

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

西村一彦 本位田真一

(株) 東芝 システム・ソフトウェア技術研究所

概要

ソフトウェアの仕様化過程において、より高品質なプロダクトを効率的に得るためには、与えられた問題に応じて仕様化プロセスを適切に選択したり、あるいは複数の仕様化プロセスを用いた仕様化環境の構築が必要である。この要望に対して、ソフトウェアプロセスモデルの研究が盛んに行なわれている。ソフトウェアプロセスモデルはソフトウェアの開発行為を理解し、モデル化するための基礎概念を与える。従来、仕様化プロセスを成果物（中間的に得られるものを含む）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が、このアプローチでは、成果物の構成や意味はプロセス毎に規定されるために、それらの表面的な違いによって、本質的には類似した行為や相違点がモデル化できない。

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、作業に対する注意点や判断基準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、判断基準を「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。次に、これらの5つの視点間の依存関係と推移関係によって、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ない、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。

Multiple Viewpoints Analysis of Specification Process

Kazuhiko Nishimura and Shinichi Honiden

Systems and Software Engineering Laboratory, Toshiba corporation

Abstract

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 a product is determined for each process, disabling the modeling of underlying similarities and differences from their exterior characteristics.

This paper proposes a method for specification process modeling which 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 criteria and notations for each step, and information obtained through experience in a certain domain, are organized according to the traditional "who, when, where, what and how" views. Afterwards, a process model based on these five views are introduced. Additionally, OOA, JSD, and FOREST specification process models are evaluated and verified by specification of a library system.

1 はじめに

OOA[7], JSD[3], SA[2]などの仕様化プロセスは、仕様化の過程で行なわれる作業を目的、成果物、判断基準などによって規定したものである。だが、これらは作業者の能力や適用範囲の特性に依存する部分が多く、より高品質なプロダクトを効率的に得るためには、与えられた問題に応じて仕様化プロセスを適切に選択したり、あるいは複数の仕様化プロセスを用いた仕様化環境の構築が必要である。

この要望に対して、ソフトウェア開発プロセスをモデル化し(ソフトウェアプロセスモデル)、そのモデルのもとで仕様化プロセスを分類する研究が盛んに行なわれている[8, 4, 9]。このソフトウェアプロセスモデルは仕様化プロセスの特徴やプロセス間の関係を明らかにするとともに、プロセスを表現するための言語体系を構築するための基礎概念を与える。従来、仕様化プロセスを成果物(中間的に得られるものを含む)の状態によってモデル化するアプローチがあるが、このアプローチでは、成果物の構成や意味はプロセス毎に規定されるために、それらの表面的な違いによって、本質的には類似した行為や相違点がモデル化できない[8]。

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、作業に対する注意点や判断基準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、判断基準を「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。次に、これらの5つの視点間の依存関係と推移関係によって、プロセスモデルを作成する。本論文ではOOA, JSD, FOREST[5]の仕様化プロセスモデルについて考察を行ない、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。

2 仕様化プロセスのモデル化

2.1 仕様化プロセスの定義

仕様化とは実世界の「対象」とその「振舞い」を認識し、記述する作業であり、仕様化プロセスは「対象」と「振舞い」を認識するための基準と、抽象化した概念をある一定の表現形式に基づいて定義するための一連の作業ステップや、その結果の表現形式を規定する。表1-3はOOA, JSD, FORESTの仕様化プロセスを作業ステップの名前(Name)、ステップの目的(Objective)、目的を達成するためにどのような点に注目すべきか、各手法で述べられている判断基準や経験則(Criteria)によりまとめたものである。

2.2 複合ビューポイント

一般に仕様化とは与えられた問題を4W1H(Who, Where, What, When, How)の面から明らかにすることである。従って、仕様化プロセスの規定する注意点や判断基準はこの5つの視点に関連するものと考えられる。我々は、この5つの分析視点を次の3つ(AR perspective, SD perspective, TC perspective)に分類した。図1に示すように、3つのperspectiveはそれぞれ、3つの視点(viewpoint)から構成される。

AR perspective(Absolute to Relative) 実世界に「だれがいて、どこにいるか」という物理的な側面に対応する。判断基準が個々の要素を対象とする時はAbsolute

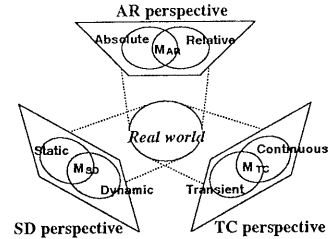


図1: 複合ビューポイント

viewpoint, 要素間の関係を対象とする時はRelative viewpointとする。また、両者に関係するものを M_{AR} とする。

SD perspective(Static to Dynamic) 「実世界で何がなされるのか」という、実世界の構成要素の機能的側面に対応する。我々は、この機能的側面を静的なもの(Static viewpoint)と、動的なもの(Dynamic viewpoint)に分類する。Staticとは実世界の「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「振舞い」の面を明らかにする。静的かつ動的なものを M_{SD} とする。

TC perspective(Transient to Continuous) 「いつ対象が現れ、どのように振舞うか」といった時間的な側面に対応する。これは瞬間的な出来事やタイミングを判断する基準(Transient viewpoint)と、不変的事実に対する基準(Continuous viewpoint)、両方の性質を持つ M_{TC} がある。

2.3 観点間の関係

上述の分類は1つの仕様化プロセスを異なる視点から分析したものである。故に、各視点はお互いに関係があるはずである。そこで、異なる観点間の関係を分析する。具体的には、観点間の類似度を計算することにより、観点間の依存関係をj得て、次にあるperspective上での観点間の推移について考察する。

ある2つの観点 V_a, V_b の類似度 $Sim(V_a, V_b)$ は次の式により得られる。

$$Sim(V_a, V_b) = \frac{size(V_a \cap V_b)}{\min(size(V_a), size(V_b))} \quad (1)$$

$$H(V_a, V_b) = \begin{cases} Dissimilar, & Sim(V_a, V_b) < 0.5 \\ Similar, & 0.5 \leq Sim(V_a, V_b) \leq 1.0 \end{cases} \quad (2)$$

ここで $size(S)$ は集合 S の要素の個数を表す。 $H(V_a, V_b)$ は類似の判定を行なう。

3 仕様化プロセスの分析

実世界の認識や抽象化の判断基準は「対象」を中心とする方法、「振舞い」を中心とする方法に大別できる。

本研究では「対象」指向として OOA[7] を、「振舞い」、すなわち、機能指向として、FOREST[5] を、そして、両者のどちらでもない方法として JSD[3] を選んだ。

本章ではこの3つの仕様化プロセスを上記の方法に従ってモデル化した結果について述べる。

3.1 OOA

OOA[7] は A) オブジェクト記述, B) 属性記述, C) 関係記述, D) オブジェクトの再構成, E) 状態モデル, F) プロセスモデルの6つの作業ステップからなる(表1)。A)-D) は実世界の事象ないしは実体を抽象化し、その特徴を属性としてまとめ、実体間の結び付きを関係として抽象化し、概念の構成と意味構造を定義する。

図2は OOA の判断基準を分類した結果である。OOA は 20 の判断基準を規定し、その間には 23 の依存関係がある。

AR Absolute は A) オブジェクト記述と B) 属性記述の判断基準が多く、Relative は C) 関係記述に関する判断基準が多い。観点の推移は Relative から M_{IR} への推移が目立つ。また、B) 属性記述の作業は M_{AR} を経由せずに Absolute から Relative へ直接推移する。

SD Static は A) オブジェクトの記述と B) 属性記述の全ての判断基準を含む。一方、Dynamic は E) 状態モデルと F) プロセスモデルの判断基準からなり、 M_{SD} は C) 関係記述の判断基準を中心とする。各観点に属している判断基準の数は Static が多い。Static, Dynamic 間の直接の関係は一つだけである。

TC Transient に対する判断基準は少ない。しかし、Continuous の判断基準は多く、特に B) 属性記述におけるほとんどの判断基準が含まれる。 M_{TC} から Continuous への推移が多い。

3.2 JSD

JSD[3] は A) 実体行動ステップ, B) 実体構造ステップ, C) 初期モデルステップ, D) 機能ステップ, E) システムタイミングステップの5つの作業ステップからなる(表2)。C) をはさんで前半の2つの作業ステップは実世界を実体とその行動という点からモデル化する部分であり、後半はシステム世界でのモデル化を行なう。

JSD は 22 の基準に対して、32 の関係が得られた(図3)。

AR A) 実体行動ステップのほとんどの判断基準は Absolute に含まれる。そして作業が進行するにつれて Relative へと判断基準が移行する。観点間の推移関係は Absolute と Relative の間、Relative と M_{AR} の間に多い。特に、後半の作業では Absolute と Relative 間の直接的な推移関係が目立つ。

SD Dynamic から Static への推移が目立っている。また、ほとんどの作業ステップの判断基準が Static と Dynamic の両方に分かれていることが特徴的である。

TC Continuous は B) 実体構造ステップの判断基準が多い。JSD の他の観点に比べて Transient は観点内部での推移関係が多い。

3.3 FOREST

FOREST は形式的な仕様化作業に至るまでに A) 動作主識別, B) データフロー分析, C) 行動分析, D) 実体分析, E) 因果分析の5つの作業ステップを順次行う(表3)。

FOREST は 19 の判断基準と 28 の関係を得た(図4)。

AR ある一つの作業に対する判断基準が各観点にほぼ均等に分かれている。Absolute は D) 実体分析の判断基準が多く、Relative は C) 行動分析に対する判断基準が多い。推移関係は Absolute と Relative の直接関係と Relative と AR の間の推移が多い。

SD 実体に関係する(A,D)ほとんどの判断基準が Static に属し、Dynamic は行動と因果解析(C,E)の判断基準からなる。 M_{SD} はデータフローに関係する(B)判断基準が多く、静的側面と動的側面を切替える間接的な観点と言える。Static と Dynamic 間の直接関係はない。また、動作主から行動を得る段階では観点間の推移が頻繁だが、プロセスの後半は Static と Dynamic 内部の推移に変わる。

TC Continuous の判断基準が他に2つに比べて少ない。これに対し、Transient は行動に関するほとんどの判断基準を含んでいる。また、いずれの観点も内部の推移が多く、他の観点との関係が少ない。

4 プロセスモデルの比較

本章では OOA, JSD, FOREST に対して得られた仕様化プロセスモデルについて考察する。

3つの仕様化プロセスに対するモデルを図5に示す。観点間の細線は式(2)が Similar なものを、太線はすべての関係の中で類似度が最大であったものを示す。矢印はその観点の中での観点の推移関係を示す。この推移関係は図2-4の観点間の推移関係を抽象化したものである。

これらのプロセスモデルは仕様化プロセスにおける2つの側面を強調している。

(観点の依存) これは仕様化プロセスにおける分析の観点間の相互依存。つまり、5つの分析の観点間の関係を表している。例えば、Static と Transient が依存しているということは、そのプロセスが静的な側面をモデル化する時に対象世界の瞬間的な部分に分析の観点を置いていることを示す。

(観点の変化) これは仕様化プロセスの分析の観点がどのように変化するかを表している。つまり、5つの分析観点の内部の変化が他の観点にどのように影響するかが明らかになる。例えば、機能の観点の推移関係と構成要素の観点が接近している時、プロセスモデルは機能の意味の変化が実世界のどこを見るかを変えていることを表す。

表4はこれらの二つの側面に対し、(1) そのプロセスモデルだけが持つ依存関係、(2) すべての依存関係の中で最も強い関係、(3) そのプロセスモデルにのみない関係、(4) 異なる

表 1: Specification Process of OOA

Name	Objectives	Criteria
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 independency 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. 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. 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: Specification Process of JSD

Name	Objectives	Criteria
A	Identify actions List attributes Identify entities	1. List of 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: Specification Process of FOREST

Name	Objectives	Criteria
A	Identifying agents Describe the agents	1. Identify the agents relevant to the system be specified 2. Construct the hierarchy of agents 3. Elaborate the agents' description
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. Examination of 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 Make causal table	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. Signal data flow between the two actions 4. Tabulate the causal relations between two actions

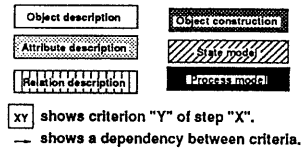
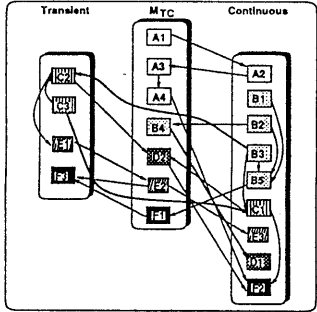
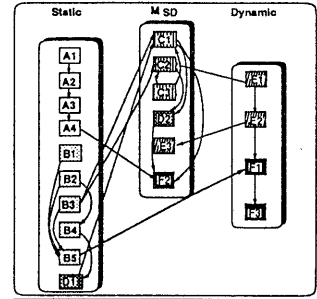
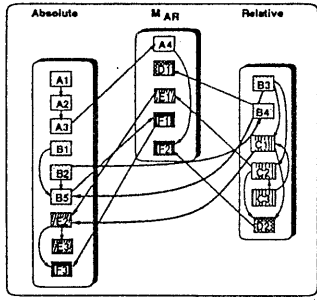


图 2: 分類結果-OOA

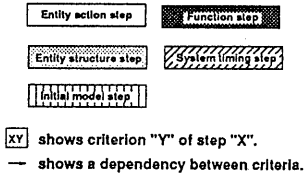
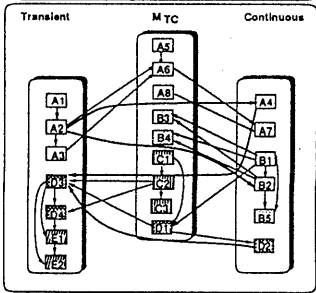
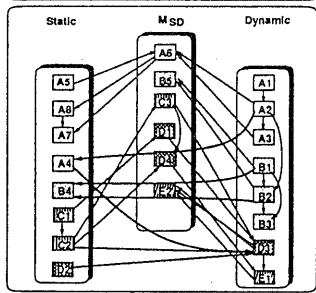
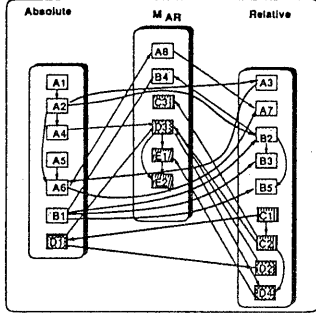


图 3: 分類結果-JSD

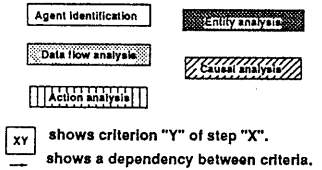
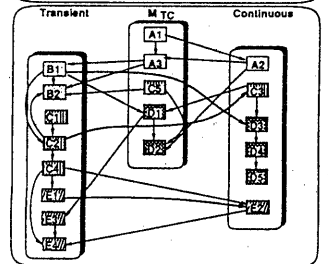
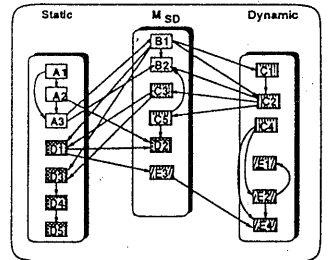
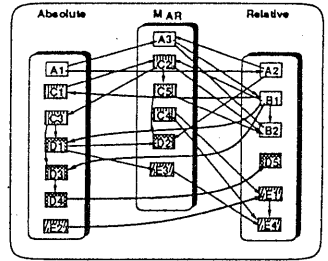


图 4: 分類結果-FOREST

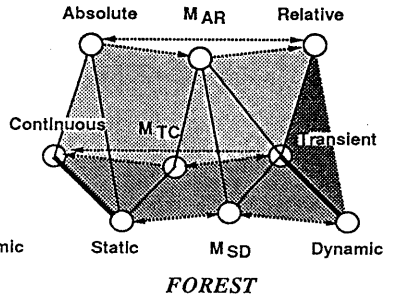
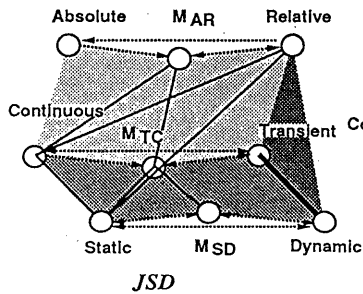
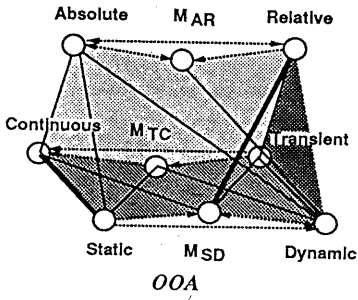


图 5: 依存關係

perspectiveの観点の推移関係の4つの結果をまとめたものである(O:OOA, J:JSD, F:FOREST).

表4: モデルの比較

	(1)	(2)	(3)	(4)
O	A:D M _{AR} :D R:M _{SD} D:M _{TC} M _{SD} :T M _{SD} :C R:S	R:M _{SD} S:C	M _{AR} :M _{TC}	A ↔ M _{AR} ↔ R S ↔ D ↔ M _{SD} C ↔ M _{TC} ↔ T
J	R:C M _{AR} :C M _{TC} :M _{SD}	D:T	A:S A:C R:T	A ↔ M _{AR} ↔ R M _{SD} ↔ D ↔ S M _{TC} ↔ T ↔ C
F	M _{AR} :M _{SD}	D:T S:C	none	A ↔ M _{AR} ↔ R S ↔ M _{SD} ↔ D C ↔ M _{TC} ↔ T

A:Absolute, R:Relative, S:Static, D:Dynamic, T:Transient, C:Continuous

4.1 OOA

OOAでは, AbsoluteはStatic, Dynamic, Continuousと, RelativeはTransientと依存している. また, 観点の推移はSD perspective上でStaticへの推移がない. 表4より, 観点がAbsoluteからRelativeへ移動する時, 静的な観点から動的な観点へ移り, TC perspective上ではContinuousからTransientへ推移する. 従って, OOAの仕様化プロセスは次のことが言える.

- (O1) 実世界の個々の構成要素に注目する場合, 実世界の静的な側面が明らかにされ, 構成要素間の関係への視点移動は動的な側面への移動をもたらす.
- (O2) 実世界の瞬間的な事実に対する観点と構成要素間の関係の観点は接近しており, また, 個々の要素に対する観点は持続的な事実に対する観点に近い.

他の2つの仕様化プロセスと比較すると, Dynamicと依存関係にある観点が4つと極めて多いこと, そして, うち2つはStaticと関係している. また, SDではStaticが作業の始まりとなることから, 次のことが結論付けられる.

- (O3) 仕様化は実世界の静的な側面を中心とする.

4.2 JSD

JSDはRelativeとContinuous, Staticの間に依存関係がある. これはJSDに特徴的な関係である. OOAやFORESTではRelativeとTransientが接近している. 表4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な視点から始まる. この結果からJSDは次の特徴を持つ.

- (J1) 実世界の個々の構成要素に注目する場合, 実世界の動的な側面を明らかにし, 実世界の静的な側面へと移る.
- (J2) 実世界の持続的な事実を考慮する時は, 構成要素間の関係に視点を置く時である.

また, AbsoluteとStatic, AbsoluteとContinuous間の関係はJSDにのみない関係である. 故に次の結論を得る.

- (J3) 実世界の個々の構成要素に対する観点は実世界の静的な側面への観点と無関係である.

4.3 FOREST

FORESTはTransientとDynamic間の関係が強く, また, ContinuousとStaticが接近していることを特徴とする. つまり, 実世界に対して, 機能的側面の視点と時間的側面の視点が密接に関係している. ARとSD, ARとTCの関係は, FORESTとOOAでは類似している. 従って, FORESTの仕様化プロセスは次の特徴を持つ.

- (F1) 実世界の個々の構成要素に注目する場合, 実世界の静的な側面が明らかにされ, 構成要素間の関係への視点移動は動的な側面への移動をもたらす.
- (F2) 実世界の瞬間的な側面は動的な側面と, 不変的な側面は静的な側面と密接である.

他の2つの仕様化プロセスとは異なり, RelativeはSDとの関係を持たない. これはすなわち,

- (F3) 構成要素間の関係は機能的側面全体に影響する.

5 図書館問題の記述

本章では仕様化プロセスモデルの妥当性を確認するために, 前章で明らかになったモデルの特徴を図書館の設計問題[6]の記述例に基づいて, 説明する. 以下で述べる記述の例はOOAを除いて既に発表されたものである[1, 5]. OOAによる記述は[7]に基づき, 我々が記述した結果を用いた.

5.1 OOA

図6-7は図書館問題の情報構造図, 状態遷移図である. Library, Book, Memberオブジェクトが定義され, Memberのサブクラスとして, Staff, Ordinaryがある. オブジェクト間では, 例えば, LibraryはBookの所有者(owns)であり, かつMemberを管理する(manages)という関係がある.

- (O1) 静的側面を表すLibraryの属性は, Libraryオブジェクトに注目する場合に得られる. Libraryの状態LoanはMemberとBook間の関係より得られ, 結果的にLend, Returnというオブジェクトの行動を導く.
- (O2) 図7において, Libraryに注目することで4つの状態が得られる. そして, 状態間の関係に視点を移すことによってイベントAdd, Leave等が得られる.
- (O3) OOAでは情報モデルが状態モデル, プロセスモデルのベースとなる. 例えば, Libraryの状態OwnはLibraryとBookの関係により明らかになる.

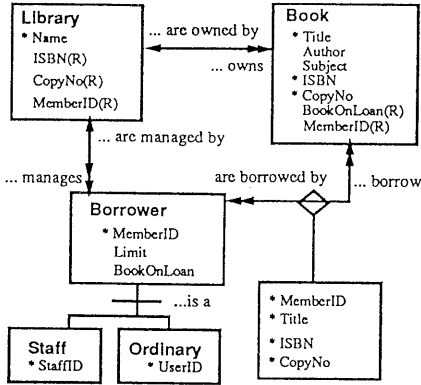
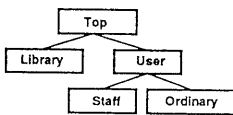


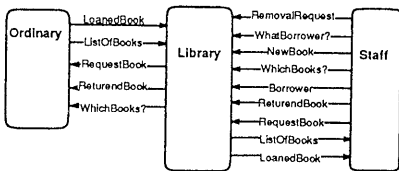
図 6: 情報構造図

Entity	Action	Definition and attributes
Book	Acquire	The library acquires the book. ID,Date,Title,Author,ISBN,Price.
Book,User	Lend	Someone borrows the book. ID,Date Borrower.
Book	Renew	The borrower renews the loan. ID,Date.
Book,User	Return	The borrower returns the book to the library. ID,Date.
Book	Sell	The book is sold. ID,Date, Vendor,Price.
User	Join	A new member join the library. Member-ID,Name,Address,Lend-Limit,Date.
User	Leave	A member leaves the library, or through inactivity is deemed to have left. Member-ID,Date.

図 8: 実体と行動の定義



Agent diagram



Data flow diagram

Action table

Staff	Source	Input	Action	Output	Destination
Library	Borrower	DesireBorrower	Read	ReturnedBook	Library
	LoanedBook	RequestBook	RequestBook	RequestBook	Library
	ListOfBooks	DesireBook	NewBook	NewBook	Library
		AddNewBook	NewBook	NewBook	Library
		QueryListOfBooks	WhichBooks?	WhichBooks?	Library
		QueryBorrower	WhatBorrower?	WhatBorrower?	Library
		RemoveBook	RemovalRequest	RemovalRequest	Library

図 10: 記述結果 -FOREST(1)

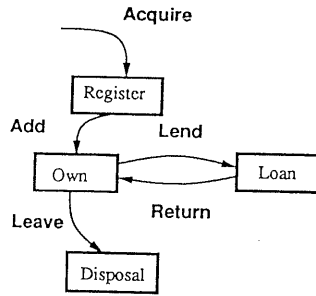


図 7: 状態遷移図

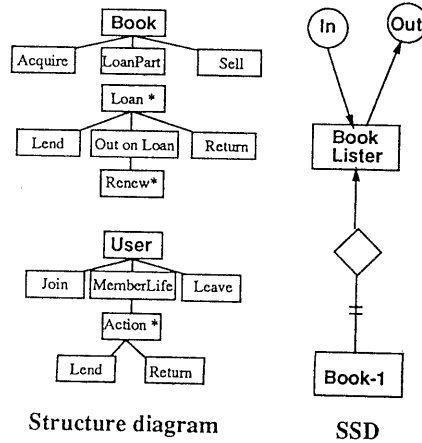
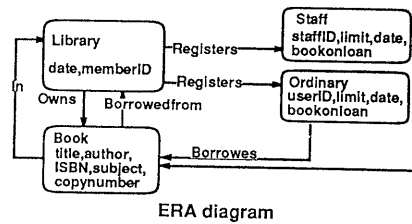


図 9: 構造図と SSD



ERA diagram

Action: DesireBook

Enablers:
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

Disablers:
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))

Action description

図 11: 記述結果 -FOREST(2)

5.2 JSD

JSDによる図書館問題の仕様化の結果を図8-9に示す[1]。図8は、BookとUser実体に対する定義と属性、図9は構造図と「著者に対する本のリストを返す」機能に対するSSDが示されている。SSDは入力(In)として「著者名」を受けとり、出力(Out)として「本のリスト」を返す。モデルプロセスとこの機能BookListerは状態ベクトル結合である。

- (J1) JSDではBook, Userといった個々に視点がある場合、動的側面の分析を行なう。視点が要素間の関係に移るとき、例えば、Bookに対する行動間の関係を分析することは、Bookという実体の構造が得られる。
- (J2) 実世界の持続的な側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) 実世界の静的な側面である実体の属性は行動に関連する属性によって間接的に得られる。このために分析を行なう必要がない。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。

5.3 FOREST

FORESTによる図書館問題の記述結果を図10-11に示す。

- (F1) 個々の構成要素を注目するときは主に構成要素の構造や属性分析といった実世界の静的側面の分析が中心となる。例えば、Staffの属性staffID、BookonloanなどはStaffの本質的な特徴である。これはオブジェクト間の関係の分析を必要としない。一方、StaffとLibraryの関係に注目すると、Staffの行動DesireBookが得られる。
- (F2) 動的側面は入力データを持つ行動として記述される。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行動として扱われる。例えば、行動表で、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) 動作主の構造(動作主図)や実体間の関係(ERA図)は問題の静的な側面を表す。一方、動的側面を表す行動記述において、DesireBookとLibraryの行動CheckOutBookは義務関係である。これらの例から、構成要素の関係は機能的側面全体に関連する。

6 結論

本論文では、仕様化プロセスの本質的な部分を顕在化するためのモデル化の方法について述べた。本手法は、1)仕様化プロセスが規定する作業ステップに対する判断基準や経験的な知識を明らかにし、2)作業者が分析の対象を認識する時の視点(perspective)に基づいて、それらの判断基準を分類し、3)視点の間の関係を分析する、という3つのステップにより、プロセスモデルを作成する。そして、本手法により、

OOA, JSD, FORESTの3つの仕様化プロセスモデルを作成し、各々のプロセスの特徴について考察を行なった。

その結果、OOAは実世界の構成要素に注目すること、機能的側面に注目することが密接である。また、構成要素間の関係に注目する時、瞬間的な事実を明らかにし、個々の要素に注目する時は不変の事実が明らかになる。そして、実世界の静的な側面の分析を軸としたプロセスである。

JSDは、実世界の構成要素の動的な側面を明らかにすることから始まり、要素間の関係に視点を移動することによって静的な側面を得る。特に個々の構成要素と静的な側面が無関係であることを特徴とする。また、要素間の関係に視点がある時、実世界の不変的な部分が明らかになる。

FORESTは実世界の構成要素の観点と静的側面の観点が、構成要素間の関係の観点と動的側面の観点到接近する。また、機能的側面と時間的な側面に対する視点が密接であり、瞬間的な事実は構成要素の関係と、持続的な事実は個々の要素と関係があることを特徴とする。さらに、構成要素の関係を明らかにする基準は実世界の機能的側面全体に影響する。

また、プロセスモデルの妥当性を確認するために、図書館システムの問題を記述した。

今回の結果は仕様化プロセス間の本質的な類似点や相違点を明らかにする。故に、複数の異なるプロセスを用いて協調的に仕様化を行なう時、プロセスモデルはプロセスの切替や中間的成果物の共有などを行なうための知識として利用できる。今後は、今回の結果をもとにして、複数の仕様化プロセスを利用した開発環境の構築を目指す。

[謝辞]

貴重なコメントをいただいた情報処理振興事業協会古宮誠一氏および東京工業大学佐伯元司助教に感謝します。また、本研究の機会を与えて下さった当研究所の西島誠一所長、田村信介部長、春木和仁主任研究員に感謝します。

References

- [1] J. R. Cameron. An Overview of JSD. *IEEE Trans. on Software Engineering*, 12(2):222-240, 1986.
- [2] T. DeMarco. *Structured Analysis and System Specification*. Yourdon Press, 1978.
- [3] M. Jackson. *System Development*. Prentice-Hall, Inc., 1983.
- [4] M. I. Kellner and G. A. Hansen. Software Process Modeling: A Case Study. *22nd Annual Hawaii International conference on System Sciences*, volume 2, pp. 175-188, 1989.
- [5] C. Potts, A. Finkelstein, M. Aslett, and J. Booth. "Structured common Sense": A Requirements Elicitation and Formalization Method for Modal Action Logic. Alvey FOREST Project Deliverable Report 3, 1986.
- [6] Problem set for the 4th IWSSD, 1987.
- [7] S. Shlaer and S. J. Mellor. *Object-Oriented Systems Analysis Modeling the Worlds in Data*. Prentice-Hall, Inc, 1988.
- [8] L. G. Williams. Software Process Modeling: A Behavioral Approach. *10th ICSE*, pp. 174-186, 1988.
- [9] 古宮他. 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, pp.1-8, 1989.

盛光印刷所