2007－SE－158 (1)
2007／11／2

アクタ関係行列を用いたｉスターフレームワーク作成方法の提案

井部 己文，山本 修一郎，佐藤 友合子
(株)NTTデータ 技術開発本部 システム科学研究所
ibek@nttdata.co.jp yamamotosui@nttdata.co.jp satouyr@nttdata.co.jp

要旨

ｉ＊フレームワークは，アクタ間の依存関係を表現する有効な手法として期待されている[1][2]。しかし，その有効性や限界については，まだ十分解明されてはいない。実際のビジネス構造をｉ＊フレームワークを用いてモデル化し，適用評価を行ったところ，(1) 各アクタ間の関係としてのソフトゴールの網羅性を確認することができないこと (2) アクタがおかれている利用シーンが表現できないといった課題があることがわかった[3]。本稿では，これらの課題を解決するため，アクタ関係行列を用いたゴール指向要求分析方法を提案する。

Proposal of i＊ framework development method based on actor relationship matrix

Komon Ibe , Shuichiro Yamamoto, Yuriko Sato
NTT Data Corporation Research Institute for System Science
ibek@nttdata.co.jp yamamotosui@nttdata.co.jp satouyr@nttdata.co.jp

Abstract

i* framework is expected as an effective technique for analyzing the dependence between actor [1][2]. However, neither the effectiveness nor the limits have been clarified enough. When an actual business structure is modeled by using i* framework, and the application evaluation was done, we found

(1) Covering a soft goal as the relations between each Actor should be not able to be confirmed. (2)The following problems, the use scenes of Actors are not expressible [3]. To solve these problems in this text, we propose the method of the goal oriented requirements analysis using the Actor relationship matrix.

１．はじめに

ｉ＊フレームワークは，アクタ間のゴール，ソフトゴール，資源，タスクの依存関係を表現する有効な手法として期待されている[1][2]。しかし，現実の多様な産業界の問題に適用する上での有効性や限界については，必ずしも十分に解明されていない。そこで実際の業界の問題を対象にしてｉ＊フレームワークの有効性評価を行い，次のような課題があることを明らかにした。[３]

(1)アクタの数が多い場合，アクタ間の関係が複雑で，網羅性がｉ＊フレームワークの図だけでは効率的に確認できない

(2)アクタの置かれたシーンによってアクタが抱える問題や，それに伴うゴール，ソフトゴール，資源，が，変化する可能性がある。すなわち，ある特定のシーンにおけるアクタと意図の関係だけを表現する，ｉ＊フレームワークではこれらの変化を記述できない。すなわち，シーン，問題，意図とアクタとの関係を総合的にモデル化して関係を分析する必要がある。この課題の背景には，対象とした業界構造の特徴とアクタが置かれたシーンが，アクタ自身の問題やゴールに影響を及ぼしているという点がある。

本稿では，アクタが置かれる問題状況ごとにアクタ間の関係を行列で定義する方法を提案する。次に，このアクタ間関係行列を用いて，網羅的にSD図を作成する方法を示す。

2.シーン別アクタ関係行列 ARM*の提案

2.1 概要

１章で述べた課題を解決するため，以下のような「段階的 ｉ＊フレームワーク開発手法」を提案する。

①シーン別アクタ行列　ASM の作成

シーン毎に登場するアクタを抽出する。

②アクタ関係行列　ARM の作成

アクタ間の内部ゴールとソフトゴール，資源，タスクの相互関係を行列として表現する。

③シーン別アクタ関係行列　ARM*の作成

複数のシーンに関する関係行列 ARM を統合する。

④SD 図の作成

ARM*に基づいて，SD 図を作成する。

2.2 シーン別アクタ行列　ASM

2.2.1　ASM の定義

【定義】ASM

シーン $S_1,\cdots S_m$ とアクタ $A_1,\cdots A_n$ に関する，n 行 m 列の行列 ASM の，i 行 j 列の要素は，シーン S_i に A_j が存在するとき「1」，そうでないとき「0」である。

アクタが置かれるシーンによって，登場するアクタが変化したり，同じアクタであっても問題点が変化しアクタ間の意図の依存関係も変化する。アクタの問題全体を扱ったり，アクタを固定しての全てのシーンについての分析を行うためには，シーン別にどのアクタが登場し，どのような意図の因果関係を持つかを分析することが必要である。

2.2.2　ASM の作成方法

表の行には，アクタが置かれるシーンに項番を振って記す。表の列には，アクタ名を書く。そして，シーンごとに登場するアクタが存在する時は「1」，そうでないときは「0」を記入する。シーンによっては，同一のアクタが登場することがあるが，その場合も互いが持つ意図が変化する場合があるので，別に考える必要がある。

この表の構成要素が，ASM の行列要素となる。

表 2.1　ASM

シーン＼アクタ	アクタA	アクタB	アクタC
S_1	1	0	1
S_2	1	1	0

←ASTの要素

Sx:アクタが置かれたシーンx

2.3　アクタ関係行列　ARM

2.3.1　ARM の定義

【定義】ARM $(A_1,\cdots,A_n)$

アクタ $A_1,\cdots,A_n$ に対する ARM $(A_1,\cdots,A_n)$ は n 行 n 列の行列である。ここで各行及び列は，アクタ $A_1,\cdots,A_n$ に対応する。第 i 行 j 列は，アクタ Ai が Aj に対して望む意図 Iij である。

また，第 i 行 i 列はアクタ Ai が持つ内部ゴール集合 G_{Ai} である。

$ARM_{ii} = \{G_{ik} \mid k=1,\cdots l_i\}$

$ARM_{ij} = \{I_{ijk} \mid k=1,\cdots l_{(i,j)}\}$

【定義】ARM 要素集合

ARM の要素集合は，S（ARM $(A_1,\cdots,A_n)$）で，ARM $(A_1,\cdots,A_n)$ の行列要素からなる集合を表す。

2.3.2　ARM の作成方法

ARM は，ゴールと意図としてのソフトゴール，資源，タスクを整理したものである。ARM (A_i, B_j) には，受益者 A_i から提供者 B_j への意図 I_{ij} を記入する。また，対角要素 ARM (A_i, A_i) には，アクタ A_i の内部ゴール G_i を記入する。

表 2.2　ARM

受益者＼提供者	アクタA	アクタB	アクタC
アクタA	G_{AB} , G_{AC}	I_{AB}	I_{AC}
アクタB	I_{BA}	G_{BA}, G_{BC}	I_{BC}
アクタC	I_{CA}	I_{CB}	G_{CA}, G_{CB}

←ARTの要素

G_{ij}:アクタiのjに対する内部ゴール，

I_{ij}:アクタiのjに対する意図

ここでは 3 次元 ARM を例にとっているが，n 次元に拡張することで，n 個のアクタ間の依存関係を表現することができる。

2.4　シーン別アクタ関係行列 ARM*

2.4.1　ARM*の定義

【定義】ARM*

アクタ $A_1,\cdots A_n$ とシーン $S_1,\cdots S_k$ に関する，n 行 m 列の行列 ARM*の要素は次で定義され

る。

ARM* (i,j) = {(Iij,Sk) |シーン Sk にアクタ Ai と Aj が存在,ASM(k,i)=ASM (k,j) =1) , (i≠j)}

ARM* (i,i) = {(Gij,Sk) |ASM(k,i)= ASM (k,j) =1) , (i≠j)}

2.4.2 ARM*の作成方法

ASM によって，シーン Sx 毎に登場するアクタを決定する。そのアクタ全体で ARM*を構成する。これを ARM*とする。

表 2.3 ARM*

提供者 / 受益者	アクタA	アクタB
アクタA	{(GAB, S1), (GAB, S2)}	{(IAB, S1), (IAB, S2)}
アクタB	{(IBA, S1), (IBA, S2)}	{(GBA, S1), (GBA, S2)}

←ART*の要素

(Gij, Sx) :シーンxに存在するアクタiのアクタjに対する内部ゴール
(Iij, Sx) :シーンxに存在するアクタiのアクタjに対する意図

ARM* (Ai, Bj) には，受益者 Ai から提供者 Bj への意図 Iij をシーン Sx ごとに記入する。また，対角要素 ARM* (Ai, Ai) には，アクタ Ai の内部ゴール Gi をシーン Sx ごとに記入する。

ASM と ARM*を組み合わせることによって，シーン S の変化に伴うアクタ A，意図 I の変化を表現することができる。これらの有用性は以下の通りである。

（1）あらゆるシーンにおけるアクタの意図の依存関係の網羅性が高まる。

（2）アクタを固定した場合の全てのシーンにおける意図の因果関係を分析することができる。

2.4.3 ARM*から i＊フレームワークの SD 図への変換

ARM*から i＊フレームワークの SD 図への変換規則は，以下の通りである（表 2.4)。

(規則 1)ARM*の ARM* (Ai , Bj) を，Ai から Bj のソフトゴールとする。

(規則 2)ARM*の ARM* (Bj, Ai) を，Bj から Ai のソフトゴールとする。

(規則 3)それぞれのソフトゴールを，アクタと矢印で結ぶ。

(規則 4)矢印の向きは，Ai→ Ia → Bj, Bj → Ib → Ai とする。

(規則 5)各ソフトゴールに，アクタが置かれる状況 Sx を記述する。

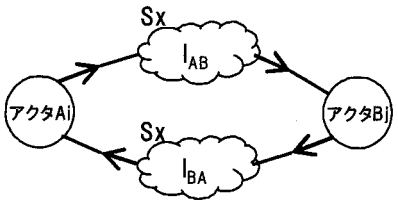

図 2.1 SD 図の例

3. シーンを考慮した i＊フレームワークの具体例

3.1 ASM の作成

ここでは，犬と飼い主の生活におけるシーンの例として，2 つのシーンを想定した。

S1：ペットフードをペットショップで買うシーン。

S2：ペットフードを動物病院で勧められて買うシーン。

これら 2 シーンに登場するアクタを，それぞれ ASM を用いて設定する。S1 で登場するアクタは，飼い主，犬，ペットショップ，ペットフード会社である。

S2 で登場するアクタは飼い主，動物病院，犬，ペットフード会社である。

このときの ASM は，表 3.1 のようになる。

表 3.1 ASM の例

	いぬ	飼い主	動物病院	ペットフード会社	ペットショップ
S1	1	1	0	1	1
S2	1	1	1	1	0

S1:ペットフードをペットショップで買う状況
S2:ペットフードを動物病院で勧められて買う状況

3.2 ARM*の作成

シーンごとに ARM を作成し，2 つの ARM を結合させて，ARM*を作成する。シーン S1 とシーン S2 は，「犬の健康を保つ」という飼い主の内部ゴールは同じであるが，それぞれのシーンによって，飼い主と犬，フード会社以外の登場するアクタが

異なり，アクタ間のソフトゴールも異なる。

表3.2 ARM*具体例

	いぬ	飼い主	動物病院	ペットフード会社	ペットショップ
いぬ					
飼い主	{安全安心なフードを食べさせたい,S_1} {安全安心なフードを食べさせたい,S_2}	{犬の健康を守る,S_1} {犬の健康を守る,S_2}	{信頼性のあるフード情報を教えて欲しい,S_1}	{安全なフードを製品化,S_1} {安全なフードを製品化,S_2}	{商品説明,S_2}
動物病院	{健康状態を見たい,S_1}		{認知度向上,S_1}	{宣伝をして欲しい,S_1}	
ペットフード会社		{嗜好を知りたい,S_1} {嗜好を知りたい,S_2}	{飼い主に斡旋して欲しい,S_1}	{売上げ,S_1} {認知度向上,S_2} {売上げ,S_1} {認知度向上,S_2}	{売れ筋を知りたい,S_2}
ペットショップ		{人気商品を知りたい,S_2}		{売れ筋製品化,S_2}	{売上げ向上,S_2}

シーンS1ではショップに対して，フードの安全性や商品情報を要求するが，シーンS2においては，獣医師に対して犬の健康状態を改善するためのフードの情報を，医学的見地から教えてもらいたいという要求をする。この例で言えば，S2の方がより犬の健康に拘った飼い主の思いが想像できる。

3.3 SDの作成

2章で述べたARM*からSD図への変換規則を用いると，シーン毎のSD図及び，シーンをマージしたSD図を記述することができる。例えば，ペットフードをペットショップで買う状況S1と，ペットフードを動物病院で買う状況S2をマージした結果のSD図を図3.1に示す。

ここで，2つのSD図をマージするとき，シーンS1，S2において，同一のアクタと同一の意図については，1つにまとめ，そのほかのアクタや意図については，それぞれ記述すると，2つのシーンにおける共通アクタや，共通意図をまとめることができる。また，各ノードにシーン識別子S1，S2を添付することで，2つのシーンを比較することができる。

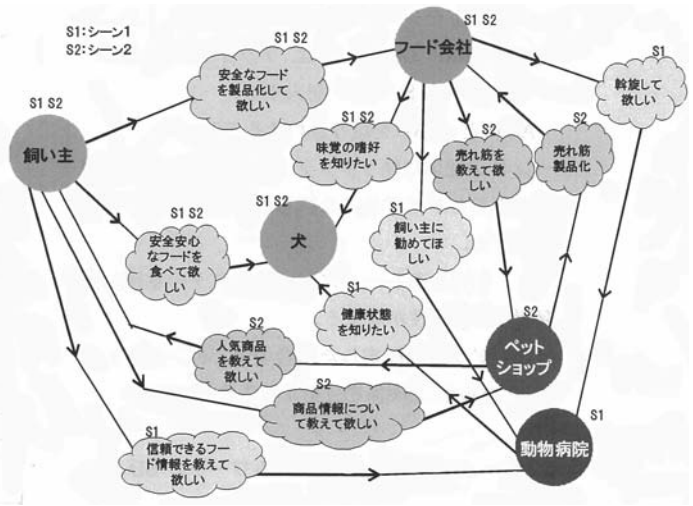

図3.1 2シーンをマージしたSD図の例

4. 考察

4.1 ARMの有用性

ARMには次の3つの有用性がある。

(1) アクタ間の関係の網羅性を確認できる
(2) アクタの関連性の類似性がわかる
(3) 登場しつつも，他のアクタとの関連性がないアクタを，行列の疎の部分として見極めることができる

4.2 SD図の一意性

複数のSD図がある場合，そのSD図が一意かどうか SD 図を見ただけでは容易に確認することはできない。しかしその SD 図に対応した ARM*を比較すれば，その一意性を容易に確認することができる。

アクタの集合：A，ソフトゴール集合：S，ゴール集合：G，タスク集合：T，資源集合：R
アクタ間の依存関係：D　とする。

D はアクタ間の関係を示す3項組み〈A_0, A_1, S ∪ G ∪ T ∪ R〉である。ここで，A_0 は受益者集合，A_1 は提供者集合，S ∪ G ∪ T ∪ R は，これらのアクタ間で必要となるソフトゴール，ゴール，タスク，資源の集合である。 換言すると，受益者が提供者に依存する対象の集合が，S ∪ G ∪ T ∪ R　である。

ここで，M を SD 図とし，

M＝〈A, S, G, T, R, D〉と表現する。

【定義】　SD 図の等価性

異なる SD 図 M_1，M_2

M_1＝〈$A_1, S_1, G_1, T_1, R_1, D_1$〉

M_2＝〈$A_2, S_2, G_2, T_2, R_2, D_2$〉

について，

A_1＝A_2，S_1＝S_2，G_1＝G_2，T_1＝T_2，R_1＝R_2，D_1＝D_2であるとき，M_1とM_2は等価であるといい，M_1＝M_2　で表す。

【定義】ARM*行列φ（M）

SD 図 M に関する ARM*行列φ（M）の i 行 j 列要素を次式で定義する。

行列　φ（M）(i,j) ＝ {Iij | M にアクタ Aj から Ai への意図 Iij がある}

以上の定義を用いることにより，SDM と ARM*の等価性を証明する。

2つの SDM　M_1，M_2が存在するとき，

φ（M_1），φ（M_2）を作ることができる。

φ（M_1）＝φ（M_2）ならば，任意の意図 I ∈ φ（M_1）＝φ（M_2）に対して，I の提供者である A_x と，受益者 A_y が存在する。もしも，あるアクタ A_x，A_y があって，M_1には存在するが M_2には存在しないとする。φの定義から，φ（M_1）(x,y)＝I_{xy}　となる行列要素が存在するはずである。ところが，I_{xy}は，φ（M_2）の要素でもあるので，このようなアクタが M_2に存在するはずである。これは矛盾である。従って，M_1とM_2のアクタ集合は，一致する。

次にある意図 I_{xy}が M_1には存在するが，M_2には存在しないとする。このときも同様にφの定義から矛盾が導かれる。

よって，M_1，M_2に対して，φ（M_1）＝φ（M_2）ならば，M_1＝M_2　であり，逆も成り立つ。

4.3 異なるSD図のマージにおけるARM*の有効性について

異なる SD 図を1つの SD 図としてマージする場合，アクタや意図の共通部分をくくりだすなど，SD 図のままマージするのは，困難であるが，それぞれの SD 図に対応した ARM*を用いることで，マージの手順化ができる。

M_1, M_2に対して，それぞれ，φ（M_1）とφ（M_2）を求める。これらの行列をマージするにあたり，以下のような方法を定義する。

このとき，φ（M_1）＋φ（M_2）をφ$_{12}$（M_1,M_2）で定義する。

φ$_{12}$（M_1,M_2）に対する SD 図 M_{12}が，M_1, M_2をマージした SD 図である。

αをφ（M_1）の添え字集合，βをφ（M_2）の添え字集合とする。

(1) m,n∈ α，m,n∈ βでないときは，
φ$_{12}$（M_1,M_2）(m,n) ＝φ（M_1）(m,n)

(2) m,n∈ αでなく m,n∈ βのときは，
φ$_{12}$（M_1,M_2）(m,n) ＝φ（M_2）(m,n)

(3) m,n∈ α,βで，且つφ（M_1）(m,n)＝φ（M_2）(m,n) のときは
φ$_{12}$（M_1,M_2）(m,n) ＝φ（M_1）(m,n)

(4) m,n∈ α,βで，且つφ（M_1）(m,n)≠φ（M_2）(m,n) のときは
φ$_{12}$（M_1,M_2）(m,n) ＝φ（M_1）(m,n)，φ（M_2）(m,n)

(5) m∈ α,mがβに含まれず，nがαに含まれず，n∈ βのとき及びmがαに含まれず,m∈ β，n∈ α，nがβに含まれないとき

ϕ_{12} (M_1,M_2) (m,n) ＝要素なし

4.4 シーンと内部ゴールの対応付け

アクタの持つ内部ゴールは，アクタがおかれたシーンによって変化するので，その対応付けが必要である。SD 図ではこれを表現することはできないが，SR 図におけるアクタの内部ゴールとシーンを対応付けて表現する方法を考える必要がある。

4.5 ARM マトリクスの疎密性と SD 図の関連

ARM マトリクスは全てのカラムが密な場合だけではない。密でない場合は疎の部分を省くことで，マトリクスを分割することができ，結果的に密なマトリクスを得ることができる。つまり，SD 図を分割することができる。また，マトリクスの疎の部分を見ることで，その状況に関係のないアクタを発見することができる。

4.6 ゴール分解方法について

従来の i＊フレームワークではゴールを分解する場合，そのゴール間の関係が必ずしも明確ではなく，分解の手順化がされていなかったため，ゴール自由度が高く，第三者がそのゴール分解を見たときに背景を理解しにくいという問題があった。本提案では，ゴールをシーンごとに分解する手順を示すことで，この問題の解決を図った。

5. おわりに

本稿では，アクタ関係行列を用いたシーン別の SD 図の作成手順について述べた。今後は，シーンの変化によるアクタの内部ゴールの変化によって，そのゴールを達成するためのタスクや資源が，外部の意図とどのような関係になるか SR 図を中心に分析を進める。また，ステレオタイプ[4] を用いたゴールと問題の対応関係に関する，i＊フレームワークの拡張についても研究を進める予定である。

6.参考文献

[1] i*homepage. http://www.cs.toronto.edu/km/istar/

[2] 山本修一郎　著　～ゴール指向による！！～システム要求管理技法，ソフトリサーチセンター，2007

[3] 井部己文, 山本修一郎：利用シーンを考慮した，i スターフレームワークの適用上の課題，第 157 回ソフトウェア工学研究発表会，2007

[4] Hans-Erik Eriksson, Magnus Penker 著 UML によるビジネスモデリング，ソフトバンク パブリッシング〈株〉，2002