Vol.2017-SE-195 No.10
2017/3/12

抽出ルールに基づいた要求記述からのゴールモデルの構築支援

島田 裕紀[1,a)]　中川 博之[2,b)]　小島 英春[2]　土屋 達弘[2]

概要：近年開発されるソフトウェアは大規模・複雑化し，多様な機能を実装する必要があるため，設計にかかるコストは増大し，要求分析時に要求の把握漏れが生じやすくなってきている．これを防ぐための一手段として，要求モデルの効果的な活用が挙げられる．本研究では要求モデルの 1 つであるゴールモデルに着目し，自然言語で書かれた要求記述からゴールモデルの構築を支援する手法を提案する．本手法では，まず要求記述をシステムの機能ごとに分解し，分解した記述ごとに本研究で提案する抽出ルールを適用することでゴールモデルを構築する．その後ゴールモデルのテンプレートに従い，個別に構築したゴールモデルを集約する．また本研究では提案する手法の妥当性を評価するために，ソフトウェア工学分野で知られている 2 つのシステムの要求記述を用いてゴールモデル構築実験を実施した．さらに本研究では，一部の抽出ルールを実装し，ゴールモデル構築の自動化についても検討した．

1. はじめに

近年においてソフトウェアの果たす役割は大きく，同時にソフトウェア開発の規模も増大している．そのため要求記述に書かれている要求，つまりソフトウェアが満たすべき要件の記述は膨大であり，要求をもれなく把握することも困難になる．しかしながら，要求の把握もれが致命的な問題を引き起こす可能性もある．特に医療システムや金融システム，人員運搬システムなど人命や人々の生活に関わるシステムにおいて要求の把握もれはクリティカルな問題となる．Standish グループの Chaos レポートによれば，開発プロジェクトが最終的に失敗する原因で最も多かったものは，抽出すべき要求の不完全性である [1]．そのため要求分析時には要求モデルを構築し，要求記述に書かれた要求を視覚化，および検証することで整合性のとれた要求の網羅的な獲得が試みられている．現在要求分析工学においては UML やユースケース [2] など，様々な要求モデルが提案されている．

本研究では要求モデルの 1 つであるゴールモデルに着目し，ゴールモデルを構築する手順と要求抽出のルールを提案する．また本手法の妥当性を評価するため，ソフトウェア工学分野で知られている 2 つのシステムの要求記述を対象として実験を行った．その結果抽出すべき要求の多くを抽出し，さらに構造化できていることを確認できた．またソフトウェアの大規模化による，構築されるモデルの大規模化，複雑化は避けられない．そこでモデル構築の省力化を図るため，自然言語で書かれた要求記述を入力として自然言語処理により自動でモデルを構築できるかどうかを検討するため，ルールを自動で適用するプログラムを作成した．その結果実装したルールについては，要求抽出を行い構造化を行うことができた．

本論文の構成は以下のとおりである．2 章では背景としてソフトウェア開発時に行う要求分析とその分析モデルであるゴールモデルについて説明し，また関連研究についても述べる．3 章ではゴールモデル構築の手順と要求抽出のためのルールについて説明する．4 章では 3 章で述べたルールの妥当性を評価するための実験について述べる．5 章では 3 章で述べたルール適用の自動化について説明する．最後に 6 章で本研究の結論と今後の課題について述べる．

2. 背景

2.1 要求分析

ソフトウェアを開発する際には，目的とするソフトウェアがどのような要求を満たしていなければならないのかを設計段階で分析する必要がある．これを要求分析といい，ステークホルダー間の認識のずれをなくすために重要なプ

[1] 大阪大学基礎工学部情報科学科，豊中市
Department of Information and Computer Sciences, School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan
[2] 大阪大学大学院情報科学研究科，吹田市
Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan
a) h-simada@ist.osaka-u.ac.jp
b) nakagawa@ist.osaka-u.ac.jp

ロセスである．要求分析が不十分であると，開発段階で無駄な手戻りが発生してしまう，あるいは思わぬバグを生む可能性もある．ステークホルダー間での認識を共有するために，要求モデルが用いられる．モデルを用いて要求を構造化することで，要求のもれを視覚的にわかりやすくする利点がある．

2.2 ゴールモデル

要求分析に用いるモデルの 1 つとしてゴール指向要求モデルがある．ゴールモデルとは，ソフトウェアが満たすべき 1 つの状態（要求）を 1 つのゴールとし，達成されるべき最終目的のゴールとその前提として達成されるべきゴール（サブゴール）を木構造で記述したものである．親ゴールと子ゴールの関係は and 関係あるいは or 関係で表される．親ゴールと子ゴールが and 関係で接続されている場合は，全ての子ゴールが達成されていれば親ゴールも達成されたとみなされる．親ゴールと子ゴールが or 関係で接続されている場合は，1 つ以上の子ゴールが達成されれば親ゴールも達成されたとみなされる．最終目的のゴールが達成されたとき，ソフトウェアは目的の要求を満たしていると判断できる．提案されているゴール指向要求モデルとして，KAOS[3]，i*[4]，NFR[5] が挙げられる．本研究では，あらかじめ要求記述が存在し，システムが責務を持つべき要求を網羅的に構造化するという観点から，特に KAOS ゴールモデルを想定したゴール記述を用いる．

2.3 関連研究

2.3.1 モデル構築とその自動化

本研究では，ゴールモデルの構築手法を提案しており，さらにルール適用の自動化についても議論する．Rahimi ら [6] は，自然言語で書かれた要求記述からソフトウェアの品質要求を抽出し，ゴールモデルとしての構造化及び視覚化を自動で行う手法を提案している．また Weber-Jahnke ら [7] は，自然言語で書かれた文書からセキュリティゴールモデルを構築する手法を提案し，ツールとして実装している．しかしながら，これらの研究では機能要求ではなく非機能要求の抽出のみにとどまっている．本研究では機能要求，非機能要求をともに抽出する手法を提案している．

またゴールモデル以外にも，自然言語で書かれた記述からモデルを自動生成する手法を提案する研究がある．Arora ら [8] は，自然言語で書かれた要求記述からドメインモデルを自動で抽出する手法を提案している．Zhai ら [9] は Java API ライブラリを自動でモデル化し，簡易な Java コードを生成する手法を提案している．Deeptimahanti ら [10] は，自然言語で書かれた要求記述から，自動で UML モデルを構築するツールを提案している．Thakur ら [11] は，ユースケース記述からシーケンス図を自動で生成する手法を提案している．本研究では，システムが責務を持つべき要求を網羅的に構造化するという観点からゴールモデルの構築手法を扱う．

2.3.2 ゴールモデルを再利用したソフトウェア開発

ソフトウェア開発プロセスの改善にゴールモデルを利用する手法は数多く提案されている．中村ら [12] は，複数のゴールモデルから共通しているゴールを判別する手法を提案し，中川ら [13] は，制御モデルとして知られる Control loop をゴールモデル内で同定し，ソフトウェア進化時の支援を行う手法を提案している．また Uno ら [14] は複数のゴールモデルを統合して 1 つのゴールモデルを構築し，そこからフィーチャモデルを構築してゴールモデルとの関連性を示す手法を提案し，さらに根岸ら [15] は，開発するソフトウェアが法律に適合しているかどうかをゴールモデルの記述と法律の記述のマッチングを行うことで判断し，法律に適合するようゴールモデルを修正する手法を提案している．本研究では，2.3.1 で述べた関連研究と同様にモデルを新規に構築する手法を扱う．

3. ゴールモデル構築手順と要求抽出ルール

本章では，ゴールモデルを構築するための手順と要求抽出のためのルールについて説明する．

3.1 ゴールモデル構築手順

手順は大きく分けて次の 3 つのステップからなる．各ステップについて以降説明を行う．

Step1. 要求記述をソフトウェアの機能ごとに分解する

最初のステップとして，要求記述をソフトウェアの機能ごとの記述の集合に分解する．その際に要求に関係しないと判断できる記述は除外する．特に要求記述が章や節立てで記述されている場合は，章や節，段落ごとの記述に分解する．

Step2. 各記述についてゴールモデルを構築する

次に各記述集合をもとに，ゴール記述を取り出し構造化を行う．この際に 3.2 で述べる “抽出ルール”に従い，ゴールモデルの構築を行う．

Step3. 各ゴールモデルをテンプレートに合致するよう集約する

図 1 は最終的な出力となるゴールモデルのテンプレートである．最終ゴールとしてシステムの達成目標，その子ゴールとして “機能要求”，“非機能要求”が and 関係で接続されている．非機能要求については要求工学についての教科書 [16] で紹介されているカテゴリ分類を用いてゴールを詳細化する．前ステップで構築された各ゴールモデルを機能要求と非機能要求に分類して集約し，最終的に構築されるゴールモデルとする．

3.2 ゴールモデル抽出のルール

前節で述べたステップの 2 つ目である断片的なゴールモデルの構築においては，要求記述から要求同士の関係を判

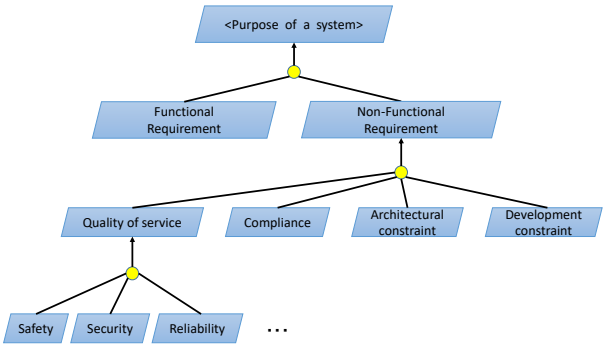

図 1　ゴールモデルのテンプレート

断し構造化することが必要となる．本節では要求（ゴール）を構造化するための抽出ルールを提案する．抽出ルールは大きく次の 2 つに分けることができる．本節ではこれらのルールから一部を抜粋して説明する．

- キーワードを用いるルール
  - 指示語の置換
  - 節構造の抽出
  - 具体例の抽出
  - 特定の場合の抽出
  - トピックの区切りの抽出
  - 抽出した記述の整形
- その他のルール
  - 通常処理と例外処理の集約
  - 列挙記述の扱い
  - 括弧部分の扱い
  - “,” がない場合の名詞の区切りの扱い
  - 親ゴール記述の設定

3.2.1　キーワードを用いるルール

このルール群は，多くの要求記述に含まれる単語に対して，ゴール記述としての文章の整形やゴール記述の抽出及びゴールモデルとしての構造化を行うものである．本論文では，“指示語の置換” と “節構造の抽出” というルールについて説明する．

- **指示語の置換**：要求記述からゴール記述を抽出する際に文脈が失われるため，ゴール記述のみから this，that，he などの指示語が指す内容を解釈することが困難となる．そのためあらかじめ指示語の内容で置換を行っておく．
- **節構造の抽出**：次に挙げるキーワードが見つかったときは，文中の節を取り出して構造化することができる．
  - **after，before**：これらのキーワードは達成される状態の前後関係を表す表現であり，文中には大きく 2 つの節を包含していると考えられる．そのためこの 2 つの節をそれぞれゴール記述として左からゴールが達成される順番に並べ，and 関係で親ゴールと接続する．
  - **then**：then が含まれる文は前文を受けた内容が書かれていると考えられる．そのため前文から抽出したゴールと同じ階層の子ゴールとして親ゴールと接続する．
  - **to-不定詞，in order to，subject to，thanks to，by …ing**：これらのキーワードが含まれる文は目的と手段を表す表現であり，大きく 2 つの節を包含していると考えられる．そのためこの 2 つの節をそれぞれゴール記述とし，“thanks to” や “by …ing” の場合はキーワードの直後の節を子ゴール，残りの節を親ゴールとし，その他の場合は逆にキーワードの直後の節を親ゴール，残りの節をその子ゴールとして接続する．
  - **“and”，“or”**：これらのキーワードは複数の節を並列に接続するものである．そのため各節をゴール記述とし，and 関係または or 関係で親ゴールと接続する．

3.2.2　その他のルール

このルール群は，キーワードに対する処理ではなく要求記述の意味内容を把握した上で，要求の構造化を行うためのルールである．本論文では，“通常処理と例外処理の集約”，および “列挙記述の扱い” というルールについて説明する．

- **通常処理と例外処理の集約**：要求記述において，通常処理と例外処理の記述が検出された場合はそれぞれのゴールモデルを and 関係で親ゴールと接続する．
- **列挙記述の扱い**：箇条書きや文中での列挙が検出された場合は，各項目についてゴールモデルを構築し，and 関係または or 関係で親ゴールと接続する．

4.　評価実験

4.1　評価方法

本研究では次の 2 点に着目して提案手法を評価した．

- **要求が抽出できているか**
  要求工学についての教科書 [16] に掲載されている会議調整システムの要求記述に対し，手作業でルールを適用することでゴールモデル M_1 を構築した．次にこの教科書で構築例として示されているゴールモデル M_2 と比較し，必要な要求がどの程度抽出できているかを確認した．
- **ルール適用により構造化ができているか**
  本研究では，構造化の指標として同教科書で良い構造化として紹介されている構造パターンを用いた．評価対象として要求工学分野でよく知られている ATM システムの要求記述に対して手動でルールを適用し，ゴールモデルを構築した．会議調整システムと ATM システムのゴールモデルの中に構造パターンがどれだけ含まれているかを調べ，構造化がどの程度できているかを確認した．

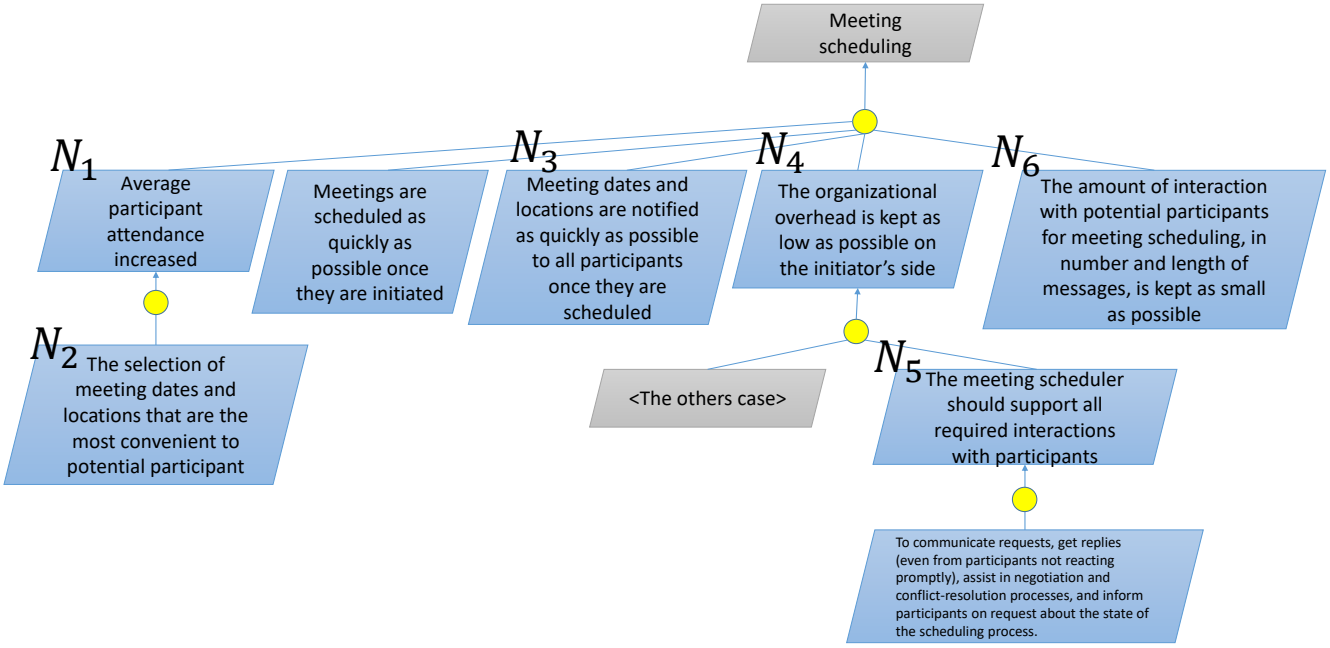

図 2　ルールを適用し構築した会議調整システムのゴールモデル M_1（一部）

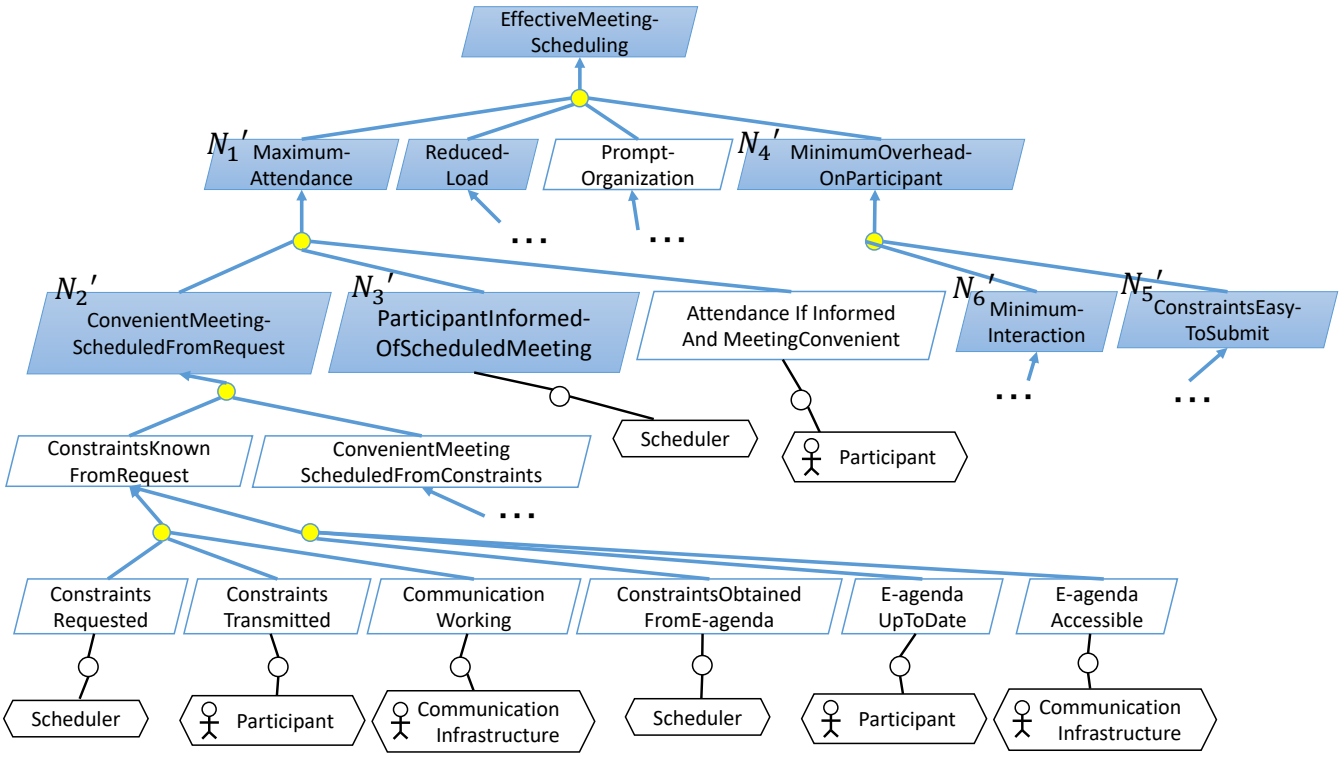

図 3　ゴールモデル構築例 M_2'

4.2　実験結果

まず，手作業で本手法を適用することで構築した会議調整システムのゴールモデル M_1 の一部を図 2 に示す．構築したモデルの各ゴールはゴール記述の種類によって次のよ

表 1　対応する要求記述

ノード番号	手作業で構築したゴールモデル (M_1)	ゴールモデル構築例 (M_2')
1	Average participant attendance increased	MaximumAttendance
2	The selection of meeting dates and locations that are the most convenient to potential participant	ConvenientMeetingScheduledFromRequest
3	Meeting dates and locations are notified as quickly as possible to all participants once they are scheduled	ParticipantInformedOfScheduledMeeting
4	The organizational overhead is kept as low as possible on the initiator's side	MinimumOverheadOnParticipant
5	The meeting scheduler should support all required interactions with participants	ConstraintsEasyToSubmit
6	The amount of interaction with potential participants for meeting scheduling, in number and length of messages, is kept as small as possible	MinimumInteraction

うに色分けしている．

- 灰色のゴール：分析者が独自で設定した記述
- 青色のゴール：上記以外で要求記述から抽出した記述

次に，書籍で構築例として示されているゴールモデル M_2 の要求が M_1 で抽出できているかどうかでゴールの色分けを行い，ゴールモデル M_2' とした．ゴールモデル M_2' を図 3 に示す．青色のゴールが M_1 で抽出できたゴール，白色のゴールが M_1 で抽出できなかったゴールである．

さらに ATM システムの要求記述に対しても本手法を適用し，ゴールモデル M_3 を作成した．

4.3 考察

4.3.1 要求の抽出

ゴールモデル M_1 で抽出された要求がゴールモデル M_2' のゴールに対応していることを示すため，それぞれに N_1 から N_6，N_1' から N_6' で対応するラベルを付けている．それぞれのゴール記述は次の表 1 のとおりである．対応する記述を比較すると必要な要求が抽出できていることがわかる．

一方で白色のゴールについては，要求抽出を行うことができなかった．その理由として，その要求が要求記述に含まれていない要求であったことや，システムの機能ではなくアクターの動作，いわゆる期待を表すゴールであったことが挙げられる．これらのゴールについては，青色のゴールの分解を進めることで抽出できる．なお，ゴールモデル M_2' では一部ゴールの記述が省略されていたが，これらのゴールについても同様に分解を行うことで抽出が可能である．

4.3.2 要求の構造化

要求工学の書籍 [16] では，以下に効果的なゴールモデルの構造パターンが紹介されている．

- **The milestone-driven refinement pattern** : あるゴールに対して，開始状態 A と最終状態 B の間に中間状態 C があり，そのゴール（$A \to B$）が 2 つの子ゴール（$A \to C$，$C \to B$）に分解されている．
- **The decomposition-by-case pattern** : あるゴールが達成されるまでの流れが複数存在する場合，各流れがそれぞれ子ゴールとして分解されている．
- **The guard-introduction pattern** : あるゴールに対して，開始状態 A と最終状態 B があり，そのゴール ($A \to B$) が達成される時に最終状態に至るための条件であるガード G が存在する場合，ガードがその子ゴール ($A \to G$) となり，加えて 2 つの子ゴール ($(A \wedge G) \to B$，$\neg B \to A$) に分解されている．
- **The divide-and-conquer pattern** : あるゴールが 2 つの最終状態を持つ場合，それぞれの状態が子ゴールとして分解されている．
- **The unmonitorability-driven refinement pattern** : エージェントがあるゴールに責任を持っていて，エージェントがそのゴールの中で監視できない状態が存在する場合，状態監視が可能な別のエージェントに責任を割り当てるようゴール分解されている．
- **The uncontrollability-driven refinement pattern** : エージェントがあるゴールに責任を持っていて，エージェントがそのゴールの中で制御できない状態が存在する場合，制御監視が可能な別のエージェントに責任を割り当てるようゴール分解されている．

本手法を適用して作成した会議調整システムと ATM システムのゴールモデル (M_1, M_3) にこれらの構造パターンが含まれているかどうかを調べた．結果を表 2 に示す．

表 2　構造パターンの有無

構造パターン	M_1	M_3
The milestone-driven refinement pattern	×	○
The decomposition-by-case pattern	○	○
The guard-introduction pattern	×	×
The divide-and-conquer pattern	×	○
The unmonitorability-driven refinement pattern	×	×
The uncontrollability-driven refinement pattern	×	×

The milestone-driven refinement pattern，The decomposition-by-case pattern，The divide-and-conquer pattern については，どちらか一方か双方のシステムで構造が確認できた．これらはルールの適用によって構造化できたことによるものである．一方で，残る構造パターンについては，どちらのシステムでも確認できなかった．これは，これらの構造を構築するにはドメイン知識やエージェントの詳細な設定が必要となるためである．

4.3.3 自動化できる範囲の検討

本研究では，キーワードを用いたルールとその他のルールという大きく分けて 2 つのルールを提案している．キーワードを用いたルールについては，要求記述中に含まれるキーワードを検出して対応する処理を行う単純なものであるため，ルールの適用を自動化することは可能である．一方でその他のルールについては，分析者が要求記述の意味内容を把握する必要があるため，適用の自動化は困難であると考えられる．図 2 のモデルはゴール記述の種類によって色分けを行っている．各種ゴールについて，灰色のゴールについては，分析者が要求記述の意味内容を把握している必要があるため，ゴール記述を自動で設定することは困難である．ただし，記述のないゴールを自動的に作成しておき，モデルの自動生成を終えた後にゴール記述を手作業で設定するよう分析者に促すことでゴールモデル構築の支援とすることができる．一方青色のゴールについては，要求記述にルールを適用する，あるいはそのまま抽出してできるゴール記述である．そのため自動での構築が可能である．

5. ゴールモデル構築の自動化について

本研究ではゴールモデル構築の省力化のため，ゴールモデル構築の自動化についても検討した．

5.1 自然言語処理

本研究におけるゴールモデル構築のプロセスは自然言語で書かれた要求記述を入力とする．そのため自動でゴールモデル構築を行うためには，自然言語処理を行う必要がある．これを行うことにより，要求記述を人間が読解可能な情報の単位に分割することができる．本研究では既存の自然言語処理ツールである Stanford CoreNLP[17] を用いることにより目的の達成を試みた．

5.1.1 Stanford CoreNLP

Stanford CoreNLP は自然言語処理ツールであり，構文解析や単語の見出し語化，指示語内容特定など様々な処理を行うことができる．本研究で 3.2.1 のキーワードを用いたルールの一部を適用するよう Java 言語により実装し，ゴールモデルの自動構築を試みた．

5.1.2 処理の流れ

まず入力を自然言語で書かれた要求記述とする．この文書の各文について構文解析を行い，ルールを適用しながら中間モデルを生成する．さらにこのモデルからルールに従って変換を行い，本プログラムの出力となるゴールモデルを構築する．なお，本プログラムの出力であるゴールモデルは dot 言語 [18] で書かれたスクリプトとして出力される．視覚化のためには dot ファイルから画像形式ファイルに変換する Graphviz[19] を用いる必要がある．

5.1.3 実装したルール

本研究で実装したルールを以下に列挙する．

- 節構造の抽出 (after，before，and，or)
- 具体例の抽出（for example，for instance，such as）
- 特定の場合の抽出（in particular，when，if）

5.2 実行結果

5.1.3 で列挙したルールに基づき自動でゴールモデルを生成した結果を図 4 に示す．出力されたゴールモデルは横に長いため，ルールが適用され構造化された部分を拡大して右下に示している．左下のゴールモデルは，図 2 で示した手作業でルールを適用し構築したゴールモデルの一部である．

5.3 考察

5.3.1 要求文の構造化

図 4 で示しているゴールモデルの一部では，特定の場合の抽出が自動で適用できている．一方で具体例の抽出が適用できていないのは，具体例の適用範囲が明確にできなかったことによるものである．また，Stanford CoreNLP の構文解析の誤りによるゴール記述抽出の誤りも存在した．

5.3.2 ゴールモデルの整形

今回作成したプログラムは要求記述の文単位で要求の構造化を行い，それらを and 関係により集約している．そのためゴールモデルが非常に横に長くなってしまっている．この問題を解決するためには，類似するゴール記述の特定や文同士の関係を把握することなどにより，モデルを整形する必要がある．

5.3.3 ゴール記述の整形

本研究で抽出されるゴール記述は，要求記述における文脈から切り離された記述となっている．例えば “and” が含まれる文から生成されたゴール記述は要求記述の文脈から切り離され，主語が欠落する場合がある．ゴール記述として簡潔かつ解釈可能なものとするために，記述の補完を自動的に行う仕組みを導入する必要がある．

5.3.4 ルールの網羅性

今回作成したプログラムでは，未実装のルールが多く存在する．これらを実装することは今後の課題であるが，実装の難しさが現時点の問題点として挙げられる．例えば “節構造の取り出し” ルールのキーワードである to-不定詞において，ルールの適用対象となる文章は “…するために，…する” という目的と手段を表すような，to-不定詞の副詞的用法が用いられているものである．この to-不定

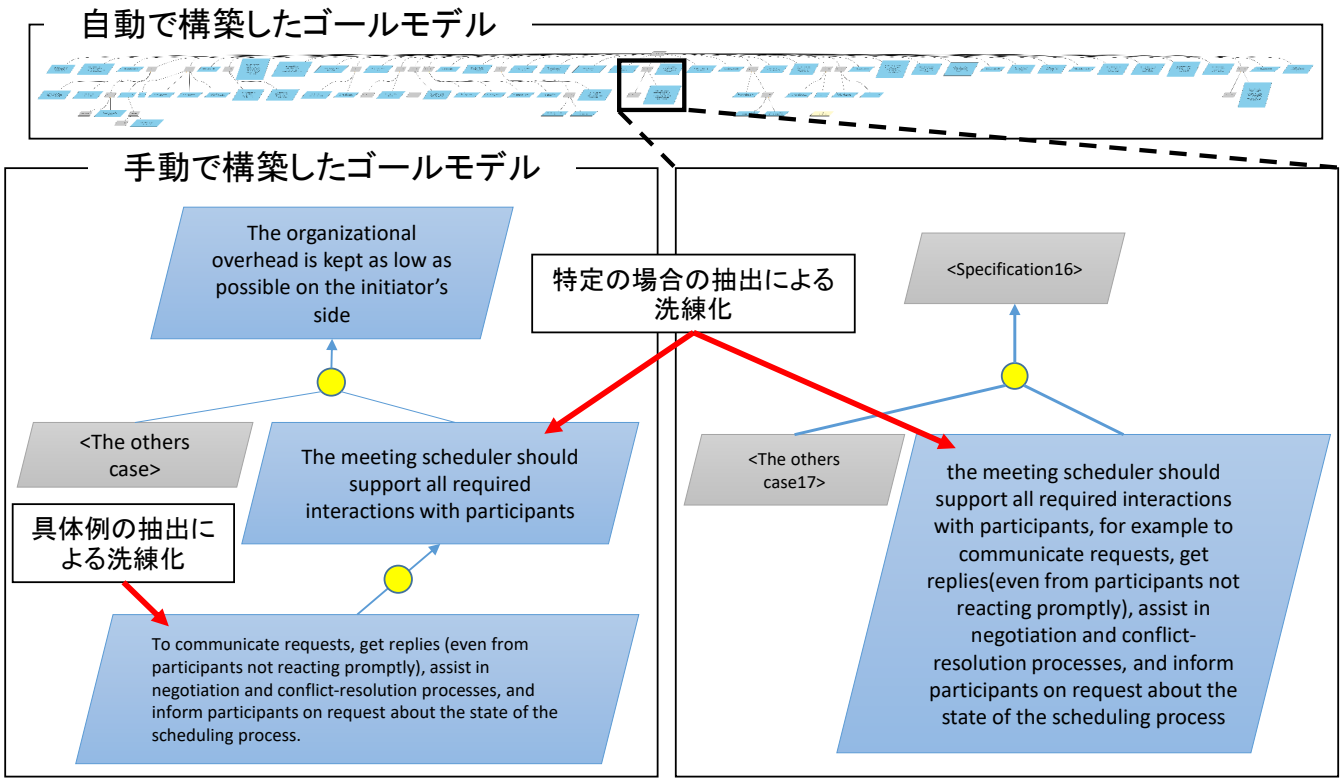

図 4　上：自動で構築したゴールモデル，左下：手動で構築したゴールモデル（一部），
右下：自動で構築したゴールモデル（一部）

詞の用法の判別が Stanford CoreNLP ではできていないのが現状である．またキーワードを用いない “その他のルール” についても，要求記述の意味内容を把握できなければ適用できないために実装が難しい．また今回作成したプログラムは 3 章で述べたルールの一部の実装のみにとどまっている．

6. 結論

6.1 本研究について

本研究では要求分析に用いられるゴールモデルの構築を支援するため，自然言語で書かれた要求記述から要求を取り出して構造化する手順と要求抽出のための抽出ルールを提案した．提案したルールの妥当性を実験するため，要求工学の教科書で扱われていた会議調整システムについて，手作業で抽出ルールを適用しゴールモデルを構築した．これと同書籍で構築例として示されていたゴールモデルとの比較を行い，必要な要求が抽出できているかどうかを調べた．その結果アクターによる操作の期待を表すゴールや要求記述にないゴールを除いては，ほとんどの要求を抽出できた．また書籍で紹介されている構造パターンが構築したゴールモデルにどれだけ含まれているかを調べ，要求の構造化ができているかを調べた．その結果抽出ルールの適用によっていくつかの構造パターンが発見された．さらにゴールモデル構築の省力化を図るため，抽出ルール適用の自動化が可能であるかどうかプログラムを作成して実験を行った．その結果，手動で構築したゴールモデルにおいて適用されていたルールが自動でも適用できていることが確認できた．

6.2 今後の課題

本研究では 2 つのシステムの要求記述に対してルール適用を行ったが，より厳密な評価を行うためにはルールの適用例を増やす必要がある．特にキーワードを用いるルールについては，本研究では見つけることができなかった要求構造化の基準となるキーワードが存在する可能性もあるため，適用例を増やすことは今後の課題である．さらに本手法では要求記述からの要求抽出のみを扱う．本研究で想定している KAOS ゴールモデルは単一のエージェントに割り当てる責務を明確にすることができる [20] という利点を持つが，現段階ではゴール分解が不十分となる可能性があり，エージェントに割り当てる責務を明確化できない．そのためより細かくゴール分解を行うことが今後の課題である．またゴールモデル構築の省力化について，現状では文単位での構文解析及び要求の構造化を行っている．そのためゴールの集約が不十分であり，モデルとしての実用性は低い．このことからゴール記述やゴールモデル全体の整形，残る抽出ルールの実装が必要である．さらに本研究で実装したのは 3.1 で述べた手順の一部である．本手法を自

動化するためにはその他のステップの実装も必須となる．本研究における課題は多く残っているが，今後より分析可能性の高いゴールモデルを構築するための研究の礎となると考える．

参考文献

[1] The Standish Group: The Standish Group Report Chaos (1995).
[2] Rosenberg, D. and Stephens, M.: *Use Case Driven Object Modeling with UML : Theory and Practice*, Apress (2007).
[3] Dardenne, A., van Lamsweerde, A. and Fickas, S.: Goal-directed Requirements Acquisition, *SCIENCE OF COMPUTER PROGRAMMING*, pp. 3–50 (1993).
[4] Yu, E. S. K.: Towards modelling and reasoning support for early-phase requirements engineering, *Proceedings of the Third IEEE International Symposium on Requirements Engineering, 1997.*, pp. 226–235 (1997).
[5] Mylopoulos, J., Chung, L. and Nixon, B.: Representing and using nonfunctional requirements: a process-oriented approach, *IEEE Transactions on Software Engineering*, Vol. 18, No. 6, pp. 483–497 (1992).
[6] Rahimi, M., Mirakhorli, M. and Cleland-Huang, J.: Automated extraction and visualization of quality concerns from requirements specifications, *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pp. 253–262 (2014).
[7] Weber-Jahnke, J. H. and Onabajo, A.: Mining and analysing security goal models in health information systems, *2009 ICSE Workshop on Software Engineering in Health Care*, pp. 42–52 (2009).
[8] Arora, C., Sabetzadeh, M., Briand, L. and Zimmer, F.: Extracting Domain Models from Natural-language Requirements: Approach and Industrial Evaluation, *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, MODELS '16, New York, NY, USA, ACM, pp. 250–260 (2016).
[9] Zhai, J., Huang, J., Ma, S., Zhang, X., Tan, L., Zhao, J. and Qin, F.: Automatic Model Generation from Documentation for Java API Functions, *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, New York, NY, USA, ACM, pp. 380–391 (2016).
[10] Deeptimahanti, D. K. and Babar, M. A.: An Automated Tool for Generating UML Models from Natural Language Requirements, *2009 IEEE/ACM International Conference on Automated Software Engineering*, pp. 680–682 (2009).
[11] Thakur, J. S. and Gupta, A.: Automatic Generation of Sequence Diagram from Use Case Specification, *Proceedings of the 7th India Software Engineering Conference*, ISEC '14, New York, NY, USA, ACM, pp. 20:1–20:6 (2014).
[12] 中村祐貴，本田耕三，中川博之，田原康之，大須賀昭彦：ソフトウェア再利用に向けた共通ゴール判別手法の提案，コンピュータソフトウェア， Vol. 31, No. 2, pp. 67–83 (2014).
[13] 中川博之，大須賀昭彦，本位田真一：ゴール指向要求記述の整形に基づいたソフトウェアシステム進化手法，情報処理学会論文誌，Vol. 53, No. 10, pp. 2328–2344 (2012).
[14] Uno, K., Hayashi, S. and Saeki, M.: Constructing Feature Models Using Goal-Oriented Analysis, *2009 Ninth International Conference on Quality Software*, pp. 412–417 (2009).
[15] 根岸由，林晋平，佐伯元司：法律に適合した要求獲得のためのゴールモデル作成支援，技術報告 22，東京工業大学 (2016).
[16] van Lamsweerde, A.: *Requirements Engineering From System Goals to UML Models to Software Specifications*, Wiley (2009).
[17] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J. and McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit, *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60 (2014).
[18] The DOT Language. `http://www.graphviz.org/content/dot-language`.
[19] Graphviz. `http://www.graphviz.org/`.
[20] Letier, E.: Reasoning about Agents in Goal-Oriented Requirements Engineering, PhD Thesis, University of Louvain (2001).