テクニカルノート

デュアルユース型冗長構成方式における機能検証手順

久野 巧 †

筆者らは重要なデータの処理および通信だけを首尾一貫して多重化することにより，コストパフォーマンスとフォールトトレランスの両立を可能にするデュアルユース型冗長構成方式をすでに提案してきた．提案方式に基づくシステムは，平常時には機能検証を行い，非常時（フォールト発生時）にはフェイルソフトな動作に移行する．本論文では，特にデュアルユース型冗長構成方式における機能検証手順について説明する．実験システムに対して機能検証手順を適用した結果，ネットワーク切断や輻輳をフォールトとして検出し，その有効性を確認することができた．

A Functional Verification Procedure for the Dual-Use Redundant System Architecture

Takumi Hisano†

We have already proposed a new redundant system architecture called Dual-Use Redundant System, which involves multiplexing of all modules to improve fault tolerance and using abstract data to reduce the amount of redundant modules. In this paper we describe a functional verification procedure for the Dual-Use Redundant System and show experimental results of the verification.

1. はじめに

ユビキタスコンピューティングの浸透にともなって，今後，あらゆる人たちが情報支援システムからのサービスを享受できるようになると考えられる．他方において，情報支援システムの存在があたりまえになった社会では，システムの障害が近代的な生活の営みをすべて停止させてしまう危険性もはらんでいる．筆者らは，そのような危険を回避するためのデュアルユース型冗長構成方式を提案している[1)]．

フォールトトレランス（耐フォールト性）の向上を目指す冗長構成方式は過去に数多く提案され，すでにいくつかは実用化されている[2)]．しかし，従来方式は同じ構成要素の多重化を基本としているために，構築コストの大幅な上昇を招いていた．デュアルユース型冗長構成方式は，重要なデータの処理および通信だけを首尾一貫して多重化することにより，コストパフォーマンスとフォールトトレランスの両立を可能にする．本論文では，特にデュアルユース型冗長構成方式の機能検証手順を取り上げ，その有効性確認のための初期的な実験について述べる．

2. デュアルユース型冗長構成方式の概要

デュアルユース型冗長構成方式の基本形を**図 1** に示す．通常のデータを扱う部分を第 1 レベルモジュール，重要なデータ☆だけを扱う部分を第 2 レベルモジュールと呼ぶ．ここで「重要なデータ」とは要求仕様で定められた機能実現上不可欠なデータをいう．

この基本形を 2 重の通信ネットワーク経由で双方向に結合することによって，複数のサブシステムからなるデュアルユース型システムを構築する（**図 2** 参照）．

本方式の要点は次のとおりである．

(1) システム動作にかかわる重要なデータの処理および通信を生成段階から最終出力段階まで首尾一貫して多重化する．
(2) 平常時にマルチレベルシミュレーション法による機能検証を実行する．
(3) 通常のデータを扱う部分（第 1 レベル）でフォー

† 独立行政法人産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)

☆ 重要なデータは後述の抽象化操作によって通常のデータから抽出される．

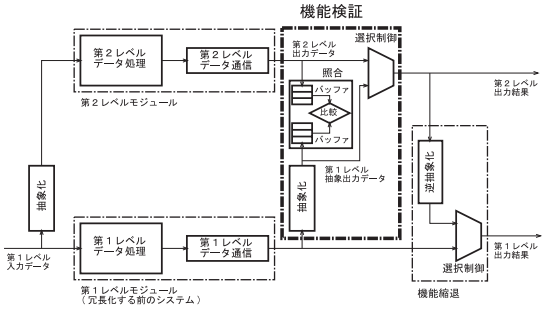

図 1 デュアルユース型冗長構成方式の基本形
Fig. 1 Base unit of the dual-use redundant system architecture.

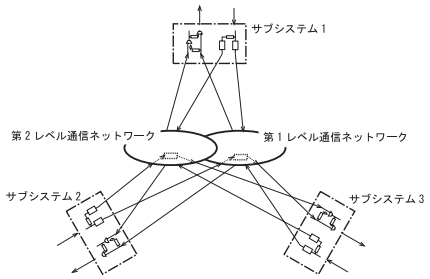

図 2 基本形を結合したデュアルユース型システムの構成例
Fig. 2 Typical configuration of the dual-use redundant system composed of base units.

ルトが発生した場合には，第 2 レベル出力データを用いてフェイルソフトな動作に移行する☆．重要なデータを扱う部分（第 2 レベル）でフォールトが発生した場合には，平常時と同じ結果（第 1 レベル出力データ）を出力する．

また，その目的はシステムに以下の特徴を付与することにある．

特徴 (a) 情報量の少ない重要なデータだけを扱うモジュールを使って多重化することにより，システムの構築コストの増大を抑制する☆☆．

特徴 (b) 多重化しない部分をデータ生成段階と最終出力段階に限定することで，広範囲のフォールトに対応した処理を可能にする☆☆☆．

本論文で述べる機能検証手順は特徴 (b) を裏付ける手段となる．従来のエラー検出や機能検査（functional test）[2] との違いは次のとおりである．

通常の冗長システムにおいて非常時用に設けた予備モジュールは，エラー検出や機能検査のための比較対象としても利用される．たとえば，2 重化モジュールを用いた静的冗長システムは，予備モジュールと現用モジュールの出力を比較し，モジュール単独のエラー検出結果と総合して出力を切り替える．また，ソフトウェアの多重化である N-Version 法やその拡張である N-Self Checking Programming 法[3] においても，複数モジュールの出力を直接比較して最終出力や構成を制御する．本方式における機能検証は，モジュール出力データの単純な一致比較ではなく，異なる箇所（第 2 レベルモジュールの入力部と第 1 レベルモジュールの出力部）での抽象化操作を経たデータを照合する点が従来方式と異なる．

3. 機能検証

デュアルユース型冗長構成方式の機能検証手順について述べる．機能検証は，2 つの表現レベルで表された機能的に等価な処理モジュールに等価なデータを入力すれば等価な出力データが得られるという原理に基づく．この検証法は当初，論理回路設計におけるマルチレベルシミュレーション[4] で採用された．所与の機能表現と構造表現に基づいて処理された出力結果を照合することで設計の正しさを検証する．この場合，システムの要求仕様を表した機能表現とその仕様を実現した構造表現が 2 つの等価なモジュールを構成している．

本方式の機能検証手順は，論理回路設計用マルチレベルシミュレーションの検証手順を拡張したものである．要求仕様で定められた機能実現上不可欠な重要データを扱う第 2 レベルモジュールが機能表現の処理モジュールに，通常のデータを扱う第 1 レベルモジュールが構造表現の処理モジュールにそれぞれ対応する．論理回路設計用マルチレベルシミュレーションとの大きな違いは，検証対象を稼働中のシステムにし

☆ フェイルソフトな動作に移行することを機能縮退という．フォールト発生時のデータ選択は，抽象化操作や誤り検出の際に単調減少する値を用いて図 1 の選択制御部が行う．本論文では機能縮退の具体的な説明を省略する．詳細は文献 1) を参照．

☆☆ 同じ仕様のハードウェアを複数用意し，さらに同じ仕様のソフトウェアを独立に複数実装することによって N 多重化した冗長システムでは，通常 N 倍を超える構築コストを想定しなければならない．本方式の第 2 レベルモジュールは，第 1 レベルモジュールに比べて低性能のハードウェアと制約の緩やかなソフトウェアで実現可能になる．構築コストに関する要求が冗長化する前のシステムの 2 倍未満であるような領域が本方式の適用範囲である．

☆☆☆ コストパフォーマンスを優先した従来の冗長システムでは，統計情報や経験をもとにしてフォールトの起きやすい部分を特定し，該当部分だけを多重化する．また，構築コストよりもフォールトトレランスを最優先にする特殊な分野においては，ソフトウェアを含むすべての構成要素を首尾一貫して多重化する完全多重化システムを採用する．多重化していない部分を最小限にした本方式では，従来の完全 2 重化システムと同程度のフォールト検出が可能となる．なお，本論文では機能的に等価なものを重複させる形態を多重化と呼んでいる．

たことにある．この拡張によって，機能検証用テストパターンの作成を省略可能にしただけでなく，完全多重化システム以外では実現困難であった過負荷やソフトウァア設計上のバグ等の広義のフォールトを発見できるようになった．

3.1 機能検証の前提条件

機能検証の対象となるシステムは，次の前提条件を満足しているものとする．

(1) 第1レベルモジュールと第2レベルモジュールが機能的に等価である．
(2) 第1レベルモジュールと第2レベルモジュールの入出力データを対応付ける手続き（抽象化操作）が存在する．
(3) 第1レベルモジュールと第2レベルモジュールにおける処理時間や通信時間の差を補償する機構が存在する．

3.2 機能検証手順

機能検証手順を以下に示す．

入力 第1レベルモジュールへの入力データ．
出力 検証成功時「true」，検証失敗時「false」．
手順

1. 特定の入力データを第1レベルモジュールに入力する．
2. 手順1の入力データから抽象化操作によって情報量を縮減し，その縮減したデータを第2レベルモジュールに入力する．
3. 第1レベルモジュールと第2レベルモジュールを同時並行的に動作させる．
4. 第1レベルモジュールの出力データから抽象化操作によって情報量を縮減し，第1レベル抽象出力データを得る．
5. 第2レベル出力データと第1レベル抽象出力データを最大 Td 時間保持する．
6. 対応する第2レベル出力データと第1レベル抽象出力データを比較し内容が一致していれば，検証は成功する．さもなければ検証は失敗する．
7. 手順6において，対応する第2レベル出力データと第1レベル抽象出力データが存在しなければ，検証は失敗する． □

検証出力結果が false のとき，第1レベルモジュールあるいは第2レベルモジュールのいずれかにフォールトが発生したと判断する．

3.3 モジュール出力データ間の時間差の補償

本方式のように処理/通信能力の異なるモジュールでデータの生成段階から最終出力段階までを一貫して多重化する場合，従来のように故障しやすい部分を同一構成要素で多重化する場合に比べて，第1レベルモジュールと第2レベルモジュールの出力データ間の時間差は大きくなる可能性がある．以下では，検証手順5で仮定した，データを一定時間保持することで時間差を吸収する機構について説明する．この機構の特徴はメタデータとしての単調増加数と照合部入力端のバッファにある．

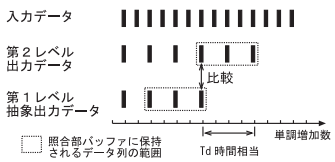

図 3 システム各部を通過したデータの履歴
Fig. 3 Data sequences through the system components.

システムの入力データには生成段階で処理単位ごとに単調増加数を付加する．いったん付加された単調増加数が処理途中あるいは通信途中で変化することはない．また，照合部の入力端に設けた2つのバッファは，到着時から Td 時間分に相当するデータを単調増加数とともに保持する．その単調増加数に基づいてバッファからデータを取り出す．Td は許容時間差である．具体的な Td の値はシステムが正常に稼働している状況での時間遅れを計測して決定する．

図 3 は，特定時刻までにシステム各部を通過した，入力データと第2レベル出力データ，第1レベル抽象出力データのイメージを表す．第1レベル出力データに抽象化操作を適用したものが第1レベル抽象出力データである．図中の小矩形が処理単位ごとのデータを示し，点線の枠が各バッファに保持されるデータ列の範囲を示す．

バッファ内の同じ単調増加数を持つデータに対して検証手順6の比較操作が適用される．

上記機構によって，ハードウェア故障だけでなく，過負荷や輻輳もフォールトとして検出可能になる．すなわち，第1レベルのデータ処理/通信において過負荷や輻輳を起因とする遅れが発生し，第2レベルとの時間差が Td を超えて増大すると，2つのバッファ間で同じ単調増加数を持つデータが存在しなくなり，比較操作が実行不能となる．この時点で検証は失敗し，フォールトが発生したと判断される．

3.4 実験システムにおける機能検証

機能検証の有効性を確認する目的で簡単な画像情報処理を行うデュアルユース型実験システムを構築した．

3.4.1 実験システムの構成と環境

実験システムは「来訪者の到着を知るために施設入口にビデオカメラを設置し，距離的に離れた受付や待

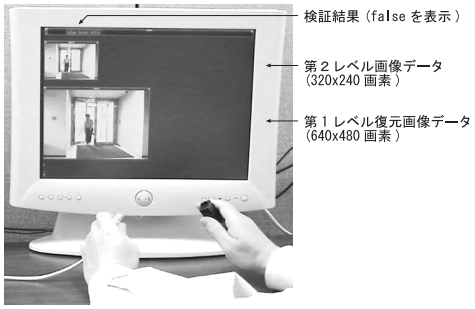

図 4 実験システムにおけるネットワーク切断時の機能検証の実行例
Fig. 4 Example of functional verification in the experimental system.

機場所に置いたモニタテレビに来訪者の姿を表示するシステム」である．

第 1 レベルモジュールは，ビデオカメラからの画像データを蓄積し転送する機能を持つ．第 1 レベルモジュールの通信路は 100 Mbps を上限とするネットワークである．第 1 レベルデータは 640×480 画素 10 fps カラー画像データである．第 2 レベルモジュールは，第 1 レベルデータの画像データから情報量を縮減した，320×240 画素 1 fps モノクロ画像データを処理する．第 2 レベルモジュールの通信路は 10 Mbps を上限とするネットワークである．モニタテレビが検証結果と出力画像データを表示する．

単調増加数として 10 msec 単位で整数化した時刻を用いた．抽象化操作は，10 フレーム分の 640×480 画素カラー画像データを 1 フレーム分の 320×240 画素モノクロ画像データに変換する処理である．

検証手順 6 の比較操作は，640×480 画素 10 fps カラーの第 1 レベル出力データに抽象化操作を適用したもの（320×240 画素 1 fps モノクロ画像データ）と第 2 レベル出力データ（320×240 画素 1 fps モノクロ画像データ）に対して実行される．

この実験システムは 3.1 節の前提条件をすべて満足している．すなわち，第 1 レベルモジュールと第 2 レベルモジュールは画像処理とデータ転送に関して機能的に等価であり，第 1 レベルデータを第 2 レベルデータに対応付ける抽象化操作および処理/通信時間の差を補償する機構（Td=1 sec）が存在する．

3.4.2 実験結果と考察

実験システムを稼働させたところ，平常時にはシステムが正常であることを示す「true」と画像データをモニタテレビに表示した．意図的にネットワーク切断や輻輳を生じさせた場合，モニタテレビに「false」を表示し，フォールト発生を検出した．**図 4** に 100 Mbps ネットワーク切断時のモニタテレビの表示例を示す．

この実験結果から，時間遅れをともなうデータ処理/通信環境において，提案した機能検証手順が有効にフォールトを検出できることが分かった．

4. おわりに

デュアルユース型冗長構成方式における機能検証手順について述べた．簡単な実験システムを構築して機能検証手順を適用した結果，ネットワーク切断や輻輳をフォールトとして検出し，その有効性を確認することができた．

なお，今回の実験は時間遅れをともなう環境での機能検証手順の有効性確認が主目的であったため，データベース等の内部状態を参照/更新する処理内容は含まれていない．また，機能検証のための前提条件がどの範囲の処理内容にまで適用可能かも明確になっていない．

今後は，広範な処理内容を含む，より本格的な冗長システムの構築実験等を通して定量的な評価を実施したいと考えている．

参考文献

1) 久野　巧：フェイルソフトな社会情報システムのためのデュアルユース型冗長構成方式，情報処理学会研究報告，Vol.2002, No.115, pp.163–170 (2003).
2) Geffroy, J.C. and Motet, G.: *Design of Dependable Computing Systems*, Kluwer Academic Pub. (2002).
3) Lyu, M.: *Handbook of Software Reliability Engineering*, McGraw-Hill and IEEE Computer Society Press (1996).
4) 久野　巧：マルチレベルシミュレーションによる多階層モデルの検証，電子情報通信学会論文誌，Vol.J76-D-II, No.4, pp.908–913 (1993).

(平成 16 年 1 月 29 日受付)
(平成 16 年 5 月 11 日採録)

久野　巧（正会員）

昭和 30 年生．昭和 54 年通商産業省工業技術院電子技術総合研究所入所．設計支援システム，協調計算アルゴリズムに関する研究に従事．現在，独立行政法人産業技術総合研究所主任研究員．フェイルソフト型エージェントシステムに関する研究に従事．電子情報通信学会，ソフトウェア科学会，人工知能学会各会員．