

ET ロボコンへの PFD と USDM の適用

八木将計[†] 深谷直彦[†] 大原貴都[†] 小川秀人[†]

[†] 株式会社 日立製作所 横浜研究所

〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地

E-mail: [†] {masakazu.yagi.zd, naohiko.fukaya.ha, takatoshi.ohara.me, hideto.ogawa.cp}@hitachi.com

概要 近年、組込みソフトウェア開発分野における若年層・初級エンジニアの育成が重要となってきた。この状況に対して、「ET ロボコン」はコンテストという形式で、ソフトウェアの分析・設計の教育機会を提供することを目的としている。日立製作所 横浜研究所は、ET ロボコン 2011 に参加し、ロボコン向けソフト開発に PFD と USDM を適用した。ET ロボコンに求められる成果物を精査し、PFD を用いることで、ET ロボコンに特化した開発プロセスを設計した。また、ET ロボコンのソフト設計に USDM を用いることの有効性を示した。

キーワード ET ロボコン, PFD, USDM, 短 TAT, Strategy パターン, 追跡可能性

Application of PFD and USDM to Software Development of Embedded Technology Software Design Robot Contest

Masakazu YAGI[†], Naohiko FUKAYA[†], Takatoshi OHARA[†], and Hideto OGAWA[†]

[†] Hitachi, Ltd., Yokohama Research Laboratory

292, Yoshida-cho, Totsuka, Yokohama, Kanagawa, 244-0817 Japan

E-mail: [†] {masakazu.yagi.zd, naohiko.fukaya.ha, takatoshi.ohara.me, hideto.ogawa.cp}@hitachi.com

Abstract Embedded Technology software design robot contest (ET robocon) offers an opportunity to study the embedded software development for junior engineers. We participated in ET robocon 2011 and applied PFD (Process Flow Diagram) and USDM (Universal Specification Describing Manner) to the software development of ET robocon. In ET robocon, participating teams are ranked according to the running speed of the robot and the abstract document of its software design. We designed a specialized software development process in ET robocon using PFD. Moreover, we clarified an efficacy of USDM for ET robocon.

Keyword ET robocon, PFD, USDM, TAT reduction, Strategy pattern, traceability

1. はじめに

近年、組込みソフトウェアの重要性が増加してきており [1], それに伴い、組込みソフトウェア開発分野における若年層および初級エンジニアの育成が重要な課題の一つになっている。

この状況に対して、ソフトウェアの分析・設計モデリングの教育機会提供を目的とした、産官学連携によるオープン参加型の ET ソフトウェアデザインロボットコンテスト(以下、ET ロボコン)が開催されている [2].

ET ロボコンは、決められた自律型ライントレースロボット (以下、走行体) で指定コースを自

律走行する競技であり、同一のハードウェアに UML®などで分析・設計したソフトウェアを搭載し競うコンテストである。コンテストは、走行体の走行競技速度を競う「競技部門」と、UML®などで記述されたソフトウェア設計モデルの内容(以下、モデル図)の審査である「モデル部門」からなり、それら二つの部門の総合で全体の順位が決定する。

日立グループは、この ET ロボコンに積極的に参加しており、2008 年度 4 事業所 6 チームから、2011 年度 10 事業所 20 チームへと、参加を拡大している。そのなか、日立製作所 横浜研究所は、

組込みソフトウェア生産技術の研究部署の若手研究者(主に入社1年目から5年目社員)を中心としたチームでETロボコン2011に参戦した。

我々の部署は研究部門であるため、ETロボコンの参戦にあたって、教育目的だけではなく、様々なソフト工学の技術を応用することを目的の一つにおいた。用いた技術・ツールは、UML®によるオブジェクト指向設計技術、USDM(Universal Specification Describing Manner)[3]、マインドマップ®[4]、QFD(Quality Function Deployment:品質機能展開)[5]、LTSA(Labelled Transition System Analyser:モデル検査技術)[6]、タグチメソッド(実験計画法)[7]、PFD(Process Flow Diagram)[8]、CCPM(Critical Chain Project Management:プロジェクトマネジメント手法)[9]、TOC-TP(Theory of Constraints - Thinking Process:問題解決手法)[10]である。本報告では、これらETロボコンの開発で用いた技術の中のPFDとUSDMの事例について述べる。

我々のチームは、ほぼETロボコン初参加である上、メンバー全員が研究者であるため実際の開発経験が十分ではなく、どのような開発プロセスなるか想定できないという課題があった。本報告では、その課題に対し、ETロボコンに求められる成果物を精査してPFDを用いることでETロボコンに特化した開発プロセスを設計した。結果、短TAT(Turn Around Time)開発を実現するアーキテクチャを実現し、その効果を実証した。

また、ETロボコンのモデル部門では、「トレーサビリティ(追跡可能性)」という評価項目があり、過去のモデル部門評価の傾向から、その項目が不十分になりやすいという課題があった。本報告では、その課題の解決のため、ソフト設計にUSDMを用い、その有効性を確認した。

2. ETロボコンの特徴と課題

2.1. ETロボコンの特徴

ETロボコンは、走行を競う「競技部門」とソフト設計を競う「モデル部門」からなる。

「競技部門」は、図1に示す走行体(スペックは表1)で、図2に示す競技フィールドの走行速度を競う。走行フィールドは、アウトコースとインコースの2コースあり、2チームが交互に両方を走行する。また、競技コース上には、難所と呼ばれる障害物があり、それら難所をクリアすることで、ボーナスタイムを獲得できる。競技は、イ

ンコース・アウトコースの合計走行タイムから獲得したボーナスタイムを差し引いた総合タイムで競う[11]。

「モデル部門」は、走行体に搭載するソフトウェアの設計であるモデル図(図3)の良し悪しを競う。モデル図は、「機能」「構造」「振舞い」等といった基準で審査される[12]。モデル審査基準では、機能は「UMLではユースケース図などが該当します」、構造は「UMLではクラス図やオブジェクト図などが該当します」、振舞いは「UMLでは状態マシン図、アクティビティ図などが該当します」とあり[12]、多くの場合、モデル図にはUML®が用いられる。

審査はモデル審査員によって行われるが、審査には、それほど多くリソースを確保できない事情から、モデル図はA3用紙5ページ以内と規定されている。この紙面の制限により、モデル図には全ての詳細な設計情報は記載できないため、モデル審査員に分かりやすいよう、設計情報からチームが主張したい点について、まとめることが重要となる。つまり、モデル図は、プレゼン資料などに近い物となる。よって、モデル図は、通常の開発ドキュメントとは異なり、開発自体に直接寄与するものではなく、ETロボコンに特有なドキュメントである。

表1 走行体(LEGO Mindstorms NXT®)のスペック

CPU	ARM7@ 48MHz
OS	TOPPERS / OSEK
駆動系	車輪モータ(左/右), 尻尾駆動モータ
各種センサ	超音波, ジャイロ, 光学, タッチ
その他	Bluetooth®, USB



図1 走行体概観図

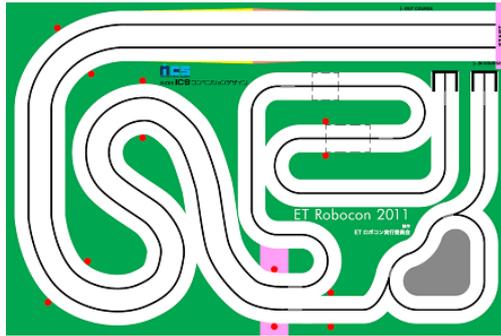


図 2 ET ロボコン 2011 競技フィールド (5460mm×3640mm)

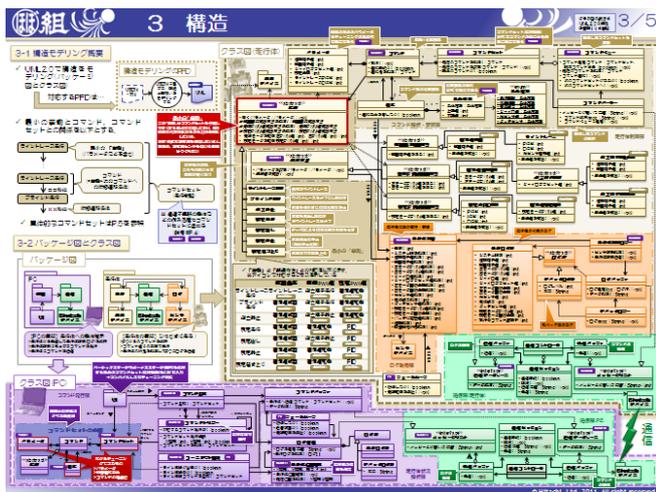


図 3 モデル図

2.2. ET ロボコンの課題

我々のチームが ET ロボコンに参加する上での課題は以下であった。

【課題 1】：不明確な開発プロセス

ET ロボコンはコンテストという特徴から開発プロセスも一般的なソフト開発とは異なる可能性が考えられたが、チームメンバーのほとんどが初参加である上、メンバー全員が研究者であるため実際のソフト開発経験が十分とは言えず、どのような開発プロセスになるか想定できなかった。

また、日立グループ内の過去の参加者によると、プロジェクトのコントロールが非常に大変になりやすく、徹夜や休日出勤で対応することが多いということであった。そのような状況を防ぐ意味でも開発プロセスを明確にする必要があった。

【課題 2】：モデル図のトレーサビリティ

ET ロボコンのモデル審査基準には、「トレーサビリティ(追跡可能性)」という項目がある。ET ロボコンにおけるトレーサビリティ(追跡性)は、通常の開発におけるトレーサビリティ(追跡可能性)

とは異なり、モデル図上で各モデルの対応関係が読み取れるように配慮することを示す。

UML®は、クラス図などの「構造面」と状態マシン図など「振舞い面」の対応関係は図示しやすいが、「機能面」であるユースケース図は他の図との対応を紙面では示しにくい。そのため、機能と構造のトレーサビリティは UML®では示しにくく、過去のモデル審査結果を調査しても、機能と構造に関するトレーサビリティ(追跡可能性)に関する指摘・減点が多い。

我々のチームでも、モデル図には、主に UML®を用いることとしたため、機能と構造のトレーサビリティが問題になると想定された。

3. PFD による ET ロボコンのソフト開発プロセスの分析と設計

2.2 節に示した、ET ロボコンの【課題 1】を解決するために、PFD を適用した。まず、ET ロボコンの一般的なソフト開発プロセスを PFD により分析し、ET ロボコンプロジェクトのコントロールが困難になる要因を明確にした。その課題に対応するため、PFD で ET ロボコンに特化した開発プロセスを設計し、それを實現するアーキテクチャを採用した。

3.1. ET ロボコンの一般的なソフト開発プロセスの分析

一般的なロボコンにおけるソフト開発の PFD を図 4 に示す。図 4 より、ロボコンでは、ロボットのハード仕様と競技規約(競技フィールドや難所の仕様)からどのように走行・難所クリアをするかという「走行戦略」を立てる(P1)。その後、その走行戦略からソフト設計・実装を行う(P2,P3)。結果、作成したソースコードを用いて実際にロボットをレプリカコースで走行させて、走行性能を検証する(P4)。検証結果を受けて、より良い走行を實現するため、「走行戦略」「設計モデル」「ソースコード」にフィードバックを行う(P5)。つまり、速く安定した走行が實現できるまで、反復開発を繰り返すこととなる。

ロボコンの競技部門は、他チームとの競争であり、走行性能に明確な目標がない。そのため、チームが納得するまで、永遠に反復を繰り返し、走行戦略、設計モデル、ソースコードに手を入れ続けることになる。このことが、プロジェクトをコントロールしづらいものにし、徹夜や休日出勤で対応しなければならない要因であると考えられる。

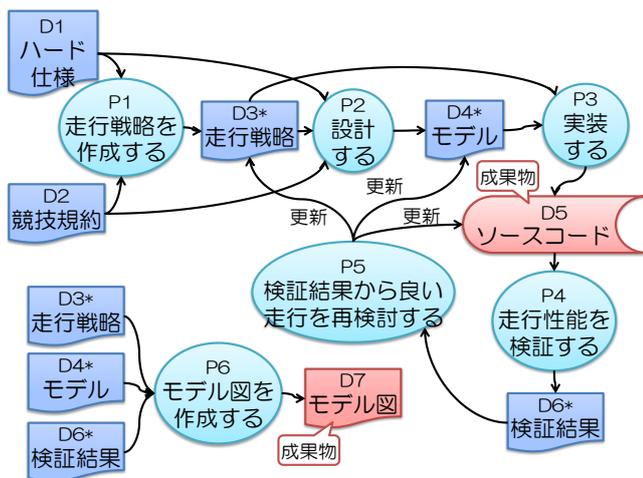


図 4 一般的な ET ロボコンの開発プロセス

3.2. ET ロボコンに特化したソフト開発プロセスの設計

前節に示す ET ロボコンにおける開発プロセスの問題を解決するため、我々のチームでは、ET ロボコンに求められる「成果物」を精査することで、図 5 に示す PFD の開発プロセスを設計した。図 4 に示す一般的な ET ロボコン開発は「ソースコード」を最終成果物としているのに対し、我々が採用した開発プロセス(図 5)は、「走行性能」そのものを最終成果物として扱っている。つまり、「ソースコード」ではなく、走行性能を「走行戦略」ととらえ、その生成(最適化)コストが最小となるように開発プロセス設計した結果が図 5 の PFD である。

図 5 の開発プロセスでは、上流での走行戦略の立案を行わず、「チューニング」というプロセスで、走行戦略を Try&Error で決定していく(P4,P5)。よって、チューニング工程に入ったあとは、設計モデルもソースコードも変更しない。これにより、反復部分が最小限に収められたため、短 TAT 開発を実現し、上流工程(設計・実装工程)への大きな手戻りは発生せず、プロジェクトがコントロールしやすくなるものと考えた。

3.3. プロセス設計に基づき採用したアーキテクチャ

前節の開発プロセスを実現するため、走行体の Bluetooth®通信機能を用いて、PC から走行戦略を送ることで、走行を切り替えるアーキテクチャを設計した。走行体のソフトには、Strategy パターン[13]を採用し、ロボットの最小挙動を用意した。PC で、それら挙動を状態遷移でつないで「走

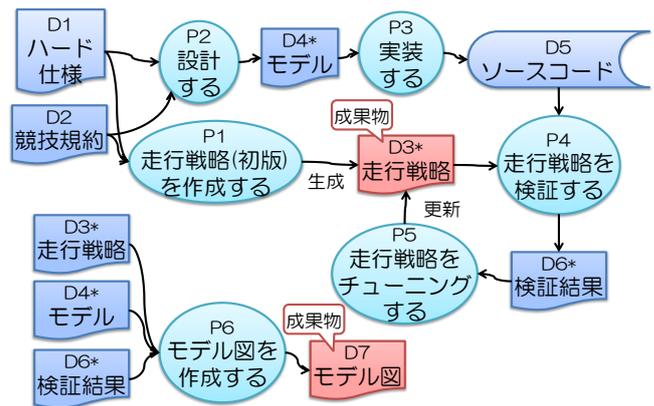


図 5 採用した開発プロセス

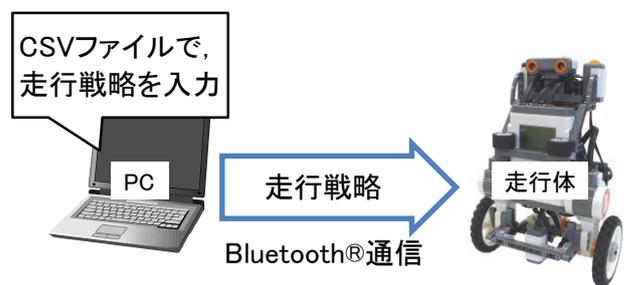


図 6 採用したアーキテクチャ

行戦略」を作成する(CSV ファイルで記述)。その走行戦略を PC から Bluetooth®通信で走行体へ送信して、走行体は送信された走行戦略とおりに走行する(図 6)。このアーキテクチャを用いることで、ソースコードに手を入れることなくチューニングを実現できると考えた。

4. USDM によるトレーサビリティの確保

2.2 節に示した、ET ロボコンの【課題 2】を解決するため、USDM で「機能」を表現し、TM(Traceability Matrix)で「構造」に対応する UML®(パッケージ図)との関係を記述することとした。モデル図に記載した USDM+TM を図 7 に示す。ただし、過去のモデル図の「機能」の内容調査より、ロボコンでの「機能」は USDM での「下位要求」に対応すると考えた。また、実際に作成した USDM 全てを記載する紙面がないため、主要部分のみを記載した。

※ロボコンでの「要求」はUSDMの上位要求に対応し、「機能」は下位要求に対応する

P.3パッケージ図とのトレーサビリティ (○が関連を示す)
 ※実際はクラス・メソッドとの関係記載するが表示しきれないため割愛
 ※クラス図との関係はP.3にアイコン **RUN###** で表示

要求	要求ID	機能ID	内容	PC		走行体							
				戦略	UI	通信	B T デ バ イス	走行	制御 デ バ イス	通信	B T デ バ イス	ログ	セン サ デ バ イス
要求 要求2	RUN		走行体は最小の走行セットを持ち、PCがそれらの組合せを走行戦略として送信。走行体はその走行戦略通りに走行する	○		○	○	○	○	○	○	○	○
	理由		今年から競技中のBluetooth通信が可能になったから、QFDで要求重要度が最も高かったから、										
	説明		競技中に随時、Bluetooth通信でPCから走行体に走行戦略を送信する										
			《走行体》										
	機能	RUN001	走行体は最小の「拳動」を持ち、拳動を指令する「コマンド(PCから送信)」通りに走行する					○	○				
		理由	走行のための拳動をコマンド化することで、チューニングを容易にする										
		説明	最小の「拳動」とは、倒立制御/ライトレース走行 など										
	機能	RUN002	PCからのコマンドをある程度の塊(コマンドセット(走行戦略))で受信し、キューに保持する					○	○	○			
		理由	細かな拳動をその都度PCから指示すると次の拳動に間に合わない可能性があるから										
		説明	このコマンドセットが「走行戦略」になる										
	機能	RUN003	走行中、随時、走行体情報(ログ)をPCに送信する 要求10							○	○	○	○
		理由	PCでコマンドセット(走行戦略)を組み立てるために走行体情報が必要だから、PCからコマンドセットを送信するタイミングを算出するためにも必要だから、										
		説明	走行体情報(ログ)は、システム時刻/速度指示値/各種センサー値(光、ジャイロ)など										
			《PC》										
	機能	RUN011	走行体からの走行体情報からマップを用いて走行体位置を推定する 要求14	○									
		理由	コマンドセットの送信タイミングを把握するため、コースアウトを検知するため、										
		説明	走行体は自己位置推定の機能を持たない										
	機能	RUN012	走行体の位置推定に基づいて、コマンドセット(走行戦略)を作成する	○									
		理由	コマンドセットを走行体の状況に応じて変更できるようにしたいから										
		説明	走行体位置以外でコマンドセットを変更する可能性が低いため、位置のみを見る										
	機能	RUN013	走行体が規定の位置に着たとき、対応するコマンドセット(走行戦略)を走行体に送信する	○		○	○						
		理由	走行体は自己位置推定を行わないため										
		説明	各難所のコマンドセット(走行戦略)を難所に到達する前にあらかじめ送信しておく										
	機能	RUN014	走行体位置推定からコースアウトを検知したら、緊急ライン復帰コマンドを送信する			○	○						
		理由	走行体は自己位置推定を行わないので、コースアウトを検知できないから										
		説明	緊急コマンドは、現状のコマンドセットを強制的に入れ替える→P.5										
要求 要求6-8	TN		制御パラメータチューニングと走行戦略の組み替えをダイナミックに実施する	○	○								
	理由		タグチメソッドを適用しやすくしたい										
	機能	TN001	PCから容易に「拳動」のパラメータを変更できる 要求6	○	○								
	機能	TN002	PCから容易に走行戦略(コマンドセット)を変更できる 要求7	○	○								
要求 要求15	BSC		ベーシック・ステージをコマンドセットを要所要所で切り替えて高速に走行する	○	○								
	理由		優勝するために高速に走行する必要があるから										
要求 要求17	BNS		ボーナス・ステージですべての難所を攻略する	○	○								
	理由		優勝するには、多くのボーナスポイントを獲得する必要があるから										
要求 要求19-22	FS		フェールセーフ機能をPCと走行体のどちらでも発動できる	○				○	○				
	理由		ダブルチェックをした方が信頼性が高いから、競技規約に推奨されているから										
要求 要求22-24	NA		ネガティブアクターの影響			○	○	○	○	○	○		
	理由		センサーや通信への悪影響を軽減しなければ、安定した走行ができないから										
			《センサーへの影響》										
			《通信部》										
	機能	NA001	通信が切断された場合、再接続する 要求23			○	○			○	○		
	機能	NA002	パケットがロスしたら、再送する(PC→走行体、走行体→PCとも) 要求24			○	○			○	○		

図 7 モデル図に記載した USDM

5. PFD と USDM の適用効果の分析

5.1. ロボコンの戦績とプロジェクトのデータ

ロボコンの戦績を表 2 に示す。また、今回のロボコン開発に関わったメンバーは 11 名であり、全体の工数は表 3 に示すとおりとなった。開発には C++ で使用し、開発規模は約 9KLOC であった。

表 2 ET ロボコン 2011 参戦成績

	競技部門	モデル部門	総合
地区大会	13 位(44.7 秒)	2 位(ランク A)	6 位
CS 大会	6 位(-24.8 秒)	37 位(ランク B)	27 位

※CS 大会: チャンピオンシップ大会(全国大会)

表 3 ET ロボコン開発各フェーズの工数(人・月)

	地区大会	CS 大会	合計
調査・要件定義	1.1	0.1	1.2
設計	5.0	0.1	5.1
実装	4.7	0.6	5.3
モデル図作成	1.5	0.1	1.6
チューニング	0.9	0.9	1.8
合計	13.2	1.8	15.0

5.2. PFD の適用効果

今回、我々のチームにとって、ロボコン開発は未知のプロジェクトではあったものの、開発早期の段階で PFD を用いて開発プロセスを明確にできたため、後工程における想定外の作業は発生しなかった。

表 3 に示すとおり、設計・実装工程に多くの工数がかかっているが、チューニング工程に入ってから、設計・実装には手戻りをしておらず、チューニングの工数も 1.8 人・月と少ない。よって、PFD による開発プロセス設計の狙い通り、チューニングで走行戦略を追い込む短 TAT 開発が実現できたと考えている。また、表 2 より、短 TAT 開発を実現できたため、地区大会から CS 大会(全国大会)で、大幅に走行性能が向上(69.5 秒短縮)できたと考えられる。

5.3. USDM の適用効果

ET ロボコンにおけるモデル図の「トレーサビリティ(追跡可能性)の評価項目」は減点法で評価されるが、USDM と TM を用いることで、機能・構造のトレーサビリティ(追跡可能性)については、減点はなかった。ただし、モデル審査は、詳細な所までフィードバックなされないため、USDM の有効性について、それ以上の評価はできていない。

また、今回の開発プロセスが実現できたのは、

3.3 節に示したアーキテクチャを採用したためである。このアーキテクチャの検討は USDM で行なったが、設計・実装以降に USDM に対する手戻りは発生しなかった。

6. 今後について

今回のロボコン開発では、プロジェクトマネジメント手法として、CCPM を用いた。CCPM は、個々のタスクに含まれる余力(バッファ)をプロジェクト全体に集めて管理し、そのバッファを各タスクで融通し合うことでプロジェクトを回す手法である[9]。

この CCPM は、通常 PERT 図などを用いて、クリティカルパス(クリティカルチェーン)を構成していくが、未知のプロジェクトで成果物・タスクが明確でない場合、その実施が困難であった。しかし、PFD により成果物・タスクがはっきりしたことで、CCPM がうまく機能したと考えている。また、CCPM の線表の作成方法は、最終のタスクからクリティカルチェーンを作成する逆線表であるため、最終成果物からプロセスを設計する PFD との相性が良いと感じた。よって、CCPM を実行する上でも、PFD を用いることは非常に有効であると考えられる。

また、CCPM は、プロジェクトバッファを監視して、早期にリスクを抽出して対処するという特徴があり、通常は、発生リスクに対して、人員の追加、スコープの絞り込み、開発期間の延長等で対処する。今回の開発においても発生リスクに対して人員の配置を変更することで対処した。しかし、PFD を用いている場合、PFD でプロセスを再設計することでリスクに対処可能であると考えている。今後、この点について、研究していきたい。

7. まとめ

本論文では、ET ロボコンの開発に対して、PFD と USDM の適用を提案した。一般的な ET ロボコンにおけるソフト開発を PFD を用いて分析した結果、多くの反復がプロジェクトをコントロールしづらくしていることがわかった。本論文では、この課題に対し、ET ロボコンに求められる成果物を精査して PFD を用いることで ET ロボコンに適した開発プロセスを設計した。また、ET ロボコン特有の成果物であるモデル図に求められるトレーサビリティ(追跡可能性)について USDM を用いることが有効であることを示した。

文 献

- [1] 2010 年度版組込みソフトウェア産業実態調査, 経産省,
http://www.meti.go.jp/policy/mono_info_service/joho/downloadfiles/2010software_research/10project_houkokusyo.pdf.
- [2] ET ロボコン 2011 ホームページ,
<http://www.etrobo.jp/2011/index.php>
- [3] 清水吉男, [改訂第 2 版] [入門 + 実践] 要求を仕様化する技術・表現する技術, 技術評論社, (2010).
- [4] Tony Buzan, Barry Buzan, ザ・マインドマップ, ダイヤモンド社, 2005.
- [5] 赤尾洋二, 品質展開入門, 日科技連出版社, 1990.
- [6] LTSA - Labelled Transition System Analyser,
<http://www.doc.ic.ac.uk/ltsa/>
- [7] 矢野宏, おはなし品質工学, 日本規格協会, 2001.
- [8] PFD (Process Flow Diagram) の書き方,
http://homepage3.nifty.com/koha_hp/process/PFDform3.pdf
- [9] 西原隆, 栗山潤, TOC/CCPM 標準ハンドブックークリティカルチェーン・プロジェクトマネジメント入門, 秀和システム, 2010.
- [10] Eliyahu Goldratt, ザ・ゴール 2ー思考プロセス, ダイヤモンド社, 2002.
- [11] ET ロボコン 2011 「競技規約」,
<http://www.etrobo.jp/2011/gaiyou/kiyaku.php>
- [12] ET ロボコン 2011 「モデル審査基準」
<http://www.etrobo.jp/2011/gaiyou/model.php>
- [13] Erich Gamma, et al., オブジェクト指向における再利用のためのデザインパターン[改訂版], ソフトバンククリエイティブ, 1999.