

(19) 日本国特許庁(JP)

再公表特許(A1)

(11) 国際公開番号

W02012/073686

発行日 平成26年5月19日 (2014. 5. 19)

(43) 国際公開日 平成24年6月7日 (2012. 6. 7)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 11/30 (2006.01)	G06F 11/30 A	5B042
G06F 9/44 (2006.01)	G06F 9/06 62OK	5B376
G06F 11/36 (2006.01)	G06F 9/06 62OR	

審査請求 有 予備審査請求 有 (全 98 頁)

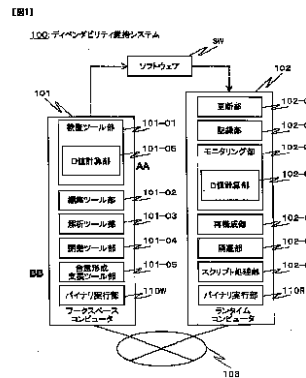
出願番号 特願2012-546760 (P2012-546760)	(71) 出願人 503360115 独立行政法人科学技術振興機構 埼玉県川口市本町四丁目1番8号
(21) 国際出願番号 PCT/JP2011/076219	(74) 代理人 110000338 特許業務法人HARAKENZO WORLD PATENT & TRADEMARK
(22) 国際出願日 平成23年11月14日 (2011. 11. 14)	(72) 発明者 横手 靖彦 日本国東京都港区虎ノ門3-14-1-2004
(11) 特許番号 特許第5280587号 (P5280587)	(72) 発明者 所 真理雄 日本国東京都大田区田園調布本町5-15
(45) 特許公報発行日 平成25年9月4日 (2013. 9. 4)	(72) 発明者 山本 修一郎 日本国愛知県名古屋市港区千鳥2-7-20-301
(31) 優先権主張番号 特願2010-267461 (P2010-267461)	
(32) 優先日 平成22年11月30日 (2010. 11. 30)	
(33) 優先権主張国 日本国 (JP)	
(31) 優先権主張番号 特願2011-177322 (P2011-177322)	
(32) 優先日 平成23年8月12日 (2011. 8. 12)	
(33) 優先権主張国 日本国 (JP)	

最終頁に続く

(54) 【発明の名称】 ディペンダビリティ維持システム、変化対応サイクル実行装置、障害対応サイクル実行装置、ディペンダビリティ維持システムの制御方法、制御プログラムおよびそれを記録したコンピュータ

(57) 【要約】

本発明のワークスペースコンピュータ(101)および/またはランタイムコンピュータ(102)は、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データに基づいて、対象システムのディペンダビリティの価値を定量的に示すD値を求める。



- 100 Dependency maintenance system
- 101 Workspace computer
- 101-01 Verification tool unit
- 101-02 Edit tool unit
- 101-03 Analysis tool unit
- 101-04 Development tool unit
- 102 Runtime computer
- 102-01 Update unit
- 102-02 Recording unit
- 102-03 Monitoring unit
- 102-04 Recognition unit
- 102-05 Aggregation unit
- 102-06 D value calculation unit
- 102-07 Script processing unit
- 102-08 Binary execution unit
- 110W, 110R Software
- SW Software
- AA D value calculation unit
- BB Consensus formation supporting tool unit

【特許請求の範囲】**【請求項 1】**

対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置であって、

対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得手段と、

上記記述データ取得手段が取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定手段と、を備えることを特徴とするディペンダビリティ維持装置。

【請求項 2】

上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルの少なくとも何れか一方を実行することを特徴とする請求項 1 に記載のディペンダビリティ維持装置。

【請求項 3】

上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記障害対応サイクルを実行するとともに、

上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、当該対象システムを再構成する再構成手段を備えることを特徴とする請求項 1 に記載のディペンダビリティ維持装置。

【請求項 4】

上記ディペンダビリティ記述データには上記再構成の手順が記載されており、

上記再構成手段は、上記ディペンダビリティ記述データに記載されている再構成の上記手順に従って、当該対象システムを再構成することを特徴とする請求項 3 に記載のディペンダビリティ維持装置。

【請求項 5】

上記障害発生または障害予兆を検知し、上記対象システムの停止が不回避であるとき、上記変化対応サイクルを実行する装置に、上記ディペンダビリティ記述データの変更の要求を送信する変更要求送信手段を備えることを特徴とする請求項 3 または 4 に記載のディペンダビリティ維持装置。

【請求項 6】

上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記変化対応サイクルを実行するとともに、

上記障害対応サイクルを実行する装置から、上記ディペンダビリティ記述データの変更の要求を受信したとき、上記ディペンダビリティ記述データを変更することを特徴とする請求項 1 に記載のディペンダビリティ維持装置。

【請求項 7】

上記ディペンダビリティ記述データは、互いに関連づけられたゴールノードとモニタノードとの組を規定するデータであり、

上記ゴールノードは、要求・仕様がゴール形式にて記述されたノードであり、

上記モニタノードは、上記ゴールノードに記述されたゴールが充足されていることを表明するエビデンスであるとともに、上記対象システムの監視点に対応付けられており、

上記ディペンダビリティ値決定手段は、対応付けられた上記監視点から取得されたデー

10

20

30

40

50

タが変動許容範囲に対して良好であるモニタノードに基づいて、上記評価値を計算することを特徴とする請求項 1 から 6 のいずれか 1 項に記載のディペンダビリティ維持装置。

【請求項 8】

上記ディペンダビリティ値決定手段は、上記対象システムをモニタして取得されたモニタ値が変動許容範囲に対して良好であったエビデンスに基づいて、上記評価値を計算することを特徴とする請求項 1 から 6 のいずれか 1 項に記載のディペンダビリティ維持装置。

【請求項 9】

上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記障害対応サイクルを実行するとともに、

10

上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに含まれるスクリプトを実行するスクリプト処理手段を備え、

上記スクリプトは上記対象システムを変動許容範囲の状態に回復させるシナリオを含むことを特徴とする請求項 1 に記載のディペンダビリティ維持装置。

【請求項 10】

対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、

20

上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクルを実行するワークスペース装置と、

上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行するランタイム装置とを含み、

上記ワークスペース装置および上記ランタイム装置の少なくともいずれか一方が、上記ディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めることを特徴とするディペンダビリティ維持システム。

【請求項 11】

他のディペンダビリティ維持システムとネットワークを介して接続されていることを特徴とする請求項 10 に記載のディペンダビリティ維持システム。

30

【請求項 12】

対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置の制御方法であって、

対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得ステップと、

上記記述データ取得ステップにて取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定ステップと、を含むことを特徴とするディペンダビリティ維持装置の制御方法。

【請求項 13】

40

請求項 1 から 9 のいずれか 1 項に記載のディペンダビリティ維持装置の上記各手段としてコンピュータを機能させるための制御プログラム。

【請求項 14】

請求項 13 に記載の制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 15】

監視制御対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置であって、

上記監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出し、読み出したディ

50

ペンダビリティ記述データから上記障害監視制御用データを生成するディペンダビリティ記述変換部を備えたことを特徴とするディペンダビリティ維持装置。

【請求項 16】

上記障害監視制御部は、上記監視制御対象システムの状態を監視し、必要な場合に対策を実行するために使用するモジュールを、複数のモジュールから選択して制御可能であり、

上記ディペンダビリティ記述データは、1つの記述が、上記モジュールを特定するためのモジュール特定情報を値として設定可能な変数を少なくとも含む形式であり、

上記ディペンダビリティ記述変換部は、あらかじめ設定された、上記モジュール特定情報と上記障害監視制御用データとの対応関係を示す情報に基づいて、上記ディペンダビリティ記述データに含まれる変換対象の記述を、該記述に含まれるモジュール特定情報に対応する障害監視制御用データに変換することを特徴とする請求項 15 に記載のディペンダビリティ維持装置。

10

【請求項 17】

上記ディペンダビリティ記述データの1つの記述から変数部分を除いた固定部分に対応して、上記対応関係が設定されており、

上記ディペンダビリティ記述変換部は、上記ディペンダビリティ記述データの記述に含まれる上記固定部分に対応する上記対応関係を参照して、該記述に対応する障害監視制御用データに変換することを特徴とする請求項 16 に記載のディペンダビリティ維持装置。

【請求項 18】

20

請求項 15 から 17 のいずれか 1 項に記載のディペンダビリティ維持装置と、

上記ディペンダビリティ記述格納部と、

上記障害監視制御部とを含み、

上記ディペンダビリティ維持装置が上記ディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから生成した障害監視制御用データに従って、上記障害監視制御部が動作することにより、上記監視制御対象システムの状態を監視し、必要な場合に対策を実行することを特徴とする障害対応システム。

【請求項 19】

上記監視制御対象システムの状態を監視する 1 以上のモニタモジュールと、

必要な場合に上記監視制御対象システムに対して対策を実行する 1 以上のアクションモジュールと、

30

上記障害監視制御部の制御により、上記モニタモジュールの選択および動作の制御を行うモニタ管理部と、

上記障害監視制御部の制御により、上記アクションモジュールの選択および動作の制御を行うアクション管理部と、を含むことを特徴とする請求項 18 に記載の障害対応システム。

【請求項 20】

監視制御対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置の制御方法であって、

40

上記監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出す読み出しステップと、

読み出したディペンダビリティ記述データから上記障害監視制御用データを生成する変換ステップとを含むことを特徴とするディペンダビリティ維持装置の制御方法。

【請求項 21】

請求項 15 から 17 のいずれか 1 項に記載のディペンダビリティ維持装置としてコンピュータを機能させるための制御プログラムであって、コンピュータを上記ディペンダビリティ記述変換部として機能させるための制御プログラム。

【請求項 22】

50

請求項 2 1 に記載の制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、時間軸上で変化可能なディペンダビリティ要求に対して、ある時刻における当該要求を表現したディペンダビリティ記述データを備える対象システムにおけるディペンダビリティの価値を計測、評価することを可能にし、当該対象システムのディペンダビリティを維持する装置等に関するものである。

【背景技術】

【0002】

昨今、銀行のオンラインシステムの停止、携帯電話やIP電話の通信障害、各種商用サービスのセキュリティ障害、等の重要な社会インフラストラクチャを担うシステムやサービス停止が目立ってきており、我々の生活に影響を与えている。その原因は、ひとえにこれらの組み込み型コンピュータシステムを利用した商品やサービスにおいて、その規模や複雑度が劇的に増してきていることに帰することができる。その原因をさらに掘り下げて調べてみると、人為的なミスが原因であることがかなりのケースで存在している。

【0003】

従来から、コンピュータシステムの信頼性、可用性、保守性、安全性、完全性、機密性に関しては、ディペンダビリティというコンピュータシステムの備えるべき性質として議論されてきている（非特許文献3）。組み込みシステム開発においては、最初に開発計画を立て、対象システムやサービスの機能要件および非機能要件を仕様としてきっちり書き出し、検証やテストを長期にわたって行い、デプロイする手法がとられてきた。しかし、上述のように、障害が日に日に件数を増してきている。CMMIやISO 26262をはじめとする規格では、人為的エラーを減らす試みもなされている。しかし、これら既存の技術・規格では、オープン環境におけるシステムという特性に対する考慮が欠けている。

【0004】

従来からの手法は、開発の開発時点における仕様が、コンピュータプログラムとして確実に実装され、かつその仕様が商品やサービスのデプロイ後も不変であるという仮定に基づいている。しかし、オープン環境では、開発の開発時点とデプロイ時点では環境が変化している。さらに、デプロイ後も環境は変化している。その結果、それらの変化への対応が求められる。

【0005】

そこで、独立行政法人科学技術振興機構は、CRESTプログラムにおいてDEOS (Dependable Embedded Operating Systems / Dependability Engineering for Open Systems) プロジェクト (<http://www.crest-os.jst.go.jp/>) を立ち上げ、組み込みシステム用のディペンダブル・オペレーティングシステムの研究開発を行っている。DEOSプロジェクトでは、オープン環境におけるディペンダビリティを、オープンシステムディペンダビリティとして次のように定義している。「現代の大規模ソフトウェアシステムは機能、構造、システム境界が時間的に変化し、これに起因する不完全さと不確かさを完全に排除することができず、未来に障害となりうる要因（開放系障害要因）を本質的に抱えている。オープンシステムディペンダビリティとは、それらの要因を顕在化する前にできる限り取り除き、また、顕在化した後に迅速かつ適切に対応し、影響を最小とするようにマネージし、利用者が期待する便益をできる限り安全にかつ継続的に提供し、社会への説明責任を全うし、およびそれらを継続的に行う能力を言う。」（非特許文献1参考）

また、従来、組み込みシステム開発においては（当該開発には限定されないが）、ステークホルダ群からの要求、及び当該要求に対する仕様書を作成し、それに基づいてシステム開発が行われている。具体的には、対象システムの機能要求、および非機能要求をまとめた仕様書群に従ってシステムは開発される。そして、運用中にシステムの一部を変更する場合には、仕様書群と当該システムの実装とが無矛盾に更新される。

10

20

30

40

50

【 0 0 0 6 】

当該仕様書群と当該システム実装の更新が無矛盾に行われる必要がある理由の一つとして、当該システムのディペンダビリティ（非特許文献1あるいは非特許文献3等を参照）は、いかに環境が変化しても維持しなければならないことがあげられる。そのため、当該仕様書に対応したディペンダビリティ記述データの更新と、それを実現するためのシステムを監視制御するモジュールの開発・追加が、常に当該仕様書群と無矛盾に行われることを保証しなければならない。

【 0 0 0 7 】

また、非特許文献2には、Safety Caseと呼ばれるシステムの安全性を示すドキュメントを、システムのライフサイクル（概念、開発、運用、保守、更新、廃棄など）を通じて、システムの変更に応じて更新することの必要性が記載されている。

10

【 0 0 0 8 】

Safety Caseは、イギリスなどで、原子力発電所など、高い安全性が求められるシステムを開発・運用する際に認証機関に提出が義務付けられるまでに普及している、システムの安全性の根拠（エビデンス）を示すための、構造化されたドキュメントである。自動車の機能安全規格であるISO 26262でも提出が義務付けられるなど世界的に普及しつつある。

【 先行技術文献 】

【 特許文献 】

【 0 0 0 9 】

20

【 特許文献 1 】 米国特許第 7 7 5 6 7 3 5 号明細書（ 2 0 1 0 年 7 月 1 3 日 ）

【 非特許文献 】

【 0 0 1 0 】

【 非特許文献 1 】 Mario Tokoro, "Challenge to Open Systems Problems," September 29, 2010. <http://www.stanford.edu/class/ee380/Abstracts/100929.html> [平成 23 年 7 月 7 日 検索]

【 非特許文献 2 】 Peter Bishop and Robin Bloomfield, "A Methodology for Safety Case Management," Safety Critical System Symposium, 1998. <http://www.adelard.com/papers/sss98web.pdf> [平成 23 年 7 月 7 日 検索]

【 非特許文献 3 】 Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11-33, Jan.-Mar. 2004, doi:10.1109/TDSC.2004.2.

30

【 非特許文献 4 】 所眞理雄、JST-CREST「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム-DEOSプロジェクト-White Paper Version 2.0」、DEOS-FY201-WP-02J、2010/12/01

【 非特許文献 5 】 Jin Nakazawa, Yutaka Matsuno, and Hideyuki Tokuda, "Evaluating Degree of Systems' Dependability with Semi-Structured Assurance Case" Proceedings of the 13th European Workshop on Dependable Computing (EWDC 2011), 2011.

40

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 1 】

システム規模の拡大や複雑度の増大は、当該システムやそれが提供するサービスに関する要求・仕様をも複雑にしている。そのため、すべての要求を完全にシステム開発前に抽出したり、全ての仕様を完全にシステム開発前に記述することが不可能（仕様の不完全性）になっている。仕様の不完全性に対応する実装も完全ではなく（実装の不完全性）、当該システムが提供するサービスの振る舞いも完全には把握することができない。その結果、「何をどこまで保証したら良いのか？」「保証可能なのか？」も分からない。また、これらの不完全性はステークホルダ間でのシステムやサービスに対する要求理解の違い（誤解）を招いたり、人為的なミスを誘導したりする。

50

【 0 0 1 2 】

さらに、先に述べた環境の変化に対応するための要求が新たに発生する。また、開発時点の要求にも修正が必要になる。「環境がどのように変化するか？」は事前には分かり得ないので、「現在のシステムの動作が、環境変化に確実に対応できるか？」もわからない。このような変化に対する不確実性はシステム動作予測を困難にし、システム障害に繋がる。

【 0 0 1 3 】

これらの不完全性や不確実性はシステム障害が発生した際に、その原因調査に時間がかかるようになり、またその適切な対応を困難にし、ステークホルダが自らの説明責任を果たすことも難しくなる。

10

【 0 0 1 4 】

このような状況下において、従来、非特許文献 2 においては、システム開発時点で想定される障害対応を Safety Case として記述し、ステークホルダ間で合意することで、障害発生時点でのステークホルダの説明責任の達成を可能にしている。しかし、環境の変化に起因する Safety Case の変更には対応できない。

【 0 0 1 5 】

また、非特許文献 3 は、Faults-Errors-Failures という変化で障害をとらえることで障害の発生そのものを抑えようとしている。しかし、前記不完全性や不確実性を扱うことはできない。

【 0 0 1 6 】

また、非特許文献 5 には、前記不完全性や不確実性を扱うための手法に関して記載されている。しかし、その記述は実行環境とは分離されており、また、記述をモデル化しディペンダビリティの価値を計測・評価することに関しては設計時のみにて可能であるに過ぎない。

20

【 0 0 1 7 】

一方、特許文献 1 は Enterprise Architecture における複雑性を数学的に制御する手法に関して述べている。複雑性が制御できればディペンダビリティの向上につながる事が期待できるが、前記不完全性や不確実性までは扱っていない。

【 0 0 1 8 】

また、従来のシステム開発の手法では、複数のステークホルダ群からの要求は合意された仕様書にまとめられていなかったり、仕様書にまとめることが容易でないことに加えて、運用後にシステムの一部に変更があった場合に、仕様書とシステム実装の変更とを無矛盾に維持することができなかった。

30

【 0 0 1 9 】

例えば、上記 Safety Case でも、Safety Case に記載されている、ステークホルダが Safety Case 上で合意した障害対応内容と、実際の対象システムの障害対応やシステム監視制御部の開発・追加などが無矛盾であることを維持する実用的なシステムは存在していない。このような Safety Case と対象システムとの無矛盾性は、ほぼ人手で行わなければならない、システムの更新と、Safety Case の更新を無矛盾に行うことは、時間的なずれが生じるなど、問題点が多い。一般に、Safety Case のような、認証機関などを含む、広範囲のステークホルダが理解可能な形式で記述されたドキュメントと、システムの実際の動作との間の無矛盾性を保証する実用的なシステムは、発明者が知る範囲においては無い。

40

【 0 0 2 0 】

本発明は、上記の問題点に鑑みてなされたものであり、その目的は、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援することを可能にすると共に、対象システムの仕様書群と当該システムの実装を無矛盾に維持することを容易にするディペンダビリティ維持装置等を実現することにある。なお、本発明では、システムのディペンダビリティを非特許文献 3 などの定義に、複数のステークホルダ間での合意を含んだものに拡張する。

【 課題を解決するための手段 】

50

【0021】

上記課題を解決するために、本発明に係るディペンダビリティ維持装置は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置であって、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得手段と、上記記述データ取得手段が取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定手段と、を備えることを特徴としている。

【0022】

また、本発明に係るディペンダビリティ維持装置の制御方法は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置の制御方法であって、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得ステップと、上記記述データ取得ステップにて取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定ステップと、を含むことを特徴としている。

10

【0023】

上記の構成によれば、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データに基づいて、対象システムのディペンダビリティの価値を定量的に示す評価値を求めることができる。なお、上記要求・仕様は、ステークホルダ群によって合意されたものであることが必要がある（少なくとも、合意されたものであることが望ましい）。

20

【0024】

それゆえ、対象システムのディペンダビリティの価値を定量的に表現でき、したがって、例えば、要求変更に伴ってディペンダビリティ記述データを変更するときや、対象システムの運用時に対象システムの状態を確認するとき、対象システムのディペンダビリティの価値を分かりやすく客観的に提示することができる。

【0025】

よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となるという効果を奏する。すなわち、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援することが可能となるという効果を

30

【0026】

また、本発明に係るディペンダビリティ維持システムは、対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクルを実行するワークスペース装置と、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行するランタイム装置とを含み、上記ワークスペース装置および上記ランタイム装置の少なくともいずれか一方が、上記ディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めることを特徴としている。

40

【0027】

なお、上記のディペンダビリティ維持装置は、コンピュータによって実現してもよく、この場合には、コンピュータを上記の各手段として動作させることにより上記ディペンダビリティ維持装置をコンピュータにて実現させるディペンダビリティ維持装置のプログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

【0028】

また、本発明に係るディペンダビリティ維持装置は、監視制御対象システム（以下、本発明では、対象システムが特に監視対象に係る場合には、当該対象システムを監視制御対

50

象システムとも呼ぶ)の状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置であって、上記監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出し、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成するディペンダビリティ記述変換部を備えたことを特徴としている。

【0029】

また、本発明に係るディペンダビリティ維持装置の制御方法は、監視制御対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置の制御方法であって、上記監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出す読み出しステップと、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成する変換ステップとを含むことを特徴としている。

10

【0030】

上記の構成によれば、ディペンダビリティ維持装置がディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから障害監視制御用データを生成する。そして、これに従って障害監視制御部が動作することにより、障害監視制御部が監視制御対象システムの状態を監視し、必要な場合に対策を実行することができる。

【0031】

ここで、ペンダビリティ記述データは、監視制御対象システムのディペンダビリティに関わる要求・仕様を記述したものである。そして、監視制御対象システムの各ステークホルダが監視制御対象システムのディペンダビリティに関して合意した際、その結果がペンダビリティ記述データとして記述され、ディペンダビリティ記述格納部に格納されたものである。

20

【0032】

このように、障害監視制御部が用いる障害監視制御用データが、ディペンダビリティ記述格納部に格納されたディペンダビリティ記述データから、ディペンダビリティ維持装置によって生成されることにより、ディペンダビリティ記述データと障害監視制御部の動作とが常に無矛盾に維持されることが可能となる。

30

【0033】

また、本発明に係る障害対応システムは、上記記載のディペンダビリティ維持装置と、上記ディペンダビリティ記述格納部と、上記障害監視制御部とを含み、上記ディペンダビリティ維持装置が上記ディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから生成した障害監視制御用データに従って、上記障害監視制御部が動作することにより、上記監視制御対象システムの状態を監視し、必要な場合に対策を実行するように構成してもよい。

【0034】

なお、上記のディペンダビリティ維持装置および障害対応システムは、コンピュータによって実現してもよく、この場合には、コンピュータを上記ディペンダビリティ記述変換部として動作させることにより上記ディペンダビリティ維持装置をコンピュータにて実現させるディペンダビリティ維持装置のプログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

40

【発明の効果】

【0035】

以上のように、本発明に係るディペンダビリティ維持装置は、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得手段と、上記記述データ取得手段が取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定手段と、を備える構成である。

50

【0036】

また、本発明に係るディペンダビリティ維持装置の制御方法は、対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得ステップと、上記記述データ取得ステップにて取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定ステップと、を含む方法である。

【0037】

それゆえ、対象システムのディペンダビリティの価値を定量的に表現できる。よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となるという効果を奏する。すなわち、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援することが可能となるという効果を奏する。

10

【0038】

また、本発明に係るディペンダビリティ維持装置は、監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出し、読み出したディペンダビリティ記述データから障害監視制御用データを生成するディペンダビリティ記述変換部を備えた構成である。

【0039】

また、本発明に係るディペンダビリティ維持装置の制御方法は、監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出す読み出しステップと、読み出したディペンダビリティ記述データから障害監視制御用データを生成する変換ステップとを含む方法である。

20

【0040】

それゆえ、障害監視制御部が用いる障害監視制御用データが、ディペンダビリティ記述格納部に格納されたディペンダビリティ記述データから、ディペンダビリティ維持装置によって生成されることにより、ディペンダビリティ記述データと障害監視制御部の動作とが常に無矛盾になるように維持することが可能となるという効果を奏する。

【0041】

本発明のさらに他の目的、特徴、および優れた点は、以下に示す記載によって十分に分かるであろう。また、本発明の利点は、添付図面を参照した次の説明で明白になるであろう。

30

【図面の簡単な説明】

【0042】

【図1】本発明の一実施形態を示すものであり、ディペンダビリティ維持システムの構成の概略を示す機能ブロック図である。

【図2】図1に示したディペンダビリティ維持システムのワークスペースコンピュータおよびランタイムコンピュータのハードウェア構成を示すブロック図である。

【図3】図1に示したディペンダビリティ維持システムで用いるソフトウェアの構成例を示すブロック図である。

【図4】図1に示したディペンダビリティ維持システムで処理するアプリケーションの一例を示す説明図であり、(a)はアプリケーションが3層構造モデルをしていることを示し、(b)は3層構造モデルを実装する際の一構成例を示す。

40

【図5】図1に示したディペンダビリティ維持システムで処理するアプリケーションのディペンダビリティ記述データの一例(正常系)を示す説明図である。

【図6】図1に示したディペンダビリティ維持システムで処理するアプリケーションのディペンダビリティ記述データの一例(リスクベース)を示す説明図である。

【図7】図6に示したディペンダビリティ記述データと、ステークホルダ要求変更に伴って変化したディペンダビリティ記述データとの差分を示すディペンダビリティ記述データの記述例を示す説明図である。

【図8】図1に示したディペンダビリティ維持システムおよび図4に示した障害対応シ

50

システムにおいて、ディペンダビリティ記述データとして用いる、D-Caseの基本構造を示す説明図である。

【図9】図5に示したディペンダビリティ記述データの一部のD-Case記述による表現を示す説明図である。

【図10】図1に示したディペンダビリティ維持システムにおいて、ディペンダビリティ記述データが表現するシステムのディペンダビリティの価値を計測、評価する手法を示す説明図であり、(a)は各エッジを要素とする多次元ベクトル値をD値とする手法を示し、(b)は有効エビデンス/総エビデンスをD値とする手法を示し、(c)はモニタノード群のグラフ構造そのものをD値とする手法を示す。

【図11】非特許文献5に記載の手法を示す説明図である。

【図12】図1に示したディペンダビリティ維持システムのD値計算部の構成例を示す説明図である。

【図13】図1に示したディペンダビリティ維持システムが処理する、ステークホルダ群により合意されたD-Case記述に電子署名を付与する例を示す説明図である。

【図14】図1に示したディペンダビリティ維持システムが処理する、階層構造を有するモニタノードの例を示す説明図である。

【図15】図1に示したディペンダビリティ維持システムが有する、変化対応サイクルおよび障害対応サイクルの概略を示す説明図である。

【図16】図15に示したディペンダビリティ維持システムが有する反復的プロセスを実現するためのアーキテクチャを示すブロック図である。

【図17】図1に示したディペンダビリティ維持システムが有する、変化対応サイクルおよび障害対応サイクルの一連の手順を示すフローチャートである。

【図18】図17に示した変化対応サイクルにおける3つの処理を示すフローチャートである。

【図19】図17に示した障害対応サイクルのうち、障害発生検出時の2つの処理を示すフローチャートである。

【図20】図19に示した障害発生検出処理の一例を示す図であり、(a)はディペンダビリティ記述データの一部を示し、(b)はモニタノードを用いて障害検出するためのディペンダビリティ維持システムの構成例を示すブロック図である。

【図21】図20に示した障害発生検出処理において使用する、モニタノードと監視モジュールとの対応関係を定義するテーブルの一例を示す説明図である。

【図22】図19に示した障害発生検出処理において使用するスクリプトの構成概要を示す説明図である。

【図23】図17に示した障害対応サイクルのうち、障害予兆検出時の2つの処理を示すフローチャートである。

【図24】図19に示した障害発生検出処理において使用するスクリプトの一例を示す説明図である。

【図25】図1に示したディペンダビリティ維持システムのランタイムコンピュータのソフトウェア階層の一例を示す説明図である。

【図26】図1に示したディペンダビリティ維持システムのランタイムコンピュータの隔離部が実現する各隔離項目に関する機能要件の一例をまとめたテーブルである。

【図27】図1に示したディペンダビリティ維持システムにおけるディペンダビリティ記述データの変更抽出処理の手順を示すフローチャートである。

【図28】図1に示したディペンダビリティ維持システムにおける、ディペンダビリティ記述データを介したワークスペースコンピュータとランタイムコンピュータとの連携を示す機能ブロック図である。

【図29】図5に示したディペンダビリティ記述データを、計算機表現の一例としてXMLで記述したリストの一部である。

【図30】図1に示したディペンダビリティ維持システムの構成例であって、ディペンダビリティ記述データデータベースと、ワークスペースコンピュータおよびランタイムコン

10

20

30

40

50

コンピュータとの関係を示す説明図である。

【図3 1】図1に示したディペンダビリティ維持システムのランタイムコンピュータにおけるコマンド実行の処理手順を示すフローチャートである。

【図3 2】図1に示したディペンダビリティ維持システムにおけるディペンダビリティ記述データに関連付けられたプログラムの処理内容の一例を示すフローチャートである。

【図3 3】図1に示したディペンダビリティ維持システムにおけるディペンダビリティ記述データに関連付けられたプログラムの処理内容の他の例を示すフローチャートである。

【図3 4】図1に示したディペンダビリティ維持システムのワークスペースコンピュータが備えるツール群の各機能と、ディペンダビリティ記述データとの関係を示す説明図である。

10

【図3 5】図1に示したディペンダビリティ維持システムにおける、ベンチマーク機能とD-Case Editorとの連携を示す説明図である。

【図3 6】図1に示したディペンダビリティ維持システムを2つ接続した構成例を示すブロック図である。

【図3 7】図3 6に示した2つのディペンダビリティ維持システムを接続した構成における、独立した2つのディペンダビリティ維持システム間での連携の一例を示す説明図である。

【図3 8】図1に示したディペンダビリティ維持システムを2つ、本体側および部品側として統合した構成例を示すブロック図である。

20

【図3 9】図3 8に示した2つのディペンダビリティ維持システムを本体側および部品側として統合した構成における、独立した2つのディペンダビリティ維持システム間での連携の一例を示す説明図である。

【図4 0】図3 8に示した2つのディペンダビリティ維持システムを本体側および部品側として統合した構成における、本体に部品を統合する処理手順を示すフローチャートである。

【図4 1】図1に示したディペンダビリティ維持システムのワークスペースコンピュータによる、ディペンダビリティ記述データの表示例を示す説明図である。

【図4 2】本発明の他の実施形態を示すものであり、障害対応システムの構成の概略を示す機能ブロック図である。

【図4 3】図4 2に示した障害対応システムにおいて、ディペンダビリティ記述データとして用いる、D-Caseの具体例を示す説明図である。

30

【図4 4】図4 2に示した障害対応システムの処理を示すフローチャートである。

【図4 5】図4 2に示した障害対応システムが用いる、D-Caseパターンを含むD-Caseの具体例を示す説明図である。

【図4 6】図4 2に示した障害対応システムが用いる、D-Caseパターン<=>モジュール対応テーブルの一例であって、(a)はモニタモジュールについての対応テーブルを示し、(b)はアクションモジュールについての対応テーブルを示す。

【図4 7】図4 2に示した障害対応システムが用いる、D-Caseパターンを含むD-Caseの具体例を示す説明図である。

【図4 8】図4 2に示した障害対応システムが用いる、D-Caseパターン<=>モジュール対応テーブルの一例であって、(a)はモニタモジュールについての対応テーブルを示し、(b)はアクションモジュールについての対応テーブルを示す。

40

【図4 9】図4 2に示した障害対応システムが用いる、D-Caseパターン<=>モジュール対応テーブルの一例であって、モニタモジュールについての対応テーブルを示す。

【図5 0】図4 2に示した障害対応システムにおいて、ディペンダビリティ記述データとして用いる、D-Caseの他の具体例を示す説明図(左半分)である。

【図5 1】図4 2に示した障害対応システムにおいて、ディペンダビリティ記述データとして用いる、D-Caseの他の具体例を示す説明図(右半分)である。

【図5 2】図5 0、図5 1に示したD-CaseをXML形式で記述した例を示す説明図であり、一つのサンプルからの抜粋を図5 2～図5 5に分割して示す。

50

【図53】図50、図51に示したD-CaseをXML形式で記述した例を示す説明図であり、一つのサンプルからの抜粋を図52～図55に分割して示す。

【図54】図50、図51に示したD-CaseをXML形式で記述した例を示す説明図であり、一つのサンプルからの抜粋を図52～図55に分割して示す。

【図55】図50、図51に示したD-CaseをXML形式で記述した例を示す説明図であり、一つのサンプルからの抜粋を図52～図55に分割して示す。

【図56】図50、図51、および図52～図55に示したD-Caseから変換した監視制御データ（障害対応スクリプト）の例を示す説明図である。

【発明を実施するための形態】

【0043】

10

〔実施の形態1〕

本実施の形態に係るワークスペースコンピュータ101および/またはランタイムコンピュータ102は、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援するものである。そのため、後述するように、本実施の形態に係るワークスペースコンピュータ101および/またはランタイムコンピュータ102は、対象システムのディペンダビリティに関する仕様を記述したディペンダビリティ記述データ（時間軸上でモデルの変化を可能にした上で対象システムの構造を計算可能なモデルとして表現することを可能にする差分構造モデルで記述されるデータであってもよい。）に基づいて、対象システムのディペンダビリティの価値を定量的に示すD値を求める。

20

【0044】

以下、図1から図41に基づいて、本発明の一実施の形態について詳細に説明する。

【0045】

〔1.ハードウェア構成〕

図1は、本実施の形態に係るディペンダビリティ維持システム100の構成の概略を示す機能ブロック図である。

【0046】

図1に示すように、ディペンダビリティ維持システム100は、ディペンダビリティ維持装置としてのワークスペース（Workspace）コンピュータ（ディペンダビリティ維持装置、ワークスペース装置）101およびランタイム（Runtime）コンピュータ（ディペンダビリティ維持装置、ランタイム装置）102を含んで構成され、それらはネットワーク103で接続される。ワークスペースコンピュータ101で処理されたソフトウェアSWは、ランタイムコンピュータ102に展開されて処理される。また、対象システム、あるいはアプリケーションシステムとその実行に必要なライブラリ、ミドルウェア、システムサービス等のサポートシステムはランタイムコンピュータ102に含まれる。ソフトウェアSWは、バイナリ実行部110（110W、110R；ワークスペースコンピュータ101のバイナリ実行部110Wとランタイムコンピュータ102のバイナリ実行部110Rを区別する場合は、W、Rを付して記す。）で処理可能な表現形式を有している。ソフトウェアSWは対象システムで稼働するアプリケーションとその実行に必要なライブラリ、ミドルウェア、システムサービス等の実行環境を含む。

30

40

【0047】

ワークスペースコンピュータ101は、検証ツール部101-01、編集ツール部101-02、解析ツール部101-03、開発ツール部101-04、合意形成支援ツール部101-05、バイナリ実行部110Wを備えて構成される。なお、機能ブロックは必ずしもこれらに限定されるものではないが、これら代表的な機能について簡単に説明する。

【0048】

検証ツール部101-01は、ソフトウェアSWを検証する。編集ツール部101-02は、ソフトウェアSWを編集する。解析ツール部101-03は、ソフトウェアSWを解析する。その際、ランタイムコンピュータ102からの情報を用いても良い。開発ツ

50

ル部 101-04 は、ソフトウェア SW を開発するために利用される。

【0049】

また、ランタイムコンピュータ 102 は、更新部 102-01、記録部 102-02、モニタリング部 102-03、再構成部（再構成手段）102-04、隔離部 102-05、スクリプト処理部（スクリプト処理手段）102-07、バイナリ実行部 110R を備えて構成される。なお、機能ブロックは必ずしもこれらに限定されるものではないが、これら代表的な機能について簡単に説明する。

【0050】

更新部 102-01 は、ランタイムコンピュータ 102 で処理されているソフトウェア SW を更新する。記録部 102-02 は、ランタイムコンピュータ 102 の内部の状態を記録する。記録に当たっては、ソフトウェア SW の指示に従っても良いし、あらかじめ決められた所定の設定に従って記録しても良い。モニタリング部 102-03 は、ランタイムコンピュータ 102 の内部の状態を取り出すと同時に、後述する D 値を算出する。状態の取得は、ソフトウェア SW の指示に従っても良いし、あらかじめ決められた所定の設定に従って取得しても良い。再構成部 102-04 は、ランタイムコンピュータ 102 の内部の構成を変更する。構成の変更にあたっては、ソフトウェア SW の指示に従っても良いし、あらかじめ決められた所定の設定に従って変更しても良いし、スクリプトの指示に従っても良い。隔離部 102-05 は、ランタイムコンピュータ 102 の内部の一部の構成を切り離して独立させる。スクリプト処理部 102-07 は、後述のディペンダビリティ記述データから導出されるスクリプトを実行する。

10

20

【0051】

ここで、ディペンダビリティ維持システム 100 は、図 1 に示すように、別個の二台のコンピュータで構成されるシステムであっても良いし、1台のコンピュータで構成されるシステムであっても良いし、各々が 2 台あるいはそれ以上の台数のコンピュータから構成されるシステムであっても良い。2 台あるいはそれ以上で構成されるときには、各コンピュータはネットワーク 103 で接続されていけばよい。

【0052】

図 2 に、ワークスペースコンピュータ 101 およびランタイムコンピュータ 102 のハードウェア構成を示す。もっとも基本的な構成としては、ワークスペースコンピュータ 101 およびランタイムコンピュータ 102 は、命令バスとデータバスで接続された演算装置 151 と、制御装置 152 と、メモリ装置 153 と、入出力装置 154 とを備えた電子計算機である。入出力装置 154 から入力されたビットデータの情報に基づき、演算装置 151 において、算術演算、論理演算、比較演算、シフト演算、等が実行される。実行されたデータは必要に応じて、メモリ装置 153 に記憶され、入出力装置 154 から出力される。これら一連の処理は、メモリ装置 153 に記憶されたソフトウェアプログラムに従って、制御装置 152 によって制御される。本実施の形態におけるワークスペースコンピュータ 101 およびランタイムコンピュータ 102 は、上記のコンピュータとしての基本機能を備えたハードウェアであり、オペレーティングシステムやデバイスドライバ、ミドルウェア、アプリケーションソフトウェアといったプログラム群によって制御されている。

30

40

【0053】

図 3 に本実施の形態に係わるソフトウェア SW の構成例を示す。版管理情報 104-01 は、当該ソフトウェア SW に関する版管理に関する情報であり、タイムスタンプを含んでも良い。ディペンダビリティ記述データ 104-02 は、当該ソフトウェア SW に関する後述のディペンダビリティ記述データに関する情報である。バイナリ・コード 104-03 は、当該ソフトウェア SW の処理をバイナリ実行部 110 が解釈実行可能な表現形式で記述した情報である。外部参照情報 104-04 は、当該ソフトウェア SW が必要とする、あるいは依存している外部のソフトウェアに関しての情報である。

【0054】

〔2. ディペンダビリティ記述データ〕

50

図4は、一例としてのアプリケーションを示す説明図である。ここでは、WEBアプリケーションを例に説明する。

【0055】

図4(a)は、当該アプリケーションが3層構造モデルを成していることを示している。プレゼンテーション層(Presentation)200-01は、当該アプリケーションの表示(及び入力)を担当する。データアクセス層(Data access)200-03は、当該アプリケーションが実行を進めるに当たって必要となるデータを格納し管理する。アプリケーションロジック層(Application logic)200-02は、ユーザからの入力の基づき当該アプリケーションとしての必要な計算処理を実行し、結果をデータアクセス層200-02に渡す。あるいは、当該アプリケーションが必要なデータをデータアクセス層200-02から入手し、処理をしてからプレゼンテーション層200-01に渡して利用者に提示する。

10

【0056】

図4(b)は、上記3層構造モデルを実装する際の一構成例を示している。WEB Server 201-02が、上記プレゼンテーション層200-01に相当する。Client 201-01は、当該アプリケーションの利用者であり、ネットワーク201-05経由で、WEB Server 201-02を利用する。Client 201-01、ネットワーク201-05、及びWEB Server 201-02を含めてプレゼンテーション層200-01に相当すると考えても良い。App Logic 201-03は、上記アプリケーションロジック層200-02に相当する。App Logic 201-03は、WEB Server 201-02とは通信路201-06を介して情報の交換を行い、DBS 201-04とは通信路201-07を介して情報の交換を行う。DBS 201-04は、データベースであり、App Logic 201-03の処理した、又は処理に必要なデータの格納、獲得、等の管理を行う。WEB Server 201-02、App Logic 201-03、及びDBS 201-04は、ランタイムコンピュータ102で実行されるが、それらが1台のコンピュータ上で実行されても、それぞれ1台ずつのコンピュータ上で実行されても良い。

20

【0057】

図5は、上記アプリケーションのディペンダビリティ記述データの一例であり、サービス継続の観点からディペンダビリティ記述データを記述している。ディペンダビリティ記述データとは、アプリケーションのディペンダビリティに関するステークホルダ間の合意された要求を記述したモデルである。

30

【0058】

以下に、図5に記載の一例の内容を説明する。「WEBサービスの継続」(202-01)は上記アプリケーションのディペンダビリティに関するゴールであり、WEBサービスが継続できることを意味している。このゴール(202-01)を実現するためには、当該ディペンダビリティ記述データでは、「Clientの動作継続」(202-02)、「WEB Serverの動作継続」(202-03)、「App Logicの動作継続」(202-04)、「DBSの動作継続」(202-05)の4つの特性が満足される必要がある。それぞれ、図4(b)のclient 201-001、WEB Server 201-02、App Logic 201-03、及びDBS 201-04に対応している。

40

【0059】

これら4つの特性は、さらにブレイクダウンされ次のように記述される。「Clientの動作継続」(202-02)のためには、「デバイスの適切な動作」(202-06)と「WEB Serverの適切な反応」(202-10)という2つの特性の充足が必要である。「WEB Serverの動作継続」(202-03)のためには、「リクエストの適切な送受信」(202-07)と「App Logicの適切な反応」(202-11)の2つの特性の充足が必要である。「App Logicの動作継続」(202-04)のためには、「正常な業務処理」(202-08)と「DBSの適切な反応」(202-12)の2つの特性の充足が必要である。「DBSの動作継続」(202-05)のためには、「DBの一貫性」(202-09)と「データの有効性」(202-13)の2つの特性の充足が必要である。

50

【0060】

上記複数の特性の充足は、図5では、楕円形ノードに示された各種の検証を始めとする、実行時のモニタリング結果によって、把握、判断することができる。例えば、「デバイスの適切な動作」(202-06)は包括的なテストによる「デバイス検査の合否」(202-20)によって当該特性が充足されていると判断できる。「WEB Serverの適切な反応」(202-10)や「リクエストの適切な送受信」(202-07)は当該サーバへのアクセスベンチマークを始めとする、実行時のログ検証による「反応検査の合否」(202-21)や「遅延検査の合否」(202-22)によって当該特性が充足されていると判断できる。「App Logicの適切な反応」(202-11)や「正常な業務処理」(202-08)も同様に包括的なテストによる「テストケース合否」(202-23と202-24)によって当該特性が充足されていると判断できる。「DBSの適切な反応」(202-12)、「DBの一貫性」(202-09)、及び「データの有効性」(202-13)の特性もベンチマーク、ストレステスト(過負荷試験)、アノマリーテスト(耐異常試験)による、あるいは実行時のデータベースアクセスログの検証による各々「ベンチマーク合否」(202-25)、「ストレステスト合否」(202-26)、「アノマリーテスト合否」(202-27)によって当該特性が充足されていると判断できる。本実施の形態では、これら楕円形ノードに記述された検証を、対応ノードの特性が充足していることを示す根拠(エビデンス)として扱う。なお、これらのエビデンスは、後述するモニタノードによる計測値も利用される。

10

20

【0061】

図6は、図5がシステムの正常系に関してのディペンダビリティ記述データであったのに対し、リスクを考慮したシナリオベースのディペンダビリティ記述データの例である。すなわち、「・・・が発生したら、・・・の対応が取れるか？」というシナリオを考えることで、ディペンダビリティ記述データを記述している。例えば、図6の例では、ノードのトップは「DBのHDDの容量が限界(データベースシステムを構成するハードディスク容量に空きがなくなったらどのような対応が取れるか?)」(203-01)という、リスクが記述されている。このリスクが実際に発現しないための、4つの対策を次レベルのノードに記述している。「HDD容量の拡張」(203-02)はハードディスクの追加やより容量の大きなディスクドライブへの置き換え等の手段によって容量が拡張可能であることを記述している。「バックアップDBSによる継続」(203-03)はデータベースシステムにバックアップ系が用意されており、それを用いてデータベース機能が継続可能であることを記述している。「キャッシュによるApp Logicの継続」(203-04)はデータベースが利用できない状況でも一部キャッシュされたデータを用いてアプリケーションが継続可能であることを記述している。「丁寧なエラーを利用者に返す」(203-05)はアプリケーションの実行を続けられなくなっても何らかの説明を利用者に通知可能であることを記述している。

30

【0062】

それらのノード(203-02~203-05)は、その可能性を裏付ける機能をそれらのサブノードで記述している。例えば、「HDD容量の拡張」(203-02)は「HDDデバイス活線挿抜(ホットスワップ、すなわち停止させることなくディスクドライブを交換できること)」(203-06)機能、あるいは「論理ボリュームのOSサポート(ボリューム容量はディスクドライブの容量に制限されないシステム機能)」(203-07)機能によって実現可能になる。「バックアップDBSによる継続」(203-03)は「スタンバイ機再起動(データベースシステムを複数用意しておき1つの系統が停止したときに待機していたシステムを起動させて機能を引き継ぐ)」(203-08)機能、あるいは「2重系の片方のみでの継続(2系統のデータベースシステムで実現した機能をその1系統が停止しても継続できる)」(203-09)機能によって実現可能になる。「キャッシュによるApp Logicの継続」(203-04)は「DBへの変更の遅延書き込み(データベースへのアクセスをそれが復旧するまで遅らせる)」(203-10)機能、あるいは「データのタイムスタンプ(タイムスタンプを参照することで古いデータでもApp Logic

40

50

を継続できる)」(203-11)機能によって実現可能になる。最後の「丁寧なエラーを利用者に返す」(203-05)は「ネットワークが到達可能(利用者までのネットワーク通信が可能)」(203-12)によって実現可能になる。

【0063】

なお、これらエッジノード(203-06~203-12)は、図5における楕円形ノード(エビデンス)によってそれらの特性の充足が記述されるが、図6では省略している。

【0064】

上記図5に記載のディペンダビリティ記述データと図6に記載のディペンダビリティ記述データは、どちらかをステークホルダ間で合意されたディペンダビリティ記述データとして用いても良い。また、より上位のスコープのディペンダビリティ記述データの一部として、図5は正常系、図6はリスク系として、共にステークホルダ間で合意されたディペンダビリティ記述データとして用いても良い。

【0065】

図7は、ステークホルダ要求変更に伴って変化したディペンダビリティ記述データと、図6に示したディペンダビリティ記述データとの差分を示すディペンダビリティ記述データの記述例を示す説明図である。

【0066】

図6では、「HDD容量の拡張」(203-02)の可能性が当該アプリケーションの継続には重要であった。これに対して、図7では、何らかの環境の変化、例えば、ビジネスの好転によって、ステークホルダがデータベースシステムへの投資額を増やし強化する決定をした場合における、図6とは異なった時刻におけるディペンダビリティ記述データを記述している。すなわち、図6の「バックアップDBSによる継続」(203-03)の代わりに、「多重系によるDBSの無停止」(204-01)機能を採用することを決定した場合を記述している。データベースが無停止であることによって、図6の「キャッシュによるApp Logicの継続」(203-04)は不要になっている。その結果、図7の網かけされたノードが図6との差分となる。

【0067】

図5から図7に示したディペンダビリティ記述データは、Safety Case表現(非特許文献2)を用いて記述しても良いし、図8、図9に示すように、D-Case表現を用いて記述しても良い。

【0068】

ここで、図8を用いて、D-Case表現の基本構造を説明する。対象システムのディペンダビリティに関するステークホルダ群からの要求をトップゴール210-01として記述する。トップゴールとは、対象システムに対してステークホルダ間で合意すべき命題を表す。例えば、「対象システムは機能安全規格IEC61508で定義されているSafety Integrity Level 3を満たす」などである。当該トップゴールが満たされていることを木構造により詳細化していき、詳細化されたサブゴール210-05に対しても同様に木構造により詳細化していく。サブゴールとは、トップゴールを示すために、示すべき事を分割した命題を表す。サブゴールは、さらに小さなサブゴールに分割される。本実施の形態では、トップゴール及びサブゴールを総称してゴールと呼ぶ。当該詳細化の過程は「議論の構造」210-02と呼ばれる。

【0069】

当該ゴールをサブゴールに分割する際に、当該分割の補助説明(理由・理屈)を示すストラテジを記述しても良い。例えば、図5に記載のディペンダビリティ記述データの一部をD-Case記述で表現すると図9になる。ゴールが「WEBサービスの継続」(202-01)であり、当該ゴールにリンクしている4項目である「Clientの動作継続」(202-02)、「WEB Serverの動作継続」(202-03)、「App Logicの動作継続」(202-04)、「DBSの動作継続」(202-05)は各々サブゴールであり、それら4項目をサブゴールとする理由がストラテジ「サブシステムの継続性の観点で分割」(211

10

20

30

40

50

- 01)である。

【0070】

また、ゴールあるいはストラテジにはコンテキストがリンクしても良い。コンテキストはゴールやストラテジの内容を補足する情報である。例えば、図9では、ゴール202-01に対してコンテキストとして当該ゴール「WEBサービスの継続」において「継続」に関する補足説明である「サービスが停止する場合でも1分以内で回復すること」が付与されている。

【0071】

ゴールには、それが妥当であることを表明しているエビデンス210-03、210-04が存在する。エビデンスは分割され詳細化されたゴールを最終的に保証する。エビデンスの妥当性はステークホルダ間の合意に基づく。エビデンスの存在していないゴールは何らかの問題を内在している。

10

【0072】

また、図5から図9では、ディペンダビリティ記述データを木構造で表現した。しかし、ノード間の依存関係を表現し一般的なグラフ構造で表現しても良いし、あるいは表形式で表現しても良い。以下、本実施の形態では、ディペンダビリティ記述データとして前記D-Case表現を用いて説明する。

【0073】

本実施の形態では、ディペンダビリティ記述データから計算されるD値を用いてディペンダビリティの価値を計測、評価する。D値の計算は、ワークスペースコンピュータ101においては、検証ツール部101-01のD値計算部（ディペンダビリティ値決定手段）101-05が、ランタイムコンピュータ102においては、モニタリング部102-03のD値計算部（ディペンダビリティ値決定手段）102-06が行う。D値は、ワークスペースコンピュータ101においては、例えば、ディペンダビリティ記述データの検証のために用いられ、ディペンダビリティ記述データが変更される度に計算される。また、ランタイムコンピュータ102においては、例えば、ディペンダビリティ記述データに従って動作する対象システムの監視のために用いられ、対象システムの動作中はリアルタイムに計算される。

20

【0074】

図10にディペンダビリティ記述データが表現するシステムのディペンダビリティの価値を計測、評価する手法を示す。ディペンダビリティ維持システム100にはディペンダビリティ記述データが複数存在する。トップノードは特定の特性に対するディペンダビリティの価値と定義できる。例えば、図5の「WEBサービスの継続」（202-01）は失われると困るものであり資産（asset）と考えられるので、本実施の形態では「ディペンダビリティの価値」と定義し、「D値（評価値）」と呼ぶことにする。

30

【0075】

トップノードは複数のサブノードに分解されるので、各エッジ（リンク）を要素とする多次元ベクトルが構成できる（図10（a））。その際に、エッジに重みを付けても良い。図5を例にするなら、例えば、ノード202-05をノード202-04よりも重要である場合、202-01と202-05のエッジの重みを、202-01と202-04のエッジより大きくしても良い。また、あるノードのサブノードの数を重みとしても良い。サブノードが多いと言うことは十分に対策が検討されていることにもなるからである。このように重み付けされたエッジの多次元ベクトル値をD値とすることができる。次元数が多くなると多次元ベクトルは扱いにくいので、主成分分析法やフィッシャー判別分析法等を用いて次元削減を行っても良い。また、設計時におけるエビデンスを教師データとして学習し、後述のモニタノードによる変化を、例えばマハラノビス距離計算によって外れ値として検出することで異常の発生検知に用いても良い。

40

【0076】

こうすることで、グラフ構造にエッジの重要性を加味した評価値が得られる。ここで得られるD値は多次元ベクトル値であるが、目的に応じた任意の変換手法によりスカラー値

50

に変換できることはいうまでもない。

【0077】

また、図10(b)は、当該特性において、有効エビデンス/総エビデンスをD値とした例である。図5では総エビデンス数は8であるが、例えば、その中で有効数が4であるときにはD値は $4/8=0.5$ となる。

【0078】

また、図10(c)は、モニタノード群におけるモニタノードでの監視点が取得したデータが変動許容範囲内であるノードと当該変動許容範囲外のノードとを色分けし、当該グラフ構造そのものをD値とする例である。なお、詳細については後述する。

【0079】

こうすることで、エビデンスの状態に応じた評価値が得られる。エビデンスは要求変更に応じて変化するため、ここで得られるD値も要求変更に応じて変化する。すなわち、D値を確認することで、要求変更がディペンダビリティに与える影響を評価することができる。

【0080】

また、非特許文献5の手法を用いても良い。当該文献ではD-Case記述を図11に記載の項目で再構成している。図11に示すように、「フェーズ」、「目的」、「ターゲット」、「異常」の種類の4層にD-Caseノードを再配置することで、エッジの重みを各層で一貫性のある値として扱うことが可能になり、D値を各層で比較することも可能になる。

【0081】

ここで、システムに求められるディペンダビリティ要件は、そのシステムの目的により異なる。また、システムのライフサイクルの各フェーズで、ステークホルダ間の合意のための作業を円滑化する必要があるが、そのためには、ディペンダビリティ要件やその実現度合いを定量的に表現する必要がある。そこで、上記のように、D値としてディペンダビリティを定量的に表現することにより、議論するための指標(ディペンダビリティメトリクス)が実現できる。

【0082】

上記のように、D値を用いることで、システム開発者は、そのシステムにおける特定のディペンダビリティ支援に関する重みを考慮して、要求そのもの、あるいは要求の実現度合いを定量化できる。また、システムの運用時には、そのシステムに不具合が発生した場合、その時点で満足し得るディペンダビリティをリアルタイムに定量化することができる。すなわち、ベンチマークや検証により得られるエビデンスとD値とを組み合わせ、ディペンダビリティ記述データを評価することが可能となる。また、動作中システムからリアルタイムにD値を得て、システムの状況を判断することが可能となる。

【0083】

ここで、図12を用いて、ランタイムコンピュータ102における、モニタリング部102-03のD値計算部102-06の一実施例を説明する。D値を当該ランタイムコンピュータ102においてステークホルダ合意に基づいて実行時に計算するために、図8に記載のD-Caseに対象システム内部のどの機能を、いつ、どのように監視するかをステークホルダ間での合意の上で記述するD-Caseモニタノードを導入する(図12)。D-Case記述におけるゴールを表現するノードには、当該ゴールが満足しているエビデンスを表明するノードを関連付けることができる。この際のエビデンスに相当するノードに対象システム内部の監視点からのデータを収集し、当該データが変動許容範囲にあるかどうかを判断させるように構成できる。

【0084】

変動許容範囲とは、監視点から得られたデータが基準値であるか、基準値に準じているかの範囲を示している。例えば、ネットワーク帯域の場合には、1Mbps~2Mbpsの範囲にあれば通常の動作であると見なすことができる範囲と定義できる。アプリケーションのメモリ消費であれば、100MB以内の消費であれば通常の動作であると見なすことができる範囲と定義できる。なお、当該変動許容範囲はステークホルダ合意に基づいて

10

20

30

40

50

設定されている。

【0085】

図12において、ゴール220-01に対してはモニタノード220-03が、ゴール220-02に対してはモニタノード220-04が、それぞれリンクしている。モニタノード220-04は実行環境220-07における監視点1(220-05)からのデータを取得し、当該データが変動許容範囲にあるときにはゴール220-02を満足していることを意味している。また、モニタノード220-03は実行環境220-07における監視点2(220-06)からのデータを取得し、当該データが変動許容範囲にあるときにはゴール220-03が満足していることを意味している。

【0086】

ここで、監視点1(220-05)の取得するデータはアプリケーション220-10内部の監視対象センサ220-08からデータを取得している。当該センサには、例えば、当該アプリケーションの消費しているメモリ量であったり、当該アプリケーションが外部と通信している遅延や帯域幅であったりする。一方、監視点2(220-06)の取得するデータはアプリケーション220-10内部の開始対象ログ220-09から取得している。一般に、ログから対象システムの監視対象が変動許容範囲内にあることを判断するには、ログ内部を精査する必要がある。そこで、本実施例では一例として、スクリプトを導入し、ログを精査して監視点対象が変動許容範囲にあることを判断するように構成する。なお、当該スクリプトもステークホルダ合意に基づいている。

【0087】

本発明の一実施例として、ステークホルダによる当該合意を電子署名によって表現し、実行時に確認するように構成しても良い。図13にその一例を示す。図12におけるD-Case記述の概略は図13のように描けるが、ここでは3つの電子署名の有効範囲を示している。範囲211-01は全体に対する電子署名であり、範囲221-02は1組のゴールとモニタノードに対する電子署名であり、範囲221-03は1組のゴールとモニタノードとスクリプトに対する電子署名である。

【0088】

前記電子署名は、その有効期限を定めるように構成可能である。例えば、開発時にのみ有効な電子署名をD-Case記述に付与したり、状況に依存した期間で有効な電子署名をD-Case記述に付与しても良い。これにより、モニタノードによるデータ取得の信頼性を高め、エビデンスとしての利用をより確実にできる。

【0089】

また、電子署名のあるD-Case記述と電子署名のないD-Case記述とが混在する環境においては、電子署名のないD-Case記述を処理する場合にはデフォルトの動作を決めておくように構成できる。一般的にデフォルトの動作は、電子署名が無効である場合と同様の拒絶の処理をすべきである。しかし、状況によってはオペレータやユーザに確認を求めるように構成しても良い。

【0090】

前記D-Caseモニタノードにて取得されるデータをD値計算に用いても良い。図10(c)に一例を示す。本実施例ではD-Case記述はモジュール毎に存在するように構成できる。当該D-Case記述そのものは木構造であり、その一部であるD-Caseモニタノードも木構造を構成するが、対象システム全体のD-Caseモニタノードの構造は図10(c)のような一般化されたグラフ構造を用いて構成できる。ここでは、ノード間の距離を評価するように構成しても良い。ノード群同士の直接リンクをノード間に複数のリンクがある構造よりも評価値を大きくできる。また、当該モニタノード群におけるモニタノードでの監視点が取得したデータが変動許容範囲内であるノードと当該変動許容範囲外のノードとを色分けし、当該グラフ構造そのものをD値として構成しても良い。当該色分けに際して、変動許容範囲に深刻度(Severity)を定義しても良い。例えば、ハードディスクの残容量に関する変動許容範囲の場合には、残容量50%と残容量10%とは異なった深刻度を与えた方が望ましい。当該グラフ構造にはモジュール間の依存関係を反映しているので、グラフ構造を

10

20

30

40

50

D値とすることは、当該依存関係を考慮したD値をディペンダビリティの価値として扱うことができる。

【0091】

さらに、図14示すように、エッジのゴール群(220-04乃至222-07)以外に、途中のゴール群(222-01乃至222-03)に前記モニタノードをリンクしても良い(223-01乃至223-03)。あるゴールにリンクされたモニターノードと当該ゴールのサブノード群にリンクされたモニターノード群との関係は、対応する監視点群の関係に相当する。例えば、ゴール2(222-02)が「商品購入サイトのサービス継続」である場合、そのサブゴールを2個に分割し、ゴール4(222-04)が「電子決済システムの継続」であり、ゴール5(222-05)が「商品DBの継続」であるように構成できる。当該ゴール群に対して、各々次のモニタノードをリンクする。モニタノード2(223-02)はゴール2(222-02)にリンクされ、例えば、スクリプトによって包括的なサービス継続シナリオを実行することで、当該ゴール2が満足していることを確認する。モニタノード4(223-04)はゴール4(222-04)にリンクされ、例えば、スクリプトによって電子決済システムに異常が無いことを確認し当該ゴール4が満足していることを確認する。モニタノード5(223-05)はゴール5(222-05)にリンクされ、例えば、商品DBに具備された死活センサからデータを受領することで当該ゴール5が満足していることを確認する。

10

【0092】

また、次のように構成しても良い。例えば、ゴール3(222-03)が「商品候補表示システムの継続」である場合、そのサブゴールを2個に分割し、ゴール5(222-05)が「規定値アクセス数以下での正常動作」であり、ゴール6(222-06)が「規定値遅延以内での正常動作」であるように構成しても良い。当該ゴール群に対して、各々次のモニタノードをリンクする。モニタノード6(223-06)はゴール6(222-06)にリンクされ、例えば、スクリプトによって規定値アクセス数以下で正常に動作していることを確認することで、当該ゴール6が満足していることを確認する。モニタノード7(223-07)はゴール7(222-07)にリンクされ、例えば、対象システムのWEBサーバに組み込まれたセンサのデータを取得して規定値以下の遅延で正常に動作していることを確認することでゴール7が満足していることを確認する。この場合には、ゴール3(222-03)にリンクされたモニタノード3(223-03)は、サブゴールである当該ゴール6と当該ゴール7が共に満足していることを確認することで、当該ゴール3が満足されていることを確認したことにように構成できる。以上のように、モニタノード群をグラフ構造によって構成することで、上記D値をグラフ構造を計算することで取得し、システムの状況を判断することができる。

20

30

【0093】

なお、本発明は以下のように構成してもよい。

【0094】

本発明のディペンダビリティ維持装置(ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102)は、不完全性と不確実性が潜在的に存在するオープン環境におけるシステム開発及びシステム運用、さらにシステムが提供するサービスにおけるディペンダビリティを持続的に維持する目的で、かつ、ステークホルダ間における要求の誤解、環境の変化に対する対応不能、障害対応の失敗という3つの問題に対応する目的で、ディペンダビリティに関して時間軸上で変化可能なある時刻におけるディペンダビリティ記述データ(データ構造としては、例えば、木構造モデル)を備えるシステムにおいて、当該ディペンダビリティ記述データを計算することができ、それによって当該ディペンダビリティ記述データが表現するシステムのディペンダビリティの価値を計測、評価するものであってもよい。

40

【0095】

このように、前記ディペンダビリティ維持装置は、ディペンダビリティに関して時間軸上で変化可能なある時刻におけるディペンダビリティ記述データを備えるシステムにおい

50

て、ディペンダビリティの価値を計測、評価することができる。したがって、ステークホルダ間における要求の誤解、環境の変化に対する対応不能、障害対応の失敗という3つの問題に対処することでディペンダビリティを維持することが可能となるという効果を奏する。

【0096】

また、本発明は以下のように構成してもよい。

【0097】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102）は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置であって、対象システムのディペンダビリティに関する仕様を記述したディペンダビリティ記述データを取得する記述データ取得手段（ディペンダビリティ記述データ入力部901-01、ソフトウェア入力部902-01）と、上記記述データ取得手段が取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値（D値）を求めるディペンダビリティ値決定手段（D値計算部101-05、D値計算部102-06）と、を備えて構成されても良い。

10

【0098】

これにより、対象システムのディペンダビリティの価値が定量的に表現できる。したがって、例えば、要求変更に伴ってディペンダビリティ記述データを変更するときや、対象システムの運用時に対象システムの状態を確認するときに、対象システムのディペンダビリティの価値を分かりやすく客観的に提示することができる。よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となる。

20

【0099】

さらに、本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102）は、上記ディペンダビリティ記述データがノード間のエッジに重み付けが可能な木構造あるいはグラフ構造を有するデータであり、上記ディペンダビリティ値決定手段（D値計算部101-05、D値計算部102-06）は、上記エッジに重み付けされた重みを要素とする多次元ベクトル値を上記評価値（D値）とするように構成されても良い。

30

【0100】

これにより、木構造あるいはグラフ構造に、エッジの重要性を加味した評価値が得られる。よって、評価値を確認することで、エッジに重み付けの変更がディペンダビリティに与える影響を評価することができる。なお、ここで得られる評価値は多次元ベクトル値であるが、目的に応じた任意の変換手法によりスカラー値に変換できることはいうまでもない。

【0101】

さらに、本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102）は、上記ディペンダビリティ値決定手段（D値計算部101-05、D値計算部102-06）は、上記ディペンダビリティ記述データに含まれるエビデンスの総数である総エビデンス数に対する、上記対象システムをモニタして取得されたモニタ値が変動許容範囲に対して良好であったエビデンスの数である有効エビデンス数の割合を上記評価値（D値）として計算するように構成されても良い。

40

【0102】

これにより、エビデンスの状態に応じた評価値が得られる。エビデンスは要求変更に応じて変化するため、ここで得られる評価値も要求変更に応じて変化する。すなわち、評価値を確認することで、要求変更がディペンダビリティに与える影響を評価することができる。

【0103】

さらに、本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100

50

0、ワークスペースコンピュータ101、ランタイムコンピュータ102)は、上記ディペンダビリティ記述データをグラフ構造として評価し、上記ディペンダビリティ値決定手段(D値計算部101-05、D値計算部102-06)は、当該グラフ構造を計算して、例えばノード間距離や深さを計算式に組み込み評価値(D値)を算出しても良い。

【0104】

これにより、モニタノード群により得られるデータに応じた評価値が得られる。モニタノード群は対象システムの実行時の状況に応じて変化するため、ここで得られる評価値も当該実行状況に応じて変化する。すなわち、当該評価値を確認することで、実行状況がディペンダビリティに与える影響を評価することができる。

【0105】

また、本発明のディペンダビリティ維持システム100は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを仕様の変更に合わせて変更する変化対応サイクルP300を実行するワークスペース装置(ワークスペースコンピュータ101)と、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルP301を実行するランタイム装置(ランタイムコンピュータ102)とを含み、上記ワークスペース装置および上記ランタイム装置の少なくともいずれか一方が、上記ディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値(D値)を求めるように構成されても良い。

【0106】

さらに、本発明のディペンダビリティ維持システム100は、他のディペンダビリティ維持システム100とネットワークを介して接続されていても良い。すなわち、本発明のディペンダビリティ維持システム100は、複数、ネットワークを介して接続されていても良い。

【0107】

{3.二つのプロセス}

図15を用いて、ディペンダビリティ維持システム100が有する反復的プロセスに関して、また、図16を用いて、当該反復的プロセスを実現するためのアーキテクチャに関して説明する。

【0108】

オープンシステムディペンダビリティ、すなわち、機能、構造、システム境界が時間とともに変化し続けるシステムに対するディペンダビリティの実現の為には、反復的プロセスとしてのアプローチが必須である。当該反復的プロセスは、目的(Objectives)や環境の変化に対してシステムを継続的に変更していくためのサイクル(変化対応サイクルP300)と、障害に対して迅速に対応するためのサイクル(障害対応サイクルP301)を備えている必要がある。当該反復的プロセスは、構成要素として要求マネジメントプロセス250-01、開発プロセス250-02、通常運用プロセスS302、障害対応サイクル250-04、説明責任遂行プロセス250-05等を含む「プロセスのプロセス(Process of Processes)」である。当該構成要素プロセスは相互に有機的に結びつけられる必要がある。本実施の形態では当該反復プロセスをDEOSプロセスと呼ぶ。

【0109】

対象システムのディペンダビリティに関する利害関係者をDEOSプロセスにおいては「ステークホルダ」と呼ぶ。ステークホルダとしては、以下を想定している。1)サービス・製品の利用者(顧客、社会的インフラの場合は社会全体)、2)サービス・製品の提供者(事業主)、3)システム提供者(設計開発者、保守運用者、ハードウェア供給者)、4)サービス・製品認可者(規制監督官庁)。

【0110】

ステークホルダは時間の経過や環境の変化によってそれぞれの目的を変化させ、機能やサービスに対する要求を変化させる可能性がある。これらの変化をここでは「目的・環境

10

20

30

40

50

変化」と呼ぶこととする。これらの変化に対し、当該ステークホルダは熟慮し、相互に合意したうえで、適切な時期にシステムの変更を要求する。DEOSプロセスはこのような要求に対するサイクルとして、「変化対応サイクル」(P 3 0 0)を備える。

【0 1 1 1】

対象システムは不完全さと不確実さに起因する障害を完全に回避することがきわめて困難である。障害の予兆を検出した場合には障害を未然に回避し、不幸にも障害が発生してしまった場合には迅速に対応して被害を最小化し、原因を究明し、説明責任を遂行する必要がある。DEOSプロセスではそのような状況に対応するために「障害対応サイクル」(P 3 0 1)を備える。

【0 1 1 2】

新規にシステムを開発し、また、目的・環境の変化に対応してシステムの変更を行う場合、その理由やステークホルダ間で行われた議論の過程、合意内容などを詳細に記録するための合意記述データベース2 5 0 - 0 6を備えていることが効果的な前記反復的プロセスを実現し、説明責任を遂行するために必須である。当該データベースにはディペンダビリティを達成するための議論や根拠を記述した前記D-Case、障害に対してサービスを継続するためのシナリオを基に障害予兆の検出や障害発生に迅速に対応するための実行手続きを記述したスクリプト(詳しくは後述するが図中ではD-Scriptとして記載)が含まれている。そして、これらの合意記述を基に開発プロセス2 5 0 - 0 2、障害対応サイクル2 5 0 - 0 4、通常運用プロセスS 3 0 2が実行され、また、説明責任遂行プロセス5 0 - 0 5を支援することができる。合意記述データベース2 5 0 - 0 6は前記構成要素プロセスを有機的に繋ぐ重要な役割を果たす。

【0 1 1 3】

前記DEOSプロセスの特徴をまとめる。1)「通常運用」(S 3 0 2)から開始される「変化対応サイクル(P 3 0 0)」と「障害対応サイクル(P 3 0 1)」の2つのサイクルから成り立っていること。2)システム変更要求のための「ステークホルダ合意」(2 5 0 - 0 7)と、システム変更や障害対応の「説明責任遂行」(2 5 0 - 0 5)の2つのフェーズが組み入れられていること。3)ステークホルダ間で利害関係を調整し、ディペンダビリティを達成するために議論した過程、論拠、結果等を記述した前記「D-Case」と障害に迅速に対応するための実行手続きを記述した「スクリプト」を含む合意記述データベース2 5 0 - 0 6を備え、構成要素プロセスを有機的に結合すること。

【0 1 1 4】

図15において通常運用S 3 0 2はディペンダビリティ維持システム1 0 0が通常運用状態にあることを示している。当該通常運用S 3 0 2はシステムがステークホルダ間で合意されたサービスレベル変動許容範囲(In-Operation Range)から大幅な逸脱がなく、ユーザに対してサービス提供を継続している状態である。変化対応サイクルP 3 0 0は通常運用と並行して実行され、サービスの提供を継続しつつシステムの変更が行われることが望ましい。同様に、障害対応サイクルP 3 0 1も通常運用を継続しながら実行されることが望ましい。実際、システムが異常の予兆を検知しても、当該スクリプトに記されたサービス・機能レベル変動許容範囲内で自動的に回避処理が働いてサービスが継続される場合がある。あるいは一部の機能を縮退してサービスを継続している場合もある。しかしながらサービスの提供が完全に停止されてしまう場合もある。

【0 1 1 5】

通常運用状態S 3 0 2において実行されるプロセスには、日常的な動作記録の点検、プロセスの定期的な見直し・改善、要員の訓練・しつけ・教育など、継続的なディペンダビリティ向上活動が含まれる。システムの稼働状況を記録し、日々点検する事により保守担当者や運用担当者が、何かの兆候をそこから見出す事ができる可能性がある。また、システムのメモリー資源を常にクリーンな状態にすることも、非常に有効な日常保守・改善活動である。あるいは、積極的に予行を行うことも有効である。障害はある時間が経過してある状態に達した時発生する。であれば、時間を先に経過させると障害の発生を事前に知ることができる。いわゆるリハーサルである。情報システムの提供するサービスの運用時

10

20

30

40

50

において、どの程度適切なりハーサルができるのかはその時の状況による。

【0116】

障害対応サイクルP301に関して概説する。障害対応サイクルP301は障害に対して迅速に対応して障害による被害を最小化するためのサイクルである。DEOSプロセスでは「障害」をステークホルダ間で合意されたサービス・機能レベル変動許容範囲から逸脱する事象と定義する。障害対応サイクルにおける主要なフェーズは、「未然回避」(250-10)、「迅速対応」(250-09)、「原因究明」(250-08)であり、障害が発生した場合は「説明責任遂行」が必須である。当該3フェーズはそれぞれ別個に、かつ順番に行われるとは限らない。多くの場合、これらはお互いが関連しあい、渾然一体となった事象・活動となる。

10

【0117】

未然回避フェーズ250-10は、システムの運用中に障害が発生する前に障害発生を予知したり、あるいは障害が起きる可能性の増大を検出すると、障害を回避するように対応・動作するフェーズである。障害の予知が障害の発生予想時刻の十分に前であれば効果的な対策が打てる。例えばシステムの資源を制限してスループットを下げてシステムダウンを回避したりシステムダウンまでの時間を稼いだりすることが行われる。直前に予知した場合には障害の影響の最小化に努力することになる。また、原因解析に有効な、障害に至るまでのシステムの内部情報を記録することができる。予知のための具体的な方法としては、過去の障害パターンから類似の障害を判別することなどがある。未然回避シナリオはスクリプトとして事前に記述され、オペレータやシステム管理者と協調して未然回避動作が実行される。

20

【0118】

迅速対応フェーズ250-09は、障害が起きた時にその影響を最小化するためのフェーズである。障害に対する迅速対応のシナリオはスクリプトとして事前に記述されており、自動的に行われるのが望ましい。しかしながら、想定しない障害に対応しなければならない場面もある。対応分野や領域ごとの目的に応じたサービス継続のための緊急対応計画(責任者や対応組織、手順、エスカレーションパスなどが記されている)を事前に立てて、ステークホルダ間で合意しておくことが求められる。当該計画の指示に基づきオペレータやシステム管理者と協調して迅速に障害による影響を最小化することになる。すなわち、障害を分離して影響の局所化を行い、サービス全体のダウンを回避する。そのために障害が発生したアプリケーションやシステムの一部のオペレーションを中断し、リセットし、その後オペレータやシステム管理者による復帰活動が行われる。

30

【0119】

原因究明フェーズ250-08は、障害対応サイクルP301と変化対応サイクルP300に関連したフェーズである。サイクルにより深さの違う判断がなされる。障害対応サイクルP301では、どのような迅速対応が可能であるかを見極めることを目的とした原因究明がおこなわれる。その結果によっては変化対応サイクルP300が開始される。

【0120】

説明責任遂行フェーズ250-05では、サービス提供者、特に社会インフラサービス提供者や社会に広く使われる製品提供者が、障害発生時にサービス利用者、製品使用者、社会に対し、障害状況、迅速対応、今後の見通しなど説明する。これは利用者や社会からの信頼を維持し、インフラサービス提供上のコンセンサスを醸成し、ひいてはサービス提供者のビジネス遂行上の便益を守るという大変重要な役目を持つ。合意記述データベース特にD-Case記述と、システム監視記録が説明責任遂行に大いに役立つ。

40

【0121】

変化対応サイクルP300に関して概説する。変化対応サイクルP300はステークホルダの目的の変化や、各種外部環境の変化に対応するためのサイクルである。このサイクルにおける主要なフェーズは、システム変更のための「要求抽出・リスク分析」(250-11)、「ステークホルダ合意」(250-07)、「設計・実装・検証・テスト」(250-02)である。大きな変化に対応する場合は「説明責任遂行」(250-05)

50

が必須となる。障害対応サイクルP301における原因究明フェーズ250-08の実行の結果、システムの根本的な改良の要求が発生した場合も、変化対応サイクルP300が開始される。

【0122】

要求抽出・リスク分析フェーズ250-11は目的や環境の変化によりステークホルダからの要求が変化（新規の要求も含む）した場合、あるいは障害発生に迅速に対応した後、原因究明を行った結果、システムを変更する必要がある場合が始まる。いずれの場合も、事業主のサービス目的をベースにユーザの要求、システム環境、関連する法律や国際標準を勘案し、システムの機能要件を抽出する。また同時に、サービス目的からシステムのサービス継続シナリオを作成してリスク分析を行い、ディペンダビリティ要件を含む非機能要件を抽出する。

10

【0123】

ステークホルダ合意フェーズ250-07では、何をどのように変更するのかを、すべてのステークホルダに分かりやすく、誤解のないように記述し、ステークホルダ間の議論を経て、合意をD-Caseとして記述する。またサービス継続シナリオを作成し、その実行手続きであるスクリプトを作成する。要求抽出・リスク分析フェーズ250-11と当該ステークホルダ合意フェーズ250-07が「要求マネジメントプロセス」（250-01）を構成する。

【0124】

設計・実証・検証・テストフェーズ250-02は、いわゆる設計開発のプロセスである。ここでは、これまで多くの研究がなされ、多くの手法やツールが出されている。

20

【0125】

説明責任遂行フェーズ250-05では、目的や環境変化によるステークホルダの要求変化を満たすためにシステムを変更した場合、その経緯と、いつからどのようにサービスや機能がよくなるのか（変化するのか）を説明する。また、日常のサービス遂行状況や設計開発・保守運用プロセスに関する説明が必要なときもこれに対応する。これは利用者や社会からの信頼を維持し、インフラサービス提供上のコンセンサスを醸成し、ひいてはサービス提供者のビジネス遂行上の便益を守るという大変重要な役目を持つ。合意記述データベース250-06に記録されている特にD-Case記述が説明責任遂行に大いに役立つ。

【0126】

30

〔4. DEOSアーキテクチャ〕

前記DEOSプロセスは広範なオープンシステムに対するディペンダビリティを実現するための反復的プロセスを提供している。このプロセスをより具体的に対象とするシステムに適用した場合、対象のカテゴリ毎にプロセス実行のためのアーキテクチャを考える必要がある。本実施例では、組込みシステムを含む現代の大規模かつ複雑なソフトウェアシステムへの適応を念頭に考案されたアーキテクチャ（DEOSアーキテクチャと呼ぶ）について述べる。前記DEOSプロセスと当該DEOSアーキテクチャを並べて眺めるとDEOSプロセスが実際のシステムでどのように実行されるかが理解できる。

【0127】

図16を用いて、当該DEOSアーキテクチャに関して説明する。当該アーキテクチャは次の構成要素から構成される。1) 要求抽出・リスク分析フェーズを支援するツール群260-01(101-05)、2) ステークホルダ合意フェーズを支援する合意形成支援ツール群260-02(101-05)、3) 合意の記述であるD-Caseとサービス継続シナリオの実行手続きであるスクリプト(図中ではD-Scriptとして記載)を含む合意記述データベース250-06、4) DEOS実行環境260-03、5) プログラム検証ツールとベンチマーキングならびにフォールトインジェクションテストのためのツール群を含むDEOS開発支援ツール260-04。

40

【0128】

要求抽出・リスク分析フェーズ250-11は事業主のサービス目的260-05をベースにユーザの要求、システム環境、関連する法律や国際標準を勘案し、システムの機能要件

50

を抽出し、想定される障害に対するサービス継続シナリオを作成してリスク分析を行い、ディペンダビリティ要件を含む非機能要件を抽出する。

【0129】

ステークホルダ合意フェーズ250-07は合意を形成するための方法と合意記述の記法に基づいて合意内容をD-Caseとして記述する。そのためのツールがD-Case Editor (260-06)ならびにD-Case Viewer (260-07)である。また、サービス継続シナリオに基づいた実行手続きスクリプト260-08も作成される。当該スクリプトはDEOSアーキテクチャにおいてD-Case記述とアプリケーションプログラムの実行を動的に結合する役割を果たしている。当該スクリプトには後述のスクリプトエンジン(図中ではD-Script Engineとして記載)が実行するシナリオが書かれている。そのシナリオはDEOS実行環境260-03に対して、1)いつ、どのようなログ情報を収集するかを指示し、また、2)障害発生時には障害に対してどのように対処するかを指示している。この時、エスカレーションルールに従ってオペレータの介入を指示する場合もある。このように、当該スクリプトは動的かつ双方向に情報を交換することによりアプリケーションプログラムの実行を柔軟に制御し、オープンシステムディペンダビリティの達成に寄与している。

10

【0130】

DEOS実行環境260-03はステークホルダ合意に基づくディペンダビリティを実現するサービスを提供するための実行環境であり、次のサブシステムから構成される。D-Visorは対象システムの再構成のため、システムの構成要素の各々の独立性を担保する仕組み(System Container)を提供する。あるSystem Container内における異常や障害が他のSystem Containerに波及することを抑える働きを担っている。D-Application Managerは複数のアプリケーションの独立性を担保する仕組み(Application Container)を提供し、各アプリケーションのライフサイクル(生成、起動、更新、停止、削除)を管理し制御する。D-Application Monitorはアプリケーションの動作監視機能を提供し、D-Caseモニターノードの記載に従ってエビデンスを収集し、D-Boxに蓄積する。D-System Monitorはシステム(オペレーティングシステムやランタイムサポートを含む)の動作監視機能を提供する。D-Application Monitor同様にD-Caseモニターノードの記載に従ってエビデンスを収集し、D-Boxに蓄積する。D-Boxはエビデンスを始め、OSD実現に有益な情報を安全・確実に記録する。D-Script EngineはD-Scriptを安全・確実に実行する役割を担い、D-Application Manager、D-Application Monitor、D-System Monitorを制御する。

20

30

【0131】

DEOS開発支援ツール260-04は事業目的や事業継続シナリオに基づいて決められた機能仕様、テスト仕様、ベンチマーキングシナリオ、さらにはログ仕様に基づいてプログラムを設計し、開発し、検証し、ベンチマーキングを行い、テストを行うための開発支援ツール群である。例えば、型理論ならびにモデル検証に基づいたソフトウェア検証ツール、ベンチマーキング並びにフォルトインJECTION機能を備えたディペンダビリティテスト支援ツール等がある。

【0132】

前記DEOSプロセスを適用することにより享受できる最大の利点は、ステークホルダ間で要求の変化に対する合意議論を充分に行うことができ、合意結果や、その結論に至った理由や、議論の経緯をD-Caseに記録することができる点である。システム開発時にD-Case記述を用いることにより、前記DEOSアーキテクチャと連携して、障害時に適切かつ迅速な対応を取ることが可能なシステムを設計することができる。またD-Case記述があることにより、障害の原因究明や説明責任を果たすことがより容易になる。

40

【0133】

前記DEOSプロセスのもう一つの利点は、要求が適切に抽出されリスクが充分検討されたのちに、システムの変更が実行される点である。それぞれのステークホルダはシステムの状態をどんな時点でも、それぞれの観点で知ることができる。これによりシステムを簡潔かつ強力に管理運用することができる。一方、要求の数は膨大である。D-Case Editor (260-06)やD-Case Viewer (260-07)などのツールが要求マネジメントにお

50

ける作業を軽減する。

【0134】

前記DEOSアーキテクチャを具現化したDEOS実行環境260-03は、モニタリング機能を備え、D-Caseを基に、解析に必要なシステムやアプリケーションの実行状態の監視と記録を実行する。当該実行環境はこの監視記録とスクリプトに従って障害時の迅速対応を実行する。またD-Case記述や監視記録から得られた情報をエビデンスとして原因分析や説明責任を遂行する。スクリプトとスクリプトエンジンはD-Caseと当該実行環境の橋渡しを果たしており、この仕組みにより、システムの自動的、あるいは必要であればオペレータを介した(D-Case記述に基づく)柔軟な対応を可能にする。

【0135】

また前記DEOS-アーキテクチャは、あるモジュールにおける障害が他のモジュールに伝搬することを隔離する機能を提供する。同様にシステムのセキュリティを守るために外部からの侵入を検知する機能も提供する。またディペンダビリティ達成のための、実行前にプログラムを検証するツール、パフォーマンスを測定するツール、フォルトを埋め込んで異常時の振舞いをテストする開発ツール等も提供する。

【0136】

上記の仕組みや機能を利用することにより、前記DEOSプロセスと前記DEOSアーキテクチャは、継続的な障害回避のための能力を備え、障害時には適切かつ迅速な対応をして、その影響を最小限とする。またサービスを継続し、説明責任を遂行することができる。前記DEOSプロセスと前記DEOSアーキテクチャはオープンシステムディペンダビリティを達成するための初めての取り組みである。

【0137】

〔5.実行(ランタイム)環境〕

図17を用いて本発明における一実施例に係る前記DEOSプロセスを構成する変化対応サイクルP300と障害対応サイクルP301の一連の手順を説明する。当該両サイクル共に通常は通常運用S302の状態にある。

【0138】

変化対応サイクルP300において、ビジネス環境の変化等の環境の変化によるステークホルダ要求の変更、開発途中における要求の変更、及び/又はシステム運用時における要求の変更が起こると、変化対応サイクルP300は環境変化S300-01の状態に遷移する。

【0139】

まず、要求の変更S300-02をワークスペースコンピュータ101(図1参照)の解析ツール部101-03を用いて行う。ここではステークホルダ群からの要求を抽出後に、当該要求の変更がランタイムコンピュータ102に与える影響を分析したり、具体的な変更手段を分析したり、変更箇所を分析したりする。

【0140】

次に、ディペンダビリティ記述データの変更S300-03を編集ツール部101-02及び/又は開発ツール部101-04を用いて、要求の変更S300-02での分析結果を基にして行う。ここでは、ソフトウェアSWにその結果が反映される。

【0141】

次に、変更が要求通りであることを、エビデンス計測S300-04にて、検証ツール部101-01を用いて確認する。

【0142】

最後に、ソフトウェアSWをデプロイメントS300-05にてランタイムコンピュータ102に導入する。その後、変化対応サイクルP300は通常運用S302になる。

【0143】

また、障害対応サイクルP301において、ランタイムコンピュータ102(図1参照)に障害が発生した場合には、障害発生S301-01の状態に遷移する。まず、対応策抽出S301-03においてランタイムコンピュータ102を停止させることなく、当該

10

20

30

40

50

障害に対応する対応策を抽出する。

【0144】

次に、緊急対応S301-04にて当該対応策を実行することで、ランタイムコンピュータ102におけるモデルに基づく処理を継続させる。

【0145】

次に、記録S301-05では当該緊急対応を記録部102-02により記録する。ここでは当該障害に関連する情報も記録部102-02により記録される。ランタイムコンピュータ102はこの時点で継続されていることが望ましいが、障害によっては当該対応策でも継続が不可能な場合がある。継続・不継続のいずれの場合でも原因究明S301-06を行う。これにより、当該障害の原因、および、ランタイムコンピュータ102がモデルに基づく処理が不継続の場合にはその原因が究明される。その結果次第では、要求の変更をとめない、変化対応サイクルP300を環境変化S300-01の状態に遷移させる。

10

【0146】

一方、障害対応サイクルP301において、故障（あるいは故障の予兆）が見つかった場合、あるいは障害の予兆が見つかった場合には、障害予兆S301-02の状態に遷移する。故障（あるいは故障の予兆）の発見はランタイムコンピュータ102のモニタリング部102-03によりなされる。障害の予兆の発見も当該モニタリング部102-03によってなされる。具体的には、ランタイムコンピュータ102のバイナリ実行部110における各種の計算資源の消費履歴を調べることによって可能である。例えば、メモリ残量が一定時間内で減少を続けている場合には近い将来メモリ不足で計算が停止することがわかる。

20

【0147】

まず、対応策抽出S301-03においてランタイムコンピュータ102における処理を停止させることなく、前記故障（あるいは故障の予兆）の発生、あるいは障害の予兆に対応する対応策を抽出する。

【0148】

次に、緊急対応S301-04にて当該対応策を実行することで、ランタイムコンピュータ102を継続する。

【0149】

次に、記録S301-05では当該緊急対応を記録部102-02により記録する。ここでは当該障害に関連する情報も記録部102-02により記録される。ランタイムコンピュータ102におけるモデルに基づく処理はこの時点で継続されていることが望ましいが、障害によっては当該対応策でも継続が不可能な場合がある。継続・不継続のいずれの場合でも原因究明S301-06を行う。これにより、前記障害の原因、および、ランタイムコンピュータ102におけるモデルに基づく処理が不継続の場合にはその原因が究明される。その結果次第では、要求の変更をとめない、変化対応サイクルP300を環境変化S300-01の状態に遷移させる。

30

【0150】

なお、上述のように、前記変化対応サイクルP300と障害対応サイクルP301は同時に存在する。

40

【0151】

なお、本発明は以下のように構成してもよい。

【0152】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100）は、前記ディペンダビリティ記述データを備えるシステムにおいて、環境の変化に起因するステークホルダの要求の変更、システム開発途中における要求の変更、及び/又はシステム運用時における要求の変更に対処するためのプロセスと、システム運用時におけるシステムが提供するサービスを障害発生に際しても継続するように対処し、及び/又はシステム運用時におけるシステムが提供するサービスを停止させないように障害の予兆を検知し障

50

害の発生を未然に回避するプロセスという2つの手段を同時に備えていてもよい。

【0153】

このように、前記ディペンダビリティ維持装置は、不完全性と不確実性とを潜在的に有するオープン環境において、環境の変化に起因するステークホルダの要求の変更、システム開発途中における要求の変更、及び/又はシステム運用時における要求の変更に対処するためのプロセスと、システム運用時におけるシステムが提供するサービスを障害発生に際しても継続するように対処し、及び/又はシステム運用時におけるシステムが提供するサービスを停止させないように障害の予兆を検知し障害の発生を未然に回避するプロセスという2つの手段を備えることにより、従来の開発フェーズと運用フェーズの垣根を無くし、ステークホルダのアカウントビリティ達成を可能にすると共に、ディペンダビリティを維持することが可能となる効果を奏する。

10

【0154】

また、本発明は以下のように構成してもよい。

【0155】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102）は、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを仕様の変更に合わせて変更する変化対応サイクルP300、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルP301の少なくとも何れか一方を実行するように構成されても良い。

20

【0156】

障害対応サイクルP301において、障害発生S300-01の状態に遷移することによる一連のステップは制御工学におけるフィードバック処理に相当し、障害予兆S300-02の状態に遷移することによる一連のステップは制御工学におけるフィードフォワード処理に相当する。従って、当該フィードバック処理の結果を学習することで、対応モデルを構築し、当該フィードフォワード処理に役立てるように構成することもできる。また、その際の当該モデルがディペンダビリティ記述データへの変更として記録することができるよう構成しても良い。また、当該フィードフォワード処理においては、他のシステムにおける対応モデルを上記ディペンダビリティ記述データに記述することで、ステークホルダ群が合意した状態で当該モデルに対応する障害予兆に対してプロアクティブな回避処理を行うように構成することもできる。

30

【0157】

オープン環境において対象システムのディペンダビリティを継続的に維持するためには、要求変化や環境変化に対応して十分に検討して対処する変化対応サイクルP300と、運用中の障害発生に迅速に対応する障害対応サイクルP301との両方が必要である。

【0158】

そこで、上記のように構成することにより、変化対応サイクルP300および障害対応サイクルP301の少なくともいずれか一方において、対象システムのディペンダビリティの価値を示す評価値を求めることができる。したがって、例えば、変化対応サイクルP300において、要求変更に伴ってディペンダビリティ記述データを変更するときに、評価値に基づいて、ディペンダビリティ記述データの変更案の適否を判断することが可能となる。また、例えば、障害対応サイクルP301において、対象システムの運用時に対象システムの状態を、評価値に基づいて判断することが可能となる。よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となる。

40

【0159】

〔6. 変化対応サイクル〕

図18は、図17に示したワークスペースコンピュータ101で作業される変化対応サイクルP300における、ステークホルダ間での合意形成処理S400、要求変更の実装処理S401、アカウントビリティ処理S402の3つのステップを図示している。

50

【0160】

ステークホルダ間での合意形成処理 S 4 0 0 は要求の変更 S 3 0 0 - 0 2 とディペンダビリティ記述データの変更 S 3 0 0 - 0 3 と関係している。ステークホルダ間での合意形成処理 S 4 0 0 は、要求変更の内容を理解するステップ（要求変更の理解 S 4 0 0 - 0 1）から始まる。それには、例えば、“IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specification” に準拠した S R S（ソフトウェア内部仕様書）を分析する。

【0161】

次に、影響分析 S 4 0 0 - 0 2 で既存のディペンダビリティ記述データに対する変更点を列挙する。

10

【0162】

最後に、当該変更点をディペンダビリティ記述データの変更 S 4 0 0 - 0 3 において当該ディペンダビリティ記述データの変更として反映させる。

【0163】

これらのステップはステークホルダ間で合意 S 4 0 0 - 0 4 されるまで繰り返される。

【0164】

また、要求変更の実装処理 S 4 0 1 はディペンダビリティ記述データの変更 S 3 0 0 - 0 3 とエビデンス計測 S 3 0 0 - 0 4 と関係している。要求変更の実装処理 S 4 0 1 は、設計 S 4 0 1 - 0 1 のステップから始まる。設計 S 4 0 1 - 0 1 では開発ツール部 1 0 1 - 0 4 を利用して要求変更を実際のコードに変換するためのソフトウェア設計が、例えば

20

【0165】

次に、実装 S 4 0 1 - 0 2 では、設計 S 4 0 1 - 0 1 での設計を開発ツール部 1 0 1 - 0 4 を利用してソフトウェア S W に変換する。

【0166】

次に、テスト S 4 0 1 - 0 3 では、全ステップでのソフトウェア S W を検証ツール部 1 0 1 - 0 1 を用いてテスト、検証する。

【0167】

これらのステップはテスト S 4 0 1 - 0 3 での検証が全て OK となるまで繰り返される。検証が全て OK となると、導入 S 4 0 1 - 0 4 で当該ソフトウェア S W をランタイムコンピュータ 1 0 2 に導入する。

30

【0168】

また、アカウントビリティ処理 S 4 0 2 はエビデンス計測 S 3 0 0 - 0 4 とデプロイメント S 3 0 0 - 0 5 と関係している。アカウントビリティ S 4 0 2 は、ベンチマーク等収集 S 4 0 2 - 0 1 のステップから始まる。ベンチマークはランタイムコンピュータ 1 0 2 で実行され計測されても良いし、ランタイムコンピュータ 1 0 2 をシミュレーションする環境上で実行され計測されても良い。

【0169】

次に、収集されたベンチマークデータが要求を満足しているか検証される（エビデンス検証 S 4 0 2 - 0 2）。

40

【0170】

最後に、ステークホルダは当該検証されたデータをエビデンスとして必要に応じて情報開示する（情報開示 S 4 0 2 - 0 3）。なお、情報開示 S 4 0 2 - 0 3 により、ステークホルダがアカウントビリティを達成（S 4 0 2 - 0 4）することができる。

【0171】

なお、本発明は以下のように構成してもよい。

【0172】

本発明のディペンダビリティ維持装置（ワークスペースコンピュータ 1 0 1）は、上記環境の変化に起因するステークホルダの要求の変更、システム開発途中における要求の変

50

更、及び/又はシステム運用時における要求の変更に対応するためのプロセスにおいて、それら要求を入力として、当該要求の変更の発生を起点とした一連のステップから構成されるループを形成する当該プロセスにおいて、当該要求の変更を確実に実装しシステム運用しサービスを提供できるようにステークホルダ間で要求の変更の合意を形成するステップと、当該要求の変更を実装し検証しシステム運用するステップと、当該要求の変更によるシステム運用とサービス提供によりステークホルダがアカウントビリティを達成するステップという3つのステップを実行しても良い。

【0173】

〔7. 障害対応サイクル(迅速対応)〕

図19は、障害対応サイクルP301のうち、障害発生検出時の2つの処理を示すフローチャートである。なお、図19における障害対応サイクルP301は図17の再掲である。

10

【0174】

障害発生S301-01はモニタリング部(障害発生検出手段)102-03によって障害の発生が検出されると当該状態に遷移する。モニタリング部102-03は、例えば、D値が所定の変動許容範囲より悪化したとき、障害発生を検出する。障害発生検出時の障害対応サイクルP301には2つの機能が関係している。システム再構成処理S502とアカウントビリティ処理S503である。

【0175】

システム再構成処理S502は対応策抽出S301-03と緊急対応S301-04に関係している。システム再構成処理S502はサービス継続判断S502-01から始まる。ここでは、障害に該当する部分のディペンダビリティ記述データを抽出する。当該ディペンダビリティ記述データからシステム再構成シナリオが抽出可能である場合には(S502-01でYES)、次のステップに進む。なお、ディペンダビリティ記述データからシステム再構成シナリオが抽出不可能である場合には(S502-01でNO)、継続不可能と判断して、ディペンダビリティ記述データの変更抽出処理S800(図27)を実行する。

20

【0176】

システム再構成シナリオはシステム設計時点であらかじめ典型的なケースに関しては組み込んでおいても良い。例えば、後述のアプリケーション・コンテナあるいはシステム・コンテナを再起動するシナリオはその一例である。また、当該再構成シナリオを後述同様にディペンダビリティ記述データにおけるノードに実行可能なプログラムを関連つけても良い。いずれの手法においても、当該再構成シナリオが抽出できない場合には、ディペンダビリティ記述データを変更するために、変化対応サイクルP300を実行する準備をする。また、当該再構成シナリオを後述のスクリプトとして記述し、ディペンダビリティ記述データに関連つけても良い。

30

【0177】

次に、障害部位隔離S502-02では当該抽出されたシナリオに従って障害部位を隔離し、ディペンダビリティ記述データの違いをランタイムコンピュータ102に反映させ、次ステップに進む。

40

【0178】

次に、サービス継続S502-03では当該シナリオに従って隔離された部位の代替機能が当該シナリオによって起動され、サービスが継続される(S502-04)。

【0179】

一方、アカウントビリティ処理S503は記録S301-05と原因究明S301-06に関係している。アカウントビリティ処理S503は障害に関係するログを記録する(S503-01)ことから始まる。

【0180】

次に、前記シナリオに従って再構成されたランタイムコンピュータ102のエビデンスを収集する(S503-02)。ここでのエビデンスとはシステム再構成によってランタ

50

イムコンピュータ102がディペンダビリティ記述データに従って適切に動作していることの記録である。

【0181】

次に、障害原因検出S503-03に進む。ここでは、障害に関係する前記ログ(S503-01)、前記エビデンス(S503-02)、及び前記障害に関係したディペンダビリティ記述データをまとめたものである。本実施の形態では、当該まとめをステークホルダによるアカウントビリティの達成とする。なお、これらのステップ(S503-01及びS503-02)は並列に処理されても良い。

【0182】

図20にモニタリング部102-03による障害発生検出処理の一例を示す。

10

【0183】

図20(a)はディペンダビリティ記述データの一部(510)を示す。ディペンダビリティ記述データを作成する際に、実行時に状況が変化するデータは、モニタノードによって実行時にモニタリングが必要であることを明示しておく。

【0184】

ノード(510-01)は「正常なDBSのHDDアクセス」に関する。当該ノードは2個に分岐され、すなわち、データベースシステムがHDD装置に対して正常にアクセスしている状態を2個のノードにブレークダウンして、その条件を記述している。具体的には、ノード(510-02)は「HDDの残容量」に関する。また、ノード(510-03)は「HDDの転送量」に関する。

20

【0185】

ここで、本例では、これら2ノードのエビデンスとして前記モニタノードを導入する。図20(a)では、ノード(510-04)とノード(510-05)がモニタノードである。ノード(510-04)はHDDの残容量をモニタする。ノード(510-05)はHDD転送速度をモニタする。

【0186】

ディペンダビリティ記述データにおけるモニタノードは、対象のシステムを外部からモニタリング可能な場合にはモニタリング部102-03が備える監視モジュール群(後述)を利用して対象のシステムをモニタリングする。一方、対象のシステムを外部からモニタリング不可能な場合には、対象のシステムに後述のプログラムを導入する手順に従って監視モジュールを送り込むことでモニタリングを行う。

30

【0187】

そして、実行時のモニタリング結果をディペンダビリティ記述データの記述と比較することにより、変動許容範囲を超える差分の場合には、設計時の基準と異なる振る舞いとして異常として特定する。また、ディペンダビリティ記述データの上位ノードを参照することで、複合的な異常により上位ノードのディペンダビリティが満たされない状況を把握できる。さらに、個々のシステム内や複数のシステム間の障害箇所を突き合わせることで、障害発生箇所を絞り込むことができる。

【0188】

図20(b)にモニタノードを用いて障害検出するためのディペンダビリティ維持システム100の構成例を示す。モニタノードはパターンによって分類することができる。例えば、図20(a)の2個のモニタノード(510-04、510-05)は『HDD「\$1」の「\$2」が「\$3」であることをモニタリングする』という形式を取ることができる。このモニタノードのパターンに対応するように監視モジュール群を予め用意し、モニタノードのパターンと監視モジュールの対応関係をテーブル化しておくことにより、実行時の状態をモニタリングできる。

40

【0189】

例えば、図20(a)の2個のモニタノード(510-04、510-05)に関しては、図21のように対応関係を示すことができる。ここで、図21は、『HDD「\$1」の「\$2」が「\$3」であることをモニタリングする』というパターンのモニタノードの場

50

合に、\$ 2 が「ディスク容量」であれば、監視モジュールとしてHDD監視モジュールを利用し、引数として\$ 1 (=HDD識別番号)と\$ 3 (=変動許容範囲)を指定すれば良いことを示している。

【0190】

ワークスペースコンピュータ101の編集ツール部101-02はディペンダビリティ記述データM1を記述し、さらに、ランタイムコンピュータ102における内部表現であるディペンダビリティ記述データ内部表現M2に変換する。詳細には、ワークスペースコンピュータ101の編集ツール部101-02において、ディペンダビリティ記述データ記述・表示ツール511-01がディペンダビリティ記述データM1を記述した後、それを、ディペンダビリティ記述データ変換部511-03によってディペンダビリティ記述データ内部表現M2に変換する。

10

【0191】

ランタイムコンピュータ102では、ワークスペースコンピュータ101から取得したディペンダビリティ記述データ内部表現M2を更新部102-01がディペンダビリティ記述データ格納部511-05に記録する。また、更新部102-01は、ディペンダビリティ記述データ内部表現M2に基づいて、モニタノードとモニタリング部102-03における各種監視モジュール(監視モジュール群511-m)との対応関係をモニタノード/モジュール対応テーブル511-06に記録する。なお、図20(b)には監視モジュール群511-mの各監視モジュールとして、CPU監視モジュール511-08、HDD監視モジュール511-09、プロセス監視モジュール511-10、が記載されているが、他の監視モジュール511-11があっても良い。

20

【0192】

モニタリング部102-03における前記監視モジュールによって得られたデータ(モニタ値)は収集部510-13によって集められ、実行状況検証部510-12によって、モニタノード/モジュール対応テーブル511-06に基づき、前記ディペンダビリティ記述データにおけるモニタノードの値として記録される。例えば、ディペンダビリティ記述データの各モニタノードに前記モニタ値の変動許容範囲を対応付けて記載しておき、モニタ値がその変動許容範囲から外れたノードを障害発生ノードとして検出する。また、前記監視モジュール群511-mからのモニタ値はD値計算部102-06においてノードに対するD値の計算に利用される。よって、例えば図20(a)のディペンダビリティ記述データの一部510では、2個のモニタノード(510-04、510-05)から取得されたモニタ値が変動許容範囲と比較されるとともに、これに基づいてD値が算出される。

30

【0193】

また、CPU利用率、メモリ使用量、プロセス死活等は、のOS機能としてあらかじめ備わっており、外部からモニタリング可能なものもあるが、「システムA内のある処理Xが正常終了した」というような、外部からのモニタリングが難しいものもある。これらは、主に、アプリケーション固有の情報であり、ランタイムコンピュータ102に監視モジュール511-14として組み込む。すなわち、モニタリング対象をモニタリングする監視モジュールが内部にない場合、監視モジュール511-14のように、外部に監視モジュールを設けることができる。この組み込みはシステム構築時であっても良いし、ランタイムコンピュータ102の実行時であっても良い。これにより、実行時のモニタリング結果が変動許容範囲を超えた場合に、ディペンダビリティ記述データのノード単位で異常を特定できる。

40

【0194】

また、実行時に各々の監視モジュールからモニタリング結果を集める(510-04、510-05)ことにより、ディペンダビリティ記述データ上のどのモニタノードで変動許容範囲を超えているかを判断できる。つまり、ディペンダビリティの観点から問題の発生を把握できる。すなわち、ディペンダビリティ記述データを当該モニタノードから上位側へ検討することで、どの範囲でディペンダビリティが維持できなくなったかを判断でき

50

る。例えば、ノード(510-04)及びノード(510-05)のモニタ値が共に変動許容範囲を超えているような状況では、ディペンダビリティを維持できなくなった原因がHDD以外にあるとして、ディペンダビリティ記述データのより上位を探索する。さらに、複数のシステムが相互作用している状況では各々のシステムに関するディペンダビリティ記述データだけでなく、相互作用に関するディペンダビリティ記述データも記述されていれば、適切なモニタノードによってシステム相互間で変動許容範囲を超える振る舞いの部位を絞り込むこと、あるいは特定することができる。

【0195】

前記監視モジュール群511-mはシステムにあらかじめ組み込まれた監視モジュール群であるが、運用中に新たな監視モジュールを組み込む場合、あるいは変動許容範囲の判断が複数のセンサからのデータを考慮しなければ行けなかつたり、ログからのデータを解析しなければ行けなかつたりする場合には、スクリプトで構成することができる。

10

【0196】

図22を用いて、スクリプトの構成概要を説明する。スクリプトは1以上の基本動作550-01(図ではD-Taskとして記載)と1以上の制御動作550-02(図ではD-Controlとして記載)から構成される。制御動作550-02は変動許容範囲を深刻度で分けて該当する深刻度の基本動作(D-Task)を実行する。例えば、ハードディスクの残容量の場合には、50%の残容量の深刻度を「低」、75%の残容量の深刻度を「中」、90%の残容量の深刻度を「高」として設定して、制御動作550-02(D-Control)を構成できる。この場合、深刻度「高」の場合にはD-Task1(550-03)を実行し、深刻度「中」の場合にはD-Task2(550-04)を実行し、深刻度「低」の場合にはD-Task3(550-05)を実行する。

20

【0197】

なお、本発明は以下のように構成してもよい。

【0198】

本発明のディペンダビリティ維持装置(ランタイムコンピュータ102)は、上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスを障害発生に際しても継続するように対処するプロセスにおいて、障害の発生を検出する手段(モニタリング機能)と、障害の発生した部分を隔離する手段(隔離機能)と、隔離された残りの部分でサービスを継続する手段と、障害の原因を検出する手段(原因追及機能)と、上記システム運用時におけるシステムが提供するサービスの継続性を判断する手段(継続性判断機能)とを備えるプロセスにおいて、当該障害の発生を起点とした一連のステップから構成されるループを形成する当該プロセスにおいて、当該障害の発生に際してサービスの停止期間を最小限にするように前記ディペンダビリティ記述データを備える対象システムを再構成するステップと、当該再構成によるシステム運用とサービス提供によりステークホルダがアカウントビリティを達成することができるステップという2つの処理を実行しても良い。

30

【0199】

また、本発明は以下のように構成してもよい。

【0200】

本発明のディペンダビリティ維持装置(ランタイムコンピュータ102)は、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを仕様の変更に合わせて変更する変化対応サイクルP300、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルP301のうち、少なくとも上記障害対応サイクルP301を実行するとともに、上記障害発生を検知したとき、上記ディペンダビリティ記述データに基づいて、当該対象システムを再構成する再構成手段(再構成部102-04)を備えて構成されても良い。

40

【0201】

これにより、障害対応サイクルP301を実行するランタイムコンピュータ102にお

50

いて、対象システムの障害発生を検知したとき、ディペンダビリティ記述データに基づいて、当該対象システムを再構成して、対象システムの運用を継続させることができる。

【0202】

そして、ランタイムコンピュータ102は、対象システムの障害発生を、D値に基づいて検知してもよい。例えば、D値が所定の変動許容範囲より悪化したとき、障害発生を検出するようにしても良い。

【0203】

〔8. 障害対応サイクル（障害未然回避）〕

図23は、障害対応サイクルP301のうち、障害予兆検出時の2つの処理を示すフローチャートである。なお、図23における障害対応サイクルP301は図17の再掲である。

10

【0204】

障害予兆S301-02はモニタリング部（障害予兆検出手段）102-03によって障害の予兆、あるいは故障（あるいは故障の予兆）が検出されると当該状態に遷移する。なお、障害の予兆に関しては各種の手法を組み合わせることで実現可能である。例えば、ハードディスクドライブ装置はS.M.A.R.T.の仕組みを使うことで障害の予兆を検出できる。また、モニタリング部102-03は、例えば、D値の時間変化の傾向が所定の基準より悪化したとき、障害予兆を検出する。また、当該モニタリング部102-03は、前記モニタリング部でモニタリングされているデータの時間変化の傾向が変動許容範囲を超えている場合に障害予兆として検出する。障害予兆検出時の障害対応サイクルP301には2つの機能が関係している。システム再構成処理S602とアカウントビリティ処理S603である。

20

【0205】

システム再構成処理S602は対応策抽出S301-03と緊急対応S301-04に関係している。システム再構成処理S602は障害部位とその影響範囲の同定S602-01から始まる。ここでは、障害の予兆と障害そのものの検出を説明の都合上同一視して障害として扱う。まず、ディペンダビリティ記述データのどの部位が障害かを同定する。そして当該ディペンダビリティ記述データからその影響範囲を同定する。影響範囲はディペンダビリティ記述データのノードを探索することにより同定する。例えば、障害部位に対応するノードが複数ある場合には、それらの共通ノードまでを影響範囲として特定できる。また、ノードにノード間の依存関係が記録されている場合には、当該依存関係を利用してよい。

30

【0206】

次に、サービス継続判断S602-02では、当該ディペンダビリティ記述データから抽出される対応シナリオが当該影響範囲を十分にカバーしてサービスが継続できるか、判断される。継続可能な場合には（S602-02でYES）、障害部位隔離S602-03に進む。なお、ディペンダビリティ記述データから対応シナリオが抽出できない場合には（S602-02でNO）、継続不可能と判断して、ディペンダビリティ記述データの変更抽出処理S800（図27）を実行する。当該変更抽出処理S800は、サービス継続判断S602-02にて継続可能な場合であっても、当該障害予兆の内容によっては実行された方がよい状況もあるので、一実施例として当該変更抽出処理を実行するように構成してもよい。

40

【0207】

次に、障害部位隔離S602-03では、抽出された対応シナリオに従って障害部位を隔離する。

【0208】

次に、サービス継続S602-04に進み、システム再構成処理S602を終了する。この時点でランタイムコンピュータ102によるサービスは継続されている（S602-05）。

【0209】

50

一方、アカウントバリティ処理 S 6 0 3 は記録 S 3 0 1 - 0 5 と原因究明 S 3 0 1 - 0 6 に関係している。アカウントバリティ処理 S 6 0 3 は内部状態の記録 S 6 0 3 から始まる。ここでは、前記障害に関連したディペンダバリティ記述データ、及び当該ディペンダバリティ記述データに関連したランタイムコンピュータ 1 0 2 の内部の状態を記録する。
【 0 2 1 0 】

次に、システム再構成処理 S 6 0 2 後のエビデンスを収集する (S 6 0 3 - 0 2)。ここでのエビデンスとはシステム再構成によってランタイムコンピュータ 1 0 2 がディペンダバリティ記述データに従って適切に動作していることの記録である。なお、本実施の形態では、当該エビデンスの収集をもってステークホルダによるアカウントバリティの達成とする (S 6 0 3 - 0 3)。

10

【 0 2 1 1 】

前記緊急対応 S 3 0 1 - 0 4 は上記スクリプト (D-Script) を用いて処理しても良い。図 2 4 に一例を示す。当該スクリプト (D-Script) は図 4 (b) 記載のWEBサービスの再起動のシナリオ例であり、6個の基本動作 (D-Task) から構成されている。当該シナリオは上記DEOSプロセスにおける要求抽出・リスク分析フェーズ 2 5 0 - 1 1 におけるサービス継続シナリオから導出される。ステークホルダは当該シナリオに関して合意している。当該サービス継続シナリオは要求抽出・リスク分析フェーズを支援するツール群 2 6 0 - 0 1 を用いて記述され、図 2 4 の形式で記述され、スクリプトに変換され、ステークホルダ合意によって電子署名される。

20

【 0 2 1 2 】

スクリプトがランタイムコンピュータ 1 0 2 内部の後述のスクリプトエンジン部 7 1 0 (1 0 2 - 0 7) において実行される際には、当該スクリプトに附随された電子署名が検証される。電子署名が有効でない場合にはスクリプトは実行されない。さらに、スクリプトの実行に際しては、その権限を確認するように構成しても良い。すなわち、誰がどのスクリプトの操作を実行可能かを確認するように構成可能である。その場合には権限を有しない状況では当該スクリプトは実行されない。さらに、スクリプト自体を暗号化してランタイムコンピュータ 1 0 2 内部に格納するように構成しても良い。これらのセキュリティに関する施策により、当該スクリプトが原因となるセキュリティ違反を防止することができる。

30

【 0 2 1 3 】

図 2 5 を用いてランタイムコンピュータ 1 0 2 に関するソフトウェアの階層構造の一例を説明する。

【 0 2 1 4 】

図 2 5 では3個のCPUコア 7 0 1 (7 0 1 - 0 1 ~ 7 0 1 - 0 3) を有するマルチコア構成例を示している。なお、図 2 5 では周辺デバイスは省略している。CPUコアの上位層にはシステム・コンテナ提供部 7 0 2 がある。これは、隔離部 1 0 2 - 0 5 (図 1) の一部であり、上位層に対して複数のOSカーネル 7 1 1 を動作可能にする、隔離されたコンテナを提供する。また、図 2 5 では各々のCPUコア 7 0 1 に対してそれぞれ1つのシステム・コンテナ 7 1 6、7 1 8、及び7 2 0 が割り当てられているが、複数のシステム・コンテナを1つのCPUコア 7 0 1 に割り当てても良いし、1つのシステム・コンテナに対して複数のCPUコア 7 0 1 ... を割り当てても良い。

40

【 0 2 1 5 】

CPUコア 7 0 1 - 0 1 にはシステム・コンテナ 7 1 6 を割り当て、そこではOSカーネル 7 1 1 上に、システムモニタ部 7 0 4、再構成部 1 0 2 - 0 4、システム記録部 7 0 6、ディペンダバリティ記述データ処理部 7 0 7 が配置される。

【 0 2 1 6 】

システム・コンテナ 7 1 6 はランタイムコンピュータ 1 0 2 に1つ存在する。システムモニタ部 7 0 4 はモニタリング部 1 0 2 - 0 3 (図 1) の一部でありランタイムコンピュータ 1 0 2 におけるシステム機能をモニタする。特に他のシステム・コンテナ (図 2 5 ではシステム・コンテナ 7 1 8 と 7 2 0) のOSカーネル 7 1 1 をモニタする。また、当該

50

システムモニタ部 704 に学習機能を持たせるように構成しても良い。例えば、対象システムにおけるシステム機能をモニタする際に、そのデータ構造に対する整合性をモニタするように構成する場合に、当該システム機能に対応した仕様書やソースコードから整合性のための条件を自動生成するように構成することができる。実行時に当該条件を学習することで、当該データ構造の全てをモニタしないでも、または、当該データ構造が変更されても当該モニタの仕組みを変えないでモニタ機能を継続することができる。

【0217】

再構成部 102 - 04 は、他のシステム・コンテナの再構成を行う。具体的には、既に存在するシステム・コンテナを停止させ、破壊し、新しい構成でシステム・コンテナを作り出し、当該構成に従って OS カーネルを始めとする機能を配置し起動する。システム・コンテナが破壊されるときには、当該システム・コンテナの内部状態を、システム記録部 706 を用いて記録しても良い。システム記録部 706 は記録部 102 - 02 (図 1) の一部であり、ランタイムコンピュータ 102 の内部状態を記録する。

10

【0218】

ディペンダビリティ記述データ処理部 707 は更新部 102 - 01 (図 1) の一部であり、ワークスペースコンピュータ 101 で生成されたソフトウェア SW の内部にあるディペンダビリティ記述データをランタイムコンピュータ 102 全体のディペンダビリティ記述データに取り込み、ディペンダビリティ記述データを処理するためのシナリオを抽出しランタイムコンピュータ 102 を、再構成部 102 - 04 を利用して再構成する。

20

【0219】

一方、CPU コア 701 - 02 にはシステム・コンテナ 718 を割り当て、そこでは OS カーネル 711 上に、アプリケーションモニタ部 708、アプリケーション記録部 709、スクリプトエンジン部 710、アプリケーション・コンテナ提供部 712、アプリケーション管理部 713 乃至 714、そしてアプリケーション群を配置している。

【0220】

ここで、アプリケーション・コンテナとは、前記システム・コンテナ 1 個に対して 1 個以上存在し、1 個以上のアプリケーションをグループ化し、アドレス空間の独立、名前空間の独立、及び CPU スケジューリングの独立を実現し、アプリケーション・コンテナ提供部 712 により提供される。図 25 ではシステム・コンテナ 718 にアプリケーション・コンテナ 717 及び 719 の 2 個を配置している。

30

【0221】

アプリケーションモニタ部 708 はモニタリング部 102 - 03 (図 1) の一部であり、アプリケーション・コンテナ内部をモニタする。アプリケーション記録部 709 は記録部 102 - 02 (図 1) の一部であり、アプリケーション・コンテナの内部状態を記録している。

【0222】

アプリケーション・コンテナ 717 にはアプリケーション管理部 713 を配置し、1 個以上のアプリケーションが利用する。同様にアプリケーション・コンテナ 719 内部にはアプリケーション管理部 714 を配置し、1 個以上のアプリケーションが利用する。

40

【0223】

システム・コンテナ 720 の内部構成はシステム・コンテナ 718 と同様である。そのため、説明および図 25 中の表示を省略する。

【0224】

図 26 にランタイムコンピュータ 102 の隔離部 102 - 05 (図 1) が実現する各隔離項目に関しての機能要件の一例をまとめる。これらの隔離項目は一部を選択して必要な機能要件のみを実現しても良い。例えば、前記システム・コンテナは、これら項目の全部が実現されているし、アプリケーション・コンテナはアドレス空間、名前空間、及び CPU スケジューリングの 3 項目が実現されている。

【0225】

なお、本発明は以下のように構成してもよい。

50

【 0 2 2 6 】

本発明のディペンダビリティ維持装置（ランタイムコンピュータ102）は、上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスを停止させないように障害の予兆を検知し障害の発生を未然に回避するプロセスにおいて、前記ディペンダビリティ記述データを備えるシステムの内部状態を記録する手段（ロギング）と、当該記録された内部状態から前記ディペンダビリティ記述データを備えるシステムの障害の予兆を検出する手段と、前記ディペンダビリティ記述データを備えるシステムの当該検出された障害に対応する部分を同定する手段と、当該検出された障害が上記システム運用時におけるシステムが提供するサービスの継続性を判断する手段とを備えるプロセスにおいて、当該障害の発生を検出を起点とした一連のステップから構成されるループを形成する当該プロセスにおいて、当該検出された障害発生、あるいは障害の予兆によるサービスの停止を回避する目的で前記ディペンダビリティ記述データを備えるシステムを再構成するステップと、当該再構成によるシステム運用とサービス提供によりステークホルダがアカウントビリティを達成することができるステップという2つの処理を実行しても良い。

10

【 0 2 2 7 】

また、本発明は以下のように構成してもよい。

【 0 2 2 8 】

本発明のディペンダビリティ維持装置（ランタイムコンピュータ102）は、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを仕様の変更に合わせて変更する変化対応サイクルP300、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルP301のうち、少なくとも上記障害対応サイクルP301を実行するとともに、上記障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、当該対象システムを再構成する再構成手段（再構成部102-04）を備えて構成されても良い。

20

【 0 2 2 9 】

これにより、障害対応サイクルP301を実行するランタイムコンピュータ102において、対象システムの障害予兆を検知したとき、ディペンダビリティ記述データに基づいて、当該対象システムを再構成して、対象システムの運用を継続させることができる。

30

【 0 2 3 0 】

そして、ランタイムコンピュータ102は、対象システムの障害予兆を、D値に基づいて検知してもよい。例えば、D値の時間変化の傾向が所定の基準より悪化したとき、障害予兆を検出するようにしても良い。

【 0 2 3 1 】

さらに、ランタイムコンピュータ102は、対象システムの障害予兆を、前記モニターボードでのモニタリングデータに基づいて検知してもよい。例えば、当該モニタリングデータの時間変化の傾向が所定の基準より悪化し変動許容範囲を超えたとき、障害予兆を検出するようにしても良い。

40

【 0 2 3 2 】

〔 9 . 障害対応サイクルからの変化対応サイクルの起動 〕

図27は、ディペンダビリティ維持システム100におけるディペンダビリティ記述データの変更抽出処理の手順を示すフローチャートである。

【 0 2 3 3 】

上述したサービス継続判断S502-01（図19）、あるいは、サービス継続判断602-02（図23）において、サービス継続判断が「NO」すなわち継続不可能と判断されたときには、ディペンダビリティ記述データの変更抽出処理S800を実行する。

【 0 2 3 4 】

まず、障害部位に関するディペンダビリティ記述データを抽出する（S800-01）。次に、そのディペンダビリティ記述データにおけるリーフノードでテストと関連のある

50

ノード全てにおいて、S 8 0 0 - 0 3 ~ S 8 0 0 - 0 5 の処理を実行する (S 8 0 0 - 0 2)。

【 0 2 3 5 】

例えば、図 5 においては、ノード群 (2 0 2 - 0 6 ~ 2 0 2 - 1 3) が該当する。そこで、当該ノード群 (2 0 2 - 0 6 ~ 2 0 2 - 1 3) に対応するテストを実行し計測する (S 8 0 0 - 0 3)。既にエビデンスとしての結果は存在しているので、ここでの計測結果と比較することでエビデンスからの差異が変動許容範囲内か否かが判断される (S 8 0 0 - 0 4)。変動許容範囲外の場合には (N O)、異常状態であるとして当該ノードの識別情報 (例えば、ノード番号) を記録する (S 8 0 0 - 0 5)。

【 0 2 3 6 】

全ノードに対して前記処理を終えると、前記識別情報が記録されたノードを変更する指示をワークスペースコンピュータ 1 0 1 に送り (S 8 0 0 - 0 6)、処理を終了する (S 8 0 0 - 0 7)。

【 0 2 3 7 】

なお、本発明は以下のように構成してもよい。

【 0 2 3 8 】

本発明のディペンダビリティ維持装置 (ランタイムコンピュータ 1 0 2) は、前記障害対応サイクル P 3 0 1 において、上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスの継続が不可能と判断された場合、前記変化対応サイクル P 3 0 0 において前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、前記障害対応サイクル P 3 0 1 において記録又は検出された情報から前記ディペンダビリティ記述データを備えるシステムのディペンダビリティ記述データの変更を前記障害対応サイクル P 3 0 1 へ要求として入力するステップを実行しても良い。

【 0 2 3 9 】

また、本発明は以下のように構成してもよい。

【 0 2 4 0 】

本発明のディペンダビリティ維持装置 (ディペンダビリティ維持システム 1 0 0、ランタイムコンピュータ 1 0 2) は、上記障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル P 3 0 0 を実行する装置 (ワークスペースコンピュータ 1 0 1) に、上記ディペンダビリティ記述データの変更の要求を送信する変更要求送信手段 (変更送出部 9 0 2 - 0 2) を備えて構成されても良い。

【 0 2 4 1 】

本発明のディペンダビリティ維持装置 (ディペンダビリティ維持システム 1 0 0、ワークスペースコンピュータ 1 0 1) は、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを仕様の変更に合わせて変更する変化対応サイクル P 3 0 0、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクル P 3 0 1 のうち、少なくとも上記変化対応サイクル P 3 0 0 を実行するとともに、上記障害対応サイクル P 3 0 1 を実行する装置 (ランタイムコンピュータ 1 0 2) から、上記ディペンダビリティ記述データの変更の要求を受信したとき、上記ディペンダビリティ記述データを変更するように構成されても良い。

【 0 2 4 2 】

これにより、障害対応サイクル P 3 0 1 を実行するランタイムコンピュータ 1 0 2 において、対象システムの障害発生または障害予兆を検知し、対象システムの停止が回避であると判断したとき、変化対応サイクル P 3 0 0 を実行するワークスペースコンピュータ 1 0 1 に、ディペンダビリティ記述データの変更の要求を送信することができる。なお、ランタイムコンピュータ 1 0 2 は、障害発生または障害予兆により、対象システムの停止が回避であるか否かを、D 値に基づいて判断しても良い。

【 0 2 4 3 】

10

20

30

40

50

そして、ワークスペースコンピュータ101は、ランタイムコンピュータ102からディペンダビリティ記述データの変更の要求を受信したとき、当該要求に応じて、上記ディペンダビリティ記述データを変更する。

【0244】

したがって、ランタイムコンピュータ102とワークスペースコンピュータ101とが連携して、対象システムの障害発生または障害予兆を検知し、対象システムの停止が回避であれば、ディペンダビリティ記述データを変更するというプロセスを円滑に実行することができる。よって、オープン環境において、対象システムのディペンダビリティを継続的に維持することが可能となる。

【0245】

なお、前記記載の構成において、対象システムの停止が回避可能である場合においても、障害予兆の内容によっては前記記載のディペンダビリティ記述データの変更手順を実行するように構成しても良い。

【0246】

〔10．障害対応サイクルと変化対応サイクルとの連携〕

図28は、ワークスペースコンピュータ101とランタイムコンピュータ102とのディペンダビリティ記述データを介した情報交換を示す機能ブロック図である。

【0247】

ワークスペースコンピュータ101における編集ツール部101-01は、主に3個の構成要素を持つ。ディペンダビリティ記述データ編集部900-01は、ディペンダビリティ記述データを記述し、編集し、記録する。当該ディペンダビリティ記述データ編集部900-01は、図16に記載のD-Case Editor(260-06)として構成しても良い。なお、ディペンダビリティ記述データ編集部900-01として、特願2010-263681に記載のツールを用いても良い。当該特願2010-263681では、要求ベースにディペンダビリティ記述データを記述する際に、常に着目ノードが表示の中心になるようにディペンダビリティ記述データを表示することで詳細化の流れを簡素にしている。

【0248】

ディペンダビリティ記述データ組込部900-02は、ディペンダビリティ記述データ編集部900-01にて作成されたディペンダビリティ記述データM1を内部表現に変換して、ディペンダビリティ記述データM2(正確には、ディペンダビリティ記述データ内部表現M2;図3のディペンダビリティ記述データ104-02に相当)を生成する。そして、ディペンダビリティ記述データ組込部900-02は、ソフトウェアSWに、当該ソフトウェアSWに対応した(あるいは関する)前記ディペンダビリティ記述データM2を組み込む。デプロイメント部900-03は当該ソフトウェアSWをランタイムコンピュータ102に導入する。

【0249】

ここで、ディペンダビリティ記述データM2には、ディペンダビリティ記述データ編集部900-01によって、各エビデンスノードについて、モニタされるモニタ値に対する変動許容範囲が記録される。これにより、ランタイムコンピュータ102のモニタリング部102-03では、モニタ値の状態を確認できる。一方、ディペンダビリティ記述データM3には、記録部102-02によって、各エビデンスノードについて、モニタされたモニタ値のすべてあるいは一部が記録される。これにより、ワークスペースコンピュータ101の解析部901-03では、システムに生じた異常の原因を検討することができる。

【0250】

また、ワークスペースコンピュータ101における解析ツール部101-03は、主に3個の構成要素を持つ。ディペンダビリティ記述データ入力部(記述データ取得手段)901-01はランタイムコンピュータ102からのディペンダビリティ記述データM3を入力とする。データベースアクセス部901-02はランタイムコンピュータ102に導入されているソフトウェア全体に関する複数のディペンダビリティ記述データにアクセス

10

20

30

40

50

する。データベースは、ワークスペースコンピュータ101に設けられても良いし、ランタイムコンピュータ102に設けられても良いし、ワークスペースコンピュータ101と通信可能な他の装置に設けられても良い。ここでは、入力されたディペンダビリティ記述データM3に関連したディペンダビリティ記述データM4を、当該データベースから抽出する。ディペンダビリティ記述データM4はオリジナルのモデルであり、ディペンダビリティ記述データM3は変更が記載されたモデルである。当該データベースから抽出されるディペンダビリティ記述データM4はディペンダビリティ記述データM3をキーに検索されるが、より広範囲のディペンダビリティ記述データを抽出しても良く、解析部901-03での処理を容易にする。解析部901-03は当該入力されたディペンダビリティ記述データM3とデータベースアクセス部901-02経由で得られるディペンダビリティ記述データM4を解析して、障害原因等を分析し、必要に応じて環境変化S300-01(図17、図18)の状態に遷移させて、変化対応サイクルP300を実行する。

10

【0251】

一方、ランタイムコンピュータ102において更新部102-01は、6個の構成要素を持つ。ソフトウェア入力部(記述データ取得手段)902-01はワークスペースコンピュータ101におけるデプロイメント部900-03からのソフトウェアSWを入力とする。当該ソフトウェアSWは更新部102-01によってランタイムコンピュータ102にインストールされ稼働される。ディペンダビリティ記述データ抽出部902-03はソフトウェアSWに含まれているディペンダビリティ記述データM2(正確には、ディペンダビリティ記述データ内部表現M2;図3のディペンダビリティ記述データ104-02に相当)を取り出す。構造検証部902-04は、当該取り出されたディペンダビリティ記述データM2の計算機表現としての妥当性を検証する。さらに、必要な機能(ライブラリ等)がランタイムコンピュータ102に備わっているかを確認する。実行管理部902-05は当該ソフトウェアSWをランタイムコンピュータ102に導入する際に必要な準備を行う。例えば、当該ソフトウェアSWが新しいサービスアプリケーションである場合には、必要な計算資源を見積もり割り当て、アプリケーション・コンテナ(図25のアプリケーション・コンテナ717等)を生成し、アプリケーションイメージを当該アプリケーション・コンテナ内に構成する。

20

【0252】

また、変更送出部(変更要求送信手段)902-02はランタイムコンピュータ102にインストールされたディペンダビリティ記述データM2における変更部分(ディペンダビリティ記述データM3)をワークスペースコンピュータ101のディペンダビリティ記述データ入力部901-01に送り出す。このとき、変更送出部902-02は、上記変更部分と共に、ディペンダビリティ記述データの変更要求を送出しても良い。変更管理部902-06はランタイムコンピュータ102において、図27に示したディペンダビリティ記述データの変更を管理する。

30

【0253】

前記において、ワークスペースコンピュータ101とランタイムコンピュータ102は、ディペンダビリティ記述データという計算機表現を介して情報交換する。前者から後者に対してはディペンダビリティ記述データM2を手段とし、後者から前者に対してはディペンダビリティ記述データM3を手段とする。ディペンダビリティ記述データM2はステークホルダ合意に対応したモデルであり、ディペンダビリティ記述データM3は当該ステークホルダ合意がランタイムコンピュータ102において満足されなくなり、その差分を表現したものであり、ステークホルダに対する通知である。

40

【0254】

なお、本発明は以下のように構成してもよい。

【0255】

本発明のディペンダビリティ維持装置(ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102)は、前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、前記変化対

50

応サイクル P 3 0 0 と前記障害対応サイクル P 3 0 1 との間での情報交換の手段を有していても良い。

【 0 2 5 6 】

〔 1 1 . ディペンダビリティ記述データとの連携 〕

図 2 9 は、図 5 に示したディペンダビリティ記述データを、計算機表現の一例として XML で記述したリストの一部である。なお、図 2 8 におけるソフトウェア SW に組み込まれたディペンダビリティ記述データ M 2、あるいはディペンダビリティ記述データ M 3 は当該 XML 表記により記述することができる。

【 0 2 5 7 】

図 3 0 は、ディペンダビリティ維持システム 1 0 0 の構成例であって、ディペンダビリティ記述データデータベースと、ワークスペースコンピュータおよびランタイムコンピュータとの関係を示す説明図である。

10

【 0 2 5 8 】

図 3 0 に示すように、ソフトウェア SW に組み込まれているディペンダビリティ記述データ M 2 は、ワークスペースコンピュータ 1 0 1 およびランタイムコンピュータ 1 0 2 とは別体のディペンダビリティ記述データベース 1 0 0 0 に格納され、ワークスペースコンピュータ 1 0 1 およびランタイムコンピュータ 1 0 2 から利用されても良い。当該データベースは図 1 5 に記載の合意記述データベース 2 5 0 - 0 6 として構成しても良い。

【 0 2 5 9 】

20

なお、本発明は以下のように構成してもよい。

【 0 2 6 0 】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム 1 0 0、ワークスペースコンピュータ 1 0 1、ランタイムコンピュータ 1 0 2）は、前記ディペンダビリティ記述データから変換されたコンピュータで処理可能な形式のデータを、当該ディペンダビリティ記述データを備えるシステムが管理する手段を有することで、前記変化対応サイクル P 3 0 0 におけるステークホルダによる変更の要求の合意のステップによる結果を、確実に当該ディペンダビリティ記述データを備えるシステムに実施する手段を有していてもよい。

【 0 2 6 1 】

30

〔 1 2 . ランタイムコンピュータでのコマンド処理 〕

図 3 1 に、ランタイムコンピュータ 1 0 2 におけるコマンド実行の処理手順を示す。本実施の形態におけるコマンドとは、あらかじめランタイムコンピュータ 1 0 2 に組み込まれた、当該ランタイムコンピュータ 1 0 2 で実行可能な一連の処理であっても良いし、後述のディペンダビリティ記述データのノードに関連付けられた当該ランタイムコンピュータ 1 0 2 で実行可能なプログラムであっても良い。

【 0 2 6 2 】

まず、コマンド実行処理 S 1 1 0 0 の開始にあたって、処理が開始された旨が記録される（S 1 1 0 0 - 0 1）。次に、処理を 1 個以上のコマンド列に分解し、それをチェーンとして構成する（S 1 1 0 0 - 0 2）。つづいて、コマンドチェーン中の全コマンドに対して（S 1 1 0 0 - 0 3）、以下の S 1 1 0 0 - 0 4、S 1 1 0 0 - 0 5 の処理が実行される。

40

【 0 2 6 3 】

すなわち、チェーン中のコマンドを 1 つ実行し（S 1 1 0 0 - 0 4）、正常終了か、異常終了かを判断する（S 1 1 0 0 - 0 5）。そして、全部のコマンドが正常終了の場合には（S 1 1 0 0 - 0 5 で YES）、終了が記録され（S 1 1 0 0 - 0 7）、当該処理は終了する（S 1 1 0 0 - 1 4）。

【 0 2 6 4 】

一方、コマンドチェーン中のコマンドが異常終了した場合（S 1 1 0 0 - 0 6 で NO）、実行されたコマンドの効果を取り消すためのリカバリチェーンが構築される（S 1 1 0

50

0 - 08)。その後、リカバリチェイン中の全コマンドに対して(S 1 1 0 0 - 09)、リカバリコマンドを実行し(S 1 1 0 0 - 10)、正常終了か、異常終了かを判断する(S 1 1 0 0 - 11)。全リカバリコマンドが正常終了の場合(S 1 1 0 0 - 11でYES)には、その旨を記録して(S 1 1 0 0 - 07)、当該処理は終了する(S 1 1 0 0 - 14)。

【0265】

一方、リカバリコマンドの実行が異常終了の場合には(S 1 1 0 0 - 11でNO)、必要な隔離処理(S 1 1 0 0 - 13)を実行し、当該処理を含むコンテナ(アプリケーション・コンテナあるいはシステム・コンテナ)を隔離し、その旨を記録して(S 1 1 0 0 - 07)、当該処理は終了する(S 1 1 0 0 - 14)。その後、前記再構成処理を実行する。

10

【0266】

前記記載の手順は、図24に記載の基本動作をコマンドチェインとするように構成しても良い。これにより、前記スクリプトをアトミック操作として確実に実行できるようになる。

【0267】

なお、本発明は以下のように構成してもよい。

【0268】

本発明のディペンダビリティ維持装置(ディペンダビリティ維持システム100、ワークスペースコンピュータ101、ランタイムコンピュータ102)は、前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、下記の〔A〕〔B〕〔C〕に記載の手段群を、当該手段群が前記ディペンダビリティ記述データを備えるシステムにおいて確実に実行されることを保証する手段を有していても良い。〔A〕障害対応サイクルP301において、上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスの継続が不可能と判断された場合、変化対応サイクルP300において前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、前記障害対応サイクルP301において記録又は検出された情報から前記ディペンダビリティ記述データを備えるシステムのディペンダビリティ記述データの変更を前記変化対応サイクルP300に対する要求として当該変化対応サイクルP300に入力する手段。〔B〕前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、変化対応サイクルP300と障害対応サイクルP301との間での情報交換する手段。〔C〕ディペンダビリティ記述データから変換されたコンピュータで処理可能な形式のデータを、当該ディペンダビリティ記述データを備えるシステムが管理する手段を有することで、変化対応サイクルP300におけるステークホルダによる変更の要求の合意による結果を、確実に当該ディペンダビリティ記述データを備えるシステムに実施する手段。

20

30

【0269】

〔13. ノードへの実行可能プログラムの関連付け〕

図32は、ディペンダビリティ記述データに関連付けられたプログラムの処理内容の一例を示すフローチャートである。

40

【0270】

図32に示すノード(203-04)は図6に示したものと同一ノードである。当該ノードは「キャッシュによるApp Logicの継続」に関する。本実施の形態では、ディペンダビリティ記述データをランタイムコンピュータ102への組み込み手順に従って導入する。このとき、ディペンダビリティ記述データの各ノードには、ランタイムコンピュータ102のバイナリ実行部110Rで実行可能なプログラムを関連付けてもよい。また、モニタノードにも実行可能なプログラムが関連付けられて、ランタイムコンピュータ102に導入される。なお、ノードへのプログラムを関連付けは、外部参照情報104-04に記載できる。また、これらの処理について図32では当該プログラムの処理内容をフローチャートで表現したが、スクリプト言語を用いても良いし、バイナリ実行部110Rで直接

50

実行可能なプログラムコードを用いても良い。

【0271】

プログラム1200には導入処理S1200-01と廃止処理S1200-11の2個の処理の手順が定義されている。

【0272】

導入処理S1200-01は実行権限の確認(S1200-02)後に、アプリケーション・コンテナ提供部712(図25)の機能を利用してアプリケーション・コンテナを生成(S1200-03)する。次に、App Logic201-03(図4(b))に対応するアプリケーションを読み込み、アプリケーション管理部714(図25)に当該プログラムコードを登録する。

10

【0273】

一方、廃止処理S1200-11は実行権限の確認(S1200-12)後に、App Logic201-03に対応するアプリケーションを停止(S1200-13)し、アプリケーション管理部714から当該プログラムコードの登録を削除し、アプリケーション・コンテナ提供部712の機能を利用して当該アプリケーション・コンテナを削除する。

【0274】

図33は、ディペンダビリティ記述データに関連付けられたプログラムの処理内容の他の例を示すフローチャートである。

【0275】

図33に示すノード(203-01)は図7に示したものと同一ノードである。当該ノードは「多重型によるDBSの無停止」に関する。すなわち、バックアップ系を有するデータベースシステムを、複数台を同時に稼働させる多重系のデータベースに関するディペンダビリティ記述データの一部であった。

20

【0276】

プログラム1201には更新処理S1201-01、導入処理S1201-11、廃止処理S1201-21の3個の処理の手順が定義されている。ディペンダビリティ記述データにおけるノードが置き換えられるとき、当該ノードにプログラムが関連付けられている場合には、そこに定義された更新処理を実行する。

【0277】

具体的には、図33では、ノード(203-04)をノード(204-01)で置き換えるので、ノードの更新処理が実行される。更新処理S1201-01の手順は、実行権限の確認(S1201-01)後に、置き換えられるノード(図33の場合には図32のノード(203-04))に関連付けられた廃止処理S1200-11の手順を実行する。当該手順の実行に当たっては、図31のコマンド実行処理S1100のフローチャートに従い結果が記録(S1201-04)される。

30

【0278】

次に、導入処理S1201-11の手順は実行権限の確認(S1201-12)後に、システム・コンテナ提供部702(図25)の機能を利用してシステム・コンテナを作成する。次に、無停止DBSをインストール(S1201-14)し、アプリケーション管理部714(図25)に当該DBSを登録する。

40

【0279】

一方、廃止処理S1201-21の手順は実行権限の確認(S1201-21)後に、無停止DBSを停止(S1201-23)し、アプリケーション管理部714から当該DBSの登録を削除(S1201-24)し、当該DBSのシステム・コンテナをシステム・コンテナ提供部702の機能を利用して削除(S1201-25)する。

【0280】

なお、本発明は以下のように構成してもよい。

【0281】

本発明のディペンダビリティ維持装置(ランタイムコンピュータ102)は、前記ディペンダビリティ記述データを備えるシステムが障害発生検出時および障害予兆検出時の障

50

害対応サイクルP301を確実に実行するために、下記的手段群〔A〕〔B〕をコンピュータで処理するための表現手段を有していても良い。〔A〕上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスを障害発生に際しても継続するように対処するプロセスにおいて、障害の発生を検出する手段（モニタリング機能）と、障害の発生した部分を隔離する手段（隔離機能）と、隔離された残りの部分でサービスを継続する手段と、障害の原因を検出する手段（原因追及機能）と、上記システム運用時におけるシステムが提供するサービスの継続性を判断する手段（継続性判断機能）。〔B〕上記システム運用時における前記ディペンダビリティ記述データを備えるシステムが提供するサービスを停止させないように障害の予兆を検知し障害の発生を未然に回避するプロセスにおいて、前記ディペンダビリティ記述データを備えるシステムの内部状態を記録する手段（ロギング）と、当該記録された内部状態から前記ディペンダビリティ記述データを備えるシステムの障害の予兆を検出する手段と、前記ディペンダビリティ記述データを備えるシステムの当該検出された障害に対応する部分を同定する手段と、当該検出された障害が上記システム運用時におけるシステムが提供するサービスの継続性を判断する手段。

10

20

30

40

50

【0282】

また、本発明は以下のように構成してもよい。

【0283】

本発明のディペンダビリティ維持装置（ランタイムコンピュータ102）は、上記ディペンダビリティ記述データには上記再構成の手順が記載されており、上記再構成手段（再構成部102-04）は、上記ディペンダビリティ記述データに記載されている再構成の上記手順に従って、当該対象システムを再構成するように構成されても良い。

【0284】

これにより、ランタイムコンピュータ102は、ディペンダビリティ記述データに記載されている再構成の手順に従って、当該対象システムを再構成することができる。よって、再構成の手順をディペンダビリティ記述データの設計段階において、当該ディペンダビリティ記述データに記述しておくことができるため、再構成時に手順を作成する必要がない。

【0285】

〔14．合意形成ツールとの連携〕

図34に、ディペンダビリティ記述データM1と、ソフトウェアSWに含まれるXML形式ディペンダビリティ記述データM2（正確には、ディペンダビリティ記述データ内部表現M2であるが、ここでは、XML形式による内部表現を選択した場合について説明するので、「XML形式ディペンダビリティ記述データM2」と記す。）と、ワークスペースコンピュータ101が備えるツール群の各機能との関係を示す。

【0286】

編集機能1300-01（図1の編集ツール部101-02が備える機能）は、ディペンダビリティ記述データM1を入力・修正・記録する。検証機能1300-02（図1の検証ツール部101-01が備える機能）は、当該ディペンダビリティ記述データM1の妥当性を検証する。ベンチマーク機能1300-03（図1の検証ツール部101-01が備える機能）は、当該ディペンダビリティ記述データM1におけるエビデンスを確認する。変換機能1300-04（図1の編集ツール部101-02および開発ツール部101-04が備える機能）は、差分検証モデルM1をソフトウェアSWへの組込形式である、XML形式ディペンダビリティ記述データM2に変換する。当該変換の際には、ディペンダビリティ記述データからモニタノード部分を抜き出し、その構造を維持しながら変換するように本実施例を構成しても良い。計測機能1300-05（図1の検証ツール部101-01が備える機能）は、当該XML形式ディペンダビリティ記述データM2のモニタノードにアクセスしてデータを計測・抽出する。

【0287】

また、図35にベンチマーク機能（図中ではDS-Benchとして記載）と図16に記載のD-

Case Editorとの連携を示す。D-Case Editor上ではモニタノードに実行すべきベンチマークを指定できるように構成している。当該D-Case Editorからのベンチマーク実行指示に基づいて実際のベンチマークが実行され、その実行結果を反映することでエビデンスとして記録する。当該ベンチマーク結果はD値の算出に用いることができる。また、D値に影響のあるベンチマークを用意することも可能であり、その場合、D値への影響が大きいほどベンチマークの有効性も高いことになる。

【0288】

当該ベンチマークには、システムの性能に関する特性以外の、障害の原因になるような特性を検証するベンチマークが含まれていてもよい。このようなベンチマークの例としては、CPU負荷を変動させたときのシステムの振る舞いに関するベンチマーク、ネットワークの帯域幅を変動させたとき、または遅延を変動させたときのシステムの振る舞いに関するベンチマーク、余剰メモリを変動させたときのシステムの振る舞いに関するベンチマーク、過剰なアクセスをシミュレートすることでシステムに総合的に負荷をかけたときのシステムの振る舞いに関するベンチマーク、システムの構成コンポーネントを意図的に異常終了した際のシステムの振る舞いに関するベンチマーク、等が挙げられる。

10

【0289】

さらに、ディペンダビリティ記述データは、例えば図8に記載のD-Case記述を用いた場合、当該記述の整合性や網羅性等の検証のために、Agda定理証明器 (<http://wiki.portal.chalmers.se/agda/pmwiki.php>) を用いるように構成しても良い。これにより、当該D-Case記述の記述エラーを自動的に取り除くことが可能になる。

20

【0290】

なお、本発明は以下のように構成してもよい。

【0291】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101）は、前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、前記変化対応サイクルP300におけるステークホルダ間での合意を形成するステップにおいて、ステークホルダからの要求の変更をコンピュータで処理可能な表現形式で記述する手段を有し、当該表現形式の論理的検証手段を有していても良い。

30

【0292】

〔15.ディペンダビリティ維持システム間の連携〕

本実施の形態に係るディペンダビリティ維持システム100は、ワークスペースコンピュータ101、ディペンダビリティ記述データM1、ランタイムコンピュータ102から構成されると言える。そして、当該ディペンダビリティ維持システム100の計算可能なディペンダビリティは、当該ディペンダビリティ記述データM1が表現している。

【0293】

ここで、図36は、ディペンダビリティ維持システム100を2つ接続した構成例を示すブロック図である。

【0294】

図36に示すように、利用者のディペンダビリティ維持システム100Uに対してディペンダビリティ記述データに基づく機能を提供する提供者のディペンダビリティ維持システム100Sと、利用者の前記ディペンダビリティ維持システム100Uとを、ネットワーク1400-03で接続した複合システムが構成できる。

40

【0295】

ここで、図37は、図36に示した2つのディペンダビリティ維持システム100U、100Sを接続した構成における、独立した2つのディペンダビリティ維持システム100U、100S間での連携の一例を示す説明図である。

【0296】

提供者のディペンダビリティ維持システム100から、利用者のディペンダビリティ維持システム100Uに対して、SLA（Service Level Agreement）を含むSLAディベ

50

ンダビリティ記述データM2 - SLAが提示される。SLAディペンダビリティ記述データM2 - SLAは、SLAの情報を含む以外は、ディペンダビリティ記述データ内部表現M2と同じである。

【0297】

SLAディペンダビリティ記述データM2 - SLAに含まれるSLAには、ディペンダビリティ維持システム100Sにて公開されるインターフェースのディペンダビリティに関してのみ記述される。これは提供者が提示する情報であり、当該インターフェースに対しての提供者のディペンダビリティに関してのコミットメントとして捉えることができる。また、当該ディペンダビリティ記述データはD値を計算可能であり、利用者はD値を計算して提供者の公開インターフェースに関してのディペンダビリティが利用者の基準に合致するかを確認できる。

10

【0298】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101）は、前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、2台以上の当該システムがネットワークによって相互接続された環境において、相互接続されていない環境において備えることが可能な上述した各手段を有していてもよい。

【0299】

〔16. サプライチェーン〕

図38は、図1に示したディペンダビリティ維持システム100を2つ、本体側および部品側として統合した構成例を示すブロック図である。

20

【0300】

図38に示すように、機能の本体を実行する本体側のディペンダビリティ維持システム100Bに対して、前記機能の一部を部品側のディペンダビリティ維持システム100Pが部品として提供することが可能である。図38では、ディペンダビリティ記述データM - Bにおけるノードの1個（1500 - 03）が部品としての提供されている例を示す。部品側のディペンダビリティ維持システム100Pに示すディペンダビリティ記述データM - Pが、部品であるノード（1500 - 03）のディペンダビリティ記述データである。

【0301】

図39は、2つのディペンダビリティ維持システム100B，100Pを本体側および部品側として統合した構成における、独立した2つのディペンダビリティ維持システム100B，100P間での連携の一例を示す説明図である。

30

【0302】

本体側のディペンダビリティ維持システム100Bにおけるワークスペースコンピュータ101では、部品1500 - 03のディペンダビリティ記述データM - Pを検証する（S1502 - 01）。当該部品1500 - 03のディペンダビリティ記述データM - Pが妥当である場合には、本体側のディペンダビリティ維持システム100Bにおけるランタイムコンピュータ102で、当該部品1500 - 03のベンチマークを計測する（S1502 - 02）。その結果のエビデンスが十分であれば、本体側のディペンダビリティ維持システム100Bにおけるワークスペースコンピュータ101は当該部品1500 - 03を本体のディペンダビリティ記述データM - Bに統合する（S1502 - 05）。

40

【0303】

図40に、部品を統合する処理手順の詳細を示す。まず、当該部品1500 - 03のディペンダビリティ記述データM - Pを検証する（S1502 - 01）。次に、ベンチマークを実行してエビデンスを計測し（S1502 - 02）、さらに、D値を計算する（S1502 - 03）。その結果、D値が良好である場合には、統合可能であり（S1502 - 04でYES）、統合のための再構成を実行する（S1502 - 05）。一方、D値が満足できない場合には（S1502 - 04でNO）、部品提供者に対して部品（すなわち、そのディペンダビリティ記述データM - P）のアップデートを求める（S1502 - 06

50

)。以降、同様の手順を繰り返す。

【0304】

本発明のディペンダビリティ維持装置（ディペンダビリティ維持システム100、ワークスペースコンピュータ101）は、前記ディペンダビリティ記述データを備えるシステムが提供するサービスを継続させる目的で、当該システムが第三者の1又は複数のハードウェア又はソフトウェア部品で構成される場合において、各々の部品が前記ディペンダビリティ記述データを備え、当該システムのディペンダビリティ記述データとの互換性の検証手段を有していても良い。

【0305】

〔17.ディペンダビリティ記述データの表示〕

図41を用いてランタイムコンピュータ102に格納されたディペンダビリティ記述データにおけるゴール群の一部が満足されなかった状況下で、ワークスペースコンピュータ101上の図16に記載のD-Case Viewer（260-07）が当該状況を表示する例を示す。ここでは、満足されなくなったゴールがフラッシュ（1600-01）されており、必要に応じてオペレータにアテンションを求める（1600-02）。なお、図41はD-Case記述例であるが、記載内容は本説明では重要ではなく、その構造のみを示す。

10

【0306】

最後に、ディペンダビリティ維持システム100の各ブロック、特にワークスペースコンピュータ101およびランタイムコンピュータ102の各ブロックは、ハードウェアロジックによって構成してもよいし、次のようにCPUを用いてソフトウェアによって実現してもよい。

20

【0307】

後者の場合、ディペンダビリティ維持システム100、あるいはワークスペースコンピュータ101およびランタイムコンピュータ102は、各機能を実現するプログラムの命令を実行するCPU（central processing unit）、上記プログラムを格納したROM（read only memory）、上記プログラムを展開するRAM（random access memory）、上記プログラムおよび各種データを格納するメモリ等の記憶装置（記録媒体）などを備えている。そして、本発明の目的は、上述した機能を実現するソフトウェアであるディペンダビリティ維持システム100、あるいはワークスペースコンピュータ101およびランタイムコンピュータ102の制御プログラムのプログラムコード（実行形式プログラム、中間コードプログラム、ソースプログラム）をコンピュータで読み取り可能に記録した記録媒体を、上記のディペンダビリティ維持システム100、あるいはワークスペースコンピュータ101およびランタイムコンピュータ102に供給し、そのコンピュータ（またはCPUやMPU）が記録媒体に記録されているプログラムコードを読み出し実行することによっても、達成可能である。

30

【0308】

上記記録媒体としては、例えば、磁気テープやカセットテープ等のテープ系、フロッピー（登録商標）ディスク/ハードディスク等の磁気ディスクやCD-ROM/MO/MD/DVD/CD-R等の光ディスクを含むディスク系、ICカード（メモリカードを含む）/光カード等のカード系、あるいはマスクROM/EPROM/EEPROM/フラッシュROM等の半導体メモリ系などを用いることができる。

40

【0309】

また、上記のディペンダビリティ維持システム100、あるいはワークスペースコンピュータ101およびランタイムコンピュータ102を通信ネットワークと接続可能に構成し、上記プログラムコードを通信ネットワークを介して供給してもよい。この通信ネットワークとしては、特に限定されず、例えば、インターネット、イントラネット、エキストラネット、LAN、ISDN、VAN、CATV通信網、仮想専用網（virtual private network）、電話回線網、移動体通信網、衛星通信網等が利用可能である。また、通信ネットワークを構成する伝送媒体としては、特に限定されず、例えば、IEEE1394、USB、電力線搬送、ケーブルTV回線、電話線、ADSL回線等の有線でも、IrDAやリモコ

50

ンのような赤外線、Bluetooth（登録商標）、802.11無線、H D R、携帯電話網、衛星回線、地上波デジタル網等の無線でも利用可能である。なお、本発明は、上記プログラムコードが電子的な伝送で具現化された、搬送波に埋め込まれたコンピュータデータ信号の形態でも実現され得る。

【0310】

（1）本発明に係るディペンダビリティ維持装置は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置であって、対象システムのディペンダビリティに関する（ステークホルダ群によって合意された）要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得手段と、上記記述データ取得手段が取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定手段と、を備えることを特徴としている。

10

【0311】

（12）また、本発明に係るディペンダビリティ維持装置の制御方法は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持装置の制御方法であって、対象システムのディペンダビリティに関する（ステークホルダ群によって合意された）要求・仕様を記述したディペンダビリティ記述データを取得する記述データ取得ステップと、上記記述データ取得ステップにて取得したディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定ステップと、を含むことを特徴としている。

20

【0312】

上記の構成によれば、対象システムのディペンダビリティに関する（ステークホルダ群によって合意された）要求・仕様を記述したディペンダビリティ記述データに基づいて、対象システムのディペンダビリティの価値を定量的に示す評価値を求めることができる。

【0313】

それゆえ、対象システムのディペンダビリティの価値を定量的に表現できる。したがって、例えば、要求変更に伴ってディペンダビリティ記述データを変更するときや、対象システムの運用時に対象システムの状態を確認するときに、対象システムのディペンダビリティの価値を分かりやすく客観的に提示することができる。

【0314】

よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となるという効果を奏する。すなわち、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援することが可能となるという効果を奏する。

30

【0315】

（2）さらに、本発明に係るディペンダビリティ維持装置は、上記（1）に記載の構成において、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを（ステークホルダ群によって合意された）要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルの少なくとも何れか一方を実行することを特徴としている。

40

【0316】

上記の構成によれば、さらに、変化対応サイクルおよび障害対応サイクルの少なくともいずれか一方において、対象システムのディペンダビリティの価値を示す評価値を求めることができる。したがって、例えば、変化対応サイクルにおいて、要求変更に伴ってディペンダビリティ記述データを変更するとき、評価値に基づいて、ディペンダビリティ記述データの変更案の適否を判断することが可能となる。また、例えば、障害対応サイクルにおいて、対象システムの運用時に対象システムの状態を、評価値に基づいて判断することが可能となる。よって、対象システムのディペンダビリティの維持を円滑に行うことが可能となるという効果を奏する。

50

【0317】

(3) さらに、本発明に係るディペンダビリティ維持装置は、上記(1)に記載の構成において、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを(ステークホルダ群によって合意された)要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記障害対応サイクルを実行するとともに、上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、当該対象システムを再構成する再構成手段を備えることを特徴としている。

【0318】

上記の構成によれば、さらに、障害対応サイクルを実行するディペンダビリティ維持装置において、対象システムの障害発生あるいは障害予兆を検知したとき、ディペンダビリティ記述データに基づいて、当該対象システムを再構成して、対象システムの運用を継続させることができるという効果を奏する。なお、対象システムの障害発生または障害予兆を上記評価値に基づいて検知してもよい。例えば、上記評価値が所定の変動許容範囲より悪化したとき、あるいは、上記評価値の時間変化の傾向が所定の基準より悪化したとき、障害発生を検出するようにしても良い。

【0319】

(4) さらに、本発明に係るディペンダビリティ維持装置は、上記(3)に記載の構成において、上記ディペンダビリティ記述データには上記再構成の手順が記載されており、上記再構成手段は、上記ディペンダビリティ記述データに記載されている再構成の上記手順に従って、当該対象システムを再構成することを特徴としている。

【0320】

上記の構成によれば、さらに、ディペンダビリティ維持装置は、ディペンダビリティ記述データに記載されている再構成の手順に従って、当該対象システムを再構成することができる。よって、再構成の手順をディペンダビリティ記述データの設計段階において、当該ディペンダビリティ記述データに記述しておくことができるため、再構成時に手順を作成する必要がないという効果を奏する。

【0321】

(5) さらに、本発明に係るディペンダビリティ維持装置は、上記(3)または(4)に記載の構成において、上記障害発生または障害予兆を検知し、上記対象システムの停止が不回避であるとき、上記変化対応サイクルを実行する装置に、上記ディペンダビリティ記述データの変更の要求を送信する変更要求送信手段を備えることを特徴としている。

【0322】

上記の構成によれば、さらに、障害対応サイクルを実行するディペンダビリティ維持装置において、対象システムの障害発生または障害予兆を検知し、対象システムの停止が不回避であると判断したとき、変化対応サイクルを実行する装置に、ディペンダビリティ記述データの変更の要求を送信することができる。なお、障害発生または障害予兆により、対象システムの停止が不回避であるか否かを、上記評価値に基づいて判断しても良い。

【0323】

一方、変化対応サイクルを実行する装置は、上記ディペンダビリティ維持装置が送信したディペンダビリティ記述データの変更の要求を受信したとき、当該要求に応じて、上記ディペンダビリティ記述データを変更する。

【0324】

したがって、障害対応サイクルを実行するディペンダビリティ維持装置と、変化対応サイクルを実行する装置とが連携して、障害対応サイクルにおいて、対象システムの障害発生または障害予兆を検知し、対象システムの停止が不回避であれば、変化対応サイクルにおいて、ディペンダビリティ記述データを変更するという、一連のプロセスを円滑に行うことができる。よって、オープン環境において、対象システムのディペンダビリティを継続的に維持することが可能となるという効果を奏する。

10

20

30

40

50

【0325】

(6)さらに、本発明に係るディペンダビリティ維持装置は、上記(1)に記載の構成において、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを(ステークホルダ群によって合意された)要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記変化対応サイクルを実行するとともに、上記障害対応サイクルを実行する装置から、上記ディペンダビリティ記述データの変更の要求を受信したとき、上記ディペンダビリティ記述データを変更することを特徴としている。

10

【0326】

上記の構成によれば、さらに、変化対応サイクルを実行するディペンダビリティ維持装置において、障害対応サイクルを実行する装置から、ディペンダビリティ記述データの変更の要求を受信したとき、当該要求に応じて、上記ディペンダビリティ記述データを変更することができる。

【0327】

したがって、障害対応サイクルを実行する装置と、変化対応サイクルを実行するディペンダビリティ維持装置とが連携して、障害対応サイクルにおいて、対象システムの障害発生または障害予兆を検知し、対象システムの停止が不回避であれば、変化対応サイクルにおいて、ディペンダビリティ記述データを変更するという、一連のプロセスを円滑に実行することができる。よって、オープン環境において、対象システムのディペンダビリティを継続的に維持することが可能となるという効果を奏する。

20

【0328】

(7)さらに、本発明に係るディペンダビリティ維持装置は、上記(1)から(6)のいずれかに記載の構成において、上記ディペンダビリティ記述データは、互いに関連づけられたゴールノードとモニタノードとの組を規定するデータであり、上記ゴールノードは、(ステークホルダ群によって合意された)要求・仕様がゴール形式にて記述されたノードであり、上記モニタノードは、上記ゴールノードに記述されたゴールが充足されていることを表明するエビデンスであるとともに、上記対象システムの監視点に対応付けられており、上記ディペンダビリティ値決定手段は、対応付けられた上記監視点から取得されたデータが変動許容範囲に対して良好であるモニタノードに基づいて、上記評価値を計算することを特徴としている。

30

【0329】

上記の構成によれば、さらに、互いに関連づけられたゴールノードとモニタノードとの組を規定するデータ、例えば、木構造あるいはグラフ構造のデータから、多次元ベクトル値として評価値を求めることができる。

【0330】

また、本発明に係るディペンダビリティ維持装置は、上記(1)から(6)のいずれかに記載の構成において、

・上記ディペンダビリティ記述データは、ゴールノードとモニタノードとを組にすることが可能であるとともに、当該組を1以上有する木構造あるいはグラフ構造を有するデータであって、

40

・上記ゴールノードは、(ステークホルダ群によって合意された)要求・仕様がゴール形式にて記述することが可能なノードであり、

・上記モニタノードは、上記ゴールノードに記載のゴールが満たされていることを表明するエビデンスであり、

さらに、上記モニタノードは、上記対象システム内部の対応した監視点からのデータ取得を可能にした上で、当該監視点データが変動許容範囲に対して良好であることを判断する手段を備えており、

上記ディペンダビリティ値決定手段は、上記エビデンスが良好であるモニタノードを上

50

記木構造あるいはグラフ構造において計算することで上記評価値とすることを特徴としている。

【0331】

また、上記ディペンダビリティ維持装置は、上記ディペンダビリティ記述データがノード間のエッジに重み付けが可能な木構造あるいはグラフ構造を有するデータであり、上記ディペンダビリティ値決定手段は、上記エッジに重み付けされた重みを要素とする多次元ベクトル値を上記評価値としてもよい。

【0332】

上記の構成によれば、さらに、木構造あるいはグラフ構造のエッジに重み付けされた重みを要素とする多次元ベクトル値を評価値として求めることができる。

10

【0333】

それゆえ、木構造あるいはグラフ構造に、エッジの重要性を加味した評価値が得られる。よって、評価値を確認することで、エッジに重み付けの変更がディペンダビリティに与える影響を評価することができるという効果を奏する。なお、ここで得られる評価値は多次元ベクトル値であるが、目的に応じた任意の変換手法によりスカラー値に変換できることはいうまでもない。

【0334】

(8)さらに、本発明に係るディペンダビリティ維持装置は、上記(1)から(6)のいずれかに記載の構成において、上記ディペンダビリティ値決定手段は、上記対象システムをモニタして取得されたモニタ値が変動許容範囲に対して良好であったエビデンスに基づいて、上記評価値を計算することを特徴としている。

20

【0335】

上記の構成によれば、さらに、対象システムをモニタして取得されたモニタ値が変動許容範囲(基準値)に対して良好であったエビデンスに基づいて、評価値を計算することができる。その計算方法としては、例えば、ディペンダビリティ記述データに含まれるエビデンスの総数(総エビデンス数)に対する、モニタ値が変動許容範囲(基準値)に対して良好なエビデンスの数(有効エビデンス数)の割合を評価値として計算することができる。

【0336】

それゆえ、エビデンスの状態に応じた評価値が得られる。エビデンスは要求変更に応じて変化するため、ここで得られる評価値も要求変更に応じて変化する。すなわち、評価値を確認することで、要求変更がディペンダビリティに与える影響を評価することができるという効果を奏する。

30

【0337】

(9)さらに、本発明に係るディペンダビリティ維持装置は、上記(1)に記載の構成において、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを(ステークホルダ群によって合意された)要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルのうち、少なくとも上記障害対応サイクルを実行するとともに、上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに含まれるスクリプトを実行するスクリプト処理手段を備え、上記スクリプトは上記対象システムを変動許容範囲の状態に回復させるシナリオを含むことを特徴としている。

40

【0338】

上記の構成によれば、さらに、障害発生または障害予兆を検知したとき、ディペンダビリティ記述データに含まれるスクリプトを実行することによって、対象システムを変動許容範囲の状態に回復させることができる。

【0339】

なお、上記スクリプトは、ステークホルダによって合意されていることが好ましい。また、上記スクリプトは、対象システムに組み込まれたスクリプトエンジンによって実行さ

50

れる。また、上記スクリプトは、対象システム内部で取得されているログから、対象システムの状態が変動許容範囲にあるかを判断してもよい。また、対象システムを変動許容範囲の状態に回復させるシナリオを実行する際、当該シナリオは、GUIを介してオペレータから操作を受け付けてもよい。

【0340】

(10)また、本発明に係るディペンダビリティ維持システムは、対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、上記対象システムの開発時あるいは更新時に、上記ディペンダビリティ記述データを(ステークホルダ群によって合意された)要求・仕様の変更に合わせて変更する変化対応サイクルを実行するワークスペース装置と、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行するランタイム装置とを含み、上記ワークスペース装置および上記ランタイム装置の少なくともいずれか一方が、上記ディペンダビリティ記述データに基づいて、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めることを特徴としている。

10

【0341】

(11)さらに、本発明に係るディペンダビリティ維持システムは、上記(10)に記載の構成において、他のディペンダビリティ維持システムとネットワークを介して接続されていることを特徴としている。

【0342】

なお、上記のディペンダビリティ維持装置は、コンピュータによって実現してもよく、この場合には、コンピュータを上記の各手段として動作させることにより上記ディペンダビリティ維持装置をコンピュータにて実現させるディペンダビリティ維持装置のプログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

20

【0343】

〔実施の形態2〕

本実施の形態に係るディペンダビリティ維持装置20は、要求・仕様書およびシステムの更新の間の無矛盾性を維持することを容易にするものである。そのため、後述するように、本実施の形態に係るディペンダビリティ維持装置20は、監視制御対象システム2のディペンダビリティに関する仕様を記述したD-CaseをD-Case格納部10から読み出し、監視制御対象システム2の状態を監視し、必要な場合に対策を実行する障害監視制御部30の動作を制御する監視制御データを、上記読み出したD-Caseから生成し、障害監視制御部30に供給するD-Case変換部21を備える。

30

【0344】

以下、図42から図56、および図8に基づいて、本発明の一実施の形態について詳細に説明する。

【0345】

図42は、本実施の形態に係る障害対応システム1の構成の概略を示す機能ブロック図である。

40

【0346】

本実施の形態に係る障害対応システム1は、監視制御対象システム2の動作を監視し、障害が発生した場合に、必要な対応措置を行うものである。なお、監視制御対象システム2としては、任意のコンピュータシステムに適用可能であるが、特に、単一のシステムではなく、複数のステークホルダが存在する、高度で複雑なディペンダビリティが必要とされているシステムに好適である。具体的には、社会インフラシステムの基盤であるようなコンピュータシステム、例えば、監視システム、電子決済システム、交通・航空管制システム、さらには、それらを包含するクラウドシステムに好適である。

【0347】

そして、特に、本実施の形態に係る障害対応システム1は、監視制御対象システム2の

50

各ステークホルダの合意の記述を容易にするとともに、監視制御対象システム 2 の一部更新があった場合に、合意の記述と監視制御のための実装モジュールの間が無矛盾に更新されるように維持することを特徴とするものである。すなわち、障害対応システム 1 は、ディペンダビリティ記述データで表現された監視制御対象システム 2 のディペンダビリティに関わる仕様と、監視制御対象システム 2 を監視制御するためのモニタモジュールやアクションモジュールの制御との間を、無矛盾に開発・更新することを可能とする。

【 0 3 4 8 】

そのために、障害対応システム 1 は、ディペンダビリティ記述データとして、各ステークホルダの合意の記述を容易に行うことができるという特徴を持つ D-Case を用いて、監視制御対象システム 2 のディペンダビリティを記述する。また、障害対応システム 1 は、D-Case のパターン (D-Case パターン) と監視制御対象システム 2 を監視制御するためのモニタモジュールおよびアクションモジュールとの対応を管理する。これにより、監視制御対象システム 2 の変更等に応じて、モニタモジュールやアクションモジュールが追加・変更されても、ディペンダビリティ記述とモニタモジュールおよびアクションモジュールとの対応を維持することができる。

10

【 0 3 4 9 】

(D-Case)

図 8 は、D-Case の基本構造を示す説明図である。また、図 4 3 は、D-Case の具体例を示す説明図である。なお、図 4 3 は、顔認識を用いた監視システムを監視制御対象システム 2 とした場合の D-Case である。

20

【 0 3 5 0 】

上記の通り、本実施の形態では、監視制御対象システム 2 のディペンダビリティに関する仕様を記述するディペンダビリティ記述データとして、D-Case を使用する。

【 0 3 5 1 】

D-Case とは、Safety Case という、主にイギリスで用いられている、システムの安全性を保証するために用いられる構造化されたドキュメントを元としている。本明細書では、Safety Case を元にしたディペンダビリティ記述データを「D-Case」と称するものとする。D-Case によれば、可用性などの一般的なディペンダビリティ属性は、図 4 3 のようなシステムレベルの議論に詳細化される。

30

【 0 3 5 2 】

D-Case は、監視制御対象システム 2 に関わる各ステークホルダが合意するための構造化されたドキュメントである。詳細には、図 8 に示すように、D-Case は、システムのディペンダビリティをトップゴールとして、それが満たされていることを木構造により詳細化していき、詳細化されたゴールに対してエビデンスを置いた木構造を有する。D-Case は、後述するように X M L (Extensible Markup Language) によって記述することができる。

【 0 3 5 3 】

ここで、図 4 3 中、“G” を付したブロックがトップゴールである。トップゴールとは、対象システムに対して、ステークホルダ間で合意すべき命題を表す。例えば「対象システムは、安全性規格 X で定義されている Safety Integrity Level 3 を満たす」などである。

40

【 0 3 5 4 】

“SG” を付したブロックがサブゴールである。サブゴールとは、トップゴールを示すために、示すべきことを分割した命題を表す。例えば、トップゴールが「A である、かつ、B である」であるとき、サブゴール「A である」、「B である」に分割する。サブゴールは、さらに小さなサブゴールに分割されうる。トップゴール、サブゴールを総称して、ゴールと呼ぶ。

【 0 3 5 5 】

“St” を付したブロックがストラテジである。ストラテジとは、ゴールが成り立つことを、そのサブゴール群によってどのように議論されるのかの補助説明を示す。例えば、ゴールが「システムは考えられる障害すべてに対応する」であるとき、ストラテジとして

50

、「考えられる障害ごとに議論する」をあげる。この場合、サブゴールは、「システムは考えられる障害 1 に対応する」、…、「システムは考えられる障害 N に対応する」となる。

【0356】

“E”を付したブロックがエビデンスである。エビデンスとは、分割され、詳細化されたゴールを、最終的に保証するリーフノードである。例えば、サブゴールSG「システムXのサブコンポーネントYは、障害Zに対処できる」というゴールに対して、「FTA解析結果」というエビデンスをサブゴールSGの直下におく。サブゴールSGをさらにサブゴールに分割することも考えられるが、無限にゴールを分割することはできない。分割されたサブゴールを最終的に保証するものとして、エビデンスがおかれる。エビデンスの妥当性は、ステークホルダ間の合意に基づく。

10

【0357】

また、図47に示されるように、“C”を付したブロックがコンテキストである。コンテキストは、ゴールやストラテジの内容を補足する情報を示す。例えば、ストラテジ「考えられる障害ごとに議論する」である場合、コンテキストとして「考えられる障害のリスト」がそのストラテジに付けられうる。ストラテジやコンテキストは、D-Caseを読むステークホルダが、ゴールの分割を追いながら読むときの助けとなる補助情報を示す。

【0358】

なお、ストラテジ、コンテキストは、木構造化するときには読みやすさのために記入するが、ランタイム時にはコメント扱いである。

20

【0359】

また、図50に示される、“Monitor:M_1”等のブロックがモニタである。モニタとは、システムが障害対応を行う際に必要になるランタイム時のシステムの情報を表す。例えば、ランタイム時の「プロセスXのCPU使用率のモニタリング」などがモニタノードとなる。モニタはエビデンスのサブクラスである。障害対応システム1のモニタ管理部40により得られる情報をエビデンスとして用いる場合に、そのエビデンスをモニタノードとして表す。

【0360】

つづいて、図42および図44を参照しながら、上記障害対応システム1の構成および処理フローについて詳細に説明する。

30

【0361】

図42に示すように、障害対応システム1は、D-Case格納部（ディペンダビリティ記述格納部）10、ディペンダビリティ維持装置20、障害監視制御部30、モニタ管理部40、モニタモジュール群50、アクション管理部60、アクションモジュール群70を備えて構成されている。

【0362】

上記D-Case格納部10は、監視制御対象システム2についての、ステークホルダ間のディペンダビリティに関する合意を記述したディペンダビリティ記述データであるD-Caseを格納する。なお、障害対応システム1では、各ステークホルダが合意したD-Caseが入力となる。

40

【0363】

上記ディペンダビリティ維持装置20は、監視制御対象システム2を監視制御するための実装モジュール（モニタモジュール、アクションモジュール）の制御とD-Caseパターンとの対応を示すD-Caseパターン<=>モジュール対応テーブル（後述する）を使用して、D-Caseから監視制御データ（障害対応スクリプト）を生成する。そのために、ディペンダビリティ維持装置20は、D-Case変換部（ディペンダビリティ記述変換部）21、対応テーブル格納部22を備えて構成されている。

【0364】

上記D-Case変換部21は、障害監視制御部30の動作を制御する監視制御データ（障害監視制御用データ）を、D-Case格納部10に格納されているD-Caseから生成する。

50

【0365】

ここで、D-Caseは、その一部分がパターン化されている。そして、上記対応テーブル格納部22は、D-Caseパターンと監視制御データとの対応を表わすテーブル(D-Caseパターン<=>モジュール対応テーブル)を格納している。

【0366】

D-Case変換部21は、対応テーブル格納部22に格納されているD-Caseパターン<=>モジュール対応テーブルを参照して、D-Caseから監視制御データを生成する。

【0367】

つぎに、上記障害監視制御部30は、監視制御対象システム2の状態を監視(モニタ)し、必要な場合に対策(アクション)を実行する。具体的には、障害監視制御部30は、ディペンダビリティ維持装置20においてD-Case変換部21が生成した監視制御データに従って、モニタ管理部40とアクション管理部60を制御する。監視制御データは、モニタ管理部40によるモニタモジュール(モニタモジュール群50)の選択および動作の制御、並びに、アクション管理部60によるアクションモジュール(アクションモジュール群70)の選択および制御を規定している。

10

【0368】

上記モニタ管理部40は、監視制御データに従って、1以上のモニタモジュール(モニタモジュール群50)を管理する。本実施の形態では、モニタモジュールの例として、CPU監視モジュール51、メモリ監視モジュール52を挙げるが、本発明はこれに限定されない。

20

【0369】

上記アクション管理部60は、監視制御データに従って、1以上のアクションモジュール(アクションモジュール群70)を管理する。本実施の形態では、アクションモジュールの例として、CPU制限モジュール71、メモリ制限モジュール72、プロセス死活モジュール73を挙げるが、本発明はこれに限定されない。

【0370】

図44は、障害対応システム1の処理を示すフローチャートである。

【0371】

まず、D-Case管理者が、ステークホルダ間の合意のもとで作成、変更されたD-Caseを、D-Case格納部10に格納する(S1;D-Case格納ステップ)。

30

【0372】

次に、D-Case変換部21が、D-Case格納部10からD-Caseを読み出し(S2;D-Case読み出しステップ)、読み出したD-Caseから、対応テーブル格納部22に格納されているD-Caseパターン<=>モジュール対応テーブルを参照して、障害監視制御部30の動作を制御する監視制御データを生成する(S3;D-Case変換ステップ)。

【0373】

なお、D-Caseの変更に伴って監視制御データを生成する場合には、D-Case変換部21は、変更されたD-Caseの差分のみに対応する監視制御データを作成してもよい。

【0374】

次に、障害監視制御部30が、作成された監視制御データに従って、監視制御対象システム2の状態を監視(モニタ)し、必要な場合に対策(アクション)を実行する(S4;障害監視実行ステップ)。

40

【0375】

ここで、上記障害監視実行ステップ(S4)について説明する。

【0376】

まず、モニタ管理部40には、管理すべきモニタモジュールがあらかじめ制御可能に登録されている。また同様に、アクション管理部60には、管理すべきアクションモジュールがあらかじめ制御可能に登録されている。

【0377】

そして、障害監視実行ステップ(S4)では、障害監視制御部30が、監視制御データ

50

に従ってモニタ管理部 40 に適切なモニタモジュールを起動するよう指示する。そして、障害監視制御部 30 は、モニタ管理部 40 から通知されたモニタモジュールの実行結果が条件を満たす場合、さらに、アクション管理部 60 に適切なアクションモジュールを起動して障害に対処するよう指示する。このとき、モニタ管理部 40 およびアクション管理部 60 は、それぞれ、障害監視制御部 30 からの指示に従ってモジュールを起動し、適切な引数を渡す。

【0378】

(D-Case変換)

つぎに、図 45 ~ 図 49 を参照しながら、D-Caseから監視制御データへの変換処理について詳細に説明する。

10

【0379】

障害対応システム 1 では、D-Caseから、監視制御対象システム 2 を監視制御するためのモニタモジュールおよびアクションモジュールの動作を制御する監視制御データへの変換に、D-Caseパターン<=>モジュール対応テーブルを用いる。

【0380】

ここで、D-Caseパターンとは、監視制御対象システム 2 のD-Caseの一部であり、可用性などが、システムレベルでどのように維持されるかを示すものである。すなわち、システムの可用性とは、システムが障害などでサービスを行えない状態になったとき（システムダウン状態）、できるだけ早急に復旧し、利用者がサービスを用いたい時にいつでも用いられるようにできるシステムの性質である。障害が発生した場合、保守管理者などの人員により迅速に行うこと、およびシステム自体が自動的に障害復旧機能により復旧を行う。一般にシステムは、OSレベルにおいては、CPU資源、メモリ資源、ネットワーク資源などを用いて、サービスを提供する。したがって、CPU資源などが不足した場合、サービスの遅延などが発生し、可用性が減じる可能性がある。よって、あるサービスに用いるCPU資源が減じた場合、他の重要度の低いサービスで用いられているCPU資源を転用するなどの対応をシステムはOSレベルで自動的に行う。監視制御対象システム 2 のD-Caseは、上記観点から、システムレベルでの可用性について議論している。

20

【0381】

(D-Caseパターン)

図 45 は、D-Caseパターンを含むD-Caseの具体例を示す説明図である。

30

【0382】

D-Caseパターンとは、D-Caseの部分木であり、可変な部分を含むものである。図 45 はパターンの一例であり、プロセスの属性が適切に保たれることを表現する場合に使用される。サブゴールSG22とエビデンスE25、E26の「」の部分が可変（変数）であり、他は固定されたD-Caseパターンである。図 45 は、「」内に既に「画像処理」、「CPU使用率」、「50%以下」を代入した状態を示している。このD-CaseパターンがD-Caseの一部として使用される場合に、「」部分にはステークホルダ間の合意に基づく値が代入され、その代入される値により、モニタモジュールやアクションモジュールと適切に関連づけることが可能となる。また、他は固定されたD-Caseである。D-Caseパターン、すなわちD-Caseの「」部分には、ステークホルダ間の合意に基づく値が代入される。また、D-Caseの「」部分に値が代入されることにより、モニタモジュールやアクションモジュールと適切に関連づけることが可能となる。

40

【0383】

図 46 は、D-Caseパターン<=>モジュール対応テーブルの一例であって、(a) はモニタモジュールについての対応テーブルを示し、(b) はアクションモジュールについての対応テーブルを示す。

【0384】

図 46 (a) は、プロセス「\$1」の「\$2」が「\$3」であることをモニタするというD-Caseを、D-Case変換部 21 が変換するときに参照するテーブルである。このD-Caseでは、\$1、\$2、\$3が変数である。

50

【0385】

例えば、図45のD-Caseを処理する際、D-Case変換部21は、エビデンスE25の記述プロセス「画像処理」の「CPU仕様率」が「50%以下」であることをモニタするに当てはめることができるテーブル(図46(a))を対応テーブル格納部22から読み出す。そして、エビデンスE25の記述から抽出した変数\$1、\$2、\$3の値を用いて、D-Caseパターンとモジュールとの対応関係を特定し、監視制御データを作成する。具体的には、変数\$2が「CPU仕様率」であることから、モジュール名「CPU監視」および引数“\$1、\$3”を特定し、「CPU監視」モジュールを呼び出し、「画像処理」「50%以下」を引数として渡すという内容の監視制御データを作成する。

【0386】

同様に、D-Case変換部21は、エビデンスE26の記述プロセス「画像処理」の「CPU仕様率」を「50%以下」に制限するに当てはめることができるテーブル(図46(b))を対応テーブル格納部22から読み出す。そして、変数\$2が「CPU仕様率」であることから、モジュール名「CPU制限」および引数“\$1、\$3”を特定し、「CPU制限」モジュールを呼び出し、「画像処理」「50%以下」を引数として渡すという内容の監視制御データを作成する。

【0387】

上記のように、エビデンスの記述のうち、固定されたD-Caseの文字列(記述の固定部分)に応じて、D-Caseパターン<=>モジュール対応テーブルが選択される。そして、一部の変数(上記の例では、変数\$2)の値に応じて、対応するモジュールが選択される。

【0388】

このように、本実施の形態では、エビデンスの記述を一部に変数が設定された表現形式とし、そのエビデンスに則した値を変数に設定して、エビデンスを記述する。

【0389】

つぎに、図47、図48を参照して、図45のD-Caseが変更された場合の処理について説明する。

【0390】

図47は、D-Caseパターンを含むD-Caseの具体例を示す説明図である。

【0391】

図47では、図45に対して、コンテキストC31が追加されるとともに、エビデンスE26がエビデンスE32に変更されている。これは、C31:他のプロセス「顔認識」を制限してでも「画像処理」にCPUを十分与えたいため、E32:プロセス「顔認識」を「kill」することで、各ステークホルダが合意したことを示している。

【0392】

図48は、このときのD-Caseパターン<=>モジュール対応テーブルの一例である。図48(a)はモニタモジュールについての対応テーブルを示し、図48(b)はアクションモジュールについての対応テーブルを示す。図48(a)は、図46(a)と同じである。

【0393】

そして、図47のD-Caseを処理する際、D-Case変換部21は、エビデンスE32の記述プロセス「顔認識」を「kill」するに当てはめることができるテーブル(図48(b))を対応テーブル格納部22から読み出す。そして、エビデンスE32の記述から抽出した変数\$1、\$2の値を用いて、D-Caseパターンとモジュールとの対応関係を特定し、監視制御データを作成する。具体的には、変数\$2が「Kill」であることから、モジュール名「プロセス死活」および引数“\$1”を特定し、「プロセス死活」モジュールを呼び出し、「Kill」「顔認識」を引数として渡すという内容の監視制御データを作成する。

【0394】

このように、D-Caseパターン<=>モジュール対応テーブルには、引数として、変数だけでなく、定数値を設定しておくこともできる。すなわち、図48(b)では、変数\$2の値が「Kill」であれば、プロセス死活モジュール73へ渡す1番目の引数は「Kill」となり

10

20

30

40

50

、変数\$2の値が“Restart”であれば、渡す引数は“Restart”となる。また、変数\$2の値が“Migration”であれば、プロセス死活モジュール73ではなく、Migrationモジュール（図示せず）がアクションとして実行されることになる。

【0395】

つぎに、図49を参照して、新たにモニタモジュールやアクションモジュールを追加した場合の処理について説明する。

【0396】

新たにモニタモジュールやアクションモジュールを追加した場合には、モニタ管理部40やアクション管理部60に登録する前に、D-Case管理者が、D-Caseパターン<=>モジュール対応テーブルに対応関係を規定するデータ（変換規則）を追加する。

10

【0397】

図49は、D-Caseパターン<=>モジュール対応テーブルの一例であって、モニタモジュールについての対応テーブルを示す。例えば、ネットワーク転送量を監視するモジュールを新たに追加した場合、図49のように、図46(a)に1行追加される。

【0398】

（まとめ）

以上のように、上記障害対応システム1では、モニタ管理部40によるモニタモジュールの選択および動作の制御、並びに、アクション管理部60によるアクションモジュールの選択および制御と、D-Caseパターンとの対応を示すD-Caseパターン<=>モジュール対応テーブルを使用して、D-Caseから監視制御データを生成する。すなわち、D-Case格納部10に格納されたD-Caseは、D-Caseパターン<=>モジュール対応テーブルを参照しながら、障害監視制御部30の監視制御データに変換される。このように、障害監視制御部30が用いる監視制御データが、D-Case部10に格納されたD-Caseから、D-Case変換部21によって生成されることにより、D-Caseと障害監視制御部30の動作との間が常に無矛盾になるように維持される。

20

【0399】

また、D-Case格納部10に格納されているD-Caseを変更した場合、その変更はD-Case変換部21によって監視制御データの変更として障害監視制御部30に通知される。なお、D-Caseの変更をディペンダビリティ維持装置20あるいはD-Case変換部21が検知し、変更後の監視制御データを障害監視制御部30に自動的に通知するようにしてもよい。

30

【0400】

また、D-Case管理者は、D-Caseに新たにD-Caseパターンを追加した場合、D-Caseパターン<=>モジュール対応テーブルに対応関係を規定するデータ（変換規則）を追加する。

【0401】

また、モニタモジュールやアクションモジュールを修正・追加した場合、障害対応システム1はD-Case管理者に通知する。D-Case管理者は、修正・追加されたモニタモジュールやアクションモジュールに対応するD-Caseパターン<=>モジュール対応テーブルも合わせて修正する。なお、修正・追加されたモニタモジュールやアクションモジュールに対応するD-Caseパターン<=>モジュール対応テーブルが修正されない場合、モニタモジュールあるいはアクションモジュールがD-Caseと対応できてないので、監視制御データが使用できない。修正されたD-Caseパターン<=>モジュール対応テーブルを使用して監視制御データを修正することによって、D-Caseと障害監視との間の無矛盾性が維持される。

40

【0402】

また、D-Caseパターン<=>モジュール対応テーブルは、データベースに蓄積しておき、D-Caseを変更する際、適切なD-Caseパターン<=>モジュール対応テーブルを参照するように、データベースから抽出して利用するようにしてもよい。これにより、D-Caseの変更の度にD-Caseパターン<=>モジュール対応テーブルを作成する必要がなくなり、低コストでD-Caseの変更に対応することが可能となる。

【0403】

つづいて、図50～図56を参照しながら、他の具体例について説明する。

50

【0404】

図50および図51は、D-Caseの他の具体例を示す説明図である。図50と図51は連結部Aで連結されて、一つの木構造となる。また、図52～図55は、図50、図51に示したD-CaseをXML形式で記述した例を示す説明図である。なお、図52～図55には、図50、図51に示したD-Caseから一部を抜粋して示している。図56は、図50、図51、および図52～図55に示したD-Caseから変換した監視制御データ（障害対応スクリプト）の例を示す説明図である。

【0405】

図52～図55に示すように、本具体例では、D-CaseのXMLファイルに障害監視制御部30を制御する障害対応スクリプトが埋め込まれている。そして、図56が、D-CaseのXMLファイル（図52～図55）の「Monitor:M_1」～「Monitor:M_5」からそれぞれ抽出されたスクリプトである。なお、図50および図51は、図52～図55をグラフィカルに表示したものであり、データには障害対応スクリプトも含まれている。

10

【0406】

このように、D-Caseに障害対応スクリプトが埋め込まれている点が、図45～図49を参照しながら説明した、D-Caseパターン<=>モジュール対応テーブル具体例とは異なっている。しかしながら、ステークホルダの合意を示すD-Caseと、監視制御対象システムの障害対応スクリプトとの間を無矛盾に更新することができる点では同じである。

【0407】

本発明は以下のように構成してもよい。

20

【0408】

本発明のディペンダビリティ維持装置20は、監視制御対象システム2の状態を監視し、必要な場合に対策を実行する障害監視制御部30の動作を制御する障害監視制御用データ（監視制御データ）を生成し、上記障害監視制御部30に供給するディペンダビリティ維持装置20であって、上記監視制御対象システム2のディペンダビリティに関する仕様を記述したディペンダビリティ記述データ（D-Case）をディペンダビリティ記述格納部（D-Case格納部10）から読み出し、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成するディペンダビリティ記述変換部（D-Case変換部21）を備えて構成されていてもよい。

【0409】

また、本発明のディペンダビリティ維持装置20の制御方法は、監視制御対象システム2の状態を監視し、必要な場合に対策を実行する障害監視制御部30の動作を制御する障害監視制御用データを生成し、上記障害監視制御部30に供給するディペンダビリティ維持装置20の制御方法であって、上記監視制御対象システム2のディペンダビリティに関する仕様を記述したディペンダビリティ記述データ（D-Case）をディペンダビリティ記述格納部（D-Case格納部10）から読み出す読み出しステップ（S2）と、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成する変換ステップ（S3）とを含む方法であってよい。

30

【0410】

さらに、上記ディペンダビリティ維持装置20は、上記障害監視制御部30は、上記監視制御対象システム2の状態を監視し、必要な場合に対策を実行するために使用するモジュールを、複数のモジュール（モジュラモジュール群50、アクションモジュール群70）から選択して制御可能であり、上記ディペンダビリティ記述データは、1つの記述（エビデンス）が、上記モジュールを特定するためのモジュール特定情報（変数\$2；パターン）を値として設定可能な変数を少なくとも含む形式であり、上記ディペンダビリティ記述変換部は、あらかじめ設定された、上記モジュール特定情報と上記障害監視制御用データとの対応関係を示す情報（D-Caseパターン<=>モジュール対応テーブル）に基づいて、上記ディペンダビリティ記述データに含まれる変換対象の記述を、該記述に含まれるモジュール特定情報に対応する障害監視制御用データに変換するように構成されていてもよい。

40

【0411】

50

さらに、上記ディペンダビリティ維持装置 20 は、上記ディペンダビリティ記述データの 1 つの記述から変数部分を除いた固定部分に対応して、上記対応関係が設定されており、上記ディペンダビリティ記述変換部は、上記ディペンダビリティ記述データの記述に含まれる上記固定部分に対応する上記対応関係を参照して、該記述に対応する障害監視制御用データに変換するように構成されていてもよい。

【0412】

また、本発明の障害対応システム 1 は、上記のディペンダビリティ維持装置 20 と、上記ディペンダビリティ記述格納部 (D-Case 格納部 10) と、上記障害監視制御部 30 とを含み、上記ディペンダビリティ維持装置 20 が上記ディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから生成した障害監視制御用データに従って、上記障害監視制御部 30 が動作することにより、上記監視制御対象システム 2 の状態を監視し、必要な場合に対策を実行するように構成されていてもよい。

10

【0413】

さらに、上記障害対応システム 1 は、上記監視制御対象システム 2 の状態を監視する 1 以上のモタモジュール (モタモジュール群 50) と、必要な場合に上記監視制御対象システム 2 に対して対策を実行する 1 以上のアクションモジュール (アクションモジュール群 70) と、上記障害監視制御部 30 の制御により、上記モタモジュールの選択および動作の制御を行うモタ管理部 40 と、上記障害監視制御部 30 の制御により、上記アクションモジュールの選択および動作の制御を行うアクション管理部 60 と、を含んで構成されていてもよい。

20

【0414】

また、本発明は以下のように構成してもよい。

【0415】

本発明に係るコンピュータシステム (障害対応システム 1) は、ステークホルダ間のディペンダビリティに関する合意を記述したディペンダビリティ記述データ (D-Case) を格納するディペンダビリティ記述格納部 (D-Case 格納部 10) と、対象システム (監視制御対象システム 2) 内の状態を監視 (モニタ) し必要な場合に対策 (アクション) を実行する障害監視制御部 (障害監視制御部 30) と、障害監視制御部の動作を制御する障害監視制御用データ (監視制御データ) をディペンダビリティ記述データから生成するディペンダビリティ記述変換部 (D-Case 変換部 21) とを持ち、障害監視制御用データが常にディペンダビリティ記述格納部に格納されたディペンダビリティ記述データからディペンダビリティ記述変換部によって生成されることによりディペンダビリティ記述と障害監視制御部の動作とが無矛盾に維持されるように構成されていてもよい。

30

【0416】

さらに、上記コンピュータシステムは、ディペンダビリティ記述はその一部分がパターン化されており、ディペンダビリティ記述のパターンと障害監視制御用データの対応を表わすテーブル (D-Case パターン <=> モジュール対応テーブル) を使用して障害監視制御用データをディペンダビリティ記述から生成するように構成されていてもよい。

【0417】

さらに、上記コンピュータシステムは、障害監視制御部はモタ管理部 (モタ管理部 40) とアクション管理部 (アクション管理部 60) を持ち、モタ管理部は 1 以上のモタモジュール (モタモジュール群 50) を管理し、アクション管理部は 1 以上のアクションモジュール (アクションモジュール群 70) を管理し、モタ管理部によるモタモジュールの選択および動作の制御、並びに、アクション管理部によるアクションモジュールの選択および制御と、ディペンダビリティ記述のパターンとの対応を示すテーブルを使用して障害監視制御用データをディペンダビリティ記述から生成するように構成されていてもよい。

40

【0418】

さらに、上記コンピュータシステムの制御方法は、ステークホルダ間のディペンダビリティに関する合意を記述したディペンダビリティ記述データを格納するディペンダビリテ

50

ィ記述格納ステップ(D-Case格納ステップS1)と、障害監視制御部が対象システム内の状態を監視し必要な場合に対策を実行する障害監視制御ステップ(障害監視実行ステップS3)と、障害監視制御ステップの動作を制御する障害監視制御用データをディペンダビリティ記述から生成するディペンダビリティ記述変換ステップ(D-Case読み出しステップS2、D-Case変換ステップS3)とを持ち、障害監視制御用データが常にディペンダビリティ記述格納ステップによって格納されたディペンダビリティ記述データからディペンダビリティ記述変換ステップによって生成されることによりディペンダビリティ記述と障害監視制御部の動作とが無矛盾に維持されるように構成されていてもよい。

【0419】

また、障害対応システム1の各ブロック、特にディペンダビリティ維持装置20のD-Case変換部21は、ハードウェアロジックによって構成してもよいし、次のようにCPUを用いてソフトウェアによって実現してもよい。

【0420】

後者の場合、障害対応システム1は、各機能を実現するプログラムの命令を実行するCPU(central processing unit)、上記プログラムを格納したROM(read only memory)、上記プログラムを展開するRAM(random access memory)、上記プログラムおよび各種データを格納するメモリ等の記憶装置(記録媒体)などを備えている。そして、本発明の目的は、上述した機能を実現するソフトウェアである障害対応システム1あるいはディペンダビリティ維持装置20の制御プログラムのプログラムコード(実行形式プログラム、中間コードプログラム、ソースプログラム)をコンピュータで読み取り可能に記録した記録媒体を、上記の障害対応システム1あるいはディペンダビリティ維持装置20に供給し、そのコンピュータ(またはCPUやMPU)が記録媒体に記録されているプログラムコードを読み出し実行することによっても、達成可能である。

【0421】

上記記録媒体としては、例えば、磁気テープやカセットテープ等のテープ系、フロッピー(登録商標)ディスク/ハードディスク等の磁気ディスクやCD-ROM/MO/MD/DVD/CD-R等の光ディスクを含むディスク系、ICカード(メモリカードを含む)/光カード等のカード系、あるいはマスクROM/EPROM/EEPROM/フラッシュROM等の半導体メモリ系などを用いることができる。

【0422】

また、上記の障害対応システム1あるいはディペンダビリティ維持装置20を通信ネットワークと接続可能に構成し、上記プログラムコードを通信ネットワークを介して供給してもよい。この通信ネットワークとしては、特に限定されず、例えば、インターネット、イントラネット、エキストラネット、LAN、ISDN、VAN、CATV通信網、仮想専用網(virtual private network)、電話回線網、移動体通信網、衛星通信網等が利用可能である。また、通信ネットワークを構成する伝送媒体としては、特に限定されず、例えば、IEEE1394、USB、電力線搬送、ケーブルTV回線、電話線、ADSL回線等の有線でも、IrDAやリモコンのような赤外線、Bluetooth(登録商標)、802.11無線、HDDR、携帯電話網、衛星回線、地上波デジタル網等の無線でも利用可能である。なお、本発明は、上記プログラムコードが電子的な伝送で具現化された、搬送波に埋め込まれたコンピュータデータ信号の形態でも実現され得る。

【0423】

上記説明では、機能ブロックおよび手順を図示し参照したが、機能の分離併合または手順の移動は上記機能を満たす限り可能であり、上記説明が本発明を限定するものではない。

【0424】

(1)本発明に係るディペンダビリティ維持装置は、監視制御対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置であって、上記監視制御対象システムのディペンダビリティに関する(ステークホルダ群によって合意さ

10

20

30

40

50

れた)要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出し、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成するディペンダビリティ記述変換部を備えたことを特徴としている。

【0425】

(6)また、本発明に係るディペンダビリティ維持装置の制御方法は、監視制御対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給するディペンダビリティ維持装置の制御方法であって、上記監視制御対象システムのディペンダビリティに関する(ステークホルダ群によって合意された)要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出す読み出しステップと、読み出したディペンダビリティ記述データから上記障害監視制御用データを生成する変換ステップとを含むことを特徴としている。

10

【0426】

上記の構成によれば、ディペンダビリティ維持装置がディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから障害監視制御用データを生成する。そして、これに従って障害監視制御部が動作することにより、障害監視制御部が監視制御対象システムの状態を監視し、必要な場合に対策を実行することができる。

【0427】

ここで、ペンダビリティ記述データは、監視制御対象システムのディペンダビリティに関わる仕様を記述したものである。そして、監視制御対象システムの各ステークホルダが監視制御対象システムのディペンダビリティに関して合意した際、その結果がペンダビリティ記述データとして記述され、ディペンダビリティ記述格納部に格納されたものであることが望ましい。

20

【0428】

このように、障害監視制御部が用いる障害監視制御用データが、ディペンダビリティ記述格納部に格納されたディペンダビリティ記述データから、ディペンダビリティ維持装置によって生成されることにより、ディペンダビリティ記述データと障害監視制御部の動作とが常に無矛盾になるように維持することが可能となる。

【0429】

(2)さらに、本発明に係るディペンダビリティ維持装置は、上記(1)に記載の構成において、上記障害監視制御部は、上記監視制御対象システムの状態を監視し、必要な場合に対策を実行するために使用するモジュールを、複数のモジュールから選択して制御可能であり、上記ディペンダビリティ記述データは、1つの記述が、上記モジュールを特定するためのモジュール特定情報を値として設定可能な変数を少なくとも含む形式であり、上記ディペンダビリティ記述変換部は、あらかじめ設定された、上記モジュール特定情報と上記障害監視制御用データとの対応関係を示す情報に基づいて、上記ディペンダビリティ記述データに含まれる変換対象の記述を、該記述に含まれるモジュール特定情報に対応する障害監視制御用データに変換することを特徴としている。

30

【0430】

上記の構成によれば、さらに、ディペンダビリティ記述データに含まれる変換対象の記述を障害監視制御用データに変換する際、該記述に含まれるモジュール特定情報によって特定されるモジュールを障害監視制御部に選択させ、制御させる障害監視制御用データを生成する。

40

【0431】

これにより、障害監視制御部が制御可能なモジュールが複数ある場合であっても、ディペンダビリティ記述データの記述から適切なモジュールを制御対象とした障害監視制御用データを生成することが可能となる。

【0432】

(3)さらに、本発明に係るディペンダビリティ維持装置は、上記(2)に記載の構成において、上記ディペンダビリティ記述データの1つの記述から変数部分を除いた固定部

50

分に対応して、上記対応関係が設定されており、上記ディペンダビリティ記述変換部は、上記ディペンダビリティ記述データの記述に含まれる上記固定部分に対応する上記対応関係を参照して、該記述に対応する障害監視制御用データに変換することを特徴としている。

【0433】

上記の構成によれば、さらに、ディペンダビリティ記述データに含まれる変換対象の記述を障害監視制御用データに変換する際、該記述から含まれる変数部分を除いた固定部分に対応した対応関係を参照して、該記述に対応する障害監視制御用データに変換する。

【0434】

これにより、ディペンダビリティ記述データに含まれる記述の固定部分毎に、異なる対応関係に対応付けておくことが可能となる。よって、モジュール特定情報等の変数が同じであっても、固定部分が異なれば、異なる対応関係を用いて、ディペンダビリティ記述データに含まれる記述を障害監視制御用データに変換することができる。それゆえ、多様な変換を簡易な規則によって実現することができる。

10

【0435】

(4)また、本発明に係る障害対応システムは、上記(1)から(3)のいずれかに記載のディペンダビリティ維持装置と、上記ディペンダビリティ記述格納部と、上記障害監視制御部とを含み、上記ディペンダビリティ維持装置が上記ディペンダビリティ記述格納部から読み出したディペンダビリティ記述データから生成した障害監視制御用データに従って、上記障害監視制御部が動作することにより、上記監視制御対象システムの状態を監視し、必要な場合に対策を実行するように構成してもよい。

20

【0436】

(5)さらに、本発明に係る障害対応システムは、上記(4)に記載の構成において、上記監視制御対象システムの状態を監視する1以上のモジュールと、必要な場合上記監視制御対象システムに対して対策を実行する1以上のアクションモジュールと、上記障害監視制御部の制御により、上記モジュールの選択および動作の制御を行うモニタ管理部と、上記障害監視制御部の制御により、上記アクションモジュールの選択および動作の制御を行うアクション管理部と、を含むように構成してもよい。

【0437】

なお、上記のディペンダビリティ維持装置および障害対応システムは、コンピュータによって実現してもよく、この場合には、コンピュータを上記ディペンダビリティ記述変換部として動作させることにより上記ディペンダビリティ維持装置をコンピュータにて実現させるディペンダビリティ維持装置のプログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

30

【0438】

上記説明では、機能ブロックおよび手順を図示し参照したが、機能の分離併合または手順の移動は上記機能を満たす限り可能であり、上記説明が本発明を限定するものではない。

【0439】

発明の詳細な説明の項においてなされた具体的な実施態様または実施例は、あくまでも、本発明の技術内容を明らかにするものであって、そのような具体例にのみ限定して狭義に解釈されるべきものではなく、本発明の精神と次に記載する特許請求事項との範囲内で、いろいろと変更して実施することができるものである。

40

【産業上の利用可能性】

【0440】

本発明のディペンダビリティ維持装置によれば、不完全性と不確実性が潜在的に存在するオープン環境において、ステークホルダ間における要求の誤解、環境の変化に対する対応不能、障害対応の失敗という3つの問題に対するソリューションを提供することができる。オープン環境でのシステム開発及びシステム運用に広く利用することができる。

【0441】

50

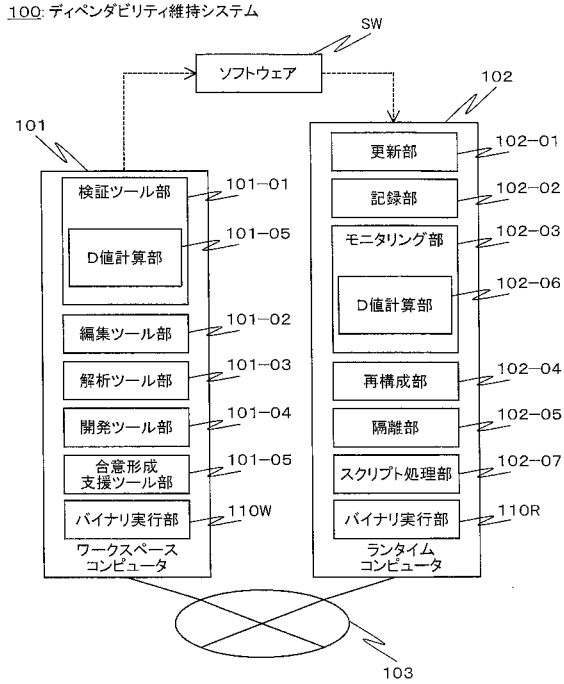
また、本発明の障害対応システムおよびディペンダビリティ維持装置によれば、各ステークホルダの合意の記述を容易に行うとともに、監視制御対象システムの一部に更新があった場合に、合意の記述とモジュールの実装とが無矛盾に開発・更新されるように維持することができる。よって、例えば、組込みシステムから、インターネットなどでつながれた多数システムまで、多様なシステムのディペンダビリティを維持するため装置・方法として好適である。

【符号の説明】

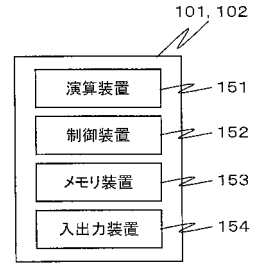
【0442】

100 (100U, 100S, 100B, 100P)	ディペンダビリティ維持システム	10
101	ワークスペースコンピュータ (ディペンダビリティ維持装置、ワークスペース装置)	
101-05	D値計算部 (ディペンダビリティ値決定手段)	
102	ランタイムコンピュータ (ディペンダビリティ維持装置、ランタイム装置)	
102-04	再構成部 (再構成手段)	
102-06	D値計算部 (ディペンダビリティ値決定手段)	
102-07	スクリプト処理部 (スクリプト処理手段)	
901-01	ディペンダビリティ記述データ入力部 (記述データ取得手段)	
902-01	ソフトウェア入力部 (記述データ取得手段)	
902-02	変更送出部 (変更要求送信手段)	20
P300	変化対応サイクル	
P301	障害対応サイクル	
1	障害対応システム	
2	監視制御対象システム	
10	D-Case格納部 (ディペンダビリティ記述格納部)	
20	ディペンダビリティ維持装置	
21	D-Case変換部 (ディペンダビリティ記述変換部)	
22	D-Caseパターン<=>モジュール対応テーブル	
30	障害監視制御部	
40	モニタ管理部	30
50	モニタモジュール群 (モジュール)	
60	アクション管理部	
70	アクションモジュール群 (モジュール)	
S2	読み出しステップ	
S3	変換ステップ	

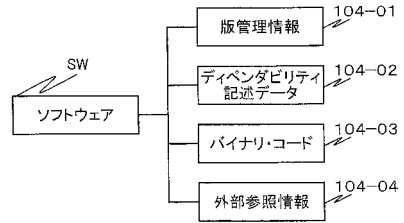
【 図 1 】



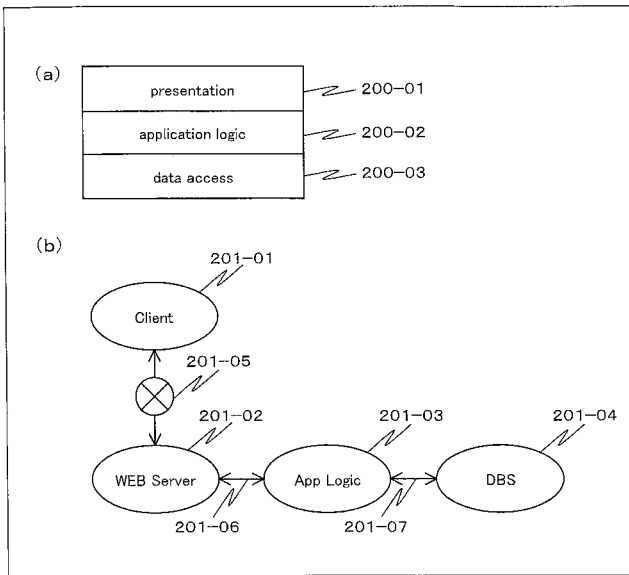
【 図 2 】



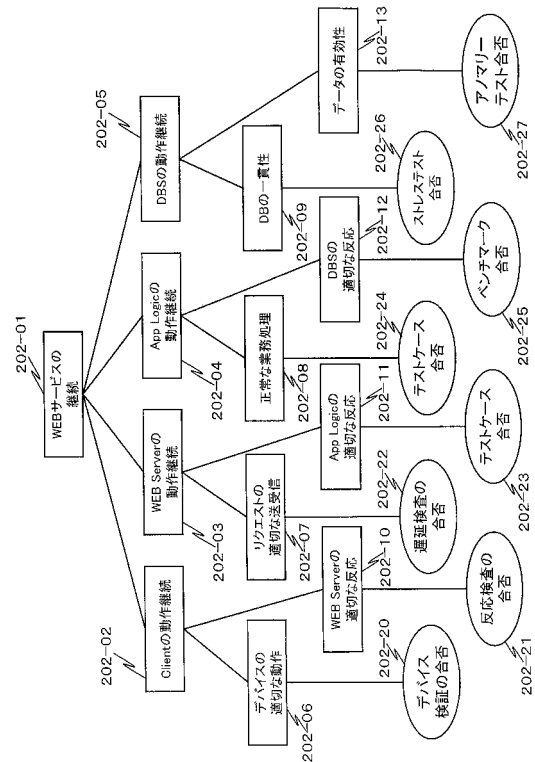
【 図 3 】



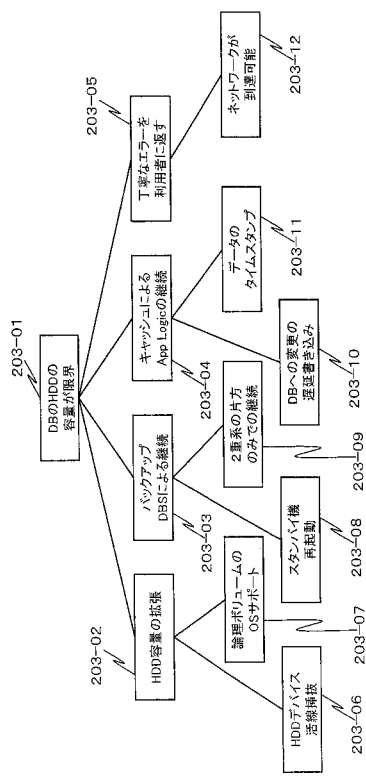
【 図 4 】



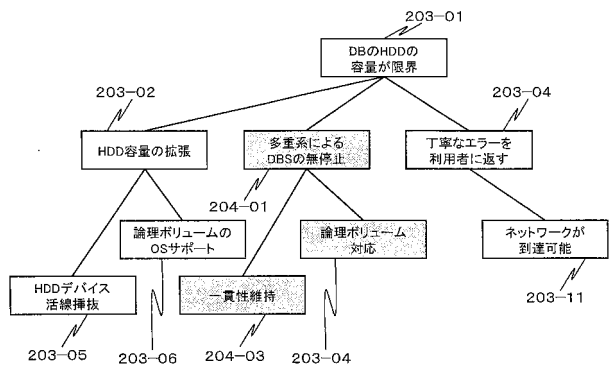
【 図 5 】



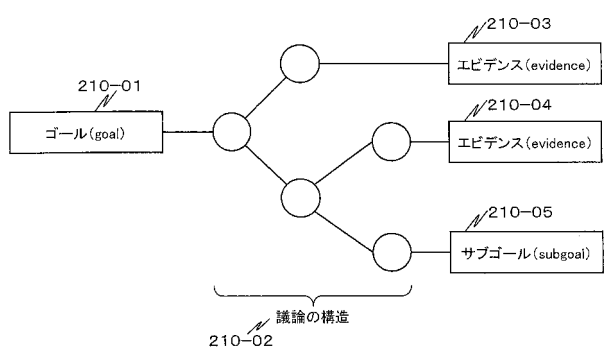
【 図 6 】



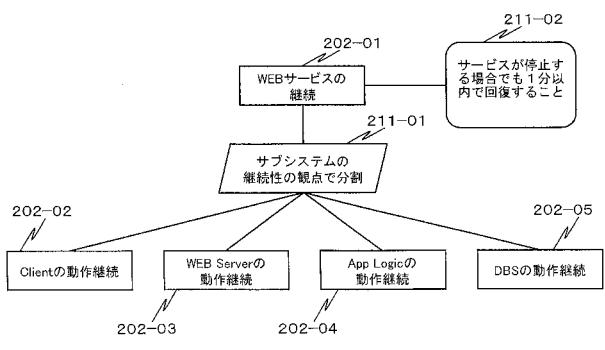
【 図 7 】



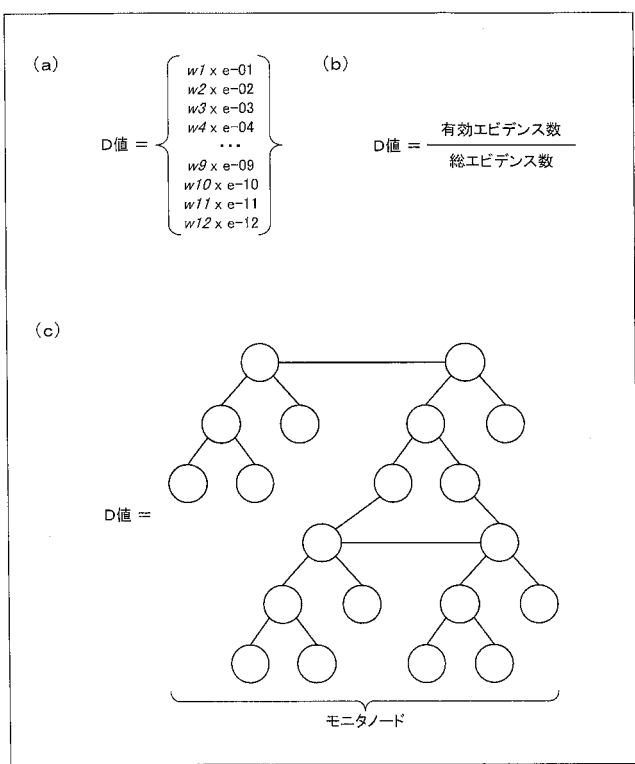
【 図 8 】



【 図 9 】



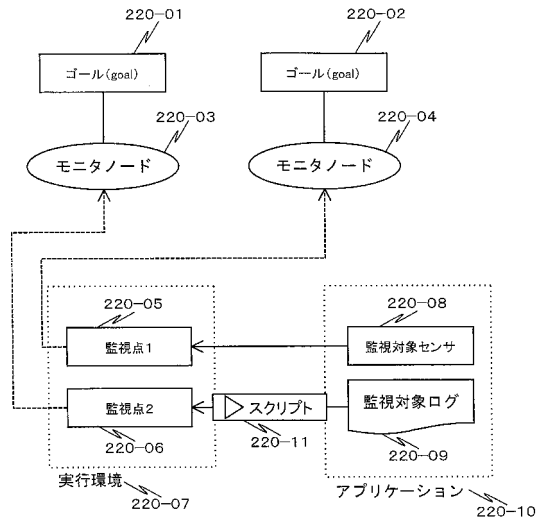
【 図 10 】



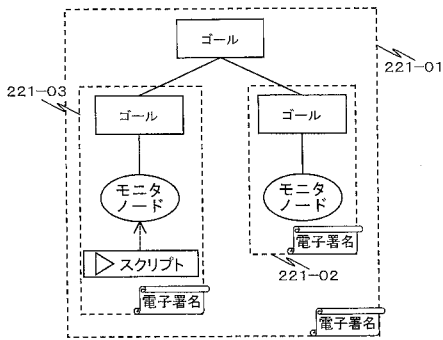
【 図 1 1 】

異常の種類	目的	フェーズ	ターゲット
自然災害	可用性	仕様	CPU
人的エラー	信頼性	設計	メモリ
悪意あるアタック	安全性	統合	ファイルシステム
ハードウェア障害	完全性	検証	ネットワーク
	保守性	導入	I/O
		保守	消費電力

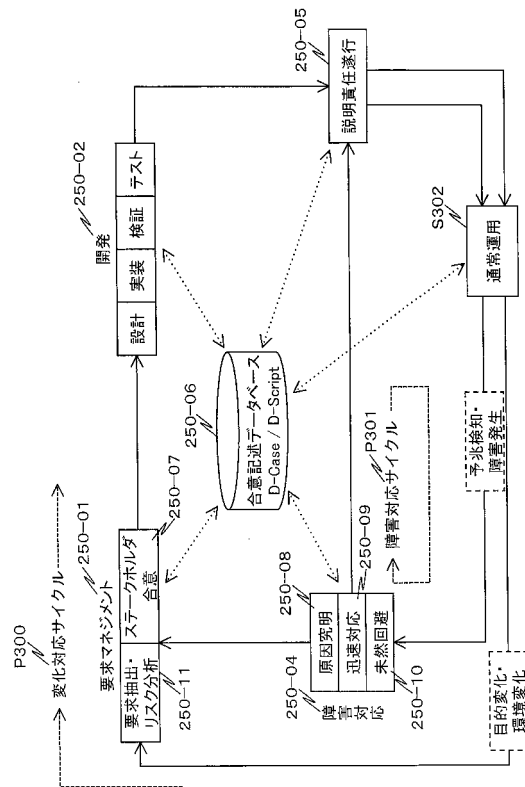
【 図 1 2 】



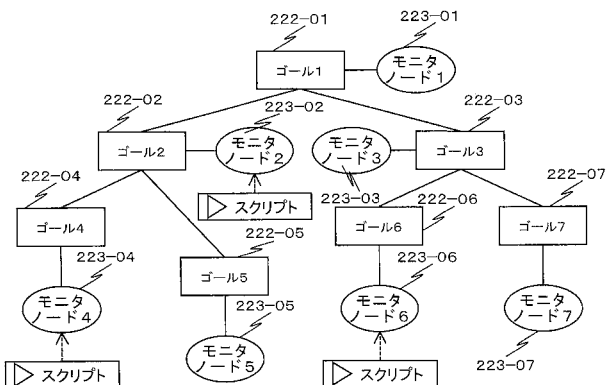
【 図 1 3 】



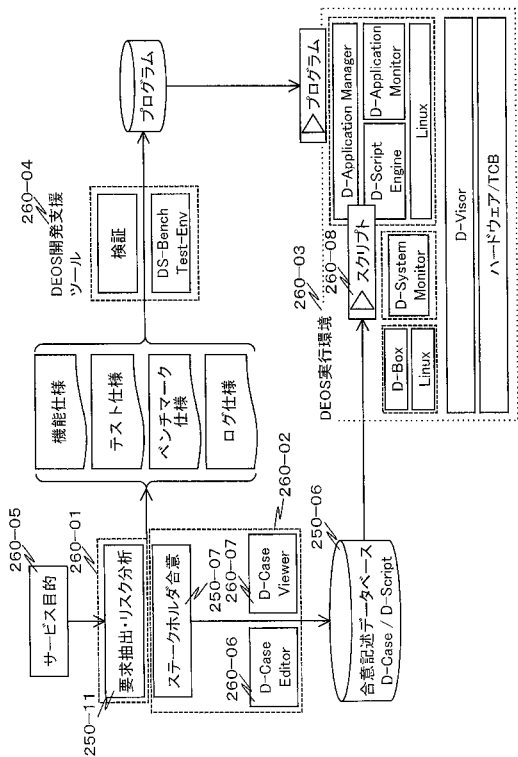
【 図 1 5 】



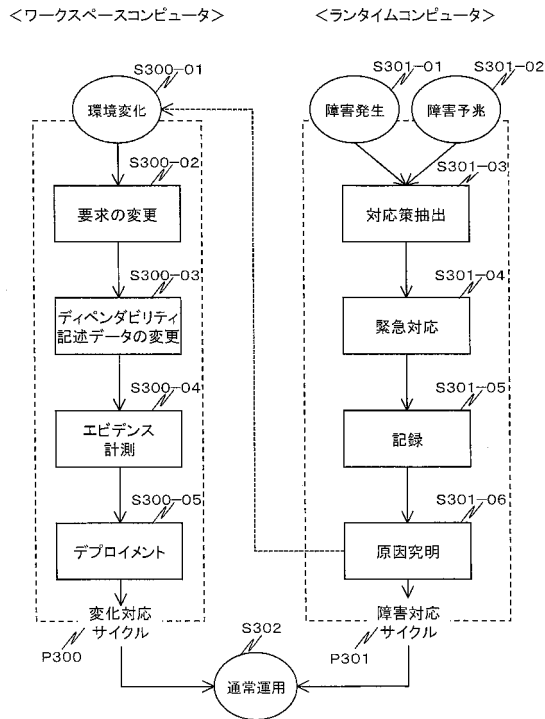
【 図 1 4 】



【図16】

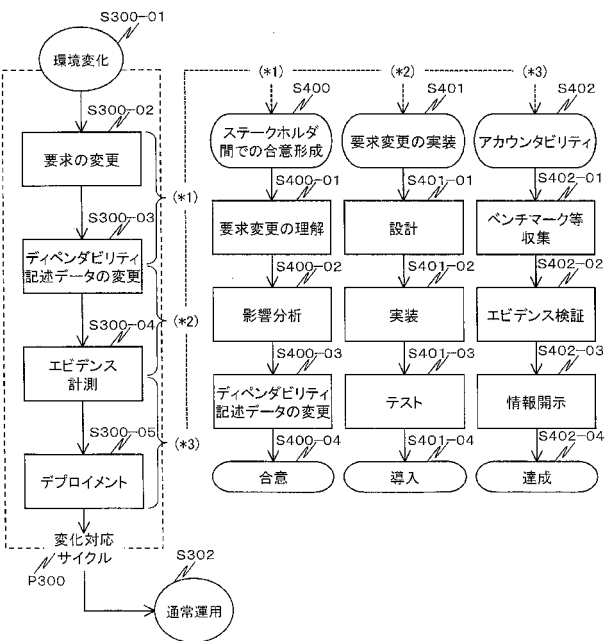


【図17】



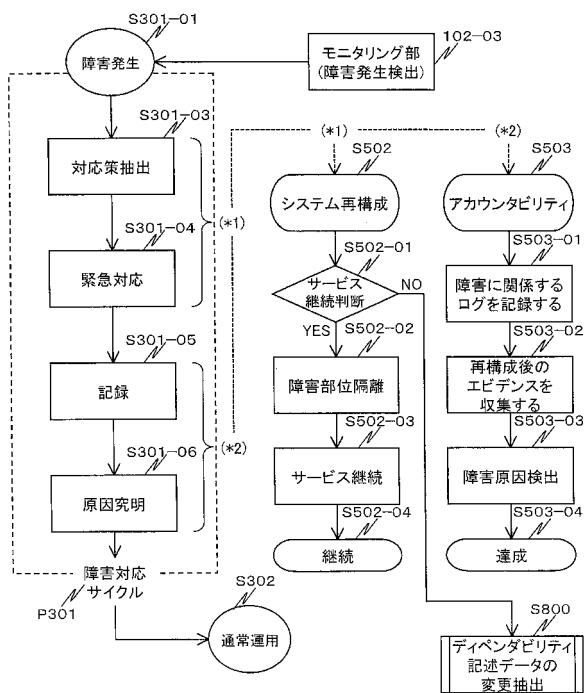
【図18】

<ワークスペースコンピュータ>

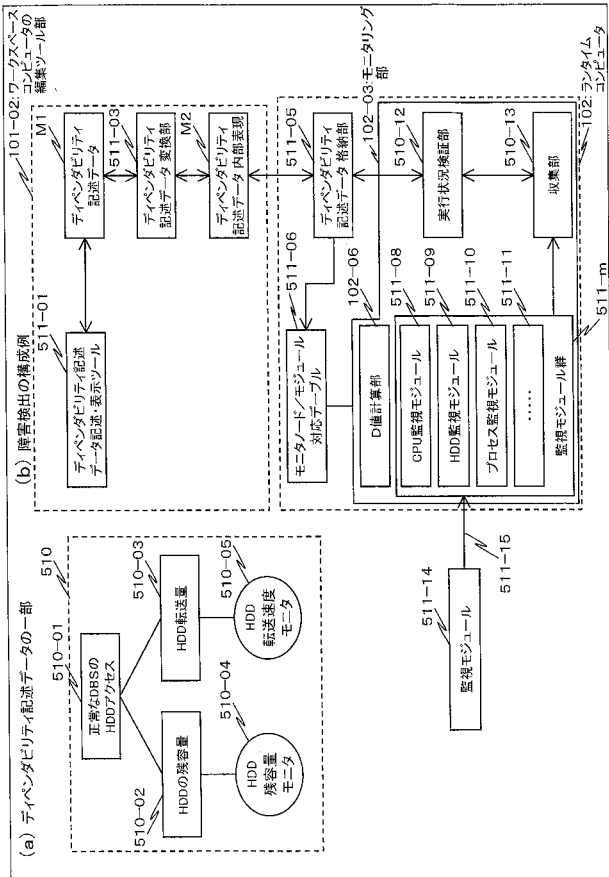


【図19】

<ランタイムコンピュータ>



【図 20】

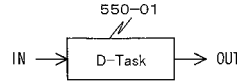


【図 21】

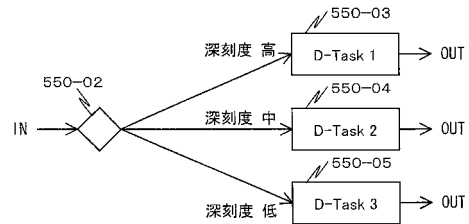
モニタノード	モジュール	
\$2	モジュール名称	引数
ディスク容量	HDD監視	\$1, \$3
ディスク転送量	HDD監視	\$1, \$3

【図 22】

基本動作 (D-Task)

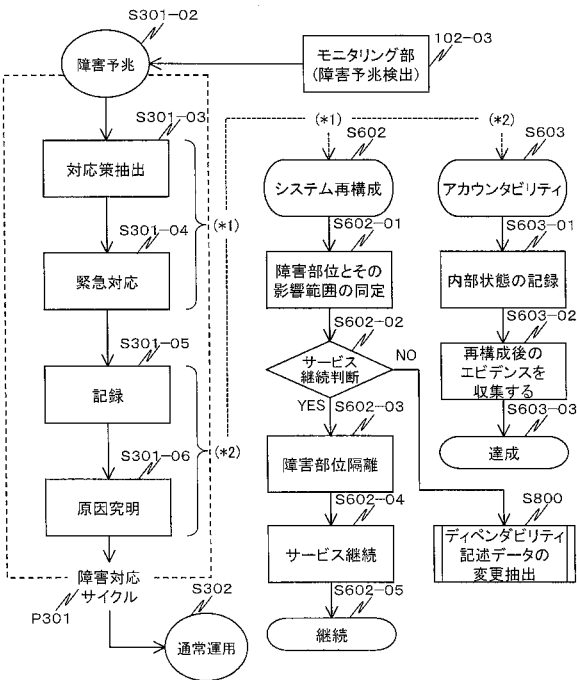


制御動作 (D-Control)

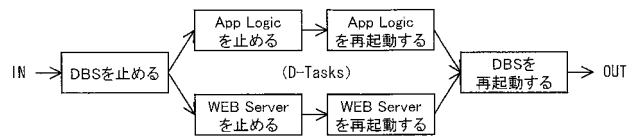


【図 23】

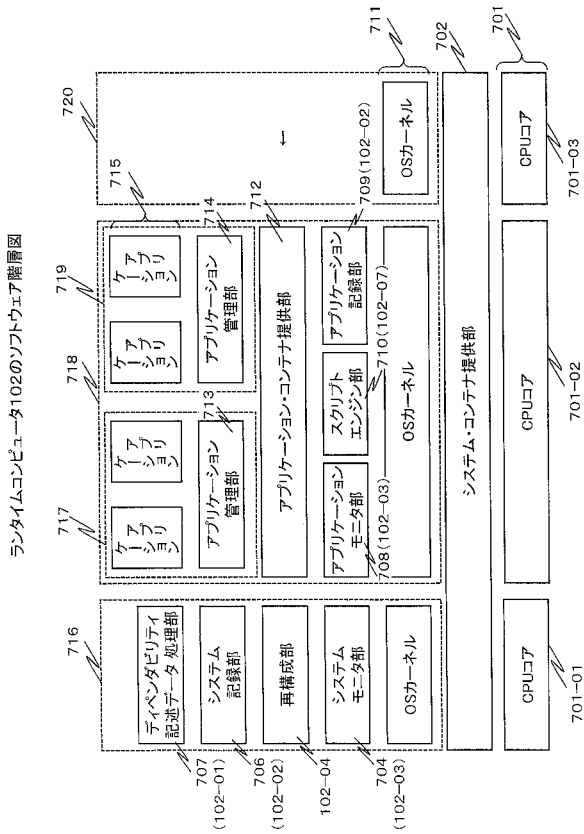
<ランタイムコンピュータ>



【図 24】



【図 25】

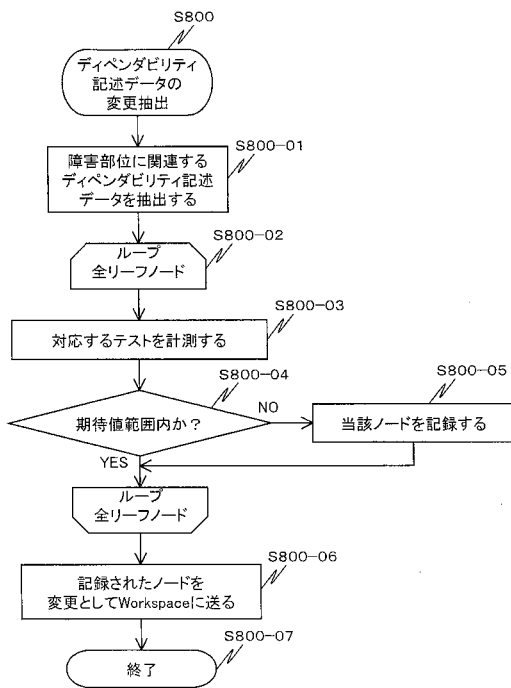


【図 26】

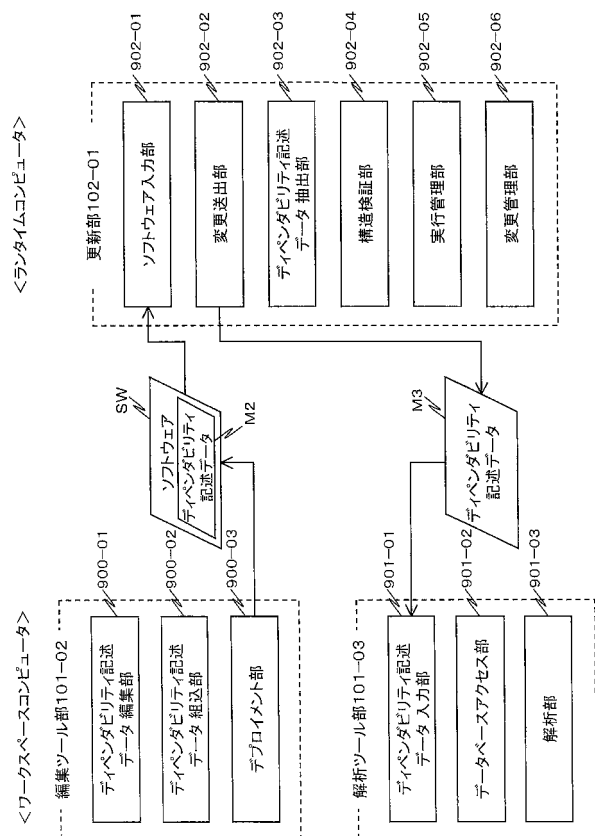
隔離項目に関する機能要件

隔離項目	機能要件
アドレス空間	独立したアドレッシングが可能であること。
名前空間	ファイル名、プロセスID、等のIDが独立していること。
実メモリ	独立した実メモリアクセスが可能であること。
キャッシュメモリ	キャッシュメモリの影響が独立していること。
CPUスケジューリング	CPUを独立して利用できること。
CPUアサインメント	マルチコア構成の場合にCPU割り当てが独立していること。
I/Oバンド幅	I/Oのバンド幅が独立していること。
バス・バンド幅	バスのバンド幅が独立していること。
割り込み	割り込み処理が独立していること。
Time-of-day	日付と時刻が独立していること。
特権	独立した特権が与えられていること。

【図 27】

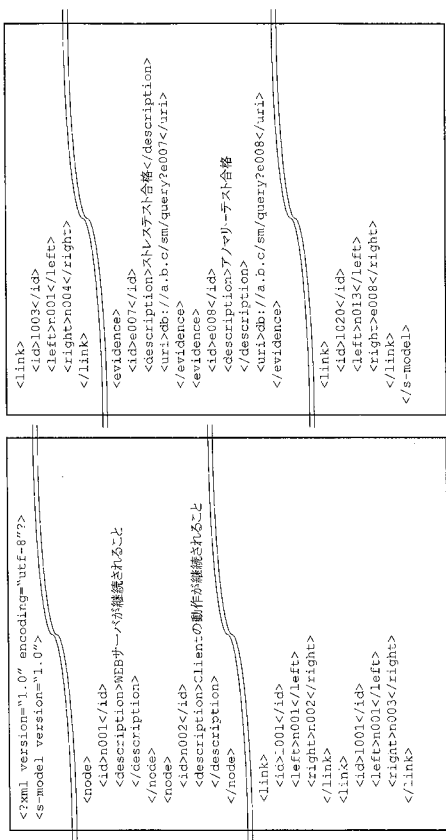


【図 28】

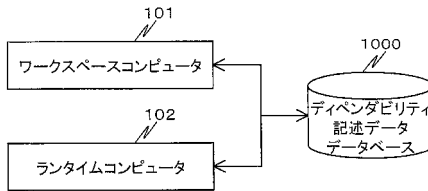


【図 29】

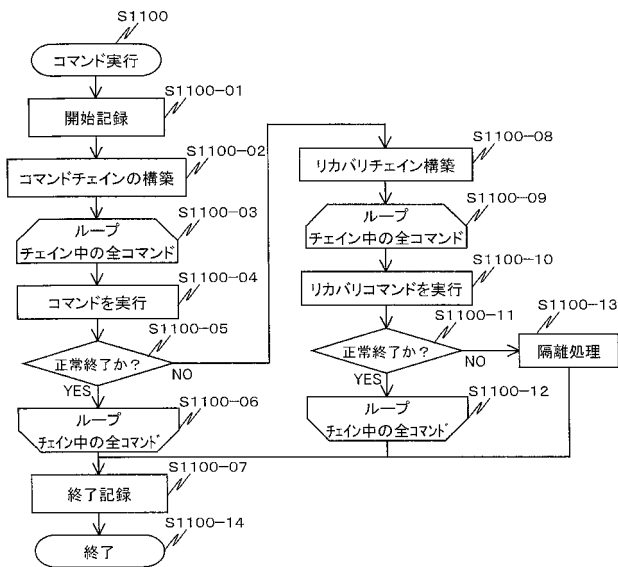
ディメンタビリティ記述データの計算機表現の一部



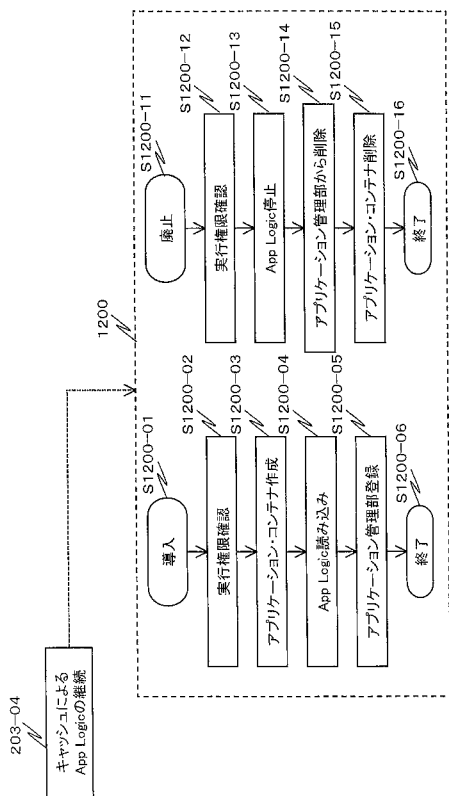
【図 30】



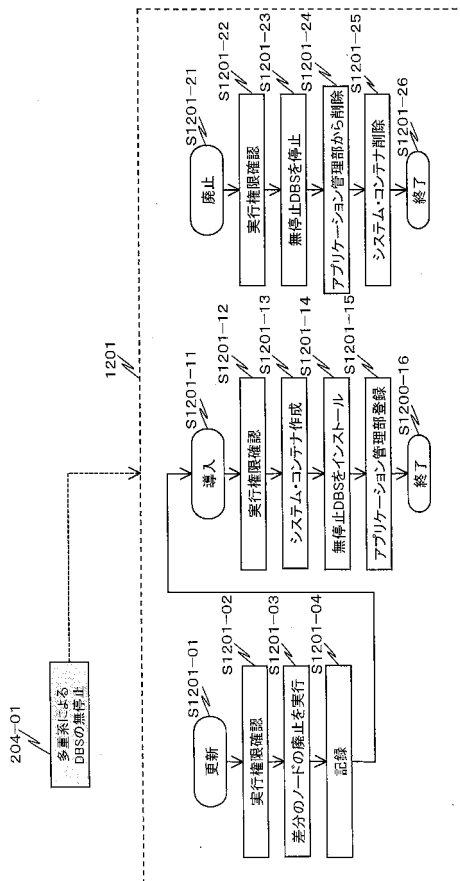
【図 31】



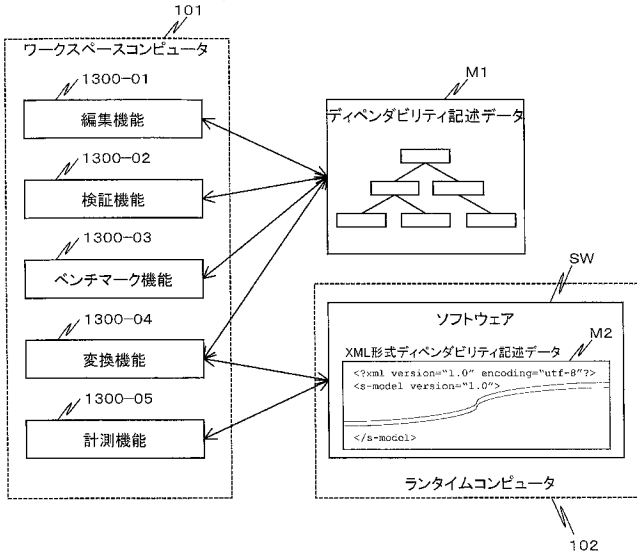
【図 32】



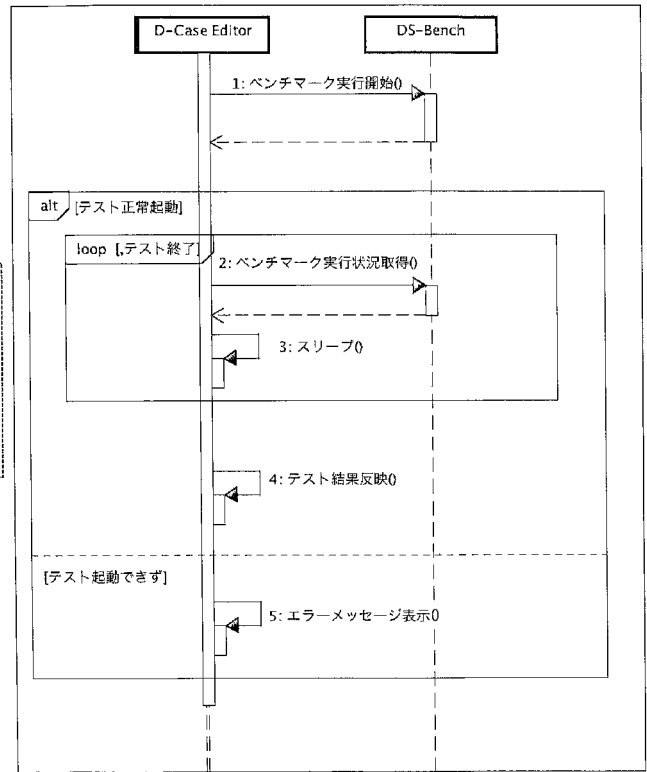
【図 33】



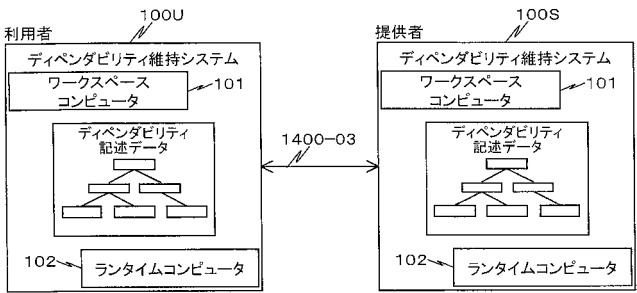
【 図 3 4 】



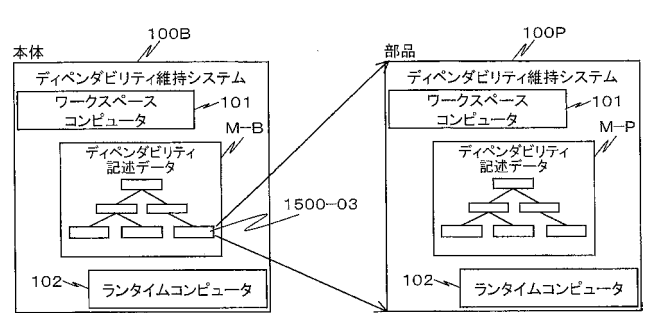
【 図 3 5 】



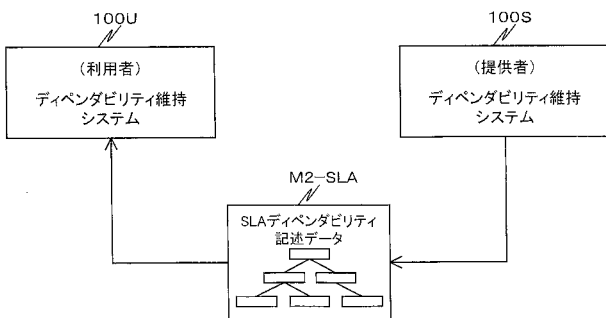
【 図 3 6 】



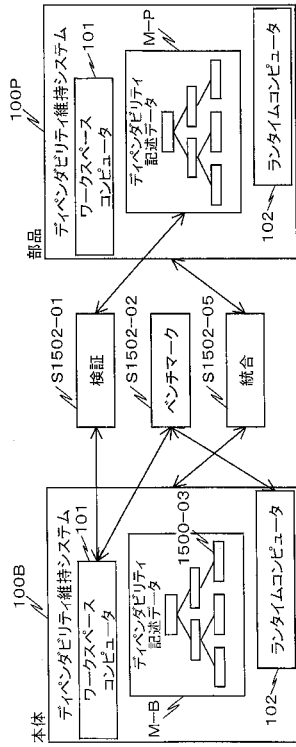
【 図 3 8 】



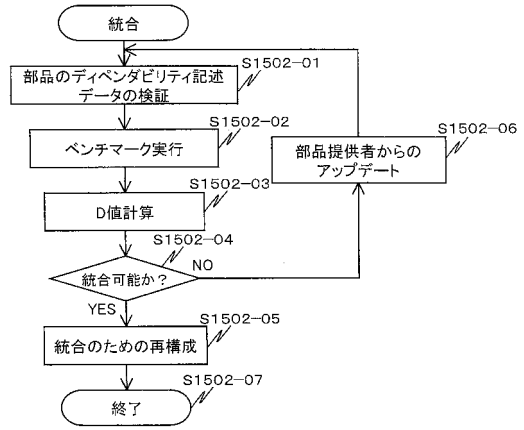
【 図 3 7 】



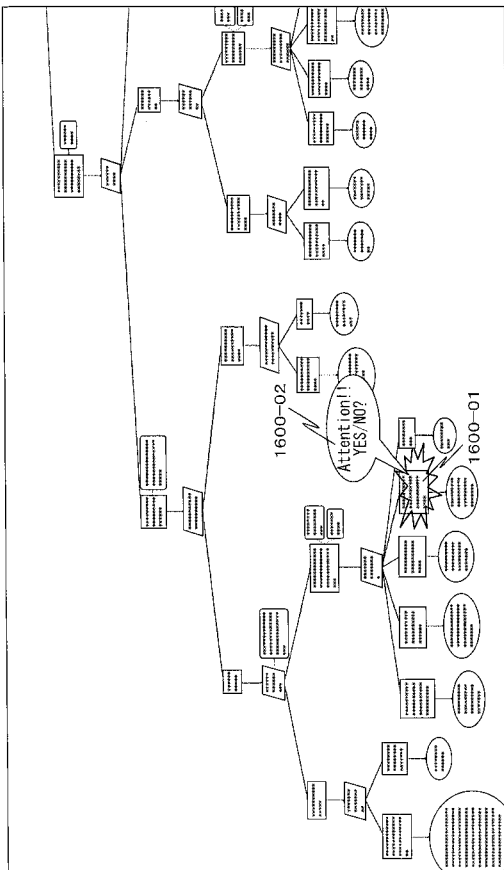
【 図 3 9 】



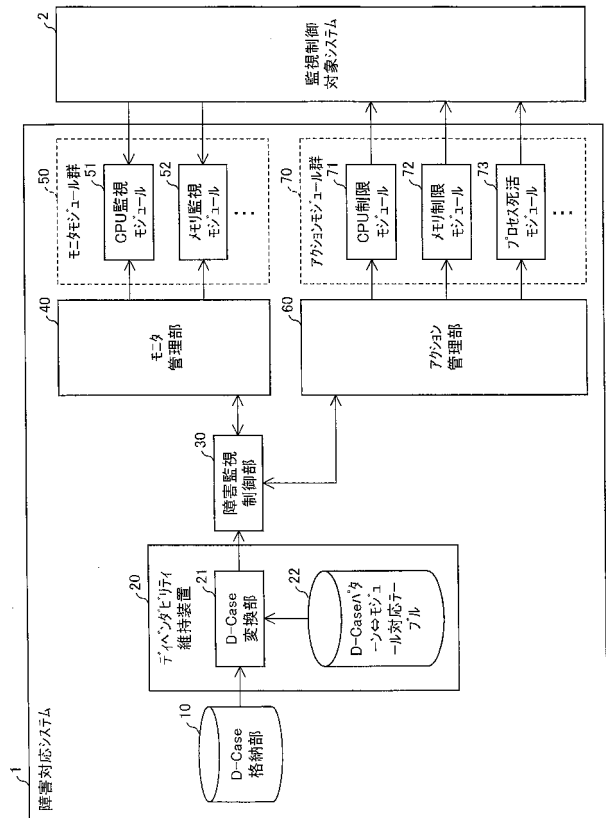
【 図 4 0 】



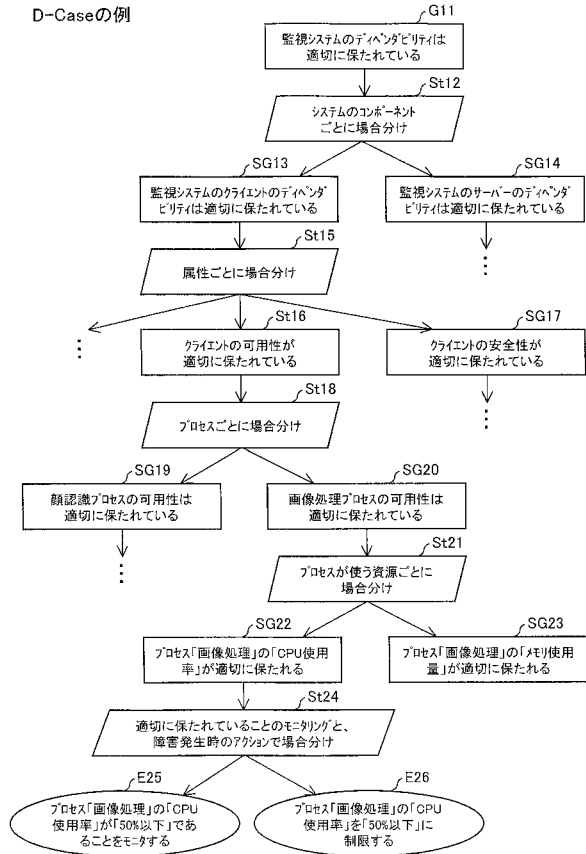
【 図 4 1 】



【 図 4 2 】



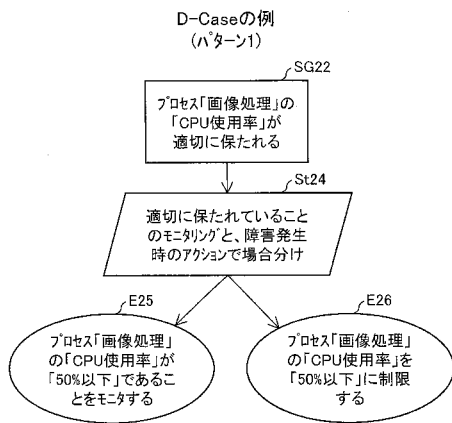
【 図 4 3 】



【 図 4 4 】



【 図 4 5 】



【 図 4 6 】

(a)

パターン1(モニタモジュール):プロセス「\$1」の「\$2」が「\$3」であることをモニタする

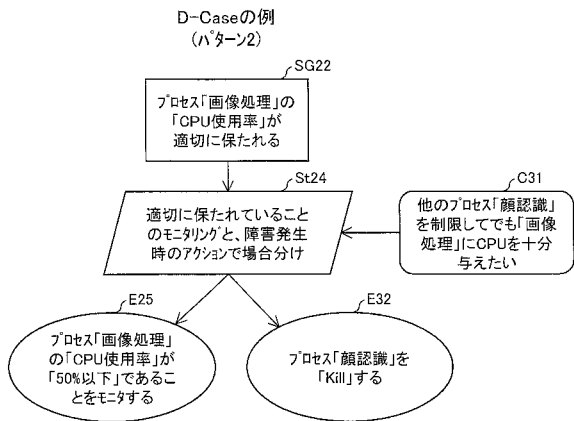
パターン	モジュール	
\$2	モジュール名	引数
CPU使用率	CPU監視	\$1, \$3
メモリ使用量	メモリ監視	\$1, \$3
ディスク転送量	ディスク監視	\$1, \$3

(b)

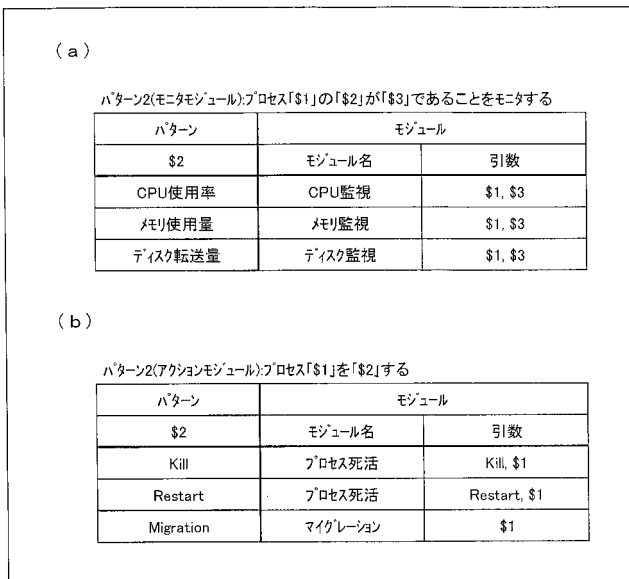
パターン1(アクションモジュール):プロセス「\$1」の「\$2」を「\$3」に制限する

パターン	モジュール	
\$2	モジュール名	引数
CPU使用率	CPU制限	\$1, \$3
メモリ使用量	メモリ制限	\$1, \$3
ディスク転送量	ディスク制限	\$1, \$3

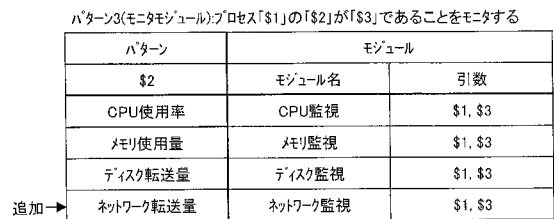
【 図 4 7 】



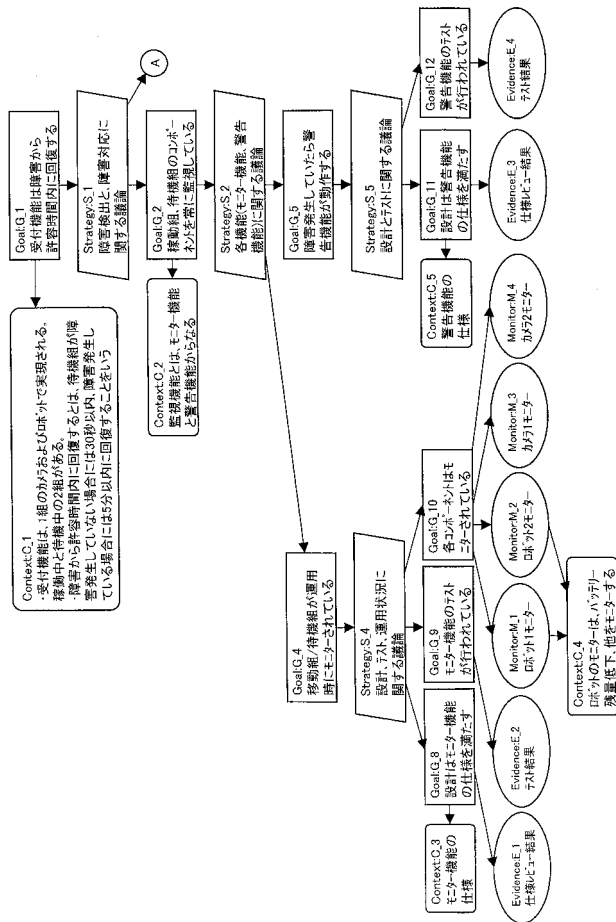
【 図 4 8 】



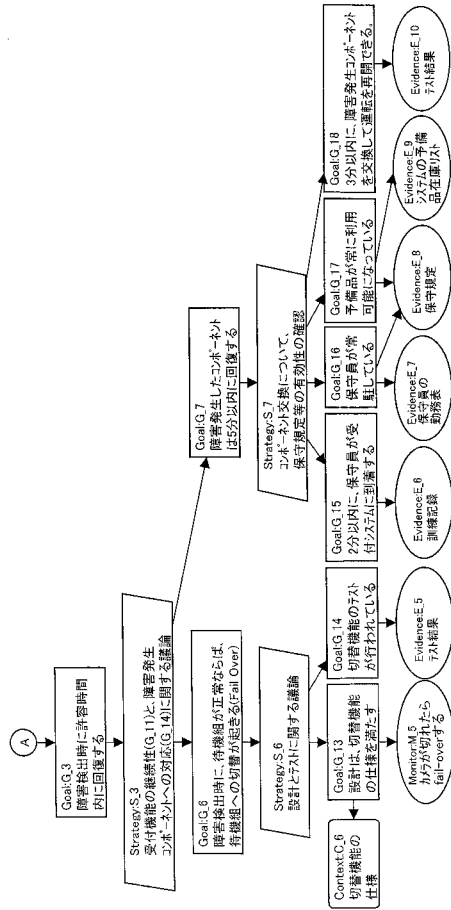
【 図 4 9 】



【 図 5 0 】



【 図 5 1 】



【 図 5 2 】

```
<?xml version="1.0" encoding="UTF-8" ?>
<dcase:dcase xmlns:dcase="http://www.sample_dcase.co.jp/2010/06/dcase"
id="QdCjCPOIEd-p_d8e3tzhBw">
<dcase:description />
<dcase:nodes>
<dcase:node type="Context" id="id1" name="C_1">
<dcase:description>受付機能は、1組のカメラおよびロボットで実現される。稼働中と待機中の2組がある。障害から許容時間内に回復するとは、待機組が障害発生していない場合には30秒以内、障害発生している場合には5分以内に回復することをいう。
</dcase:description>
</dcase:node>
<dcase:node type="Goal" id="id2" name="G_1">
<dcase:description>受付機能は障害から許容時間内に回復する</dcase:description>
</dcase:node>
<dcase:node type="Strategy" id="id3" name="S_1">
<dcase:description>障害検出と、障害対応に関する議論</dcase:description>
</dcase:node>
<dcase:node type="Goal" id="id4" name="G_4">
<dcase:description>移動組/待機組が運用時にモニターされている</dcase:description>
</dcase:node>
<dcase:node type="Strategy" id="id5" name="S_4">
<dcase:description>設計、テスト、運用状況に関する議論</dcase:description>
</dcase:node>
<dcase:node type="Context" id="id6" name="C_3">
<dcase:description>モニター機能の仕様</dcase:description>
</dcase:node>
<dcase:node type="Goal" id="id7" name="G_8">
<dcase:description>設計はモニター機能の仕様を満たす</dcase:description>
</dcase:node>
<dcase:node type="Evidence" id="id8" name="E_1">
<dcase:description>仕様レビュー結果</dcase:description>
</dcase:node>
</dcase:nodes>

```

【 図 5 3 】

```
<dcase:node type="Goal" id="id9" name="G_9">
<dcase:description>モニター機能のテストが行われている</dcase:description>
</dcase:node>
<dcase:node type="Evidence" id="id10" name="E_2">
<dcase:description>テスト結果</dcase:description>
</dcase:node>
<dcase:node type="Goal" id="id11" name="G_10">
<dcase:description>各コンポーネントはモニターされている</dcase:description>
</dcase:node>
<dcase:node type="Monitor" id="id12" name="M_1">
<dcase:description>ロボット1モニター</dcase:description>
<dcase:property name="Userdef002" value="main" />
<dcase:property name="Userdef005" value="ロボット[NUMBER]モニター" />
<dcase:property name="Userdef006" value="if exec
/home/et2010/damd/RobotCheck.sh main id="[id]" [n]then exec
/home/et2010/damd/RobotError.sh main" />
<dcase:property name="Userdef007" value="NUMBER=1" />
</dcase:node>
<dcase:node type="Monitor" id="id13" name="M_2">
<dcase:description>ロボット2モニター</dcase:description>
<dcase:property name="Userdef002" value="backup" />
<dcase:property name="Userdef005" value="ロボット[NUMBER]モニター" />
<dcase:property name="Userdef006" value="if exec
/home/et2010/damd/RobotCheck.sh backup id="[id]" [n]then exec
/home/et2010/damd/RobotError.sh backup" />
<dcase:property name="Userdef007" value="NUMBER=2" />
</dcase:node>
<dcase:node type="Monitor" id="id14" name="M_3">
<dcase:description>カメラ1モニター</dcase:description>
<dcase:property name="Userdef002" value="Camera" />
<dcase:property name="Userdef005" value="カメラ[NUMBER]モニター" />
<dcase:property name="Userdef006" value="if exec
/home/et2010/damd/CameraCheck.sh [NUMBER] id="[id]" [n]then exec
/home/et2010/damd/CameraStatus.sh [NUMBER] camera-status" />
<dcase:property name="Userdef007" value="NUMBER=1" />
<dcase:property name="Userdef008" value="Score=0,Weight=1" />
</dcase:node>

```

【 図 5 4 】

```
<dcase:node type="Monitor" id="id22" name="M_4">
<dcase:description>カメラ2モニター</dcase:description>
<dcase:property name="Userdef002" value="Camera" />
<dcase:property name="Userdef004" value="" />
<dcase:property name="Userdef005" value="カメラ[NUMBER]モニター" />
<dcase:property name="Userdef006" value="if exec
/home/et2010/damd/CameraCheck.sh [NUMBER] id="[id]" [n]then exec
/home/et2010/damd/CameraStatus.sh [NUMBER] camera-status" />
<dcase:property name="Userdef007" value="NUMBER=2" />
<dcase:property name="Userdef008" value="Score=0,Weight=1" />
</dcase:node>
<dcase:node type="Goal" id="id15" name="G_13">
<dcase:description>設計は、切替機能の仕様を満たす</dcase:description>
</dcase:node>
<dcase:node type="Monitor" id="id26" name="M_5">
<dcase:description>カメラが切れたらfail-overする</dcase:description>
<dcase:property name="Userdef002" value="OK" />
<dcase:property name="Userdef005" value="カメラが切れたら[ACTION]する" />
<dcase:property name="Userdef006" value="if exec /home/et2010/damd/videocheck.sh
id="[id]" [n]then exec /home/et2010/damd/demo1.sh [ACTION] camera-error" />
<dcase:property name="Userdef007" value="ACTION=fail-over" />
<dcase:property name="Userdef008" value="Score=0,Weight=1" />
</dcase:node>
<dcase:node type="Context" id="id16" name="C_6">
<dcase:description>切替機能の仕様</dcase:description>
</dcase:node>
<dcase:links>
<dcase:link id="id17" name="LINK_72" source="id15" target="id26">
<dcase:description />
</dcase:link>
<dcase:link id="id18" name="LINK_71" source="id11" target="id14">
<dcase:description />
</dcase:link>

```

【 図 5 5 】

```

-<dcase:link id="id19" name="LINK_73" source="id11" target="id12">
<dcase:description />
<dcase:properties />
</dcase:link>
-<dcase:link id="id20" name="LINK_74" source="id11" target="id13">
<dcase:description />
<dcase:properties />
</dcase:link>
-<dcase:link id="id21" name="LINK_70" source="id11" target="id22">
<dcase:description />
<dcase:properties />
</dcase:link>
-<dcase:link id="id23" name="LINK_78" source="id12" target="id24">
<dcase:description />
<dcase:properties />
</dcase:link>
-<dcase:link id="id25" name="LINK_79" source="id13" target="id24">
<dcase:description />
<dcase:properties />
</dcase:link>
</dcase:links>
</dcase:dcase>

```

【 図 5 6 】

```

/* Monitor:M_1 から生成されたスクリプト「ロボット1モニター」*/
if exec /home/et2010/damd/RobotCheck.sh main id="id12"
then exec /home/et2010/damd/RobotError.sh main

/* Monitor:M_2 から生成されたスクリプト「ロボット2モニター」*/
if exec /home/et2010/damd/RobotCheck.sh backup id="id13"
then exec /home/et2010/damd/RobotError.sh backup

/* Monitor:M_3 から生成されたスクリプト「カメラ1モニター」*/
if exec /home/et2010/damd/CameraCheck.sh 1 id="id14"
then exec /home/et2010/damd/CameraStatus.sh 1 camera-status

/* Monitor:M_4 から生成されたスクリプト「カメラ2モニター」*/
if exec /home/et2010/damd/CameraCheck.sh 2 id="id22"
then exec /home/et2010/damd/CameraStatus.sh 2 camera-status

/* Monitor:M_5 から生成されたスクリプト「カメラが切れたらfail-overする」*/
if exec /home/et2010/damd/videocheck.sh id="id26"
then exec /home/et2010/damd/demo1.sh fail-over camera-error

```

【 手続 補正書 】

【 提出日 】平成24年9月24日(2012.9.24)

【 手続 補正 1 】

【 補正対象書類名 】特許請求の範囲

【 補正対象項目名 】全文

【 補正方法 】変更

【 補正の内容 】

【 特許請求の範囲 】

【 請求項 1 】

対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、

上記対象システムの開発時あるいは運用時における上記対象システムの更新時に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルを実行する変化対応サイクル実行装置と、

上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行する障害対応サイクル実行装置と、

上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信手段と、を備えることを特徴とするディペンダビリティ維持システム。

【 請求項 2 】

上記ディペンダビリティ記述データを格納するディペンダビリティ記述格納部を備え、上記変化対応サイクル実行装置および上記障害対応サイクル実行装置が、上記ディペン

ダビリティ記述格納部に格納されている上記ディペンダビリティ記述データを共有しながら、上記変化対応サイクルおよび上記障害対応サイクルを一方ずつまたは両方同時に実行することを特徴とする請求項1に記載のディペンダビリティ維持システム。

【請求項3】

上記障害対応サイクル実行装置が上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、当該対象システムを再構成する再構成手段を備えることを特徴とする請求項1に記載のディペンダビリティ維持システム。

【請求項4】

上記ディペンダビリティ記述データには上記再構成の手順が記載されており、

上記再構成手段は、上記ディペンダビリティ記述データに記載されている再構成の上記手順に従って、当該対象システムを再構成することを特徴とする請求項3に記載のディペンダビリティ維持システム。

【請求項5】

上記ディペンダビリティ記述データの変更要求を受信したとき、上記変化対応サイクル実行装置が上記ディペンダビリティ記述データを変更することを特徴とする請求項1に記載のディペンダビリティ維持システム。

【請求項6】

上記ディペンダビリティ記述データを上記ディペンダビリティ記述格納部から取得し、上記対象システムのディペンダビリティの価値を定量的に示す評価値を求めるディペンダビリティ値決定手段を備え、

上記ディペンダビリティ値決定手段が、上記評価値を、上記変化対応サイクルまたは上記障害対応サイクルにおいて決定することを特徴とする請求項1に記載のディペンダビリティ維持システム。

【請求項7】

上記ディペンダビリティ記述データは、互いに関連づけられたゴールノードとモニタノードとの組を規定するデータであり、

上記ゴールノードは、要求・仕様がゴール形式にて記述されたノードであり、

上記モニタノードは、上記ゴールノードに記述されたゴールが充足されていることを表明するエビデンスであるとともに、上記対象システムの監視点に対応付けられており、

上記ディペンダビリティ値決定手段は、対応付けられた上記監視点から取得されたデータがモニタノードに関連づけられた変動許容範囲に基づいて、上記評価値を計算することを特徴とする請求項6に記載のディペンダビリティ維持システム。

【請求項8】

上記ディペンダビリティ値決定手段は、上記対象システムをモニタして取得されたモニタ値が変動許容範囲に対して良好であったエビデンスに基づいて、上記評価値を計算することを特徴とする請求項7に記載のディペンダビリティ維持システム。

【請求項9】

上記障害対応サイクル実行装置が上記障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに含まれるスクリプトを実行するスクリプト処理手段を備え、

上記スクリプトは上記対象システムを変動許容範囲の状態に回復させるシナリオを含むことを特徴とする請求項1に記載のディペンダビリティ維持システム。

【請求項10】

上記対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部と、

上記ディペンダビリティ記述データを格納するディペンダビリティ記述格納部と、

上記障害監視制御部の動作を制御する障害監視制御用データを、上記ディペンダビリティ記述格納部から読み出した上記ディペンダビリティ記述データから生成して、上記障害監視制御部に供給するディペンダビリティ記述変換手段と、を備え、

上記障害監視制御用データに従って、上記障害監視制御部が動作することにより、上記対象システムの状態を監視し、必要な場合に対策を実行することを特徴とする請求項1に

記載のディペンダビリティ維持システム。

【請求項 1 1】

上記対象システムの状態を監視する 1 以上のモニタモジュールと、
必要な場合に上記対象システムに対して対策を実行する 1 以上のアクションモジュールと、

上記障害監視制御部の制御により、上記モニタモジュールの選択および動作の制御を行うモニタ管理部と、

上記障害監視制御部の制御により、上記アクションモジュールの選択および動作の制御を行うアクション管理部と、を含むことを特徴とする請求項 1 0 に記載のディペンダビリティ維持システム。

【請求項 1 2】

他のディペンダビリティ維持システムとネットワークを介して接続されていることを特徴とする請求項 1 に記載のディペンダビリティ維持システム。

【請求項 1 3】

請求項 1 に記載のディペンダビリティ維持システムを構成する変化対応サイクル実行装置。

【請求項 1 4】

請求項 1 に記載のディペンダビリティ維持システムを構成する障害対応サイクル実行装置。

【請求項 1 5】

上記ディペンダビリティ記述データをディペンダビリティ記述格納部から読み出すとともに、

上記対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御するための障害監視制御用データを、読み出した上記ディペンダビリティ記述データから生成するディペンダビリティ記述変換手段を備えたことを特徴とする請求項 1 4 に記載の障害対応サイクル実行装置。

【請求項 1 6】

上記障害監視制御部は、上記対象システムの状態を監視し、必要な場合に対策を実行するために使用するモジュールを、複数のモジュールから選択して制御可能であり、

上記ディペンダビリティ記述データは、1 つの記述が、上記モジュールを特定するためのモジュール特定情報を値として設定可能な変数を少なくとも含む形式であり、

上記ディペンダビリティ記述変換手段は、あらかじめ設定された、上記モジュール特定情報と上記障害監視制御用データとの対応関係を示す情報に基づいて、上記ディペンダビリティ記述データに含まれる変換対象の記述を、該記述に含まれるモジュール特定情報に対応する障害監視制御用データに変換することを特徴とする請求項 1 5 に記載の障害対応サイクル実行装置。

【請求項 1 7】

上記ディペンダビリティ記述データの 1 つの記述から変数部分を除いた固定部分に対応して、上記対応関係が設定されており、

上記ディペンダビリティ記述変換手段は、上記ディペンダビリティ記述データの記述に含まれる上記固定部分に対応する上記対応関係を参照して、該記述を対応する障害監視制御用データに変換することを特徴とする請求項 1 6 に記載の障害対応サイクル実行装置。

【請求項 1 8】

対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムの制御方法であって、

上記ディペンダビリティ維持システムの変化対応サイクル実行装置が、上記対象システムの開発時あるいは運用時における上記対象システムの更新時に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルと、

上記ディペンダビリティ維持システムの障害対応サイクル実行装置が、上記対象システ

ムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルと、

上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信ステップと、を含むことを特徴とするディペンダビリティ維持システムの制御方法。

【請求項 19】

上記ディペンダビリティ記述データをディペンダビリティ記述格納部から読み出す読み出しステップと、

上記対象システムの状態を監視し、必要な場合に対策を実行する障害監視制御部の動作を制御するための障害監視制御用データを、読み出した上記ディペンダビリティ記述データから生成する変換ステップと、を含むことを特徴とする請求項 18 に記載のディペンダビリティ維持システムの制御方法。

【請求項 20】

請求項 1 から 12 のいずれか 1 項に記載のディペンダビリティ維持システムの上記各手段としてコンピュータを機能させるための制御プログラム。

【請求項 21】

請求項 15 から 17 のいずれか 1 項に記載の障害対応サイクル実行装置の上記ディペンダビリティ記述変換手段としてコンピュータを機能させるための制御プログラム。

【請求項 22】

請求項 20 または 21 に記載の制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正の内容】

【0001】

技術分野

[0001]

本発明は、時間軸上で変化可能なディペンダビリティ要求に対して、ある時刻における当該要求を表現したディペンダビリティ記述データを備える対象システムにおけるディペンダビリティの価値を計測、評価することを可能にし、当該対象システムのディペンダビリティを維持する装置等に関するものである。

背景技術

[0002]

昨今、銀行のオンラインシステムの停止、携帯電話や IP 電話の通信障害、各種商用サービスのセキュリティ障害、等の重要な社会インフラストラクチャを担うシステムやサービス停止が目立ってきており、我々の生活に影響を与えている。その原因は、ひとえにそれらの組み込み型コンピュータシステムを利用した商品やサービスにおいて、その規模や複雑度が劇的に増してきていることに帰することができる。その原因をさらに掘り下げて調べてみると、人為的なミスが原因であることがかなりのケースで存在している。

[0003]

従来から、コンピュータシステムの信頼性、可用性、保守性、安全性、完全性、機密性に関しては、ディペンダビリティというコンピュータシステムの備えるべき性質として議論されてきている（非特許文献 3）。組み込みシステム開発においては、最初に開発計画を立て、対象システムやサービスの機能要件および非機能要件を仕様としてきちり書き出し、検証やテストを長期にわたって行い、デプロイする手法がとられてきた。しかし、上述のように、障害が日に日に件数を増してきている。CMMI や ISO 26262 をはじめとする規格では、人為的エラーを減らす試みもなされている。し

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0004

【補正方法】変更

【補正の内容】

【0004】

, vol. 1, no. 1, pp. 11 - 33, Jan. - Mar. 2004, doi: 10.1109/TDSC.2004.2.

非特許文献4：所眞理雄、JST-CREST「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム - DEOSプロジェクト - White Paper Version 2.0」、DEOS-FY201-WP-02J、2010/12/01

非特許文献5：Jin Nakazawa, Yutaka Matsuno, and Hideyuki Tokuda, "Evaluating Degree of Systems' Dependability with Semi-Structured Assurance Case" Proceedings of the 13th European Workshop on Dependable Computing (EWDC 2011), 2011.

非特許文献6：DEOS Core Team, "Open Systems Dependability Core - Dependability Metrics - オープン・システム・ディペンダビリティ・コア - ディペンダビリティメトリクス - ", [online], 2009.09.04, 科学技術振興機構, [2012年2月3日検索], インターネット<URL: <http://www.crest-os.jst.go.jp/topics/deos2009/metrics.pdf>>

発明の概要

発明が解決しようとする課題

[0011]

システム規模の拡大や複雑度の増大は、当該システムやそれが提供するサービスに関する要求・仕様をも複雑にしている。そのため、すべての要求を完全にシステム開発前に抽出し、全ての仕様を完全にシステム開発前に記述することが不可能（仕様の不完全性）になっている。仕様の不完全性は対応する実装も完全ではなく（実装の不完全性）、当該システムが提供するサービスの振る舞いも完全には把握することができない。その結果、「何をどこまで保証したら良いのか？」「保証可能なのか？」も分からない。また、これらの不完全性はステークホルダ間でのシステムやサービスに対する要求理解の違い（誤解）を招いたり、人為的なミスを誘導したりする。

[0012]

さらに、先に述べた環境の変化に対応するための要求が新たに発生する。また、開発時点の要求にも修正が必要になる。「環境がどのように変化するか？」は事前には分かり得ないので、「現在のシステムの動作が、環境変化に確実に対応できるか？」もわからない。このような変化に対する不確実性はシステム動作予測を困難にし、システム障害に繋がる。

[0013]

これらの不完全性や不確実性はシステム障害が発生した際に、その原因調査に時間がかかるようになり、またその適切な対応を困難にし、ステークホルダが自らの説明責任を果たすことも難しくなる。

[0014]

このような状況下において、従来、非特許文献2においては、システム開

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0005

【補正方法】変更

【補正の内容】

【0005】

発時点で想定される障害対応を Safety Caseとして記述し、ステークホルダ間で合意することで、障害発生時点でのステークホルダの説明責任の達成を可能にしている。しかし、環境の変化に起因する Safety Caseの変更には対応できない。

[0015]

また、非特許文献3は、Faults - Errors - Failuresという変化で障害をとらえることで障害の発生そのものを抑えようとしている。しかし、前記不完全性や不確実性を扱うことはできない。

[0016]

また、非特許文献5、6には、前記不完全性や不確実性を扱うための手法に関して記載されている。しかし、その記述は実行環境とは分離されており、また、記述をモデル化しディペンダビリティの価値を計測・評価することに関しては設計時のみにて可能であるに過ぎない。

[0017]

一方、特許文献1はEnterprise Architectureにおける複雑性を数学的に制御する手法に関して述べている。複雑性が制御できればディペンダビリティの向上につながることを期待できるが、前記不完全性や不確実性までは扱っていない。

[0018]

また、従来 of システム開発の手法では、複数のステークホルダ群からの要求は合意された仕様書にまとめられていないし、仕様書にまとめることが容易でないことに加えて、運用後にシステムの一部に変更があった場合に、仕様書とシステム実装の変更とを無矛盾に維持することができなかつた。

[0019]

例えば、上記 Safety Caseでも、Safety Caseに記載されている、ステークホルダが Safety Case上で合意した障害対応内容と、実際の対象システムの障害対応やシステム監視制御部の開発・追加などが無矛盾であることを維持する実用的なシステムは存在していない。このような Safety Caseと対象システムとの無矛盾性は、ほぼ人手で行わなければならない、システムの更新と、Safety Caseの更新を無矛盾に行うことは、時間的なずれが生じるなど、問題点が多い。一般に、Safety Caseのような、認証機関などを含む、広範囲のステークホルダが理解可能な形式で記述されたドキュメントと、システムの実際の動作との間の無矛盾性を保証する実用的なシステムは、発明者が知る範

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0006

【補正方法】変更

【補正の内容】

【0006】

図においてははない。

[0020]

本発明は、上記の問題点に鑑みてなされたものであり、その目的は、不完全性と不確実性が潜在的に存在するオープン環境において、対象システムのディペンダビリティの維持を支援することを可能にすると共に、対象システムの仕様書群と当該システムの実装を無矛盾に維持することを容易にするディペンダビリティ維持装置等を実現することにある。なお、本発明では、システムのディペンダビリティを非特許文献3などの定義に、複数のステークホルダ間での合意を含んだものに拡張する。

課題を解決するための手段

[0021]

上記課題を解決するために、本発明に係るディペンダビリティ維持システムは、対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムであって、上記対象システムの開発時あるいは運用時における上記対象システムの更新時に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルを実行する変化対応サイクル実行装置と、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行する障害対応サイクル実行装置と、上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信手段と、を備えることを特徴としている。

[0 0 2 2]

また、本発明に係るディペンダビリティ維持システムの制御方法は、対象システムのディペンダビリティを維持するためのディペンダビリティ維持システムの制御方法であって、上記ディペンダビリティ維持システムの変化対応サイクル実行装置が、上記対象システムの開発時あるいは運用時における上記対象システムの更新時

【 手 続 補 正 7 】

【 補 正 対 象 書 類 名 】 明 細 書

【 補 正 対 象 項 目 名 】 0 0 0 7

【 補 正 方 法 】 変 更

【 補 正 の 内 容 】

【 0 0 0 7 】

に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルと、上記ディペンダビリティ維持システムの障害対応サイクル実行装置が、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルと、上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信ステップと、を含むことを特徴としている。

[0 0 2 3]

上記の構成によれば、障害対応サイクル実行装置において、対象システムの障害発生または障害予兆を検知し、対象システムの停止が回避であると判断したとき、変化対応サイクル実行装置に対して、ディペンダビリティ記述データの変更の要求を送信することができる。

[0 0 2 4]

これにより、変化対応サイクル実行装置は、障害対応サイクル実行装置が送信したディペンダビリティ記述データの変更の要求を受信したとき、当該要求に応じて、上記ディペンダビリティ記述データを変更することが可能となる。

[0 0 2 5]

したがって、障害対応サイクル実行装置と、変化対応サイクル実行装置とが連携して、障害対応サイクルにおいて、対象システムの障害発生または障害予兆を検知し、対象システムの停止が回避であれば、変化対応サイクルにおいて、ディペンダビリティ記述データを変更するという、一連のプロセスを円滑に実行することができる。よって、オープン環境において、対象システムのディペンダビリティを継続的に維持することが可能となるという効果を奏する。

[0 0 2 6]

なお、上記の変化対応サイクル実行装置および障害対応サイクル実行装置を含むディペ

ンダビリティ維持システムは、コンピュータによって実現してもよく、この

【手続補正 8】

【補正対象書類名】明細書

【補正対象項目名】0008

【補正方法】変更

【補正の内容】

【0008】

場合には、コンピュータを上記の各手段として動作させることにより上記ディペンダビリティ維持システムをコンピュータにて実現させるディペンダビリティ維持システムのプログラム、およびそれを記録したコンピュータ読み取り可能な記録媒体も、本発明の範疇に入る。

[0027]

[0028]

[0029]

[0030]

[0031]

[0032]

[0033]

[0034]

発明の効果

[0035]

以上のように、本発明に係るディペンダビリティ維持システムは、対象システムの開発時あるいは運用時における上記対象システムの更新時に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルを実行する変化対応サイクル実行装置と、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルを実行する障害対応サイクル実行装置と、上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信手段と、を備える構成である。

[0036]

また、本発明に係るディペンダビリティ維持システムの制御方法は、ディペンダビリティ維持システムの変化対応サイクル実行装置が、対象システムの開発時ある

【手続補正 9】

【補正対象書類名】明細書

【補正対象項目名】0009

【補正方法】変更

【補正の内容】

【0009】

いは運用時における上記対象システムの更新時に、上記対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データを上記対象システムに関する要求・仕様の変更に合わせて追加・修正する変化対応サイクルと、上記ディペンダビリティ維持システムの障害対応サイクル実行装置が、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルと、上記障害対応サイクル実行装置が障害発生または障害予兆を検知し、上記対象システムの停止が回避であるとき、上記変化対応サイクル実行装置に対して、上記ディペンダビリティ記述データの変更要求を送信する変更要求送信ステップと、を含む方法である。

[0037]

それゆえ、障害対応サイクル実行装置と、変化対応サイクル実行装置とが連携して、障害対応サイクルにおいて、対象システムの障害発生または障害予兆を検知し、対象システムの停止が回避であれば、変化対応サイクルにおいて、ディペンダビリティ記述データを変更するという、一連のプロセスを円滑に実行することができる。よって、オープン環境において、対象システムのディペンダビリティを継続的に維持することが可能となるという効果を奏する。

[0 0 3 8]

[0 0 3 9]

[0 0 4 0]

[0 0 4 1]

本発明のさらに他の目的、特徴、および優れた点は、以下に示す記載によって十分に分かるであろう。また、本発明の利点は、添付図面を参照した次の説明で明白になるであろう。

図面の簡単な説明

[0 0 4 2]

[図 1]

本発明の一実施形態を示すものであり、ディペンダビリティ維持システムの構成

【 手 続 補 正 1 0 】

【 補 正 対 象 書 類 名 】 明 細 書

【 補 正 対 象 項 目 名 】 0 0 1 0

【 補 正 方 法 】 変 更

【 補 正 の 内 容 】

【 0 0 1 0 】

の概略を示す機能ブロック図である。

【 手 続 補 正 1 1 】

【 補 正 対 象 書 類 名 】 明 細 書

【 補 正 対 象 項 目 名 】 0 0 5 3

【 補 正 方 法 】 変 更

【 補 正 の 内 容 】

【 0 0 5 3 】

ディペンダビリティ記述データM1を記述し、さらに、ランタイムコンピュータ102における内部表現であるディペンダビリティ記述データ内部表現M2に変換する。詳細には、ワークスペースコンピュータ101の編集ツール部101-02において、ディペンダビリティ記述データ記述・表示ツール511-01がディペンダビリティ記述データM1を記述した後、それを、ディペンダビリティ記述データ変換部511-03によってディペンダビリティ記述データ内部表現M2に変換する。

[0 1 9 1]

ランタイムコンピュータ102では、ワークスペースコンピュータ101から取得したディペンダビリティ記述データ内部表現M2を更新部102-01がディペンダビリティ記述データ記録部(ディペンダビリティ記述データ格納部)511-05に記録する。また、更新部102-01は、ディペンダビリティ記述データ内部表現M2に基づいて、モニターノードとモニタリング部102-03における各種監視モジュール(監視モジュール群511-m)との対応関係をモニターノード/モジュール対応テーブル511-06に記録する。なお、図20(b)には監視モジュール群511-mの各監視モジュールとして、CPU監視モジュール511-08、HDD監視モジュール511-09、プロセス監視モジュール511-10、が記載されているが、他の監視モジュール511-11があっても良い。

[0 1 9 2]

モニタリング部102-03における前記監視モジュールによって得られたデータ(モニタ値)は収集部510-13によって集められ、実行状況検証部510-12によって

、モニタノード/モジュール対応テーブル511-06に基づき、前記ディペンダビリティ記述データにおけるモニタノードの値として記録される。例えば、ディペンダビリティ記述データの各モニタノードに前記モニタ値の変動許容範囲を対応付けて記載しておき、モニタ値がその変動許容範囲から外れたノードを障害発生ノードとして検出する。また、前記監視モジュール群511-mからのモニタ値はD値計算部102-06においてノードに対するD値の計算に利用される。よって、例えば図20(a)のディペンダビリティ記述データの一部510では、2個のモニタ

【手続補正12】

【補正対象書類名】明細書

【補正対象項目名】0108

【補正方法】変更

【補正の内容】

【0108】

在的に存在するオープン環境において、ステークホルダ間における要求の誤解、環境の変化に対する対応不能、障害対応の失敗という3つの問題に対するソリューションを提供することができる。オープン環境でのシステム開発及びシステム運用に広く利用することができる。

[0441]

また、本発明の障害対応システムおよびディペンダビリティ維持装置によれば、各ステークホルダの合意の記述を容易に行うとともに、監視制御対象システムの一部に更新があった場合に、合意の記述とモジュールの実装とが無矛盾に開発・更新されるように維持することができる。よって、例えば、組込みシステムから、インターネットなどでつながれた多数システムまで、多様なシステムのディペンダビリティを維持するため装置・方法として好適である。

符号の説明

[0442]

100 (100U, 100S, 100B, 100P) ディペンダビリティ維持システム

101 ワークスペースコンピュータ(ディペンダビリティ維持装置、ワークスペース装置、変化対応サイクル実行装置)

101-05 D値計算部(ディペンダビリティ値決定手段)

102 ランタイムコンピュータ(ディペンダビリティ維持装置、ランタイム装置、障害対応サイクル実行装置)

102-04 再構成部(再構成手段)

102-06 D値計算部(ディペンダビリティ値決定手段)

102-07 スクリプト処理部(スクリプト処理手段)

901-01 ディペンダビリティ記述データ入力部(記述データ取得手段)

902-01 ソフトウェア入力部(記述データ取得手段)

902-02 変更送出部(変更要求送信手段)

P300 変化対応サイクル

P301 障害対応サイクル

【手続補正13】

【補正対象書類名】明細書

【補正対象項目名】0109

【補正方法】変更

【補正の内容】

【0109】

1 障害対応システム

2 監視制御対象システム

10 D-Case格納部(ディペンダビリティ記述格納部)

- 2 0 ディペンダビリティ維持装置
- 2 1 D - C a s e 変換部 (ディペンダビリティ記述変換部、ディペンダビリティ記述
変換手段)
- 2 2 D - C a s e パターン < = > モジュール対応テーブル
- 3 0 障害監視制御部
- 4 0 モニタ管理部
- 5 0 モニタモジュール群 (モジュール)
- 6 0 アクション管理部
- 7 0 アクションモジュール群 (モジュール)
- S 2 読み出しステップ
- S 3 変換ステップ

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/JP2011/076219
A. CLASSIFICATION OF SUBJECT MATTER G06F11/30(2006.01)i, G06F11/36(2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F11/30, G06F11/36 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2012 Kokai Jitsuyo Shinan Koho 1971-2012 Toroku Jitsuyo Shinan Koho 1994-2012 Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DEOS Core Team, "Open Systems Dependability Core -Dependability Metrics-", [online], 04	1, 7-8, 11-15, 18-22
Y	September 2009 (04.09.2009), Japan Science and Technology Agency, [retrieval date 03 February 2012 (03.02.2012)], Internet <URL: http://www.crest-os.jst.go.jp/topics/deos2009/metrics.pdf > particularly, pages 1 to 18	2-4, 6, 9-10
A		5, 16-17
Y	JP 2009-169657 A (Hitachi, Ltd.), 30 July 2009 (30.07.2009), paragraphs [0249] to [0256], [0291] to [0297]; fig. 14A, 15 & US 2009/0182866 A1	2-4, 6, 9-10
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 06 February, 2012 (06.02.12)		Date of mailing of the international search report 14 February, 2012 (14.02.12)
Name and mailing address of the ISA/ Japanese Patent Office		Authorized officer Telephone No.
Facsimile No.		

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/076219

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2008-537203 A (Lehman Brothers Inc.), 11 September 2008 (11.09.2008), paragraphs [0016] to [0020] & US 2006/0190766 A1 & EP 1851632 A & WO 2006/091400 A2	2-4, 6, 9-10
Y	JP 2000-99359 A (International Computers Ltd.), 07 April 2000 (07.04.2000), paragraph [0016] & US 6460144 B1 & GB 9819523 A & GB 9819524 A & GB 2345769 A & GB 9819523 A0 & EP 987630 A2 & DE 69927223 D & DE 69927223 T & AU 4738899 A & ZA 9905736 A & AU 753898 B	2-4, 6, 9-10
A	JP 2007-249739 A (NEC Corp.), 27 September 2007 (27.09.2007), paragraphs [0003], [0039], [0063] to [0066]; fig. 5 to 6, 9 (Family: none)	16-17

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/076219

Box No. II	Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)
<p>This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:</p> <p>1. <input type="checkbox"/> Claims Nos.: because they relate to subject matter not required to be searched by this Authority, namely:</p> <p>2. <input type="checkbox"/> Claims Nos.: because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:</p> <p>3. <input type="checkbox"/> Claims Nos.: because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).</p>	
Box No. III	Observations where unity of invention is lacking (Continuation of item 3 of first sheet)
<p>This International Searching Authority found multiple inventions in this international application, as follows:</p> <p style="padding-left: 40px;">See extra sheet</p> <p>1. <input type="checkbox"/> As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.</p> <p>2. <input checked="" type="checkbox"/> As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.</p> <p>3. <input type="checkbox"/> As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:</p> <p>4. <input type="checkbox"/> No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:</p> <p>Remark on Protest</p> <p><input type="checkbox"/> The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.</p> <p><input type="checkbox"/> The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.</p> <p><input type="checkbox"/> No protest accompanied the payment of additional search fees.</p>	

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/076219

Continuation of Box No.III of continuation of first sheet(2)

Document 1 discloses an invention of a dependability supporting method which is provided when an apparatus is executed, developed and updated on its maintenance, and in which a dependability score (corresponding to the evaluation value) is derived (digitalized) using D-case (corresponding to the dependability description data).

Therefore, the invention in claims 1 and 12-14 does not appear to be novel to the invention disclosed in document 1, and does not have a special technical feature. Accordingly, claims include two inventions (invention groups) having the following special technical features.

Invention 1: The invention in claims 1 - 14

An invention which executes at least either one of: a change adapting cycle in which dependability description data is changed in accordance with a change in requirements/specifications when an object system is developed or updated; and a failure adapting cycle in which system stop of the object system is avoided on the basis of the dependability description data when an occurrence of failure or a sign of failure is detected while the object system is being operated.

Invention 2: The invention in claims 15 - 22

An invention which reads dependability description data describing requirements/specifications as to dependability of a monitoring control object system out of a dependability description storing unit, generates failure monitoring control data for controlling an operation of a failure monitoring control unit on the basis of the dependability description data thus read, and supplies the data thus generated to the failure monitoring control unit.

国際調査報告		国際出願番号 PCT/JP2011/076219									
A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. G06F11/30(2006.01)i, G06F11/36(2006.01)i											
B. 調査を行った分野 調査を行った最小限資料 (国際特許分類 (IPC)) Int.Cl. G06F11/30, G06F11/36											
最小限資料以外の資料で調査を行った分野に含まれるもの <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922-1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971-2012年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996-2012年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994-2012年</td> </tr> </table>				日本国実用新案公報	1922-1996年	日本国公開実用新案公報	1971-2012年	日本国実用新案登録公報	1996-2012年	日本国登録実用新案公報	1994-2012年
日本国実用新案公報	1922-1996年										
日本国公開実用新案公報	1971-2012年										
日本国実用新案登録公報	1996-2012年										
日本国登録実用新案公報	1994-2012年										
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)											
C. 関連すると認められる文献											
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号									
X	DEOS Core Team, "Open Systems Dependability Core -Dependability Metrics-", [online], 2009.09.04, 科学技術振興機構, [2012年2月3日検索], インターネット <URL: http://www.crest-os.jst.go.jp/topics/deos2009/metrics.pdf> 特に、p.1-18の記載を参照	1, 7-8, 11-15, 18-22									
Y	JP 2009-169657 A (株式会社日立製作所) 2009.07.30, 段落【0249】 - 【0256】, 【0291】 - 【0297】, 第14A, 15図 & US 2009/0182866 A1	2-4, 6, 9-10 5, 16-17									
A		2-4, 6, 9-10									
Y		5, 16-17									
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。											
* 引用文献のカテゴリー		の日の後に公表された文献									
「A」特に関連のある文献ではなく、一般的技術水準を示すもの		「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの									
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの		「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの									
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)		「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの									
「O」口頭による開示、使用、展示等に言及する文献		「&」同一パテントファミリー文献									
「P」国際出願日前で、かつ優先権の主張の基礎となる出願											
国際調査を完了した日 06.02.2012		国際調査報告の発送日 14.02.2012									
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 稲垣 良一	5B 4680								
		電話番号 03-3581-1101 内線 3545									

国際調査報告

国際出願番号 PCT/JP2011/076219

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y	JP 2008-537203 A (リーマン・ブラザーズ・インコーポレーテッド) 2008.09.11, 段落【0016】-【0020】 & US 2006/0190766 A1 & EP 1851632 A & WO 2006/091400 A2	2-4, 6, 9-10
Y	JP 2000-99359 A (インターナショナル コンピューターズ リミテッド) 2000.04.07, 段落【0016】 & US 6460144 B1 & GB 9819523 A & GB 9819524 A & GB 2345769 A & GB 9819523 A0 & EP 987630 A2 & DE 69927223 D & DE 69927223 T & AU 4738899 A & ZA 9905736 A & AU 753898 B	2-4, 6, 9-10
A	JP 2007-249739 A (日本電気株式会社) 2007.09.27, 段落【0003】, 【0039】, 【0063】 - 【0066】, 第5-6, 9 図 (ファミリーなし)	16-17

国際調査報告

国際出願番号 PCT/J P 2 0 1 1 / 0 7 6 2 1 9

第II欄 請求の範囲の一部の調査ができないときの意見 (第1ページの2の続き)

法第8条第3項 (PCT17条(2)(a))の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

1. 請求項 _____ は、この国際調査機関が調査をすることを要しない対象に係るものである。つまり、
2. 請求項 _____ は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、
3. 請求項 _____ は、従属請求の範囲であってPCT規則6.4(a)の第2文及び第3文の規定に従って記載されていない。

第III欄 発明の単一性が欠如しているときの意見 (第1ページの3の続き)

次に述べるようにこの国際出願に二以上の発明があるとこの国際調査機関は認めた。
特別ページ参照。

1. 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求項について作成した。
2. 追加調査手数料を要求するまでもなく、すべての調査可能な請求項について調査することができたので、追加調査手数料の納付を求めなかった。
3. 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったため、この国際調査報告は、手数料の納付のあった次の請求項のみについて作成した。
4. 出願人が必要な追加調査手数料を期間内に納付しなかったため、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求項について作成した。

追加調査手数料の異議の申立てに関する注意

- 追加調査手数料及び、該当する場合には、異議申立手数料の納付と共に、出願人から異議申立てがあった。
- 追加調査手数料の納付と共に出願人から異議申立てがあったが、異議申立手数料が納付命令書に示した期間内に支払われなかった。
- 追加調査手数料の納付はあったが、異議申立てはなかった。

様式PCT/ISA/210 (第1ページの続葉(2)) (2009年7月)

文献1には、機器の実行時、開発時、保守更新時に提供されるディペンダビリティ支援方法であって、D-case (ディペンダビリティ記述データに対応) を用いて、ディペンダビリティスコア (評価値に対応) を導出 (数値化) する、ディペンダビリティ支援方法の発明が記載されている。

したがって、請求項 1, 12-14 に係る発明は、文献1に記載された発明に対して新規性が認められず、特別な技術的特徴を有しない。したがって、請求の範囲には、以下の特別な技術的特徴を有する2の発明 (群) が含まれる。

(発明1) 請求項 1-14 に係る発明

対象システムの開発時あるいは更新時に、ディペンダビリティ記述データを要求・仕様の変更に合わせて変更する変化対応サイクル、および、上記対象システムの運用時に、障害発生または障害予兆を検知したとき、上記ディペンダビリティ記述データに基づいて、上記対象システムの停止を回避する障害対応サイクルの少なくとも何れか一方を実行する発明。

(発明2) 請求項 15-22 に係る発明

監視制御対象システムのディペンダビリティに関する要求・仕様を記述したディペンダビリティ記述データをディペンダビリティ記述格納部から読み出し、読み出したディペンダビリティ記述データから障害監視制御部の動作を制御する障害監視制御用データを生成し、上記障害監視制御部に供給する発明。

フロントページの続き

(81) 指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN

(72) 発明者 松野 裕

日本国千葉県柏市あけぼの4 - 4 - 5 - 2 0 1

Fターム(参考) 5B042 JJ06 KK17

5B376 BB19 DA21 DA28

(54) 【発明の名称】ディペンダビリティ維持システム、変化対応サイクル実行装置、障害対応サイクル実行装置、ディペンダビリティ維持システムの制御方法、制御プログラムおよびそれを記録したコンピュータ読み取り可能な記録媒体

(注) この公表は、国際事務局(WIPO)により国際公開された公報を基に作成したものである。なおこの公表に係る日本語特許出願(日本語実用新案登録出願)の国際公開の効果は、特許法第184条の10第1項(実用新案法第48条の13第2項)により生ずるものであり、本掲載とは関係ありません。