

user's view を上位、関連・実体・属性を下位オブジェクトとし、Object-Z の記述に至る分析手法

1 T-3 OOA-USA: An approach of Object-Oriented Analysis for Object-Z, taking USer's views As primary objects

田中 智

(株)野村総合研究所

1 はじめに

オブジェクト指向への期待プロセスと比較してデータは安定していると言われ、データ中心分析設計が試みられた。更に状態とその振る舞いの観点から、カプセル化出来る単位を見だし、全体を構成するオブジェクト指向の試みが行われている。オブジェクト指向の単位が見いだせれば、分析設計思想の強力な共有基盤となるので期待される。しかし提唱されている方法は、問題記述文の名詞からオブジェクト候補を見いだすなどで、開拓の余地が大きい。

2 提案する Object-Attribute Matrix を用いた分析手法と手順

以下具体例は、山崎利治氏の共通問題(情報処理 vol.25, No.9)を使用している。また、本手法の対象はビジネスアスリケーションである。

2.1 primitive user's view の作成とリファイン

1. 対象を表現する台帳、伝票、帳票を用いてシステムをシミュレーションする。
2. 同一の属性は同じ縦一列に並ぶように整理する。これは整備されてデータディクショナリになっていく。(表1)
3. 同時にエンティティ、およびプロセスとコントロールの発見、理解がなされる。

2.2 分析の要点

1. 台帳、伝票、帳票(user's view)は、実世界のモデルに他ならない。
2. 台帳、伝票、帳票(user's view)は各々状態であり、自身の状態の生成や変化を自身で司るもの、つまりオブジェクトであると見なせる。
3. USER'S VIEW は、単独のエンティティであることは希で、殆ど何らかのリレーションとなっている。エンティティは安定している。リレーションは、制度やビジネスルールを反映しており変わり易い。エンティティを見極める事は有用であるが、こと更にリレーションを分解し、よりプリミティブなリレーションを求め、静的なデータ構造として実現することには余り利益はない。
4. また、オブジェクト列の全てのオブジェクトを有するオブジェクトは、アプリケーションシステムとしてのトップレベルのオブジェクトとなる。この例では、「倉庫システム」となる。

2.3 プロセスとコントロールの記述

Z (およびObject-Z)では、状態を中心として、プロセスと同期制御を同一視野に表現出来る。Object-Attribute Matrixでも同様に表現する。(表2)

2.4 Object-Attribute Matrix

リファインされた Object-Attribute Matrix は、

1. relation

2. list(agggregation)

3. entity

4. basic type (system's, and user's)

から構成され、relation から entity を経て basic type で終端する。(表3)

2.5 Object-Z での記述

記述例示。(表4) 要旨は、

1. 上位のオブジェクト(リレーション)は、下位のオブジェクトを擁して一つの状態を保持し、そのレベルでの変化を司る。
2. 但し、下位オブジェクトの属性の詳細には直接かかわらない。初期化や属性入力のコントロールをするのみ。
3. generic type や inheritance が効果的に適応可能である。しかし、モデリングの中心となる性格とは必ずしも言えない。

3 結語

多人数が協議して効率よく開発を進めるためには、分析や設計の基本思想が、共有されることが極めて重要である。ここで言う共有には、時間を経てのコミュニケーションつまりメンテナンスも含まれる。本文では、システムがシミュレーションする外部実世界に対応する状態変数を最も自然に表現するのは、帳票、台帳の素朴なモデルである事と、Z (Object-Z) の状態、プロセス、コントロールの表現パラダイムの両者を融合する分析手法として Object-Attribute Matrix を提案した。最後に、本研究は日本IBM, SE 研究所主催(主催者山村氏)の yarnStudyGroup での Object-Z 評価分科会がきっかけとなった。この場を借りて主催者並びに分科会メンバーに謝意申し上げる。

参考文献

- [1] J.M.Spivey: "The Z Notation", Prentice Hall
- [2] Ben Potter: "An Introduction to Formal Specification and Z", Prentice Hall
- [3] Antoni Diller: "Z AN INTRODUCTION TO FORMAL METHODS"
- [4] R.Duke, et al: "Z and Object-Z for use in ODP", ISO/IEC JTC1/SC21/WG7 N
- [5] David Carrington, et al: "Object-Z: An Object-Oriented Extension to Z", Formal Description Techniques
- [6] James Rumbaugh, et al: "OBJECT-ORIENTED MODELING AND DESIGN", Prentice Hall
- [7] Peter Coad, Edward Yourdon: "OBJECT-ORIENTED ANALYSIS", Prentice Hall
- [8] Grady Booch: "Object Oriented Design", The Benjamin/Cummings

OOA-USA: An approach of Object-Oriented Analysis for Object-Z, taking USer's views As primary objects

Satoshi TANAKA

Nomura Research Institute, Ltd.

Data dictionary							
attr-spec	00000 ..99999	日付連番					
attr-name (alias)	コンテナ番号	搬入年月日	品名	数量	送り先	出庫依頼番号	空印

倉庫システム

objects	attributes						
積荷票	12345	92/08/01/15	Jack Daniel J&B I.W Herper	15 22 14			
出庫依頼票			Old Per	12	京都		
出庫指示書	23456 45678		Old Per	8 4	京都	98765 98765	有 空
不足リスト			Jack Daniel Swing	24 18	大阪 横浜		

表 1: primitive user's view

入力属性値		⇒	出力属性値	
objects	属性		objects	属性
R 積荷票	事前 状態	⇒	R 積荷票	事後 状態
R 出庫依頼票			R 出庫依頼票	
R 出庫指示書			R 出庫指示書	
L 出庫指示書			L 出庫指示書	
L 不足リスト			L 不足リスト	
r 積荷			r 積荷	
L 積荷			L 積荷	
L 在庫リスト			L 在庫リスト	
E コンテナ			E コンテナ	
E 品物			E 品物	
E 顧客	E 顧客			

表 2: State transition on OAMs

倉庫システム

r積荷票: R積荷票
r出庫依頼票: R出庫依頼票
r出庫指示書: R出庫指示書
l出庫指示書: L出庫指示書
l不足リスト: L不足リスト
r積荷: R積荷
l積荷: L積荷
l在庫リスト: L在庫リスト

INIT ≡ l在庫リスト.INIT ∧ l不足リスト.INIT
入庫 ≡ r積荷票.起票 ∥ l在庫リスト.追加
注文処理 ≡ r出庫依頼票.起票 ∥
(l在庫リスト.出庫処理 ⇒ l出庫指示書.起票)
(¬ l在庫リスト.出庫処理 ⇒ l不足リスト.追加)

R積荷票

maz: N₁

eコンテナ: Eコンテナ
l積荷: L積荷
l積荷.搭載総数 ≤ maz

INIT ≡ eコンテナ.INIT ∧ l積荷.INIT

記入
Δ(eコンテナ, l積荷)

eコンテナ' = eコンテナ.記入
e積荷' = e積荷.記入

起票 ≡ INIT ∧ 記入

表 4: Object-Z での記述例

Data dictionary									
attr-spec	integer 0 ≤, ≤ 99999	日付連番							
attr-name (alias)	e コンテナ	b 年月日	e 品物	b 数量	e 顧客	r 出庫依頼票	b 空印	b 品名	b 住所

倉庫システム

objects	attributes									
R 積荷票	e コンテナ	(b 年月日)	l 積荷					(@b 空印)		
R 出庫依頼票			e 品物	b 数量	e 顧客	r 出庫依頼票*				
R 出庫指示書	e コンテナ		(e 品物)	b 数量	(e 顧客)	r 出庫依頼票	(@b 空印)			
L 出庫指示書			{r 出庫指示書} ^{1..N}							
L 不足リスト						{r 出庫依頼票} ^{0..N}				
R 積荷			e 品物	b 数量						
L 積荷			{r 積荷} ^{0..N}					@b 空印		
L 在庫リスト			{r 積荷票} ^{0..N}							

Entity dictionary									
E コンテナ	e コンテナ*	b 年月日							
E 品物			e 品物*					b 品名	
E 顧客					e 顧客*				b 住所

rItem, RItem stands for relation object
lItem, LItem stands for list or aggregation object
eItem, EItem stands for entity object
{Item} ^{0..N} stands for structure of list
eItem* is objectID in itself
@Item is calculated from other attributes
(Item) is derived from other objects

表 3: Object-Attribute Matrix