ソフトウェア工学 122－15
（1999．3．19）

命題論理に基づいた並行システムの階層的仕様記述

宮木 衛　　高橋 薫

仙台電波工業高等専門学校
〒989-3124　仙台市青葉区上愛子字北原1番地
email: mamo@104.net, kaoru@cc.sendai-ct.ac.jp
電話: 022-391-5584　FAX: 022-391-6144

あらまし　システム設計のための形式記述技法の多くは状態遷移の概念を基本とし，システムの挙動を陽に記述することでシステムの形式仕様を得ている．しかしながら，このような仕様ではシステム機能の論理的な性質が不明瞭である．そこで，命題論理に基づいてシステムの仕様を記述する方法が提案されている．本研究では，システムの大規模化に伴う仕様記述の複雑化に対処するために，仕様記述に階層化の概念を取り入れた階層型機能要求記述および形式仕様を導出する手法を提案する．

キーワード　並行システム，命題論理，機能要求，形式仕様，状態遷移システム，階層化

Hierarchical Specification of a Concurrent System based on Propositional Logic

Mamoru MIYAKI and Kaoru TAKAHASHI

Sendai National College of Technology
Kitahara 1, Kamiayashi, Aoba-ku, Sendai, 989-3124, JAPAN
email: mamo@104.net, kaoru@cc.sendai-ct.ac.jp
Phone: +81-22-391-5584　FAX: +81-22-391-6144

Abstract　Many formal description techniques are based on the concept of state transition and a system is specified by explicitly describing its behavior. However, in such a specification, the logical property of the system becomes unclear. To cope with this problem, a description method for concurrent system specification based on a propositional logic has been proposed. In this paper, we extend the method by equipping with hierarchical description ability so that the description of complex and large-scale concurrent systems can be well dealt with.

Keywords　Concurrent System, Propositional Logic, Functional Requirement, Formal Specification, State Transition System, Hierarchical Description

1 はじめに

情報システムの複雑化に伴い，信頼性の高いシステム設計を達成するための形式記述技法 (FDTs: Formal Description Techniques) の必要性が認識されている [1]．FDT の多くは状態遷移の概念を基本とし，システムの挙動を陽に記述することでシステムの形式仕様を得ている．しかしながら，このような仕様では，システム機能の論理的な性質が不明瞭であり，また，仕様の一部の修正・変更が仕様全体に影響を与えることがある．

そこで，システムの論理的な機能要求，および，個々の機能実行のための前提条件，実行における入出力，機能実行による事後条件に着目し，命題論理に基づいてシステムの仕様を記述する方法が提案されている [2] [3]．また，並行システム仕様記述への拡張 [6] や，述語論理を用いた仕様記述への拡張 [7] も提案されている．ところが，システムが大規模になるにつれ仕様記述も複雑になり，また機能要求 (要求仕様) における素命題の増加とともに，対応する状態遷移システム (形式仕様) の状態数も爆発的に増加する．この問題に対処する方法として階層分割の方法が考えられる．

そこで本研究では，仕様記述に階層分割の概念を取り入れた階層型機能要求記述および形式仕様を導出する手法を提案する．

2 要求仕様と形式仕様

並行システムの要求仕様の記述として機能要求を用いる．並行システムはいくつかのサブシステムから成ると考え，機能要求は命題論理に基づいて記述される各サブシステムの機能の集まりから構成する．一方，並行システムの形式仕様の記述には，システムの挙動を陽に反映しうる状態遷移システムを用いる．

2.1 命題論理

$\mathcal{A}$ を素命題の全体とする．素命題は，システムを構成しているそれぞれの機能において，機能が使用可能になるための条件と機能によって変更される条件を表す．それによって，システムの状態，条件を意味する命題は，命題論理式として与えることができる．

定義 1 $\mathcal{A}$ から生成される命題論理式を次のように帰納的に定義する:

1. $A \in \mathcal{A}$ は命題論理式である．
2. f が命題論理式であるとき，$\neg f$ も命題論理式である．
3. f と g が命題論理式であるとき，$f \wedge g$，$f \vee g$，$f \Rightarrow g$ も命題論理式である． □

以下では，命題論理式を単に命題ともいう．

定義 2 素命題 A あるいは素命題の否定 $\neg A$ を A のリテラルと呼ぶ． □

定義 3 $\mathcal{A}$ から生成される命題の集合を $\mathcal{L}$ とする．このとき，命題の意味を解釈 $\mathcal{I} : \mathcal{L} \to \{\mathbf{true}, \mathbf{false}\}$ を用いて表す．ここで，**true**, **false** は命題の真偽値である．$\mathcal{I}$ と演算子 $\neg$, $\wedge$, $\vee$, $\Rightarrow$ との関係を，A を素命題，f と g を命題として，以下に定義する:

1. $\mathcal{I}(A) = \mathbf{true}$ または $\mathcal{I}(A) = \mathbf{false}$
2. $\mathcal{I}(f) = \mathbf{true}$ のとき，$\mathcal{I}(\neg f) = \mathbf{false}$
3. $\mathcal{I}(f) = \mathbf{false}$ のとき，$\mathcal{I}(\neg f) = \mathbf{true}$
4. $\mathcal{I}(f) = \mathbf{true}$, $\mathcal{I}(g) = \mathbf{true}$ のとき，$\mathcal{I}(f \wedge g) = \mathbf{true}$，それ以外のとき，$\mathcal{I}(f \wedge g) = \mathbf{false}$
5. $\mathcal{I}(f) = \mathbf{false}$, $\mathcal{I}(g) = \mathbf{false}$ のとき，$\mathcal{I}(f \vee g) = \mathbf{false}$，それ以外のとき，$\mathcal{I}(f \vee g) = \mathbf{true}$
6. $\mathcal{I}(f) = \mathbf{true}$, $\mathcal{I}(g) = \mathbf{false}$ のとき，$\mathcal{I}(f \Rightarrow g) = \mathbf{false}$，それ以外のとき，$\mathcal{I}(f \Rightarrow g) = \mathbf{true}$

□

定義 4 f を命題，$\mathcal{I}$ を解釈とする．$\mathcal{I}(f)$=**true** のとき，f は $\mathcal{I}$ の下で真であるといい，$\mathcal{I}(f)$=**false** のとき，f は $\mathcal{I}$ の下で偽であるという．$\mathcal{I}$ の下で f が真であるならば，$\mathcal{I}$ は f を満たすという．f を満たす解釈が存在するとき，f は無矛盾であるという． □

2.2 サブシステムとその機能

並行システムは互いに相互作用を行ういくつかのサブシステムから構成されると考える．サブシステムをそれぞれ関連させるために，ポートの概念を導入する．ポートを通してサブシステム間で相互作用 (入出力) を行うのである．サブシステム k のポートの有限集合を P_k と表す．また，入力の有限集合を I_k，出力の有限集合を O_k と表す．これらを用いて

サブシステムの入力アクションと出力アクションを以下のように定義する:

定義 5 サブシステム k の入力アクション Σ_k と出力アクション Δ_k は次の集合である:

$$\Sigma_k \subseteq P_k \times I_k \qquad \Delta_k \subseteq P_k \times O_k$$

□

サブシステムの動作として，ポートを通した外部からの入力，ポートを通した外部への出力，外部とは独立にサブシステム内で起こりうる内部アクション(記号 ε で表す)の3つを考えることができる．また，サブシステムが，何らかの条件を満たしているときのみ，動作が可能であると考えられる．このことから，サブシステムの状態の有限集合を考えたときに，ある動作が実行可能な状態の有限集合と，実行不可能な状態の有限集合に分ける事ができる．さらに，動作を実行することにより，条件が変更され，新たに実行可能になる動作と実行不可能になる動作が存在する．

定義 6 サブシステム k および素命題の有限集合 Θ ($\Theta \subset \mathcal{A}$) に対し，$\Theta$ から生成される命題の集合を $\mathcal{L}_\Theta$ とする．このとき，サブシステム k および素命題の有限集合 Θ に関する機能 ρ は以下の3項組である:

$$\rho = \langle f_{in},\ a,\ f_{out} \rangle$$

ここで，f_{in} は機能 ρ を実行するための前提条件 ($f_{in} \in \mathcal{L}_\Theta$)，$a$ は入力アクションまたは出力アクションまたは内部アクション ($a \in \Sigma_k \cup \Delta_k \cup \{\varepsilon\}$)，$f_{out}$ は機能 ρ の実行後に満たされるべき事後条件 ($f_{out} \in \mathcal{L}_\Theta$) である． □

分かりやすさのため，機能 ρ を $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ と記述する．入力の場合 $\rho : f_{in} \stackrel{a?}{\Rightarrow} f_{out}$ と記述し，機能を入力機能と呼ぶ．出力の場合 $\rho : f_{in} \stackrel{a!}{\Rightarrow} f_{out}$ と記述し，機能を出力機能と呼ぶ．また，内部アクションの場合 $\rho : f_{in} \stackrel{\varepsilon}{\Rightarrow} f_{out}$ と記述し，機能を内部機能と呼ぶ．入力機能および出力機能のとき，アクション a をポートの部分 p と入力/出力の部分 e に分け，$p : e$ のように表す．

2.3 機能要求の階層的記述

システムが複雑になった場合に，その機能要求記述の柔軟性の向上，記述の理解の明確化を図るために，階層的記述能力を備えた機能要求を考える．

サブシステムの機能要求記述を階層分割し，それを制約条件によって束縛されているいくつかの階層記述から成るとする．一つのサブシステムの機能要求記述中に最上位階層記述はただ一つである．ここで，最上位階層記述とは，制約条件によって束縛されていない階層記述をいう．最上位階層記述を除いて，個々の階層記述に親階層記述は必ず一つしか存在しない．親階層記述とは，制約条件で束縛している階層記述をいう．親階層記述は複数の子階層記述を持つことができる．子階層記述とは，制約条件によって束縛されている階層記述をいう．親階層記述より下の階層記述を一括して子孫階層記述ともいう．親階層記述が制約条件によって表す条件を満たしているときのみ，子孫階層記述の機能が有効になると考える．

以上のことを，図1の例を使って説明する．同図の例では，並行システムは2つのサブシステムから成る．これらサブシステムはポート p_1 と p_2 を用いて互いに相互作用，即ち，入出力を行う．図中，左側のサブシステムは3つの階層記述から成る．階層記述 HR_{11} はこのサブシステムの最上位階層記述である．HR_{11} は，階層記述 HR_{12} と HR_{13} の親階層記述でもある．HR_{12} は HR_{11} の子階層記述である．各親子階層記述間には必ず制約条件があり，どの階層記述もそれによって束縛されている．

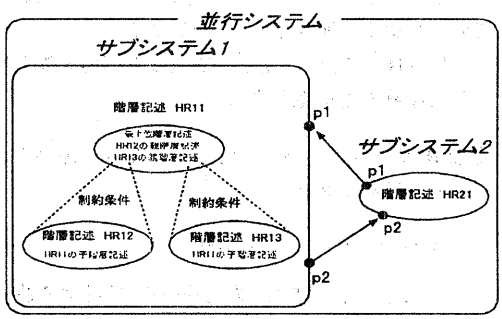

図 1: 並行システムの階層的記述例

以上の考え方の下で，並行システムの階層型機能要求を以下のように形式化する．

まず，サブシステムの機能要求の集まりとして並行システムの機能要求を定義する．

定義 7 並行システムの機能要求 $\mathcal{R}$ は次の集合で

ある:

$$\mathcal{R} = \{\mathcal{R}_k\} \quad (1 \leq k \leq n)$$

ここで，$\mathcal{R}_k$ はサブシステム k の機能要求であり，n はサブシステムの数を表す． □

次に，サブシステムの機能要求をいくつかの階層記述の集まりとして定義する．

定義 8 サブシステム k の機能要求 $\mathcal{R}_k$ は次の 5 項組である:

$$\mathcal{R}_k = \langle \{\mathcal{HR}_{ki}\}, P_k, \Sigma_k, \Delta_k, \mathcal{A}_k \rangle \quad (1 \leq i \leq m)$$

ここで，$\mathcal{HR}_{ki}$ はサブシステム k の i 番目の階層記述であり，m は階層記述の数を表す．P_k はサブシステム k のポートの有限集合，Σ_k はサブシステム k の入力アクションの集合，Δ_k はサブシステム k の出力アクションの集合，$\mathcal{A}_k$ はサブシステム k の素命題の有限集合 ($\mathcal{A}_k \subset \mathcal{A}$) である． □

なお，他の任意のサブシステム l ($1 \leq l \leq n, l \neq k$) の素命題の集合 $\mathcal{A}_l$ に対して $\mathcal{A}_k \cap \mathcal{A}_l = \phi$ とし，$\bigcup_{a=1}^{n} \mathcal{A}_a = \mathcal{A}$ とする．

$m = 1$ のときは，$\mathcal{R}_k$ が階層化されてはおらず，フラットに記述されていることになる．この場合，筆者らの従来の記述法 [6] と一致する．

制約条件を満たすときに有効となる機能の集まりとして階層記述を定義する．

定義 9 サブシステム k の i 番目の階層記述 $\mathcal{HR}_{ki}$ は次の 4 項組である:

$$\mathcal{HR}_{ki} = \langle R_{ki}, C_{ki}, \gamma_{0ki}, \mathcal{A}_{ki} \rangle$$

ここで，R_{ki} は $\mathcal{A}_{ki}$ の素命題を用いて定義される機能の有限集合である．C_{ki} は制約条件であり，**1** または，他のある 1 つの階層記述 $\mathcal{HR}_{kj}$ ($1 \leq j \leq m$, $j \neq i$) の素命題の有限集合 $\mathcal{A}_{kj}$ の素命題のリテラルからなる無矛盾な連言である．γ_{0ki} は初期条件であり，$\mathcal{A}_{ki}$ 中の全ての素命題のリテラルからなる無矛盾な連言である．$\mathcal{A}_{ki}$ はこの階層記述の素命題の有限集合 ($\mathcal{A}_{ki} \subset \mathcal{A}_k$) である． □

なお，$C_{ki} = \mathbf{1}$ であるような階層記述はサブシステム k の機能要求に，ただ 1 つだけ存在することとする．また同じサブシステムの他の任意の階層記述 $\mathcal{HR}_{kj}$ ($1 \leq j \leq m$, $j \neq i$) の素命題の集合 $\mathcal{A}_{kj}$ に対して $\mathcal{A}_{ki} \cap \mathcal{A}_{kj} = \phi$ とし，$\bigcup_{b=1}^{m} \mathcal{A}_{kb} = \mathcal{A}_k$ とする．

$C_{ki} = \mathbf{1}$ であることは $\mathcal{HR}_{ki}$ が最上位階層記述であることを意味する．そうでない場合は，定義中で示した $\mathcal{HR}_{kj}$ が親階層記述であることを意味する．

例 1 次の $\mathcal{R}$ は図 1 に対応する階層型機能要求の例である．

$\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2\}$
$\mathcal{R}_1 = \langle \{\mathcal{HR}_{11}, \mathcal{HR}_{12}, \mathcal{HR}_{13}\}, \{p_1, p_2\}, \{p_1 : a\}, \{p_2 : b\}, \{A, B, C\} \rangle$
$\mathcal{HR}_{11} = \langle \{\rho_1 : \neg A \overset{p_1:a?}{\Longrightarrow} A\}, \mathbf{1}, \neg A, \{A\} \rangle$
$\mathcal{HR}_{12} = \langle \{\rho_2 : B \overset{p_2:b!}{\Longrightarrow} \neg B\}, A, B, \{B\} \rangle$
$\mathcal{HR}_{13} = \langle \{\rho_3 : C \overset{\varepsilon}{\Longrightarrow} C\}, \neg A, C, \{C\} \rangle$
$\mathcal{R}_2 = \langle \{\mathcal{HR}_{21}\}, \{p_1, p_2\}, \{p_2 : b\}, \{p_1 : a\}, \{D\} \rangle$
$\mathcal{HR}_{21} = \langle \{\rho_4 : \neg D \overset{p_1:a!}{\Longrightarrow} D, \rho_5 : D \overset{p_2:b?}{\Longrightarrow} \neg D\}, \mathbf{1}, \neg D, \{D\} \rangle$

2.4 形式仕様

形式仕様とはシステムの挙動を陽に記述する仕様のことを指す．一般に，その多くは状態，遷移の概念を用いた状態遷移システムを意味モデルとする．そこで，ここでは合成の対象となる形式仕様として，状態遷移システムを用いる．

定義 10 状態遷移システム $\mathcal{M}$ は次の 4 項組である:

$$\mathcal{M} = \langle Q, E, \rightarrow, q_0 \rangle$$

ここで，Q は状態の有限集合，E はイベント (アクション) の有限集合，$\rightarrow$ は遷移関係 ($\rightarrow \subseteq Q \times E \times Q$)，$q_0$ は初期状態 ($q_0 \in Q$) である． □

以下，遷移 $(p, a, q) \in \rightarrow$ を $p \overset{a}{\rightarrow} q$ と書く．$p \overset{a}{\rightarrow} q$ は「状態 p において a というイベントが起きたときシステムの状態は q に遷移する」という意味である．

3 形式仕様の合成

機能要求記述によって，機能の観点からのシステムの要求仕様が得られるが，システム挙動の予測，シミュレーション，安全性検証やプロトタイプ作成

の基礎として，要求仕様に対応する形式仕様 (状態遷移システム) を得ておくことが一般に望ましいと考えられる．

本節では，与えられたサブシステムの機能要求記述からその挙動を陽に反映できる状態遷移システムを合成する手法の基本的な考え方を示す．合成の形式的な取り扱いおよび並行システム全体の対処に関しては今後の課題である．

サブシステムの機能要求 $\mathcal{R}_k$ と合成対象の状態遷移システム $\mathcal{M}$ を以下としたとき，$\mathcal{M}$ の 1) 状態，2) イベント，3) 遷移の合成の基本的な考え方を示す．

$$\mathcal{R}_k = \langle \{\mathcal{HR}_{ki}\}, P_k, \Sigma_k, \Delta_k, \mathcal{A}_k \rangle \qquad (1 \leq i \leq m)$$

$$\mathcal{M} = \langle Q, E, \rightarrow, q_0 \rangle$$

1) 状態生成　前節で述べたように，機能要求の基礎を成す素命題はシステム機能の活性の基本条件を規定するものであり，これらを用いて $\mathcal{M}$ の状態を構成することは自然である．つまり，$\mathcal{A}_k$ 中に現われるすべての素命題の無矛盾な連言 (即ち，無矛盾な命題) の各々を $\mathcal{M}$ の状態に対応づけ，その全体によって状態空間 Q を構成することができる．各階層記述 $\mathcal{HR}_{ki}$ 中で規定される初期条件 γ_{0ki} の連言をとることで，$\mathcal{M}$ の挙動の出発点としての初期状態 q_0 を構成する．各階層記述の素命題集合 $\mathcal{A}_{ki}$ 間の独立性と各 γ_{0ki} の無矛盾性から，初期状態を構成する命題は無矛盾である．

2) イベント　各階層記述 $\mathcal{HR}_{ki}$ における機能集合 R_{ki} 中の機能に現われるアクションをそのまま $\mathcal{M}$ のイベントと対応づけることができる．

3) 遷移生成　最も基本的な考え方は，階層記述 $\mathcal{HR}_{ki}$ 中の機能の実行の意味を考え，機能の前提条件を $\mathcal{M}$ の遷移元の状態，機能の事後条件を $\mathcal{M}$ の遷移先の状態と対応づけることである．そして，$\mathcal{M}$ におけるそれら状態間の遷移はまさに，機能実行の際のアクション (イベント) の生起に付随すると考える．機能中の前提/事後条件を状態と対応づけやすくするため，各機能を積和標準形に変換し，リテラルの連言だけからなる機能に分解する．前提条件に現われ，事後条件に現われないリテラルは，事後条件に連言として加えて，機能実行の意味を変えずに体裁を整える．前提/事後条件の $\mathcal{M}$ の具体的な状態への対応づけは，対応づけられるべき状態が条件を満たしているかどうかにより決定する．

階層記述の扱いを次に述べる．$m = 1$，つまり $\mathcal{R}_k$ がフラットな場合には，合成は上で述べた通りになる．$m \geq 2$ の場合は，各階層記述 $\mathcal{HR}_{ki}$ 中の各機能について，制約条件 $\mathcal{C}_{ki}$ を考慮した遷移を考える必要がある．本研究における制約条件の導入の意図は，

> その階層記述中の機能は，示された制約条件が親階層記述において満足されているときにのみ有効，つまり，機能の実行が可能である

ということである．従って，機能からの遷移の生成に関して，次のような考え方をとる:

- 当該階層記述中の機能実行に対応する状態遷移の生成は，親階層記述での制約条件の充足時にのみ行う
- 機能の前提/事後条件と状態との対応は，状態が制約条件を満たすと同時に，前提/事後条件も満たしているときにのみとる

この考え方の特殊な場合は，$\mathcal{C}_{ki} = \mathbf{1}$ であるような階層記述，つまり最上位階層記述中の機能の場合である．最上位階層記述は親階層記述を持っていないため，遷移生成に関する上の考え方においては，制約条件が常に満たされているものとすればよい．

$\mathcal{M}$ が子階層記述の機能を反映している状態にあるときに，親階層記述の機能の実行に対応する状態遷移を起こしたときには，この遷移によって何らかの例外措置により子孫階層記述の機能の実行を無効にする必要がある．これは，親階層記述が子孫階層記述の機能の活性/不活性を制約（制御）しているということからきている．なお，親階層記述において前提条件と事後条件が同じ状態で遷移される機能は状態の意味合い上，例外的に機能の実行を無効にしないこととする．この措置法としては次のものが考えられる:

- 初期条件リセット型: 現状態から親階層記述の遷移アクションとともに，遷移先 (親階層記述の状態) へ遷移させる．但し，遷移先の状態では，子孫階層記述に関わる命題は子孫階層記述の初期条件となっている．
- 状態保存型: 上と同様であるが，遷移先の状態では，子孫階層記述に関わる命題は現状態の対応する部分と同一である．

ここでは，初期条件リセット型の措置法を採用する．その理由として，状態保存型の措置法では状態の意味合いが取れないという点，さらに状態の数，遷移の数が急激に増加してしまうとにより状態遷移システムが複雑になってしまうことである．初期条件リセット型の措置法であれば，状態の意味合いが多少は崩れる点もあるが，ほぼ意味合いも取ることができ，状態，遷移の数を少なくできることにより状態遷移システムの明確化が図られる．ただ，状態保存型がユーザの記述の仕方や記述対象システムによっては有効となる場合もある．

なお，状態保存型については，次節の適用例で簡単に説明する．

以上述べた考え方の下，サブシステムの形式仕様を合成する．具体的な適用の仕方については，例とともに次節で与える．

4 適用例

本節では，並行システム機能要求記述の例と対応する形式仕様（状態遷移システム）の合成例を示す．具体的なシステムへの適用例として簡単なモバイルシステムの機能要求記述と形式仕様の合成を行った．

このシステムは次の3つのサブシステムから成る：

1. ms—モバイルステーション（移動局）
2. $base1$—移動局と交信する基地局1
3. $base2$—移動局と交信する基地局2

移動局と基地局との交信においては，まずユーザの on により移動局の電源を入れ，$enter$ により交信を開始し，$talk$ で信号のやりとりを，そして，$leave$ で交信を終了する．移動局は一度に一つの基地局とだけ交信でき，$move$ で移動することにより，もう一つの基地局に交信をスイッチする（ハンドオーバ）．

このモバイルシステムの機能要求 $\mathcal{R}$ は以下のように記述される(図2参照)．ここで，$\mathcal{R}_1$ は ms，$\mathcal{R}_2$ は $base1$，$\mathcal{R}_3$ は $base2$ の記述である．

$\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$

$\mathcal{R}_1 = \langle\{\mathcal{HR}_{11}, \mathcal{HR}_{12}, \mathcal{HR}_{13}, \mathcal{HR}_{14}, \mathcal{HR}_{15}\}, \{u, p_i \mid 1 \le i \le 2\}, \{u : on, u : off, p_i : talk \mid 1 \le i \le 2\}, \{p_i : move, p_i : enter, p_i : talk, p_i : leave \mid 1 \le i \le 2\}, \{power, loc, conn_i, Normal_i \mid 1 \le i \le 2\}\rangle$

$\mathcal{HR}_{11} = \langle R_{11}, \mathbf{1}, \neg power \wedge loc, \{power, loc\}\rangle$

$\mathcal{HR}_{12} = \langle R_{12}, power \wedge loc, \neg conn_1, \{conn_1\}\rangle$

$\mathcal{HR}_{13} = \langle R_{13}, power \wedge \neg loc, \neg conn_2, \{conn_2\}\rangle$

$\mathcal{HR}_{14} = \langle R_{14}, conn_1, Normal_1, \{Normal_1\}\rangle$

$\mathcal{HR}_{15} = \langle R_{15}, conn_2, Normal_2, \{Normal_2\}\rangle$

$\mathcal{R}_2 = \{\mathcal{HR}_{21}, \mathcal{HR}_{22}\}, \{p_i \mid 1 \le i \le 2\}, \{p_i : move, p_1 : enter, p_1 : talk, p_1 : leave \mid 1 \le i \le 2\}, \{p_1 : talk\}, \{bloc_1, bconn_1\}\rangle$

$\mathcal{HR}_{21} = \langle R_{21}, \mathbf{1}, bloc_1, \{bloc_1\}\rangle$

$\mathcal{HR}_{22} = \langle R_{22}, bloc_1, \neg bconn_1, \{bconn_1\}\rangle$

$\mathcal{R}_3 = \{\mathcal{HR}_{31}, \mathcal{HR}_{32}\}, \{p_i \mid 1 \le i \le 2\}, \{p_i : move, p_2 : enter, p_2 : talk, p_2 : leave \mid 1 \le i \le 2\}, \{p_2 : talk\}, \{bloc_2, bconn_2\}\rangle$

$\mathcal{HR}_{31} = \langle R_{31}, \mathbf{1}, \neg bloc_2, \{bloc_2\}\rangle$

$\mathcal{HR}_{32} = \langle R_{32}, bloc_2, \neg bconn_2, \{bconn_2\}\rangle$

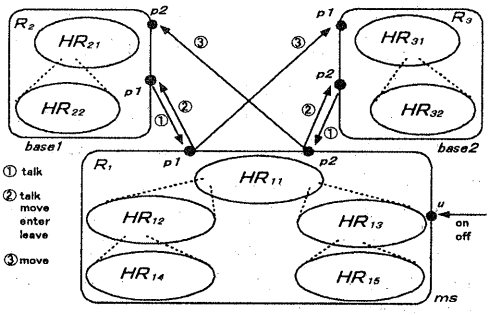

図 2: モバイルシステムの記述の枠組み

$\mathcal{R}_1$ の各階層記述の機能は表1に示す通りであり，用いている各素命題とその意味は表2に示す通りである．前節で述べた手法によって $\mathcal{R}_1$ から合成される形式仕様（状態遷移システム）を図3に示す．初期状態は1であり，各状態と命題との対応は表3に示す通りである．なお，$\mathcal{R}_2$，$\mathcal{R}_3$ については，紙面の都合上省略する．

機能要求記述，特に制約条件に関して $\mathcal{HR}_{11}$ と $\mathcal{HR}_{12}$ を用いて説明する．$\mathcal{HR}_{11}$ は，移動局の電源 on/off および移動局がどちらのエリアにいるかということを示す階層記述である．$\mathcal{HR}_{12}$ は，$base1$ との接続可否を示す階層記述である．移動局 ms が $base1$ と接続できるためには，移動局の電源が on で $base1$ のエリアにいることが条件となる．それ以外の場合には $base1$ と接続できないことになる．従って $\mathcal{HR}_{11}$ と $\mathcal{HR}_{12}$ の間の制約条件は $power \wedge loc$ になっている．これは，$\mathcal{HR}_{11}$ がこの制約条件によって表す条件を満たしているときのみ，$\mathcal{HR}_{12}$ の機能が有効になることを表している．

合成された形式仕様（図 3）の遷移に着目し，制約条件の形式仕様への反映を説明する．状態 1 から状態 3 への遷移は電源を on にするイベントが起こったとき成立する．このとき，状態に対応する命題論理式（表 3）のはじめの $power$ の部分だけが変化し，その他の部分は変化しない．これは，親階層記述 $\mathcal{HR}_{11}$ の機能の実行に関わる遷移を表している．$\mathcal{HR}_{12}$ では，その機能実行が制約条件 $power \wedge loc$ により束縛されるため，この条件を満たす状態（3 と 5）でのみ対応する遷移が可能になっている．また，前節で述べたように親階層記述の機能の実行，遷移によって子孫階層記述の機能の実行を無効にするなどといった階層化から派生する遷移がある．

それらの一部を以下に示す：

1. 状態 5 $\xrightarrow{u:off?}$ 状態 1
   $base1$ と接続中に強制電源 off

2. 状態 5 $\xrightarrow{p_2:move!}$ 状態 4 $\xrightarrow{p_2:enter!}$ 状態 6
   $base1$ と接続中に $base2$ のエリアに移動し，接続する．(暗黙的にハンドオーバが生じている)

例えば上の 1 は，ms が $base1$ と接続中にもかかわらず親階層記述の電源 off のイベントにより，強制的に電源が切れてしまう．このとき，その全ての子孫階層記述の条件の部分（表 3 の状態 5 中 $conn_1$ 以降の部分）は子孫階層記述の初期条件となるという，前節で述べたような初期条件リセット型の形になっている．またこれに対して状態保存型をとった場合には，電源 off のイベントにより強制的に電源が切れてしまうのはよいが，上で述べた無関係な部分も保存しているので電源が切れているにも関わらず，$base1$ と接続したままになってしまい，意味合いが取れていないことになる．更に，図 3 にはない状態への遷移も生成することになるため，状態数も増加してしまう．なお，これは前節でも述べたように記述の仕方や対象システムによっては有効になる場合も考えられる．

この例からわかるように，階層型機能要求記述により，従来の方法 [6] に比べ仕様記述が簡単になり，機能自体が明確化されるようになった．また，対応する状態遷移システムに変換したときも状態の数が従来の考えよりも格段に軽減されるため，状態，遷移の関係が明確化になる．

表 1: $\mathcal{R}_1$ 各階層記述の機能

R_{11}	$\neg power \overset{u:on?}{\Longrightarrow} power$
	$power \overset{u:off?}{\Longrightarrow} \neg power$
	$power \wedge loc \overset{p2:move!}{\Longrightarrow} \neg loc$
	$power \wedge \neg loc \overset{p1:move!}{\Longrightarrow} loc$
R_{12}	$\neg conn_1 \overset{p_1:enter!}{\Longrightarrow} conn_1$
	$conn_1 \overset{p_1:leave!}{\Longrightarrow} \neg conn_1$
	$conn_1 \overset{p_1:talk!}{\Longrightarrow} conn_1$
	$conn_1 \overset{p_1:talk?}{\Longrightarrow} conn_1$
R_{13}	$\neg conn_2 \overset{p_2:enter!}{\Longrightarrow} conn_2$
	$conn_2 \overset{p_2:leave!}{\Longrightarrow} \neg conn_2$
	$conn_2 \overset{p_2:talk!}{\Longrightarrow} conn_2$
	$conn_2 \overset{p_2:talk?}{\Longrightarrow} conn_2$
R_{14}	$Normal_1 \overset{\varepsilon}{\Longrightarrow} \neg Normal_1$
	$\neg Normal_1 \overset{\varepsilon}{\Longrightarrow} Normal_1$
R_{15}	$Normal_2 \overset{\varepsilon}{\Longrightarrow} \neg Normal_2$
	$\neg Normal_2 \overset{\varepsilon}{\Longrightarrow} Normal_2$

表 2: ms における素命題とその意味

素命題	意　味
$power$	移動局 ms の電源が ON
loc	移動局 ms は基地局 $base1$ のエリア内
$conn_1$	移動局 ms が基地局 $base1$ と接続中
$conn_2$	移動局 ms が基地局 $base2$ と接続中
$Normal_1$	移動局 ms が基地局 $base1$ において正常電波範囲内
$Normal_2$	移動局 ms が基地局 $base2$ において正常電波範囲内

表 3: ms の形式仕様における状態と命題の対応

状態	命題論理式
1	$\neg power \wedge loc \wedge \neg conn_1 \wedge Normal_1 \wedge \neg conn_2 \wedge Normal_2$
2	$\neg power \wedge \neg loc \wedge \neg conn_1 \wedge Normal_1 \wedge \neg conn_2 \wedge Normal_2$
3	$power \wedge loc \wedge \neg conn_1 \wedge Normal_1 \wedge \neg conn_2 \wedge Normal_2$
4	$power \wedge \neg loc \wedge \neg conn_1 \wedge Normal_1 \wedge \neg conn_2 \wedge Normal_2$
5	$power \wedge loc \wedge conn_1 \wedge Normal_1 \wedge \neg conn_2 \wedge Normal_2$
6	$power \wedge \neg loc \wedge \neg conn_1 \wedge Normal_1 \wedge conn_2 \wedge Normal_2$
7	$power \wedge loc \wedge conn_1 \wedge \neg Normal_1 \wedge \neg conn_2 \wedge Normal_2$
8	$power \wedge \neg loc \wedge \neg conn_1 \wedge Normal_1 \wedge conn_2 \wedge \neg\neg Normal_2$

5 むすび

本研究では，仕様記述の明確化と状態遷移システムでの状態数を減少させるために，仕様記述に階層分割と制約条件の概念を取り入れた階層型機能要求記述法を提案した．更に，その挙動を陽に表現する形式仕様としての状態遷移システムを合成する手法についても併せて与えた．また，本手法の適用可能性を示すために，簡略化したモバイルシステムの記述を与え，同時に，その挙動を表す形式仕様の合成を行った．

本研究の今後の課題および展開としては，以下のことがあげられる．

- 形式仕様の合成における形式的な取り扱いおよび並行システム全体への合成．
- 状態の意味合いを正確にするために，初期条件リセット型または状態保存型以外の階層化を考慮した合成の仕方の考察．
- 機能要求記述と形式仕様合成の支援システムの開発

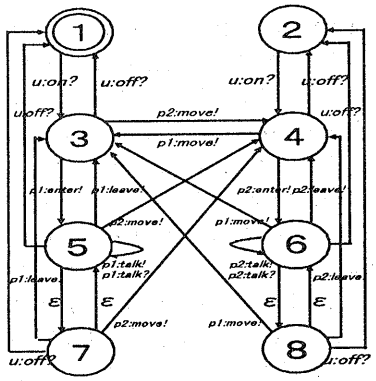

図 3: ms の形式仕様

参考文献

[1] K. J. Turner, "Using Formal Description Techniques," JOHN WILEY & SONS, 1993.

[2] 宋国換, 富樫敦, 白鳥則郎, "命題論理に基づいた要求記述法と状態遷移システムによる意味記述," 情報処理学会論文誌, Vol.37, No.4, pp.511-519, 1996.

[3] K. H. Song, A. Togashi and N. Shiratori, "Verification and Refinement for System Requirements," IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E78-A, No.11, pp.1468-1478, 1995.

[4] Y. Hirakawa and T. Takenaka, "Telecommunication Service Description using State Transition Rules," Proc. 6th Int. Work. Software Specification and Design, pp.140-147, 1991.

[5] J. Spivey, "The Z Notation," Prentice-Hall, 1992.

[6] K. Takahashi, K. Sugawara, T. Ando, Y. Kato, N. Shiratori, "Specification of a Concurrent System Based on Propositional Logic," 情報処理学会論文誌, Vol.40, No.1, pp.322-332, 1999.

[7] 金指文明, 陸暁松, 富樫敦, "システム要求からの形式仕様の導出方法," 情報処理学会論文誌, Vol.40, No.1, pp.310-321, 1999.