組込みシステムのシステム要求からの段階的なモデル駆動開発手法

伊藤　邦彦†　　松浦　佐江子†

芝浦工業大学大学院工学研究科電気電子情報工学専攻†

1. はじめに

組込みシステム開発では，センサーやアクチュエータといった機械部品を制御するソフトウェアを開発する．しかし，システム要求に変更がない場合でもハードウェア構成の違いにより制御フローを変更する必要がある．また，システム構成の違いのすべてを実機で確認することはできないため，システム構成を考慮したシミュレーションを行う必要がある．そこで，システム要求をハードウェア構成に非依存に定義し，段階的にハードウェア構成に依存したモデルの構築を行い，自動変換によりxUML[1][2]にすることによってシミュレーションを行う．本稿では，システム要求から段階的モデル構築によるプロダクト開発までの段階的開発手法を提案し，プロダクトがシステム要求を満たしていることを検証する．

2. 開発プロセス

業務系システム開発では，要求を分析する際，利用者や外部システムを含めた業務フローを十分に分析することが重要である．組込みシステムにおいても，組込み機器を含めたシステム全体の作業フローを分析し，その中での組込み機器の要件を検討する必要がある．例えば，迷路探索ロボットの場合では，「自律走行可能なロボットが，複雑に入り組んだ道を辿って，ゴールまで辿り着くこと」がシステム全体の要求である．すなわち，「迷路探索の解法を自律走行可能なロボットが実現する」ことが要求されていると考えることができる．しかし，ここで問題となるのは，迷路探索の解法を実現可能なロボットの要件はセンサーの種類，数，位置等，多様であり，全ての場合を実機によって確認することはできない．そのため，入出力の多様性を考慮し，アルゴリズムと入出力動作の分割を行う．また，ハードウェア要件決定の根拠が明確にならないと，システム全体の要求自体が変化した場合に対応が困難となる．物理環境を媒介として動作するため，実行環境である外部環境へのアクチュエータの影響が必ず正しく行われるとは限らない．そこで，入出力と外部環境の関係を分割し，入出力が正しく行われないことを考慮し，誤差を導入したシミュレーション実行を行う．それぞれの観点から段階的に分析を行う方法を提案する．図1が提案方法の開発プロセスである．

2.1. システムの要求定義

システムの要求定義では解決すべき問題からそのアルゴリズムとデータを特定する．アルゴリズムはアクティビティ図を用い，データはクラス図を用いて記述する．また，アクティビティ図にはアクション実行後にある属性の事後条件を明示的に記述することにより，アクションが満たさなくてはいけない実行動作を特定する．このように作成されたモデルを初期モデルとする．

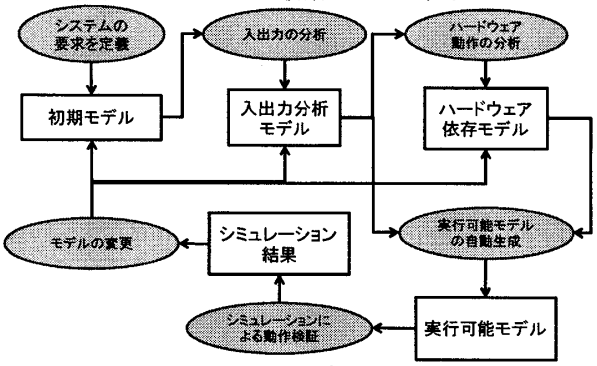

図1　開発プロセス

2.2. 入出力の特定

センサー・アクチュエータの配置位置及び動作を想定し，オブジェクトノードによってデータ操作を限定することで入出力アクションの特定を行う．ここで，入力はどのようなデータを取得し判別するか，また出力はどのデータを変更するかを決定する．初期モデルに入力・出力パーティションを追加し，入出力を特定したアクションのみを用いて初期モデルアクションの条件を満たすようにアクション系列を決定する．また，外部環境パーティションを設け外部環境への影響を記述する．

2.3. ハードウェア・外部環境の特定

ハードウェア・外部環境の大きさやその間の関係を特定する．また，アクチュエータが外部環境に与える変化量を具体的な数値で決定し，その結果センサーで得られる具体的な数値を入力アクションにおいて判定する閾値の決定を行う．ここで，誤差が生じるデータに対応する誤差属性を外部環境クラスに追加する．

2.4. 実行可能モデルの生成

クラス図，アクティビティ図からxUMLに必要なステートマシン及びアクション言語によるステートの振舞いを生成する．アクティビティ図からxUMLのステートマシン図への変換ルールが表1である．

表1　モデル変換表

	アクティビティ図	ステートマシン図
1	対象クラスのパーティション内のアクション	状態
2	対象パーティション内から次の対象パーティションへのエッジ	遷移
5	オブジェクトノード	オブジェクトノード直前のアクションに対応する状態のプロシージャ
6	ガード	ガード直前のアクションに対応する状態のプロシージャ ガード直後のアクションに対応する状態に到達する遷移のイベント

2.5. シミュレーション実行による検証

シミュレーション実行では入出力のアクション系列の検証とハードウェアの精度を考慮するため誤差による影響を検証する必要がある．アクション系列の検証では初期アクティビティ図の開始ノードから終了ノードまで到達可能か検証する．到達不可能であった場合入出力分析モデルの再検討を行う．誤差による検証では，入出力時

Model-Driven Developments Methods from System Requirements of Embedded System

†Kunihiko Ito　†Saeko Matsuura

†Department of Electronic Engineering and Computer Science, Graduate School of Engineering, Shibaura Institute of Technology

に想定されるハードウェア・外部環境による誤差の許容範囲をデータに入れることで検証を行う．この検証によってシステムの要求を満たす誤差の許容範囲をハードウェア的またはソフトウェア的に解決するかを検討する．

3. 適用事例

事例として迷路を探索し目標位置まで移動する自律走行ロボットを作成する．

3.1. 初期モデル

要求を満たすためのアルゴリズムとして右の壁を伝うことによって探索を行う右手法を採用した．右手法による探索アルゴリズムと探索の終了条件をアクティビティ図にしたものが図 2 の制御パーティションである．また，扱うデータは二次元の方位を持つ座標系を想定している．

3.2. 入出力分析モデル

初期モデルを実現する入出力の案として，センサーを前方に一つ，アクチュエータにより座標を変えずにその場での回転および前進ができるものとし，入出力のアクションを表 2と定義した．初期モデルで定義したオブジェクトノードの値を満たすようにアクション系列を作成する．外部環境のデータを扱うクラスを追加する．

表 2　入出力モデルのアクション

パーティション	アクション	動作
入力	現在の向きの壁の有無を調べる	現在の向きの壁の有無を取得する
出力	90度右を向く	現在の向きを右方向変更する
	90度左を向く	現在の向きを左方向変更する
	一区画前に進む	現在の向き応じて座標を変更する

3.3. ハードウェア依存モデル

使用するセンサーを超音波センサーとし，入力として壁との距離を取得する．ハードウェア・外部環境の具体的な数値を決定する．一区画の長さと移動距離を決定し，入力から取得する距離を用いて壁の有無を判断する際の閾値を決定する．シミュレーション実行することで実機なしで閾値の決定ができる．また，迷路クラスに区画座標とは異なる連続的な座標を扱う属性を追加する．図 2 が超音波センサーを用いた時のアクティビティ図である．また，扱うデータは図 3 のクラス図である．図 2，図 3 を xUML へ変換しシミュレーションを行う．

4. 評価と考察

xUML を外部環境として定義した仮想上の迷路でシミュレーション実行し，ロボットの属性である座標が終了座標になるか実験する．さらに，誤差を入れ実験を行い，その結果生じる問題を解決する方法について考察する．

本実験では迷路の一区画を 10cm とし出力の誤差がない時 10cm 移動する．誤差を±0.1cm ずつ加えた時の結果が表 3である．「探索終了」は Robot クラスの座標 x,y が終了座標 x,y を満たした場合「○」とし，「物理到達」は迷路クラスの座標が終了座標に到達したかどうかを表している．探索中に壁に衝突したものは両項目が「×」としている．この結果から正常に動作する誤差範囲は±0.1cm である．誤差がマイナスであれば終了条件を満たすが，実際の終了位置とは異なる場所で止まる．また Robot の座標が終了座標に到達しないため無限ループに陥った．しかし，実行結果をみると迷路を探索し終了位置まで走行した．これは，ロボットの内部の終了条件を満たさなかったが物理的には終了位置まで走行可能であることを示している．実機では，探索終了位置まで走行するため一見正常に動作したかに見える．そのため，アルゴリズムの欠陥か，内部データと外部環境との差異よるものか原因の発見が困難となる．

表 3シミュレーション結果

誤差	-1	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1
探索終了	×	○	×	×	○	○	○	○	○	○
物理到達	×	×	×	○	×	×	×	×	×	○

誤差	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
探索終了	○	×	×	×	×	×	×	×	×	×
物理到達	○	○	○	×	×	○	×	×	×	○

欠陥をなくすために，誤差を 0.1 以内にできない場合制御方法を変更する必要がある．入出力分析モデルの修正により問題を解決する．壁との接触を避けるため，壁との距離を調整する．この変更では「半区画前進する」といった新たなアクションの追加が考えられる．初期モデルの条件を満たしつつ変更を行う．壁の有無だけでなく，壁との距離を判断し，それに応じた出力を行うことで探索アルゴリズムを変更せずに入出力動作のみを変更を行うことができる．また，入力回数，センサー数の追加といった変更も考えられる．

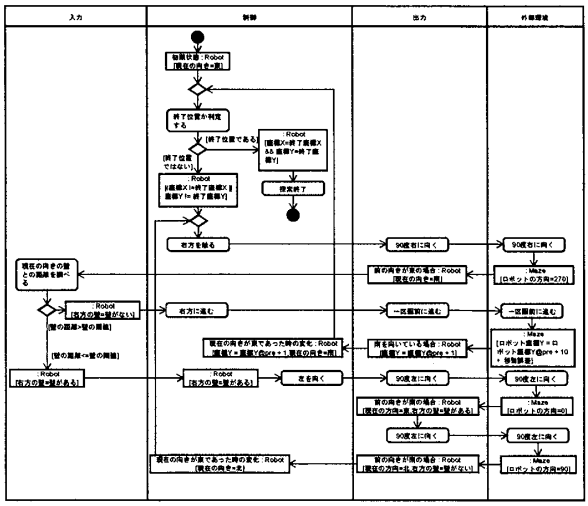

図 2　超音波センサー特化モデル

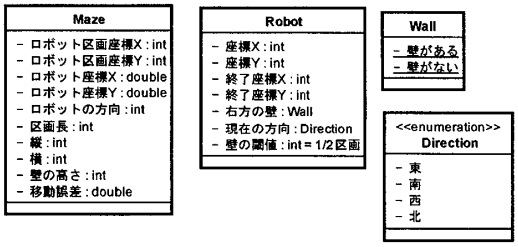

図 3　クラス図

5. まとめ

本手法では組込みシステムにおけるハードウェアの多様性に対応するため，アルゴリズム・入出力・ハードウェアの種類を段階的に決定する手法を提案した．システム要求をハードウェア・外部環境に依存しないモデルとして定義し，モデルの条件を満たしながらモデルを特化していくことで，システム要求を満たしたモデルを作成することができた．また，入出力パーティションを分け定義しているため，アルゴリズムを変更する場合であっても入出力のアクションを再利用することができる．

参考文献

[1] S.J.メラー：”MDA のエッセンス”, 翔泳社 (2004)
[2] S.J.メラー,M.J.バルサー：”Executable UML”翔泳社 (2003)