(7) 悪性疾患の場合、告知、説明の状況
(8) 入院の場合、病室などの本人の希望
(9) 現在の処方内容

退院時要約の記載項目

(1) 患者プロフィール
　氏名、年令、性別、その他嗜好、職業歴、住環境、薬物服用歴、アレルギー（薬物によるものも含む）

(2) 診断名と転帰
　入院中の問題点、確定診断名、必要に応じて暫定診断名、症状、データ異常

(3) 入院病歴
　主訴（あるいは入院目的）、現病歴、既往歴、家族歴、理学所見、入院時検査所見、入院後経過

(4) 考察
　診断根拠、治療方針決定の根拠、症例の独自の事項（社会上、診断上、治療上等）、外来主治医へのメッセージ

これらの記載項目（あるいはさらに詳細な記載項目）が、受け側の情報システムで誤りなく解釈されるためには、これらに標準的なコードが割り当てられている必要がある。

このような目的で用いられるコードとしては、国際的には HL7 で想定されている LOINC、国内では、MEDIS-DC の標準データセット（J-MIX）がある。

モデルベース診療文書の実現にあっては、記載項目の表現方法として、これらのコードの使用法を定める必要がある。

5．2 LOINC

LOINC とは、Logical Observation Identifier Names and Codes の略で、Regenstrief Institute が開発・管理する検査、診察等で得られる情報の項目名とそのコードの規格である。

LOINC の用語には、診療用語（ Clinical Term Classes ）、検査用語（ Laboratory Term Classes ）、診療（請求）明細（ Attachment Term Classes ）の３種類があり、このうち、診療用語が、ここでの記載項目に相当する。CDA の標準仕様では、必須ではないが、記載項目のコード化にこの LOINC を想定しており、Release Two でも踏襲される見込みである。

なお、検査用語については、後述の MEDIS-DC の標準臨床検査マスターとの関連を検討する必要がある。また、診療（請求）明細については、文書コードの規定となっており CDA にも関連する。したがって、モデルベース診療文書としての診療情報提供書、退院時要約の詳細検討の際には、その文書コードの決定の際に充分検討する必要がある。

5．3 標準データセット

平成１１年４月の当時の厚生省健康政策局長、医薬安全局長、保険局長の連名による通知「診療録等の電子媒体による保存について」により、診療録等の電子媒体による保存が認められることになったことを受け、MEDIS-DC が平成１１年度の厚生省の委託事業である「高度情報社会医療情報システム構築推進事業」の一環として、診療録等の電子保存を実施している医療機関相互間において患者の診療録情報を電子的に交換する場合に必要となる標準的なデータ項目セットについて検討した結果を報告書として取りまとめたものが標準データセットである。
電子カルテに記録されるべきと考えられているデータ項目を机上で収集するのではなく、さまざまな情報交換のユースケースで実際に使用されているさまざまな書面や様式に収載されているデータ項目を優先的に収集する方針で作成されている。
また、情報交換の場面では、個々のデータ項目を指定するために、日本語標準ラベルまたは英語標準ラベルを用いることとし、それに対応するデータ項目コードを用いることをしない方針で作成されている。

日本の診療事情に合わせて作成されたものであるので、モデルベース診療文書の記載項目のモデルとして適当であるが、国際的な標準化を見据えて LOINC との対応をとり、HL7 の XML 記法にあわせた情報交換用ラベルの生成規則を検討する必要がある。

6　標準マスターや国際規格に基づく各種マスター

6．1 キーワードの表現

ここでいう、キーワードとは、あらかじめ規定された語彙の中から表現される用語で、これにより用途別のシステム間での情報のインターフェースや参照や統計処理の際の検索キーとして用いられる情報である。

実際に記載される用語に対してコードを付与した用語コードがメインであるが、ある集合概念に対してコード化する分類コードも用いられる。実際には用語コードのマスターにリンク情報として記録されている分類コードを用いた処理が行われる。

なお、これらのキーワードが、診療支援、患者安全のために有効に利用されるには、副作用情報や、診療ガイドラインといった公開のコンテンツに含まれるキーワードも、これらのキーワードおよびコードとマッチしている、あるいは少なくとも変換テーブル、対照表が用意されている必要がある。
しかしながら、これらのコンテンツは現在電子的に取得可能な形で公開されている場合でも、このようなキーワードの対応ができるようなコード化された形で公開されている例は非常に少ない。今後は、このようなコンテンツの提供にあたっては標準的なコードでキーワードをコード化することを推進する必要がある。

6．2 標準病名マスター

正式名称を「ICD10 対応電子カルテ用標準病名マスター」と呼び、1999 年 4 月の第 1 版が公開されたが、2001 年には標準化を一歩進めるための、いわゆるリードタームを明確化した階層構造をもつマスターとして再構成され第 2 版となった。さらに 2002 年 6 月からは、同時公開された診療報酬請求用の傷病名マスター第 2 版とのコードの相互持合による統合化が実現した。

ICD10 を分類コードとして採用しているが、収録されている病名自体は臨床上で使われている病名であり、第 2 版からは、各々病名を表すコード自体は ICD10 と無関係となっており、用語コードと分類コードの違いが明確になっている。
４桁の情報交換用コードにて共通する疾病概念をあらわすほか、補助テーブルとしてシソーラスを提供するなど、病名という範囲に限定されるが、概念志向の用語コードに近い取り組みがされている。

６．３ 標準手術・処置マスター

レセプト電算処理システム用手術・処置マスター（以下レセ電算名称）、日本医学会および同分科会の用語集（以下学会用語）、既存の手術・処置マスター（以下一般名称）を基本ファイルとし、収載名称として、基本ファイルの中でも「レセ電算名称」および「学会用語」を中心に、補完するファイルとして「一般名称」を用いて作成された。
基本分類コードには ICD・9・CM の手術・処置分類を採用している。

６．４ 標準医薬品マスター

現在国内で使用されている医薬品コードの現状は、医薬品の承認、市販後調査、副作用報告、流通、薬価、レセプト処理などの目的別に１０種類を越えるコード体系が利用されているのに加えて、各医療機関で使用されている医薬品コードの多くが独自コードである。
このうち、流通、薬価、レセプト処理など、各医療機関で利用頻度が高い思われる４つのコード体系（薬価基準収載医薬品コード、個別医薬品コード（YJ コード）、JAN コード、レセコード）の対応テーブルとそれを管理するための永久不変・欠番の基本コード（ＨＯＴコード）を設定したもの。
モデルベース診療文書の観点から見た課題としては、一般名と商品名の関連が分かるような情報や、周辺コンテンツとして添付文書の計算機可読化があげられる。

６．５ 標準臨床検査マスター

日本で行われている臨床検査項目を JLAC10 のコード（１５桁）で表現し、さらにそれぞれのレコードに診療行為マスターの検査の診療行為コードを対応づけたもの。
このマスターに含まれる検査の領域は、検体検査（血液学的検査、細菌検査など）、病理学的検査および生体検査（呼吸循環機能検査、超音波検査など）である。JLAC10 は原則として検体検査（病理学的検査を含む）を表現するコード体系であるが、このマスターでは JLAC10 を拡張し、生体検査もサポートしている。

JLAC10 とは、日本臨床検査医学会が制定した「臨床検査項目分類コード(第 10 回改訂)」のことであり、(1)分析物コード、(2)識別コード、(3)材料コード、(4)測定法コード、(5)結果識別コードの 5 つの要素区分よりなる。

将来、国際的なデータ交換において、HL7 と密接な関係にある LOINC とのマッピング作業も求められることになるかもしれない。

６．６ 標準医療材料データベース

日本医療機器関係団体協議会の「医療材料商品コード・バーコード標準化ガイドライン」に基づき、多種・多様な医療材料について、一元的に商品コードを管理する目的で、

MEDIS-DC が、厚生労働省の委託を受け、各企業が自社製品情報を登録してインターネット環境上で広く関係機関がその情報を利用する「医療材料データベース」を開発し、運用している。

マスターではなくデータベースとして提供されるため、モデルベース診療文書でどのように活用するのか、システム運用の面を含めて十分検討する必要がある。

6．7 看護用語マスター

現在検討作業中であるが、モデルベース診療文書の観点では、家族歴、生活歴などについて医と看護の両面からの検討が必要と思われる。

7　概念志向に基づく総括的な用語コード体系

7．1 任意の概念の表現

従来の診療文書の電子化では、今回キーワードとして定義した、あらかじめ規定された語彙の中から表現される用語レベルがコード化されるに過ぎなかったが、これでは必要に応じて任意の詳細度まで計算機で解釈できる形で表現できることにはならない。もちろんまったく事前の規定なしではそのようなことはできないのであるが、概念志向のコード化を行うことで、既存の概念を組み合わせて新たな概念を表現することができる。

このような用語コードとしては、医療全般を対象とした SNOMED-CT が代表的であるが、残念ながら日本語化されていない。日本語化されたものとしては、対象分野は限定されるものの、医薬品に関連した情報（効能や副作用など）についての国際的に共通な用語集 MedDRA の日本語版 MedDRA/J が存在する。また MEDIS-DC の標準病名マスターは病名という限定された範囲であるが、4 桁の情報交換用コードにて共通する疾病概念をあらわすほか、補助テーブルとしてシソーラスを提供するなど、概念志向の用語コードに近い取り組みがされている。

7．2 SNOMED-CT

日本医療情報学会の課題研究会、用語モデル研究会では、研究課題として次のようなテーマを掲げている。

「現在、ISO,CEN といった国際的な標準化機構で、医療用語モデルの標準化が行われている。中でも CEN の Categorical Structure は、SNOMED-CT といった大規模用語集で採用されている概念志向モデル、セマンティックリンクといった技術を集約した規格であり、現在、この規格に基づいた領域別用語集モデル（術式、看護用語など）が ISO,CEN で提案されている。日本では、用語開発において概念志向モデルが採用されることが少なく、このような国際規格も十分周知されているとはいえない。しかし、これらの規格は、各領域で用語開発を行う際の共通基盤となる技術である。本研究会は、このような用語モデルに関する技術の普及を図り、日本における用語集モデルの採用・開発の可能性について検討する。」

米国 HL7 は、e-Government の推進の一環として医療 IT の発展のために SNOMED-CT をベースに電子カルテのモデルを構築することを決定し、HL7 にその電子カルテのモデル作り

を委託している。また英国では自国政府で作成した用語集 Clinical Terms を SNOMED-CT に統合することで、やはり HL7 Version 3 ベースの電子カルテシステムを国策として推進する上で、概念志向の用語モデルを取り入れている。また第1回国際 CDA 会議の舞台となったドイツについては、SNOMED-CT にドイツ語版がすでに用意されている。

日本でも、本格的な電子カルテの推進にあたっては、記載内容を精密かつ詳細にコード化できる何らかの概念志向の用語モデルを取り入れる必要がある。SNOMED-CT の日本語版を作成するのかどうか、しないのであれば、どのようなモデルを作成するのか早急に方針を決める必要がある。

8　今後の検討課題

今回提唱するモデルベース診療文書の構想を、実例を元に検証し、より具体的で詳細な要件としてまとめる必要がある。これについては、JAHIS 診療支援システム委員会電子カルテコンテンツ検討 WG の平成 16 年度のテーマとして取り組む予定である。なお、モデルベース診療文書に密接に関わる用語コードや、診療文書交換標準などについては、JAHIS 単体で取り組むことは出来ないため、関連する標準化に関わる組織に対して、標準制定活動への参加や要望や意見の提言活動にも取り組む必要がある。

9　まとめ

電子カルテの効能としてとりあげられることの多い、情報の共有化や後利用について、その実効ある実現には一定の技術的要件が必要なことを明らかにし、モデルベース診療文書の構想としてまとめた。

今後、問題点が解決され、このようなモデルベース診療文書を含めた電子カルテシステムの標準化が、真の診療支援を実現するための基盤となることを希望する。

付録・用語コード一覧

用語・概念のカテゴリ（SNOMED-CT の分類による）	用語コード	概念コード	分類コード
Clinical Finding（症状、病名）	標準病名マスター	SNOMED-CT MedDRA/J	ICD10 ICD-O ICD9 ICD9-CM Diagnosis) ICPC
Procedure/intervention（手術・処置）	標準手術処置マスター	SNOMED-CT	ICD9-CM Procedure CPT
Observable entity（検査、観察項目）	標準臨床検査マスター LOINC	SNOMED-CT	
Body structure（部位）	(*1)	SNOMED-CT	
Organism（生物、有機体）	(*2)	SNOMED-CT	
Substance（物質）	(*2)	SNOMED-CT	
Pharmaceutical/biologic product（薬品）	標準医薬品マスター (*2)	SNOMED-CT	
Specimen（検体）	(*2)	SNOMED-CT	
Physical object（物品）	標準医療材料データベース	SNOMED-CT	
Physical force（物理力）		SNOMED-CT	
Events（事象）		SNOMED-CT	
Environments/geographical locations（環境、場所）		SNOMED-CT	
Social context（社会的）		SNOMED-CT	
Context-dependent categories（文脈依存）		SNOMED-CT	
Staging and scales（評価スケール）		SNOMED-CT	
Attribute（属性）		SNOMED-CT	
Qualifier value（修飾語）		SNOMED-CT	
Special Concept（特別の概念）		SNOMED-CT	

(*1)　標準病名マスターの修飾語に含まれる
(*2)　標準臨床検査マスターや LOINC を構成する部分コードに含まれる

平成 15 年度厚生労働科学研究
標準的電子カルテシステムのアーキテクチャ（フレームワーク）に関する研究
総括研究報告書

(資料５)

電子カルテシステムのコンポーネントモデルの試作

――――目次――――

１．はじめに 3
１．１　経緯 3
１．２　開発内容 3
２．コンポーネントモデルの注射オーダドメインでの例 5
２．１　注射オーダ生成 6
２．２　注射オーダ更新 9
２．３　注射オーダ削除 12
２．４　注射オーダ検索 15
２．５　注射オーダ承認 18
２．６　注射オーダ承認取り消し 21
２．７　注射オーダ指示受け 24
２．８　注射オーダ指示受け解除 27
２．９　注射オーダ指示受け返却 30
２．１０　注射オーダ保留 33
２．１１　注射オーダ保留解除 36
２．１２　注射取り揃え実績生成 39
２．１３　注射取り揃え実績削除 42
２．１４　注射混合実績生成 45
２．１５　注射混合実績削除 48
２．１６　注射計画生成 51
２．１７　注射計画更新 54
２．１８　注射計画削除 57
２．１９　注射計画検索 60
２．２０　注射実績生成 63

２．２１　注射実績削除……66
２．２２　注射実績検索……69
２．２３　注射開始実績生成……72
２．２４　注射終了実績生成……75
２．２５　注射速度変更実績生成……78
２．２６　注射途中確認実績生成……81
２．２７　注射中止実績生成……84

１．はじめに

「電子カルテシステムのコンポーネントモデルの試作」は、以下の経緯のもとに実施した。なお、本モデル開発おいて、現在筑波大学附属病院で稼働中のオーダエントリシステムである CAFE の技術情報を参考にした。

１.１　経緯

保健医療福祉情報システム工業会（JAHIS）では、医療機関における情報システムの品質を高めるための研究を継続して行っている。電子カルテシステムの研究としては、平成 14 年度の厚生労働科学特別研究事業の「コンポーネントの標準化による電子カルテ開発」があり、本モデル開発はこの成果に基づいて行った。

１.２　開発内容

本モデル開発では、分散処理システム構築における標準フレームワークである RM-ODP(Reference Model for Open Distributed Processing)とモデルの表記法である UML(Unified Modeling Language)を使用し、電子カルテシステムのサーバサイドに注目したコンポーネントモデルを開発した。

サーバサイドのコンポーネントは、大きくファサードと内部コンポーネントに分けられる。ファサードは、クライアントサイドからの処理要求に対する窓口であり、トランザクション単位に一連の処理がまとめられている。各コンポーネントは３７の領域に分類され、全体で４１７個のファサードが導き出された。

作成されたモデルは、一つのファサードに対し、モジュール構成図、サブプロセス、インタフェース情報一覧表の３種類によって構成される。

- モジュール構成図

  ファサードと、ファサードから呼び出される内部コンポーネント、及び外部システムのコンポーネントの構成を表す。登場するコンポーネントはアーキテクチャに対応する以下のステレオタイプによって分類される。

  - 《Facade》クライアントサイドからの処理要求を受け付け、一連の処理を行う。
  - 《Business Logic》計算、判断などの最も基本的な処理を行う。
  - 《Data Access》永続情報（データベースなど）に対して情報の入出力を行う。

  また、上記アーキテクチャのステレオタイプに加え、UML Profile for EDOC においてのステレオタイプである《Process Component》（機能的コンポーネント）、《Entity》（情報コンポーネント）、《Protocol Port》（コンポーネントのインタフェース）も付加した。

- サブプロセス

  モジュール構成図にて明らかにされた内部コンポーネントが、ファサードから呼び出される順序を記述した。UML Profile for EDOC においての以下のステレオタイプを付加し、ファサードからの内部コンポーネントの呼び出しを明確に示した。

  - 《responds》クライアントサイドからの呼び出しを表す。
  - 《initiates》内部コンポーネントの呼び出しを表す。

- インタフェース情報定義一覧表

  ファサードがクライアントサイドから利用される際に行う入出力の情報を一覧にした。

２　コンポーネントモデルの注射オーダドメインでの例

今回作成したコンポーネントモデルのうち、「注射オーダ」ドメイン（２７ファサード）のモデルを以下に示す。

資料５に含まれる図やテーブルは報告書（配付版）には掲載することができませんでしたので、下記の報告書（電子版）を参照いただけますようお願い申し上げます。

保健医療福祉情報システム工業会
電子カルテシステムモデル特別プロジェクトのページ
http://www.jahis.jp/StandardEPRS/index.htm

なお、こちらの報告書（電子版）は、本研究班の報告書に加え、「電子カルテ導入における標準的な業務フローモデルに関する研究」（代表研究者　飯田修平　全日本病院協会理事）報告書の一部を含め、JAHIS「電子カルテシステムモデル特別プロジェクト」の報告書という形式となっており、より包括的な報告書となっています。

平成15年度厚生労働科学研究
標準的電子カルテシステムのアーキテクチャ（フレームワーク）に関する研究
総括研究報告書

(資料６)

MDAによる開発事例とその技術的基盤

――――目次――――

１．はじめに 3
２．実行モデル 4
　２．１．実行モデルとは 4
　２．２．基本プロセス 5
３．分析 5
　３．１．目的 5
　３．２．ユースケース分析 5
　３．３．ユースケースのチェック 6
　３．４．ユースケースアクティビティ 7
　３．５．ユースケースアクティビティのチェック 8
　３．６．ドメイン分析 9
　３．７．ユースケースの利点 9
　３．８．カタログ作成 10
　　３．８．１．業務項目の抽出 10
　　３．８．２．業務ロジックの抽出 10
　３．９．フィージビリティー計画 11
　３．１０．テスト計画 11
　３．１１．その他の成果物 12
４．モデル設計 12
　４．１．ビジネスパターン 12
　４．２．モデル図 12
　４．３．コンポーネントアクティビティ 12
５．実装 12
　５．１．実装インターフェイス 12

５．２．手続きによる業務ロジックの実装 14
６．その他 14
７．実行モデルの利点 17
８．CAFE 17
８．１．CAFEとは 17
８．２．特徴 17
８．３．モデル図 18
８．４．開発規模 18
８．５．利点 20
９．最後に 21
参考文献 21

１．はじめに

多くの企業・団体が自らの業務を効率化または省力化しようと、毎年膨大な数のシステム開発プロジェクトが遂行され稼働を開始し、稼働後もさらなる業務効率化を求めて修正され続けている。しかしこれだけ多くのシステムが稼働しているにも関わらず、納期・品質・顧客満足度の 3 点をすべて満たすことの出来たシステムは数少ないと言われているのが現状である。

従来システム開発を依頼した顧客のプロジェクトで重視されていた点は完成したシステムの品質であった。品質さえ一定水準以上であれば、納期が少々遅れようとあまり問題とされないのが一般的であった。しかし近年の厳しい経済状況はそのような甘えを許さなくなってきている。厳しい経済状況は顧客の情報化投資予算への圧力となり、システム構築予算の縮小傾向に拍車をかけていることは自明である。これは開発者に対しプロジェクト予算内でシステムを開発させるために納期の厳守という新たな要求を突きつけ、品質でさえあまり保てていなかった開発現場をさらに混乱に追い込んだ。そして見逃せない点は顧客がシステムを単なる業務データの記録媒体としての扱いから顧客自身のビジネスプロセスやビジネスルールを直接扱う戦略ツールとして位置づけ始めたことである。システム上に記録された膨大なデータから未来の事業戦略はどうあるべきかを考察したり、新商品の開発に利用したりと、データの再利用を行う事例が急速に増加した。競争に勝ち抜くためにも絶えず変化が求められる現状にあるため、顧客は開発者に対してシステム上の業務ルールのタイムリーな変更を要求するようになった。追加要求がなくても品質・納期を守ることすら出来ない開発現場にとって頻繁に起こる細かな業務ルール・項目変更は、開発現場を混乱の渦に巻き込んだ。結果上記の 3 点を達成出来ないプロジェクトの増加に拍車がかかり、動かないコンピュータの見本が増産されているのが現状である。

本稿ではシステム開発の品質面を打開するため考案した開発プロセス・プログラミングフレームワークである実行モデルを紹介する。この実行モデルは仕様策定から実装に至るまで、フェーズ毎に目標、成果物、チェック項目を明確に定義したプロセスを通じ開発することで品質の高いシステムを構築し、開発途中で必ず発生するであろう仕様変更に対しても柔軟に対応出来るよう考慮された開発プロセスである。さらに分析・設計結果で作成された成果物からプログラムコードを自動生成する仕組みを備えており、分析結果と実装との間で違いが発生しにくいプロセス・構造になっている。まずこの実行モデルのプロセス、成果物、チェック項目などを解説し、どのような効果が期待できるかを説明する。そしてこの実行モデルを具体的に検証するため、実際にこの実行モデルを用いて開発された、現在筑波大学附属病院で稼働中のオーダエントリシステム CAFE を紹介し、その開発に関す

る分析を行う。

２．実行モデル

２．１．実行モデルとは

従来のシステム開発では、要件をあいまいにしたまま開発を始めたために、成果物に対する検証が不能になることが多発した。加えて現状の検証すら出来ないため、このまま開発を進めて良いのかの判断すら不可能になることが頻繁に発生した。そのような場合、業務に詳しいエキスパートを開発チームに投入し難局を乗り切ろうとするが、結局問題解決する仕事がそのエキスパートに集中するため、エキスパート自身が開発のボトルネックとかし、開発効率を下げてしまう現象が起きた。

次に実装面を見てみると、業務ロジックの変更や業務データ項目の追加に対して非常に工数がかかるなど、拡張性に関する問題点は相変わらずであった。これは従来の業務ロジックがデータストア（リレーショナルデータベースなど）から業務オブジェクト（データ）をどのように取得・保存するかを記述する部分と、取得した業務データへの処理を記述した部分が分散していたことによるものであり、言い換えればデータストア上のテーブル構造と密接に関係していたためである。

開発上の諸問題を解決しシステムの拡張性を確保する試みとして、オブジェクト指向開発の導入が叫ばれた。結果としてオブジェクト指向開発ではデータストアから得た業務オブジェクトを隠蔽（カプセル化）し、業務オブジェクトを生成するロジック等の実装面では有効に利用された。しかしオブジェクトの再利用については成果を上げることが出来ないのが現実であった。これは業務ロジックならびに業務オブジェクトを記述する一貫した手段が提供されていなかったからである。

以下で紹介する実行モデルは、最新のオブジェクト指向開発法を取り込み、さらに一貫した業務ロジックならびに業務オブジェクトの記述法を提供することで、業務ロジックの変更や業務データ項目の追加に対して柔軟性のなるシステムが構築出来るような手段を開発者に提供する、プロセスならびに開発用フレームワークである。そして実行で開発したシステムが長期間利用可能であり、柔軟性のあるシステムとなることが最終的な目的としている。

２．２．基本プロセス

実行モデルでは成果物中心主義で開発を進め、各開発フェーズで成果物に対しルールとチェックリストによる検証を必ず行う。実行モデルでは開発プロセスとして以下のフェーズが定義されている。

a. 分析
システムが必要とする機能を抽出し、ユーザ、開発者双方で検証可能かつ、変更影響範囲が特定可能な形式で記述する。
b. モデル設計
ビジネスパターンに基づく振る舞いベースのモデル設計を行う。
c. 実装
上記の分析・設計結果を直接利用し実装する。その際業務ロジックの実装が分散しないような仕組みを提供する。

そして各フェーズで作成される成果物は必ずフェーズ関連者（ユーザ、開発者など）のいずれからも理解・検証が可能であり、他フェーズでの成果物との対応関係がトレース可能でなければならないことを求めている。成果物がこれらの条件を満たすよう、一部はUML(Unified Modeling Language)を使用している。

また変更が発生した場合は、速やかに変更が解決可能なフェーズまで戻り、変更点を修正あるいは追加し開発を再開する。これを繰り返し行い顧客の要望に迅速に対応し、求められているシステムを確実に開発していく。

３．分析

３．１．目的

このフェーズでは業務を達成するためにどのような機能が必要かを抽出することが最初の目的である。言い換えればシステムのユーザが何をしようとしているのかを導き出すことが当面の目標である。そして導出された機能一覧からその後の開発に対してどのような方針で開発を進めていくかを決定する。

３．２．ユースケース分析

ユースケースは、ユーザ（関係者、関係システム）から見たシステムが果たすべき外部機能を表現したものである。典型的な業務フロー（シナリオ）を想定し、そのシナリオに関わる人（アクタ）を見つけ記述する。

通常アクタごとに業務を行う上で必要な機能を見つけ、ユースケースを作成していく。記述される項目は以下のものである。

a. アクタ　　シナリオ関係する人もしくはもの
b. 機能　　業務を達成するために必要な機能
c. 前提条件　シナリオを開始する前に達成しておかなければならない条件
d. 終了条件　シナリオが終了する時に達成しているはずの条件

３．３．ユースケースのチェック

業務を達成するのに必要なユースケース一覧が作成出来たら、以下の点で全ユースケースに対してチェックをかける。

a. ユースケースの妥当性の検証

ユースケースをアクタを主語にした一文にしてみて意味が通じるかを検証する。アクタが必ず存在しているか、また同一の業務レベルで記述されているか、また途中でアクタ、時間、場所が変更されず単独で利用することが可能か中心に検証する。

ID	アクター名	参加者	メモ
	患者情報登録者	Learn可能ユーザー	
	患者オーダ編集者	上級医 初級医 学生	

ID	ユースケース名	アクター	Activity	メモ
FFAC-01	患者基本情報を表示する	患者情報参照者	Activity	患者の基本情報を表示する
FFAC-02	担当患者一覧を表示する	患者オーダ編集者	Activity	アクターが担当している患者の一覧を表示する
FFAC-03	病棟患者一覧を表示する	患者情報参照者	Activity	病棟に属する患者の一覧を表示する
FFAC-04	休まれ患者一覧を表示する	患者情報参照者	Activity	休まれの患者リスト一覧が表示される
FFAC-05	患者詳細情報を表示する	患者情報参照者	Activity	より詳細な患者の情報を表示する
FFAC-06	オーダ発行対象患者を選択する	患者オーダ編集者	Activity	オーダを発行する患者を選択する
FFAC-07	オーダ発行対象患者が解除される	患者オーダ編集者	Activity	オーダが発行された後に選択されていた患者が元に戻る
FFAC-08	院長指導画面を表示する	患者情報参照者	Activity	院長系の初期画面を表示する
FFAC-09	病棟指導画面を表示する	患者情報参照者	Activity	病棟系の初期画面を表示する
FFAC-10	口答モード画面を表示する	患者情報参照者	Activity	口答モード画面を表示する
FFAC-11	リマインダーを表示する	患者情報参照者	Activity	月別、当日からのスケジュールリマインダー（実施日、予定日）が表示される
FFAC-12	オーダー覧を表示する	患者情報参照者	Activity	当日行われる予定の有効な指示、データの一覧を表示する
FFAC-13	病名一覧を表示する	患者情報参照者	Activity	患者の病名リスト一覧表示する
FFAC-14	病名を編集する	患者オーダ編集者	Activity	病名の編集（追加）を行う
FFAC-15	薬剤を検索する	患者情報参照者	Activity	マスタの薬剤の検索を行う
FFAC-16	実施を表示する	患者情報参照者	Activity	オーダの実施を表示する
FFAC-17	実施を入力する	患者情報参照者	Activity	オーダの実施を入力する
FFAC-18	実施を修正する	患者情報参照者	Activity	オーダの実施を修正する
FFAC-19	レポートを検索する	患者オーダ編集者	Activity	レポートの検索を行う
FFAC-20	レポートを表示する	患者オーダ編集者	Activity	レポートの表示を行う
FFAC-21	オーダを作成する	患者オーダ編集者	Activity	オーダの作成を行う
FFAC-22	オーダを検索する	患者オーダ編集者	Activity	オーダの検索を行う
FFAC-23	オーダを表示する	患者オーダ編集者	Activity	オーダを表示する
FFAC-24	オーダをコピーする	患者オーダ編集者	Activity	オーダをコピーする

図 2.1. ユースケース一覧

b. ユースケースの混在可能性の検証

異なる複数アクタを持つユースケースが存在する場合、そのユースケースはアクタごとに分割出来ないか、またあるアクタが片方のアクタの代理として振る舞っていないか検証する。さらに前提条件が存在する場合、その前提条件が他のユースケースで実現されているか検証する。

c. 運用可能性の検証

最後にすべての記述されてユースケースで業務の運用が可能か検証し、可能と判断されればユースケースの抽出は終了する。

３．４．ユースケースアクティビティ

ユースケースアクティビティはユースケース毎にその機能を実現するのに必要な業務手順（アクティビティ）に細分化し、実際に行うべき作業を明確化したものである。
アクティビティは入力項目ないし出力項目を基準に分割し記述する。その際システムの内部的な動作は記述しない。これはこの段階ではユースケースのうちどこまでをシステム化

するかまだ決定していないためである。

またプロセス内にアクタによる排他的な選択肢が存在すればそれも記述する。

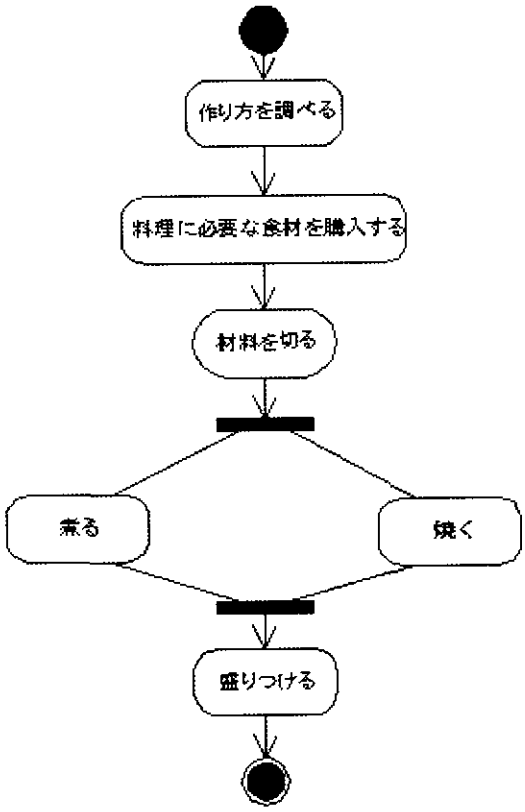

図3.1.ユースケースアクティビティ図

３．５．ユースケースアクティビティのチェック

a. アクタから見たアクティビティか検証

アクタ中心にアクティビティが記述出来ているかチェックする。その場合システム的なプロセス表現が含まれていたらこれを排除する必要がある。

b. アクティビティ内の項目の検証

次にアクティビティ内の項目がアクタから見て、一度に決定する項目のみで記述されているかを検討する。一度に決定できないようであれば、アクティビティの分割あるいはユースケースレベルの分割を考える。

c. 曖昧な表現の排除

最後にアクティビティ内の記述で形容詞を使った曖昧な表現が使われているならば、そこで使用されている形容詞は必ず名詞を使った具体的な表現に変換し、すべて業務項目を抽出する。