

ゴール指向要求分析法とジャクソン木に基づく 設計法によるソフトウェア開発法の提案

A Proposal of a Software Development Method by Goal-Oriented Requirents Analysis Method in combination with Software Design Method Based on Jackson Tree

始沢翔太[†]

Shota Shizawa[†]

大谷雄平[‡]

Yuhei Otani[‡]

古宮誠一[‡]

Seichi Komiya[‡]

[†] 芝浦工業大学 デザイン工学部

[‡] 芝浦工業大学大学院 知能ソフトウェア工学研究科

[†] School of Arts and Sciences, Shibaura Institute of Technology.

[‡] Graduate School of Engineering and Science, Shibaura Institute of Technology.

要旨

どのようなソフトウェアを開発するかという要求を抽出する方法として有効なゴール指向要求分析法[1]と、構造化プログラミングを支援する設計法として有効なジャクソン木に基づく設計法の利用を考える。ゴール指向要求分析法は目的樹木による分析法とも呼ばれ、ジャクソン木に似た木構造を利用するので、目的樹木からジャクソン木への変換は容易だと考えられる。本稿では、この両者を連続して使用するソフトウェア開発法を提案する。

1. はじめに

開発するソフトウェアに求められる要求を分析することは、システムを作る上で非常に重要なことである。要求を分析する作業は開発工程の一番初めに行われるため、この工程で誤りが生じると以降の工程にも誤りが含まれたまま作業が行われてしまう。このため、誤りを含まない形で、要求を洩れなく抽出をして行くことがソフトウェア開発では非常に重要である。しかし、抽出し確定した要求を、次工程である設計工程でもそのまま利用して設計作業を行う手法は、未だ提案されていない。そこで本研究では、効率よく要求を抽出することができるゴール指向要求分析法と、構造化されたプログラムを、ジャクソン木を使って効率よく作り出す設計法とを連続して適用することにより、効率よくソフトウェア開発を行う手法を提案する。

2. ゴール指向要求分析法により、O×ゲームに求められる要求の抽出

ゴール指向要求分析とは、ソフトウェア開発するということは開発するソフトウェアによって何らかのゴール（目的）を達成することであるという考えに基づき、開発するソフトウェアに対する要求を抽出する方法である[1]。本稿では、3目並べを行うシステムを例に、その手法を明らかにする。

2.1. ゴール指向要求分析法とは

ゴールは2種類の分解により階層化される。ひとつはAND分解であり、これは分解された部分ゴールがすべて満たされれば、ゴールが満たされることを意味する。もうひとつはOR分解であり、部分ゴールの1つが満たされれば、目標が満たされることを意味する。

2.2. ゴール指向要求分析法を用いた要求抽出作業の方法

ゴール指向要求分析法は最初に最終目標を1つ設定し、AND分解とOR分解を使って、その達成を可能にする幾つかの部分ゴールへ分解する。次に、部分ゴールのそれぞれを最終目標と見なし、幾つかの部分ゴールへ分解する。それ以上分解出来なくなるまでこの操作を繰り返す。目的樹木と呼ばれる木の枝を下方へ段階的に伸ばして行くことで具体的な要求を抽出することができる。機能的な目標でも非機能

的な目標でも部分ゴールへ分解することができる。目標を最終的には手段へ分解することになるため、最初に設定した最終目標を達成するには具体的に何をすればよいか分かるので理解し易い。

ここでは、その具体例として〇×ゲームを作りたいという最終目標を設定した。この目標を達成するにはまず〇×ゲームということで勝敗を決めたいという要求が浮かぶ。次にこの結果を知りたいので表示するという要求も浮かぶ。勝敗を決めるには〇と×を入力し、結果を判断する必要があるため入力と判断の要求も必要となる。最後にこれらを使うための初期条件が必要なため初期条件を作る要求を加える。その結果を図1に示す。

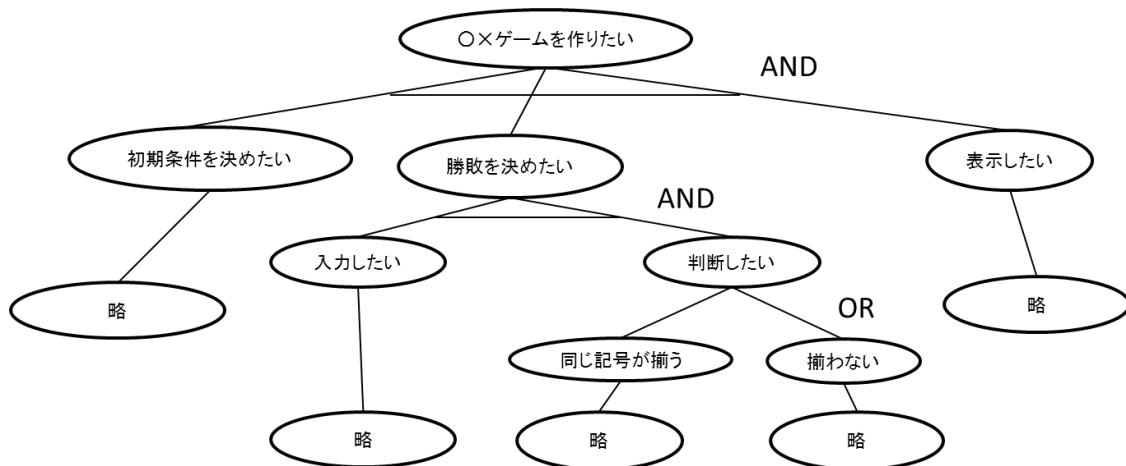


図1 ゴール指向要求分析法での〇×ゲームの要求抽出一部例

2.3. ゴール指向要求分析法の長所・短所

ゴール指向要求分析法を用いることの長所と短所を下記に列挙する。

長所：

- 目的樹木を利用するので、ビジュアルに作業を進めることができる。
- 目的樹木の枝を下方へ段階的に伸ばして行く形で作業を進めるので、必要とあれば枝を下方へ幾らでも伸ばすことができる
- 抽象的な達成目標から具体的な達成目標へと、段階的に達成目標を捉らえることができるので、抽出作業が具体的に行える

短所：

- 上位の目標を達成するために同じ操作の繰り返しが必要な場合、それを表現する表記法がない
- AND 分解によって抽出された複数の手段の間に、それらの適用順序が表現されていない

2.4. ゴール指向要求分析法から Jackson 木への変換

ゴール指向要求分析法も Jackson 木に基づくソフトウェア設計法も木を利用した手法であり、しかも木で表現されている意味が類似している。このため、目的樹木から Jackson 木への変換も容易であると考えた。従って、ゴール指向要求分析法と、Jackson 木に基づくソフトウェア設計法とを連続して適用すれば、シームレスな開発が行えると考えた。

3. Jackson 木への変換による、〇×ゲームのアルゴリズム設計

Dijkstraは、構造化プログラミング[2]と呼ばれる、「接続」「選択」「繰り返し」というプログラムの実行を制御する3つの構造を使えば、どのようなプログラムのアルゴリズムでも構築できることを証明し

た。構造化プログラミングの原理を遵守したプログラムは、プログラムのアルゴリズムを理解し易いので、アルゴリズムの変更作業も容易であり、アルゴリズム・エラーの混入を防ぐ効果があると言われていいる。つまり、プログラムの保守性に優れていると言われていいる。マイケル・ジャクソンは、構造化プログラミングの原理を遵守したプログラムを導くJSP法[3]と呼ばれるソフトウェア設計法を開発した。そこで使われているのがJackson木である。

3.1. Jackson 木とは

Jackson 木は以下の3つの構造を使って階層的な表現ができるプログラム設計図である。

- 1) 接続：各階層の行動は左から右へという順序で処理される。
 - 2) 選択：階層内のどれか1を選択して処理をする。選択に属するノードはノードの右上に○印を付けて表記する。
 - 3) 繰り返し：繰り返して処理する。繰り返しに属するノードはノードの右上に*を付けて表記する。
- ここでは図1で示したものをJackson木に変換したものを図2に示す。

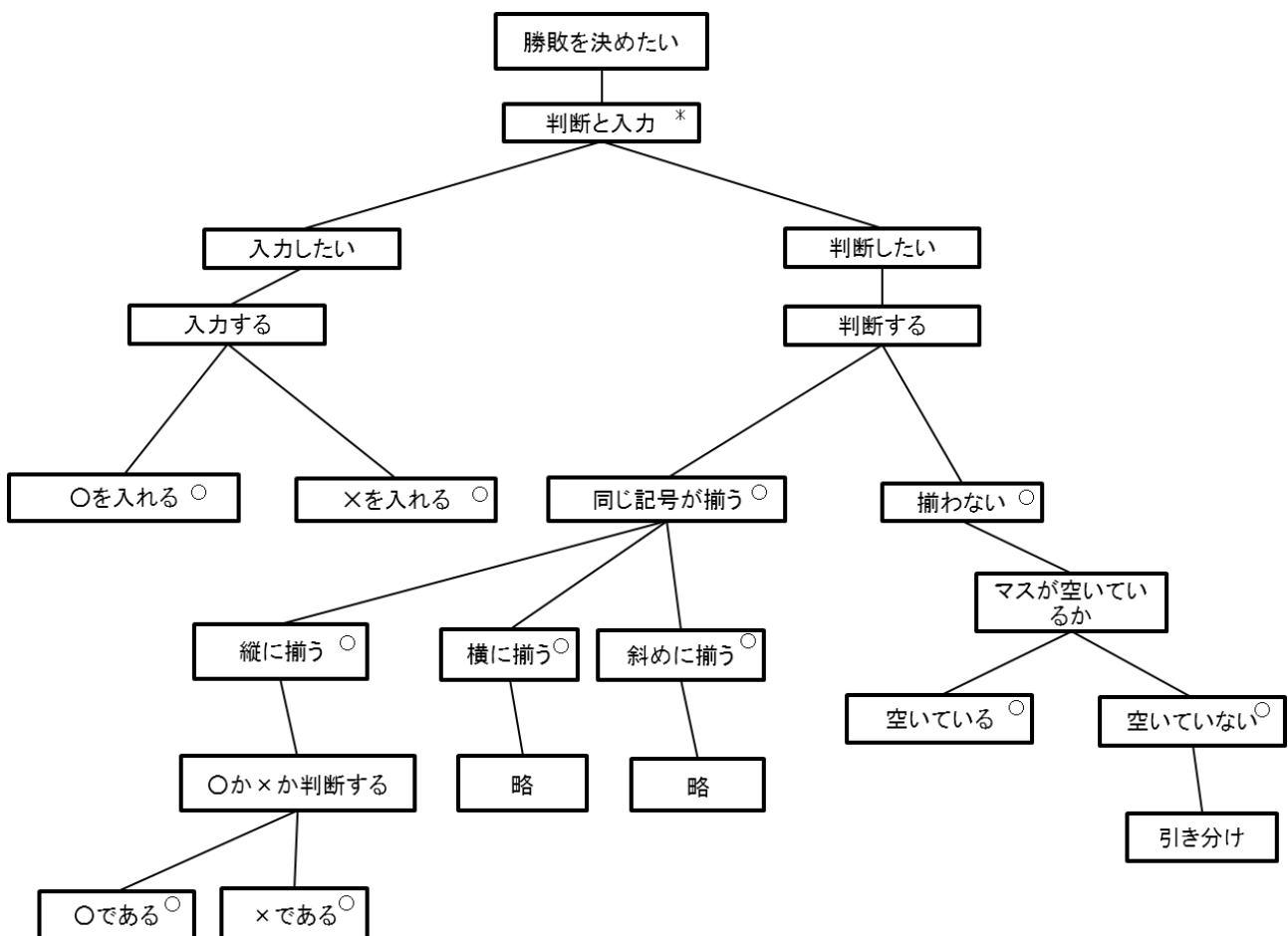


図2 Jackson 木による○×ゲームの要求抽出一部例

3.2. Jackson 木に基づくソフトウェア設計法の利点

Jackson 木を用いたことの利点をここに述べる。

- 同じノードから分解されてできた子ノードの実行順序を、左から右に並べることによって表記するというルールを採用することにより、接続のノードの実行順序を明示的に表現できる
- ゴール指向では表記できなかった繰り返しが表現できる

3.3. Jackson 木とフローチャートとの比較

最も普及しているプログラム設計図として、フローチャートを挙げるができる。フローチャートは流れ図とも呼ばれ、処理の実行順序を表記する。流れ図を使って処理の流れ(実行順序)を、原則として上から下へという順序で表記するものと約束する。また、できるだけ流れ図が左から右へと展開するように表記する。流れ線は互いに交差してもよく、この場合、これらの間には互いに論理的関係はないものと約束する。Jackson 木に基づくソフトウェア設計法では「接続」「選択」「繰り返し」という制御構造ごとに、対応するフローチャートの表記法が用意されている。このため、Jackson 木からフローチャートへと容易に変換できる。ここでは Jackson 木とフローチャートとの比較をすることで Jackson 木の利点について述べる。

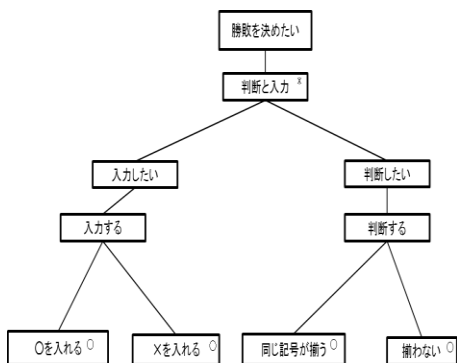


図3 Jackson 木による表現

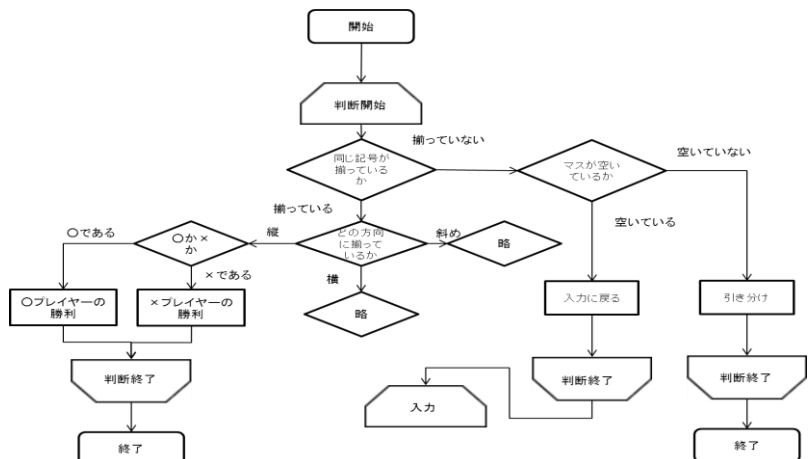


図4 フローチャートによる表現

図3は図2を簡略化して表記したものである。図4は判断するところを詳細化したフローチャートである。この二つを見比べると、フローチャートも Jackson 木も目で追うことで処理の流れを視界的に理解できるメリットがある。しかし、フローチャートは始めから順にアルゴリズムを追っていかないとアルゴリズムを理解できない。また、選択処理や反復処理が理解しにくいという欠点もある。その点では、Jackson 木は、接続処理における実行順序に関する表記法については、Jackson 木における左からの右へという表記ルールと、フローチャートにおける上から下へという表記ルールとが対応するので、アルゴリズムの理解し易さという点では等価であろう。選択処理や反復処理に関する表記法については、Jackson 木による表記法のほうが、フローチャートよりも高い抽象度で表記できるので、アルゴリズムを理解しやすいと考えられる。

4. まとめ

本稿では、ゴール指向要求分析法と、Jackson 木に基づく設計法を連続して適用する、シームレスなソフトウェア開発法を提案した。2章では、ゴール指向要求分析法を使って、開発するソフトウェアに対する要求をどのように抽出するかということ、○×ゲームソフトウェアを例に説明した。3章では、○×ゲームソフトウェアを例に、ゴール指向要求分析法と Jackson 木に基づく設計法を連続して適用するソフトウェア開発法の有用性について述べた。

参考文献

- [1] 山本 修一郎, “～ゴール指向による!!～システム要求管理技法,” ソフト・リサーチ・センター, 2007.
- [2] E. W. Dijkstra, “Structured Programming,” In *Software Engineering Techniques*, B. Randell and J.N. Buxton, (Eds.), NATO Scientific Affairs Division, Brussels, Belgium, pp. 84-88, 1970.
- [3] M. A. Jackson, “Principles of Program Design,” Academic Press, 1975.