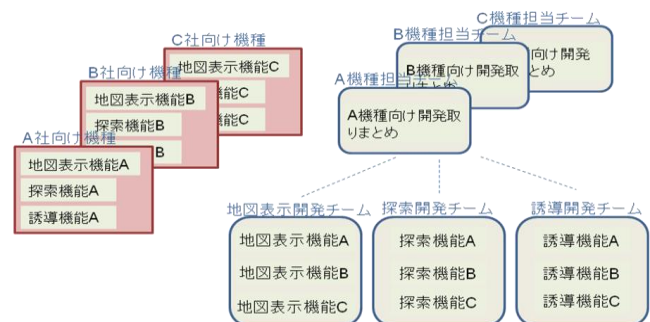


図 1 カーナビの開発体制

Figure 1 Developmental regime of car navigation system

複数の開発プロジェクトを一連のプロダクト生産としてみる手法として、プロダクトライン開発が有効である。プロダクトライン開発では、一般にフィーチャモデリングによりソフトウェアの特性を分析する。どのような製品系列とするのか、どのようにソフトウェアを資産化して再利用するのがポイントとなってくる。しかし、実際の開発現場においては、そもそも要求仕様が煩雑化して把握しにくい状況にある。例えば地図表示機能は、紙の地図帳を開くようなシンプルなものではない。地図の中に自車の位置を表示し、自車の動きと連動して地図も動く。地図スケール

<sup>†</sup>電気通信大学  
The University of Electro-Communications

変更, 地図向き変更, 地図画面スクロール, 道路だけでなく 3D ビル表示, といったように実に様々な機能を持っている (図 11, 図 12 参照). 実際, カーナビ製品の取扱説明書は 300 ページを超えるものもある [7][8]. また, 短期間で多機能なソフトウェアを開発しなければならないため, 多数の開発メンバによる業務分担もさらに要求仕様全体を把握しにくくさせる状況にある. 仕様に対する理解が顧客と開発者で一致していないこともある. その結果, 顧客の想定しているものと別のソフトウェアを開発したり, 似たような機能がすでにあるのに気付かず重複して開発したり, 類似の仕様の中で再利用を想定したソフトウェアの設計ができず, 複数の開発プロジェクトを一連のプロダクト生産として考えるプロダクトライン開発は適用しにくい状況にある.

本研究では, カーナビ等, 多機種・多機能な製品を対象としたプロダクトライン開発において, 要求仕様の煩雑化改善に着目する. ゴール指向要求分析手法 KAOS を適用し, 複数製品の要求仕様を統合して管理する. KAOS 要求分析モデルをベースにフィーチャモデルの機能もサポートできる要求分析手法を提案する. 要求仕様を共通部分と各顧客固有の部分に分けつつ統合して管理し, 要求仕様変更に対しても要求分析モデルとしてメンテナンスし再利用できるようにする. さらに, 情報可視化手法を用いて, 複数製品の要求仕様が同時に確認できるようにする.

本論文の構成は以下の通りである. 第 2 章では, フィーチャモデルと KAOS 要求分析モデルについて概説する. 第 3 章では製品開発の課題と要件について述べる. 第 4 章では, 本研究提案手法の詳細について述べる. 第 5 章では, カーナビ開発を想定し, 本研究提案手法の評価実験と考察を行う. 第 6 章では従来研究について述べる. 第 7 章では, 結論と今後の課題について述べる.

## 2. フィーチャモデルと KAOS 要求分析モデル

フィーチャモデルと KAOS 要求分析モデルについて概説する. フィーチャモデルは製品の特徴のモデリング, KAOS 要求分析モデルは顧客要求分析のモデリングにそれぞれ適したモデリング手法である.

### 2.1 フィーチャモデル

フィーチャモデルはプロダクトライン開発における製品の特徴, すなわちフィーチャを定義し, 共通性と可変性を表現する. 図 2 はフィーチャのモデリング表記法である [1]. 必須フィーチャは複数の製品で共通性が高く, オプションフィーチャは製品ごとに含むか含まないか選択するものである. 包含的論理和フィーチャは, フィーチャのいずれかもしくはすべてを含むケースを表現している. 例えば次図 2 の支払方法(PaymentTypes)にはデビットカード(DebitCard), 注文書(PurchaseOrder), クレジットカード(CreditCard), があるが, デビットカードのみ, クレジット

カードのみ, もしくは注文書とクレジットカードの両方といった場合の記述に用いる. 排他的論理和フィーチャはフィーチャのいずれかを選択するとき残りは選択しないケースを表現している.

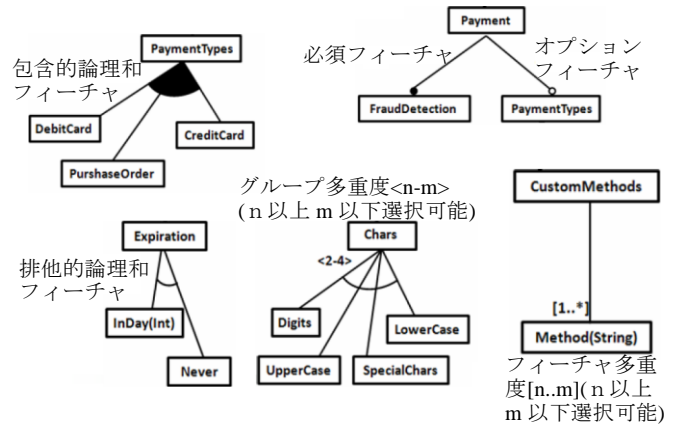


図 2 フィーチャモデル表記法 [1]

Figure 2 Feature model concepts.

### 2.2 KAOS 要求分析モデル

ゴール指向要求分析はニーズをシステムが達成すべき目標, すなわちゴールととらえ, なぜ・どうやって達成するのか, という観点で分析する要求工学の重要なアプローチである. KAOS はゴール指向要求分析の代表的な分析手法のひとつである. システムゴールを 1)目標状態, 2)システムが達成責任を持つ操作要求と, 3)環境や人が達成責任をもつ仮説の 3 種類に分類し, 系統的に分析する. KAOS の要求分析モデルには, ゴールモデル, オブジェクトモデル, 責任モデル, 操作モデルがあり, ゴールモデルを中心とした要求分析を行う. 本論文では, データ部分をオブジェクトとして表現するためオブジェクトモデルを使用する (4.3 参照). ゴールモデルは, 目標とする状態を系統的に分析したモデルである. 機能的ゴール, 非機能的ゴールを AND/OR リンクと障害リンクにより結合し構造的に表現する. 次図 3 は, それぞれ左から AND リンク, OR リンク, 障害リンクの例である.

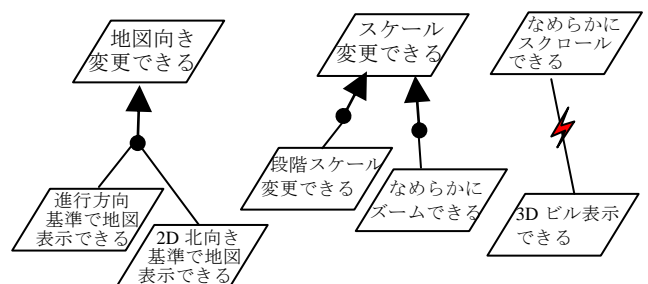


図 3 AND/OR リンクと障害リンク

Figure 3 AND link, OR link and obstacle link.

オブジェクトモデルはゴールの定義に用いられる用語をオブジェクトとして定め, ゴールとオブジェクト間の関係を

表現したモデルである。たとえば次図 4 で『登録地設定できる』というゴールは、『登録地』というオブジェクトに関心を持っている。すなわち、『登録地』オブジェクトの状態が、『登録地設定できる』ゴールの成否に影響する。『登録地』オブジェクトは、『名称』、『読み仮名』、『電話番号』、『場所』および『地点マーク』というオブジェクトの集約である。以降、KAOS 要求分析モデルのオブジェクトモデルを KAOS モデルと記述する。

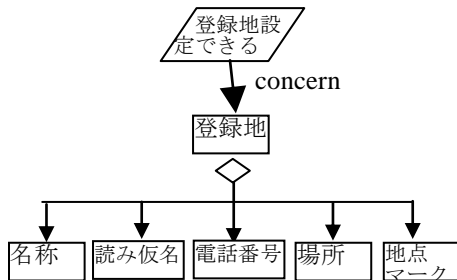


図 4 オブジェクト  
 Figure 4 Object.

### 3. 製品開発の課題と要件

本研究では、カーナビ等、多機種・多機能な製品を対象としたプロダクトライン開発の要求仕様の煩雑化改善に着目する。実際の製品開発では多機種かつ多機能で大規模なソフトウェアを対象とするため、要求仕様は非常に煩雑である(図 11, 図 12 参照)。また、大勢の開発メンバーで分担して開発するため、要求仕様を全体的に把握しにくく、他メンバーがどんな仕様の開発をしているか認識していないケースはまれではない。そのような開発状況では、顧客の意図と異なるソフトウェアを開発したり、類似の仕様を重複して開発したり、再利用を想定した設計ができない場合もある。そこで、要求仕様を全体的に把握しやすいようにする必要がある。

製品開発では短期間での開発であるにも関わらず頻繁に要求仕様の変更がある。そこで、要求仕様の変更に対してメンテナンスしやすいようにする必要がある。

多機種の製品を同時に開発するが、例えば地図表示開発チームは A 社、B 社、C 社、…、すべての顧客向けの地図表示機能開発を担当する。異なる顧客でも共通の機能であれば、ソフトウェアは再利用できる。そこで、複数製品の要求仕様が同時に確認や比較できることが望ましい。同様に、新規に開発対象の製品の要求仕様と既存の類似した製品仕様が比較できれば、仕様そのものの抜け漏れがチェックでき品質の高い仕様を作成できる。例えば、新規製品の仕様にスケール変更機能がある場合、新旧の仕様が比較できれば、どのようにスケール変更するかも決めておかなければならないことがすぐにわかる。

以上より、次の要件に着目する。

- ・要求仕様が全体的に把握しやすいこと。

- ・要求仕様の変更に対するメンテナンスがしやすいこと。
- ・複数製品の要求仕様が同時に確認できること。

### 4. 本研究提案手法

本研究では、要求仕様の煩雑化改善を目標とし、顧客要求の分析に着目する。要求をもれなく分析し、品質の高い要求仕様を作成するためにゴール指向要求分析手法 KAOS を適用する。プロダクトラインの要求仕様を統合した KAOS モデルとして表現する。KAOS モデルはルートのゴールに近いほど機能がまとめて表示されており、要求仕様の概要を把握しやすくなっている。プロダクトライン開発では、一般にフィーチャモデルを用いて分析するが、フィーチャモデルではなぜその要求仕様が存在するのかという由来や、どの顧客向けの仕様であるかが不明である。一方、KAOS 要求分析モデルは要求仕様の由来は明確であるが、製品ごとに指定する必須・オプションの指定など複数製品の表現をサポートしていない。そこで、仮にゴールモデルとフィーチャモデルを併用した場合、ゴールとフィーチャのわかりやすい相互の対応づけが必要になってくる。また、仕様変更や新機種追加による要求分析モデルのメンテナンスを行う場合、ゴールグラフとフィーチャモデルどちらも修正を行うことは、多機能で大規模なソフトウェアを対象とする実際の製品開発では非効率的である。そこで、KAOS モデルをベースにフィーチャモデルの機能もサポートできる要求分析手法を提案する。次図 5 のように、要求仕様を共通部分と各顧客固有の部分に分け、要求分析モデルとして再利用できるようにする。さらに、情報可視化手法を用いて、複数製品の要求仕様が同時に確認できるようにする。KAOS モデルとして統合するための手法および可視化手法を用いた視認性の改善について順次説明する。

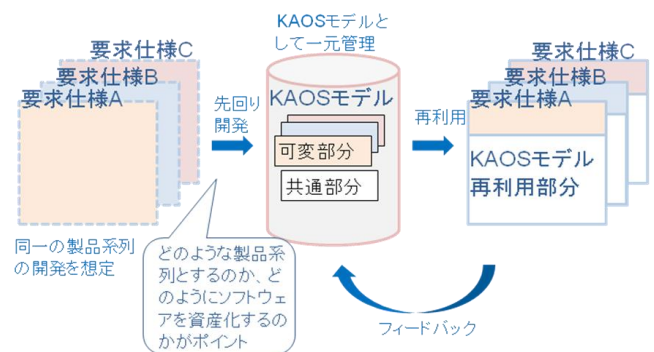


図 5 要求仕様の統合

Figure 5 Integration of requirements specifications.

#### 4.1 ゴールの設定

類似の仕様でバリエーションのある多機能なソフトウェア開発では、すでに開発した製品をベースとして若干の新規機能を追加するケースが多い。すなわち、すでに要求仕様が機能レベルに落とされているものが大半である。機能



レベルに落とされた仕様，もしくはフィーチャモデルでフィーチャとして表現される仕様を KAOS モデルでゴールとして設定する．ルートのゴールには開発対象の製品システムを実現したいというゴールを設定する．ルートのゴールと，機能を定義したゴールの間を，その機能は製品を実現するためになぜ必要なのかという観点で埋めていく．カーナビを例にすると，ルートゴールは『カーナビを実現したい』である．機能のゴールとしては『地図表示できる』『ルートを設定できる』『案内表示できる』などあげられる．次図 6 右のように，①『地図表示できる』のサブゴールとして②『画面スクロールができる』という機能を定義するとき，画面スクロール機能はなぜ必要か，という観点で③『任意の地点を地図表示できる』というゴールを設定する．次図左側は地図表示についてのフィーチャモデル(図 12 抜粋)，右側はフィーチャモデルに該当する KAOS モデルである(図 11 抜粋)．

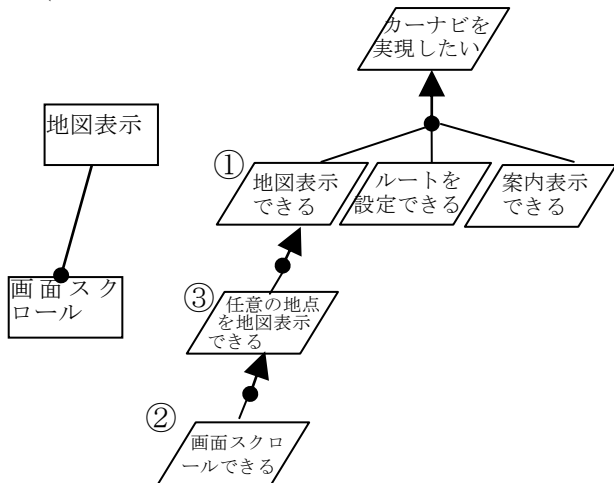


図 6 ゴールの設定

Figure 6 Goals configuration.

#### 4.2 共通仕様・固有仕様の表現

フィーチャモデルでは製品に共通の特徴は必須のフィーチャとして，製品ごとに異なる特徴はオプションフィーチャとして表現される．なお，どの製品がオプションを選択する・しないはフィーチャモデルと別に管理されている．一方，KAOS モデルでは，オプションの表現法をサポートしていない．そこで，提案手法では次図 7 右側に KAOS モデルをベースになぜフィーチャモデルのオプション(図 7『スクロール方面名称表示』)に該当するゴール(図 7『方向を確認しながらスクロールできる』)を間に追加し，フィーチャモデルで必須の要求(図 7『上下左右斜めスクロールできる』)は必ず含まれるようにリンクする．次図左側は画面スクロールについてのフィーチャモデル(図 12 抜粋)，右側はフィーチャモデルに該当する KAOS モデルである(図 11 抜粋)．

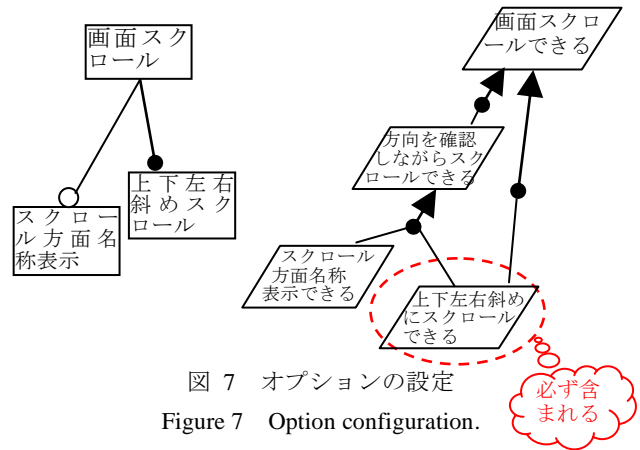


図 7 オプションの設定

Figure 7 Option configuration.

#### 4.3 オブジェクトの設定

フィーチャモデルにおいてデータとして表現されるフィーチャは KAOS モデルではオブジェクトとして表現する．次図左側は登録地設定についてのフィーチャモデル(図 12 抜粋)，右側はフィーチャモデルに該当する KAOS モデルである(図 11 抜粋)．

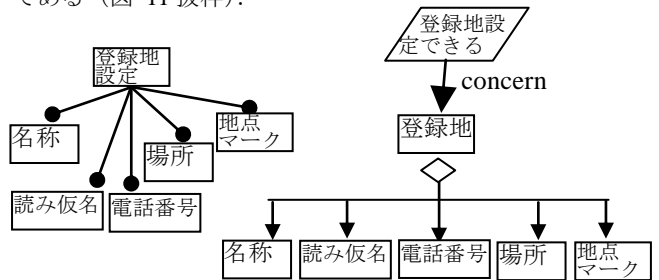


図 8 データの設定

Figure 8 Data configuration.

#### 4.4 情報可視化の適用：複数製品の表現

KAOS モデルでは，複数の製品を同時に表現するケースを想定していない．そこで，次図のようにゴールを対象顧客ごとに色分けして表現する．色を割り当てる際，全製品共通色は白，全製品でないが複数製品共通色は薄い系統の色，各製品独自色は濃い系統の色というように，カテゴリに応じて濃淡を設定すると共通性の度合いが表現できる．

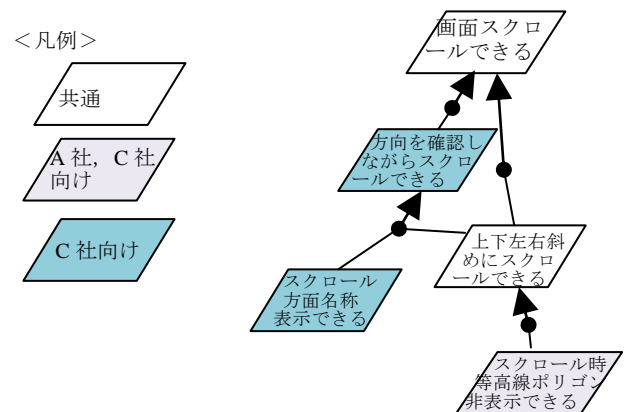


図 9 顧客ごとの色分け

Figure 9 Color coding according to stakeholders.

4.5 情報可視化の適用：ゴールの強調

特定のゴールを強調することにより、着目している要求が一目で確認できるようにする。例えば、製品の特徴のみを確認したいときは、フィーチャに該当するゴールを拡大して強調し、それ以外のゴールを縮小して表示する。次図左側はフィーチャモデル（図 12 抜粋）、右側はフィーチャモデルに該当する KAOS モデルである（図 11 抜粋）。

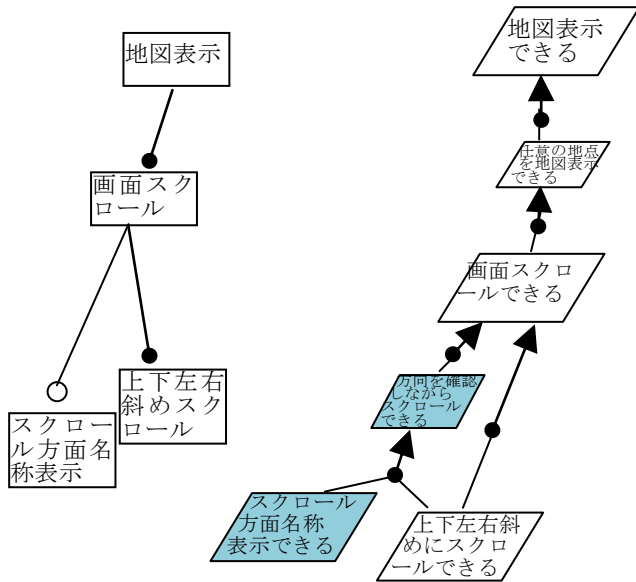


図 10 ゴールの強調  
 Figure 10 Emphasizing goals.

5. カーナビへの適用と評価

5.1 カーナビへの適用

図 11 は本研究手法をカーナビへ適用した場合の KAOS モデル、同様に図 12 はフィーチャモデルである[7][8][9]。

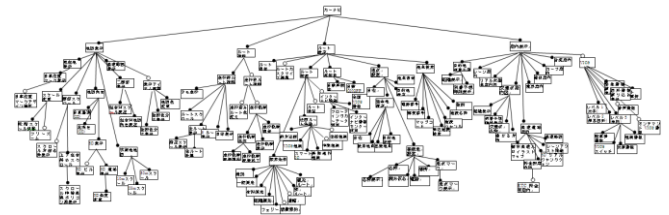


図 12 フィーチャモデル  
 Figure 12 Feature model.

図 11 は『カーナビを実現したい』というゴールをルートとして、カーナビの各機能について KAOS モデルを展開している。色が塗られていないゴールは各製品共通のゴールである。濃い系統の色が塗られているゴールは各顧客 A 社、B 社、C 社、D 社独自のゴールである。薄い系統の色が塗られているゴールは複数の顧客に共通ではあるが全製品共通ではないゴールである。図 11, 図 12 のゴールやフィーチャからなるノードの個数はそれぞれ表 1 のとおりである。既存の手法として、各顧客向けに分析した KAOS モデルおよびフィーチャモデルを併用して開発するケースと本研究提案の KAOS モデルを比較する。前提として、顧客は A 社、B 社、C 社、D 社の 4 社あり、各顧客向けにそれぞれ製品が 1 個存在すると仮定する。既存の手法として、

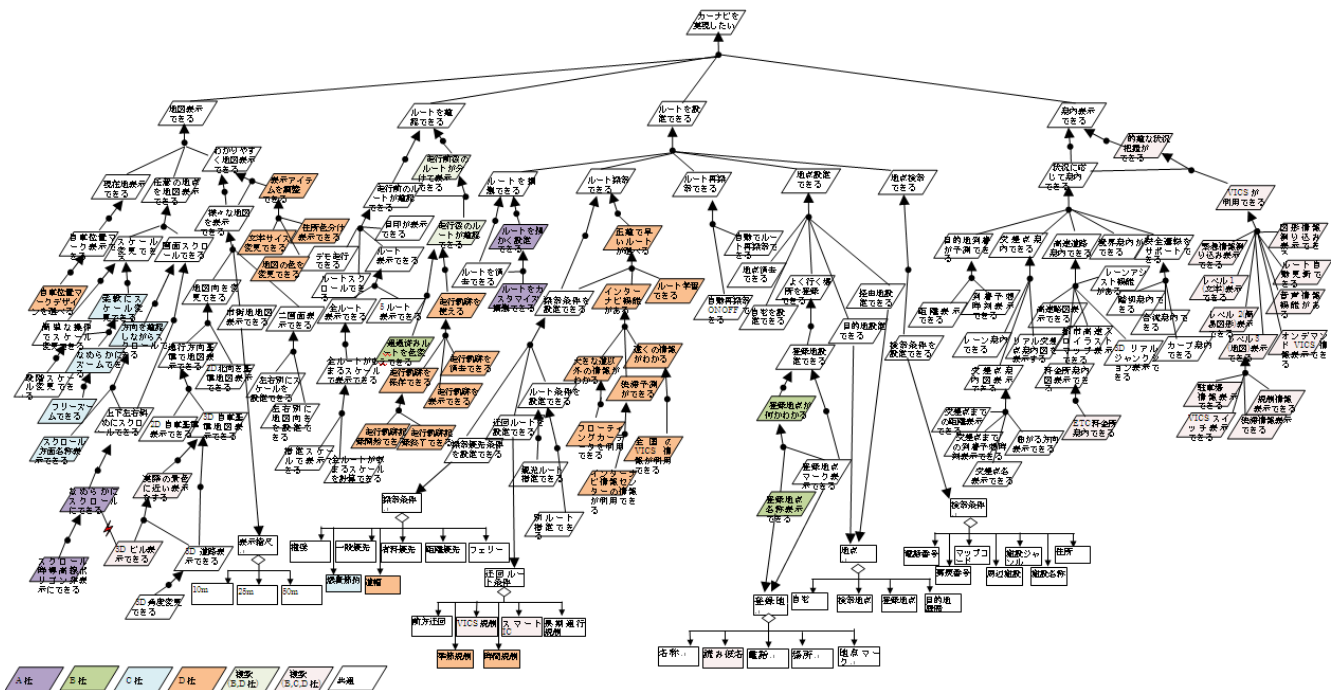


図 11 KAOS モデル  
 Figure 11 KAOS model.

表 1 ノードの個数

Table 1 Number of nodes

要求分析モデル	ノードの個数
本研究提案 KAOS モデル	170
A 社向け KAOS モデル	136
B 社向け KAOS モデル	117
C 社向け KAOS モデル	138
D 社向け KAOS モデル	153
フィーチャモデル	133

各顧客向け KAOS モデルとフィーチャモデルを併用して開発する場合、管理が必要なノードの個数は表 1 よりトータル 136+117+138+153+133=677 個である。一方、本研究提案の手法では KAOS モデルのみで統合して管理するため、170 個である。本研究提案のように、各顧客向け KAOS モデルを一つの KAOS モデルにまとめると、管理が必要なノードの数は  $170 \div 677 \approx 1/4$  となる。従って、ノードに関するメンテナンス作業が  $170 \div 677 \times 100 = 25\%$  程度へ削減できる。なお、顧客の数が増えるほど、本研究提案のように一つの KAOS モデルにまとめて要求仕様を統合して管理するほうが管理が必要なノードの個数について効率的であることは明らかである。また、まとめて表示しているの各顧客向けの仕様が同時に確認できる。

次に、実際の開発シナリオとして次のケースを考える。

1) 3D ビル表示機能: A 社向け製品でバグが発見された場合、本提案手法により 3D ビル表示は A 社、B 社、C 社の共通機能であるため B 社と C 社でもバグ改修が必要な可能性があることがすぐにわかる。また、B 社の顧客から自然な見栄えになるよう自車が道を曲がったときは地面だけでなく空も回転してほしいという要望がでたとき、A 社と C 社に同様な仕様の提案ができる。

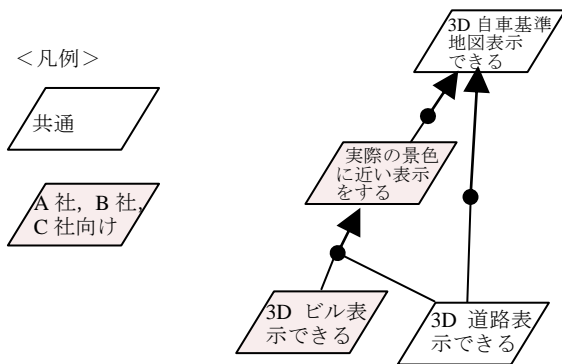


図 13 3D ビル表示  
 Figure 13 3D buildings.

2) 画面スクロール機能: スクロール時に等高線ポリゴンを描画する仕様がなくなる。消去する理由は描画処理を省くことでなめらかにスクロールできるよう性能を確保するためであり、単にスクロール中ポリゴンを描画を消去したいのではない。実機が新しい高性能なものになって、等高線ポリゴン

を描画してもスクロール性能に影響なくなったとき、等高線ポリゴンを消去する理由がわかっていたら顧客へスクロール時も等高線ポリゴンを描画するよう提案できる。

次図は試作ツール Visual-K[3]を用いて、特定のゴールを強調したものである。次図は図 10 右側の KAOS モデルであり、フィーチャに該当するゴールを拡大している。

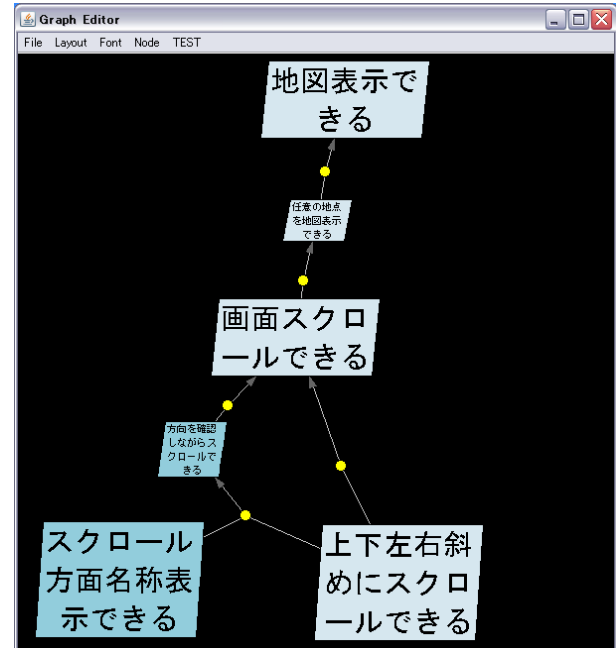


図 14 フィーチャに該当するゴールの強調  
 Figure 14 Emphasizing goals: feature.

同様に次図は図 10 右側の KAOS モデルであり、共通仕様に該当するゴールのみ強調表示したものである。

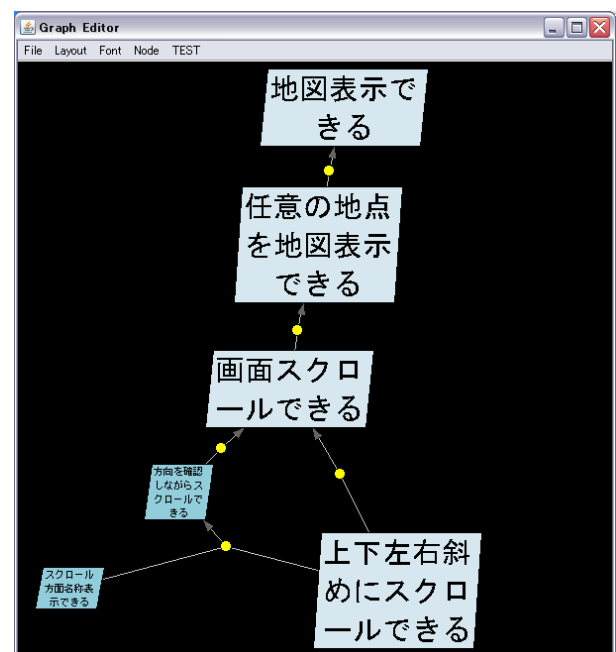


図 15 共通仕様に該当するゴールの強調  
 Figure 15 Emphasizing goals: common specifications.

5.2 評価実験

本研究提案手法の有効性について、情報科学科の研究室から10名のユーザを選び評価実験を行った。前提として、カーナビの顧客A社～D社の4社があり、カーナビは各社1製品の合計4製品である。4製品は顧客ごとに若干異なる仕様であるが、共通部分も多く、共通部分は同一のソフトウェアを使用している。各顧客(A～D社)からの要求仕様変更依頼があった場合、変更の影響範囲の確認しやすさについて次のKAOSモデルa), b)を用いて評価する。

- a) 図11の本研究提案KAOSモデル。ノード数170個。
- b) a)のモデルを各顧客向けの製品別に分析したKAOSモデル(図16, 図17参照)。ノード数は表1参照。

実験1) 要求仕様変更で影響する製品数のカウント

変更依頼してきた顧客(A社, B社, …)と変更対象の要求仕様に該当するゴールを合計15個提示した。なお、15個のうち3個はKAOSモデルに存在しないゴールを設定した。KAOSモデルa), b)それぞれから該当ゴールを探し、変更に影響のある製品数をカウントしてもらった。ここで、変更に影響のある製品数とは、例えばA社から『5ルート表示できる』仕様の変更を依頼された場合、『5ルート表示できる』はA社～D社向け製品共通の仕様なので4個とした。また、D社から『走行軌跡表示できる』仕様の変更を依頼された場合、『走行軌跡表示できる』はD社向け製品のみ仕様なので1個とした。

<提示したゴールの例→解答例>

- ・A社: 5ルート表示できる→共通仕様なので、4個
- ・D社: 走行軌跡表示できる→D社独自仕様なので、1個

実験2) ある要求仕様変更で影響する要求仕様のカウント

実験1)同様、顧客と変更予定のゴールを合計10個、うち3個はKAOSモデルに存在しないゴールを提示した。KAOSモデルa), b)それぞれから該当ゴールを探し、各社製品ごとに葉ゴールの数をカウントしてもらった。ここで、葉ゴールとは末端のゴールで子ゴールをもたないゴールのことである。

<提示したゴールの例→解答例>

- ・D社: 走行前後のルートが分けて表示できる  
 →A社0個, B社5個, C社0個, D社5個
- ・B社: 走行後のルートが確認できる  
 →A社0個, B社1個, C社0個, D社1個

表2 実験結果

Table 2 Experimental result

	平均正解率%		平均解答時間	
	本研究提案 a)	製品別 b)	本研究提案 a)	製品別 b)
実験1	95	97	12分31秒	14分30秒
実験2	88	92	9分25秒	11分11秒

表3 不正解の原因

Table 3 The cause of error

	本研究提案 a)			製品別 b)	
	色付きゴールの解釈間違い	指定ゴールが発見できない	設問読み間違い	指定ゴールが発見できない	カウント間違い
実験1	0	3	1	2	0
実験2	5	0	1	2	2

実験結果を表2および表3に示す。平均正解率は製品別b)のほうが本研究提案a)より若干よいが、概ね9割以上となった。正解率の差は、本研究提案a)の不正解の半数が色付きゴールの解釈間違いであることから、色付きゴールすなわち複数製品の定義が若干複雑であり、製品別のモデルのほうがシンプルな形状であるためと思われる。平均解答時間は本研究提案a)のほうが短く、製品別b)にかかる時間の85%程度であった。本研究提案a)のノードの個数が製品別b)より少なく確認しやすいためであると思われる。

なお、ゴールの検索については目視ではなくコンピュータ上でのテキスト検索も考えられるが、仕様変更やバグ改修での依頼は要求分析時に用いている用語と同じとは限らない。よって、目視での見やすさは必要である。

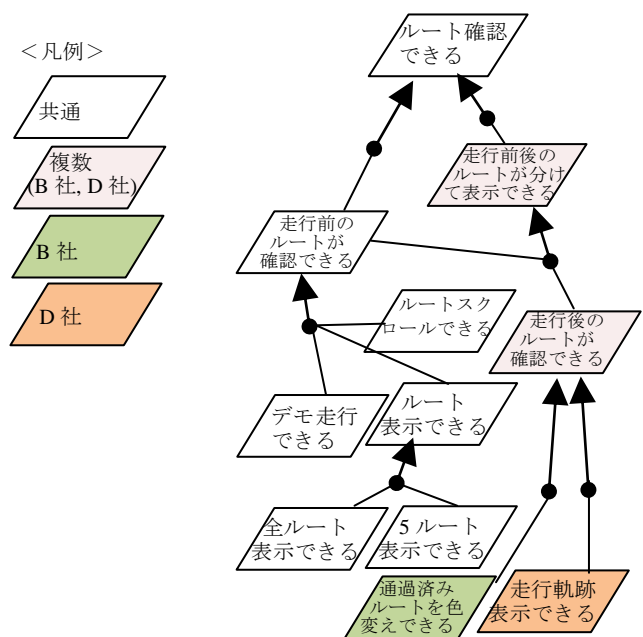


図16 統合したKAOSモデル(図11抜粋)

Figure 16 Integrated KAOS model (part of Figure 11)

以上のように、多機能な複数製品の要求仕様を一つのKAOSモデルに統合し、要求仕様を全体的に把握しやすくした。また、要求仕様のメンテナンス対象を削減した。ゴールを色分けすることにより、特定の要求仕様がどの製品に割り当てられているか一目で確認できるようにし、複数製品の要求仕様が同時に確認できるようにした。

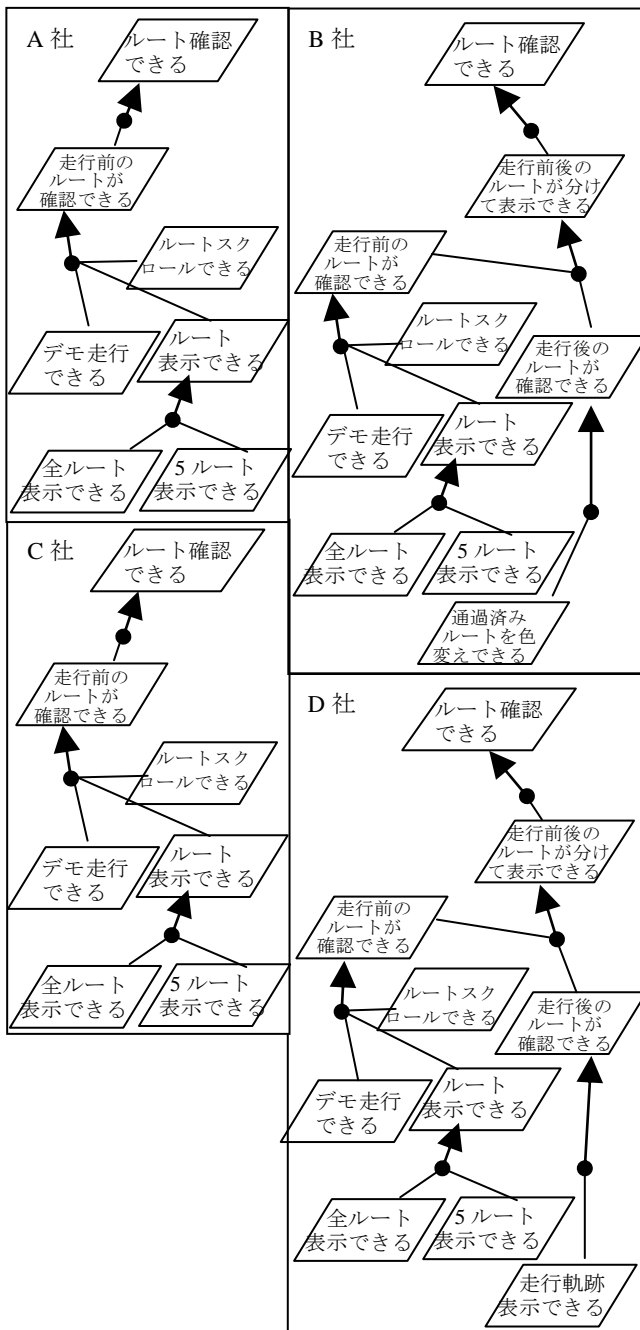


図 17 製品ごとの KAOS モデル (抜粋)

Figure 17 KAOS model for each product (modified Figure 16)

## 6. 従来研究

フィーチャモデルからゴールグラフへのマッピングについて Bobra らの研究がある[1]. しかし, ゴールグラフは i\*手法を対象とした取り組みであり, またどの製品がどのゴールに対応しているかの割り当てについては触れていない. 宇野らは複数のゴールモデルを統合する取り組みを行っている[2]. しかし目的は, フィーチャモデルへ変換するためであり, 統合したゴールモデルについてもどの製品がどのゴールに対応しているかはやはり触れていない. 他に, Yu らや Asadi らの研究もあるが, どちらもゴールモデルから

フィーチャモデルのマッピングに関するものである[4][5]. Heidenreich らはフィーチャモデルから実際のモデルへのマッピングにおいて情報可視化を考慮しているが, マッピング元の使われないモデルをグレイアウトするというものであり, 複数製品の仕様を同時に表示するためのものではない[6]. いずれの既存研究も KAOS 手法を用いて複数製品の要求仕様を統合して管理しようとしていない. また, 実際の製品開発では多機能な仕様であるため大きく複雑な要求分析モデルとなることが予測されるが, 視認性の改善についても十分研究されていない.

## 7. おわりに

本論文では, プロダクトライン開発の実際の製品開発を想定したシステムの要求分析モデリング手法を提案した. KAOS 要求分析モデルをベースにフィーチャモデルの機能も含み, 多機能な複数製品の類似した要求仕様をひとつの KAOS モデルで統合して管理できるよう工夫した. また, 情報可視化手法を適用し, 要求分析モデルの視認性を改善した. 今後の課題としては, 複雑な要求間関係をわかりやすく表現する手法や, モデリングツールの開発があげられる.

## 参考文献

- 1) Borba, C. and Silva, C. : A Comparison of Goal-Oriented Approaches to Model SPLs Variability, Proc. ER Workshops, pp244-253 (2009).
- 2) 宇野 耕平, 林 晋平, 佐伯 元司: ゴールグラフからのフィーチャモデル導出, 情報処理学会研究報告, Vol.2009, No.31, pp. 1-8 (2009).
- 3) ジリエ 陽子, 本田 耕三, 中川 博之, 田原 康之, 大須賀 昭彦: Visual-K : ゴール指向要求分析手法 KAOS のモデリング可視化支援ツールの試作, 情報処理学会研究報告, Vol.2011-SE-171, No.29, pp. 1-7 (2011).
- 4) Yu, Y., Leite, J.C.S.D.P., Lapouchnian, A., and Mylopoulos, J. : Configuring features with stakeholder goals, Proc. 23rd ACM Symposium on Applied Computing(SAC'08), pp645-649 (2008).
- 5) Asadi, M., Bagheri E., Gasevic D., and Hatala M. : Goal-driven software product line engineering, Proc. 26th ACM Symposium on Applied Computing(SAC'11), pp691-698 (2011).
- 6) Heidenreich, F., Kopcsek, J., and Wende, C.: FeatureMapper: mapping features to models, ICSE Companion, pp943-944 (2008).
- 7) HONDA ナビゲーション取扱説明書  
<http://www.honda.co.jp/manual-access/navi/>
- 8) 三菱電機 カーナビゲーションシステム取扱説明書  
<http://www.mitsubishielectric.co.jp/carele/carnavi/manual/manual.html>
- 9) 三菱自動車 ナビゲーション取扱説明書  
<http://www.mitsubishi-motors.co.jp/support/manual/index.html>