Vol.2013-SE-181 No.16
2013/7/18

テスト対象と実行条件を考慮したテストタイプに基づくシステムテスト計画書導出手順の開発

大橋恭子[†1]　麻岡正洋[†1]　笹川文義[†1]　若杉賢治[†2]　難波功[†1]

コンピュータシステム開発の上流工程で作成する文書からシステムテストの計画書を高精度に作成する方法を提案する．
システムテストではアプリケーションの機能や非機能求に対する検証に加え，システム基盤や利用者への展開手順等の検証など多様な側面からの検証が必要となる．しかし，システムテストの計画書や仕様書を作成するための手順は標準化されておらず，それら文書の品質は属人性が高いという課題がある．
この課題解決のためテストタイプの定義と上流工程文書からシステムテスト計画書を導出する手順を開発した．テストタイプ定義にあたり，テストタイプの構造を反映したテストモデルを作成した．このテストモデルの要素を要件定義工程や基本設計書と追跡可能な程度にまで詳細化し，さらにいくつかの要素を組み合わせてテストタイプを定義した．さらにシステムテスト計画書導出手順の精度を実プロジェクトの文書を用いて評価し，ST のスキルを持たなくても高精度のテスト計画を作成することを確認した．

Development of a test plan document generation method which is based on test type

KYOKO OHASHI[†1]　MASAHIRO ASAOKA[†1]　FUMIYOSHI SASAGAWA[†1]
KENJI WAKASUGI[†2]　ISAO NAMBA[†1]

We propose the system-test (ST) plan generation method with high precision from the document which is made by the early phase of computer-systems development. In addition to the verification of the application functions, in ST, the verification from the various sides, such as verification of a system infrastructure, the deployment process to a user, etc., is required. However, the procedure for drawing up the plan and specifications of ST is not standardized. And, there is a problem that the quality of test plan document and test specification is uneven.
We developed a test type definition and a method which derives a test plan document from an early phase's document. For the test type definition, we made a test model which is a reflection of test case's structure. We broke down model's elements that can trace to early phase documents. And each test type was defined by combination of some instances of a test model. We have verified that normal skill person can generate a highly precise test plan document by our proposing generation method.

1. はじめに

現在，コンピュータシステムは企業活動の業務支援だけでなく，広く社会全般で利用されている．そのため，稼働後に障害が発生すると社会的な影響は大きい[1]．また，企業活動の業務改善を目的に開発されたコンピュータシステムであっても，使い手を無視した仕様で役に立たないといわれる場合も多い[2].

このような状況を防ぐ“最後の砦”の一つが，システムテスト(ST)である．システムテストは国際ソフトウェアテスト資格認定委員会(ISTQB)において，テストレベルの一つとして位置付けられている[3]．各テストレベルでは，機能テストや性能テスト等いくつかのテストタイプを実施する．ISTQB のソフトウェアテスト標準用語集では，テストタイプという用語を次のように定義している[4].

> コンポーネント又はシステムをテストするためのテスト活動をまとめたものであり、たとえば機能テスト、使用性テスト、回帰テストなどのように特定のテスト目的に焦点を当てている。テストタイプは一つ又は複数のテストレベル又はテストフェーズで行われる。

また，文献[3]には各テストレベルで何を確認すべきかが記述されている．しかし，どのテストタイプに重きを置くべきかが記述されていない．また，同じテストタイプを複数のテストレベルで実施する場合がある．テストレベルが異なれば同じテストタイプであってもその内容には違いがあるはずだが，その違いが記述されていない．このようにシステムテスト計画書やシステムテスト仕様を作成するために必要な情報が不足している．そのためシステムテストの品質は，システムテスト計画作成者やシステムテスト仕様書作成者のスキルに依存する．

システムテスト計画書やシステムテスト仕様書は，どのような文書を参照して，作成されているのだろうか．ソフトウェア開発の V 字モデル[5]では, 開発工程とテスト工程の対応関係が示されている．このモデルで ST に対応するのは基本設計工程である．しかし， ISTQB のソフトウェアテスト標準用語集において，システムテスト(system testing)は以下のように定義されている[4].

†1 (株)富士通研究所 Fujitsu Laboratories Ltd.
†2 富士通(株) Fujitsu Ltd.

統合されたシステムが、指定された要件を満たすことを実証するためのテストの手順。

この定義に記述されているように, ST には基本設計工程だけでなく要件定義工程の情報が必要である．つまり，システムテスト計画やシステムテスト仕様の作成スキルには，単にテストの知識だけでなく，顧客業務に関する知識や顧客要求の詳細や要求の優先順位の把握といった要素も含まれる．そのため ST に関する高いスキルは容易には習得できず，結果として高品質のシステムテスト計画書やシステムテスト仕様書を作成可能な者が限定されてしまうという課題が生じている．以降，要件定義工程と基本設計工程で作成する文書を，上流工程文書と呼ぶ．

本論文では，第２章で課題解決のための基本的なアプローチを説明する．この章の中でテストモデルとテストタイプ定義と我々が導出目標とするテスト計画書について説明する．第３章では，漏れがなくかつ無駄のないテスト計画を作成するための工夫について述べる．第４章では，システムテスト計画書導出手順について述べる．社内の実プロジェクトで作成した文書を利用して，システムテスト計画書導出手順(以後，単に導出手順と呼ぶ場合がある)の有効性を評価した．第５章ではその評価について述べる．第６章では関連研究について述べ，第７章で本論文の結論と今後の課題を示す．

2. アプローチ

我々は１章に示した課題を解決するための第一歩として，ST で実施すべきテストタイプを定義し，その定義に基づいたシステムテスト計画書導出手順を作成するアプローチを取った．テストタイプの定義に先だってテストモデルを作成し，そのモデルを利用することで，ST で実施すべきテストタイプの厳密な定義を目指した．

この章では，テストモデルとテストモデルを利用したテストタイプの定義と我々が導出目標とするシステムテスト計画書について述べる．

2.1 目標

まず，我々のアプローチの目標を示す．

- ST で実施すべきことは漏れなくテストできること
- 無駄なテストは行わないこと
- 高スキルを持ったテスト技術者でなくても高品質のシステムテスト計画書やシステムテスト仕様書が作成可能であること

一つ目の目標を達成するためにテストモデルを定義する．テストモデルでは，システムテスト計画書，システムテスト仕様書の雛形を表すための要素を定義する．二つ目の目標を達成するためにテストタイプを定義する．テストモデルについては 2.2 節，テストタイプについては 2. 3 節に詳細に示す．以後，システムテスト計画書やシステムテスト仕様書を，テスト計画書やテスト仕様書と省略することがある．

これら二つの目標を達成するためには，テストタイプごとに上流工程文書からテスト計画書やテスト仕様書を導出するための手順を整備すればよいと考えた．しかし， 実プロジェクトの最終的な成果物となるテスト計画書やテスト仕様書には，自然言語による文章でテストでの実施内容が記述されていた．自然言語で記述された文章を導出する手順を定義することは現実的ではない．そこで，テスト計画書やテスト仕様書の雛形を導出の目標とした．ここでの雛形とは，単なる文書の書式ではなく，ST を漏れなくかつ無駄なく実施するための“ キーワード ”を含んだものである．2.4 節に我々が導出の目標とするシステムテスト計画書を示す．

なお三つ目の目標を達成するために，2.4 節に示したシステムテスト計画書の雛形を導出する手順を定義する．この導出手順については４章に詳細に示す．

2.2 テストモデル

テスト計画書やテスト設計書を漏れなく作成するためは，まずテスト計画作成作業やテスト設計作業で考慮すべきことを明確にする必要がある．そこで，これらの作業時に考慮すべき要素と要素間の関係を表すテストモデルを作成した．UML のクラス図で表したテストモデルを図 1に示す．

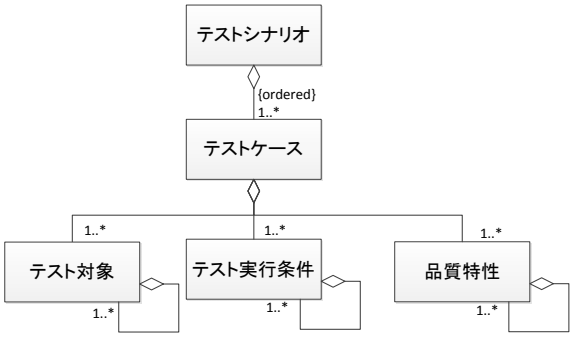

図 1：テストモデル

このモデル中の要素を定義する．

- テストケース：テスト実行の最小単位．
- テストシナリオ：テストケースの集合．テストケースをある順序で並べたもの．
- テスト対象：テストによって実行されるもの．例えば，業務，業務プロセス，機能といったコンピュータシステムの構成要素であり，テストの実行によってその正しさや妥当性が検証されるもの．
- テスト実行条件：テストを実行する上で，事前に満たしていなければならない条件．例えば，データ量やテストケースを実行する日時等である．
- 品質特性：テストケースの実行によって，確認される性質．

1-1	業務機能テスト

説明

業務フローに従った一連の業務タスクの組み合わせが要件や仕様通り動作することを確認する。

○品質特性(どうなっていることを)

機能性/合目的性	・業務遂行に支障がないこと
機能性/正確性	・業務の流れにしたがってデータが連携すること(業務フローどおりに動作すること)

○テスト対象(何が)

業務・業務プロセス	・代表業務、定常

○テスト実行条件(どういった条件のもとで)

(where)	・[実行環境] − [本番環境(新システム)]or[疑似本番環境(テスト環境)]
(who)	・[テスト実行者] − [ベンダ] or [顧客(情シス・開発)] or [顧客(情シス・運用)] ・[利用者に関わる条件] − [利用者の役割] − [エンドユーザ] or [業務運用](業務の種類によっては実施する)
(what)	・[入出力に関わる条件] − [データのバリエーション] − [業務フローの全パスを通るようにデータを揃える]

図 2：テストタイプ定義の例

図 1 のクラス図にも示したように，テスト対象，テスト実行条件，品質特性の各クラスのインスタンスは階層的に詳細化できる．この詳細化は，上流工程文書と相互に追跡可能な程度まで実施する．

以降では，これらのクラスの中で，テストタイプを定義する上で特に重要なテスト対象，テスト実行条件，品質特性の各クラスについて着目する．

2.3 テストモデルを利用したテストタイプの定義

前節のテストモデルに示すように，テストケースの雛形は，テスト対象とテスト実行条件，品質属性のインスタンスを組み合わせて作成する．しかし，単純に組み合せただけでは意味のない無駄なテストケースができてしまう．そこで，要素の適切な組み合わせを定義することで無駄なテストケースの生成を防止する．この適切な要素の組み合わせがテストタイプの定義である．図 2 にテストタイプ定義の例を示す. この例に示すように, テストタイプ定義では, テスト対象とテスト実行条件，品質属性のインスタンスの適切な組み合わせを示している．

各クラスのインスタンスの作成方法やテストタイプ定義の作成方法は 3 章で詳細に述べる．

2.4 導出目標とするテスト計画書

一般にテスト計画書には，テストの工程定義，スケジュール，推進体制，作業分担，テスト用のシステム環境，テスト項目等が記述される．このテスト項目とは，類似したテストケースを集めたカテゴリである．各カテゴリをさらに詳細化したものがテスト仕様書に記述されるテストケースやテストシナリオになる．

大分類		中分類	
ID		ID	
1	システム起動／停止	1	以下順番で○○システムを起動し、正常稼働できることを確認する。 1.MDB連携サーバ起動→2.検索サーバ起動→3.業務APサーバ起動→4.ポータルサーバ起動→5.業務アプリ起動→6.業務規制解除→7.業務確認
1		2	以下順番で○○システムを停止し、正常停止できることを確認する。 1.業務規制実施→2.業務アプリ停止→3.ポータルサーバ停止→4.業務APサーバ停止→5.検索サーバ停止→6.MDB連携サーバ停止→7.ソーリーコンテンツ表示確認
2	状態監視	1	ログ監視として××に対し、○○システムからログトラップ通知が正常に通知されていることと、証跡管理として△△のログ保管サービスを利用し、対象ログが正常にNASストレージ上に保管されていることを確認する。
2		2	リソース監視として○○システムのCPU／メモリ／ディスク使用量において閾値超過時に警告／異常通知を××に対し正常に通知することを確認する。また正常値に復帰した際の通知メッセージに関しても××に対して通知されることを確認する。
2		3	プロセス監視として○○システムで登録申請しているプロセスを停止し、警告／異常通知を××に対し正常に通知することを確認する。また停止しプロセスを復旧し、正常な状態に復帰した際の通知メッセージを確認する。
2		4	ジョブネット監視として○○システムが登録しているジョブが正常終了/警告終了/異常終了した際、イベント通知としてジョブの状態を××に対し正常に通知することを確認する。
2		5	IJServerクラスタ監視として業務アプリの起動／停止を実施し、正常に状態監視できることを確認し、××に対し正常に通知することを確認する。

図 3：システムテスト計画書(テスト項目)の例

テスト計画書のテスト項目でカテゴリを漏れなく無駄なく作成することが，テストケースの漏れと無駄防止につながる．そこで我々は，テスト計画書の中のテスト項目を上流工程文書から導出することを目標とした．

複数の実プロジェクトで作成したシステムテスト計画書のテスト項目を調査したところ次の 2 つの特徴があった．一つは，カテゴリは 2，3 程度の階層で構成されていること．もう一つは，各カテゴリでどのようなテストを実施するのかが文章で記載されていることである．図 3 は，ある実プロジェクトで作成したテスト項目である. この図に示すように，カテゴリが“大分類”，“中分類”の 2 階層で構成されている．また，中

分類には各カテゴリにおいてどのようなテストを行うのかを表す文章が記述されている．この文章の中には，中分類の各カテゴリを特徴づける単語が含まれていた．我々はそれを“キーワード”と呼ぶ．このキーワードは，テストモデルのテスト対象もしくはテスト実行条件に該当するものである．品質特性は，同じテストタイプ内であれば共通しているので，カテゴリを特徴づけるキーワードにはならない．

中分類に記述されている文章を上流工程文書から導出することは現実的ではない．そこで，導出するテスト項目に記述するのはキーワードのみとした．キーワードが漏れなく無駄なく導出されていればテスト漏れや無駄は防止できる．さらに導出されたテスト項目に対して，テスト設計者がテストタイプ定義を参照すれば，中分類に記述されている文章も記述可能だと判断した．図 4 に我々が導出の目標とするテスト項目の例を示す．これは，以下の２つの特徴を持つ．

- テスト項目は大分類と中分類の２階層で構成していること
- 大分類，中分類には各々を分別するのに必要なキーワードが設定されていること

大分類		中分類	
ID		ID	
1	システム起動／停止	1	起動
1		2	停止
2	状態監視	1	ログ監視
2		2	リソース監視
2		3	プロセス監視
2		4	ジョブネット
2		5	IJServerクラスタ監視

図 4：導出目標のシステムテスト計画書(テスト項目)の例

3. テストタイプの定義

本章では，2.3 節に示したテストタイプ定義の作成方法を詳細に述べる．以後，3.1 節で ST において実施すべきテストタイプの抽出方法について述べ，3.2 節でテストモデルの中のテスト対象，テスト実行条件，品質特性のインスタンスを作成する手順を述べる．その後， 3.1 節で抽出したテストタイプと 3.2 節で得られたインスタンスを用いてテストタイプを定義する方法を 3.3 節に示す．

3.1 テストタイプの抽出

本節では ST で実施すべきテストタイプの抽出手順を示す．抽出は，以下の２ステップで実施した．

- テストに関する文献調査し，テストタイプの名称とそのテストタイプに属するテストケースの特徴を収集する．
- 収集したテストタイプの中から，テストケースの特徴に基づいて ST で実施すべきものを選択する．

3.2 品質特性，テスト対象，テスト実行条件の詳細化

2 章でテストモデルを示した．このモデル中の要素は概念的なものであった．上流工程文書からテスト計画書を導出するには，品質特性，テスト対象，テスト実行条件の各インスタンスが上流工程文書と追跡可能になる程度に詳細化しなければならない．そこで，実プロジェクトのシステムテスト計画書やシステムテスト仕様書を利用してテストモデルのインスタンスを作成した．

以降の節では，品質特性，テスト対象，テスト実行条件の各クラスのインスタンスを作成する手順を述べる．

3.2.1 品質特性

ソフトウェアに求められる品質は，日本工業規格(JIS)の JIS X0129-1 で既に規定されている[6]．この規格では，ソフトウェアの品質を６個の品質特性とそれを詳細化した品質副特性で表している．

そこで，以下のステップでテストモデルの品質特性クラスのインスタンスを階層的に作成する．

1. JIS X 0129-1 で規定している６個の品質特性を最上位のインスタンスとし，その品質副特性をその下位階層のインスタンスとする．
2. 実プロジェクトのシステムテスト計画書やシステムテスト仕様書から品質特性クラスの具体例に該当するものを抽出する．抽出したものを抽象化してインスタンスとする．抽象化した例を次に示す．
   例１：業務遂行に支障がないこと
   例２：エラーメッセージが仕様どおりであること
3. ステップ 2 で抽出したインスタンスを，いずれか一つの品質副特性の下位階層に位置づける．

上記のステップ 2 で実プロジェクトのシステムテスト計画書やシステムテスト仕様書から抽出した具体例を抽象化した理由を示す．ステップ 2 では，多数のプロジェクトについて調査を実施した．それらの中には同じ性質を確認しているにも関わらず異なる表現になっているものがあった．また，開発対象のシステムに固有な機能名や画面名を使用している場合もあった．そのため，実プロジェクトから抽出しただけでは，他プロジェクトでは使用しにくかった．そこで，複数の表現を一つにまとめたり，他プロジェクトでも適用可能にするための抽象化を実施した．

3.2.2 テスト対象

要件定義工程や基本設計工程では，顧客業務の分析，システム化範囲の決定，システム化する機能(システム機能と呼ぶ)の抽出，システム機能の仕様定義等が行われる．要件定義工程での分析結果の概念モデルである要求モデル[7]が過去に提案されている．

以下のステップでテスト対象クラスのインスタンスを階層的に作成する．

1. 要求モデルで規定している要素の中からシステムテストにおいて実行単位となるものを選択し，それを

最上位の階層インスタンスとする．

2. 実プロジェクトのシステムテスト計画書やシステムテスト仕様書からテスト対象クラスの具体例に該当するものを抽出する．抽出したものを抽象化してインスタンスとする．抽象化した例を次に示す．
   例１：代表業務，定常業務
   例２：システム運用(通常運用)
3. ステップ 2 で抽出したインスタンスを，ステップ１のいずれか一つのインスタンスの下位階層に位置づける．

ステップ 2 で抽象化を実施した理由は，品質特性クラスの場合と同様である．

3.2.3 テスト実行条件

テスト実行条件に相当するものは，例えば実行日時，テストデータの特徴，あるいは，システムにアクセスが集中している場合等，多岐にわたる．そこで，テスト実行条件を次の A）B）に分類した．

A) 個々のユーザや連携している外部システムから，テスト対象システムへのアクセスする時の条件
B) テスト対象のシステムの状態を示す条件

A)に属するテスト実行条件は，文章を明確に記述するための基本である 5W に基づいて詳細化を実施する．

B）に属するテスト実行条件は，複数のユーザや外部システムからのアクセスによって生じるシステムの状態を示す条件と，システムを構成しているミドルウェアや機器等の設定によって決定されるシステム自身の状態を示す条件に詳細化する．前者の一例はリソース不足，後者の一例はサーバー機器のセキュリテイ設定である．

以下のステップでテスト実行条件クラスのインスタンスを階層的に作成する．

1. 5W の中の４つの W である“いつ”,“どこで”,“誰が”,“何を”とシステムの状態とシステムの設定を最上位のインスタンスとする．
2. 実プロジェクトのシステムテスト計画書やシステムテスト仕様書からテスト実行条件クラスに該当する具体例を抽出する．抽出したものを抽象化してインスタンスとする．抽象化したテスト実行条件の例を２つ示す．例１は“いつ”の，例２は“誰が”のカテゴリに属するものである．
   例１：業務を実施する日時に関わる条件
   例２：未登録ユーザ，権限のあるユーザ，権限のないユーザ
3. ステップ 2 で抽出したインスタンスを，ステップ１のいずれか一つのインスタンスの下位階層に位置づける．

ステップ 2 で抽象化を実施した理由は，品質特性クラスの場合と同様である．

3.3 テストタイプ定義の作成

3.1 節で抽出した各テストタイプに対し，3.2 節の手順で詳細化したテストモデルの品質特性，テスト対象，テスト実行条件の各インスタンスの適切な組合せを決定する．この時，テストタイプの調査時に取得したテストタイプの特徴を反映したものにする．例えば，年度をまたがって実施される業務の動作を確認することが特徴であれば，テスト実行条件の中から,“いつ”のカテゴリの中から“業務を実施する日時に関わる条件”を用いた組み合わせを作成する．

今回の取り組みで, ST で実施すべきテストタイプとして 34 個を抽出し, 各々に対して図 2 のようなテストタイプ定義を作成した．また，テスト対象，テスト実行条件の各クラスのインスタンスを階層的に定義した．階層の最下層のインスタンス数は，品質特性クラスが 60 個，テスト対象クラスが 33 個，テスト実行条件クラスが 59 個で，計 152 個である．なお，１つのインスタンスが複数のテストタイプ定義で使われることがあるので，全テストタイプ定義中のインスタンスの出現数は計 404 個である．

4. システムテスト計画書導出手順

本章では，2.4 節に示したシステムテスト計画書を導出する手順について述べる．この導出手順は，誰でも漏れなくシステムテスト計画を作成することを目標としたものである．以後の節で，手順の構成や特徴を説明する．

4.1 システムテスト計画書導出手順の構成

我々は，2.4 節に示した形式のテスト計画書を導出するための手順をテストタイプ毎に作成した. この手順は,“参照文書”と“抽出する情報”と“作業内容”の３つの項目から構成されている．各々の項目について説明する．

参照文書は，キーワードを作成する元となる情報を取得する文書や参照すべき章，節，図表名等箇所を記載するものである．参照文書は，社内標準として規定されている文書名や記述項目名を用いた．

抽出する情報は，参照文書から抜き出す情報を示したもので，例えば，サーバー名，システム機能名，外部インタフェース名である．

作業内容は，抽出した情報をキーワードに変換する手順を示したものである．変換手順には，抽出した情報がそのままキーワードになるパターンと，いくつかの情報を組み合わせてキーワードにしているパターンがあった．

導出手順を上記の構成にした理由を示す．我々は，実プロジェクトで作成した上流工程文書や ST 工程で作成した文書を比較・分析し，高品質のシステムテスト計画を作成できる者は，以下のステップを踏んでいると推測した．

1. 要件定義工程や基本設計工程で作成した文書を参照し，その中からテストタイプに応じた情報を抽出．

2. 抽出した情報からある変換手順に従ってテスト計画書の大分類や中分類への展開時に必要なキーワード作成.
3. 中分類には, キーワードを含んだテスト内容を記述.

我々は ST に高スキルを保持している者のノウハウは，ステップ 1 で参照するする上流工程文書やそこから抽出する情報，ステップ 2 のキーワードを作成する変換手順に現れていると考えた．そこで，これらを明示するための項目を導出手順に設けた．

4.2 システムテスト計画書の導出

本節では，前節に示した導出手順を利用して，あるテストタイプのシステムテスト計画書を導出するステップを述べる．

(1) 文書の準備

導出手順の参照文書項目には，キーワード導出のために参照すべき文書が社内標準で規定している文書名や記述項目名で指定されている．各プロジェクトで作成した文書の中から，指定された標準文書や記述項目に該当する文書や記述項目を選択する．

例えば，導出手順においてハードウェア一覧とハードウェアに対する多重障害発生時の業務継続性要件が記載されている文書が参照文書として指定されている場合， 図 5 のようなシステムで使用するハードウェアが列挙された表を含む文書を選択する．

No.	システム		冗長化構成	台数
	環境	サーバ名		
1	本番環境	業務APサーバ	並列構成 VMware HA構成	2
2		ポータルサーバ	並列構成 VMware HA構成	2
3		DBサーバ	運用待機構成	2
4		検索サーバ	VMware HA構成	1

図 5：上流工程文書(ハードウェア一覧)の例

同様に，多重障害発生時の業務継続性要件が記載された文書も選択する．以後のステップでは，上図の例に含まれるポータルサーバーを例にして説明する．

(2) 情報の抽出

(1)で準備した文書や記述項目の中から，導出する項目に記載された情報を抜き出す．

例えば，ハードウェア一覧から抽出する情報としてサーバーの名称や台数，冗長化構成の有無が指定されていた場合は，図 5 の例に示された表から“ポータルサーバー”が“2 台”で,“冗長化構成である”いう情報を抽出する．他のサーバーについても同様に情報を抽出する． また，多重障害発生時の業務継続性要件が記載された文書から抽出する情報として，業務継続性が必要なサーバー名と何重障害まで動作確認するか，という二つが指定されていた場合には，参照文書からポータルサーバーは業務継続性が必要であり，２重障害発生時の動作確認を行う，という情報を抽出する．

(3) キーワードの導出

(2)で抽出した情報を用いて，作業内容項目に記載された手順に従って情報を転記したり組み合わせたりしてキーワードを導出する．導出したキーワードは，システムテスト計画書の雛形の大分類や中分類に記述する．

例えば，作業内容項目に

- 各サーバーと単一障害時の組み合せを作成．サーバーが N 台ある場合は，N 台分作成する．
- N 重（＝冗長化数）障害時の業務継続性が必要なサーバーに対して，N 重以内の障害の組み合せを作成

と書かれている場合，ポータルサーバーに関しては以下のようなキーワードを導出する．他のサーバーに対しても同様にキーワードを導出し，テスト項目の大分類，中分類に記述する．

1	ポータルサーバー#1, 二重冗長化, 単一障害発生時
2	ポータルサーバー#2, 二重冗長化, 単一障害発生時
3	ポータルサーバー#1, #2, 二重冗長化, 二重障害発生時

図 6：導出されたキーワードの例

(4) 中分類の内容記述

中分類の各カテゴリに対して，テストで実施すべき内容を記述する．テストで実施すべき内容は，(3)で導出したキーワードやテストタイプ定義を参照しながら記述する．

5. 評価

我々は４章に示した導出手順の有効性を，正しさと効率性向上の２側面から評価した．

導出手順の正しさは，導出手順によって上流工程文書から，システムテスト計画書の大分類や中分類の作成に必要なキーワードをどの程度漏れなく無駄なく抽出できたかで評価する．また，導出手順を利用することによる効率性向上は，１キーワードあたりの導出時間で評価する．

5.1 評価対象

社内の実プロジェクト２件(プロジェクト A, プロジェクト B)から開発工程で作成した要件定義書，基本設計書，運用関係および ST 関連の文書を提供いただいた．各プロジェクトの規模とシステムテスト計画書に記載された中分類数を表 1 に示す．

表 1：プロジェクト規模

プロジェクト名	A		B
業種	金融		産業・流通
テストタイプ	Type1	Type2	Type3
成果物数 (ファイル数)	1308		3698
中分類数	39	16	67

5.2 評価方法

次に評価方法を示す．今回の評価は次の４ステップで実施した．作業効率性を測定するため，下記ステップの(2)と(3)の作業時間を計測した．なお下記ステップの(1)～(3)は，4.2 節に示した利用手順の(1)～(3)に該当する．

(1) **文書の準備**

被験者は導出手順の参照文書に記載された文書を準備する．

(2) **情報の抽出**

被験者は，(1)で準備した文書から，キーワード導出に必要な情報を抽出する．

(3) **キーワードの導出**

抽出した情報を用いて，作業内容に従ってキーワードの集合を導出する．今回の検証では，中分類のキーワードのみを導出した．

導出したキーワードの集合を E と呼ぶ．また，実プロジェクトのシステムテスト計画書に記載されているキーワードの集合を R 呼ぶ．

(4) **キーワードの対応付けと分類**

次に集合 E と，集合 R に含まれるキーワードの対応付けを行い，その結果からキーワードを次の３つに分類した．

- (i) 導出のみ
  集合Ｅにのみ含まれるキーワード．
  E∩ ¬R に該当する集合．
- (ii) 対応付け可
  集合Ｒ，Ｅ両方の集合に含まれるキーワード．
  E∩R に該当する集合．
- (iii) 実物のみ
  集合Ｒのみに含まれるキーワード．
  ¬E∩R に該当する集合．

5.3 評価基準

我々は，導出手順を用いることで漏れのないテスト計画が作成できたかを評価するため，実プロジェクトのテスト計画書に記述された中分類に含まれるキーワードうち，我々が作成した導出手順によって抽出できた割合に着目した．この割合を導出率と呼ぶ．

また，導出手順を用いることで無駄のないテスト計画が作成できたかを評価するため，我々が作成した導出手順によって導出したキーワードのうち，実プロジェクトのテスト計画書の中分類にも含まれるキーワードの割合に着目した．この割合を適合率と呼ぶ．

以下に導出率と適合率の定義を式(1)(2)に示す．どちらも値が大きいほど漏れや無駄のないキーワード抽出ができたことを表す．

$$導出率(\%) = \frac{|E \cap R|}{|R|} \times 100 \qquad (1)$$

$$適合率(\%) = \frac{|E \cap R|}{|E|} \times 100 \qquad (2)$$

5.4 測定結果と考察

表 2 にキーワードを３種類に分類した結果と導出率，適合率，キーワードの導出に要した作業時間を示す．

今回の測定では，導出率はいずれも 80%以上であった．この値は十分満足のいくものではないが，ある程度の正しさでテスト計画書が導出できたと判断した．

表 2：測定結果

プロジェクト		A		B
テストタイプ		Type1	Type2	Type3
分類	(i)導出のみ (E∩ ¬R)	5	13	49
	(ii)対応付け可 (E∩R)	32	14	56
	(iii)実物のみ (¬E∩R)	7	2	11
導出率 (ii)/{(ii)+(iii)}		82%	88%	84%
適合率 (ii)/{(i)+(ii)}		86%	52%	53%
作業時間 （秒/キーワード）		51.89	26.67	－

一方，適合率はプロジェクト A のテストタイプ Type1 では 80%以上であるが，他の２件では 50%を少し超えた程度である．適合率が低くなる原因として，次の二つを想定した．

①上流工程文書の品質が低い，
　もしくは，後工程の仕様変更が反映されていない．
②導出手順が不適切．

しかし，適合率だけでは，どちらが原因かが判断できない．そこで，導出率と組み合わせて次のように判断した．適合率が低いが導出率が高い場合は，導出手順が大きく誤っているとは可能性が低いと考えられる．そこで②の可能性は低い，つまり主たる原因は①だと判断する．適合率も導出率も低い場合は，原因は主に②だと判断する．

今回の測定結果では導出率はいずれも高いので，適合率が低い理由は主に①だと判断した．

導出の作業時間は，キーワード１個あたり１分以内で抽出できている．本評価の被験者は，ST のスキルは持たず，また，評価対象プロジェクトのメンバでもないことを鑑みると，十分に短時間で導出できたと判断した．

後日，プロジェクト A のメンバに，上記測定結果の説明を実施した．その結果適合率を下げた原因が分かった．プロジェクト A で作成したソフトウェアは，別プロジェクトが作成する統合基盤上に構築していた．要件定義工程や基本設計工程では，統合基盤で提供する機能についても網羅的に記述したが，ST ではテスト計画書のテスト項目から除外していた．そのため，上流工程文書には抽出すべき情報として存在するが，テスト計画書には含まれず，結果として適合率の低さとして現れたことが分かった．導出率が

100%でない理由も統合基盤に起因するものであった．これらは，導出手順においてアーキテクチャに対する考慮が不足していたためと判断した．なお，大分類や中分類の作成方法は，我々が想定したように上流工程文書から情報を抽出し，それを転記したり組み合わせたりして作成していることが確認できた．

6. 関連研究

過去，Myers や Beizer らは，主にプログラムテストや結合テストで利用できる技術，例えば制御フローやドメインテストについて詳細な説明をしている[8][9]. これらの技術はシステムテストでも利用可能としているが，システムテストでは業務運用性や性能等の非機能要件の確認に重点が置かれるため，これらの技術だけでは漏れなくテストを行うには不十分である．

また国際ソフトウェアテスト資格認定委員会(ISTQB)が作成したテスト技術者資格制度のシラバス[3]でもシステムテストについて説明しているが，抽象的な説明にとどまっており,テストの実行対象やテスト実行時の条件として，何を選択すれば漏れなくかつ無駄なくテストができるのかを示していない．

OMG の Testing Profile[10]の中でテストの構造を表すモデルが提案されている．このモデルは，作成済みのテストケースの構造や，テスト実行に必要な各種要素について示している．一方，テスト計画やテスト仕様書の作成段階で必要となる情報やその関係を表すものは要素として含まれていない．我々のモデルは，テスト計画やテスト仕様書の作成段階で必要となる情報やその関係を表す点が，Testing Profile のテストモデルとは異なる．

Srikanth や Li らは，Value-based Testing の研究を実施している[11][12]．これらは，テストケースにステークホルダーにとっての価値を設定し，その価値によってテストケースの実行順序を決定する技術である．今後テストケースの導出手順が整備されると，非常に多数のテストケースが導出される可能性がある．ステークホルダーにとってのテストケースの価値を設定することで，重大障害の発見漏れを防ぎつつ効率的にテストを実施することが期待される．

7. まとめと今後の課題

本論文では，テスト対象，品質特性，テスト実行条件に基づいたテストタイプの定義と，そのテストタイプ毎にテスト計画を漏れなく無駄なく導出する手順について述べた．この導出手順を用いることで，“ 漏れ ” の面からは 80%以上の精度でシステムテスト計画書が生成できた．また，システムテストで実施すべきテストタイプをテストモデルの要素を用いて定義した. テストタイプ定義を用いることで，システムテスト計画やシステムテスト仕様の品質向上が期待できる．今後は以下の課題に取り組む．

- テストタイプ定義，システムテスト計画書導出手順のブラッシュアップ．
  特に，ソフトウェアのアーキテクチャを考慮した導出手順へのブラッシュアップ
- テスト仕様作成方法の確立
- テストタイプやテスト仕様の優先順位づけを実施し，無駄のない効率的なテストの実施

参考文献

1) 情報処理推進機構ソフトウェア・エンジニアリング・センター(IPA/SEC)，高信頼化ソフトウェアのための開発手法ガイドブックガイドブック－予防と検証の事例を中心に－，http://sec.ipa.go.jp/users/publish/SEC-TN10-005.pdf, p.3，2011
2) 8 割が役に立たないシステムを経験，「目的が不明確」「トップがダメ」「使い手を無視」が理由，http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20050123/155108/
3) ISTQB 著, JSTQB 翻訳, テスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2011.J02, http://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2011.J02.pdf, p.27
4) ISTQB 用語集作業班著，JSTQB®技術委員会翻訳，ソフトウェアテスト標準用語集(日本語版) Version 2.0.J02, http://jstqb.jp/syllabus.html
5) 情報処理推進機構ソフトウェア・エンジニアリング・センター(IPA/SEC)，ソフトウェアテスト見積りガイドブック~品質に応じた見積りとは~，http://www.ipa.go.jp/files/000005132.pdf , p.8, 2009
6) JIS X 0129-1：ソフトウェア製品の品質―第 1 部：品質モデル，http://www.jisc.go.jp/
7) 栗原英俊, 宗像一樹, 野津昌弘, 山本晃治, 大橋恭子, 山本里枝子， ビジネスシステム向け要求モデルに基づく要求定義手法の提案と評価，ソフトウェアエンジニアリングシンポジウム 2008, pp. 131-138
8) J.Myers 著，長尾真監訳，ソフトウェア・テストの技法，近代科学社，1980
9) ボーリス・バイザー著,ソフトウエアテスト技法～自動化、品質保証、そしてバグの未然防止のために～,日経ＢＰマーケティング, 1994
10) UML Testing Profile(UTP) Version1.2, http://www.omg.org/spec/UTP/1.2/,
11) H. Srikanth, S. Banerjee, Improving test efficiency through system test prioritization, The Journal of Systems and Software 85 (2012) , pp. 1176–1187
12) Li, Q. and Li, M. and Yang, Y. and Wang, Q. and Tan, T. and Boehm, B. and Hu, C., Bridge the gap between software test process and business value: A case study, Trustworthy Software Development Processes， pp. 212-223, Springer, 2009