2 組み込みソフトウェア向け開発支援環境

Embedded Software Development Environments

渡辺　政彦　キャッツ（株）
watanabe@zipc.com

組み込みソフトウェアとは，さまざまな機器に組み込まれてサービスを提供するソフトウェアを指す．本稿では，特に自動車や携帯電話の分野を中心に論じる．ここでは，組み込みソフトウェアに起きている変化について，質，量，具現化という3つの側面から検討する．これらの変化が従来のコードベース開発からモデルベース開発への移行が必要であることを述べる．次に組み込みシステムのモデリング対象と粒度を明確にし，それぞれにおける組み込みソフトウェア向けモデルベース開発手法およびその開発支援環境を解説する．

組み込みソフトウェアとは

組み込みソフトウェアの特長の1つは，それぞれの用途に特化したハードウェア上で動作することである．組み込みソフトウェアにとってのハードウェアプラットフォームは，自動車，携帯電話，複写機，ディジタルカメラ，家電，医療機器，そして衛星ロケットなどさまざまだ．一方，組み込みではないソフトウェアにとってのハードウェアプラットフォームは，パソコン，ワークステーションである．汎用的なコンピュータをソフトウェアが，経営戦略支援，顧客管理，在庫管理，給与計算，ワープロ，表計算，メイラ，ブラウザ，ゲームなどさまざまな用途で使用できるようにする．このため，組み込みソフトウェアは組み込みではないソフトウェアに比べて目立たぬ存在であった．しかし，ここ数年の間に「組み込みソフトウェア」が注目を浴びている．なぜか．携帯電話やカーナビの普及率とともに，それらに組み込まれたソフトウェアによって数多くのサービスが実現されるようになったからである．

たとえば携帯電話・自動車・半導体

たとえば数年前は会話するだけだった携帯電話は，最近ではメールやインターネットに接続され，さらにカメラが搭載された．これにより，身の回りの電子機器によって，さまざまなサービスを受けられるようになった．このため携帯電話のプログラム規模は1996年の時点で約700KLOC（Line Of Code）のサイズが，1999年の時点で約1.5MLOCのサイズに増大した[1]．3年間で倍の規模となり，その規模は100万行を超えている．携帯電話は電話機能からデータ通信機能，ディジタルカメラ機能へと提供するサービスの質が変化し，それが同時にコード量の増大を招いている．

自動車では1980年代はEFI（Electronic Fuel Injection）やABS（Anti-lock Brake System）などの車としての「走る」「止まる」といった基本性能を電子化，つまりマイコンによる制御を行っていた．2000年代に入るとカーナビや周辺監視といったITS（Intelligent Transport Systems）により，「走る」「止まる」「曲がる」の基本性能のほかに「つながる」といった利便性が新たに加わる．これにより1994年から1998年のわずか4年間で自動車に組み込まれるプログラム開発総量が約8倍となっている[2]．

組み込みソフトウェアが動作するデバイスは，MPU（Micro Processing Unit）である．半導体の高集積化により安価で高機能なMPUが提供されるようになった．このことが，携帯電話や自動車に組み込まれるソフトウェアの質，量の変化と深く関係している．携帯電話，自動車は1980年代では8bitのMPU，1990年代には16bit，そして2000年代では32bitのMPUを採用している．

ITRS（International Technology Roadmap for Semiconductors）によれば，1990年から2000年の10年間で集積度は100倍となり，58%/年で集積度が増加している[3]．100万ゲートを超えるチップの出現は，従来ソフトウェアで実現していたアルゴリズムやプロトコルといったミドルウェア機能を，ハードウェアで実現することを可能とした．さらにFPGA（Field

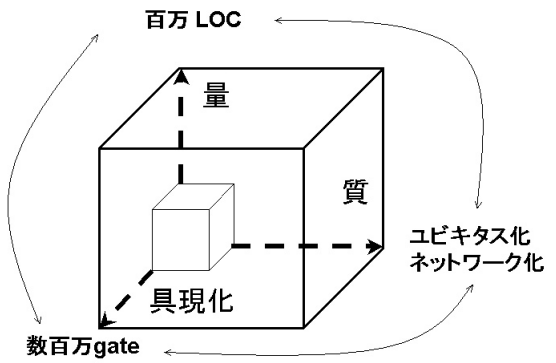


図 -1　組み込みソフトウェアの変化

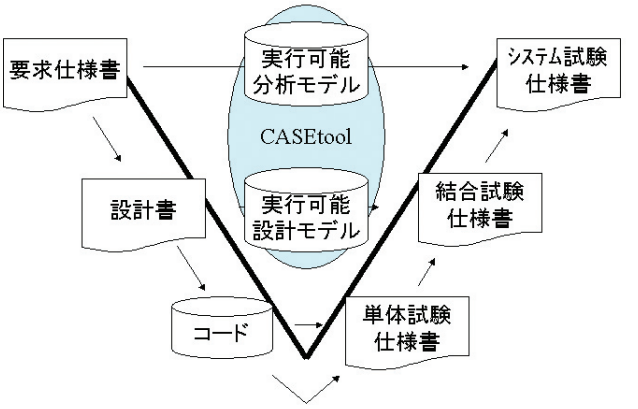


図 -2　組み込みモデルベース開発

Programmable Gate Array）や PLD（Programmable Logic Device）といったハードウェアの高性能とソフトウェアの柔軟性を合わせ持つデバイスの出現で，ハードウェアとソフトウェアの境界が変化している．従来変更しやすいようにとソフトウェアで実現していた機能を，ハードウェアで処理させることができるのである．

組み込みソフトウェアの変化による問題

これまで述べたように，組み込みソフトウェアの変化とは，(1) 新しいサービスに対応するための質の変化，(2) 開発規模が増大する量の変化，(3) 小型化，低消費電力，低コスト，処理性能向上を実現するための具現化の変化，である（図 -1）．これら 3 つの変化が，品質低下，開発スケジュールの遅れといった問題を引き起こしている．携帯電話キャリアの Web ページには，最新機種の不具合が掲載されている．国土交通省自動車交通局のリコールの原因調査・分析検討調査報告書では，自動車に組み込まれる技術の高度化・複雑化が不具合要因の 1 つであると報告している [4]．さらに，「質」，「量」，「具現化」の変化が起こっているにもかかわらず，開発期間は短縮の方向にある．

コードベース開発からモデルベース開発へ

開発対象である組み込みソフトウェアが変化しているのであるから，開発を支えるプロセスはその変化に対応すべきであろう．従来のプログラムコードを中心とした開発には限界がきている．コードベース開発では要求仕様や設計はワープロやドローイングツールを用いて文書化が行われ，レビューによって妥当性検証が実施される．仕様の動作確認を行わないため，上流工程の要求仕様フェーズでは想定していなかった使われ方，つまり想定外事象を，下流工程である実装フェーズで発見する可能性が高い．実装フェーズで要求仕様を変更するため，フィードバック工数が増加し，品質は低下する．納期優先のあまり，大量の試験項目を実施する試験時間を十分にとることができず，多くの不具合がプログラムに潜在し，リコール問題を引き起こすことになる．

コードベース開発に変わる開発プロセスとしてモデルベース開発が注目されている．モデルベース開発では要求仕様書を実行可能分析モデル，設計書を実行可能設計モデルとして，動作できるモデルを作成する（図 -2）．ソフトウェア開発工程の各フェーズのモデルを動作検証することで，フィードバック工数を削減し，品質を高める．設計モデルからコードを自動生成することで生産性を高める．モデルの動作検証，モデルからのコード自動生成を行うには，CASE ツールが必要となる．CASE ツールの導入には，要求と設計をきちんとモデリングするスキルが求められる．

従来の組み込みらしさを持つ MbR モデル

分析，および設計モデルはモデリング対象の種類と粒度ごとに表記法，方法論，それらを支援する開発環境が異なる．組み込みシステムは規格や公式等のルールからモデリングされる MbR（Modeling by Rule）と，分析からモデリングされる MbA（Modeling by Analysis）に分類できる（図 -3）．

MbR は性能を満たすための具体的な方策を主にモデリングする．従来の組み込みソフトウェアの分析・設計とは MbR が大部分を占めている．MbR は今後もなくなることなく進化する分野で，他のエンタープライズ系ソフトウェアと組み込みソフトウェアを区別するのに分かりやすい特徴を有している．MbR は連続系モデルと離散系モデルに分類される．

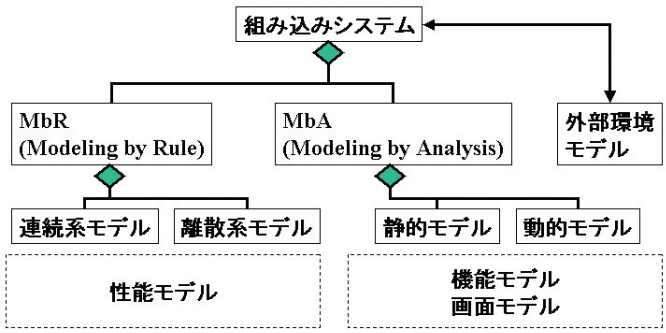


図-3　組み込みシステムモデルの種類

連続系モデルの例では，エンジン制御やトランスミッション制御，または無線電波の物理層制御などがある．連続系モデルは，システムはエンジンの回転数や電波反射といった外部の物理現象のフィードバック制御を行う．フィードバック制御のモデルにはブロック図や状態図が用いられ，外部の物理現象には原理に基づく公式が用いられる．この分野の開発支援環境にTMW社のMATLAB/Simulink（http://www.mathworks.com/）がある．

離散系モデルの例では，通信プロトコルモデルがある．IMT2000（International Mobile Telecommunications）には，W-CDMA（Wideband Code Division Multiple Access），CDMA2000，TD-SCDMA（Time Division Synchronous）の3つの規格がある．通信接続を確実なものとするために，規格は通信プロトコルとして，レイヤごとにモデリングされる．この分野の開発言語としてSDL（Specific and Description Language）があり，開発支援環境としてTelelogic社のTelelogic/Tau（http://www.telelogic.com/）がある．

これからの組み込みらしさを担うMbAモデル

サービスを実現するモデルがMbAである．性能重視であった組み込みシステムに，利便性や快適性といったサービスが求められている．このようなサービスは規格が厳密に定義されているわけではない．分析によるモデリングを通して，要求仕様が明確化される．利便性，快適性といった定性的な評価を最終工程のコードで行うことは，フィードバック工数増大につながるため，実行可能モデルによる上流工程での動作検証が重要である．

たとえば自動車のマドの場合，従来手動で開け閉めを行っていたマドを，スイッチとモータにより自動化したものが，パワーウィンドウである．モータがマドを開閉するモデルは，モータの性能から数学的にモデリングされ，スイッチを押されて何秒でマドが開閉するかをシミュレーションする．このようなモデルは前述したMbRである．「つながる」という新しいサービスを実現するこれからの自動車のマドは利便性，快適性という視点から，モデリングすることになる．携帯電話でマドを閉めることでマド閉め忘れ防止や，自動車が自立的にマドを開閉し，駐車中の車内温度を自動的に調整するといった新しいサービスをモデリングする．

ステークホルダーに対してどのようなサービスを提供するかといったモデルにはUML（Unified Modeling Language）のユースケースが用いられる．UMLを支援するツールは商用からフリーのものまで非常に数多くの製品が存在する．IBM社のRose（http://www-6.ibm.com/jp/software/rational/products/rose/），CanyonBlue社のKonesa（http://www.canyonblue.com/jp/），Foundatao社のPattern Weaver（http://www.foundatao.com/），永和システムマネジメント社のJude（http://www.esm.co.jp/）などがある．

携帯電話などのメニュー画面を，いかに使いやすくするかといったユーザビリティはMbA設計である．情報を扱うシステムは，それを使う人間を中心に設計されるべきとした人間中心設計プロセスとしてISO-13407がある．ISO-13407では人間中心設計のための方法を規定しているのではなく，プロセスマネジメントを対象としている[5]．具体的な画面モデルには画面遷移図が表記法として用いられ，プロトタイピングシミュレーションが行われる．HMI（Human Machine Interface）モデリングの開発支援環境としてeSIM社のRapidPLUS（http://www.e-sim.com/home/index-h.htm），CATS社のDrawrial（http://www.zipc.com/）などがある．

モデルの粒度，種類，抽象度の関係

MbAにはモデリングを行う視点として，構成を示す静的モデルと振舞いを示す動的モデルがある．MbRに

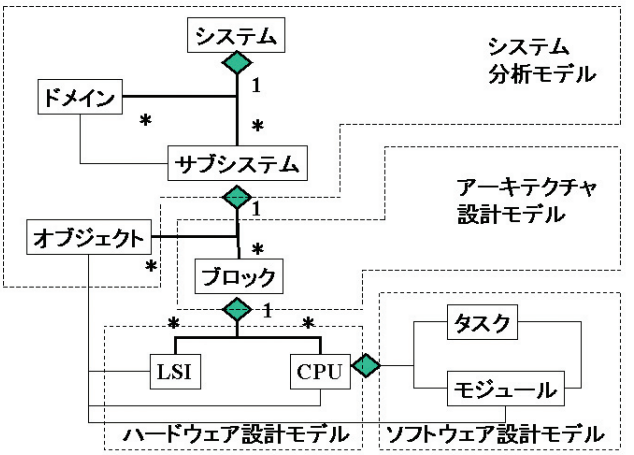


図 -4　組み込みシステムの粒度

も同様に静的モデルと動的モデルはあるが，MbA と MbR との違いは，MbR の方が MbA よりも細かい粒度からモデリングを開始する点にある．MbR は後述するアーキテクチャ設計モデルの粒度からモデリングを開始するのに対し，MbA はシステム分析モデルから開始する．

サービスや性能を実現する最大粒度であるシステムは，サブシステム，ブロックに分割される．ブロックは CPU，ROM，RAM，周辺 LSI といったデバイスに分解される．CPU 部分はソフトウェア設計で，さらにタスク分割もしくはモジュール分割される（**図 -4**）．

組み込み向けオブジェクト指向開発のベストプラクティスである eUML（Embedded UML）では，システムをサブシステムに階層化する前にドメイン分割を行う．ドメインとは問題領域に沿った分割で，システムに必要な専門知識を再利用できるように分類したものである．サブシステムが物理的な構成を示すのに対し，ドメインは論理的な構成を提供する．サービスの種類を整理してドメイン分割を行い，サービスを実行する物理的構成の変更の影響を受けないようにすることができる．ドメインが明確になれば，従来通りにサブシステムに分割する．続いて，オブジェクト構造が分析される．ブロック分割の前に再利用を考えたオブジェクト部品を抽出することで，抽出したオブジェクトが物理的構成の影響を受けないようにする．オブジェクトはハードウェアで実現するか，ソフトウェアで実現するかを，アーキテクチャ設計で選択することができる．

MbR と MbA モデルの併用

アーキテクチャ設計モデルの粒度から始まる MbR モデルと，システム分析モデルの粒度から始まる MbA モデルは併用される．組み込みソフトウェアは，エンタープライズ系ソフトウェアに見られる J2EE や .NET などの共通プラットフォームがない．組み込みソフトウェアにおけるプラットフォームは，MbR モデルで対象システムごとに開発されている．

たとえば自動車の場合では，エンジンや ABS などのパワートレイン系，パワーウィンドウやエアコンなどのボディ系，カーオーディオやカーナビなどの情報系がプラットフォームである．このプラットフォーム上に予防安全サービスとして，「ドライバが居眠りをしている場合，マドを開けるサービス」などを MbA でモデリングを行う．

新しいサービスを発見し，そのサービスの MbA モデルを通して，価値，性能，コストなどを見積もる開発が，今後の組み込みソフトウェア開発の潮流になると予測する．このような開発をサービス駆動開発（SDD：Service Driven Development）と定義する．

既存資産として蓄積された高度な性能，機能を有する MbR モデルをプリミティブ・オブジェクト（PO：Primitive Object）と定義する．PO は従来のスイッチやセンサからの入力だけでなく，サービスへのインタフェースを保有するように拡張を施す必要がある．プリミティブ・オブジェクトの集合がプラットフォームである．

SDD ではアプリケーションであるサービスと，プラットフォームであるプリミティブ・オブジェクトとのインタフェースが重要になる．抽象度の高いインタフェースにより，プリミティブ・オブジェクトとサービスの独立性を高めて，それぞれの進化がお互いに影響を受けないようにすべきである．

また，プリミティブ・オブジェクトが既存機能である制御を行っている状態で，サービス要求の事象があった場合にどうするかといった動的モデリングが必要になる．

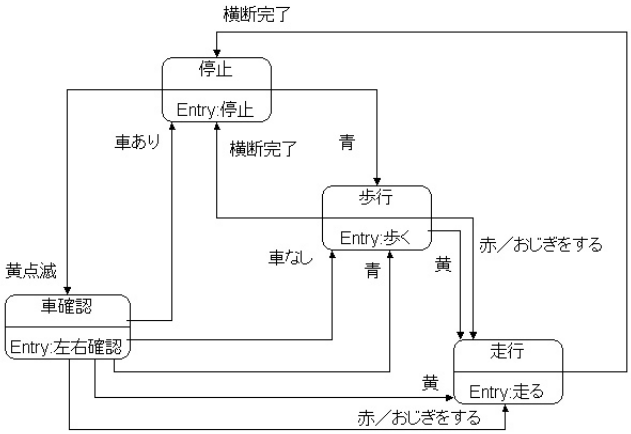


図 -5　状態図例 [7)]

動的モデルの表記法

数多くの UML を支援するツールの中で，リアルタイム制御向けや組み込み向けと特徴づけしている製品の多くは，動的モデルから C 言語といった非オブジェクト指向言語の自動生成や，RTOS へのオブジェクトのマッピング機能を備えたものである．このような開発支援環境に IBM 社の Rose Realtime（http://www-6.ibm.com/jp/software/rational/products/rose/），I-Logix 社 の Rhapsody（http://www.ilogix.com/）などがある．

オブジェクト指向では，クラスを動的モデルの対象とする．組み込みの分野では，システム，サブシステム，ブロック，タスク，モジュールを対象として動的モデリングを行うことが多い．動的モデルの表記法には，時系列的なメッセージの流れを記述するメッセージシーケンス図や，有限化した事象と状態を制御する有限状態機械を記述する状態図または状態表がある．

しばしば事故の原因となる想定外事象は動的モデルの責務である．事故に直結する事象を正確に想定することが，設計者に要求される能力であると科学技術振興事業団の失敗知識データベースでは指摘している [6)]．組み込みシステムの大半を占める処理は例外処理である．状態図に例外仕様ケースを記述すると，図が複雑になり理解しにくいものとなってしまう（**図 -5**）．それに対して状態表では事象と状態のすべての組合せが表のセルとして用意されるので，すべてのケース，正常ケースも例外ケースも気にかけやすい（**図 -6**）．状態遷移の概要を把握するには状態表では難しく，状態図が適している．状態図を用いてモデリングを支援する開発環境としては，I-Logix 社の Statemate（http://www.ilogix.com/）がある．状態表を用いてモデリングを行う開発支援環境として CATS 社の ZIPC（http://www.zipc.com/）がある．

制約をクリアするためのアーキテクチャ設計

MbA によるモデルは，アーキテクチャ設計フェーズにおいて，MbR と同様に組み込みシステムの厳しい制約条件にさらされることになる．制約条件とは面積，性能，消費電力，コストなどである．先に述べた半導体の技術進化により，機能をハードウェアで実現するか，ソフトウェアで実現するかを選択できる．機能を最適にハードウェア，ソフトウェアにマッピングするアーキテクチャ設計，もしくはハードウェア・ソフトウェア協調設計の良し悪しが，低消費電力で処理性能が高く，小さな機器を実現できるか否かにかかってくる．

このようにシステム分析モデルをアーキテクチャ設計モデルに変換するアーキテクトが，組み込みシステムでは非常に重要な役割を担っている．アーキテクチャ設計

	S	停止 Entry:停止する	車確認 Entry:左右確認	歩行 Entry:歩く	走行 Entry:走る
E		0	1	2	3
青	0	=>歩行	=>歩行	／	／
黄	1	／	=>走行	=>走行	／
赤	2	／	=>歩行 おじぎする	=>走行 おじぎする	／
黄点滅	3	=>車確認	／	／	／
車なし	4	／	=>歩行	／	／
車あり	5	×	=>停止	／	／
横断完了	6	×	×	=>停止	=>停止

図 -6　状態表例 [7)]

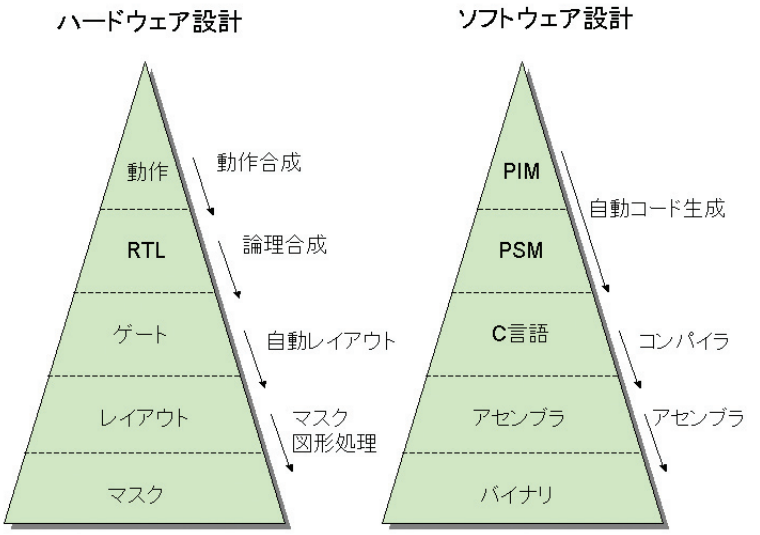


図 -7 ハードウェア設計・ソフトウェア設計支援

時にハードウェアで実現するか，ソフトウェアで実現するかを自由に交換して動作検証することが望ましい．現在，LSI 設計は従来の VHDL，VerilogHDL から，C 言語ベース設計へ移行している．C 言語ベース設計では，C 言語や C++ 言語，もしくは並列処理機能等の拡張を施した SystemC 言語や SpecC 言語などが記述言語として採用されている．C 言語ベース設計となり，CASE ツールを使い，モデルから C 言語ソースコードを自動生成することで，モデルベース開発への移行もしやすい．UML から SystemC のコードを自動生成する開発環境として，CATS 社の XModelink（http://www.zipc.com/）がある．

ハードウェア設計・ソフトウェア設計の自動化を加速するツール化

ハードウェア設計では，レイアウトデータ（シンボル図）からマスク図形データ（実体図）を自動的に生成するマスク図形処理，ゲートから自動的にレイアウトをおこす自動レイアウト，RTL からゲートを生成する論理合成は EDA ツールによって自動化されている（**図 -7**）．今後は C 言語で書かれた動作から RTL を自動生成する動作合成ツールが求められる．現在，NEC エレクトロニクス社の Cyber（http://www.necel.com/index_j.html），フューチャーデザインオートメーション社の Design Prototyper（http://www.future-da.co.jp/index2.html）などがある．

一方，ソフトウェア設計では，アセンブラ言語をバイナリ（機械語）に変換するアセンブラ，C 言語や C++ 言語をアセンブラ言語に翻訳するコンパイラ，静的モデルや動的モデルから C 言語や C++ 言語を自動生成する CASE ツールがある（図 -7）．OMG（Object Management Group）が提唱する MDA（Model Driven Architecture）はモデルを PIM（Platform Independent Model）と PSM（Platform Specific Model）とに分けることで，PIM を変更することなく多くのプラットフォーム上で動作させることを可能としている．MDA を支援する開発環境として Project Technology 社の BridgePoint（http://www.projtech.com/），Kenedy Carter 社の iUML（http://www.kc.com/）がある．

組み込みソフトウェア立国へ

いかにリソースを抑えて機能を実現するかといったコード最適化中心の開発から，どのようなサービスをどのように提供するかといったモデル洗練化の開発へシフトするだろう．コードの最適化はコンパイラと同様に CASE ツールの役割となる．モデルベース開発に必要なモデルリングスキルを身につけた，これからの世界をリードする日本の組み込み技術者を期待したい．

参考文献

1）平山雅之：組込みシステム開発における品質向上の施策，ET2002 講演資料．
2）重松　崇：自動車産業におけるソフトウェア開発戦略，ET2002 講演資料．
3）http://www.cqpub.co.jp/dwm/contents/0028/dwm002800580.pdf
4）http://www.mlit.go.jp/kisha/kisha02/09/090905_4_.html
5）堀部，山本：ユーザビリティ工学と国際規約，情報処理，Vol.44, No.2, pp.128-135（Feb. 2003）．
6）http://shippai.jst.go.jp/
7）渡辺政彦，飯田周作，石田哲史，山本修二，浅利康二：UML 動的モデルによる組み込み開発，オーム社（2003）．

（平成 15 年 11 月 26 日受付）