

インタビューによる要求抽出作業を 誘導するシステムの実現方法

古宮 誠一*1, 加藤潤三*2,
永田守男*3, 大西 淳*4, 佐伯元司*5,
山本修一郎*6, 蓬萊尚幸*7

*1 情報処理振興事業協会 (㈱日立製作所より出向中), *2 (株)コムニク創研,
*3 慶応義塾大学, *4 立命館大学, *5 東京工業大学
*6 NTTソフトウェア研究所, *7 富士通研究所

*1 〒113-6591 文京区本駒込2-28-8
(文京グリーンコート センターオフィス16階)
komiya@ipa.go.jp,

あらまし ソフトウェアに対する顧客の要求を抽出する技術をインタビュー技術であると捉え、インタビューによる要求抽出作業を誘導するシステムをWWW上に開発中である。この論文では、インタビューによる要求抽出作業を誘導するシステムの実現方法を明らかにしている。

キーワード インタビューによる要求抽出、要求抽出作業の誘導、インタビュー作業の進捗管理、カラーリング、WWW上での黒板機能、エクストラネット

A Realization Method of the system to Navigate Requirements Elicitation Work Driven by Interviews

Seiichi Komiya *1, Junzo Kato *2,
Morio Nagata *3, Atsushi Ohnishi *4, Motoshi Saeki *5
Shuichiroh Yamamoto *6, Hisayuki Horai *7

*1 Information-Technology Promotion Agency Japan (Also with Hitachi Ltd.),
*2 Comnic-Soken Corporation,
*3 Keio University, *4 Ritsumeikan University, *5 Tokyo Institute of Technology,
*6 NTT Software Laboratories, *7 Fujitsu Laboratories Ltd.

16 th Floor, Center Office, Bunkyo Green Court,
2-28-8, Honkomagome, Bunkyo-ku, Tokyo 113-6591, Japan

Abstract: The authors regard software requirements elicitation technology as an interview technique, are developing a sytem on WWW to navigate software requirements elicitation work. This paper describes a realization method of the system to navigate software requirements elicitation work.

Keywords: Requirements Elicitation Driven by Interviews,
Navigation of Requirements Elicitation Work, Coloring,
Black Board Facilities on WWW, Extra-net

1. はじめに

ソフトウェアは開発予定のソフトウェアに対する顧客の要求をまとめた要求仕様書に基づいて開発される。それ故、顧客の要求が要求仕様書に正しく、かつ、遺漏なく反映されているのでなかったら、作業のやり直しに伴う工程遅延やコスト高が発生し、悪くするとソフトウェア開発プロジェクトは致命的な打撃を受けることになる。このため、ソフトウェアに対する顧客の要求を正しく、かつ、遺漏なく抽出することが求められる。しかし、どんなに経験豊かなソフトウェア技術者(SE)でも顧客の要求を正しく、かつ、遺漏なく抽出することは容易ではない。それ故、あまり経験のないソフトウェア技術者でも正しく、かつ、遺漏なく要求を抽出できるような技術を開発することが望まれる。

ソフトウェアの要求分析過程は、これまで①要求抽出、②要求仕様記述、③要求仕様の欠陥解析、の3つであると考え、formalな要求仕様記述を基にした②と③の実現やそれらへの支援技術が研究開発の中心であり、①の要求抽出過程の研究はなおざりにされてきた。しかし、②と③は①の要求抽出が可能となって初めて可能となるソフトウェア開発プロセスである。それ故、要求分析過程を、それも特に①の要求抽出過程を研究する必要がある。

図1に、これまでの研究における要求分析プロセスと、当研究の要求分析プロセスの比較を示す。

従来の研究 (現場でのプロセスとは異なる)	現場でのプロセス	当研究 (現場でのプロセスのまま)
(要求抽出過程が欠落)	多人数による要求抽出 (結果の記述はinformal)	
	要求相互における 重複や矛盾のチェック	左記の作業を誰でもできる ように計算機で知的にガイド
	グループでの合意に 基づく仕様の決定	
formalな仕様記述 言語による要求記述	多人数による要求仕様書 (informalな記述)の執筆	要求仕様書(informalな 記述)の自動生成
formalな記述に基づく 要求仕様の欠陥解析		
(これより設計過程)		
formalな記述に基づく プログラムの自動生成	要求仕様書に基づき 設計仕様書などを 多人数で分担して執筆	GroupWork Managerで 設計視察とそれに対する グループ討論を計算機支援

図1 ソフトウェア要求分析過程の比較

これまで、要求分析は、SEが顧客と面談したり文書を交換したりして顧客の要求を聞き出し、スーパーSEの知識や経験により要求仕様書を作成していた。このため、これらの知識を持たないSEでも、顧客から要求を抽出し要求仕様書を作成できるようにしたい。そこで、我々はソフトウェア要求抽出技術をインタビュー技術であると捉え、スーパーSEの知識を備えてインタビューを誘導するシステムを研究開発する。

この研究目的を達成するために、我々は次のようなアプローチを採用する。まず最初に、インタビューによってソフトウェアの要求を抽出する作業に必要な、最低限の機能を持つツールを作成する。次に、このツールを使い、インタビューによってソフトウェアの要求を抽出する過程を実

験する。そして、この実験結果を基にツールを改良し、改良されたツールを使ってこの過程の実験を行う。納得するまでこのサイクルを繰り返すことにより、この研究を行う。

なお、一般性を確保するために、インタビューである顧客の人数は原則として複数とし、インタビューであるSEは複数でもよいという前提の下でツールを開発する。また、SEのリーダーの主導の下にこのツールを使用することを前提とする(図2参照)。

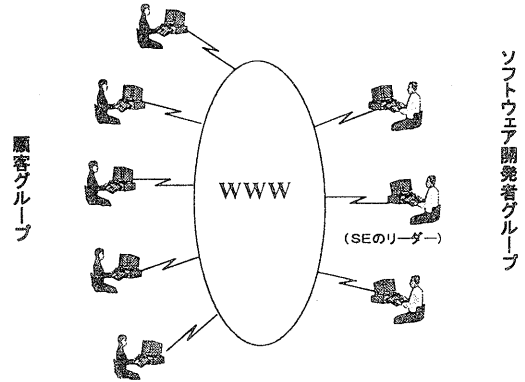


図2 ネットワーク上でのソフトウェア要求分析

また、インタビューを効果的に行うために、インタビューにおける質疑応答の履歴を画面で一覧できるようにしたい。そのために、これまで対面法や文書のやりとりで行っていたインタビューをWWW上で行うようにする。

さらに、WWWを使用することにより、離れた事業所に散在する技術者が互いに各自の職場を離れずに議論できるように、ネットワーク・セキュリティを保証する機能を実現する。このことは、ソフトウェアの要求抽出のために、技術者が客先に何度も足を運ばなくても済むようになるなどの付加価値を生む。

本論文の構成は次のとおりである。2章で、インタビューによる要求抽出作業を誘導する方法について明らかにする。3章で、その他の支援機能について明らかにする。4章で本論文のまとめを記述する。

2. 要求抽出作業を誘導する方法

2.1 シナリオによる支援方法

SEが顧客にインタビューすることによって、ソフトウェア仕様に対する顧客の要求を抽出する作業を誘導するためには、そもそもどのようなことが議論されなければならないか、ということから明らかにしなければならない。このため、仮想インタビューによる要求抽出実験を実施した[6]。ソフトウェア開発領域の例としては、「バック旅行の申し込みをインターネットで受付け、計算機で処理するシステム」を選び、そのソフトウェアに対する要求を抽出した。被験者は、1人で1組の組が3つ、2人で1組の組が1つの計4組(計5人)である。この実験で得られたデータとその分析例を表1に示す。

表1 仮想インタビューによる実験で
得られたデータとその分析例

発言者	カテゴリー	発言内容	発言のタイプ
SE	Schedule	バック旅行の申し込み受付の件ですが、いつ頃までに実現させたいのですか?	質問
顧客	Schedule	実現可能ということが判れば、できるだけ早く実現させたいです。	回答
顧客	Examples	どのような場合、通常はこのくらいの期間があれば可能なのですか?	質問

分析の結果、要求抽出のために議論されるべき内容は、

- What: 開発しようとするソフトウェアの機能
- Examples: 例えば、どのようなことができるか
- Why: 開発の理由や背景
- Budget: 開発予算
- Schedule: 開発期間
- Current system: 現行システムについての質疑
- Conditions: 設計条件など
- Constraints: 制約
- Policy: 開発方針

の9種類のカテゴリー（話題の集合）に分類できることが判明した。また、インタビューによる要求抽出では、採り上げる話題の順序によって、抽出作業の効率が良くなったり悪くなったりすると考えられる。このため、インタビューで採り上げるカテゴリーの順序を規定したものをシナリオと呼び、予めこれを登録しておき、これに基づいてインタビューを進めて行く必要がある。

ここでは、システムが用意するシナリオの考え方を示す。要求を効率よく抽出するためにSEが最初になすべきことは、インタビューにおいて、まず最初にシステムの全体像を明らかにすることである。そのためには、大雑把でよいから、開発しようとするシステムの機能についてインタビューすることから始めるのが効果的である。このことは、インタビューによる要求抽出において、最初に採り上げるべき話題の集合がWhatであることを示している。このとき、開発しようとするシステムの機能を正しく理解するには、具体例を挙げると議論がスムーズに進展することが多い。これらの話題はExamplesに属する。そして、顧客から要求のあった機能のそれぞれが、本当に必要であるか否かを正しく判断するためには、システムを開発するに至った背景や開発の理由を理解する必要がある。そのための話題はWhyに属する。また、開発しようとするシステムの機能を煮詰めるには現行システムがどうなっているかどうなっているかという議論が必要である。これらの話題はCurrent systemに属する。従って、これらのカテゴリーに属する話題は、時を同じくして一緒に議論すべきである。以上の理由から、

要求抽出のためにSEが顧客に対して行うインタビューにおいて、最初に採り上げるべきは、Whatに属する話題であり、Whatに属する話題をより充実したものにするにはExamples, Why, Current systemに属する話題が必要である。

これまでシステムの機能について議論してきたので、次に議論すべきは、どのような前提の上でそれらの機能を実現するかという、設計条件に関する話題である。Conditions, Constraints, Policyなどに属する話題がこれに該当する。そして、最後に議論すべきは、どのような開発予算、開発期間でそれらを開発するかということである。何故なら、今まで議論してきた機能のすべてがそれらの予算や開発期間で開発可能かどうか、それによって明らかになるからである。これらの話題はBudgetとScheduleに属する。

上記のような考え方によりでき上がった、システムが用意するシナリオの例を表2に示す。

表2 システムが用意するシナリオ

カテゴリー	順序	タイプ	カテゴリー	順序	タイプ
What	1 st	P	Current system	1 st	P
Examples	1 st	P	Conditions	2 nd	S
Why	1 st	P	Constraints	2 nd	S
Budget	3 rd	P	Policy	2 nd	S
Schedule	3 rd	P			

表2における順序は、要求抽出のためにインタビューで採り上げる話題の集合の順序を示している。例えば、What, Examples, Why, Current systemの1stグループと、Conditions, Constraints, Policyの2ndグループと、Budget, Scheduleの3rdグループの3つは、それぞれに属する話題を採り上げる順序が各グループの中では同一であるが、グループ間では異なっている。これは最初に1stグループに属する話題を採り上げ、次に2ndグループ、そして最後に3rdグループに属する話題を採り上げればよいことを示している。

また、表2においてPタイプとSタイプとあるのは、グループ内での話題の遷移方法の違いを示している。例えば、1stグループに属する話題では、Whatに属するすべての話題が終了しないうちにExamples, Why, Current systemなどへ話題を遷移してもよいが、2ndグループへ話題が遷移するのは1stグループに属するすべての話題が終了した後でなければならない。即ち、あるグループ（カテゴリーの集合）に属するすべてのカテゴリーのすべての話題が終了しない間は、そのグループに属さない他のカテゴリーへ話題を遷移してはならない。このような話題の遷移方法をPタイプと呼ぶ。一方、2ndグループに属する話題では、このグループに属するどのカテゴリーから話題を採り上げてよいが、一度採り上げたらそのカテゴリーに属するすべての話題が終了するまでは、他のカテゴリーへ話題を遷移することはできない。例えば、Policyに関するすべての話題が終了してからでないと、Constraintsに属する話題を採り上げることは許されない。このような話題の遷移方法をSタイプと呼ぶ。

なお、表2で、Conditions, Constraints, Policy からなるグループはSタイプとして見なされ、そのように登録しているが、これらをPタイプとする考え方もある。また、2ndグループと3rdグループでは順序が逆のほうがよいという考え方もある。このため、これらを組み合わせた4種類をシステムのシナリオとして用意する。さらに、シナリオ作成に際しては種々の考え方があるので、SEが自由に作成・登録できるように、シナリオの作成/登録/修正/削除する機能を設ける。

2.2 進捗管理テーブルによる進捗管理支援

インタビューによる要求抽出において、そもそも議論されなければならない事柄が9つのカテゴリーに分類でき、要求抽出のためにインタビューで採り上げる話題が、これらのカテゴリーの間をどのように遷移すべきかを規定するシナリオがあれば、要求抽出作業の進捗を管理できる筈である。即ち、シナリオに沿ってインタビューを進めるとともに、カテゴリーごとの進捗を管理すれば、シナリオに沿ってどこまでインタビューが進んだかを管理することができる。そこで、インタビューによる要求抽出作業の進捗を管理するために、表3のような進捗管理テーブルを用意する。

表3 進捗管理テーブル

カテゴリー	順序	タイプ	進捗
What	1 st	P	2
Examples	1 st	P	2
Why	1 st	P	2
Current system	1 st	P	2
Conditions	2 nd	S	2
Constraints	2 nd	S	1
Policy	2 nd	S	0
Budget	3 rd	P	0
Schedule	3 rd	P	0
全体の進捗	—	—	5/9

[状態] 0 : 未着手 1 : 議論継続中
2 : 議論終了 3 : 議論の必要なし

これは表2のシナリオに対して、各カテゴリーの進捗の欄が、0であればそのカテゴリーの議論が始まっていない、1であれば議論が継続中である、2であれば議論が終了した、3であれば初めから議論の必要がないという、カテゴリーごとの進捗管理情報を加えたものである。従って表3の例では、1stグループの進捗の値がすべて2なので、1stグループの議論はすべて終了した。2ndグループの進捗の欄を見ると、Conditionsが2でConstraintsが1なので、現在2ndグループにおけるConditionsに属するすべての話題が終了し、次のConstraintsに話題の議論を継続中であることが判る。また、9つあるカテゴリーのうち、すべての話題が終了したカテゴリー（進捗の欄の値が2であるカテゴリー）が5つあるので、インタビュー作業全体での進捗は5/9であることも判る。なお、各カテゴリーにおける進捗状況はSEのリーダーが判断し、進捗管理テーブルに情報をセットする。従って、表3のような進捗管理テーブルを用いれば、要求抽出作業の進捗を発言内容のレベルで

管理することができる。同様の考え方に立った技術として、知的CAIにおけるオーバーレイ・モデルがある。このモデルでは、学習者が修得した知識は教材作成者の知識の部分集合であると見なし、学習者の学習の達成度を把握する。従って、上記の進捗管理テーブルは、オーバーレイ・モデルを拡張したものと言えよう。

2.3 ナビゲーションメッセージの生成方法

経験のない人でも、ユーザの要求を正しく、かつ、遺漏なく、しかも効率良く抽出できるようにするためには、次に何をすべきかを示すナビゲーション・メッセージを生成する必要がある。

では、どのような考え方でシステムがナビゲーション・メッセージを与えるか、以下に例を挙げてその考え方を示す。

(1) 質問開始時における誘導

「開発しようとするシステムは、一言で言えば、何を

するシステムですか？」
上記は、インタビューの最初にソフトウェア技術者から発せられるべき質問である。逆に、この一言がないとインタビューを開始できない。何故なら、インタビューにより要求を抽出しようとするとき、開発しようとするシステムの機能を、大雑把に把握することから始めるのが最も効率が良い、と考えられるからである。しかし、インタビューの初めであればこの質問が常に成立するかというと、必ずしもそうとは言えない。例えば、インタビューに先立ち、開発すべきシステムの概要について、予め顧客から伝えられていたような場合には、上記のような質問はできない。この場合には、顧客が開発しようとするシステムの概要を一言で言い表わし、例えば「～するシステムを開発したいと伺いましたが、もう少し詳しくお聞かせ願えませんか？」というような言葉で切り出すことになる。どちらの質問が適切かは、インタビューするときの状況によって異なるので、ナビゲーションシステムには判断できない。かといって、どちらの質問が適切かの判断を常にインタビューにさせるのは、インタビューによっては負担になることもある。そこで、どちらの質問が適切かをナビゲーション・システムが判断できるように、状況判断に必要な情報をチェックリストの形で用意しておき、その都度、それをインタビューにマークさせる。そして、マークされたチェックリストの情報を基に、ナビゲーションメッセージをシステムが生成するという方式を採用する。即ち、システムがインタビューの文脈を理解して、ユーザが次に何をすべきかを示すメッセージを生成するという方式は採らない。インタビューにおける文脈の理解はインタビューである人間に任せ、インタビューが自分の理解に基づいてマークしたチェックリストの情報を基に、システムが必要なメッセージを生成するという方式を採用する。

(2) 進捗管理テーブルに基づくメッセージの生成

カテゴリーがPタイプかSタイプかによって、話題の遷移方法が異なるので、メッセージ出力のタイミングも

異なってくる。カテゴリがPタイプの場合には、新しいグループへ遷移するときのみ、そのグループの議論を促すメッセージを出力する。表3の例で言えば、1stグループに入るとき、2ndグループから3rdグループへ遷移するとき、の2つである。このうち、1stグループに入るときに出力するメッセージについては、(1)の質問開始時における誘導で説明した。2ndグループから3rdグループへ遷移するときには、3rdグループのカテゴリがBudgetとScheduleなので「ソフトウェア開発のための予算と開発期間について、お考えをお聞かせ下さい。」というメッセージを出力する。

一方、カテゴリがSタイプの場合には、新しいカテゴリに遷移するときのみ、そのカテゴリの議論を促すメッセージを出力する。このとき出力するのは、カテゴリごとに用意されたカテゴリ固有のメッセージである。例えば、Conditionsカテゴリへ遷移するときには、「設計条件について議論したいと思います。設計条件の例をこちらで用意致しましたので、~をご参照下さい。」というメッセージを出力する。

(3) 発言のタイプに基づくメッセージの生成

顧客とSEの発言内容から見て、インタビューにおけるその発言の役割を分類したものを**発言のタイプ**と呼ぶ。発言のタイプは下記の15種類である。

質問、回答、確認、提案、補足、訂正、資料提供、資料提供依頼、参考、動議、評価、決定、作業依頼、約束、その他

次に遷移する発言のタイプが確定する場合にのみ、その発言のタイプに基づき、次の発言を誘導するメッセージを生成する。

発言のタイプは、顧客およびSEが発言する際に、自分の発言がどの発言のタイプになるかを自ら判断し、メニュー上のそれを選んだ上で発言させるようにする(スクリーン画面上の入力フィールドを工夫して、発言のタイプが単一になるようにする)。顧客にこのような操作をさせるのは、顧客の負担を増大させるのでよくないと考える向きもあるが、顧客およびSEに自ら発言のタイプを選ばせることにより、1回1回の発言がシンプルになるという効果がある。

2.4 カテゴリ個別の支援機能

これまで述べてきた誘導の方法は、顧客の発言を促すメッセージは出すものの、その姿勢は顧客の提案をひたすら待つというものであった。これではインタビューは長くは続かない。何故なら、類似するソフトウェアを過去に開発したり使用したりしたことのない人が、開発予定のソフトウェアの仕様に対する自らの要求を正確に、かつ、遺漏なく抽出することは困難だとされているからである。しかし、SEが具体案を呈示するとともに、顧客がそれを受け容れてもよいか否かを判断するのに必要な説明と情報を提供した上で、顧客の判断を仰ぐという行動をとれば、このような顧客でも的確な判断を下せることが知られている。SEがとる上記のような行動は、謂所、提案型のSEの行動として広く知られている。

提案型のSEがとる行動を支援するために、カテゴリごとの支援機能を用意する必要がある。

(1) 1stグループの場合

1stグループでのインタビューの目的は「開発しようとするソフトウェアで必要となる機能」を列挙し、列挙された機能の1つ1つを吟味することである。機能を列挙するために有力な方法は2つある。1つは、現行のシステムに対する顧客の不満を聞き出すとともに、このシステムの情報を入手し、これを理解した上で、現行のシステムのどこをどのように改良するかを具体的に提案することである。このため、現行システムに対する顧客の不満を聞き出す作業を誘導する機能が必要となる。もう1つは、ソフトウェア領域ごとに必要と思われる機能をデータベース化して蓄え、インタビューの過程でそれを顧客に呈示して、その中から必要なものを顧客に選ばせることである。このようなことを実現するためには、本稿で提案するシステムを利用する度に、要求仕様書の形でその事例が蓄積され、インタビューの過程で事例検索できるような機能を実現する必要がある。

(2) 2ndグループの場合

Conditions, Constraints, Policyの違いを説明するとともに、それらの事例を顧客に呈示して、その中から必要なものを選ばせる機能を用意する。

(3) 3rdグループの場合

ソフトウェア開発のための予算と期間を考える上で、目安となる情報をSEに呈示して、インタビューの過程で参照できるようにする。

3. その他の支援機能

現場のソフトウェア開発プロセスと、当研究で開発するツールの支援機能との対応関係を図3に示す。

現場でのプロセス	当研究 (現場でのプロセスをそのまま支援)	支援機能の例
多人数による要求抽出 (結果の記述はinformal)		インタビューによる要求抽出 ・ナビゲーション機能 ・要求抽出作業の進捗管理
要求相互における 重複や矛盾のチェック	左記の作業を随でもできる ように計算機で自動的にガイド	重複や矛盾の検出作業ガイド
グループでの合意に 基づく仕様の決定		グループレビューへの支援 意思決定支援
多人数による要求仕様書 (informalな記述)の執筆	要求仕様書(informalな 記述)の自動生成	要求仕様書のスタイルファイル 要求仕様書の生成規則
(これより設計過程)		
要求仕様書に基づき 設計仕様書などを 多人数で分担して執筆	GroupWork Managerで 支援済み	設計提案とそれに対する グループ討論を支援

図3 開発現場での要求分析過程とそれへの支援

2章で述べた諸機能は、図3に示した現場でのプロセスを支援する機能である。

3.1 獲得した要求仕様を途中で確認する機能

インタビューの途中または終了時で、現時点までに獲得した、要求仕様を画面(ハードコピー可)に呈示する機能

である。この機能を実現する方式をその処理手順に従って以下に示す。

(1) カラーリング

メッセージの形で送られてくる要求項目を3.1節で提案した9つのカテゴリに分類し、その分類に基づいて割り当てられた記憶場所に、要求項目のそれぞれを振り分けてデータベースに記録する（この一連の処理を**カラーリング**と呼ぶことにする）。

カテゴリ分けされた個々の要求項目を記録するための内部データの記録形式も決定した。これを**R-EAD (Requirements Elicitation/Analysis/Definition) モデル**と呼ぶことにした。そして、この分類カテゴリや記録フォーマットを基に試作システム (Version 1) を構築した。

(2) カラーリングで記録された情報を基に、カテゴリまたはその集合であるグループごとに編集して画面に呈示する。

3.2 複数人の中で意見交換する過程をreplayする機能や任意の時点へ戻る機能

この機能は要求抽出過程を分析する際に必要となる。特に複数の人間が発言している場合に有効である。

3.3 グループでの意思決定支援機能

インタビューに参加した顧客のそれぞれが異なる意見を持っていたために、意見がなかなか一致しないことがある。そのような状況は様々な要求を引き出すためには好都合ではあるが、要求仕様書をまとめるためには意見の一致をみなければならない。このため、如何にしてそれらの意見を仕様としてまとめるかということが問題になる。このため、Saatyの**階層分析法 AHP (Analytic Hierarchy Process)**[11] をグループでの意思決定に応用してこの問題の解決を図る。

3.4 WWW上での黒板機能

黒板は参加者全員が同時にそれを見ることのできる共通の画面であり、予め参加を認められた人であれば誰でも自分の意見を黒板に書き込むことができる。しかし、それが許されるのは、発言を認められたときだけである。要求仕様書をレビューしたり、複数の人達の意見を調整したりするときには、関係者全員で同じ画面を見たほうが効率が良い。このため、個人別の利用しか考えていないWWWにおいて、黒板機能の実現を目指す。この場合、同一画面へ複数の人がアクセスすることになるので、そのことによって生じる黒板機能実現のために実現すべき項目は次のとおりである。

- ①画面切り替えの際の、複数の利用者画面の同期表示機能
- ②発言者が複数存在する場合の、画面やファイルへの書き込みに関する排他制御機能

3.5 IEEE Std 830-1993に基づく要求仕様書執筆支援機能

IEEEが推進している標準化作業の成果に、**IEEE Std 830-1993**[1]がある。これは、要求仕様書に記載すべき項目の、目次案とその記述内容についてまとめたもので、**ソフトウェア要求仕様書の記載項目に関するIEEEの推**

奨案とも言うべきものである。本稿の付録にその概要を示す。

IEEE Std 830-1993の内容を読むと判るように、この推奨案に沿って要求仕様書を記述しようとするれば、インタビューによる要求抽出だけで記述できるのは、この推奨案が要求する記述項目のうちの、ほんの一部分だけである。

(しかし、我々は要求仕様書を作成する上で最も困難な部分から支援しようとしているのである。)何故なら、IEEE Std 830-1993が推奨する記述内容は、開発しようとするソフトウェアに対する顧客の要求に関する部分よりも、次の工程で行う設計作業に対する条件書のような内容を記述する部分のほうがずっと多いからであり、これらの内容はSEが自ら提案し決定するものが殆どだからである。従って、要求仕様書の作成過程全体を支援するには、インタビューでは得られない記述項目に対して、2, 4節で述べた、カテゴリ別の支援機能と同様に、SEが設計条件などを自ら決定する作業を支援する機能が必要である。

3.6 要求仕様書執筆作業用進捗管理テーブル

ところで、要求仕様書の記載項目に、IEEE Std 830-1993のような推奨案があるということは、「ソフトウェアの要求仕様を定義するときに、そもそも明らかにされなければならない幾つかの項目があるということであり、それらは目次案という形で、その内容を幾つかの項目に分類できる」ということである。しかも、それらは推奨に叶うものであるから、その目次項目は、特殊な場合を除き、想定される殆どの場合を含んだものになっている筈である。とすれば、それらは要求仕様書に記載すべき項目を分類するためのカテゴリとしても利用できることになる。従って、この分類カテゴリを利用すれば、要求執筆作業の進捗状況を管理できる筈である。従って、IEEE Std 830-1993の推奨目次案を基に要求執筆作業の進捗管理テーブルを作成することができる。

3.7 要求仕様書の自動生成機能

それまでに抽出された要求を要求仕様書の形で自動生成する機能である。要求仕様書を自動生成するためには、IEEEの推奨する要求仕様書の目次案[1]に合わせてスタイルファイルをSGML (Standard Generalized Markup Language) で作成しておくとともに、予め抽出しておいたルール群を使って要求仕様書を自動生成する方式を採用する。

4. おわりに

WWWを使用し、SEが顧客にインタビューすることにより要求を抽出する作業を誘導するシステムの実現方式を明らかにした。

[参考文献]

- [1]IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1993.
- [2]Julio Cesar Sampaio do Prado Leite and Ana Paula Pinho Gilvaz: Requirements Elicitation Driven by Interviews: The Use of Viewpoints, Proceedings of Eighteenth International Workshop on Software Specification and Design (IWSSD-8), pp. 85-94

(1996).

[3]加藤潤三：要求間の制約分析に基づく要求定義手法の適用事例，電信技報KBSE95-16，pp.9-15 (1995).

[4]加藤潤三，古宮誠一，米田英剛，水野俊平，永田守男，山本修一郎：ソフトウェア要求分析法とインターネット上で構築するその支援環境，(インタビューによる要求獲得の課題と提案)，電信技報KBSE-96-17，pp.17-25 (Sept. 26, 1996).

[5]古宮誠一：インタビューによるソフトウェア要求分析過程とWWW上に構築するその支援環境，第16回技術発表会論文集，情報処理振興事業協会，pp.19-26 (1997).

[6]森田賢二，永田守男：情報系システムを対象とした視覚的表現法による要求獲得支援環境の提案とその実現法，経営学会誌，Vol.5，No.2，pp.1-22 (1996).

[7]Nagata, M. et al: Computer Supported Meta-communication Analysis of Meetings, Journal of Organizational Computing and Electronic Commerce, (1997).

[8]Russell, S. & Norvig, P.: Artificial Intelligence -A Modern Approach-, Prentice Hall, (1996).

[9]Saeki, M., et al: Structuring Utterance Records of Requirements Elicitation Meetings Based on Speech Act Theory, ICRE'96, pp.21-30 (1996).

[10]忠海均，上野正巳，山本修一郎：システム要求分析における会話特性の抽出実験，電信技報KBSE94-39，pp.57-64 (1994).

[11]刀根薫：ゲーム感覚意思決定法，日科技連出版社，(1986).

[12]山本修一郎，上野正巳：既存システムの改造における要求プロセスの評価実験，電信技報KBSE95-17，pp.17-22 (1995).

[13]山本修一郎，忠海均，上野正巳：ドメイン指向要求分析技術：DREM, NTT R&D, Vol.45, No.8, pp.711-718 (1996).

[14]上野正巳，忠海均，山本修一郎：要求分析方法論DREMの基本概念，電信技報KBSE95-5，pp.33-38 (1995).

付録 (IEEEによる要求仕様書記載項目の推奨案)

IEEEが推進している標準化作業の成果に、IEEE Std 830-1993[1]がある。これは、要求仕様書に記載すべき項目の、目次案とその記述内容についてまとめたもので、ソフトウェア要求仕様書の記載項目に関するIEEEの推奨案とも言うべきものである。以下にその概要を示す。

目次

1. はじめに

1.1 目的

- a) この仕様書の目的
- b) この仕様書が対象とする読み手の規定

1.2 適用範囲

この仕様書およびこの仕様書で規定するソフトウェアの適用単位について記述する。

- a) ソフトウェア製品名
- b) ソフトウェア製品の説明

c) ソフトウェアの適用対象を詳細に利点、目的および目標を含めて記述する

d) ハイレベルな仕様書(例えば、システム要件)がある場合、ハイレベルな仕様書にある類似の説明と矛盾がないように記述する。

1.3 定義、頭字語および略語

この仕様書の記述に必要な用語、頭字語および略語すべてを定義する。

1.4 参考文献

- a) SRS以外の参考文献一覧
- b) 各文献の表題、文献番号、日付および出版社
- c) 文献を入手できる出所

1.5 要求仕様書(SRS)の構成

- a) SRSの「はじめに」以外の項目に何があるか
- b) SRSの編成がどのようになっているか

2. 全体説明

2.1 製品の位置づけ

この仕様書に記述されるシステムが、より大きなシステムの一部であれば、それとの位置づけやインタフェースなどをそれとの関連で記述する。もし、独立のシステムであれば、そのことが判るように記述する。

- a) システム・インタフェース
- b) ユーザ・インタフェース
- c) ハードウェア・インタフェース
- d) ソフトウェア・インタフェース
- e) 通信インタフェース
- f) メモリ制約
- g) 操作
- h) 場所適応要求

2.2 製品の機能

2.3 ユーザの特性

SRSの3章で具体的に述べる特定の要求の理由を与えるために、教育的水準、経験、及び専門技術を含む、製品のユーザの一般的な特長を述べる。

2.4 制約

- a) 規制上の政策
- b) ハードウェアの制限
- c) 他アプリケーションとのインタフェース
- d) 並列オペレーション
- e) 内蔵機能チェック機能
- f) 制御機能
- g) 高次元言語の要求
- h) 信号ハンドシェイクプロトコル
- i) 信頼性の要求
- j) アプリケーションの重大度
- k) 安全と機密保護の配慮

2.5 前提と依存性

SRSに述べられた要求に影響を与える個々の前提を列挙する。前提が満たされないときは、SRSはそれに従って変更される。

2.6 システムの今後のバージョン計画

3. 仕様要件

ここでは、次のようなレベルまで詳細化した内容のソフトウェア要求を記述する。

- ・ これらの要求を満たすように設計者が設計できる。
- ・ これらの要求をシステムが満たすことを、テスト担当者がテストできる。

3章はSRSの中で最も大きくかつ最も重要な部分であることが多いため、次の原理を応用する。

- 特定の要求は、後で述べる「良い要求仕様書の特性」に従って述べられている。
- 特定の要求は、関連する初期の文書と相互参照可能である。
- すべての要求はユニークに識別できる。
- 要求仕様書の構成に配慮し、読みやすくする。

3.1 外部インタフェース

3.2 機能

3.3 性能要求

3.4 データベースへの論理的な要求

3.5 設計上の制約

3.5.1 標準や規格の遵守

3.6 ソフトウェアシステム属性

ソフトウェアには要求を満たすための多くの属性がある。要求された属性が達成されているか否かを客観的に確かめられるように、定義されることが重要である。以下に例を示す。

3.6.1 信頼性

3.6.2 可用性

3.6.3 安全性

3.6.4 保守性

3.6.5 移植性

3.7 要求仕様の構成

理解しやすい最適な方法で要求仕様を構成するよう注意深い配慮が必要である。要求仕様書の第3章において、システム内の異なるクラスでは異なった仕様定義となる。その構成のいくつかを以下に示す。

3.7.1 システムモード

あるシステムは全く違った操作モードで運用されている。ある制御システムではそのモードに依存した別々の機能セットとなるであろう。例えば、訓練時、正常運転時、緊急時というような異なる機能セットを持つ。

3.7.2 ユーザクラス

あるシステムはユーザが異なったクラスでは異なる機能セットを提供する。

3.7.3 オブジェクト

そのオブジェクトの属性と（そのオブジェクトによって行われた）機能のセットがそれぞれのオブジェクトと結び付けられる。あるシステムはオブジェクトの属性と機能のセットを記述することによって最も良く構成することができる。

3.7.4 システム特性

システム特性のそれぞれは、一般に、連続した外部入力と応答の対で記述される。あるシステムはシステム特性を記述することによって最も良く構成することができる。

3.7.5 外部入力

あるシステムは外部入力を使って機能を記述すること

によって最も良く構成することができる。

3.7.6 応答

あるシステムは応答を生成するすべての機能を記述することによって最も良く構成することができる。

3.7.7 機能階層

上記の構成のいずれが当てはまるか分からない時、全体的な機能は通常の入力、出力、あるいは内部のデータアクセスによって構成された機能階層で構成することができる。データフロー図とデータ辞書が機能とデータの間の関係を示すために使用できる。

3.7.8 追記

新しい要求仕様(SRS)を考えると、3.7.7 までで示した構成化技術の2つ以上が当てはめられることがある。このような場合には、詳細なシステムの要求仕様に合わせて多数の要求仕様を構成する。どのような追加要求も要求仕様書の最後に別の章として追加される。

付録
索引

—以上—