

複合ビューポイントに基づく仕様化プロセスの分析

西 村 一 彦[†] 本位田 真 一[†]

ソフトウェアシステムの仕様化作業を支援する数多くの技法が提案されている中で、与えられた問題に適した技法を選択したり、複数の技法を用いての仕様化作業が求められている。この要望に対して、そうした技法の規定するプロダクトや仕様化プロセスのモデル化の研究が盛んに行われている。本論文では、仕様化プロセスのモデルを形成するための新しい方法を提案する。従来、仕様化プロセスを（中間的に得られる）成果物の状態によってモデル化するアプローチが提案されている。このアプローチでは、成果物の構成や意味はプロセスごとに規定されるために、それらの表面的な違いに邪魔され、本質的に類似した行為や相違点が明らかにできない。本論文では仕様化プロセスの成果物といった表面的な情報ではなく、作業行為を遂行する際の作業者の見方や考え方によってプロセスモデルを構築する方法を提案する。具体的には、作業行為を誰が、いつ、どこで、何を、どのようにするかという視点によって整理する。さらに、作業行為の推移関係から視点の移り変わりを明らかにし、プロセスモデルを作成する。本論文では、オブジェクト指向分析（OOA）、ジャクソン法（JSD）、FORESTの仕様化プロセスモデルについて考察を行い、図書館システムの仕様化を通じてプロセスモデルが妥当であることを確認する。

An Analysis of Specification Processes from Multiple Viewpoints

KAZUHIKO NISHIMURA[†] and SHINICHI HONIDEN[†]

Software process models are seen as a basis for understanding and reasoning about software development activities. Additionally, these process models may provide a means of constructing the software specification environment to produce a high-quality product. The conventional modeling of specification processes involve the status of the products obtained in the specification process, including intermediate ones. However, the syntax or semantics of each product is determined for each process, disabling the modeling of underlying similarities and differences from their exterior characteristics. This paper proposes a new method for specification process modeling. This method is called the *Multiple Viewpoints Analysis*. Multiple Viewpoints Analysis employs the characteristics of each step taken in order to obtain the end product, while the conventional methods employ the external information of the end product. Information such as the partial knowledge about the domain for employing each activity, and the view which an engineer maintains, are classified according to the three viewpoints. These viewpoints are organized based on the traditional "who, when, where, what and how" views. Additionally, OOA, JSD, and FOREST specification process models are evaluated and verified by specification of a library system.

1. はじめに

ソフトウェア・システムの仕様化作業を支援するために、特定の分野や作業ステップに対する仕様化技法が提案されている。その中には、SA⁵⁾、JSD^{2), 8)}、OOA^{14), 15)} などのように実用化を目指したものもある。しかしながら、各々の技法はそれぞれに適した問題があり、作業者は、数多くの技法の中から適当なものを選択しなければならない¹⁷⁾。

また、一般に仕様化作業は複数の作業者によって行われる。彼らの得意とする技法は異なる場合が多く、お互いの作業を理解するためには技法間の類似点や違いを形式化することが望まれる。

これらの要求を満たす上で、(1)対象世界を表現するプロダクトモデル（例えば、リアルタイムシステムの並行プロセスをどのようなモデルで表現するか）や(2)仕様化作業のプロセスモデル（例えば、並行プロセスを仕様化するために何を考慮しなければならないか）は重要な情報である。

(1)に関しては、従来より様々な分析が行われている^{10), 17)}。(2)に関しては、a)仕様化手順が具体的な問題を中心に述べられ、一般的でない、b)説明が抽象的で作業者の判断能力に依存する、c)経験則、

[†] (株)東芝 研究開発センター システム・ソフトウェア
生産技術研究所
Systems and Software Engineering Laboratory,
Research and Development Center, Toshiba
Corporation

ノウハウが体系的に整理されていないなどの理由により、いまだ分析は不十分である。

本論文では、これらの問題点を踏まえて、仕様化作業プロセスの形式モデルを形成する新しい方法を提案する。なお、この形式モデルを仕様化プロセスモデル、あるいは単にプロセスモデルと呼ぶ。

仕様化プロセスの分析は主観的な分類による非形式的なものから、仕様化プロセスモデルの基礎概念の抽出など広範囲に渡っている。例えば、特定分野の特徴と仕様化プロセスの関係を分析した研究¹⁰⁾、プロセスを構成する基礎モデルの相違点を経験的に分類した研究^{4), 9), 11)~14), 16)}がある。本研究は後者に属するものである。仕様化プロセスモデルの基礎概念は、ソフトウェアシステムを表す概念に基づいたものが多い。例えば、Kellner⁹⁾や佐伯¹³⁾は Structure, Function, Behavior を、古宮は Process, Data, State 等の7つの概念を提案した¹¹⁾。これらの研究は、仕様化プロセスを構成する作業行為の実施結果である(中間)成果物の変化としてプロセスモデルを形成しようとするアプローチである。しかしながら、成果物の構成や意味は各々の仕様化技法によって異なり、それらの表面的な違いに邪魔されて、本質的には類似している行為や相違点を顕在化することができない¹⁸⁾。

一方、モデル化の対象も標準的なプロセスから、現実に行われた仕様化プロセス^{4), 9), 11)}まで様々である。しかしながら、具体的な事例に対してプロセスを分析するという方法は、分析結果がその事例の特徴に依存することから、事例の特徴が変わると得られたモデルの意味が失われてしまう。

本論文で提案する方法は、作業行為を遂行する際に考慮すべき情報に基づき、事例の特徴に依存しないプロセスモデルを形成することができる。すなわち、従来アプローチが対象世界を認識した結果に基づいているのに対して、我々は対象世界の認識の仕方そのものに着目している点に違いがある。この結果、技法間の本質的な類似点や違いを見出すことができる。本方法は、仕様化プロセスを作業行為に分割し、各々の作業行為の遂行時に必要な情報を誰が、いつ、どこで、何を、どのようにするかという基本的な分析視点に基づき分類する。さらに、作業行為間の推移関係から、上記の分析視点の移り変わりを明らかにする。本論文では、OOA¹⁵⁾、JSD⁹⁾、FOREST⁹⁾の仕様化プロセスのモデル化を行う。そして、各々のプロセスモデルの特徴をまとめ、図書館システムの仕様化を通じて各々の

特徴が妥当なものであることを確認する。

2. 仕様化プロセスの分析

仕様化プロセスのモデル化は(1)仕様化プロセスの階層的分割、(2)ビューポイントに基づく作業行為の分類、(3)ビューポイント間の相互関係の解析の3つのステップからなる。本章では、仕様化プロセスについて定義し、分析の基本概念であるビューポイントについて説明する。(2)および(3)については具体例を交え、次章以降で述べる。

2.1 仕様化プロセスの定義

対象世界の対象とその振舞いという概念を認識し、成果物を作成する過程を仕様化プロセスという。

定義1 仕様化プロセス

仕様化プロセス (sp) は、次に示すように階層的な構造をなす。

$$\begin{aligned} sp &= \{step_i\} \quad (i=1, \dots, k) \\ step_i &= \{activity_j\} \quad (j=1, \dots, l) \\ activity_j &= \{\{objective_x\}, \{view_y\}\} \\ &\quad (x=1, \dots, m, y=1, \dots, n) \end{aligned}$$

ここで、 $step_i$ は作業ステップ、 $objective_x$ は作業目的、 $activity_j$ は作業者が作業目的を達成するために行う最小の作業行為とする。さらに、作業行為は作業目的と、作業目的を達成するために考慮すべき見方や考え方 ($view_y$) から構成される。□

表1-表3は OOA, JSD, FOREST の標準的な仕様化プロセスを、定義1により整理したものである。

さて、仕様化プロセスは作業行為間の時間的順序関係に関する情報も含んでいる。また、別の作業行為を実施する上で必要な情報を提供するなど互いに影響しあう。このような関係を作業行為間の推移関係という。

定義2 作業行為間の推移関係

以下の条件のどちらかを満足する時、作業行為 A_1 と A_2 の間に推移関係 $A_1 \rightarrow A_2$ が成り立つ。

(i) A_1 の後に A_2 を行うという作業行為間の時間的な順序関係が技法で明示されている。

(ii) A_1 の結果である中間成果物が A_2 の遂行時に利用される。すなわち、 A_1 と A_2 の間に情報の流れがある。□

2.2 複合ビューポイント

ビューポイントとは、仕様化作業を遂行する上での対象世界の見方や意識の仕方の特徴づける概念である。例えば、与えられた問題や対象世界に対して4W

表 1 OOA の仕様化プロセス
Table 1 Specification process of OOA.

Steps	Objectives	Activities
A	Define objects	1. Identify instances 2. Describe objects 3. Name objects 4. Examine object definition
B	Classify attribute categories Determine identifications Record the values of each attribute	1. Describe fundamental features of instances 2. Define instance name or label 3. Define connection of instances 4. Define identification independently 5. Set attributes according to pre-defined category
C	Describe binary relations	1. Model relations with no conditions 2. Model relations with conditions 3. Simplify M:M relations
D	Specialize or Generalize Make relative objects	1. Incorporate all attributes except the identification 2. Identify the relations with instances
E	Determine the behavior of each objects Add the state attributes	1. Define events 2. Define states and actions of each object 3. Examine all states in life cycle
F	Identify and describe actions	1. Define data flow from attributes 2. Define data store 3. Define actions as processes

表 2 JSD の仕様化プロセス
Table 2 Specification process of JSD.

Steps	Objectives	Activities
A	Identify actions List attributes Identity entities	1. List verbs 2. Select the immediately and primitive actions 3. Resolve the conflicts by taking notice of common actions 4. Identify the attribute of each action 5. List nouns 6. Think that they can do and affect the described actions 7. Form the entity types by generalizing and abstraction 8. Name by distinct label
B	Arrange actions in their ordering by time Refine the entity structure	1. Consider whole life span of each entity 2. Describe the order which the action can happen 3. Take notice of concurrency [Add marsupial entities] 4. Specify the separate structure for each role 5. Check composite structure
C	Realize the entities and actions in a process and connections between the model and real world	1. Make the model processes 2. Connect the real world to the model 3. Merge multiple inputs stream
D	Add functions which generate required outputs Refine the model processes	1. Determine what functions are needed to generate the outputs to the model 2. Establish connections between function processes and model processes 3. Add interactive functions to generate inputs for internally generated actions 4. Introduce the time processes into the model
E	Consider some aspects of process scheduling	1. Consider potential delays which affect to function output 2. Add synchronization processes into SSD

表 3 FOREST の仕様化プロセス
Table 3 Specification process of FOREST.

Steps	Objectives	Activities
A	Identifying agents	1. Identify the agents relevant to the system to be specified 2. Construct the hierarchy of agents 3. Elaborate the agents' description
	Describe the agents	
B	Analyze data flow	1. Analyze data flow between agents 2. Check the consistency of agents' interfaces
C	Identify actions performed by each agent	1. Model what the agent can do 2. Check the production and consumption of data flows 3. Consider internal data flow 4. Constraint the co-occurrence of input and output data flows 5. Describe in natural language
D	Identify data sort and add structure to it Identify static structure of the system	1. Decompose data flows 2. Examine the agent hierarchy, data structure and action description 3. Model the passive entities 4. Identify the attributes which are tied to specific entities 5. Classify the relationships
E	Analyze permission and obligation	1. Consider condition under which each action may, can and must occur 2. Consider what enables each action, what disables it and what effects it has 3. Examine the signal data flow between the two actions 4. Tabulate the causal relations between two actions
	Make causal table	

1H (Who, Where, What, When, How) という基本的な概念がある。定義1より、作業行為は対象世界の見方・考え方 (*view*) をその構成要素とする。従って、作業行為の特徴はビューポイントに依存する。

一方、対象世界 (ソフトウェアシステム) を表すモデルは、大きく3つ (構造的側面、機能的側面、時間的側面) に分類することができる⁶⁾。従って、上述のビューポイントとこれら3つの側面の間には何らかの関係が存在する。我々は、図1のように、ビューポイントとシステムの各側面を表すモデルを対応づける。

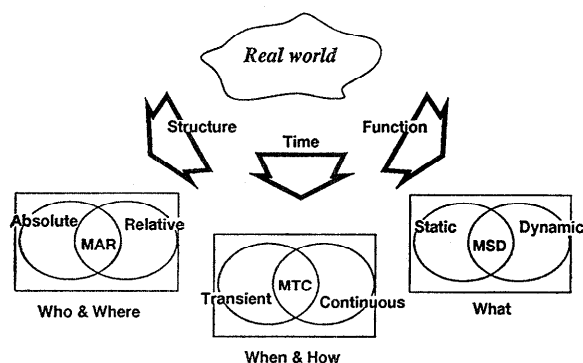


図 1 複合ビューポイント
Fig. 1 Multiple viewpoints.

その理由と各ビューポイントの定義を以下に述べる。

(1) 構造的ビューポイント

対象世界の構造的側面は、対象世界を構成する要素として何があり (**Who**)、それらが、どこに位置するか (**Where**) というビューポイントに対応する。これを構造的ビューポイントと呼ぶ。構造的ビューポイントは、技法に現れる様々な構成要素 (例えば、OOA のインスタンス、属性、JSD の実体、行動など) を定義する際に個々の構成要素のみに注目する時は **Absolute**、構成要素間の関係に注目する時は **Relative** とする。また、両者を注目する時は **MAR** とする。OOA の場合、表1の「A」オブジェクト定義の「1」インスタンスの識別は対象世界の事象のみに注目することから、**Absolute** に属す。また、「C」二項関係の記述の「1」条件なし関係のモデル化はオブジェクト間の関係に注目するので、**Relative** である。さらに、作業ステップA)の「4」オブジェクト定義の検証は、個々のオブジェクトの記述のみでなく、インスタンス間の関係にも注目するので、**MAR** に含まれる。

(2) 機能的ビューポイント

対象世界の機能的側面は、対象世界で何が行なわれるのか (**What**) というビューポイントに対応

する。これを機能的ビューポイントと呼ぶ。我々は、この機能的ビューポイントを静的なもの (Static)、動的なもの (Dynamic)、静的かつ動的なもの (MSD) に分類する。Static とは対象世界の対象とその構成や特徴といった物理的な面を明らかにし、Dynamic は対象の動作や変化の面を明らかにする。JSD の場合、表 2 の「A) 実体の識別」の「5) 名詞の列挙」は、対象世界の实体に注目するので Static である。また、作業ステップ A) の「1) 動詞の列挙」は、対象世界の動的な側面を考慮するので Dynamic である。作業ステップ D) の「1) 出力を得るための機能の決定」は、システムの振舞い (出力) という動的な側面と、どの実体と関係するかという静的な側面を考慮するので MSD である。

(3) 時間的ビューポイント

対象世界の時間的側面は、対象世界にいつ対象が現れ (When)、それがどのように振舞うか (How) というビューポイントに対応する。これを時間的ビューポイントと呼ぶ。時間的ビューポイントは、瞬間的な出来事、つまり、時点に基づいて対象世界を認識するもの (Transient)、不変的事実、つまり、時区間に基づいて対象世界を認識するもの (Continuous)、そして、両方のビューポイントに関係する MTC がある。FOREST の場合、表 3 の作業ステップ B) の「1) エージェント間のデータフロー解析」は、データフローがイベントとして用いられることから、この作業行為は Transient である。また、作業ステップ C) の「3) 内部データフローの考慮」は、エージェントの内部状態に注目するので、Continuous である。さらに、作業ステップ D) の「1) データフロー分解」は上述の B)-1) と C)-3) の両方に関係するので MTC である。

3. 作業行為の分類

OOA, JSD, FOREST は対象と振舞いという概念を明示的に区別する代表的な仕様化プロセスである。表 1-表 3 に示した各々の仕様化プロセスの作業行為を前章で述べたビューポイントに基づいて分類した結果を図 2-図 4 に示す。図中、矢印は定義 2 に述べた作業行為間の推移関係を表す。

3.1 OOA

図 2 は OOA の分類結果を示す。OOA は 20 の作業行為とそれらの間に存在する 23 の推移関係がある。

構造的ビューポイントでは、Absolute は A) オブジェクト記述と B) 属性記述の作業行為が多く、Rela-

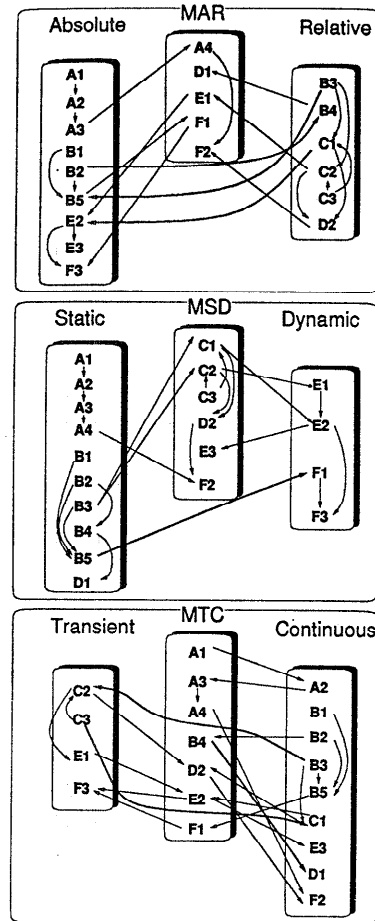


図 2 分類結果 (OOA)

Fig. 2 Classification of activities (OOA).

tive は C) 関係記述に関する作業行為が多い。そして、Relative から MAR への作業行為間の推移が目立つ。また、B) 属性記述の作業行為は MAR を経由せずに Absolute から Relative へ直接推移する。

機能的ビューポイントでは、Static は A) オブジェクト記述と B) 属性記述のすべての作業行為を含む。一方、Dynamic は E) 状態モデルと F) プロセスモデルの作業行為を含み、MSD は C) 関係記述の作業行為を中心とする。そして、Static に含まれる作業行為の数が最も多い。Static, Dynamic 間の直後の推移関係は 1 つだけである。

時間的ビューポイントでは、Transient に含まれる作業行為は少ない。しかし、Continuous の作業行為数は多く、特に B) 属性記述のほとんどの作業行為が含まれる。MTC から Continuous への推移が多い。

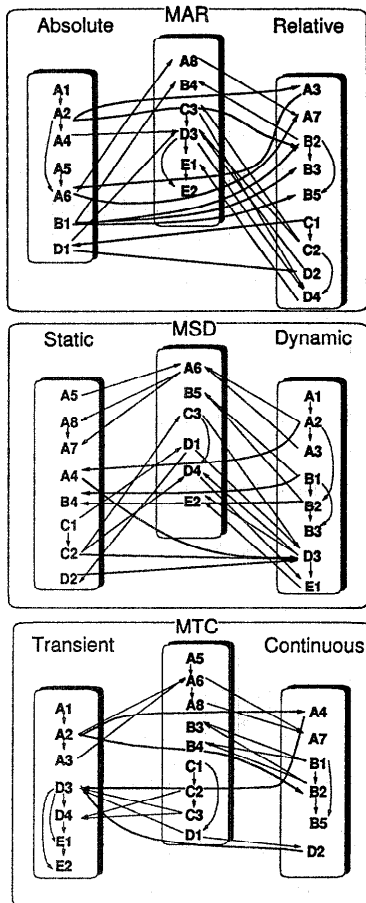


図 3 分類結果 (JSD)

Fig. 3 Classification of activities (JSD).

3.2 JSD

JSD は 22 の作業行為に対して、33 の推移関係が得られた (図 3)。

構造的ビューポイントでは、A) 実体行動ステップのほとんどの作業行為は Absolute である。しかし、A) 以外の作業ステップは MAR あるいは Relative である。つまり、作業が進むにつれて Absolute から MAR あるいは Relative へと作業が移る。作業行為間の推移関係は Absolute と Relative の間、Relative と MAR の間に多い。特に、作業ステップ A) の後半から B) の始まりにかけての作業行為では Absolute と Relative 間の直接的な推移関係が目立つ。

機能的ビューポイントでは、Dynamic から Static への直接的な推移が多い。また、ほとんどの作業ステップの作業行為が Static と Dynamic の両方に分か

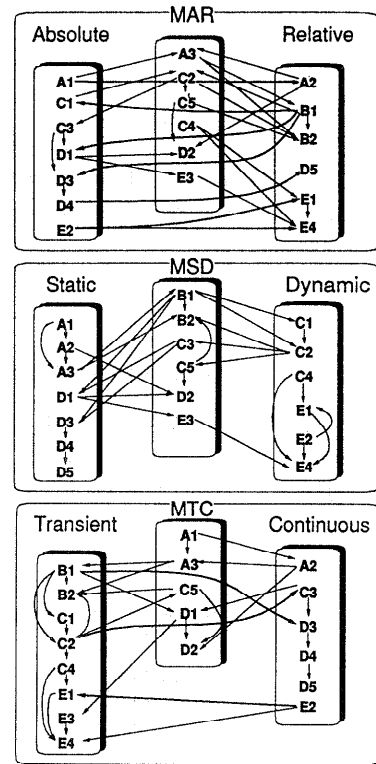


図 4 分類結果 (FOREST)

Fig. 4 Classification of activities (FOREST).

れていることが特徴的である。

時間的ビューポイントでは、Continuous は B) 実体構造ステップの作業行為が多い。また、JSD の他のビューポイントに比べて、Transient はビューポイントの内部での推移関係が多い。

3.3 FOREST

FOREST は 19 の作業行為と 29 の推移関係を得た (図 4)。

構造的ビューポイントでは、Absolute は D) 実体分析の作業行為が多く、MAR は C) 行動分析の作業行為が多い。そして、ほとんどの作業ステップが、3つのビューポイントに分かれている。推移関係は Absolute と Relative 間の直接関係と Relative と MAR 間が多い。

機能的ビューポイントでは、Static が実体に関する作業ステップ (A, D)) のほとんどの作業行為を含む。Dynamic は行動と因果解析 (C, E))、そして、MSD はデータフロー分析 (B)) の作業行為が多い。Static と Dynamic 間の直接関係はない。すなわち、MSD は Static と Dynamic を切り替えるためのビ

ューポイントと言える。また、作業ステップ A) から C) では、ビューポイント間での作業行為間の推移が頻繁だが、それ以降は Static と Dynamic の内部の推移に変わる。

時間的ビューポイントでは、Transient は行動に関する作業ステップ(C)とE))のほとんどの作業行為を含む。Continuous は実体分析(D))の作業行為を多く含む。そして、MTC は対象世界の動作主や受動的実体に関する作業ステップ(A), D))の作業行為を多く含む。

4. ビューポイント間の相互関係の解析

3章では、技法が規定する標準的な仕様化プロセスを、ある種のメタモデルである複合ビューポイントに基づいて分析した。1つの仕様化プロセスを異なる面

から分析しているので、各ビューポイントはお互いに関係があるはずである。本章では異なるビューポイント間の関係を近さと推移という点から分析し、より精密なプロセスモデルを形成する。

4.1 ビューポイントの近さ

ビューポイントの近さとは、作業行為を遂行する上で、異なる考え方や見方がどの程度同時に考慮されているかを表す。例えば、図2において、Static に含まれる作業行為と Continuous に含まれる作業行為が比較的似ている。これは OOA が対象世界の静的な面を考慮する時に、対象世界の持続的な側面にも注目する傾向が強いことを示す。

定義3 ビューポイント間の近さ

2つのビューポイント V_a, V_b の近さは、ビューポイント間の共通度 $Com(V_a, V_b)$ により得られる。

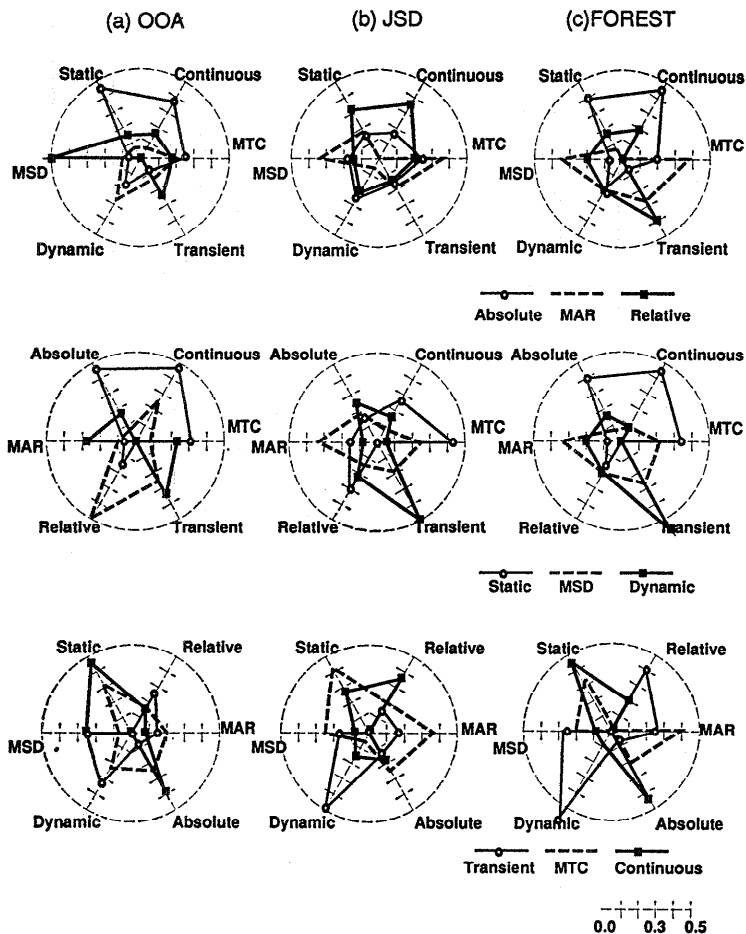


図5 ビューポイント間の近さの計算結果
Fig. 5 Adjacency among viewpoints in OOA, JSD and FOREST.

$$Com(V_a, V_b) = \frac{size(V_a \cap V_b)}{size(V_a \cup V_b)} \quad (1)$$

ここで $size(S)$ は集合 S の要素の個数 (作業行為の数) を表す。□

例えば, OOA の Static と Continuous 間では

$$Com(Static, Continuous) = 6/13 = 0.4615 \dots$$

となる。図5は各仕様プロセスのビューポイント間の共通度を計算した結果を表す。

4.2 ビューポイントの推移

ビューポイントの推移は作業行為を遂行する上での考え方や見方の移り変わりを示す。これは作業行為間の推移関係から得られる。

定義4 ビューポイントの推移

異なるビューポイント V_1 と V_2 において, V_1 に含まれる作業行為から, V_2 に含まれる作業行為間に推移関係が存在しない場合は, V_1 と V_2 の間には推移関係はない。作業行為間のすべての推移関係が同じ方向である場合 (例えば, V_1 に含まれる作業行為から V_2 に含まれる作業行為への推移関係のみ), その方向に推移があるという。作業行為間に両方向の推移関係がある場合は, V_1 と V_2 の間には両方向の推移関係があるという。□

さて, 構造的/機能的/時間的ビューポイントの間には前述の共通度があるので, あるビューポイントの推移は他のビューポイントの推移に大きく関係する。

異なるビューポイントの推移の対応関係を得るために, ビューポイント間の相関分析を行う。その手順は, a) あるビューポイント g_i において, 2つの作業行為間の推移関係を, 表4の値に対応づける。例えば, 表4の V_1, V_2, V_3 は Absolute, MAR, Relative に対応する。b) ある2つのビューポイント g_1 と g_2 に対し, a)によって得られた値の組の集合を V_{g_1} と V_{g_2} とするとき $Pairs_{g_1, g_2}$ を定義する。

$$Pairs_{g_1, g_2} = \{(x_i, y_i) | x_i \in V_{g_1}, y_i \in V_{g_2}\}$$

次に, c) 集合 $Pairs_{g_1, g_2}$ に対し, 式(2)を計算する。

$$Cor(V_{g_1}, V_{g_2}) = \frac{n \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{\sqrt{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \sqrt{n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2}} \quad (2)$$

例えば, 表5に, OOA の構造的ビューポイントと機能的ビューポイント間の相関分析の結果を示す。表中, 最も高い値を示す (Absolute, MAR, Relative)

表4 作業行為の推移のパターン

Table 4 Transition patterns of activities.

TO \ FROM	V1	V2	V3
V1	1	6	8
V2	4	2	9
V3	5	7	3

表5 OOA における構造的ビューポイントと機能的ビューポイント間の相関関係

Table 5 Correlation coefficients between Structure and Function in OOA.

Patterns	Value
(Absolute, MAR, Relative) × (Static, MSD, Dynamic)	0.149
(Absolute, MAR, Relative) × (MSD, Static, Dynamic)	0.023
(Absolute, MAR, Relative) × (MSD, Dynamic, Static)	-0.060
(Absolute, MAR, Relative) × (Static, Dynamic, MSD)	0.316
(Absolute, MAR, Relative) × (Dynamic, Static, MSD)	0.245
(Absolute, MAR, Relative) × (Dynamic, MSD, Static)	0.107

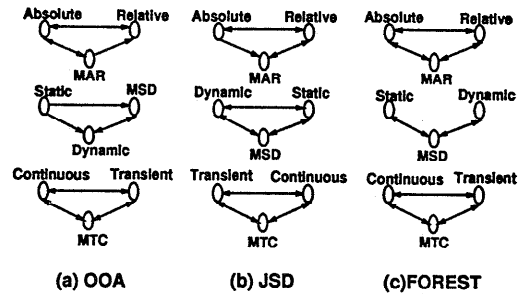


図6 観点の推移の関係

Fig. 6 Comparison of transition from a viewpoint to another.

と (Static, Dynamic, MSD) は, 構造的ビューポイントが Absolute から MAR へ, そして Relative と推移する時に, 機能的ビューポイントは Static から Dynamic へ, そして MSD と推移することを示す。

図6は, OOA, JSD, FOREST に対してすべてのビューポイント間の相関分析を行い, 各々の最も高い値を示す対応関係をまとめたものを示している。

図中, ビューポイントの推移は矢印を用いて示している。そして, 対応するビューポイントが縦に並んでいる。例えば, JSD では, 構造的ビューポイントが Absolute から MAR へ, そして Relative へ推移する時, 機能的ビューポイントでは Dynamic から MSD へ, そして Static と推移し, 時間的ビューポイントでは, Transient から MTC へ, そして Continuous へと推移する。

5. 分析結果

本章では、4章までで得られた OOA, JSD, FOREST のプロセスモデルの特徴について述べる。

5.1 OOA

OOA は、Absolute と Static, Continuous の間、そして、Relative と MSD, Transient 間の共通度が高い (図5(a))。図6より、構造的ビューポイントが Absolute から Relative へ推移する時、機能的ビューポイントは Static から Dynamic へ移り、時間的ビューポイントは Continuous から Transient へ推移する。従って、OOA は次の特徴を持つ。

(O1) 対象世界の個々の構成要素に注目する場合、対象世界の静的な面に注目する。構成要素間の関係へ見方・考え方を移動することによって、分析の視点が動的な面へと移動する。

(O2) 対象世界の瞬間的な面に注目する時、構成要素間の関係に注目する。また、個々の要素に注目する時、持続的な面に注目する傾向が強い。

また、OOA のプロセスは Static なビューポイントから始まること、そして、Static は他のビューポイントと高い割合で関係していることから、以下が言える。

(O3) 仕様化は対象世界の静的な面を重点的に考慮して行われる。

5.2 JSD

JSD は Relative と Continuous, Static 間の共通度が高い (図5(b))。これは JSD に特徴的な関係である。図6より、Absolute から Relative への推移に対して、Dynamic から Static への推移と Transient から Continuous への推移が対応している。また、仕様化プロセスは Absolute な面から始まる。この結果から JSD は次の特徴を持つ。

(J1) 対象世界の個々の構成要素に注目する場合、対象世界の動的な面に注目する。そして、対象世界の静的な面へと、その視点が移る。

(J2) 対象世界の持続的な面を考慮する時、構成要素間の関係を考慮する傾向が強い。

また、Absolute と Static, Absolute と Continuous 間の関係が比較的弱い。ゆえに次の特徴を得る。

(J3) 対象世界の個々の構成要素に注目する時には、必ずしも対象世界の静的な面を注目しない。

5.3 FOREST

図5(c)より、FOREST は Transient と Dynamic

間の関係が特に強い。また、Continuous と Static 間の共通度が高いことを特徴とする。つまり、対象世界に対して、機能的ビューポイントの視点と時間的ビューポイントの視点が密接に関係している。従って、FOREST の仕様化プロセスは次の特徴を持つ。

(F1) 対象世界の個々の構成要素に注目する場合、対象世界の静的な面に注目する。そして、構成要素間の関係へと見方が変化すると、動的な面に注目が移る傾向が強い。

(F2) 対象世界の瞬間的な面に注目する場合、対象世界の動的な面に注目する。そして、不変的な面に見方が変化すると、静的な面に注目が移る傾向が強い。

Relative は機能的ビューポイントとの間に平均的な関係を持っている。これはすなわち、

(F3) 構成要素間の関係は機能的ビューポイント全体に影響する。

6. 図書館問題の記述

本章では、図書館の設計問題¹⁾の記述結果により、前章で述べた仕様化プロセスのいくつかの特徴についてその妥当性を検証する。以下の文中で上述の特徴に対応する部分をイタリック体で表す。なお、この記述例は OOA を除いて既に発表されたものである^{2),3)}。OOA の記述は、著者らが記述した結果を用いた。図7-図9は3つの技法によって記述した図書館システムの仕様の一部を示す。

図7(a)は OOA の情報構造図、図7(b)は状態遷移図を表す。情報構造図は Library, Book, Borrower オブジェクトの定義や、サブクラス関係などの関係定義を表す。仕様化の過程では、例えば、図書館システムの静的側面である Library の属性に注目する時、Library オブジェクト自身に注目する。また、Library の状態 Loan (図7(b))を考慮するとき、Borrower と Book 間の関係に注目している (O1)。

図8(a)は JSD における Book や User などの実体定義、図8(b)は Book の構造図を表す。JSD では、個々の要素 (Book や User など) を考慮する時、それらの動作である Acquire や Lend といった動的な面に注目する。逆に、視点が要素間の関係に移ると (e. x. Book の行動間の関係)、実体の構造 (図8(b)) に視点が移る (J1)。

図9(a)は FOREST による行動表、図9(b)は ERA 図を示す。FOREST では、動作主のような個々

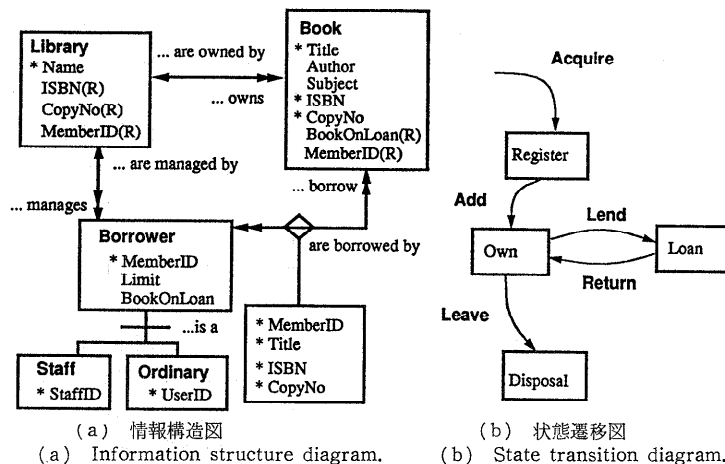


図 7 OOA による図書館問題の記述結果

Fig. 7 Part of description for the library problem by OOA.

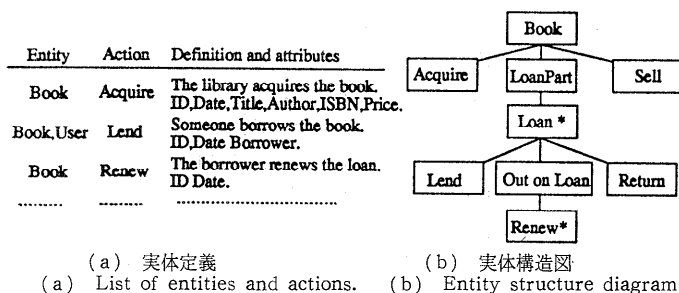


図 8 JSD による図書館問題の記述結果

Fig. 8 Part of description for the library problem by JSD.

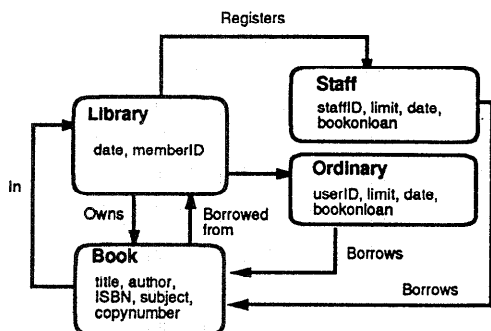
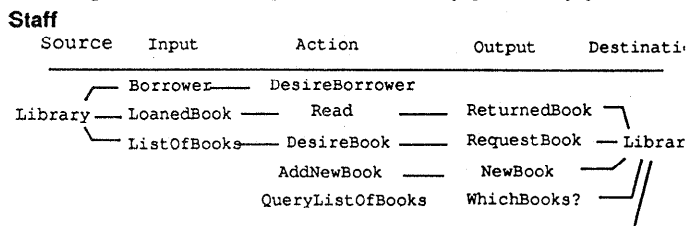


図 9 FOREST による図書館問題の記述結果

Fig. 9 Part of description for the library problem by FOREST.

の構成要素に注目するときは、主に構成要素の属性分析といった対象世界の静的な面の分析が中心となる。例えば、動作主 Staff の属性である staff-ID, bookonloan などは Staff の本質的な特徴であり、これは動作主間の関係の分析を必要としない。一方、Staff と Library の関係に注目することによって、Staff の行動 DesireBook (図 9 (a)) が得られる (F1)。

7. 考 察

本章では分析の結果得られた各仕様化技法のプロセスモデルの特徴をもとにして、各技法の比較を行う。

7.1 ビューポイントの近さ

図 5 に示したように、OOA と JSD では、ほとんどのビューポイントにおいてその近さと傾向が異なっている。例えば、OOA は構造的・機能的ビューポイントの各チャートがある軸に偏っている (構造的ビューポイントの Absolute は、Static と Continuous との共通度が高い)。これは、OOA の幾つかの作業行為が特定のビューポイントと密接な関連があることを示している。一方、JSD は各チャートの偏りが OOA に比べると小さい。例えば、図 5 (b) の構造的ビューポイントにお

ける Absolute は、他のビューポイントとの共通度が均等である。つまり、OOA とは逆に作業行為とビューポイントの関連は薄く、作業行為を遂行する時に、同時に幾つものビューポイントを意識させる仕様化プロセスといえる。

JSD と FOREST の間でも、チャートのバランスは異なっている。ただし、OOA ほど極端な違いはない。例えば、構造的ビューポイントの MAR や時間的ビューポイントの Transient, MTC が比較的似ている。そして、傾向が似ている部分の共通関係の強さは、JSD と FOREST では良く似ている。

OOA と FOREST は、全体的によく似ている。特に、機能的ビューポイントの Static、構造的ビューポイントの Absolute は全く同じである (図 5(a)(c))。ただし、構造的ビューポイントの Relative は、OOA が機能的ビューポイントに偏っているが、FOREST はバランスが良く、時間的ビューポイントに偏りがある。

ビューポイントの近さという点から各仕様化プロセスをまとめると、OOA はビューポイントを明確に分離するのに対して、JSD は複数のビューポイントを混在する。また、FOREST は OOA の傾向と JSD の傾向の双方を合わせ持った中間的な仕様化プロセスを形成している。

7.2 ビューポイントの推移

図 6 に示すように、JSD は双方向の推移が多い。これに対して、OOA は構造的と機能的ビューポイントの一部が一方に制約されている。FOREST では、機能的ビューポイントにおいて Static と Dynamic 間の直接的な流れが存在しない。以上から、OOA は仕様化プロセスの流れをある方向に規定しているが、JSD や FOREST はそうした流れを規定しない比較的自由なプロセスと言える。

ビューポイントの推移の対応関係の点で比較すると、図 6 の構造的ビューポイントにおいて、OOA、JSD、FOREST が Static に注目する時、OOA と FOREST は Absolute に注目するが、JSD は Relative に注目している。これは、ビューポイントの推移に関しても、OOA と FOREST は JSD と異なるプロセスモデルを形成していることを示している。

8. 結 論

本論文では、仕様化プロセスの本質的な特徴を顕在化するための分析方法を提案した。本方法は、1)仕様

化プロセスを作業行為に階層的に分割し、2)作業者が分析の対象を認識する時の考え方や見方 (ビューポイント)に基づき、抽出した作業行為を分類し、3)ビューポイント間の相互関係を分析するという3つのステップからなる。本方法により OOA、JSD、FOREST の3つの仕様化プロセスモデルを作成し、それらの特徴の比較を行った。

本分析方法により、以下に示すような効果が期待できる。

(1) 成果物を記述する過程の分析を行うことにより、緻密なプロセスモデルが得られる。抽象的な説明が多い仕様化プロセスにおいて、作業行為を遂行するための情報を単なる記述結果ではなく、考え方や見方といった成果物を得るための前提条件をビューポイントという概念に基づいて整理した。この結果、仕様化作業を遂行するための情報のタイプや流れを容易に把握できる。

(2) 事例の違いに左右されない一般的なプロセスモデルが得られる。本分析においては、技法の規定する標準的な作業プロセスを利用したことで、事例に依存しない一般的なプロセスモデルを得ることができた。その一方で、プロセスモデルは具体的な事例においても妥当なものであることを実証できた。

(3) 複数の技法間の類似点や違いをより顕在化することができる。従来の成果物に基づくプロセスモデルは、各々の仕様化プロセスの成果物の基礎モデルが同じならば、それらのプロセスモデルは同じである。しかしながら、本方法によれば、類似の成果物であっても、それを得る作業行為が異なるビューポイントを持つならば、プロセスモデルは異なるということを示すことができる。例えば、OOA におけるオブジェクトと JSD の実体は類似の概念である^{13),14)}が、それを得るための構造的ビューポイントは OOA では Absolute、JSD では Relative という違いがある。

以上の結果、仕様化プロセスの特徴を定性的に捉えることができた。しかしながら、この結果をより精密なものにするためには、作業行為間の後戻りの分析に加え、各々の仕様化プロセスを定量的に分析し、かつ形式的な枠組の中で議論する必要がある。例えば、仕様化プロセスを STATEMATE によって、複数のビューポイントから、時間的・空間的に分析を行う⁹⁾ことが今後の課題として考えられる。

謝辞 本研究をすすめる上で、有益な助言を頂いた査読者の方々、ならびに東京工業大学工学部佐伯元司

助教授と情報処理振興事業協会古宮誠一氏に感謝します。また、本研究の機会を与えてくださった(株)東芝研究開発センター システム・ソフトウェア生産技術研究所の西島誠一所長(現在、研究開発センター副所長)、田村信介部長に感謝します。

参 考 文 献

- 1) Problem set for the 4th IWSSD (1987).
- 2) Cameron, J.R.: An Overview of JSD, *IEEE Trans. Softw. Eng.*, Vol. 12, No. 2, pp. 222-240 (1986).
- 3) Aslett, M., Potts, C., Finkelstein, A. and Booth, J.: "Structured Common Sense": A Requirements Elicitation and Formalization Method for Modal Action Logic, Alvey FOREST Project Deliverable Report 3 (1986).
- 4) Curtis, B., Kransner, H., Shen, V. and Iscoe, N.: On Building Software Process Models under Lamppost, *Proc. 9th Int. Conf. on Software Eng.*, pp. 96-103 (1987).
- 5) DeMarco, T.: *Structured Analysis and System Specification*, Yourdon Press (1978).
- 6) Harel, D., Lachover, H., Naamad, A., Pnueri, A., Politi, M., Sherman, R. and Shtul-Trauring, A.: STATEMATE: A Working Environment for the Development of Complex Reactive Systems, *Proc. 10th Int. Conf. on Software Eng.*, pp. 396-406 (1988).
- 7) 稲田良三, 萩原剛志, 井上克朗, 鳥居宏次: ソフトウェア開発過程の形式化とその詳細化による支援システムの作成—JSDを例として—, 電子情報通信学会論文誌, Vol. J 72-D-I, No. 12, pp. 874-882 (1989).
- 8) Jackson, M.: *System Development*, Prentice-Hall, Inc. (1983).
- 9) Kellner, M.I. and Hansen, G.A.: Software Process Modeling: A Case Study, *Proc. 22nd Annual Hawaii International Conference on System Sciences*, Vol. 2, pp. 175-188 (1989).
- 10) Kelly, J.C.: A Comparison of Design Methods for Real-Time Systems, *Proc. 9th Int. Conf. on Software Eng.*, pp. 238-252 (1987).
- 11) 古宮誠一ほか: 仕様記述過程モデル化のための実験と分析, ソフトウェア工学研究会, 69-1 (1989)
- 12) 西村一彦, 本位田真一: 複合ビューポイントに基づく仕様化プロセスの定性的分析, 情報処理学会ソフトウェア工学研究会, 81-6 (1991).
- 13) 佐伯元司: ソフトウェア仕様化・設計方法論の形式化について, ソフトウェア工学研究会, 72-4 (1990).
- 14) Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W.: *Object-Oriented Modeling and Design*, Prentice-Hall, Englewood Cliffs, NJ (1991).
- 15) Shlaer, S. and Mellor, S.J.: *Object-Oriented Systems Analysis Modeling the World in Data*, Prentice-Hall, Inc. (1988).
- 16) Song, X. and Osterweil, L.J.: Toward Objective, Systematic Design-Method Comparisons, *IEEE Software*, Vol. 9, No. 3, pp. 43-53, IEEE Computer Soc. Press, Los Alamitos, CA (1992).
- 17) Webster, D.E.: Mapping the Design Information Representation Terrain, *IEEE Computer*, Vol. 21, No. 12, pp. 8-23 (1988).
- 18) Williams, L.G.: Software Process Modeling: A Behavioral Approach, *Proc. 10th Int. Conf. on Software Eng.*, pp. 174-186 (1988).

付録 図書館問題の記述の続き

6章にて説明することができなかった図書館問題の記述例による各プロセスモデルの特徴の妥当性について述べる。

1. OOA

(O2) について: 図7(b)において, Library の4つの状態は Library そのものに注目して得られる。そして, 状態間の関係を考慮する際に, イベント Add, Leave 等も考慮している。

(O3) について: OOA では対象世界の静的な面を表す情報モデルが動的な面を明らかにする状態モデル, プロセスモデルを得る上でのベースとなる。例えば, 図7(b)の Library の状態 Own は情報モデルで明らかになった Library と Book の関係により得られる。

2. JSD

(J2) について: 対象世界の持続的な面である実体の内部状態は, 実体に関連する行動の順序関係に依存する。例えば, 構造図で Book の内部状態は Acquire から Sell という行動の順序によって決定される。

(J3) について: 対象世界の静的な面である実体の属性は, 行動に関連する属性から間接的に得られる。このために属性そのものの分析を行う必要がない。例えば, 実体 Book はページ数, 出版者, 発行年数などの属性を持つと考えられるが, どれも本の行動に関係しないために不要となる。

3. FOREST

FOREST による図書館の記述結果(追加)を図10に示す。

(F2) について: 動的な面は入出力データを持つ行動として記述される(図9(a))。だが, 幾つかの行動は入力を持たないものがある。これはイベント的な

Action:	DesireBook
Enablenents:	The book is in the library and the number of books out on loan to the staff does not exceed the staff's book limit and the staff is registered with the library and the staff desires
Disablenents:	The book is not in the library or the number of books out on loan to the staff exceeds the staff's book limit or the staff is not registered with the library
Effects:	OBL(Library, CheckOutBook(Book,Self))

図 10 FOREST による図書館問題の記述結果 (付録)
Fig. 10 Part of description for the library problem
by FOREST.

行動として扱われる。例えば、行動表 (図 9 (a)) で、Library オブジェクトに対して NewBook を出力する行動 AddNewBook は、図書館システムに対する環境的なイベントとして捉えられる。

(F3) について：実体間の関係 (図 9 (b)) は問題の静的な面を表す。一方、動的な面を表す行動記述 (図 10) において、DesireBook と Library の行動 CheckOutBook は義務関係 (Effect の部分) である。これらの例から、構成要素の関係は機能的ビューポイント全体に関係する。

(平成 4 年 10 月 21 日受付)

(平成 5 年 3 月 11 日採録)



西村 一彦 (正会員)

1963 年生。1986 年東京理科大学理工学部経営工学科卒業。1988 年同大学理工学研究科経営工学専攻修士課程修了。同年 (株) 東芝入社。現在、同社研究開発センターシステム・ソフトウェア生産技術研究所に勤務。システム仕様化支援技術の研究に従事。人工知能学会、日本ソフトウェア科学会各会員。



本位田真一 (正会員)

1953 年生。1976 年早稲田大学理工学部電気工学科卒業。1978 年同大学院理工学研究科電気工学専攻修士課程修了。工学博士。同年 (株) 東芝入社。現在、同社システム・ソフトウェア生産技術研究所主任研究員。早稲田大学非常勤講師。主として、ソフトウェア工学、人工知能の研究に従事。ソフトウェアの基礎理論に興味をもつ。1986 年情報処理学会論文賞受賞。著書「ソフトウェア開発のためのプロトタイピング・ツール」(共著)、「KE 養成講座② エキスパートシステム基礎技術」(共著)、「オブジェクト指向システム分析」(共訳)。人工知能学会、日本ソフトウェア科学会、IEEE、AAAI 各会員。