Vol.2012-OS-123 No.6
Vol.2012-EMB-27 No.6
2012/12/5

アスペクト指向遷移状態言語の非正常系の問題への適用

安倍 昌輝[†1]　川村 峰大[†1]　小倉 信彦[†2]　渡辺 晴美[†1]

組込みソフトウェアは，厳しい制約，膨大な非正常系により，複雑化する問題が知られており，関心事を分離可能な開発技術が望まれている．複雑な関心事を分離可能な技術にアスペクト指向技術がある．組込みソフトウェアの振る舞いはイベント駆動型であることが多く，状態遷移図が多用される．本論文では組込みソフトウェアの複雑化の問題を解決するために，アスペクト指向状態遷移言語を提案する．
従来，状態遷移モデルにアスペクト指向を適用した技術は複数提案されてきた．我々が提案する方法では，マークを付随させた状態により，状態単位で関心事を多次元的に管理することができる点が新しい．関心事を多次元的に管理するこが可能であることを評価するために，提案言語を飛行船システムの位置・方位を収集する部分と，I2C 通信部分に適用する．

A Case study of Aspect Oriented State Machine Language for Illegal Behavior

Masaki Ambai[†1]　Takahiro Kawamura[†1]
Nobuhiko Ogura[†2]　and Harumi Watanabe[†1]

Large amounts of constraints and illegal cases are typical concerns of embedded software. Such concerns cause the well-known problem, which makes complicate embedded software. For the sake of this problem, aspect oriented technologies are expected. These technologies can be encapsulating cross cutting concerns. The paper proposes an aspect oriented state machine language, since behavior of embedded software is often modeled by state machine chart. Many aspect-oriented state machine modeling languages have been proposed and applied to embedded software. New point in this thesis is making clear multi-dimensional separation of concern by putting marks on each state. To evaluate controlling multi-dimensional separation of concerns, we will apply the language into the parts for collecting data to acquire positions and the part of communicating by I2C on a toy-airship software.

1. はじめに

組込みソフトウェアは，システムの主要な機能を表す正常系の処理に加え，膨大な非正常系の処理を有し，さらに様々な制約を受けることから煩雑化するという問題が知られている．この問題は，正常系，非正常系，制約という複数の関心事が複雑，すなわち横断的に絡み合っている状況であり，適切な関心事の分離が求められている．

制約や非正常系は横断的関心事となることが多いことから，システム全体に分散波及することがあり，オブジェクトの隠蔽による関心の分離は難しい．また，組込みソフトウェアの振る舞いはイベント駆動型であることが多く，状態遷移図が多用される．以上から本論文では非正常系および制約による組込みソフトウェアの複雑化の問題を解決するために，アスペクト指向状態遷移言語を提案する．

組込みソフトウェアにおけるモデルの複雑化やアスペクト指向に関し様々な研究が行われている．佐々木，片山らは性能に応じた UML の変形について論じている 1)．紫合は，正常処理の振舞い状態マシンをもとに，非正常処理の状態遷移を検出・追加していく方式について述べている 2)．UML へのアスペクト指向技術の適用に関する研究は多数行われており 3)4)5)6)7)8)9)，鵜林らは，組込みソフトウェア開発への適用を提案している 3)．また，状態遷移モデルの合成が可能なアスペクト指向技術もいくつか提案されている．先進的な状態遷移モデルの合成が可能なアプローチに MATA(Modeling Aspects Using a Transformation Approach)8)がある．このアプローチでは，項グラフ書き換え理論のクリティカルペア解析 10)を適用することで，状態単位での詳細な状態遷移モデルの合成を実現している．以上の技術，アプローチのように UML に基づき標準化されたモデル駆動開発技術はモデル変換を検討する際の研究基盤として有用である．

一方で UML より限定されたセマンティクスをもち実装に近いモデルを記述する言語では，プロセッサや OS，実装依存の問題が顕在化するため，組込みソフトウェアのより実際的な問題の検討に有用であることが期待される．岡山，片山の研究では，組込みソフトウェア向け言語について，状態遷移構文，テスト構文を持たせ，処理系により異なる型のチェックを強化している 11)．

我々も，C 言語に状態遷移構文を持たせることが可能な

†1 東海大学情報通信学研究科
†2 東京都市大学環境情報学部

Vol.2012-OS-123 No.6
Vol.2012-EMB-27 No.6
2012/12/5

状態遷移言語を開発した 12)．本言語の特徴は，既存の C 言語に，提案した状態遷移記述を施せば，変換器を通すことで，状態遷移構文を解釈し，既存の C 言語に変換することができる．また，グラフ形式の状態遷移モデルをリバースすることが可能である点にある．以上の言語を MDD ロボットチャレンジの自動航行飛行船システム, ET ロボットコンテストの自動走行ロボット 13)等, 1 万行程度の規模の組込みソフトウェア開発に適用し評価した．

我々が提案する言語では，上記の状態遷移言語の状態にマークを付随させることにより，状態単位で関心事を多次元的に管理することができる点が新しい．関心事を多次元的に管理するこが可能であることを評価するために，提案言語を ESS ロボットチャレンジ 2012 13)の自動航行飛行船システムの位置・方位を収集する部分と，I2C 通信部分に適用する．

以下，2 章でアスペクト指向状態遷移言語を提案，3 章では自動航行飛行船システムに適用，4 章で議論を行う．

2. アスペクト指向状態遷移言語

2.1 提案言語の概要

組込みソフトウェアはイベント駆動であることから，その振る舞いを状態遷移図によって示すことが多い．システムの主要な関心事は，機能や正常系であることから，その単位ごとに状態遷移図を表す．非正常系や制約は,主要な関心事に対し，横断的な振る舞いであるか，主要な関心事の振る舞いを総括的に制約することがある．以上から，提案方法では，図 1 に示すとおり，振る舞い部分を表す状態遷移記述と，関心事宣言記述部からなる．アスペクト記述において，関心事ごとに各状態遷移記述部を操作可能にすることで関心事の分離を図る．また，状態単位で多次元的に関心事を扱えるように，関心事に応じたマークを状態に複数宣言することができる．以下，状態遷移記述およびアスペクト記述，ウィーブ，変換器について述べる．

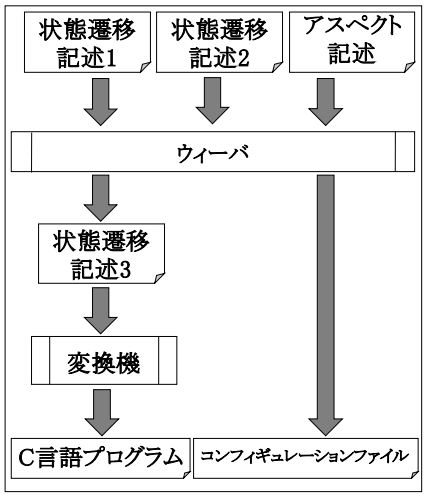

図 1 提案言語の概要

2.2 状態遷移記述

状態遷移記述とは C 言語に状態遷移モデルを記述可能とした言語であり，一般の C 言語と混在した形式で記述することにより，状態遷移モデルに基づく動作の記述が容易にできる実践的な言語を目的とする 13)．

状態遷移記述はヘッダー部，状態遷移記述部，タスク・関数部の 3 つの構成からなり，図 2 状態遷移記述の記述方法に示す．以下，各部の記述方法を示す．

(1) ヘッダー部

ここには C 言語で記述する際に必要となる include ファイルや define 定義などの宣言を記述することができる．

(2) 状態遷移記述部

状態遷移記述部の宣言は「 stm_def{} 状態遷移モデル名 1, 状態遷移モデル名 2, ... ; 」である．この際,”{”, “}” で囲まれた範囲に，状態定義，イベント定義，遷移定義を記述する．

- 状態定義は「 状態名={ is_initial か is_final, "状態説明", マーク名}; 」である．is_initial と is_final は，それぞれ開始状態と終了状態を表している．状態名は状態遷移記述内で扱うための名前で，状態説明は自然言語による理解度向上のための名前である．マーク名には後述する方法でマーク名を決定し，カンマで区切って複数個指定できる．
- イベント定義は「戻り型 イベント名 (パラメータリスト) "イベント説明"」である．イベントではユーザーの必要に応じて，戻り型とパラメータリストが指定できる．イベント名は状態遷移記述内で扱うための名前で，イベント説明は自然言語による理解度向上のための名前である．
- 遷移定義は「 戻り型 イベント名 (パラメータリスト) [ガード条件] "ガード条件説明" 遷移元状態 1,遷移元状態 2, ... -> 遷移先状態 "アクション説明" { アクション動作記述 } 」である．ここでは状態・イベント・アクションの対応関係を記述する．イベント名には上記で定義したイベント指定し，ガード条件には C 言語を受理する形で条件式を記述する．状態の遷移は「->」を使って記し，アクション動作には状態が遷移した際に行う処理を記述する．ガード条件説明とアクション説明はそれぞれ自然言語による理解度向上のための名前である．

(3) タスク部

関数・タスク部では通常の C 言語での処理をユーザーが自由に記述できる．その中で，記述内の任意の場所でイベント名を「 状態遷移モデル.イベント名() 」の形で記述することによって，イベント発行が可能である．

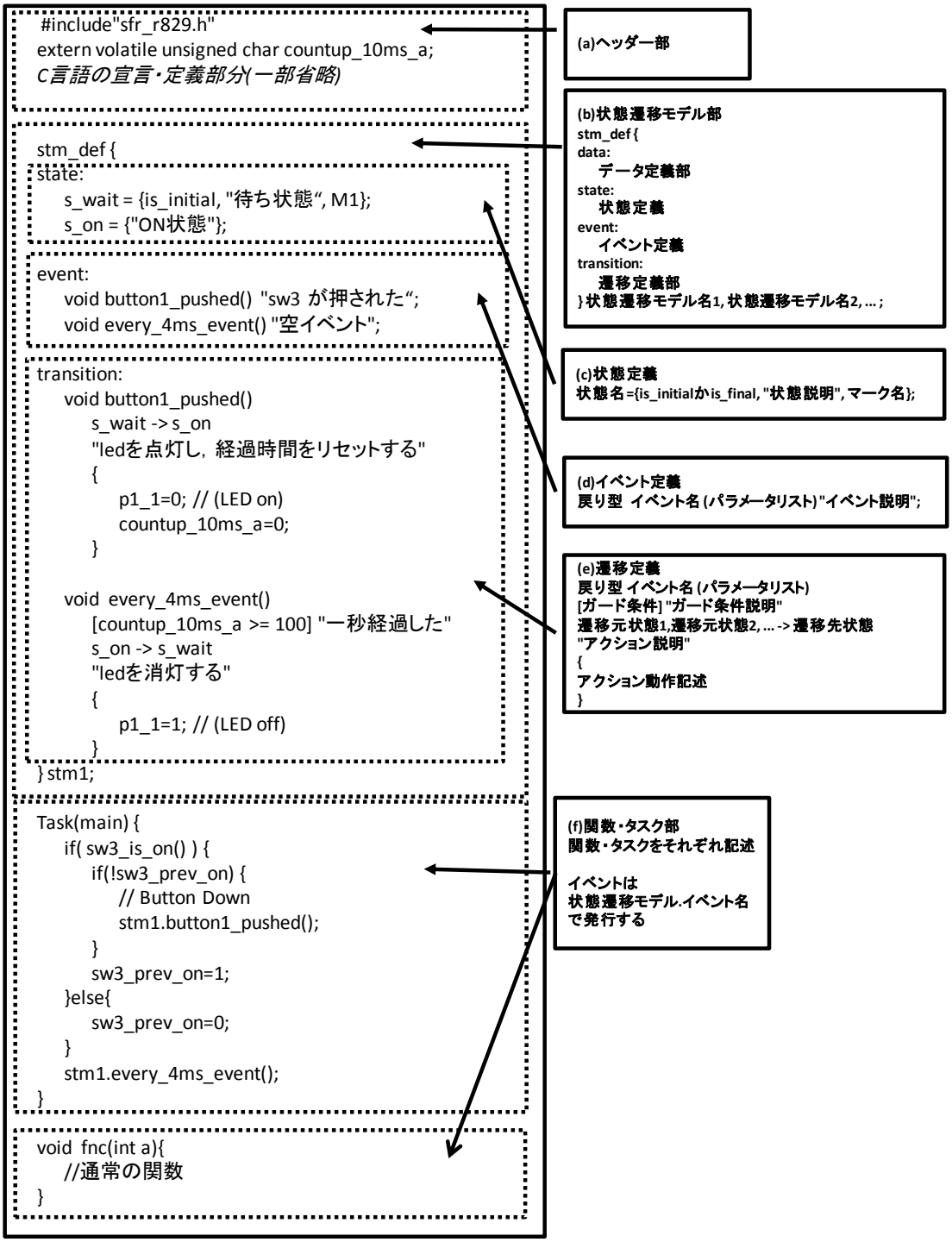

図 2 状態遷移記述の記述方法

2.3 マーク

マークとは以下の方法によって決めた，その状態を別の側面から表す記号である．

1. センサを用いている状態にはそのセンサを表すマーク名を印す．
2. 制約がある状態にはその制約を表すマーク名を印す．
3. 入力または出力を行う状態ならそれを表すマーク名を印す．

この方法を用いて，あらかじめ，コーディングを行う際に各状態へマークを印すことによって，ウィーブによる変更箇所を出戻りなくすることを可能とする．

2.4 アスペクト記述

アスペクト記述では非機能要件をまとめて管理を行う．図 3 アスペクト記述の記述方法に記述方法を示す．

(1) 関心事宣言部

関心事宣言部は「 concern 関心事名{ } 」であり，“{”, “}”で囲まれた範囲はマークとオペレーションの 2 つの宣言を記述する．マークの宣言は「 mark マーク名 1, マーク名 2, ・・・; 」である．またマークを論理積と論理和によって組み合わせ新たなマークとすることが可能である．オペレーションには合成・状態の抽出・削除・割り当ての 4 つの操作があり，以下にそれぞれの操作についての意味と記述方法を説明する．

- **合成:**合成とは指定した複数の状態に対して，別の状態を張り付ける操作とする．記述方法は「 attach(マーク名,状態遷移モデル名); 」である．マークで指定した複数の状態に状態遷移モデルがそれぞれ加わる．また特別な例として，ある状態遷移モデルの状態，すべてに合成を行いたい場合は 「 attach(all 合成先の状態遷移モデル名,合成する状態遷移モデル名); 」とする．
- **・状態遷移モデルの抽出:**状態の抽出とは指定した状態遷移モデルから一部分を抽出して，新しい状態遷移モデルとして宣言する操作である．記述方法は「新しい状態遷移モデル名 = sub_state(元となる状態遷移モデル名，抽出する一部分); 」である．新しい状態遷移モデル名は任意の文字列を指定でき，元となる状態遷移モデル名はすでに宣言してある状態遷移モデル名のみである．抽出する一部分はマークが指定でき，マークの前に^を置くことによってマーク以外の状態を指定する．
- **・削除:**削除とは指定した状態遷移モデルを削除する操作である．記述方法は「 delete(状態遷移モデル名); 」である．指定できる状態遷移モデル名はすでに宣言されているものだけである．
- **・割り当て:**割り当てとは指定した状態遷移モデルを新しいタスク上に割り当てることである．記述方法は「 assign_task(状態遷移モデル名，タスク名); 」である．状態遷移モデル名には既に宣言されたものだけ指定可能である．タスクには新しいタスクの名前を指定する．

(2) タスク宣言部

タスク宣言部は「 Task_def{ } 」であり，“{”, “}”で囲まれた範囲内には各タスクを定義する．タスク定義は「 Task タスク名{ } 」である．“{”, “}”で囲まれた範囲内に PERIODIC は周期か非周期を，RIORITY は優先度の順位を，PREEMPTION は優先度の有無を，TICK はタイマの刻み幅を，PERIOD 起床周期をそれぞれ指定する．本部分により，時間制約を扱う．

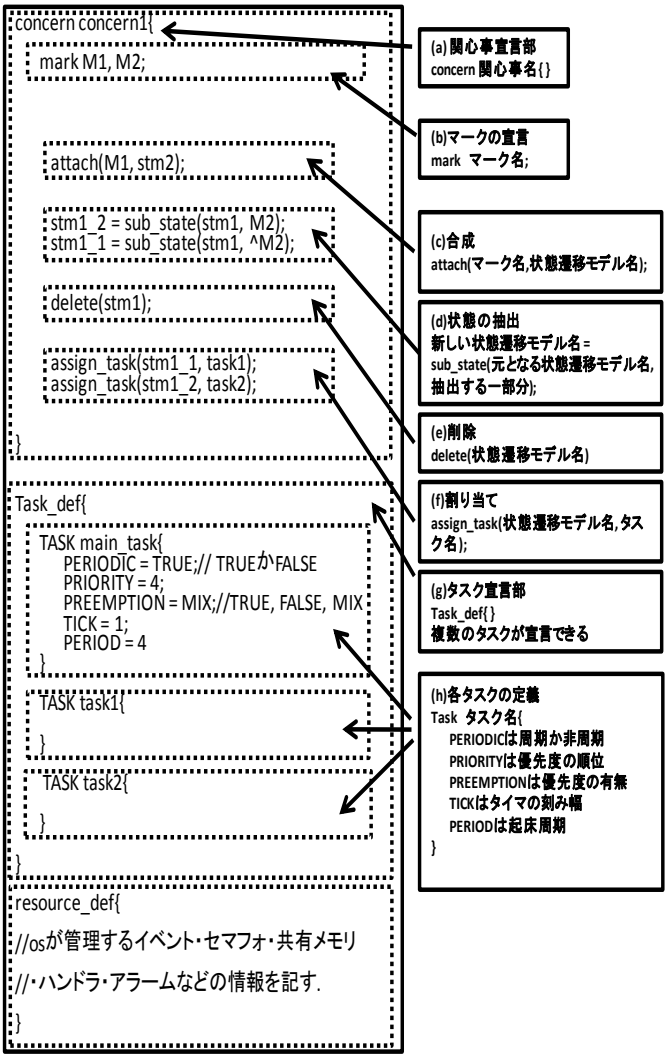

図 3 アスペクト記述の記述方法

2.5 ウィーブ

ウィーバではオペレーションで指定した操作に従って，状態言語で記述された状態遷移モデルを合成・分解する．図 9 はアスペクトの各操作の結果がどのようになされるか状態遷移モデルを用いて，説明したものである．状態遷移モデルでマークを使用するために「 状態名 ： マーク名 」と記述することで状態遷移モデルにマークを導入する．

図 4 ウィーバ適用例の(1)は attach 操作によって指定されたマーク名 M1 に状態遷移モデル stm2 を合成している．その結果，状態を 4 つ持つ状態遷移モデル stm1 になる．図 4 ウィーバ適用例の(2)では sub_state 操作によって，マーク名 M2 が印された状態を stm1_2，そうでない状態を stm1_1 としてそれぞれ状態遷移モデルに分解する．その後，delete 操作に従って，状態遷移モデル stm1 を削除している．図 4

ウィーバ適用例の(3)は assign_task 操作によって，周期を明示した task1 に tm1_1 の状態遷移モデルを割り当て，task2 に tm1_2 の状態遷移モデルを割り当てる．

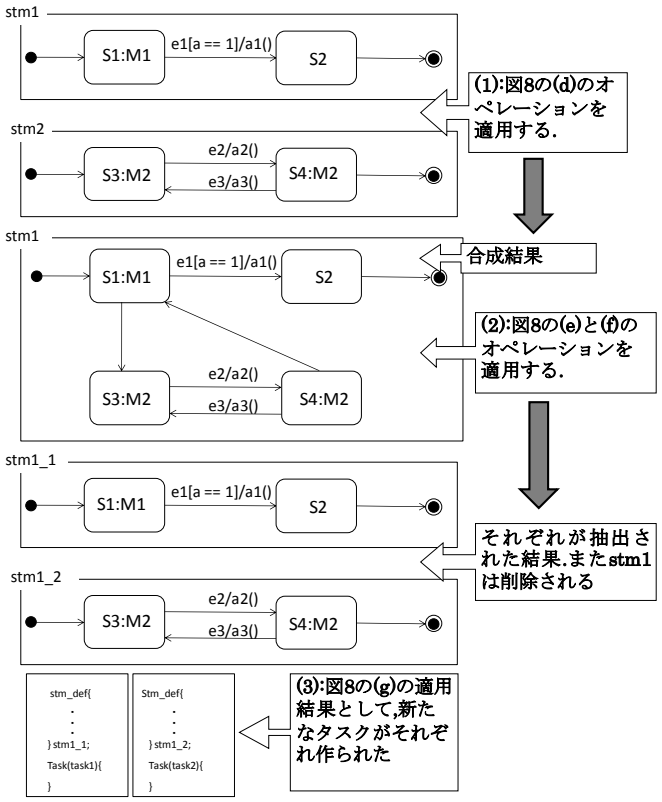

図 4 ウィーバ適用例

2.6 変換機

変換機は状態言語で記述されたプログラムを C 言語のプログラムに変換する．図 5 変換後の例には変換機の適用例を示す. この図 5 変換後の例は上記の図 2 状態遷移記述の記述方法で示した状態遷移記述を変換器に通した場合の結果である．

```
#include"sfr_r829.h"
volatile unsigned char countup_10ms_a;
//C言語の宣言・定義部分(一部省略)

// definition of stm1
struct tag_stm1{
  int state;
} stm1;

void stm1__init(struct tag_stm1 *p_stm){
  p_stm->state=0;
}

void stm1__button1_pushed(struct  tag_stm1 *p_stm){
  switch(p_stm->state){
  case 0:
          p1_1=0;
          countup_10ms_a=0;
          p_stm->state=1;
          break;
  default:
    // default case should be trapped !
  }
}

void stm1__ every_4ms_event(struct tag_stm1 *p_stm){
  switch(p_stm->state){
  case 1:
    if(countup_10ms_a >= 100)
    {
      p1_1=1;
      p_stm->state=0;
    }
    // else should be trapped!
    break;
  default:
    // default case should be trapped !
  }
}

Task(main){
  sw3_prev_on=0;
  char c;
   if(SW3_IS_ON) {
      if(!sw3_prev_on) {
         stm1__button1_pushed(&stm1);
      }
      sw3_prev_on=1;
    } else {
      sw3_prev_on=0;
    }
    stm1__ every_4ms_event(&stm1);
}

void  fnc(int a){
    //通常の関数
}
```

図 5 変換後の例

3. 自動航行飛行船システムへの適用

本章では組込みソフトウェア開発で起こりうる問題を ESS ロボットチャレンジ 2012 で用いる自動航行飛行船システムを例に述べる．

3.1 自動航行飛行船システム

提案言語は ESS ロボットチャレンジ 2012 の自動航行飛行船システムに適用した．飛行船システムの概要を図 6，構成を図 7 に示す．本システムは機上系（飛行船）と地上系からなる．飛行船から発する超音波を地上の超音波受信器で受信し，受信したタイミングと飛行船から送られてくる角度や高度と合わせて位置と進行方向を計算する．この計算結果をもとに，基地局は飛行船の振る舞いを決定し，飛行船にモータをどのように動かすかの指示を与える．図 7 **エラー! 参照元が見つかりません。**の超音波センサモジュールにより超音波を発信する．また，機上系の超音波受信モジュールにより著音波を受信することで高度を得る．ジャイロセンサモジュールから進行方向を得ることができる．このジャイロセンサとメイン MPU 間において 3.3 で述べる I2C 通信を行っている．尚，地上系のコントローラと競技用コントローラモジュールは自動航行ではなくラジコンとして動作させるための構成である．

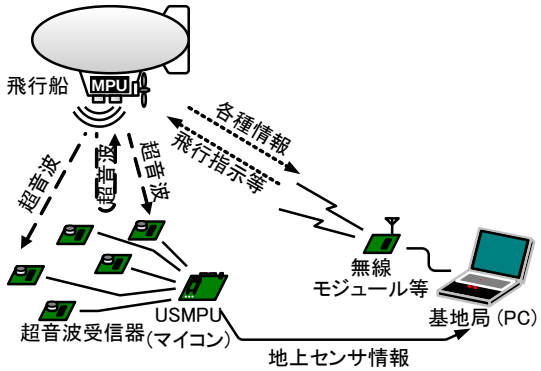

図 6　飛行船システムの概要

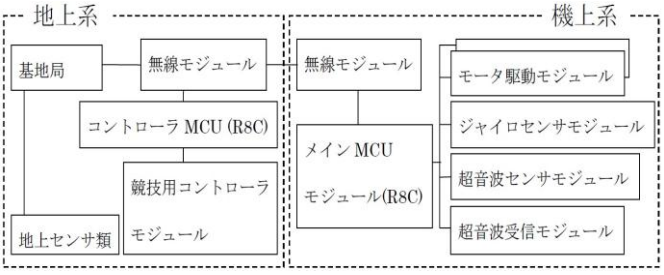

図 7 飛行船の構成

3.2 適用例 1：機上系のセンサ取得の振る舞い

本節では，正常系と非正常系の振る舞いを個別の状態遷移記述として記し，ウィーバによって合成する例を示す．図 8 は機上系のセンサ取得の振る舞いを表す正常系の状態遷移記述をグラフ形式で表した図あり，超音波センサとジャイロセンサを使って，飛行船の高度と水平角度の取得を行う処理である．この状態遷移モデルに図 9 に示す非正常系の処理を，図 10 に示すアスペクト記述部の規則に従い合成する．"M1"は"USS"と"GYRO"の二つのマークを 1 つのマークとして，attach 操作により一括に"M1"の箇所に Revise_sensor の処理を合成する．その合成後の状態遷移モデルを図 11 に示す．合成の結果，高度取得と水平角度取得の両方の状態に非正常系の処理を自動で一括に合成できた．

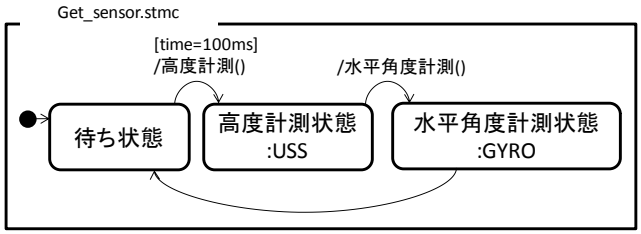

図 8 機上系のセンサ取得の状態遷移記述のグラフ形式

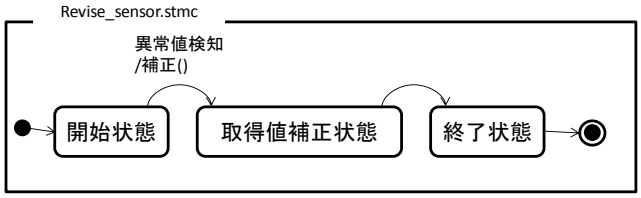

図 9 補正を行う処理の状態遷移記述のグラフ形式

```
concern error1{
 mark M1=USS & GYRO;
 attach(M1, Revise_sensor);
}
```

図 10 適用例 1 のアスペクト記述

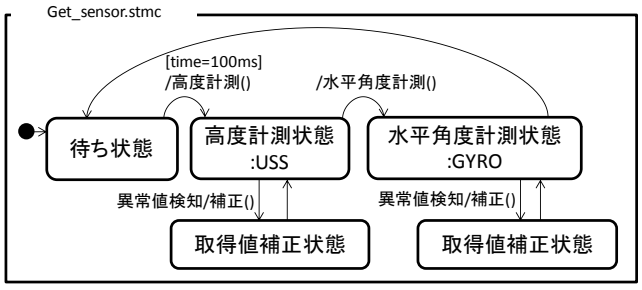

図 11 合成後のセンサ取得値の状態遷移記述のグラフ形式

Vol.2012-OS-123 No.6
Vol.2012-EMB-27 No.6
2012/12/5

3.3 適用例 2：I2C による通信の振る舞い

本節では，3.2 節よりも複雑な例として，正常系の各状態全てに同じ非正常系の振る舞いを持つ場合について示す．図 12 は I2C 通信のマスタ側の振る舞いを表す正常系の状態遷移記述のグラフ形式であり，送信と受信はそれぞれのフラグを検知することによって開始し，完了後は通信自体を終了させる処理である．I2C 通信において，ある程度の時間でバスを解放しなくてはならないので，バスを解放するための非正常系の処理を，正常系の各状態全てが持つ．バス解放の振る舞いを図 13 に示す．今回は I2C 通信のどの状態でもバスの解放が考えられるので，すべての状態に合成するための記述を図 14 に示す．本図の"all"によって I2C.stmc 上のすべての状態に状態遷移記述を追加可能である．合成後の状態遷移モデルを図 15 に示す．合成の結果，すべての状態からある一定の時間を経過したらバスを解放して通信を終了する処理が加わった．

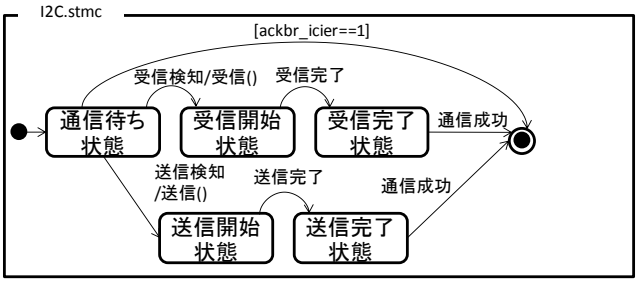

図 12 I2C 通信の状態遷移記述のグラフ形式

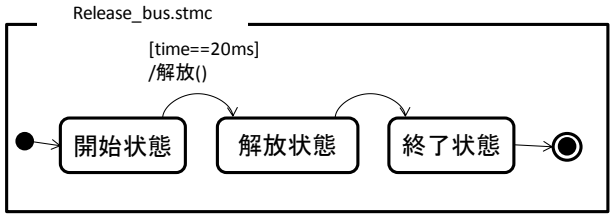

図 13 バスの解放処理の状態遷移記述のグラフ形式

```
concern error2{
attach(all I2C, Revise_sensor);
}
```

図 14 適用例 2 のアスペクト記述

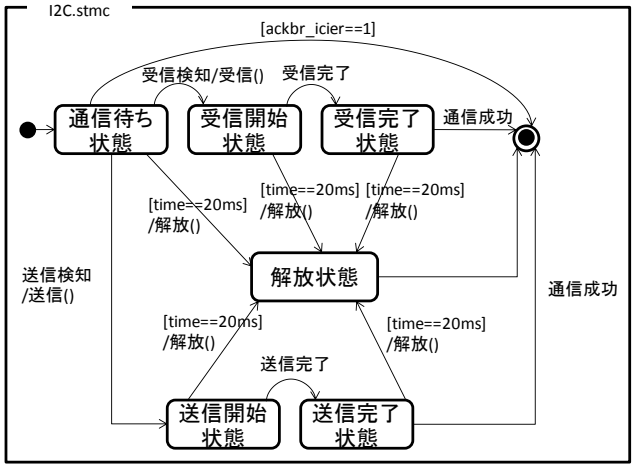

図 15 合成後の I2C 通信の状態遷移記述のグラフ形式

4. 議論

正常系と非正常系の状態遷移記述を個別に記述し，アスペクト指向記述により合成できることを 3 章で示した．3.2 では機上系のセンサ取得の振る舞いを表す正常系と，補正を行う非正常系の振る舞いを個別の状態遷移記述として表した．3.3 では I2C 通信のマスタ側の振る舞いを表す正常系と，バスを解放するための非正常系の振る舞いを個別の状態記述として表した．

従って，本言語により非正常系に関する関心事を分離できることを示した．制約に関しては，これまで ET ソフトウェアデザインロボットコンテストにおける時間制約に関する問題[15], リアルタイム OS やカーネルを搭載していないプロセッサにおいてのタイミング管理に関する問題[16]に適用し，評価してきた．

以上より，正常系，非正常系，時間制約に関する関心事について，個別の状態遷移記述として記述でき，アスペクト記述部で合成できることを確認した．

5. おわりに

本論文では，正常系，非正常系，制約等の様々な関心事により複雑化する組込みソフトウェアの問題について，アスペクト指向状態遷移言語を提案した．本言語では関心事に関連した振る舞いを状態遷移記述で記述し，アスペクト記述により，関心事間を関連付けることが可能である．また，複数のマークに各状態に宣言することで，関心事を多次元的に扱うことができる．提案言語が関心事を分離して記述し，ウィーバにより合成できることを自動航行飛行船システムに適用し示した．

今後はより多くの実際的な事例に適用し，関心事を抽出し整理していきたい．

参考文献

1) 佐々木心也, 片山徹郎 : 組込みシステムの性能に応じた UML の変形について, 組込みシステムシンポジウム 2005 論文集, 情報処理学会 pp.140-pp.143, (2005).
2) 紫合治: 組込みソフトウェアにおける非正常処理の抽出, 組込みシステムシンポジウム 2005 論文集, 情報処理学会 pp.140-pp.143, (2005).
3) 鵜林尚靖, 佐野慎治, 前野雄作, 村上聡, 片峯恵一, 橋本正明, 玉井哲雄: アスペクト指向に基づく拡張可能な MDA モデルコンパイラ, 組込みシステムシンポジウム 2004 論文集, 情報処理学会(2004).
4) T. Elrad, O. Aldawud, A. Bader: Expressing Aspects Using UML Behavioral and Structual Diagrams, Aspect-Oriented Software Development, Addison Wesley, pp. 459-478(2005).
5) J. Zhang, T. Cottenier, A. van den Berg, J. Gray: Aspect Composition in the Motorola Aspect - Oriented Modeling Weaver, Vol. 6, No. 7, Journal of Object Technology, pp.89 - pp.108 (2007).
6) N. Noda, T. Kishi: Aspect-Oriented Modeling for Embedded Software Design, 14th Asia-Pacific Software Engineering Conference (APSEC'07), pp.342-349, (2007).
7) L. Fuentes, P. Sánchez: Dynamic Weaving of Aspect-Oriented Excecutable UML Model, Transactions on Aspect-Oriented Software Development VI, pp. 1-pp.38(2009).
8) J. Whittle, P. Jayarman, A. Elkhodary, A. Moreia, João Araújo: MATA: A Unified Approach for Composing UML Aspect Model Based on Graph Transformation, Transactions on Aspect-Oriented Software Development VI, pp. 191-237(2009).
9) A. Carton, C. Driver, A. Jackson, S. Clarke: Model-Drien Theme/UML, Transactions on Aspect-Oriented Software Development VI, pp. 238-pp.266(2009).
10) M. R. Sleep, M. J. Plasmeijer, M. C. J. D van Eekelen: Term Graph Rewriting Theory and Practice, WILEY,(1993).
11) 岡山直樹, 片山徹郎: 状態遷移構文とテスト構文を導入した組込みソフトウェア向けプログラミング言語の開発, 組込みシステムシンポジウム 2010 論文集, 情報処理学会, pp. 43-48, (2010).
12) 小倉信彦, 谷川郁太, 渡辺晴美: 状態遷移言語による組込みソフトウェア開発, 情報処理学会研究報告, Vol.2011-SLDM-149 No.48, pp.1-pp. 6 (2011).
13) 満田成紀, MDD ロボットチャレンジ 2009 ワークショップ開催案内:審査員他によるワークショップ, 組込みシステムシンポジウム 2009 論文集, 情報処理学会, pp. 211-213, (2009).
14) ESS ロボットチャレンジ 2012
http://www.etrobo.jp/
15) 安倍 昌輝, 川村 峰大, 長岡 拓弥, 谷川 郁太, 原 築良, 小倉 信彦, 渡辺 晴美, 組込みソフトウェアのためのアスペクト指向による状態遷移言語の提案, 組込みシステムシンポジウム 2011 論文集, 情報処理学会, 21-1 - 21-10, (2011)
16) 川村 峰大, 安倍 昌輝, 長岡 拓弥, 谷川 郁太, 原 築良, 小倉 信彦, 渡辺 晴美, アスペクト指向状態言語の組込みソフトウェア制約への適用, 組込みシステムシンポジウム 2011 論文集, 情報処理学会研究報告, (2011)