

ゴール指向要求分析に基づく 組込みボードエミュレータ開発効率化手法の考察

ジリエ陽子^{†1†2} 本田耕三^{†2} 中川博之^{†2} 田原康之^{†2} 大須賀昭彦^{†2}

本論文では、製品開発初期から導入することで開発効率を向上させるための要求分析モデリング手法を提案する。エミュレータを用いると、組込みソフトウェアを PC 上で動作させることができる。専用の組込みボードなしで開発可能であるため、実機では難しい故障模擬テストや、多数の開発メンバーへの最新テスト環境配布ができ、さらにコスト削減にもつながる。しかし、エミュレータは工数や機能に幅があり、利用目的によって大きく異なるにも関わらず製品開発部門よりの確に要求抽出することは簡単ではない。どのような機能がいつまでにできるかを踏まえて要求仕様の収束を行う必要がある。本研究では KAOS 要求分析モデルをベースに、情報可視化手法を用いて要求分析モデルを拡張した。複数の既存エミュレータ開発案件の要求仕様やノウハウを統合し、エミュレータの利用目的によってどのように工数が変わるか確認できるようにした。また、新たに決定しなければならない仕様や確認が必要な仕様を明確化した。

Considerations on an Embedded Board Emulator Development Efficiency Method Based on Goal-oriented Requirements Analysis

YOKO GIRIER^{†1†2} KOZO HONDA^{†2} HIROYUKI NAKAGAWA^{†2}
YASUYUKI TAHARA^{†2} AKIHIKO OHSUGA^{†2}

This paper proposes a requirements analysis modeling method to introduce from the beginning of product development in order to promote development efficiency. Using an emulator, it is possible to execute embedded software on computers. Development without embedded boards enables hardware failure tests, and distribution of newest testing environment to many development actors, which is difficult with actual hardware. Furthermore, it leads to cost saving. Man-hour and functions are various depending on the purpose of the emulator. However it is not easy to extract requirements properly from product development division. It is necessary to make requirements specifications converge based on functions and man-hour. In this research we extended requirements analysis models using information visualization methods and KAOS requirements analysis models. We integrated requirements specifications and know-how of multiple emulators development cases and made it possible to verify how man-hour changes depending on the purpose of the emulator. Also, we emphasized new specifications to decide or verify.

1. はじめに

機能や性能、信頼性に対する要求から、監視制御システムでは、専用の組込みボードを用いることが多い。これらの組込みボード上で動作する組込みソフトウェアの開発規模は年々増大しており、かつ短期間での開発も重要である。信頼性の高いソフトウェアを短期間で開発するには、効率的な開発環境が必要である[4]。そこで、専用組込みボードの完成を待たず、組込みソフトウェアを PC 上で動作させるためのソフトウェアとして、エミュレータを用いた開発環境を検討する[2]。エミュレータは組込みボードの動作を模擬するため、組込みソフトウェアが組込みボードなしでも動作する。組込みソフトウェアが PC 上で動作できると、実機の無い環境で実機の故障のテストなど各種テストが行える。開発に必要な実機の個数を減らせるため、コスト削減にもつながる。海外開発拠点を含む多数の開発メンバーに最新のテスト環境を配布できるメリットもある。しかし、監視制御システムの寿命は長く、20~30年以上稼働してい

るシステムの場合、設計書や仕様書のドキュメント類があまりなく口頭で伝達していることが多い。組込みボードの仕様が明確でない場合、エミュレータ開発部門ではまずハードウェアの仕様やエミュレータの利用目的を明確にしてから設計や工数見積もりを行うことになる。エミュレータは工数や機能に幅があり、何をしたいのかによっても設計は大きく異なるにも関わらず、製品開発部門よりの確に要求抽出することは簡単ではない。一方、製品開発部門はエミュレータがいつ完成して、何ができるかがわかりにくいことも要求仕様を収束させづらい原因となっている。

本研究では製品開発初期からエミュレータ開発部門および製品開発部門が共同で検討することで製品開発効率を向上させるための要求分析モデリング手法を提案する。品質の高い要求仕様を作成するためにゴール指向要求分析手法 KAOS[6]を適用する。限られた開発期間で余分なものを作らず、かつタイムリーに必要なものを提供できるように、製品開発部門とエミュレータ開発部門で認識合わせを行うツールとして活用できるようにする。

本論文の構成は以下の通りである。第2章では、エミュレータについて概説する。第3章では KAOS 指向要求分析モデルについて述べる。第4章ではエミュレータ開発の課

†1 (株)東芝 ソフトウェア技術センター
Toshiba Corporation, Corporate Software Engineering Center
†2 電気通信大学
The University of Electro-Communications

題と要件について述べる。第5章では、本研究提案手法の詳細について述べる。第6章では、本研究提案手法のエミュレータへの適用を行う。第7章では従来研究について述べる。第8章では、結論と今後の課題について述べる。

2. 組み込みソフトウェア開発のためのボードエミュレータ

エミュレータとは、コンピュータや機械などのハードウェアを模擬するソフトウェアである。

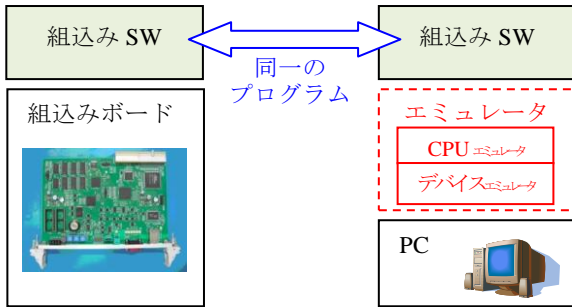


図 1 組み込みソフトウェア動作環境

Figure 1 Embedded software operation environment.

ハードウェアの設計の違いを吸収するため、元のハードウェアの“ふり”をする。専用の組み込みボード上で動作する組み込みソフトウェアの場合、エミュレータが組み込みボードの動作を模擬することにより、組み込みソフトウェアを PC 上で動作させることができる。組み込みボードは、CPU と周辺デバイスから構成される。周辺デバイスは、例えばタイマ、メモリ、ネットワークなどである。エミュレータを短期間で CPU やデバイスの進歩に追従させることは困難である。しかし近年では、QEMU[9]などオープンソースソフトウェアのエミュレータが公開されており、最新ハードウェアに対応するエミュレータを容易に入手できるようになった。表 1 は QEMU の概要を記載したものである。オープンソースソフトウェアのエミュレータをベースとして、CPU エミュレータとデバイスエミュレータを組み合わせ、組み込みボードのエミュレータを構成することができる。ここで、対象システム独自のデバイスについては、デバイスエミュレータを必要に応じて開発する。

表 1 QEMU 概要

Table 1 Summary of QEMU.

項目	説明
主な機能	異種 CPU エミュレータ。プロセッサや各種の周辺装置を含めたすべてのシステムをエミュレートできる。
サポートしているホスト OS	BeOS, FreeBSD, Linux, Mac OS X, Windows, NetBSD, OpenBSD, Solaris, ZETA
サポートしているターゲット CPU	Alpha, ARM, CRIS, i386, LM32, M68K, MicroBlaze, MIPS, OpenRISC, PowerPC, s390x, SH4, SPARC, UniCore32, Xtensa
プログラミング言語	C, Python
プログラム規模	1043950 ステップ (※コメント・空白行を除き、C および Python 有効行のみ)
ライセンス	BSD, GPL, LGPL(※コンポーネントによって異なる)
最終更新日	2012/09/05 (※2012/10/29 現在)
最新バージョン	1.2.0
URL	http://wiki.qemu.org/Main_Page

3. KAOS 要求分析モデル

要求分析はソフトウェア開発で最初にすべき、何を作りたいのか明確にさせることである。すべての要求には、なぜその要求が必要かという理由がある。逆に言えば、理由のない要求は機能化してはいけない。しかし、要求仕様書に理由が記載されることはあまりない。ゴール指向要求分析はニーズをシステムが達成すべき目標、すなわちゴールととらえ、なぜ・どうやって達成するのか、という観点で分析する要求工学の重要なアプローチである。

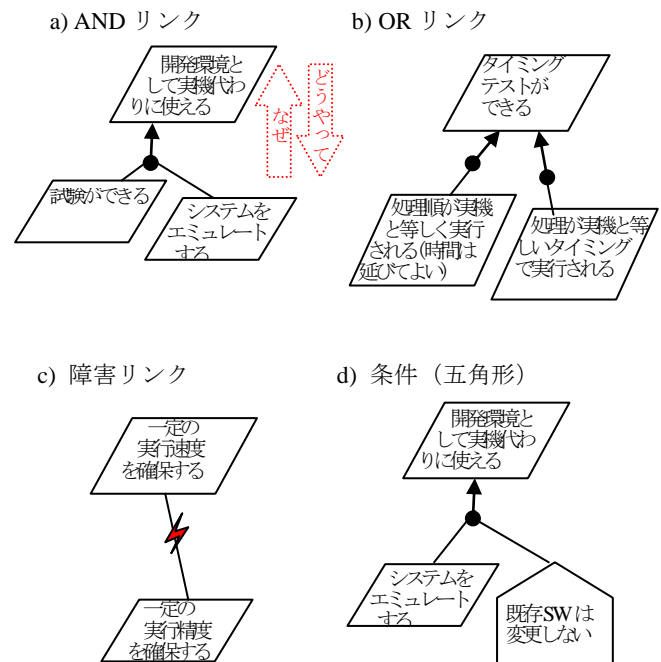


図 2 ゴールモデル表記法

Figure 2 Goal model notation.

KAOS はゴール指向要求分析の代表的な分析手法のひとつである。システムゴールを 1)目標状態, 2)システムが達成責任を持つ操作要求と, 3)環境や人が達成責任をもつ仮説の 3 種類に分類し, 系統的に分析する。本論文では KAOS の要求分析モデルとして, ゴールモデルとオブジェクトモデルを用いる。ゴールモデルは, 機能的ゴール, 非機能的ゴールを AND/OR リンクと障害リンクにより結合し構造的に表現する。図 2 は, それぞれ a) AND リンク, b) OR リンク, c) 障害リンクを用いた例である。なお, 条件は d) の 5 角形のように表示する。

本研究では, エミュレータ開発における CPU の種類, デバイスの種類といった対象をオブジェクトモデルとして表現する。次図 3 はオブジェクトモデルの記載例である。オブジェクト間の継承関係は ISA 関係として表現され, UML のオブジェクト図に類似している。

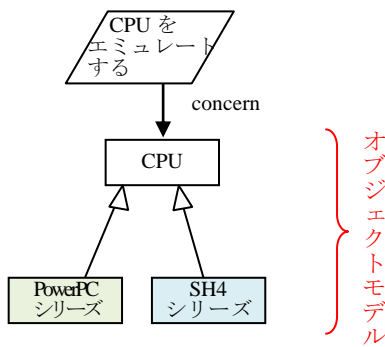


図 3 オブジェクトモデル
 Figure 3 Object model.

4. エミュレータ開発の課題と要件

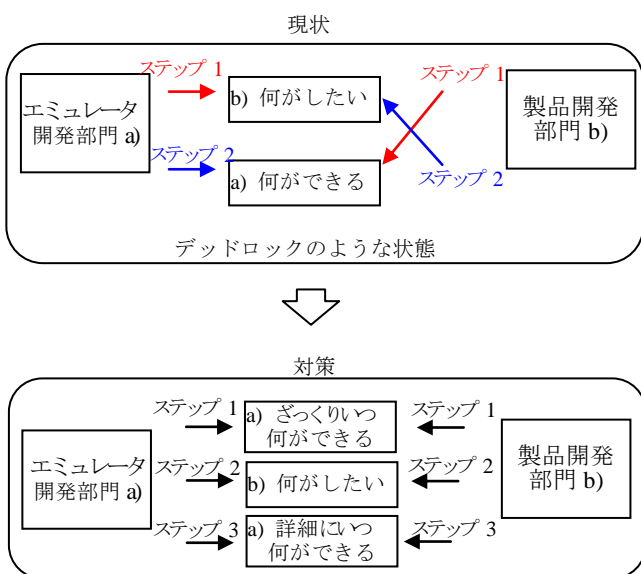


図 4 エミュレータ開発における現状と対策
 Figure 4 Current Status and Action for Emulator development

エミュレータの要求仕様は非常に複雑である。エミュレータで何がやりたいのかによって工数や機能は大きく異なってくる。工数や機能の幅, 仕様決めでネックとなるフィージビリティ調査, これらを可視化して要求仕様を収束させる必要がある。図 4 はエミュレータ開発における現状と対策を示している。現状の開発は, エミュレータ開発部門 a) はまず製品開発部門 b) は何がしたいのか知ってから何ができるか提示したい, 一方 b) はまず a) が何がしたいのか知ってから a) に何をやってもらうか考えたい, といったデッドロックのような状態である。エミュレータ開発における一番の問題は, 製品開発部門の要求をうまく聞き出せないことである。要求を的確に聞き出せない原因としては, エミュレータをどのように使うと製品開発に最も効果的か, またそのエミュレータはいつ提供可能か明確に提案しづらいことである。エミュレータがいつ提供可能か明確にしづらい, すなわち工数見積りが困難である理由は, 要求抽出が簡単ではないこと, およびハードウェアの仕様そのものに起因する。まず, エミュレータが動作を模擬する元のハードウェアである, 組込みボードの仕様が明確ではない。制御システムの初代バージョンが 20~30 年前に開発されている場合, ハードウェアにどうアクセスしているかドキュメント類があまりなく, 熟練技術者のノウハウの口頭による伝達が主になっている。他に, 組込みボードの開発段階で要点整理ができていない場合もある。すなわち, ハードウェア設計時に何をやらなければならないか明確になっていない。例えば, 組込みボードで新たに選定した CPU が, 開発が進むにつれて要件を満たさないことが明らかになり, CPU の選定からやり直しが必要なこともある。結果としてソフトウェア全般のスケジュールがさらに遅れる状況となっている。

対策として図 4 のように, まずエミュレータ開発部門 a) から製品開発部門 b) へこれまでのエミュレータ開発経験をベースにざっくりいつ何ができるか提案する。次に b) から何がしたいか聞き取る。その後改めて詳細にいつ何ができるか a) から b) へ提案することによりデッドロックのような状態を抜け出せるものと思われる。そこで, エミュレータ開発としての工数見積りをしやすくすること, およびエミュレータの効果をわかりやすく提示できることが望ましい。例えば, すべての機能はサポートしなくとも, ボトルネックとなる機能はこの程度の工数で実現可能などである。そのため, エミュレータの開発ノウハウをまとめると同時に, エミュレータ開発にあたってどのような情報をヒアリングする必要があるか明確にする必要がある。しかし, ゴール指向要求分析手法 KAOS をベースとした複数の製品開発案件を統合して管理する要求分析モデル[1]では, 工数を算出したり, 新たに決めなければならない仕様を明確にしたりするための表現手段がない。そこで, 次の 3 点に着目し 5 章で述べる拡張を行った。

- 1) エミュレータでできることとその工数がわかりやすいこと
- 2) CPU, デバイス, ボードの構成やデバイス動作仕様などヒアリングすべき項目がもれなく確認できること
- 3) フィージビリティ調査の必要な要求を明確にすること

5. 要求分析モデルの拡張

本研究では、制御ボードの選定など製品開発初期から製品開発部門とエミュレータ開発部門でハードウェアの仕様を確認しながら認識合わせを行い、エミュレータに関する要求仕様を収束させることで、製品開発効率を向上させるための要求分析モデリング手法を提案する。要求をもれなく分析し、品質の高い要求仕様を作成するためにゴール指向要求分析手法 KAOS を適用する。KAOS をベースに情報可視化手法を用いて複数の既存エミュレータ開発案件の要求仕様を統合する (図 5) [1]。統合した要求仕様やノウハウを、新規開発案件の制御ボードの選定から共同で検討し工期の見積もりを行うために有効な分析図となるよう次の手順でまとめる。

- 1) 既存の開発案件について、共通部分と固有部分を明記した要求分析図を作成する。固有部分は色分けして区別する。

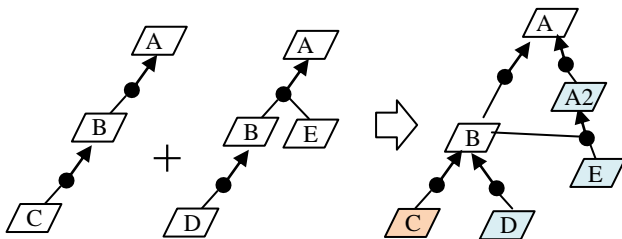


図 5 要求仕様の統合

Figure 5 Integration of requirements specifications.

- 2) 1)で作成した要求分析図に工数を記載する。
- 3) 新たに開発が必要な項目や条件を含めた表示にする。
- 4) 工数見積りの前にフィージビリティ調査が必要な項目を明確にする。

5.1 工数の表示

各要求に対する概算工数を表示し、トータルの工数見積りに役立てる。工数に幅がある場合は最小値と最大値を表示する。親ゴールは子ゴールの工数を合算して表示する。なお、本論文記載例では工数は単体試験までを想定し、単位は人日で表記する。なお技術者のレベルはエミュレータ開発経験者を前提としている。

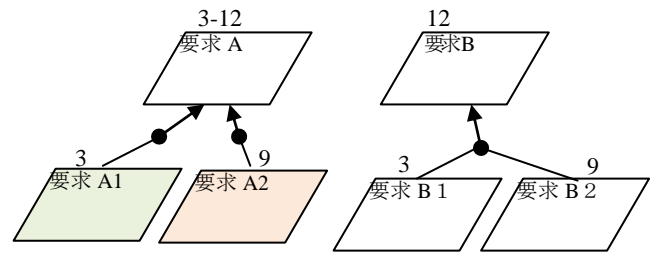


図 6 工数の表示

Figure 6 Description of man-hour.

5.2 新規開発項目の表示

開発必須の要求仕様については、既存項目に当てはまらない場合、新規開発項目を色分けし、工数と共に明確化して検討もれを防ぐ。

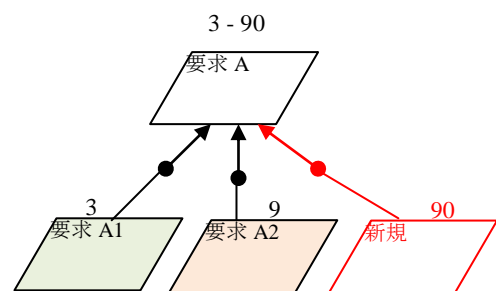


図 7 新規開発項目の表示

Figure 7 Description of new development items.

5.3 条件つき工数の差分表示

新規開発項目については、工数に違いがある場合は、条件とともに工数を次のように表示する。

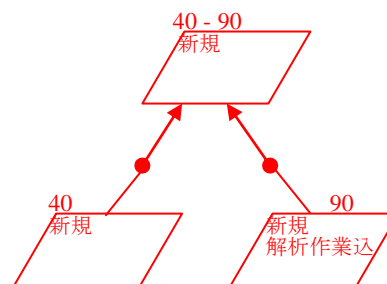


図 8 条件つき工数の表示

Figure 8 Description of conditional man-hour.

5.4 フィージビリティ調査の表示

製品部門によっては、エミュレータとして本来想定されているテスト環境・開発環境としての利用ではなく、既存ソフトウェアを新ボードにのせて製品として出荷に対する要望もある。製品の場合、厳しい性能に対する要求があり、まずはフィージビリティ調査から必要である。このように、実現可否を調査してからでない工数が算出できないもの

を項目として明確化する．なお，工数の ∞ は実現不可能な場合もあることを示す．

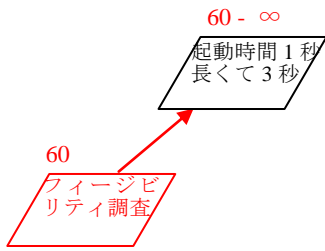


図 9 フィージビリティ調査の表示
 Figure 9 Description of feasibility study.

5.5 関連項目の地図表示

要求分析図の中で離れていても影響しあう要求仕様について，相互の位置を関連づけて表示する．要求分析モデルを列方向に分割し，A, B, …，行方向に分割し，1, 2, …，というようにラベル付する．関連する対象がそれぞれ A2, B2 にあるとき，次図の例のように記述する．

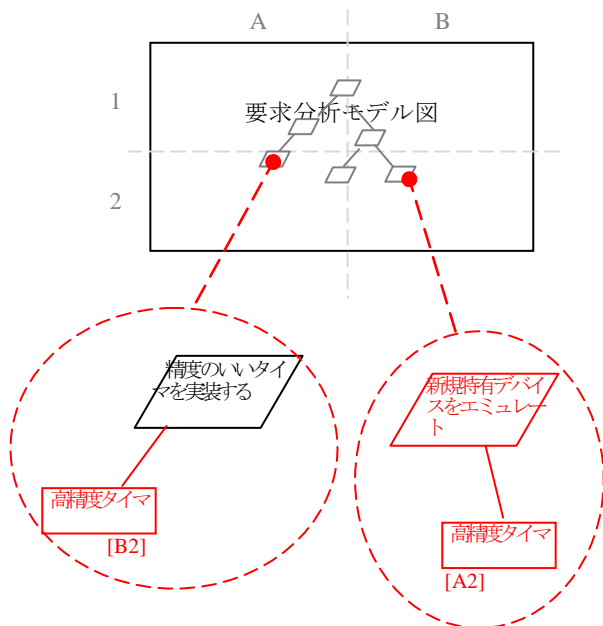


図 10 関連項目の地図スタイル表示
 Figure 10 Map-style description of related items.

5.6 オブジェクトの省略

例えば，CPU のひとつである PowerPC には多くのモデルがあり，オープンソース QEMU 上でも 600~700 程度の定義がされている．これらを分析図中にすべて描画することは効率的ではない．そこで，次の形式でオブジェクトの省略を示すことにする．なお，オブジェクトの個数を [] の中に表示する．個数に幅がある場合は最小値と最大値を表示する．また，個数の多さを重ね具合で表現する．

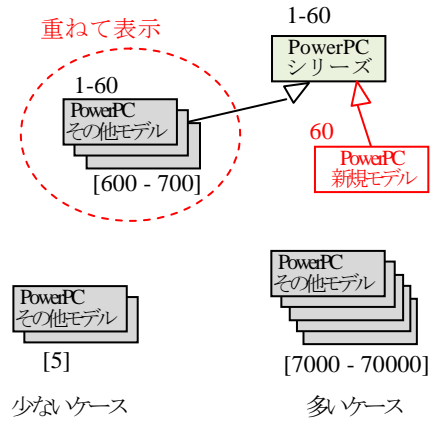


図 11 オブジェクトの省略
 Figure 11 Abbreviation of objects.

6. エミュレータへの適用

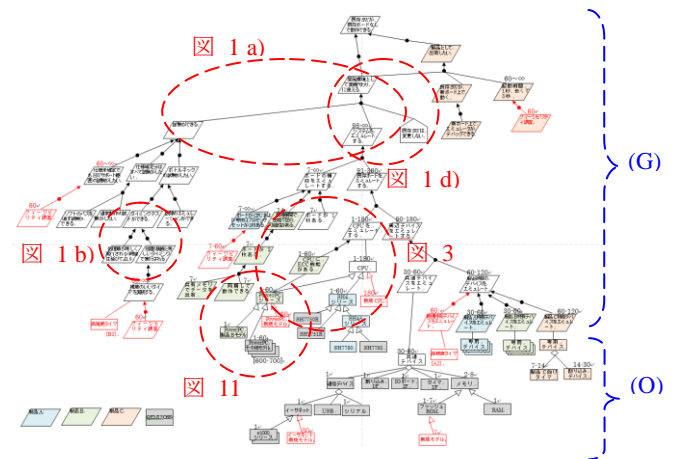


図 12 要求分析モデル全体
 Figure 12 Requirement analysis models.

図 12 は本研究提案手法を既存エミュレータ開発の要求仕様やノウハウを統合した要求分析モデルである．図の上側 (G) はゴールモデルから構成されており，下側 (O) はオブジェクトモデルから構成されている．白いゴールやオブジェクトは共通の要求仕様に関するものであり，色のついたゴールやオブジェクトは製品特有の仕様に関するものである．また，オブジェクト図で灰色の部分にはオープンソース QEMU でサポートされている可能性のあるものである．なお，QEMU では定義だけで実際にはサポートされていないものもあり，詳細はモデルごとに確認する必要がある．

次のエミュレータ開発の要件について，本研究提案手法の考察を行う．

- 1) 工数とエミュレータでできることがわかりやすいこと
 既存の開発案件の要求仕様とともに工数を記載した．ど

の要求がだいたいどのくらいの期間で実現可能かわかるようにした。また、利用目的によってどのように工数が変わるか確認できるようにした。例えば、シンプルな構成のエミュレータでまずは簡単なバグの発見を行いたいという目的の場合、

- CPU：既存のものを流用…1人日
- 周辺デバイス：最低限…90人日
- ボードの構成：標準的な1枚構成…7人日

概算工数は $1+90+7=98$ 人日となる。

しかし、CPUが既存のものではなく、新規に設計しなければならず、かつ既存デバイスエミュレータが備えている以上のタイミング精度を必要とする試験を行いたい場合、

- CPU：新規に設計…180人日
- 周辺デバイス：最低限の構成90+高精度タイミング試験のためのフィージビリティ調査 $30=120$ 人日

ボードの構成：標準的な1枚構成…7人日

従って概算工数は少なくとも $180+120+7=307$ 人日は必須となることがわかる。

2) CPU, デバイス, ボードの構成やデバイス動作仕様などヒアリングすべき項目がもれなく確認できること

オブジェクトとしてCPUや各デバイスを記入し、新規アイテムの場合は赤い枠で目立たせることにより、ヒアリングすべき項目がわかりやすいように表示した。

3) フィージビリティ調査の必要な要求を明確にすること

フィージビリティ調査は赤い枠で囲みどの要求に対するものかわかりやすく表示した。また、調査の結果実現不可の場合もあり得ることを工数の最大値を無限大 (∞) として表現した。

以上のように、エミュレータの工数、機能やフィージビリティ調査が必要なものを可視化して明確化することにより、要求仕様が収束できるようにした。

7. 従来研究

QEMUをベースとしたエミュレータ開発については、小泉ら[5]や大村ら[7]などの取り組みが行われている。しかし、開発の効率化については考慮されていない。筆者ら[1]はプロダクトラインの要求仕様を統合する要求分析モデルについて、並行して開発中の複数の製品開発案件を統合して管理する手法を提案してきた。本研究は既存の複数の製品開発案件をベースとし、新規開発案件に活用するために拡張したものである。複数のゴールモデルを統合する取り組みについて、宇野ら[8]の研究がある。しかし、目的はフィーチャモデルへ変換するためであり、統合したゴールモデルについても具体的にどの製品がどのゴールに対応しているかなどは触れていない。Heidenreichら[3]はフィーチャモデルから実モデルへのマッピングにおいて情報可視化を考慮しているが、マッピング元の使われないモデルをグレイアウトするというものであり、複数製品の仕様を同時に

表示するためのものではない。筆者らの提案を除き、いずれの既存研究もKAOS手法を用いて複数製品の要求仕様を統合して管理しようとしていない。また、実際の製品開発では多機能な仕様であるため大きく複雑な要求分析モデルとなることが予測されるが、視認性の改善についても十分研究されていない。

8. おわりに

本論文では、製品開発初期から導入することで開発効率を向上させるための要求分析モデリング手法を提案した。KAOS要求分析モデルをベースに、情報可視化手法を用いて複数の既存エミュレータ開発案件の要求仕様やノウハウを統合し、新規開発案件に活用できるよう拡張した。今後の課題としては、複雑な要求間の関係をわかりやすく表現する手法や、具体的な評価方法の検討があげられる。

謝辞 (株)東芝 ソフトウェア技術センターの皆様にご協力いただき、エミュレータ開発の状況やKAOSの拡張について議論頂いた。深く感謝いたします。

参考文献

- 1) ジリエ陽子, 本田耕三, 中川博之, 田原康之, 大須賀昭彦: プロダクトラインの要求仕様を統合する要求分析モデルの提案, 情報処理学会研究報告, Vol.2012-SE-177, No.1, pp. 1-8 (2012).
- 2) 原嶋秀次, 蔭山佳輝, 河込和宏: 仮想化技術による実機レステスト環境の構築, 東芝レビュー-Vol.67, No.8, pp. 31-34 (2012)
- 3) Heidenreich, F., Kopcsek, J., and Wende, C.: FeatureMapper: mapping features to models, ICSE Companion, pp943-944 (2008).
- 4) 河井淳, 西山由高: 組み込みシステム用ソフトウェア開発環境, 情報処理, Vol.37, No. 9, pp. 872-879, (1996).
- 5) 小泉仁志, 坂西隆之, 塙敏博, 佐藤三久, 三浦信一, 石井忠俊, 高見澤秀久: 仮想マシンとSpecC デバイスモデルを統合したデバイス故障エミュレータの実現, 情報処理学会研究報告, Vol.2010-OS-115, No.19, pp. 1-8 (2010).
- 6) Lamsweerde, van A. Requirements Engineering From System Goals to UML Models to Software Specifications. Wiley, (2009).
- 7) 大村圭, 田村芳明, 湯口徹, 盛合敏: I/O エミュレーションのロギングリプレイによる仮想マシン同期機構の高速化, 情報処理学会研究報告, Vol.2011-OS-117, No.16, pp. 1-8 (2011).
- 8) 宇野耕平, 林晋平, 佐伯元司: ゴールグラフからのフィーチャモデル導出, 情報処理学会研究報告, Vol.2009, No.31, pp. 1-8 (2009).
- 9) QEMU
http://wiki.qemu.org/Main_Page