社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＳＥ－160 (10)
2008－EMB－ 9 (10)
2008／6／20

組込みソフトウェアのためのゴール指向手法による状態モデル構築法

金子省太† 小倉信彦‡ 渡辺晴美§

組込みソフトウェアでは，様々な制約を持つことが，要求モデルと設計モデルとの間の追跡可能性を損なう要因となっている．したがって，追跡可能性を実現し，品質・生産性を向上するために，ドメイン知識を得ることにより制約を明確化することが必要となる．本稿では，追跡可能性の問題に着目し，ゴール指向モデルを利用し，ドメイン知識を獲得する方法を提案する．提案する方法では，ゴールからシナリオを記し，シナリオから状態モデルを構築する．状態モデルをもとにプロトタイピング実験を行い，その結果をもとに，次のゴールを発見する．本方法により，設計モデルに必要なドメイン知識を得るプロセスが明確になり，要求分析・設計で得る各々の状態モデル間の追跡可能性が高まる．

A state modeling technique for embedded software using goal-oriented model

SHOTA KANEKO† , NOBUHIKO OGURA‡ and HARUMI WATANABE§

Traceability has impact on quality and productivity in software development. Various constraints on embedded software, however, make it difficult to keep traceability between requirement models and design models. To clarify these constraints, acquiring domain knowledge is essential. In this paper, we propose a method to acquire domain knowledge based on goal-oriented methods. In the method, scenarios are described from goals, and state models are constructed from the scenarios. Then, experiments using prototype models results in next goals. Through repeating these process, we can clarify domain knowledge. Proposing methods consist of explicit process, which helps to improve traceability.

1. はじめに

ソフトウェア開発プロセスは複雑であり各プロセスで構築されるモデル間で関連付かないことがある．このような追跡可能性が失われる状況が，組込みソフトウェア開発において多々発生する．本稿での追跡可能性とはモデルの構成要素各々の関連性が明確であることの度合いを指す．組込みソフトウェアの開発では考慮すべき制約が多く，これらの制約を明らかにするためには，ドメイン知識の獲得が必要になる．獲得したドメイン知識を状態モデルに反映させる際に，モデルが変化し，時には構造が変わるほどの変化が起きる．この様な変更が度重なることにより，追跡困難なモデルが構築されてしまう．

本稿では，要求分析・設計の各プロセスで構築する状態モデル間の追跡可能性の問題について，ドメイン知識獲得プロセスを明確にすることで解決する．提案する方法では，ゴール指向モデルを利用する．ゴール指向とは，システムを開発するのはなんらかの目標(ゴール)を達成するためであるはずだから，その目標を構造的に明らかにすればシステムへの要求が明確になるだろうという考え方である[15]．ゴール指向モデルとはゴール指向の概念に基づいたモデルであり，ゴールをAND-OR ツリーグラフで展開する．

提案する方法では，ゴールからシナリオを構築し，状態モデルの作成，プロトタイプ実験を行い，次のゴー

†東海大学 開発工学部 情報通信工学科
Department of Information and Communication Technology, School of High-Technology for Human Welfare, Tokai University
‡武蔵工業大学 環境情報学部
Faculty of Environmental and Information Studies, Musashi Institute of Technology
§東海大学 情報通信学部
School of Information and Telecommunication Engineering, Tokai University

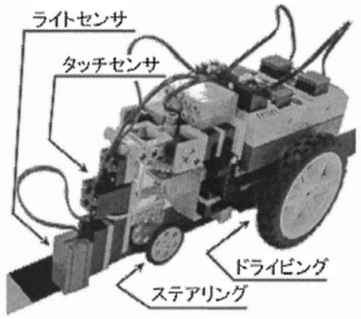

図 1 ライントレーサと各部名称

ルを発見するというプロセスにより，ドメイン知識を獲得するプロセスを明確化する．ゴール，シナリオ，状態モデルは，意図と挙動のゴールデントライアングルとも呼ばれており[1]，これらを関連付けることで追跡可能性の問題解決が期待できる．ただし，本稿で着目した制御を行う組込みソフトウェアの設計では，ドメイン知識の獲得が必要であり，そのプロセスにおいては，ゴールを明確にし，プロトタイプ実験を行うことが不可欠であることから，プロトタイプ実験を追加したプロセスを提案する．

以降の構成を以下に示す．2 節では，状態モデル間における追跡可能性低下の要因について明確にする．3 節では，問題を解決するためのゴール指向分析を利用した状態モデル表記法を提案する．4 節では，提案する表記法を組込みソフトウェアの開発事例を通して評価し考察する．5 節では，関連研究との比較を行う．6 節で本稿をまとめる．

2. 追跡可能性の低下

組込みソフトウェア特有の問題である要求分析・設計モデル間での追跡可能性低下の要因を考察し，事例を用いて説明する．

2.1. 追跡可能性低下の要因

本稿では，最初にプロトタイプ実験を開始する前までのプロセスを要求分析プロセスと呼び，要求分析で作成した状態モデルを要求分析モデルと呼ぶ．また，要求分析の結果からシナリオを通して要求を実現するためのプロトタイプを構築・実験を行うプロセスを設計と呼ぶ．設計で作成する状態モデルを設計モデルと呼ぶ．設計は，本来，制約事項を取り入れ，実装方法を示す役割を持つ．従って，設計モデルには制約事項，実装方針が盛り込まれる．本稿では，制約のみに着目

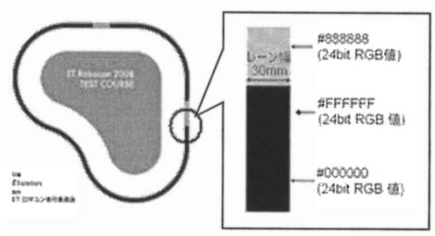

図 2 テストコース

する．機器を制御する組込みソフトウェアでは，制約事項に，電気・電子および機械の制約を含み，これらの制約を明示するには，度重なる実験が必要となる．度重なる実験の結果得た制約に関する知識をドメイン知識と呼ぶことにする．

このドメイン知識を得るプロセスでは，実験途中で様々な問題を発見し，解決していくことで，最適解を模索する．ドメイン知識を得るプロセスが明示されずに，モデルやプログラムが構築されると，追跡可能性が失われ，加えて，理解不能なモデルやプログラムになる．

2.2. 追跡可能性低下の事例

追跡可能性低下の要因について，ET ソフトウェアデザインロボットコンテスト(ET ロボコン)[16]の事例で説明する．図 1 に ET ロボコンのロボットであるライントレーサを示す．ライントレーサは，ライトセンサ，タッチセンサ，ステアリングモータ，ドライビングモータを持つ自律走行型ライントレースロボットである．走行に使用するテストコースは図 2 に示す通りであり，黒色，灰色，白色の部分から構成されている．事例では，ライントレーサを黒色のラインに沿って走行させ，灰色のマーカを感知すると停止させたいものとする．

図 3 は要求分析モデル，図 4 は幾度かのプロトタイプ実験後の設計モデルである．両者に「黒色」「白色」の同一名称の状態を見付けることはできるが，要求分析から設計へとどう変化したか，なぜこのような設計モデルが構築されたかを判断することは困難である．開発が進むに連れて，さらにその差は大きくなり，追跡することは，より一層難しくなる．この差は，上述したように，ドメイン知識の獲得のために，プロトタイプ実験を行い，その結果，設計モデルを構築したことに起因する．以下，各々の図について説明する．

図 3 の要求分析モデルは，ライトセンサは，黒色ならば小さな値，白色ならば大きな値，灰色ならば黒色と白色の値の中間を返すであろうと推測し，記した図である．

図 4 の設計モデルは幾度かのプロトタイプ実験を行った結果構築したモデルである．ドメイン知識獲得のためのプロトタイプ実験を図 5，図 6 に示す．図 5，図 6 は，

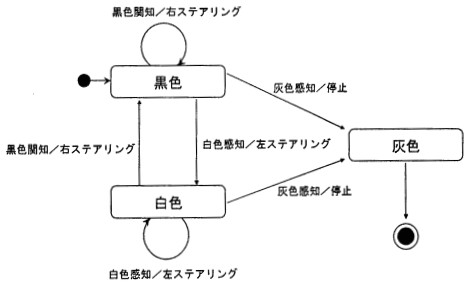

図 3 要求分析モデル

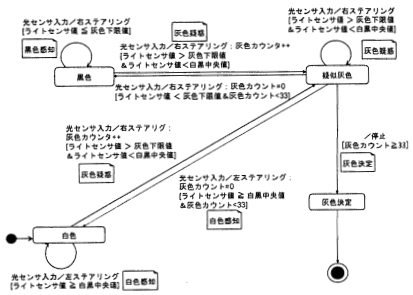

図 4 最終的な設計モデル

各々，黒線上，灰色線上を走行している際の実験結果である．いずれも縦軸がラインとセンサ値、横軸が判別回数となっている。ライトセンサ値は、ライトセンサが受ける明るさ，黒は弱く，白は強いことから、黒の値は小さく，白の値は大きい．図 5 の白黒中央値よりも、上部が白色部分で、下部が黒色部分である．図 6 の灰色下限値から白黒中央値の間が，灰色部分の明るさをセンサが獲得していることが分かる。

これらの図から，グラフの下側ピークが，黒色と灰色では異なることが分かる．ただし，黒線上を走行している際も，灰色と同じ明るさを，ライトセンサは獲得する．従って，単に要求分析モデルで想定した方法では正常に動作しない．そこで，灰色線上を走行している際の灰色判別数は，黒線上を走行している際の灰色判別数よりも多く，おおよそ 33 回程度が境目であることを実験により得たことに着目し，判別回数が 33 回よりも多いか少ないかにより．灰色を識別することにする．図 4 の「灰色カウント」が，この役割を果たしている．ただし，実験を重ねると図 4 の実装では，安定して走行することは難しく，実際には別の考え方を採用している．

実際に最終ゴールを満たす設計モデルはさらに複雑である．「停止」命令から実際に停止するまでの距離とドライビングモータとの関係，「脱線復帰」のアルゴリズム，LEGO ロボット転倒の危険性が発生し速度調整しなければならなくなる等，解決すべきゴールは多数ある．組込みソフトウェアの特徴であるハードウェア特性などの操作環境を十分に反映させた設計モデルは，要求分析モデルとの関連付けが容易ではなくなる．結果として，追跡可能性は限りなく低下してしまう．

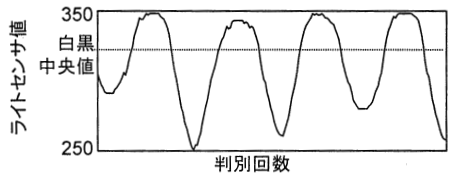

図 5 白色－黒色 閾値実験データ

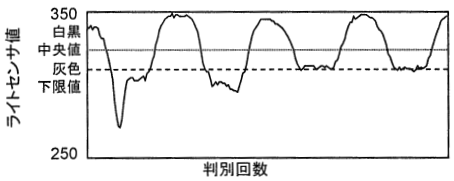

図 6 白色－灰色 閾値実験データ

3. ゴール指向モデルとドメイン知識獲得実験による追跡可能性の向上

前節で明らかにしたドメイン知識明確化の問題に対して，本節ではゴール指向分析を利用した状態モデル構築法を提案する．本方法は，図 7 の概要に示すとおり，プロセスは「要求分析」，「設計」からなる．設計プロセスは，ゴール・シナリオ生成とプロトタイプ実験の 2 つのプロセスに分かれる．図 7 の下部に記した操作環境は，ドメイン知識が必要となるハードウェア，すなわち，機械，電気・電子等である．各プロセスの詳細について，2 節で用いたライントレーサの開発事例に沿って以下で説明する．

3.1. 要求分析

要求分析プロセスは，ゴール指向モデルを展開し，最初にプロトタイプ実験を開始する前までのプロセスであり，要求分析プロセスのプロダクトは，本稿で要求モデルと呼ぶ状態モデルである．要求分析プロセスは以下の通りである．

(1) 図 8 の要求分析に示す通り，ゴールを AND-OR 分解しゴール指向モデルを構築する．

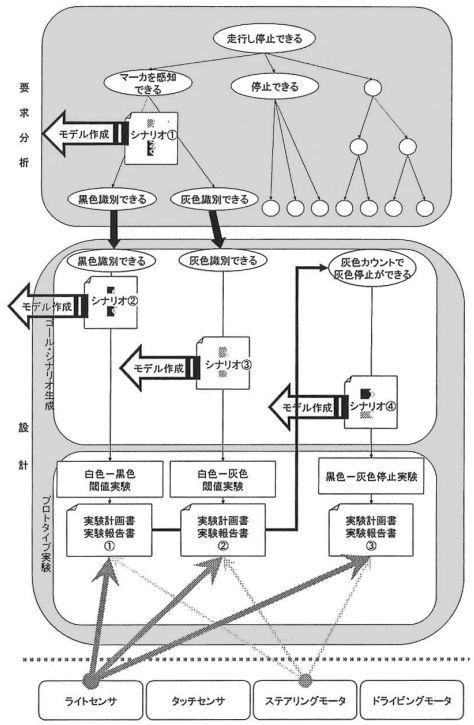

図 7 提案方法概要

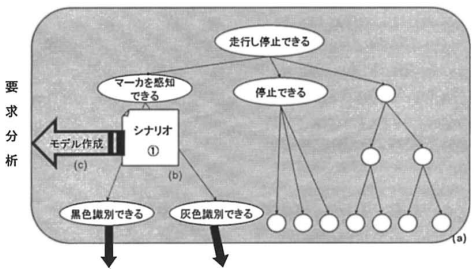

図 8 要求分析

(2) 必要に応じてゴールからシナリオを作成する. 1 つのシナリオは, 1 つ以上のゴールをもとに作成する. シナリオは, 次のゴールを発見する手がかりとするか, 要求モデルを構築する基となる. 1 つの要求モデルは, 1 つ以上のシナリオをもとに構築する. 尚, 要求分析・設計を通し, 各々のシナリオにはシナリオの他に入力・出力を一緒に記しておく.

図 8 の例では, 開発目標である黒色のラインに沿って走行し, 灰色のマーカを感知したら停止できることをゴールとして, ゴール指向要求分析手法により図 8-(a) の通りゴール指向モデルを展開していく. 途中, 灰色のマーカを検知できるためには黒色識別ができることと, 灰色識別ができることが必要であると考える. ここで灰色のマーカ感知のシナリオとして図 8-(b) の通り作成する. この図 9 に示すシナリオ①から図 3 に示す要求分析モデルを図 8-(c) の通り作成する.

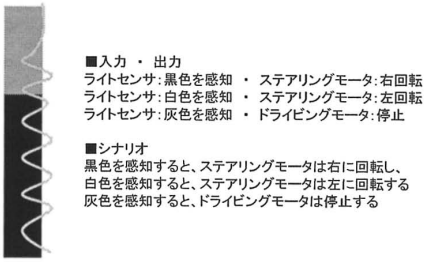

図 9 シナリオ①

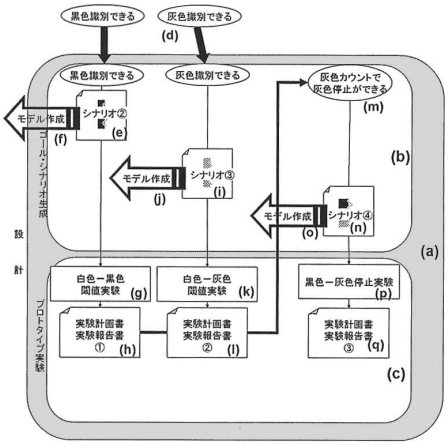

図 10 設計

3.2. 設計

設計プロセスは, ドメイン知識を獲得し, 制約事項を盛り込んだ設計モデルを構築するプロセスである. 本稿での設計モデルは, 制約事項を盛り込んだ状態モデルのことである. 上述したように, 設計プロセスはゴール・シナリオ生成, プロトタイプ実験の 2 つのプロセスに分かれる. 各々のプロセスについて以下に記す. 図 10 では, (b) が, ゴール・シナリオ生成であり, (c) がプロトタイプ実験である.

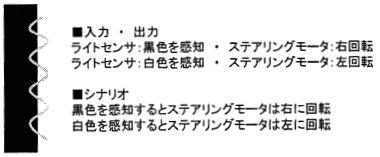

図 11 シナリオ②

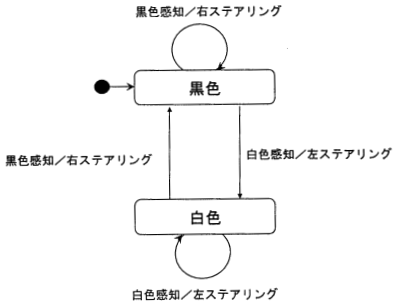

図 12 設計モデル:シナリオ②

3.2.1. ゴール・シナリオ生成

要求分析によって作成される末端のゴールが，ゴール・シナリオ生成プロセスの開始のゴールとなる．尚，要求分析で得た末端のゴールは複数あることから，設計プロセスの開始も，複数のゴールからなる．シナリオは，要求分析と同様，1 つ以上のゴールから作成する．図 10 の例では，説明を簡素化するために，ひとつのゴールから一つのシナリオを作成するようにしている．本例では，黒色識別できるというゴールから黒色を識別するシナリオを表したのが図 11 である．また，図 11 のシナリオは，図 10 のシナリオ②である．シナリオに対応する設計モデルを図 10-(f)のタイミングで作成する．ここでの設計モデルを図 12 に示す．最初の段階では，ドメイン知識を獲得していないため，図 12 で得た状態モデルは，図 3 の状態モデルと同じモデルまたは部分モデルとなる．

3.2.2. プロトタイプ実験

設計を進める中で，組込みソフトウェア開発では，ハードウェア特性などのドメイン知識の獲得が重要となる．このドメイン知識を獲得するために，プロトタイプを用いた実験を行う．実験では，単に物理的な特性を計測することもあるが，動作させて実験を行う際は，シナリオから構築した状態モデルに基づき作成したプログラムを実行し，計測する．プロトタイプ実験では，実験前に実験計画書を作成し，実験後に実験報告書を作成する．実験計画書の項目を表 1 に示し，実験報告書の項目を表 2 に示す．実験結果を操作環境の要素として整理し，次のゴールを発見する．手がかりとする．

本事例での実験報告書の結果欄に記すべき取得結果の解析データの例が，上述の図 5，図 6 である．図 5 の実験から，ラインレースする際に，どのようにセンサの値が変化するかが理解でき，黒色の最大振幅，黒色と白色との閾値を得ることができる．後に，図 6 の結果から走行戦略が決まり，閾値という名称を白黒中間値に名称を変更している．一回目の実験では，ライトセンサ，ステアリングモータ，ドライビングモータを使用したことから，図 7 では関連付けがされている．以降は，上述のプロセスを繰り返す．

表 1 実験計画書の項目

目的	ゴールから何を実験・計測するかについて記す．
背景	実験が設計にどう役立つかを記す．
期待値	実験の期待値を記す
方法	実験の方法を記す．

表 2 実験報告書の項目

条件	ハードウェアの固有情報・自然環境等．
結果	結果をグラフと説明などで示す．
比較	期待値と実験結果を比較検討し記す．
材料	実験に使用した器材，プログラムを記す．

3.3. 設計プロセスの事例

以下，設計プロセスにおいて，状態モデルが変化していく様子について記す．要求分析において作成した灰色識別できるというゴールについて考える．これを実現するために図 13 に示すシナリオ③を図 10-(i)の通り考える．このシナリオから図 14 に示す設計モデルを図 10-(j)の通り作成する．ここでシナリオと設計モデルから，プロトタイプ実験として白色－黒色閾値実験を図 10-(k)の通り作成する．実験を行い，実験報告書を図 10-(l)の通り作成する．取得結果の解析データは図 6 に示す通りであり，灰色の最大振幅，灰色と白色との閾値を得た．ここで，実験を通して把握した操作環境について関連を示すため，関連する操作環境から実験報告書へ矢印を図 7 の通り繋ぐ．この関連付けにより，最終的な設計モデルの構築，テスト，保守，再利用時に操作環境側から追跡が可能になる．

上述の設計を繰り返した本来の目的は，その 1 つ上の「マーカを検知できる」というゴールから作成された戦略である図 8-(b)に示すシナリオ①を満たすためである．

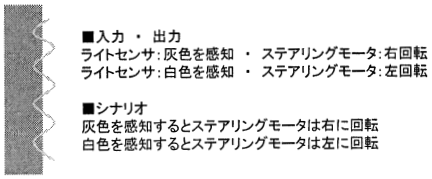

図 13 シナリオ③

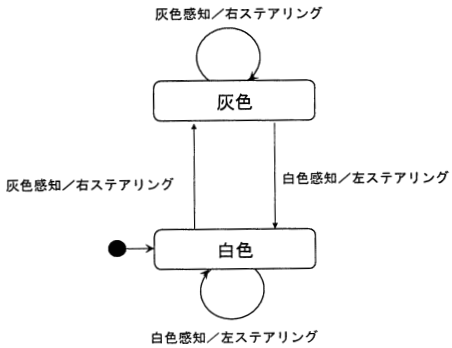

図 14 設計モデル:シナリオ③

これまでの成果物では未だにシナリオ①の実現には至ってはいないため，末端のゴールからの成果物図 10-(e)，図 10-(f)，図 10-(h)，図 10-(i)，図 10-(j)，図 10-(l)を再度検討し，新たに「灰色カウントで灰色定式ができる」というゴールを図 10-(m)の通り作成する．以降はゴール・シナリオ生成から同様に展開する．図 10-(m)に示したゴールを実現するための戦略を記したシナリオが，図 15 であり，図 10-(n)である．これを基に構築した設計モデルが図 4 である．ここでも，他のゴールと同様に，実験を通して把握した操作環境について関連を示すため，関連する操作環境から実験報告書へ矢印を繋ぐ．

以上の通り，まずゴール指向要求分析を行い，途中で必要に応じ戦略を記すシナリオと，要求分析モデルを作成する．ゴール・シナリオ生成では要求分析の末端にあるゴールからシナリオ，設計モデルを作成する．プロトタイプ実験ではシナリオと設計モデルから必要な実験を行い，操作環境のドメイン知識を獲得し実験報告書にまとめ，その結果を次々と反映させていく．以上が提案するゴール指向分析を利用した状態モデル構築法である．

4. 評価と考察

評価のため，以下の 2 点を確認する．

(1) ドメイン知識獲得前後の追跡可能性
(2) 要求分析・設計間での追跡可能性

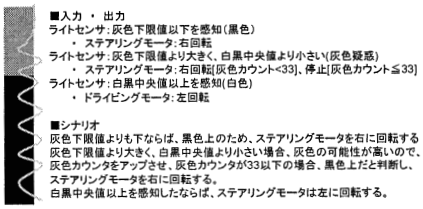

図 15 シナリオ④

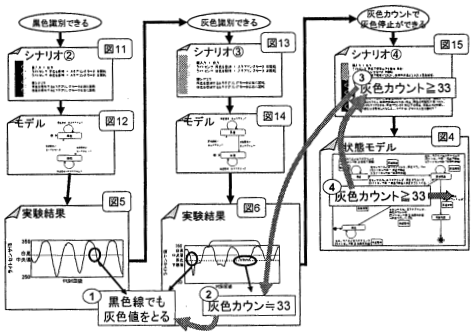

図 16 「灰色カウント≧33」追跡の例

4.1. ドメイン知識獲得前後の追跡可能性

ドメイン知識獲得前後の追跡可能性が低下する要因は 2 節から，プロトタイプ実験から得られるドメイン知識の獲得プロセスが複雑であることであった．

図 16 に示した追跡例は，図 4 の設計モデルにあるガード条件「灰色カウント≧33」を追跡する事例である．図 16 の④が出発点である．モデルのもととなったシナリオ④にも「灰色カウント≧33」がすでに記されていることが分かる．この追跡は，図 16 の④から③にあたる．このシナリオにあたるゴールを生み出した実験結果を辿る．これが，図 16 の②である．②から理解できるように，黒色線上でも灰色値を取ることが分かり，灰色線上の時は，黒色線上よりも，灰色値の値が多いこと実験から気づいたということが分かる．そして，灰色線と黒色線を分ける灰色値を検出する回数が 33 回ぐらいであるということが実験から理解できる．

以上から，本稿で提案した方法により，ドメイン知識獲得前後の追跡が可能になったと言える．尚，この例は，一つのガード条件の値を決定しただけの些細なことのように見える．しかしながら，この値を得るためには，相当な実験を繰り返したり，様々な工夫を行うことが必要となる．本稿で提案したような記録がなければ，開発者が代わった際に，引き継ぎができなくなり，再度，余計な実験を行うことになる．また，開発者が代わらなくて

も，忘却する可能性も十分にある．

要求分析・設計間での追跡可能性

この課題は，図3要求分析モデルと図4設計モデルの関連性をどのように見つけるかということである．状態モデル図12と図14は，図3とイベント名，状態名が統一されており，図3を調べるために，図12と図14を構築したことが理解できるが，図4は随分と，様子が違う．ここで，4.1.と同様に図16を辿る．図4の設計モデルのもととなるシナリオ図15に着目する．シナリオ図11，図13と図15は，状態モデルと同様，異なる単語で構成されている．ただし，図13では，「黒色感知」，「白色感知」という曖昧な表現の単語が，図15のシナリオで「灰色下限値以下を感知」，「白黒中央値以上を感知」と，各々，正確に定義されていることが伺える．また，その根拠は，図16の②①と辿ることで，理解できる．以上から，要求分析・設計モデル間で追跡が可能であることを示した．

上記の通り，追跡できることにより，要求分析モデルから，受け入れテスト項目と設計モデルとが関連付き，テスト結果を反映しやすくなるである．また，機能追加や仕様変更の際にも，開発の背景を辿れることから，影響範囲を掴みやすくなると期待できる．

5. 関連研究

本稿で提案した方法で用いているゴール，シナリオ，状態モデルについて，[1]は，意図と挙動に関するゴールデントライアングルと呼び，ゴール，シナリオから状態モデルを構築する方法について提案している．ただし，[1]のシナリオはメッセージシーケンスチャートであるが，我々のシナリオは，デバイスを明らかにし，その関係を自然言語で記述していることからメッセージシーケンスチャートは容易に構築できる．[1]の方法に対し，我々の方法では，新たにプロトタイプ実験が加わっている点で異なる．

[5]は，各ゴールにシナリオを付け，そのシナリオをもとに次のゴールを発見する方法を提案している．[5]では，ゴールとシナリオが1対1であるのに対し，我々の方法では多対多である．1つのゴールを実現するために，複数の戦略に応じたシナリオが考えられることもあり，複数のゴールを一つのシナリオで実現することもあるためである．また，シナリオから次のゴールを発見するのではなく，状態モデルを構築し，プロトタイプ実験を行った上で，次のゴールを見つける点でも異なる．シナリオから状態モデルを構築する方法として[2][6]がある．以上から，ゴール，シナリオ，状態モデルの3点が意図と挙動を結びつけるのに重要であることが理解できる．

ゴール指向モデルと類似した技術にプロダクトライン開発で用いられる特性モデルがある．代表的なプロダクトライン開発技術の一つであるFORMの特性モデルは，特性をAND-ORツリーグラフで展開する．特性は，非機能要件，機能要件であり，AND-ORツリーグラフの展開を利用することから，ゴール指向モデルと類似している．FORMでは，抽出した特性を，操作環境，ドメイン固有技術，実装技術へと関連付ける．我々の操作環境は，FORMの考え方を応用した．特性モデルを使用したプロダクトライン開発は，産業界にも適用されている[13][14]．

CAPIS Web[12]は，組込みソフトウェアのためのドメイン知識を獲得するために行ったプロセスを記録することができ，ドメイン知識を獲得するまでの試行錯誤のプロセスを辿ることができる．ドメイン知識を獲得するプロセスの明確化が重要であることが理解できる．

以上から，本稿で提案した方法は，ドメイン知識の過程を明らかにした点，すなわち，ゴール，シナリオ，状態モデル，プロトタイプ実験を組みにし，次のゴール発見を行う点で従来の研究と比較し新しい．プロダクトライン技術への適用が期待できる．

6. おわりに

本稿では，要求モデルと設計モデルの追跡可能性が低下する問題の原因として，ドメイン知識を獲得するプロセスに着目し，ゴール指向モデルのゴールからシナリオを作成し，シナリオをもとに，状態モデルを構築，状態モデルをもとにプロトタイプ実験を行い次のゴールを発見するというプロセスを提案した．

提案した方法を評価するためにライントレーサの事例を用い，問題の要因と思われる要求分析・設計間，ドメイン知識獲得前後，両状況での追跡可能性の維持について考察を行った．今後は方法の改善に努め，モデル間の規則を見出し，ツールを構築したい．

参考文献

[1] C. Damas, B. Lambeau, A. van Lamsweerde :Scenarios, Goals, and State Machines: a Win-Win Partnership for Model Synthesis, Proc. FSE'06: Intl. ACM Symposium on the Foundations of Software Engineering, Portland (OR), pp.197-207, November 2006.

[2] C. Damas, B. Lambeau, P. Dupont, A. van Lamsweerde: *Generating Annotated Behavior Models From End-User Scenarios,* IEEE Transactions on Software Engineering, Special Issue on Interaction and State-based Modeling, Vol. 31, No. 12, pp. 1056-1073, 2005.

[3] A. Dardenne, A. van Lamsweerde and S. Fickas: Goal-Directed Requirements Acquisition. Science of Computer Programming Vol. 20, North Holland, 1993, pp. 3-50.

[4] KAOS home page http://www.info.ucl.ac.be/~avl/ReqEng.html

[5] Colette Rolland, Carine Souveyet, Camille Ben Achour: Guiding Goal Modeling Using Scenarios, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 24, NO. 12, pp. 1055 –. 1071, DECEMBER 1998
[6] Uchitel, Sebastian, Brunet, Greg, Chechik, Marsha: Behaviour Model Synthesis from Properties and Scenarios, ICSE2007, pp. 34-43, 2007
[7] Kang, K. C., Kim, S., Lee, J. y. Kim, K. "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures". Annals of Software Engineering, 5:pp.143—168. 1998.
[8] Miguel A. Laguna, Bruno González-Baixauli, José M. Marqués: Seamless Development of Software Product Lines -- Feature Models to UML Traceability, Proceedings of the 6th international conference on Generative programming and component engineering, pp. 85 - 94 , 2007
[9] John Mylopoulos, Lawrence Chung, Brian Nixon: Representing and Using Nonfunctional Requirements: A Process-Oriented Approach IEEE Trans. on Software Engineering, Vol. 18. NO. 6, pp.483- 497, June 1992
[10] John Mylopoulos, Lawrence Chung, Eric Yu: From object-oriented to goal-oriented requirements analysis, Communications of the ACM, Volume 42 , Issue 1, pp.31-37, January 1999
[11] 山本 修一郎, システム要求管理技法―ゴール指向による, ソフトリサーチセンター, 2007
[12] 飛内拓弥, 八木沼修, 大山勝徳, 薗田孝造, 武内惇, 藤本洋: 設計思考過程を支援するCAPIS Web に関する一考察, 組込みシステムシンポジウム 2007 論文集, 情報処理学会シンポジウムシリーズ, Vol.2007, No.8, pp.218-223, 2007
[13] 渡辺 晴美, 福井 信二: プロダクトライン開発に向けたコア資産の構築に関する提案, 組込みソフトウェアシンポジウム 2006 論文集, 情報処理学会シンポジウムシリーズ, Vol.2006, No.9, pp.25-28, 2006
[14] 島田 和明, 安部 田章, 組込みソフトウェアの開発者によるプロダクトライン開発の実際, 組込みソフトウェアシンポジウム 2005 論文集, 情報処理学会シンポジウムシリーズ, Vol.2005, No.12, pp.82-89, 2006
[15] 玉井哲雄: ソフトウェア工学の基礎, 岩波書店, 2004
[16] ET ロボコンホームページ
http://www.etrobo.jp/