ディペンダブルシステム構築と運用の技術

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター センター長
屋代　眞

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター 研究員
髙村　博紀

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター 研究員
松原　茂

現在の社会インフラや生活環境を支える情報技術は目覚ましく進歩しているが、一方でシステム障害は無くならず、時に人命や社会に大きな影響を与えている。安全、安心、快適な生活を支えるディペンダブルな情報システムを構築し運用することを目指して科学技術振興機構 CREST※1 研究領域で取り組んできた DEOS プロジェクトの成果を紹介する。

1 はじめに

現在の社会インフラや生活環境を支えているコンピュータシステムは膨大なソフトウェアにより動作している。ソフトウェアはプログラミング言語、開発環境、開発・運用ツール、開発プロセスなどの研究により進歩を遂げてきたが、システムの障害は無くならない。科学技術振興機構はディペンダブルなシステムを構築するためのソフトウェアの基盤技術を研究するために CREST 研究領域「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」（以下、DEOS プロジェクトと略す）を2006年に立ち上げた。DEOSプロジェクトは本年度で終了するが、本稿ではその成果の中から開発・運用プロセスに関する成果を解説する。DEOS プロジェクト成果の詳細は「DEOS プロジェクト研究成果集」[1] を参照されたい。

2 なぜ情報システムの障害は無くならないか？

今日、情報システムなしに生活を送ることは不可能になっている。携帯電話やネット家電によるサービスは言うに及ばず、行政、金融、流通、医療、交通、防衛、エネルギー、通信、放送システムなどいたるところでその恩恵を受けている。現代のシステムはサービス内容が多岐にわたるため、システムの規模は巨大にならざるを得ず、またその構造は複雑なものになっている。以下、今日のディペンダブルシステム構築と運用における課題を要約する。

①ブラックボックスの存在

複雑なシステムではコスト・納期などの制約からすべてをゼロから開発することは稀であり、既存システムを局所的に改造して使用し続けることが多い。その際に COTS※2 やレガシーソフトウェアがブラックボックスとして使われることが多く、そのことがシステムの把握や障害の分析を難しくしている。

②複数の開発組織の連携

システム開発に係る開発者・運用者は複数の企業や国

【脚注】
※1　戦略的創造研究推進事業（Core Research for Evolutional Science and Technology）
※2　Commercial Off-The-Shelf : ここでは既成品として販売やリースされるソフトウェア

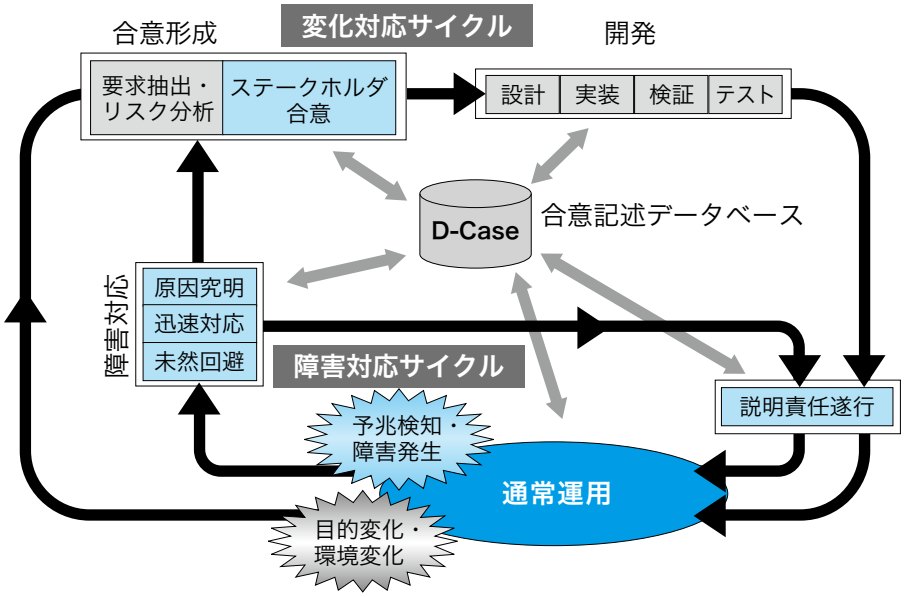

図 1　DEOS プロセス

まとめると、今日の情報システムは変化し続けるオープンシステムであり、①、②、③のような不完全性、④、⑤のような不確実性という排除できない問題がある。オープンシステムをマネージするために必要な能力、すなわち実環境の中で長期的に運用されるシステムがその目的や環境の変化に対応し、システムに関する説明責任遂行を継続的に支援しつつ、利用者が期待するサービスを継続的に提供し続ける能力を、「オープンシステムディペンダビリティ（OSD：Open Systems Dependability）」と名付けた。OSD に関する議論や内容は文献 [2] に詳しく解説されている。

をまたいで存在していることも少なくなく、システムの全容を完全に把握することが難しい。また、サプライチェーンマネジメントの問題も関わってくる。

③開発者と運用者の関係

開発者と運用者の間でお互いの現場や用語の違いの認識が不十分なことが障害に繋がることも多い。最近ではできる限り言葉を定義し共有することが普通になって来てはいるものの、完全にあいまいさを排除することはできない。

④環境の変化

インターネットを通じたサービスの提供、情報の交換、新規プログラムや更新のダウンロードなどの普及により、情報システムが使われる環境はそのライフサイクルの中で常に変化し続けることが多くなり、環境の変化により新たな問題や障害が起こることがある。

⑤要求の変化

長期にわたり継続的にサービスを提供するシステムでは、開発された時のサービスの目的や利用者の要求が、ライフサイクルを通じて変化することも多くなってきた。技術の進展や法規制・国際標準の変更などによる要求の変化もある。要求の変化により新たな問題や障害が起こることがある。

ソフトウェア開発はプログラミング言語、開発・保守プロセス、プロジェクトマネジメント手法、ツールなどの進化により、その開発のスピードや品質など著しい進歩を遂げている。既に一般的になっている CMMI、SysML などのプロセスやツールを正しく活用しても上記の問題点は必ずしも解決していない。IPA/SEC が定期的に纏めている情報システムの障害情報 [3] 等にも見られるように、単なるソフトウェアのバグと言うよりも前述の原因による障害が数多くみられる。巨大で複雑になったシステムをマネージするためには、サービス継続及び説明責任という観点で問題を捉えライフサイクルを通じて一貫した手法や技術が必要である。以下の章では、DEOS プロセスとそれを実現する核となる技術である D-Case を紹介する。

3　DEOS プロセス

社会インフラ、企業や公共の基幹システム、組込み機器のようなシステムは膨大なソフトウェアを含み、長期間使われ、その運用中にシステムは環境変化・目的変化に対応すべく修正されていく。そのためシステムの開発、運用・保守、EOL※3 という一連の流れとしてプロセスを

【脚注】
※ 3　End Of Life : 製品やサービスの終了

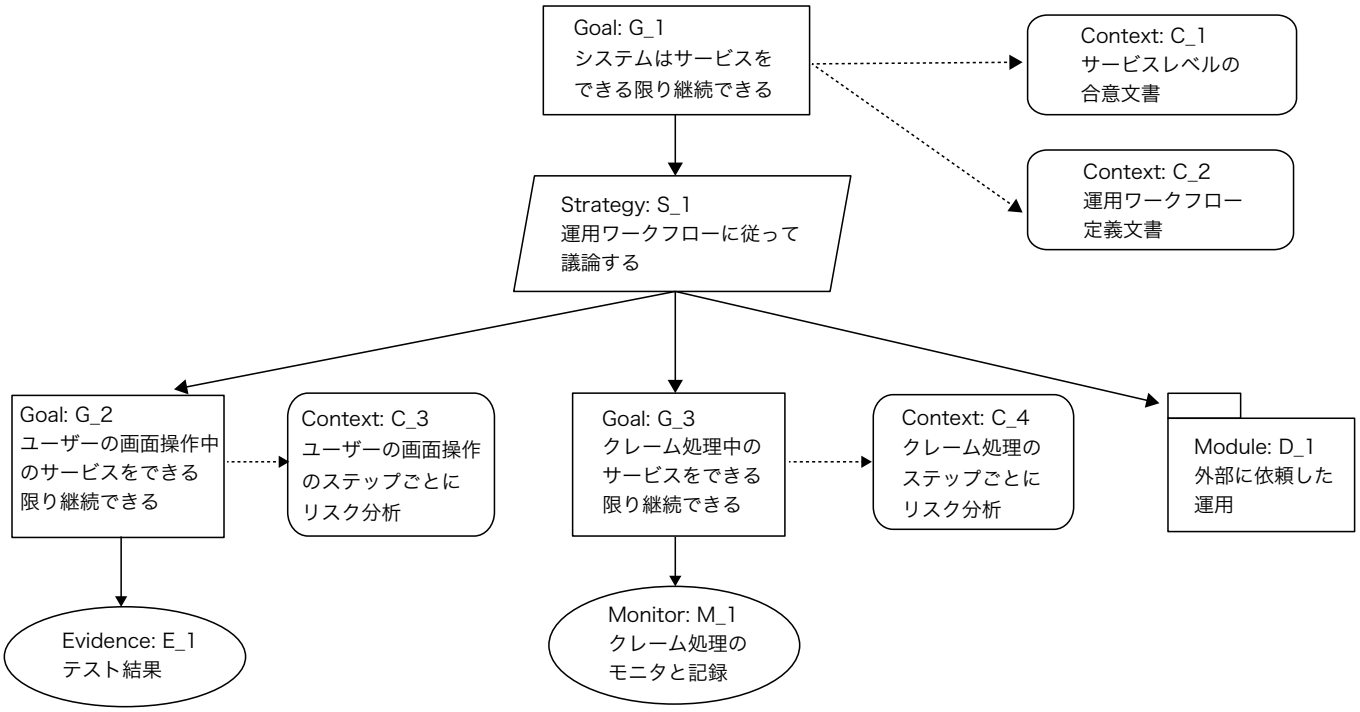

図 2　D-Case の記述例

分けて考えることは難しい。開発と運用・保守をシームレスに連携させ、システムのライフサイクルを通じて一貫したマネジメントの仕組みをもった新たなマネジメントプロセスが必要である。システムを安全・安心・快適に使い続ける、すなわちディペンダブルに開発・運用して行くためには上記で述べたように、システムによるサービスを継続し、開発・運用において実行したことに対して説明責任を果たすことが重要になる。

このような観点から開発・運用のプロセスを見直すと、システムをディペンダブルにするためには；1. 通常運用で何をすべきか、2. 障害発生時に何をすべきか、3. 変化対応で何をすべきか、を定義して実行する必要があることが分かる。多くのシステムや組織ではこれらのプロセスが存在していることが多いが、有機的にシームレスに融合・結合するための仕組みが不十分なことと、これらのプロセスが明示的になっていないことが多いことから、前述したようにシステムに障害が発生し、それによって社会的に甚大な影響を及ぼすような事態に発展することが起こっている。

我々は上記の問題を解決するために DEOS プロセスを提唱した（図 1）。DEOS プロセスは障害に対して「未然防止機能」や「障害対応機能」を含み、システムの変更を起動するための「再発防止機能」も統合した反復的プロセスである。システムは「変化対応サイクル」の中で合意形成に基づいて開発・運用され目的や環境の変化に対応する。通常運用では合意に基づいてシステムの振る舞いや環境の変化を監視する。障害や予兆が検知された場合は合意に基づきシステムの自己回復機能あるいはオペレータの動作により「障害対応サイクル」に入る。この過程で新たな合意形成が必要と判断されると再び「変化対応サイクル」に入って合意を見直し、システムを変更する。これら一連の作業は後述する D-Case に基いて指示され記録として残されると共に、説明責任を果たすためのベースとなる。DEOS プロセスは利害関係者間の合意形成、説明責任の遂行を含むシームレスなプロセスである。DEOS プロジェクトではそのプロセスを支える技術・ツールを整備した。

4 DEOS プロセスを実現する技術 — D-Case と事例 —

D-Case は利害関係者の合意形成のための手法やツールの総称である。アシュアランスケースを基に、システムをマネージする仕組みとしてモニターノード、システ

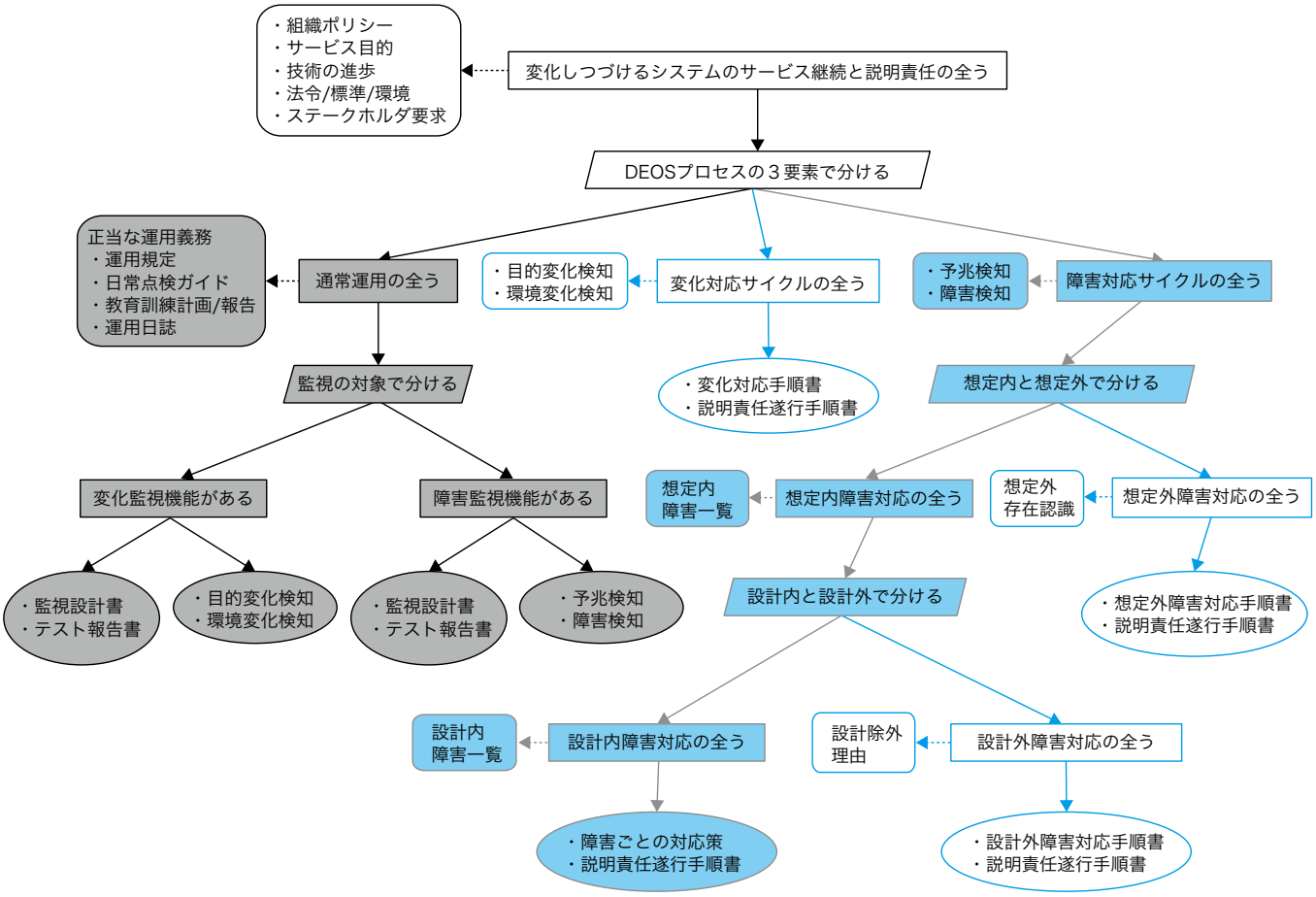

図３　DEOS プロセス基本部分の D-Case による記述

ムの動作を変更させるアクションノード、他のシステムとの連携に関する事項を記述するための外部接続ノードを追加して拡張したものである [4]。アシュアランスケースは車載システムの機能安全の国際規格 ISO 26262 でも言及されている安全ケースのもとになるもので、確信を得るための証拠が提示され構造化された議論をまとめたドキュメント群のことである。アシュアランスケースは国際規格 ISO/IEC 15026 シリーズとして制定されている。

D-Case（図 2）ではその表現に GSN※4 と呼ばれるグラフィカルな手法を採用し、まず主張したい事柄をトップゴールに掲げ、何が前提となっているのかをコンテクスト（前提）ノードに書き、どのような戦略により主張を分解していくのかを戦略ノードに記述しながら議論を構造的に詳細化していく。これによりトップゴールはサブゴールに分解され、サブゴールを保証する証拠に議論が達するまでこの分解は続けられる。D-Case では、システムの運用に関しても証拠を残さなければならないので、運用中のシステムの挙動にかかわるデータ（モニターによる測定値やログなど）も含まれる。これら運用中のシステムにかかわるデータが設定範囲を越えた場合は、通常運用から逸脱が起きていると判断し、設定範囲に収まるように障害対応を行なわなければならない。当プロジェクトではシステムにかかわるデータの収集とその障害対応は D-Script とよばれるスクリプトにより実現している。また、障害の原因究明の結果再発防止策やシステム改善が必要となり変化対応サイクルに進む場合は、再合意をおこない D-Case を更新して、システムをマネージする。このようなプロセスを確実に遂行し説明責任を果たせるようにするためには、D-Case の履歴を各種開発・運用ドキュメント群とリンクさせる仕組みが必要で

【脚注】

※４　アシュアランスケースを記述するための表記法の一つ：Goal Structuring Notation

ある。我々はこのために合意記述データベース（D-ADD）を開発した。前章で述べた DEOS プロセスは D-Case で表現して、基本的な D-Case として活用できる（図 3）。

DEOS プロセスを実現するためには、各種ドキュメント群をマネージ出来るように D-Case を記述すること、システムを柔軟に制御してディペンダビリティを達成するための記述が D-Case に明示されていることが重要となる。この中には責任者、担当者、有効期限などについても記述される。さらに、我々はシステムの監視、異常発生時のシステムの柔軟な対応を可能とする実行環境として、D-RE※5 を開発し、D-Case との連携に関する研究を進めてきた。D-Case はサービス継続のための必要事項の合意文書であるとともに、説明責任を果たす際に重要な役割を果たすことからも DEOS プロセスの核であるといえる。D-Case を記述することにより、利害関係者間の合意事項やリスクコミュニケーションが明示化され、サービス継続と説明責任の方針が明確になる。

DEOS プロセスを円滑に運用するためには、D-Case を記述し、さらに対象システムをマネージするために D-Script、D-RE、D-ADD との連携が必要となる。そのためのツールとして D-Case 記述ツールを開発し、さらに CMIS※6 をインターフェースとして開発プロセスで作成される文書を D-Case と連携させる機能、OSLC※7 を通じて SysML ツールとの連携を可能にする機能を開発し、既存のプロセスや手法との統合も図っている。

D-Case を使った実証実験の一つとして、日本科学未来館のフロア案内ロボットの開発運用に適用した。このロボットは ART-Linux※8 上に各種センサや機能を実装し、30 × 130m の展示会場をロボットが人や障害物を避けながら巡回し、来訪者との対話、デモの時刻と内容の宣伝を行うことなどを目的としている。この実証実験はロボットの機能、運用、安全、説明責任、改善の議論を D-Case として記述し、その内容をステークホルダ（サービス提供者）の日本科学未来館と合意し、一日当たり 1 万人強の来訪者が来る環境においてロボットを運用し、求められたサービスを安全に提供する事ができた [5]。

D-Case 事例や実証評価活動は D-Case 実証評価研究会ホームページに紹介されている [6]。また、DEOS センターホームページでも D-Case ツールや事例を公開している [7]。

5 技術の実用化と今後の課題

DEOS プロジェクトにおいて研究開発されたソフトウェアは DEOS センターホームページからダウンロード可能となっている。また、我々は開発してきた技術や概念が広く使われるために標準化活動をおこなっている。デジュール規格として IEC TC56 においてプロジェクトが開始され国際標準 IEC 62853（OSD 規格）の策定を進めている。また、デファクト規格としては 2013 年 7 月に The Open Group で当プロジェクトの考え方を反映した技術標準が策定された [8]。

DEOS プロジェクトにおける研究開発成果は成果集や書籍、論文、ソフトウェア等により広く利用可能な形で発表されている。これらの技術を実際に産業界で使っていくためにコンソーシアム「DEOS 協会」※9 が 2013 年 10 月に設立され、今後はコンソーシアムを中心にプロジェクトの成果が展開されていく [9]。

6 謝辞

DEOS プロジェクトは研究総括の所眞理雄氏を始め、多くの研究機関や企業から多数の方々に参加頂き成果を上げることができた。ここに謝意を表す。

【脚注】

※ 5　DEOS Runtime Environment: DEOS 実行環境
※ 6　標準化団体 OASIS の定義するコンテンツ管理システムのインターフェス標準
※ 7　Open Services for Lifecycle Collaboration : 開発ツールなどの相互連携のためのデータ交換の標準仕様
※ 8　（独）産業技術総合研究所で開発された Real-time Linux
※ 9　正式名称は「一般社団法人 ディペンダビリティ技術推進協会」

【参考文献】

[1] 所 眞理雄、他：“DEOS プロジェクト研究成果集 ”、科学技術振興機構 DEOS-FY2013-SS-01J、2013/11/15.
[2] Mario Tokoro (eds): Open Systems Dependability, CRC press, 2012.
[3] 松田 晃一、他：連載 情報システムの事故データ、SEC journal No.26, 27, 28, 30, 32, 34
[4] 松野 裕、山本 修一郎：実践 D-Case, オンデマンド出版、2012 年
[5] デジタルヒューマン工学研究センター：D-Case のロボット応用～日本科学未来館フロア移動ロボットを題材にして～ DEOS-FY2013-RA-01J
[6] D-Case 実証評価研究会 HP：http://www.dcase.jp/index.html
[7] DEOS センター HP：http://www.dependable-os.net
[8] The Open Group, Dependability through Assuredness™ Standard, 2013
[9] 一般社団法人 ディペンダビリティ技術推進協会 HP：http://deos.or.jp/index-j.html