

# 要求獲得会議を分析することによるユーザ指向要求獲得法

土井 晃一<sup>†</sup> 蓬萊 尚幸<sup>††</sup> 渡部 勇<sup>†††</sup>  
片山 佳則<sup>†††</sup> 園部 正幸

我々は USP ( User-oriented System Planning method ) 方法論という会議における要求獲得方法論を開発した。本方法論では、顧客と開発者の要求を洗練し、網羅的に獲得し、要求間の競合を解消する手助けをすることを目的とする。USP 方法論は、会議そのものを支援するオンライン法と、会議を分析することによって会議を支援するオフライン法からなる。USP 方法論による要求獲得では、オンライン法とオフライン法を交互に行うことによって要求を獲得していく。オフライン法では、機能分析、シナリオ分析、品質分析、関心事項分析という 4 つの分析を行う。このようにオフライン法は、マルチビューポイントで要求を獲得し、分析する。

## User-oriented Requirements Capturing Method in Analyzing Requirements Capturing Meeting

KOUICHI DOI,<sup>†</sup> HISAYUKI HORAI,<sup>††</sup> ISAMU WATANABE,<sup>†††</sup>  
YOSHINORI KATAYAMA<sup>†††</sup> and MASAYUKI SONOBE

We propose a requirements capturing method based on meetings, called User-oriented System Planning Method (USP Method), which helps the clients and the developers to develop and refine requirements and to capture them exhaustively. USP Method consists of Online Method which supports a meeting and Offline Method which analyses a meetings. Requirements Process by USP Method consists of an evolutionary and spiral series of meeting and analysis. Offline Method is used to analyse requirements from multi viewpoints through Functional Analysis, Scenario Analysis, Quality Analysis and Interest Analysis.

### 1. はじめに

顧客が満足するソフトウェアシステムを提供するためには、ソフトウェアプロセスの早い時点で要求を網羅的に正確に獲得することが重要である<sup>1)</sup>。しかし、要求は要求の形で直接提示されることは少ないので、要求を獲得することは困難である<sup>2)</sup>。たとえば、顧客は要求獲得プロセスの最初に要求を要求の形で認識していることは少ないので、要求獲得プロセスを通して、潜在的な要求を引き出し、要求のあいまい性を排除し、要求を明確化する必要がある。次に、顧客の全員の要求が明確になったときも、通常、それらの要求の間には競合があるので、要求獲得プロセスを通して、お互いに了解をとりながら、要求の競合を解消してい

かなければならない。要求獲得プロセスは基本的に顧客自身にしかできないプロセスである。要求獲得プロセスは顧客を助けることによって、要求を獲得し、開発者に要求を手渡すプロセスである。そこで我々は、顧客と開発者の双方が参加する要求獲得会議での要求獲得法、すなわち、USP 方法論 ( User-oriented System Planning Method ) を構築している。本論文では、USP 方法論の中のオフライン法について述べる。

このように、要求は、明確化されたり、あいまい性が排除されたりして、要求獲得の各段階によって異なる。また、要求には顧客の要求と開発者の要求がある。本論文で単に「要求」と書いたときは、ある段階での顧客の要求と開発者の要求の和集合を「要求」と呼ぶことにする。

以下、2章では、USP 方法論について概説する。USP 方法論は要求獲得会議自体を支援するオンライン法と、要求獲得会議を分析するオフライン法とからなる。USP 方法論では、オンライン法とオフライン法を交互に行うことによって要求獲得プロセスを進める。本論文では、オフライン法を主として解説する。3章で

<sup>†</sup> 奈良先端科学技術大学院大学  
Nara Institute of Science and Technology

<sup>††</sup> セレスター・レキシコ・サイエンス株式会社  
Celestar Lexico-Sciences Inc.

<sup>†††</sup> 株式会社富士通研究所  
Fujitsu Laboratories Ltd.

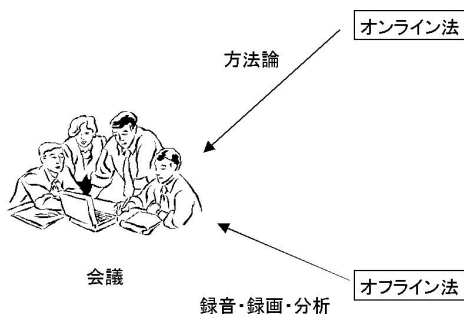


図 1 USP 要求獲得方法論

Fig. 1 USP requirements capturing method.

は、そのオフライン法について解説する。オフライン法では、機能分析、シナリオ分析、品質分析、関心事項分析を通して、マルチビューポイントにより要求を分析する。4章から8章では、オフライン法でのそれぞれの分析について説明する。9章では、オフライン法で使うCASE TOOLについて解説する。

## 2. ユーザ指向要求獲得法

本章では、USP方法論について概説する。前章で顧客のために要求を洗練し、開発者のために要求を獲得することを支援する必要性について述べた。このような方法論では、顧客や開発者の間のコミュニケーションが重要である<sup>3)</sup>。コミュニケーションに関しては、適切な方法(ツール)を選択しなければならない。コミュニケーション・ツールには、インタビュー、質疑応答、会議といったいろいろな方法がある。会議には、陳述、提案、議論、質問、応答、交渉、決定、再確認などいろいろな種類のインタラクションがある。会議はこのようにいろいろなインタラクションを含み、まさに要求を洗練するツールである。そこで、我々は会議をUSP方法論のコミュニケーション・ツールとして選択した。より実際の理由としては、現実の要求獲得プロセスでは、会議が最も頻繁に用いられる。しかし、会議を支援する方法論が少ないからである。

図1にUSP方法論の概略を示す。USP方法論は会議支援法と会議分析法からなる。以下、会議支援法のことを「オンライン法」と呼ぶ。オンライン法は、会議を効果的に効率良く進めるために会議進行中に会議参加者に直接働きかける方法である。一方、会議分析法のことを「オフライン法」と呼ぶ。オフライン法では、分析は基本的に会議が終了した後になされ、会議参加者とは直接には接触しない方法である。

図2はオンライン法とオフライン法を交互に使う様子を示したものである。オンライン法がオフライン

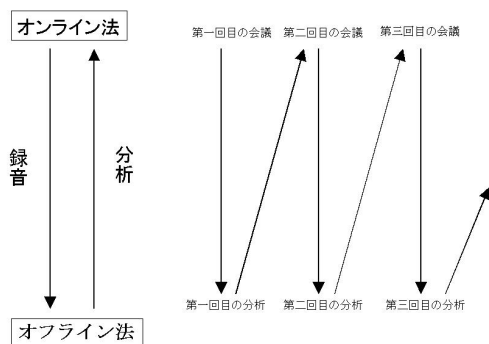


図 2 オンライン法とオフライン法

Fig. 2 Online Method and Offline Method.

法へ分析の材料を提供するのに対して、オフライン法で分析された結果は会議参加者に分析結果を示すことによって、オンライン法で効果的に使われる。こうした、オンライン法、オフライン法の繰返しによって要求は質、量ともに洗練される。

現状では、オンライン法では、オフライン法の結果を使うなどごくわずかのことしかしていない。本章では、将来的な可能性としてオンライン法との関係について説明した。本論文では、オフライン法について説明する。

## 3. USP オフライン法

要求獲得プロセスにおいて、会議は便利なツールである。しかし、残念ながら、現状では重要な内容が記録されずに失われてしまう。会議後には1,2ページの議事録が残るだけというのが現状である。そのような議事録では、通常、決定事項だけが記載されている。会議では、代替案や決定の理由など、下流の作業でも重要な情報が話されているが、残念ながらそのような議事録には記載されていないことがほとんどである。要求獲得会議での大きな問題の1つは、記録が貧弱なことである。多くの重要な情報が失われ、そしてそれらの情報はソフトウェア開発の下流では参照できない。我々のオフライン法の最も重要な特徴の1つは、会議の細目を記録し、会議から重要な情報を収集することである(USPオフライン法の網羅性)。

会議を支援する方法論としては、海谷ら<sup>4)</sup>による研究がある。海谷らの研究でも、会議から議事録へはかなりの欠落事項があることが指摘されている。海谷らの観察によると、ある機能の会話の後にその詳細が話されると欠落することが指摘されている。海谷らの会議中の発言の例と議事録の例を見ると、ある機能の会話の後にその詳細が話されている場合だけでなく、その機能が採択されなかった理由も欠落することが彼

らの文献の会議中の発話の例から見てとれる。

また、設計にきちんとした理由付けを行うことを目的とした、設計理由記録 (Design Rationale) という分野がある<sup>5)</sup>。設計理由記録の1つの研究では、理由の枠組みをあらかじめ決めておいて、その枠組みに従って設計を行う QOC 法がある。ただ、この方法では、あらかじめ枠組みが決まっているので、自由な発想をしにくいという難点がある。また、IBIS, gIBIS といったツールもあるが、その使い方はオンラインで行うことが想定されていて、ツールを操作しているときに会議が寸断するのが会議参加者の思考を妨げるのではないかという懸念がある。

本方法論では、会議に対する干渉は少ない。ただ、設計の理由付けを行うという観点から見れば、本方法論は設計理由記録と競合するものではなく、その長所をとりいれたものであるといえる。

また、設計理由記録では、設計者を支援するのに対して、本方法論では要求獲得プロセスそのものを支援するものである。

網羅性を達成するため、オフライン法では、会議を録音することから始める。エソノメソドロジーの記法を用いて<sup>6)</sup>、会議でのすべての発言を録音したテープから書き起こし、電子化する。我々はこの成果物を「コーパス」と呼ぶ。コーパスは、オフライン法の後の作業で会議の部分を特定するのに便利である。さらに、オフライン法の分析者がコーパスを作成することにより、会議の内容に対する理解が深まるという利点もある。図3にコーパスの例をあげる。コーパスの1行は、識別子(通し番号)と発者話と発話内容からなる。さらに、笑いや沈黙の時間や発話の重なりのような非言語的現象もコーパスに記述する。本方法論では、記述の仕方だけをエソノメソドロジーの記述法を採用しただけで、方法論としては採用していない。

図4にオフライン法の概略を示す。図中、文字を楕円で囲んだ部分は生成物を、文字を四角で囲んだ部分はプロセスを表す。オフライン法はいろいろな方法で会議を分析し、情報を取り出す。これは、いろいろな視点から多くの情報を取り出すためである。たとえば、機能分析では、会議で述べられている、静的な側面の情報を取り出す。シナリオ分析では、会議で述べられている、動的な側面の情報を取り出す。品質分析では、機能の品質に関する情報を取り出す。関心事項分析では、会議参加者の直接には分からない関心事項に関する情報を取り出す。このように、オフライン法の重要な特徴の1つは、会議をいろいろな観点から(マルチビューポイントで)分析することである(USP オフ

- A あの一、大体スケジュール管理、入れないでしょう。  
 A ぼくも入れてないですよ、自分では。  
 A 手帳に書いたほうが速いから。  
 B あー、スケ、スケジュール、つく、使ってない?  
 C そう、そう、そう。ぼくも使ってない。  
 A 使ってないんです。  
 B Emacs も使ってない。  
 C 手で、手でかい、書かないと覚えられないのもあるんですね。  
 A あと、結局、よく家に帰って、あした何かあるかなと思って、  
 C そう何やるか。  
 A こう、見たりするときに、書いてないんですよ。  
 A 一時使っていたことがあるけど。  
 A そうすると、手帳に書いてないもんだから、分からなくなっちゃって、結局、手帳に全部書いてたほうが、実は確実。  
 B ぼくはだから両方に書いてる。  
 A いや。  
 C ぼくもシステム手帳のプリントアウトするようなソフトをつくったことがあるんですが、あれでもだめなんですよ、なんか。  
 A うーん、やっぱり横着になってしまう。  
 B だからもうその人があれでしょう、だからシステム手帳を持ち歩かなきゃいけませんよ、あなたのあれみたいに。  
 C いや、システム手帳は持ち歩いているんですが.....  
 B いや、だからその一、管理するソフト、ソフトというかさ、そういう携帯端末で実際にスケジュールをだから電子手帳みたいに、そこで済ませると、すべて。  
 A うーん。  
 C あ、そうそう。そう。

図3 コーパス

Fig.3 Corpus.

ライン法のマルチビューポイントでの分析)。

オフライン法では作業間には依存関係がある。オフライン法での作業間の関係には入出力関係がある。いい換えると、ある作業の出力が別のある作業の入力になる。これも、オフライン法の重要な特徴の1つである(USP オフライン法のトレーサビリティ)。トレーサビリティの長所は、データの出所を明らかにできるだけではなく、異なる分析結果間の対応さえもとることができる点である。たとえば、機能グループマップのある機能とシナリオのある要素とは、共通の入力である PSG (Problem Solution Graph: 問題解決グラ

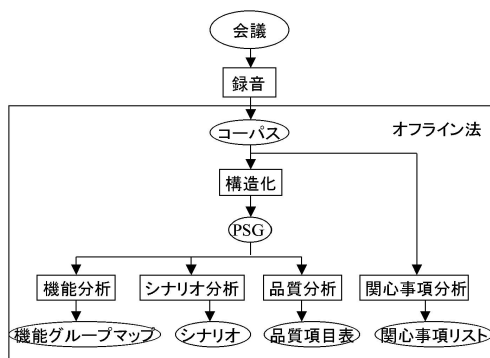


図 4 オフライン法  
Fig. 4 Offline Method.

フ. 次章で詳説する)をたどることによって対応付けできる。トレーサビリティのアンカは、コーパスの発言であり、すべての分析結果とすべての中間データは、少なくとも 1 つのコーパスの発言と対応付け可能である。この性質により、オフライン法のすべての結果は会議参加者の発言が基になっていることが保証される。分析者が自分の意見を結果に付け加えることはできない。

#### 4. 問題解決グラフ (PSG)

我々は、USP の実験を通して、要求獲得会議の大半は「問題点」と「解決策」の発言で占められているという洞察を得た。この洞察は自然に思われる。なぜなら、多くの会議は決定の場であるからである。しかしながら、我々の USP の実験から分かったことは、要求獲得会議では、決定の前に、数十秒から数分の割合で、多数の単純なアイデアや代替案が提出されている。USP オフライン法では、問題点と解決策をその関係を含めて記述するために、問題解決グラフ (Problem Solution Graph: PSG) を考案した。いい換えると、PSG は問題点と解決策の観点から要求獲得会議を要約したものである。図 5 に PSG の例を示す。PSG は有向グラフである。そのノードには、ノードの種類、要求内容、識別子、コーパスへのトレーサビリティのための属性、決定状態、他のノードへのリンク、仮の順番が記述されている。以下、PSG のノードを「要求項目」と呼ぶ。

要求獲得の分野で周知の問題点/ニーズ分析<sup>7)</sup>では、問題点とニーズを洗い出し、問題点を原因と結果でつなぐことで分析を進めていく。前述のように、会議は問題点と解決策からなっているので、ニーズではなく、解決策を採用した。

PSG は問題点と解決策を軸に分析を進めるので、以

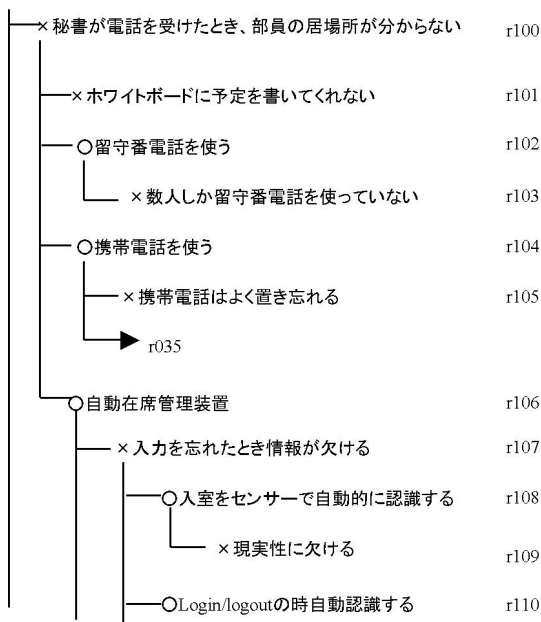


図 5 PSG  
Fig. 5 PSG.

下のように要求項目とリンクの意味を決定した。要求項目には 2 種類あり、x で表される問題点と、o で表される解決策がある。要求内容は、分析者がいくつかの発言を要約した自然言語の文または句である。図 5 では、見やすくするため、すべての属性は記述していない。

PSG のリンクの意味は、以下のとおりである。

- (1) 解決策  $S$  から問題点  $P$  へのリンクの意味は、「 $P$  は  $S$  の問題点である」。
- (2) 問題点  $P$  から解決策  $S$  へのリンクの意味は、「 $S$  は  $P$  の解決策である」。
- (3) 解決策  $S_1$  から解決策  $S_2$  へのリンクの意味は、「 $S_2$  は  $S_1$  の部分解決策である」。
- (4) 問題点  $P_1$  から問題点  $P_2$  へのリンクの意味は、「 $P_2$  は  $P_1$  の部分問題点である」。

一般的にいうと、PSG は木構造というよりも、グラフ構造である。しかしながら、我々は、図 5 のように木構造を中心に PSG を記述することにした。図の上で離れたところにある要求項目に対しては、ジャンプリンクとして「」を用いることとした。たとえば、図 5 で「 r035 」は要求項目 r104 から r035 へのリンクを意味する。

コーパスの大部分を PSG の形で構造化した後は、さらに 2 つ問題がある。

- (1) PSG の問題点/解決策は粒度が小さすぎるため、もう少し大きなかたまりにまとめ、そのかたま

りの間の関係を発見する方法論が必要である。そして、そのかたまりをソフトウェア工学の、機能やシナリオといった周知の概念にまとめる必要がある。

- (2) 問題点/解決策以外にも重要な情報がある。決定事項のように PSG の項目の属性として記述可能なものもある。しかし、PSG の中に記述できない、あるいは、関心事項のように PSG の中にはなく、コーパスの中に存在するものがある。

そこで我々は、以下で述べるように、PSG あるいはコーパスを入力とする分析を構築した。

さらに、実験を通して分かったことは、ある発言は直接ターゲット・システムについて言及しているが、システム構築のアクション・プランについて述べられている発言もあるということであった。アクション・プランはいろいろな粒度の要求項目に含まれる。たとえば「次週までにカタログで価格を調べる」のように粒度が小さい要求項目や「来年のシステム運用試験」のように粒度が大きい要求項目に含まれる。結果として、要求項目を以下の  $2 \times 2 = 4$  通りのパターンに分類した。

- (1) 解決策の形をとるもの(例: 図5の「携帯電話を使う」r104)
- (2) 解決策の形をとるもので、システム開発に関するアクションプランの提案(例: 図5の「自動在席管理装置」r106)
- (3) 問題点の形をとるもので、ターゲット・システムに関する解決策の問題点(例: 図5の「現実性に欠ける」r109)
- (4) 問題点の形をとるもので、開発プランに関する解決策の問題点(例: 「1月以内に開発する」)

我々は、仕様の観点だけではなく、アクション・プランの観点からでも(さらにアクション・プラン間の関係も)、要求を獲得するために、すべての種類の問題点と解決策を1つの PSG の中に記述することとした。我々の実験によると、2つの観点から同時に分析した方がよい会議と、別々に分析した方がよい会議があることが分かった。後者の場合、仕様 PSG とアクション・プラン PSG の2つの PSG に分割して分析を行った。以下の分析は2つの PSG に分割して行ったものである。

PSG の構築の仕方について少し補足しておく。PSG の項目が「問題点」であるのか「解決策」であるのかは、基本的に分析者に任せられている。実際に分析を行ってみると、大半のケースは問題なくどちらかに分

類される。意見が一致しない部分は、問題点とも解決策ともとれる発言であった。このような場合は、問題点としても解決策としても PSG の項目としてあげることとした。また、前述の例を凡例として分析者に与えると、PSG の項目はほぼ同じ粒度になった。また、リンクの張り方には個人差が生じる。しかし、下流でリンクの張り方に依存するプロセスでは、分析の結果を人間が見て試行錯誤を繰り返すため、実用上は問題がなかった。

## 5. 機能分析

要求獲得会議の主たる目的の1つはターゲットシステムの機能を明確にすることである。PSG の解決策は機能の候補である。しかし、解決策の粒度はまちまちであるので、解決策をそのまま機能とすることはできない。概して、PSG の解決策の粒度は小さすぎ、またその数は多すぎる。機能分析では、解決策のグルーピングを支援し、機能を抽出し、機能間の関係を明らかにする。機能分析には、以下の3つの作業がある。

- (1) USP 方法論では、解決策のグループを生成し、ソリューション・ネット内の関係を自動的に描画する。これは、次の機能抽出作業の支援となる。
- (2) 分析者は機能を抽出する。USP 方法論では、分析者のために、ソリューション・ネット内をナビゲートする機能を提供する。
- (3) USP 方法論では、抽出された機能間の関係を発見し、機能グループマップに描画する。

### 5.1 ソリューション・ネット

機能分析では、解決策のグループとそれらの間の関係を表現するソリューション・ネットというネットワークを自動的に生成する。解決策のグループとグループ間の関係は、それぞれの解決策が問題点に対する寄与の程度によって生成される。図6はソリューション・ネットの例である。ソリューション・ネットの各々のノードは、同じ問題点グループに寄与する解決策グループからなる。アークの到達ノードの寄与は、アークの出発ノードの寄与の固有集合である。

PSG からどのようにソリューション・ネットが自動的に生成されるかを説明する。まず、それぞれの解決策の寄与を計算する。PSG 上での問題点  $P$  から解決策  $S$  へのリンクは、 $S$  への  $P$  の寄与として計算する。さらに、PSG の推移性による、間接的な寄与も計算する。PSG 上でのリンクによって表される推移的關係には以下の4つがある。

- (1)  $S_2$  が  $P_1$  の解決策であり、 $P_3$  が  $S_2$  の問題点であり、 $S_4$  が  $P_3$  の解決策であれば、 $S_4$  は  $P_1$

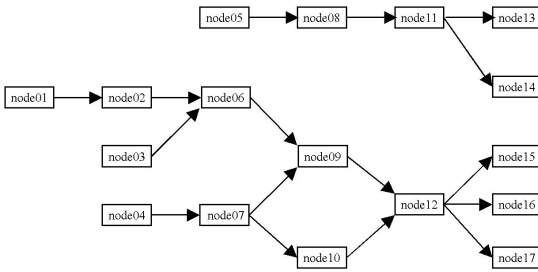


図 6 ソリューション・ネット  
Fig.6 Solution Net.

の解決策でもある。

- (2)  $S_2$  が  $P_1$  の解決策であり,  $S_3$  が  $S_1$  の部分解決策であれば,  $S_3$  は  $P_1$  の解決策でもある。
- (3)  $S_2$  が  $S_1$  の部分解決策であり,  $P_3$  が  $S_2$  の問題点であれば,  $P_3$  は  $S_1$  の問題点でもある。
- (4)  $P_2$  が  $S_1$  の問題点であり,  $P_3$  が  $P_2$  の部分問題点であれば,  $P_3$  は  $S_1$  の問題点でもある。

このような PSG 固有の推移性を利用して, 各々の解決策への寄与が計算される。

次に, 解決策の寄与間の同値関係と部分全体関係を用いて, ソリューション・ネットを生成する。部分全体関係は半順序関係であるので, ソリューション・ネットは無閉路グラフである。ソリューション・ネットの中の左の解決策は右の解決策よりも問題点に対する寄与度が高い。同じ問題点の集合へ寄与する解決策は, ノードとしてグループ化されている。

## 5.2 ソリューション・ネット内のナビゲーション

一般的にいて, ソリューション・ネットのノードは粒度が細かすぎて, そのまま機能としては使えない。分析者は機能のコア・ノードを発見して, 他のノードをそのノードの周辺に集めなければならない。この作業のために, USP 方法論では, 以下のような戦略でソリューション・ネット内をナビゲートする仕組みが必要となる。

**最左ノード戦略** 最左ノードはコア・ノードの有力候補である。なぜなら, 最左ノードは最も寄与度が高い。いい換えると, 数多くの問題点を解決しているという意味で, 最も良い解決策である。

**最大ノード戦略** 多くの解決策を含むノードはコア・ノードの有力候補である。なぜなら, 共通の問題点集合を解決するために多くの解決策が議論されたからである。

**近傍ノード戦略** アークでつながれた 2 つのノードの中にある解決策は, 同じではないが, 似た寄与を持つ。そこで, これらのノードは同じ機能となる

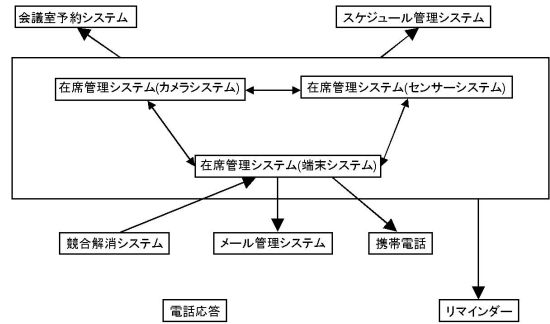


図 7 機能グループマップ  
Fig.7 Function Group Map.

候補である。

ナビゲーションする仕組みは分析者がコア・ノードの候補を得たいとき, あるいはそのヒントが必要となるときに有効である。

## 5.3 機能グループマップ

機能が抽出されると, 次に, 機能間の関係を発見し, 機能グループマップに自動的に描画される。機能グループマップ内の関係は以下のように発見される。ノード  $N_1$  中の解決策  $S_1$  が機能  $F_1$  中にあり, ノード  $N_2$  中の解決策  $S_2$  が機能  $F_2$  中にあり,  $N_1$  から  $N_2$  への線があれば,  $F_1$  から  $F_2$  への関係が抽出される。

図 7 に機能グループマップの例を示す。機能グループマップでは, 関係は矢印で表される。反射関係は表現されない。双方向関係は双方向の矢印として表現される。矢印の特長によって, 機能グループが導出される。機能グループは, 機能の集合であり, 図では四角で囲んで表現される。機能グループのすべての機能とつながっている線は機能グループの矢印として表現される。機能グループは, 分析者が矢印の数を減らそうとするときに必要となる。矢印の置き換えは自動的に行われる。

## 6. シナリオ分析

ターゲット・システムの時系列特徴を明確にすることも要求獲得会議の主な目的の 1 つである。シナリオ分析では, 時系列特徴を分析し, 「シナリオ」と呼ばれるグラフを作成する。図 8 がシナリオの例である。シナリオの各々のノードはソリューション・ネットや機能グループマップと同様に解決策の集合からなる。しかし, シナリオでは, ターゲットシステムのイベントが記述してある。シナリオの各々の線は, ソース・ノードからターゲット・ノードに向けて, 時系列の状況が記述してある。図の線の太さの違いは, 関係が強

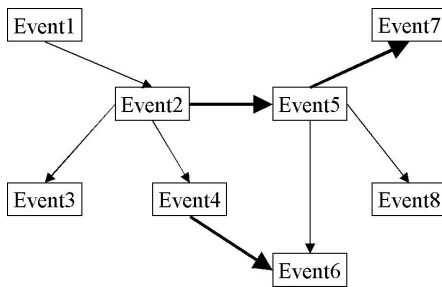


図 8 シナリオ  
Fig. 8 Scenario.

いものがより太い線で描かれている。シナリオ内の線の種類は、生成のされ方に応じて、3種類ある。それは、以下の作業となる。

- (1) USP 方法論では、解決策を自動的にカテゴリ化する。これは次章で説明する Keyword Associator によって自動的に生成される。生成されたカテゴリはシナリオのノードの候補となる。分析者は候補からノードを生成する。
- (2) 分析者は PSG の要求項目のいくつかの属性を利用して、シナリオのアークを作る。
- (3) 機能とシナリオのノードとの関係は PSG のリンクのトレーサビリティを用いて自動的に検出される。

前述の海谷らの研究では、仕様書を作ることを目的としているので、本章で述べたようなプランに関する会話は除外している。USP 方法論では、ターゲットシステムの時系列特徴を明確にすることも要求獲得会議の主な目的と考え、プランに関する情報も扱うことにした。

### 6.1 Keyword Associator

機能分析では、解決策の貢献という観点から、解決策のグループを生成するのに対して、シナリオ分析では、解決策の関係という観点から解決策のグループを生成する。Keyword Associator<sup>8)</sup> は解決策とカテゴリの関係を計算する。

各々の解決策の内容は自然言語の文である。さらに、それぞれの解決策に対して、トレーサビリティを用いて、コーパス内の関係する文を検索できる。Keyword Associator は、2つの解決策の内容と関連する発話に関する単語の共起を数え上げる。共起する単語の数をそれらの解決策の関連度とする。

次に、Keyword Associator は解決策の関連度を利用して、解決策をカテゴリ化する。関連度の強さを利用して、新しいカテゴリを生成していく。2つのカテゴリを1つのカテゴリにしてしまうわけではない。

分析者は、その生成されたカテゴリを基にシナリオのノードを作る。もし、カテゴリの数が多すぎて扱いにくいときは、Keyword Associator を使ってさらにカテゴリを進めることができる。

### 6.2 シナリオ分析での時間関係

シナリオ分析におけるアークはシナリオにおけるノードによって表されるイベントの時間関係を表す。前述のように、アークを生成する方法はいくつかある。そこで、アークの種類は以下ようになる。

- PSG での解決策は時間関係を記述した属性を持っている。シナリオのアークはこの属性から直接作られる。
- PSG での解決策は入出力を記述した属性を持っている。解決策  $S_1$  が出力  $X$  を持っていて、解決策  $S_2$  が入力  $X$  を持っていれば、 $S_1$  から  $S_2$  への時間的順序関係が存在する可能性がある。分析者が時間関係を確認すれば、 $S_1$  から  $S_2$  へのアークが生成できる。
- 分析者がノード間に時間関係を発見すれば、アークを生成できる。解決策の内容と関連した発話が時間関係を発見するときの助けになる。

### 6.3 機能とシナリオの関係

各々の機能は PSG の解決策の集合からなる。各々のシナリオも PSG の解決策の集合からなる。つまり、機能とシナリオは双方とも解決策に対してトレーサビリティを持つ。USP 方法論は、トレーサビリティを用いて機能とシナリオの間の関係を発見できる。機能とシナリオの両方の基になっている解決策があれば、その機能とそのシナリオは対応関係を持つことが分かる。

## 7. 品質分析

ターゲットシステムの品質を明確にすることは重要であり、要求獲得会議での重要な議題の1つである。我々は、ターゲットシステムの品質を機能とシステムの属性と考える。品質分析では「コストを無視している」のようなコスト要求も扱う。

機能分析とシナリオ分析が PSG の解決策を基にしていたのに対して、品質分析では、PSG の問題点を基にする。解決策の問題点は、解決策の品質の欠落である場合と、問題点の解決策は、その問題点における品質の欠落を満たすものである場合があるからである。なお、本章では、土井<sup>9)</sup>の論文の要点を解説し、さらに、USP の方法論での使い方について述べる。

品質分析は以下のような3つの作業からなる。

- (1) 分析者は PSG の問題点から品質関連表現を抽出する。たとえば、図 5 では、問題点 r107 が

機能 \ 品質	セキユリティ	正確さ	実現可能性	頑健性	有益性	使いやすさ
機能1	×	×		○		○
機能2		○	○○	×		
機能3	○		×	○	○	○
機能4	××		○		×	

図 9 品質項目表

Fig.9 Quality Element Table.

ら頑健性が、問題点 r109 からは実現可能性が抽出される。品質関連表現と問題点は多対多関係にある。分析者は、問題点だけではなく、トレーサビリティを用いて関連する発話を参照できる。

- (2) USP 方法論では、品質と機能の関係が分かる。ある品質の基になっている問題点がある解決策の下にあれば、その問題点と解決策から生成された機能は関連がある。また逆に、ある品質の基になっている問題点がある解決策の上であれば、その問題点と解決策から生成された機能には関連があることになる。
- (3) USP 方法論では、品質関連表現と機能の対応付けを行う。この対応付けした表を品質項目表と呼ぶ。図 9 が品質項目表の例である。品質関連表現が表の縦列に、機能が表の横列に表現されている。表中の ○ と × はその品質関連表現が機能に対して良い品質か悪い品質かを表す、分析者の判断である。

品質分析の結果は品質項目表の中だけで使われるのではなく、6.3 節で述べたのと同じ方法で、機能、シナリオ、品質から PSG へのトレーサビリティを利用して機能やシナリオから間接的に利用できる。

品質分析の先行研究として、QFD 法がある<sup>10)</sup>。QFD 法でも、機能と品質のマトリックスを書くのだが、マトリックスの要素として悪い品質であることが書けない。本方法論では、悪い品質も記述できるようにした。こうすることによって、下流で留意するところが明確となる。また、QFD 法自体はそのマトリックスの生成の仕方について特に報告がないが、会議の場で使うことが意図されている。本方法論では、オフラインで分析を行うため、前述のように会議参加者の思考を妨げることなく、会議を進めることができる。

### 8. 関心事項分析

機能分析、シナリオ分析、品質分析は PSG の問題点と解決策を基にして行われる。しかし、会議には問題点と解決策のほかにも多種多様な情報がある。それらの多種多様な情報の一部は問題点や解決策のように明示的に現れる。しかし、自然言語のコミュニケーションの中に埋もれてしまうものもある。

関心事項分析は、こうした自然言語のコミュニケーションに埋もれてしまう情報を掘り起こす方法論である。会議参加者の関心事項が無意識のうちにあるいは偶然に隠されてしまうのはけっして特別なことではない。そしてそれらの関心事項は会議では明示的には示されない。会議を注意深く観察すると隠された関心事項によって引き起こされる現象を発見できる。関心事項分析では、心理学/言語学的アプローチによって、隠された関心事項を明らかにすることを目的とする。関心事項分析で採用する現象は以下のとおりである。言い直し 発話者は発言を言い直すことがある。たとえば、

あ、メ、電話の取り次ぎってことね。

という例をあげて説明する。心理学による分析の結果、言い直す前の単語に関心があることがある<sup>11)</sup>。この例では「メモ」を言い直して「電話の取り次ぎ」に言い直したことが前後の文脈から分かる。「電話の取り次ぎ」よりも「電話の伝言メモ」の方に関心が高かったことが後にアンケートの結果分かった<sup>11)</sup>。このように、言い直しに注目することにより、発話者の隠された関心事項を抽出できる場合がある。

笑い 会議では笑いが起きることもある。これも隠された関心事項を示す現象である。笑いは、その背後に本音・極論・職場風土に関する情報を含むことがある。たとえば、

「で、いないときのメモの置き場にもちょっと悩むときがあって(笑)、だからーん。」

という例では、机の上が散らかっていてメモの置き場に困る状況が述べられている。この発話から、電話の伝言メモの扱い方について有用な情報が得られる。さらに、伝言メモを電子的にしたときも、多量の情報の中に埋もれてしまうことが推論できる<sup>12)</sup>。

コーパスには言い直しと笑いが記述してあるので、USP 法では、その前後の発話を参照することで、分析を行う。分析者は言い直しと笑いを調べることに



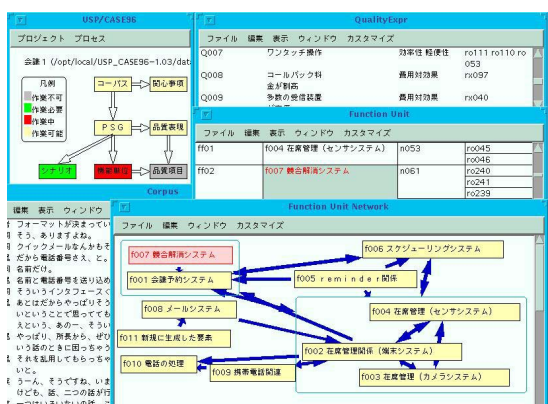


図 10 USP CASE TOOL  
Fig. 10 USP CASE TOOL.

り、発話者の隠された関心事項の候補を推論することができる。

## 9. USP CASE TOOL

我々は分析を効率良く進めるために JAVA によって CASE TOOL を開発した。

本 CASE TOOL は以下のような機能を持つ。

- (1) 機能分析でソリューション・ネットの生成を自動的に行う。また、シナリオ分析で、解決策のカテゴリー化を自動的に行う。
- (2) 機能分析でソリューション・ネット内を探索するとき、シナリオ分析で PSG の属性からアークを生成するとき、品質分析でプロセスを進行させるときのナビゲーション機能を設けた。この結果ナビゲーションが容易になった。
- (3) コーパスと要求項目の表現を正規化する機能を設けた。この結果分析結果を自動的に正規化することが可能になった。

いずれの機能も手作業で分析するのに比べると、自動化できるところは自動化できるようにし、また、入力フォームを用意することによって、出力を正規化した。この結果、分析者の工数が減り、結果が正確になるという効果が得られた。

図 10 に画面の例をあげる。図 10 は機能分析をしているときのスナップショットである。画面左上が分析プロセスの進捗状況を表すウィンドウである。図では白黒だが、実際にはカラーで、各プロセスの進捗状況が分かる。画面右上は品質分析のウィンドウである。その下が機能分析のウィンドウである。右の一番下は機能の関係を図示したウィンドウである。最後に、左下はコーパスウィンドウである。USP CASE TOOL では、ドローイングを自動的にする。文章の自動的

解析はしない。基本的に入力を補佐し、出力を正規化することに用い、高度な分析は人間が行う。このように USP CASE TOOL では、各プロセスを相互に参照しながら効率的に、また、正確に分析を進めることができる。この結果は、実際の要求獲得の現場でも有効であるとの感想をもらった。

## 10. おわりに

本論文では、会議に基づく要求獲得方法論、USP 法を提案した。本方法論では、要求を明確化し、あいまい性をなくし、潜在的な要求を引き出すことができ、会議で提供された要求を網羅的に獲得することができる。USP 方法論の重要な長所は、網羅的な獲得、マルチビューポイントでの獲得、トレーサビリティである。USP 方法論による要求獲得は、オンライン法とオフライン法との繰返しによるスパイラルモデルをとっている。オフライン法では、マルチビューポイントで獲得する。機能分析では、ターゲットシステムの機能を獲得する。シナリオ分析では、ターゲットシステムの時系列情報を獲得する。品質分析では、品質と機能の関係を獲得する。関心事項分析では、会議参加者の隠された関心事項を獲得する。

本方法論とツールは、計 3 回の会議を分析した結果に基づいて構築された。そのうち 2 回は模擬会議ではあるが、最後の 1 回は実際の要求獲得会議を対象とした。また、本方法論とツールを実際の要求獲得の現場に提案したところ意図したとおりの評価を得られた。

今後の課題は以下のとおりである。

- (1) 問題点と解決策以外の情報を記述できるように PSG を改良する必要がある。たとえば、事実の報告、単なる質問とそれに対する回答などがあげられる。改良された PSG は完全な会議録になりうる。
- (2) ソフトウェア開発のスケジュールとプラン生成も要求獲得会議の重要な議題である。オフライン法では、現状の方法と似たような方法で、これらを獲得できる。つまり、ソフトウェア開発のシナリオ分析を行えばよい。
- (3) オフライン法の結果を効果的に用いるオンライン法の構築。

謝辞 本研究の推進にあたり、熱心な議論と助言を下された、USP グループの皆様と、東京理科大学の大森晃先生に感謝いたします。

## 参 考 文 献

- 1) Finkelstein, A.: Requirements Engineering: a review and research agenda, *Proc. Asia-Pacific Software Engineering Conference* (1994).
- 2) 大森 晃, 土井晃一: オフライン要求獲得法の提案, 情報処理学会第 48 回全国大会, Vol.5, No.4K-6, pp.373-374 (1994).
- 3) Potts, A., Takahashi, K., Smith, J. and Ota, K.: An Evaluation of Inquiry-Based Requirements Analysis for an Internet Service, *Proc. IEEE International Symposium on Requirements Engineering* (1995).
- 4) 海谷治彦, 佐伯元司: プロダクトを基にしたソフトウェアの仕様作成のための会議の分析法, 電子情報通信学会論文誌, Vol.J78-D-I, No.1, pp.23-31 (1995).
- 5) Moran, T.P. and Carroll, J.M.: *Design Rationale Concepts, Techniques and Use*, Lawrence Erlbaum Associates (1996).
- 6) Atkinson, J.M. and Heritage J. (Ed): Structures of Social Action, *Studies in Conversation Analysis*, Cambridge University Press (1984).
- 7) 大木幹雄: ソフトウェア設計の基礎, pp.212-213, 日本理工出版会 (2000).
- 8) 渡部 勇, 三末和男: 単語の連想関係によるテキストマイニング, 情報処理学会情報学基礎研究会, 55-8, デジタル・ドキュメント, 19-8, pp.57-64 (July 1999).
- 9) 土井晃一: 要求獲得オフライン法での質的要求の抽出法, 情報処理学会論文誌, Vol.40, No.11, pp.4012-4020 (1999).
- 10) Daetz, D.: QFD: a method for guaranteeing communication of the customer voice through the whole product development cycle, *IEEE International Conference on Communication*, pp.1329-1333 (1989).
- 11) 土井晃一: ソフトウェアの要求獲得会議での言い直しに注目した要求獲得方法論, 自然言語処理, Vol.7, No.1, pp.3-12 (2000).
- 12) 土井晃一: 自然言語コーパスにおける「笑い」を利用したソフトウェアの要求獲得と情報検索への応用, 言語処理学会第 4 回年次大会, 3 月 (1998).

(平成 13 年 4 月 13 日受付)

(平成 14 年 10 月 7 日採録)



土井 晃一 (正会員)

1961 年生. 1991 年東京大学大学院情報工学専攻博士課程修了. 工学博士. 同年富士通研究所国際情報社会科学研究所 (現富士通研究所コンピュータシステム研究所) 入社. 自然言語理解, 人工知能, ソフトウェア工学, 情報検索等の研究に従事. 1998 年 9 月より 1999 年 10 月まで文部省学術情報センター客員助教授併任. 2001 年 (株) セレスター・レキシコ・サイエンシズ. 2001 年 10 月より奈良先端科学技術大学院大学客員助教授, 現在に至る. 日本認知科学会, 人工知能学会, ソフトウェア科学会, 言語処理学会各会員.



蓬菜 尚幸 (正会員)

1984 年東京工業大学工学部情報工学科卒業. 昭和 61 年同大学院理工学研究科情報工学専攻修士課程修了. 同年富士通 (株) 国際情報社会科学研究所入社. 平成 12 年セレスター・レキシコ・サイエンシズ (株) に入社し, 現在に至る. また, 東京工業大学情報理工学研究科計算工学専攻博士課程に在学中. ソフトウェア工学, バイオインフォマティクスの研究開発に従事.



渡部 勇 (正会員)

1961 年生. 1985 年慶應義塾大学理工学部機械工学科卒業. 1987 年同大学院制御工学専攻修士課程修了. 同年, 富士通 (株) 入社. 現在 (株) 富士通研究所 IT メディア研究所に勤務. 発想支援, 情報検索, テキストマイニング等の研究に従事. 人工知能学会会員.



片山 佳則 (正会員)

1961 年生. 1985 年東京理科大学大学院理工学研究科経営工学専攻修士課程修了, 同年富士通 (株) 入社, 1990 年 (株) 富士通研究所に編入, 現在に至る. セマンティック Web, Web サービス等に関する研究に従事, 日本ソフトウェア科学会会員.



園部 正幸(正会員)

1975年東京大学理学部物理学科卒業。同年富士通(株)入社。都市銀行大規模オンラインデータベースシステムの導入・開発支援・構築,日本語プリントシステム開発に従事した。1987年(株)富士通研究所に移籍。バッチの並列化,仕様書の自動生成,要求獲得,セキュリティシステム,仮想経済社会システム,公衆自動搬送システム(CATS),バイオインフォマティクスの研究に従事した。2000年セレスター・レキシコ・サイエンシズ(株)CLSフェロー。2002年退社し,システムの研究と企画を行っている。人工知能学会会員「グレード可変コンピュータ」の発明で関東電気協会考案賞を受賞(1991年)。 <http://sonobe.s5.xrea.com/>

---