3G1-2

問題解決プロセスモデルに基づくソフトウェア設計支援

Supporting Software Design based on Problem Solving Process Model

藤井 直樹[*1] 近藤 恵一[*1] 森田 武史[*1] 和泉 憲明[*2] 橋田 浩一[*2] 山口 高平[*1]
Naoki Fujii Keiichi Kondo Takeshi Morita Noriaki Izumi Kôiti Hasida Takahira Yamaguchi

[*1] 慶應義塾大学院 理工学研究科 [*2] 産業技術総合研究所
Graduate School of Science and Technology, Keio University National Institute of Advanced Industrial Science and Technology（AIST）

This reports proposes a support framework for specification descriptions of Web application development based on PSM ontologies. In order to reduce the dependency to an engineer's skill in a feature extraction from a requirement document, we provide a methodology both for finding lack of functional requirement and for extracting a development list item from the functional requirement. In this research, CommonKADS methodology is extended by refining and augmenting inference primitives. Through the implementation of our prototype system, we confirmed feasibility of PSM ontology enabled development of information systems.

1. はじめに

一般に，ソフトウェア開発では，要求仕様から開発項目を抽出する．ここでの精度は，ソフトウェア開発プロジェクトが成功するか否かに，大きな影響を与える重要なものである．

本研究では，抽出作業における要求定義者のスキルへの依存度の軽減を目的とし，発注者による機能要求に生じる情報の欠落の発見と，機能要求から半自動的に開発項目を抽出する手法を提案する．

ここでは，機能要求から開発項目を抽出する工程にPSM(Problem Solving Method)に基づく問題解決モデルの記述法を適用し，抽出方法の指針を与えるとともに，モデル化された開発項目（フィーチャーモデル）を抽出することを目指す．

2. Webアプリケーション開発の方法論

本研究では，発注者による機能要求の仕様としてユースケース記述を想定し，ユースケース記述を要求仕様として用いたWeb アプリケーション開発を対象とする．ユースケース記述とは，アクターとシステムのやりとりをストーリーとして書いたもので，アクターとユースケース名，事前条件，システムとアクターのやり取り（シナリオ）などで構成される．また，開発手法とは FDD（ユーザ機能駆動開発：Feature Driven Development）を想定する．

本手法では，ユーザからみた画面遷移を中心に Web アプリケーションをモデル化し，Web アプリケーション開発の開発方法論 FDD におけるフィーチャーを PSM の問題解決モデルにより構成する．

PSMの入出力オブジェクトの記述粒度を均質化するため，本手法では，ドメインオントロジーを援用する．ドメインオントロジーとは，対象領域(ドメイン) に存在するモノや人の関係を記述するオントロジーである．ドメインオントロジーは，is-a(上位下位) 関係と has-a(包含) 関係を用いた二つの階層（is-a 階層と has-a 階層）から構成され，どちらの階層も木構造である．今回，先行研究[近藤 08]によって作成されたドメインオントロジーを利用した．

3. PSMに基づくフィーチャーモデル

3.1 PSMプリミティブの定義

ここで，PSM は，CommonKADS により提供される問題解決プロセスにおける推論プリミティブのことであり，もともと，専門家の思考をモデリングしようとしたものである．PSM のアプローチでは，ドメインに現れる概念を全てオブジェクトとしてとらえ，オブジェクト操作によって，問題解決に導くことを目指している．

CommonKADS では，オブジェクト操作を表す PSM プリミティブとして 18 種類の推論プリミティブと，4 種類の通信プリミティブを提供している．本研究では，提供されている PSM プリミティブからフィーチャー抽出に適用できるものと，適用できないものに分類し，さらに，必要なプリミティブを追加した(表 1).

表1の中で，システム設計の段階では不必要なものとは，システム設計の以前，もしくは，以後に行われる処理を表しているものである．定義が抽象的過ぎるものとは，CommonKADS による定義が，本研究で想定したモデルの定義より抽象度が高かったものである．システムに行えないものとは，この推論プリミティブが人間にしか行えない複雑な思考プロセスを表してものである．

また，CommonKADS の提供している推論プリミティブには，DB のような蓄積の概念が無い．Web アプリケーションの設計には，DB の概念が不可欠であるため，search と update を追加した．さらに，present の表すオブジェクト操作を具体化し，分類する必要があるため，サブプリミティブとして，transmit，mail，print を追加した．

表 1 CommonKADS で提供されている推論プリミティブおよび通信プリミティブの分類，追加した PSM プリミティブ

適用した PSM プリミティブ		
推論プリミティブ	compare generate select sort	sort modify group
通信プリミティブ	obtain present	receive provide
適用しなかった推論プリミティブ		
システム設計では不必要なもの	propose abstract	predict verify
定義が抽象的過ぎるため使えないもの	operationalize match	specify assign
システムには実行できないもの	evaluate critique	cover
追加した PSM プリミティブ		
transmit print mail search update		

3.2 オブジェクトの定義

本研究においては，ユースケース記述に現れる概念は，情報システム上の具体的なデータやシステムとして実装されるため，データオブジェクトと参照オブジェクトとしてとらえる．ここでは，それぞれのオブジェクトをモデル化するための記述法を与える．

(1) データオブジェクト

データオブジェクトは，PSM プリミティブへの入出力となる．

モデルには，そのオブジェクトのオブジェクト型名と，そのオブジェクトが属性として持つオブジェクト型名を記述する．ドメインオントロジーを参照し，属性オブジェクトはもれなく記述することとする．また，オブジェクト型と属性オブジェクト型は，ドメインオントロジーにおいて，has-a 関係にあるものである(図 1).

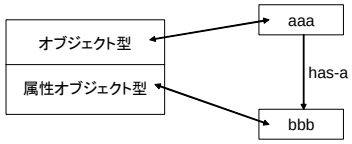

図 1　データオブジェクトの表記とドメインオントロジー

(2) 参照オブジェクト

参照オブジェクトは，DB とシステムを表すため，DB 名とシステム名のみを表記することとする．すなわち，本モデルでは，DB とシステムの内部構造はシステム開発者に委譲し，詳細に関しては定義しないこととする(図 2).

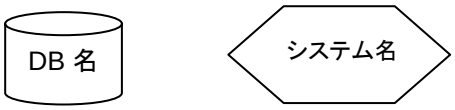

図 2　参照オブジェクトの表記の仕方

3.3 PSM に基づくフィーチャープリミティブ

本研究では，PSM プリミティブ，入出力オブジェクト，参照オブジェクトを 1 つの単位として定義し，これを，フィーチャープリミティブと呼ぶこととする．フィーチャープリミティブは，フィーチャーを構成する最小単位となる（図 3）.

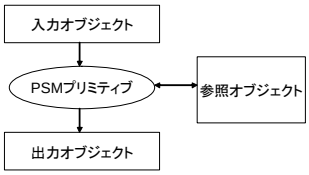

図 3　フィーチャープリミティブの表記の仕方

3.4 PSM に基づくフィーチャーモデルテンプレート

Web アプリケーションでは，画面上でユーザが処理要求をし，画面遷移がおき，次の画面に処理結果が出力される．つまり，Web アプリケーションは，1画面遷移を機能単位としている．したがって，1 画面遷移間にシステムが行う処理を機能単位（フィーチャー）と定義することができる．本研究では，1 画面遷移間にシステムが行う処理をフィーチャープリミティブでモデル化したものをフィーチャーモデルと定義する．

ここで，予備資料を分析した結果，10 種類のフィーチャーモデルが得られた(図 4). これらをフィーチャーモデルテンプレートとした．このうち，1 画面遷移間に複数のフィーチャープリミティブから構成されるフィーチャーモデルが 7 種類，単一のプリミティブのみで構成されているフィーチャーモデルが 3 種類あった．この 7 種類については，表 4 で具体的な業務を例に挙げ説明する．その他の 3 種類のフィーチャーモデルについては，表 2，3 のフィーチャープリミティブ定義のとおりである．

表 2　フィーチャープリミティブの入力・出力・参照オブジェクト定義

フィーチャープリミティブ	入力 obj	出力 obj	参照 obj
推論プリミティブ			
search	obj（検索条件）	obj 集合（検索結果）	DB
update	登録する obj	無	DB
select	obj 集合	選択された obj	無
modify	主 obj／従 obj	更新済の主 obj	無
generate	複数の obj	生成した obj	無
compare	2つの obj	真偽値	無
group	obj 集合	複数の obj 集合	無
sort	obj 集合	ソート済 obj 集合	無
classify	obj 集合	分類済 obj 集合	無
通信プリミティブ			
provide	受け渡す obj	無	無
present	受け渡す obj	無	無
print	印刷する obj	無	無
transmit	転送する obj	無	システム
mail	送信する obj	無	無
obtain	無	受け取る obj	無
receive	無	受け取る obj	無

表 3　フィーチャープリミティブの定義

フィーチャープリミティブ	定義
推論プリミティブ	
search	検索条件から，DB 検索し，結果を出力
update	obj を DB に登録
select	obj 集合の中から，いくつか選び，出力
modify	主 obj の属性に，従 obj を代入し，出力
generate	入力 obj を属性として持つ obj を出力
compare	2つの obj を比較し，結果を出力
group	obj 集合から，同属性を持つ obj 集合を出力
sort	入力 obj 集合を並び替え，出力
classify	入力 obj 集合を分類し，出力
通信プリミティブ	
provide	外部の要求に応じて obj を送信
present	内部主導により，obj を送信
print	obj を印刷
transmit	obj を他システムへ送信
mail	obj を e-mail で送信
obtain	外部に要求し obj を受信
receive	外部主導で送られてきた obj を 受信

4. PSM に基づくフィーチャー抽出方法

ここでは，ユースケース記述からフィーチャーモデルを抽出するプロセスと，ユースケース記述の情報の欠落を発見するプロセスを述べる．

(1) フィーチャーモデルの抽出

ユースケース記述のシナリオから，サブシナリオ毎にフィーチャーモデルを抽出し，フィーチャーリストを作成する．

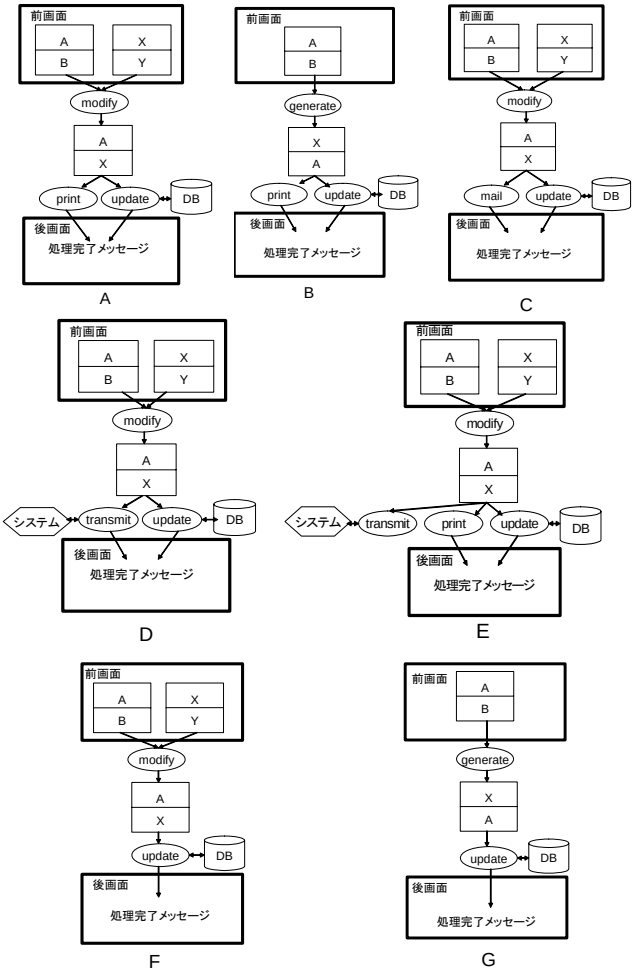

図 4　フィーチャーモデルテンプレート一覧

(2)　フィーチャー連結，フィーチャー欠落の発見

フィーチャーリスト内あるフィーチャーモデルを連結する．フィーチャーモデルを連結したものをフィーチャーフローと呼ぶ．また，フィーチャーの欠落を発見する．

(3)　孤児オブジェクトの発見

ここでは，生成プロセスなしに出現するオブジェクト(孤児オブジェクト)を同定する．

(4)　迷子オブジェクトの発見プロセス

ここでは，生成したものの，格納プロセスなしのオブジェクト(迷子オブジェクト)を同定する．

5.　ケーススタディ

5.1 実験方法

本研究では，産業技術総合研究所の人事給与システム，及び，財務会計システムのユースケース記述の主シナリオに記述されている各画面遷移間に，システムの行っている処理をフィーチャーモデルで表した(表 5)．

5.2 実験結果

PSM モデルに基づくフィーチャーリスト抽出の結果を，表 6 と表 7, 表 8, 表 9 に与える．

表 4　フィーチャーモデルテンプレートの定義と適用例

A.書類情報の修正(modify,print,update)
定義:オブジェクト A の属性 B に，入力されたオブジェクト X を代入し，更新したオブジェクト A を印刷，DB に登録．例:前画面で既存の書類情報へ修正情報を入力する．既存の書類情報を修正し，修正された書類情報を DB 登録し，印刷する．
B.書類情報の新規作成(generate,print,update)
定義:入力されたオブジェクト A を属性としてもつ，オブジェクト X を生成し，印刷，DB に登録する．例:前画面で，新規作成する書類の必要情報を入力する．書類情報を新規作成し，DB に登録し，印刷する．
C.情報修正，メール送信(modify,mail,update)
定義:前画面でオブジェクト A の属性にオブジェクト X を代入する．更新されたオブジェクト A を DB 登録し，メール送信．例:前画面で既存情報へ修正情報を入力する．既存情報を修正し，修正済情報を DB に登録し，担当者にメールを送信．
D.情報修正，他システムへ転送(modify,transmit,update)
定義:オブジェクト A の属性へオブジェクト X を代入し，更新されたオブジェクト A を，DB に登録し，他システムに送信する．例:前画面で既存情報へ修正情報を入力する．既存情報を修正し，修正済情報を DB に登録し，他システムに転送．
E.情報修正，メール送信，システムへ転送 (modify,transmit,mail,update)
定義:オブジェクト A の属性にオブジェクト X を代入する．更新されたオブジェクト A を DB に登録し，他システムへ送信し，メール送信．例:前画面で既存情報へ修正情報を入力する．既存情報を修正し，修正済情報を DB に登録し，他システムへ送信し，修正済情報を担当者にメールを送信
F.情報修正(modify,update)
定義:オブジェクト A の属性に，オブジェクト X を代入する．更新済オブジェクト A を DB 登録．例:前画面で既存情報へ修正情報を入力し．既存情報を修正し，修正済情報を DB に登録
G.情報の新規作成(generate,update)
定義:入力されたオブジェクト A を属性としてもつ，オブジェクト X を生成し，DB に登録．例:前画面で，新規作成に必要な情報を入力する．情報を新規作成し，DB に登録する

表 5　抽出対象ユースケース記述数

	対象ユースケース記述数
人事・給与システム	12
財務・会計システム	10

表 6　抽出されたフィーチャーモデル数（フィーチャーモデルテンプレート対応したもの）

フィーチャーモデル	人事給与	財務会計
obtain	9	8
select	7	5
update	2	2
generate	0	2
search	1	0
generate+update	2	0
modify+update+transmit+mail	1	0
modify+update	3	5
modify+update+mail	1	0
modify+update+print	0	1
modify+update+transmit	1	0

表 7　抽出されたフィーチャーモデル抽出数（フィーチャーモデルテンプレート対応しなかったもの）

フィーチャーモデル	人事給与	財務会計
print	1	0
update+print	0	1

表 8　発見されたユースケース記述内の情報欠落

	迷子	孤児	連結不可フィーチャーモデル
人事･給与	3	4	3
財務･会計	0	0	0

表 9　ドメインオントロジーの参照によって，補完されたオブジェクト数

	補完されたオブジェクト数
人事･給与	16
財務･会計	14

5.3 考察

(1)　ユースケース記述内の情報欠落の発見

表 8 より，孤児と迷子オブジェクト，連結不可フィーチャーモデルが発見できた．各画面遷移間にシステムが行う処理をフィーチャーモデルで表すことによって，オブジェクトのシステム内での振舞いが明確になり，その結果，迷子と孤児オブジェクトの発見が容易になった．孤児，迷子オブジェクトが発見された場合，それぞれを出力，入力オブジェクトとする処理が欠落していると考えられる．ユースケース記述に記述されている全てのシステムの処理を，フィーチャーのフローとして捉え，画面遷移の流れに沿って，フィーチャーモデルを連結し，フィーチャーフローを作成することによって，フローから孤立しているフィーチャーモデルを明らかにすることができた．連結不可フィーチャーモデルについては，そのフィーチャーモデルが不必要な処理であるか，もしくは，そのフィーチャーモデルをフィーチャーフローに連結するための処理が欠落していると考えられる．表 9 が示すように，ドメインオントロジーを参照することによって，ユースケース記述で欠損しているオブジェクトを発見，補完することができた．補完されたオブジェクトは，あるオブジェクトの属性オブジェクトとなるものが多かった．これは，ユースケース記述には，オブジェクトの属性は，抽象的に書かれていることが多い（例えば，職員_必要情報）ためである．ドメインオントロジーを参照することによって，オブジェクトの属性を明らかにすることができる． オブジェクトの属性を明らかにすることによって，システムが行う処理への入出力を正確にすることができた．

(2)　フィーチャー抽出におけるガイドライン

ユースケース記述は，自然言語により記述されているため，その記述は形式化されていない．したがって，システムの同じ処理を表す文でも，記述者によって異なる表現が使われることがある．したがって，ユースケース記述から何の枠組みもなしに，フィーチャー抽出作業を行うと，フィーチャーが発散してしまう可能性が大きい．しかし，本研究で提案したフィーチャーモデルテンプレート，及び，フィーチャープリミティブをフィーチャー抽出のガイドラインとして利用することにより，フィーチャーの発散を防ぐことができると考えられる．図 5, 6, 7 に例を示す．

連絡先：藤井直樹，山口高平，慶應義塾大学理工学部，
〒223-0061 神奈川県横浜市港北区日吉 3－14－1，
Tel：045-566-1614，E-mail:{n_fujii,yamaguti}@ae.keio.ac.jp

	3.→主アクターは、画面から居住者・非居住者判定票の登録内容を確認し、確定登録を要求する。システムは、居住者・非居住者判定票を"確定済"として登録する。

図 5ユースケース記述一部抜粋

	4.→主アクターは、画面から確定登録を要求する。システムは、扶養控除等異動申告を"確定済"として登録する

図 6ユースケース記述一部抜粋

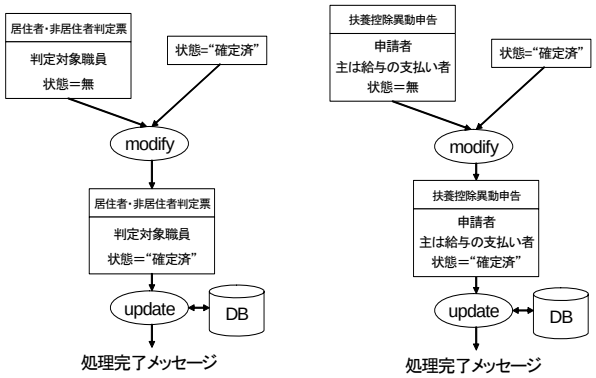

図 7　図 5,図 6 のユースケース記述より抽出したフィーチャーモデ

(3)　フィーチャーの粒度判定

画面遷移間に行われている処理をフィーチャーモデルで表すことによって，各フィーチャーモデルに存在するフィーチャープリミティブの数によって，その画面遷移間での処理の粒度を判定できる．表 7 より，フィーチャーモデルテンプレートに当てはまらない，フィーチャーモデルが抽出された．このフィーチャーモデルが対応する画面遷移では，他の画面遷移と異なる粒度の処理が行われていることが明らかとなった．

6. おわりに

本研究では，PSM の問題解決モデルに基づいてソフトウェアの設計項目を半自動的に抽出する手法について述べた．その手法を適用した結果，システム内での，オブジェクトの振る舞いや，フィーチャー同士のつながりが明確になり，フィーチャーやオブジェクトに関する情報欠落の発見が容易になった．また，フィーチャーモデルテンプレートや，フィーチャープリミティブを，フィーチャー抽出作業におけるガイドラインとして与えたことにより，ユースケース記述の曖昧性，ならびに，情報の欠損から生じるフィーチャーの不均質さの問題や，フィーチャー自身の記述方法の曖昧性により誤った解釈がなされるという問題を解決することが可能となった．

参考文献

[Schreiber 99] Guus Schreiber, A. th Schreiber :Knowledge Engineering and Management, The MIT Press ,1999.

[Gardner 01] Karen M. Gardner, Alexander Rush, Bobbin Teegarden, Michael K. Crist, Robert Konizer ：認知パターン―オブジェクト技術のための問題解決フレームワーク，Pearson Education Japan , 2001.

[近藤 08] 近藤 恵一，森田 武史，和泉 憲明 山口 高平，橋田 浩一 :情報システム開発支援のためのエンタープライズアプリケーションオントロジー，KBSE 研究会 , 2008.