

DEOSプロジェクト
Agreement Description Database
(D-ADD)



Technical White Paper

Revision: 1.0

2013/12/03

株式会社 Symphony

目次

1.	このホワイトペーパーの概要	4
2.	DEOS とは	5
2.1.	「DEOS プロセス」と「DEOS アーキテクチャ」	5
2.2.	D-Case を用いた合意形成	7
2.3.	DEOS プロジェクトと D-ADD の関係性	8
3.	合意記述データベース D-ADD	9
3.1.	要求をマネジメントする	9
3.2.	D-ADD の必要性	9
3.3.	ネットワーク構造をもった非定型データを一元管理する D-ADD	10
3.4.	ノンストップ運用を実現する D-ADD	11
3.5.	D-ADD からみたモデルの概念	11
3.6.	D-Case の運用を支援するさまざまなツール	11
3.7.	D-ADD の三層構造	12
3.8.	D-ADD 上で動作する基本ツールセット	13
3.8.1.	Assure It	13
3.8.2.	D-Case Weaver	13
3.8.3.	D-ADD dashboard	14
3.9.	D-ADD のアーキテクチャ	15
3.10.	ライセンスポリシー	16
4.	D-ADD の活用シナリオ	17
4.1.	エンタープライズアプリケーションの上流設計に関わる課題の解決	17
4.2.	企業の内部統制管理と業務遂行支援	19
4.3.	ユーザ企業とシステム・インテグレータの関係改善	19
5.	D-ADD の将来	22
5.1.	次世代のアプリケーションサーバとしての D-ADD	22
5.2.	エンタープライズリポジトリとしての D-ADD	24
5.3.	運用に関する知の集合体をベースとした、自律的自動運用の実現	25
5.4.	D-ADD の再考 - 構造化が社会を変える	26
6.	よくある質問と回答	28
6.1.	D-ADD はシステム運用監視ツールですか。	28
6.2.	D-Case は仕様記述言語ですか。	28

6.3.	DEOS は国際標準なのですか。	29
6.4.	DEOS プロセスはウォーターフォール開発とアジャイル開発のどちらに向いていますか。	29
6.5.	D-ADD はどこから入手できますか。	29
6.6.	D-ADD を使ってみたいのですが、D-Case の記述方法に馴れていません。	30
6.7.	D-ADD の機能拡張に当社の希望を伝えたいのですが。	30
7.	おわりに	31

1. このホワイトペーパーの概要

このホワイトペーパーでは、D-ADD を構成する各製品の概要と、製品化を支える基盤である DEOS の位置づけ、そして本製品の具体的な活用方法と将来のビジョンについて説明します。

D-ADD は一つの合意が形成されるまでのすべての根拠、そして議論の過程を時系列に記録し、整理し、任意の視点で辿ることができる次世代の業務基盤ソフトウェアです。

D-ADD の導入によってますます巨大化・複雑化するシステムへのビジネス継続性と説明責任を担保し、これまでのソフトウェアライフサイクルプロセスを新しい次元に導きます。

本ホワイトペーパーの構成は次のとおりです。

第 2 章「DEOS とは」では本製品の基盤となる DEOS について説明します。DEOS は日本の国家プロジェクトであり、ディペンダビリティの重要性を世界に対して発信しています。

第 3 章「合意記述データベース D-ADD」では、DEOS の中核技術である D-ADD の概要とソフトウェア構成を説明しています。

第 4 章「D-ADD の活用シナリオ」では、組織が抱える共通の問題に対して D-ADD がどのように解決に貢献できるかを具体的に説明します。

第 5 章「D-ADD の将来」では、D-ADD の発展が企業そして社会全体にどのように貢献するのかというビジョンを説明します。

第 6 章「よくある質問と回答」では、DEOS ならびに D-ADD についての一般的な質問と、その回答を紹介しています。

2. DEOS とは

現代のコンピュータシステムは常に変化しつづける外部の環境に対応し、未知の障害をマネージし、サービスをできる限り継続し、障害時には社会に対して説明責任を果たさなければなりません。この開放系対応力を「OSD：オープンシステムディペンダビリティ (Open Systems Dependability)」と呼びます。この OSD を実現するための知識・技術体系の総称が DEOS (Dependability Engineering for Open Systems) です。

DEOS は日本の国家プロジェクトとして誕生しました。科学技術振興機構 (JST: Japan Science and Technology Agency) が中心となって 2006 年にプロジェクトを発足。その中核となる DEOS 研究開発センターならびに研究領域内の各研究チームはこれまで多くの論文を発表する傍ら、DEOS コンセプトの国際標準化にも関わってきました。このホワイトペーパーで説明する「D-ADD」は、DEOS が目指す世界を具体化する成果物の一つです。

2.1. 「DEOS プロセス」と「DEOS アーキテクチャ」

DEOS は上述したように「変化し続けるシステムのためのディペンダビリティをいかに向上させるか」というテーマを掲げた研究プロジェクトです。ここで変化とは、サービスの目的・ユーザの要求・技術革新・法規制・標準化といったさまざまな外部環境要因の他に、不定性（想定外の事象）の問題も含まれます。このようなさまざまな変化に対するディペンダビリティには次のことがあげられます。

- ・ システムの障害要因を顕在化する前にできる限り取り除く。
- ・ 障害が顕在化した後に迅速かつ適切に対応する。
- ・ 障害が起こった時の影響を最小とするようにマネージする。
- ・ 同様の障害を繰り返さない。

さらに社会的にもこれらの説明責任を果たすとともに、このディペンダビリティを継続的に行う能力を維持し続けることが求められます。

DEOS はこの達成のために二つの概念「DEOS プロセス」と「DEOS アーキテクチャ」を生み出しました。DEOS プロセスとは変化に対応したサイクルと、障害に対応したサイクルという二つのサイクルを運用するものです。このプロセスを支える仕組みを DEOS アーキテ

クチャと呼びます。具体的には「要求マネジメントプロセスを支えるツールと合意記述データベース」「ディペンダブルなソフトウェアを開発するためのツール群」「システムの監視・記録や障害時の迅速対応を行う実行環境」から構成されます。

DEOS プロセスを支える DEOS アーキテクチャの核となるのが、合意記述データベースです。これを D-ADD (DEOS Agreement Description Database) と呼びます。D-ADD には主に二種類のデータ D-Case と D-Script がマネージされます。D-Case は DEOS プロセスとして遵守すべき目標が（関係者間の）合意という形で記述されたものです。合意には、そこに至る過程や、その前提となる論拠といったすべてのデータを含みます。一つの合意はさらに別の合意と関係付けられており、システム全体のディペンダビリティを定義づける合意は複雑なネットワーク構造になります。もう一つのデータである D-Script は、合意に含まれる前提条件をトリガーとして、その条件に従ってシステムに対してある「操作」を起動する振る舞いを記したものです。すなわち D-ADD とは、ディペンダビリティを維持するための（静的な構造である）D-Case と、（動的な振る舞いである）D-Script の両方を用いて DEOS プロセスを運用する基盤技術です。

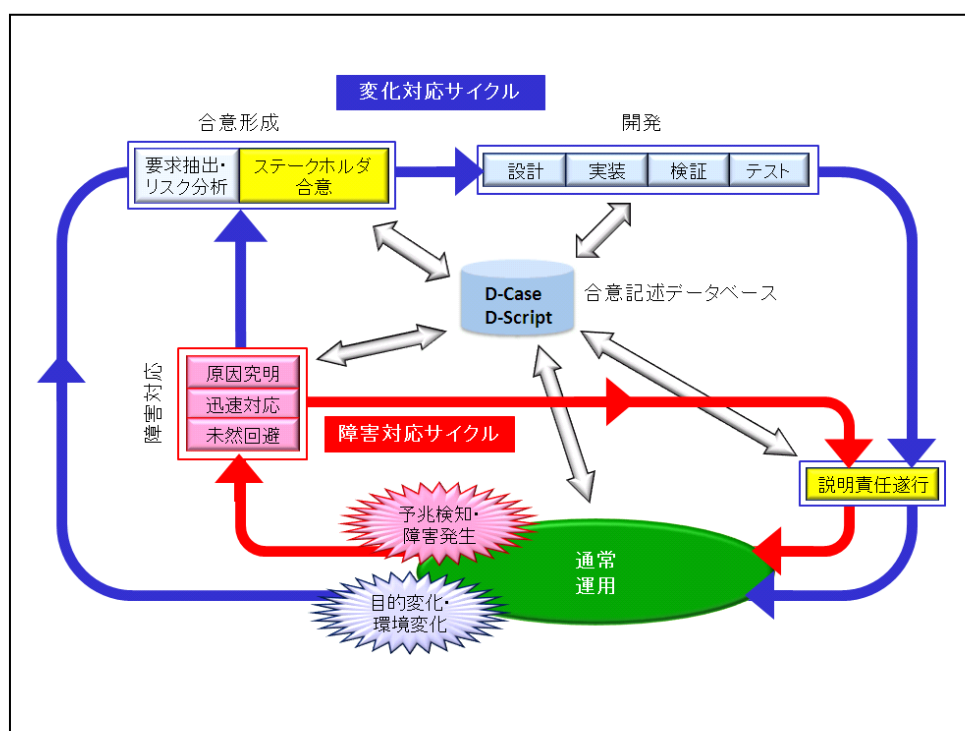


図1 DEOS プロセス

2.2. D-Case を用いた合意形成

D-Case とは「システムがディペンダブルであると関係者（利用者、開発者、社会全体）に確信を持ってもらうための議論およびドキュメント」を指します。D-Case の土台は欧米におけるシステムの安全性規格認証をとおして形成されてきた Assurance Case という概念です。これを DEOS プロセス向けに拡張したものが D-Case であり、2009 年に最初のバージョンが登場しました。D-Case は合意形成の議論のための方法やツール、そして表記法を含んでおり、これらの情報はすべて D-ADD 内で管理され、開発・運用を通じて一貫して活用されます。

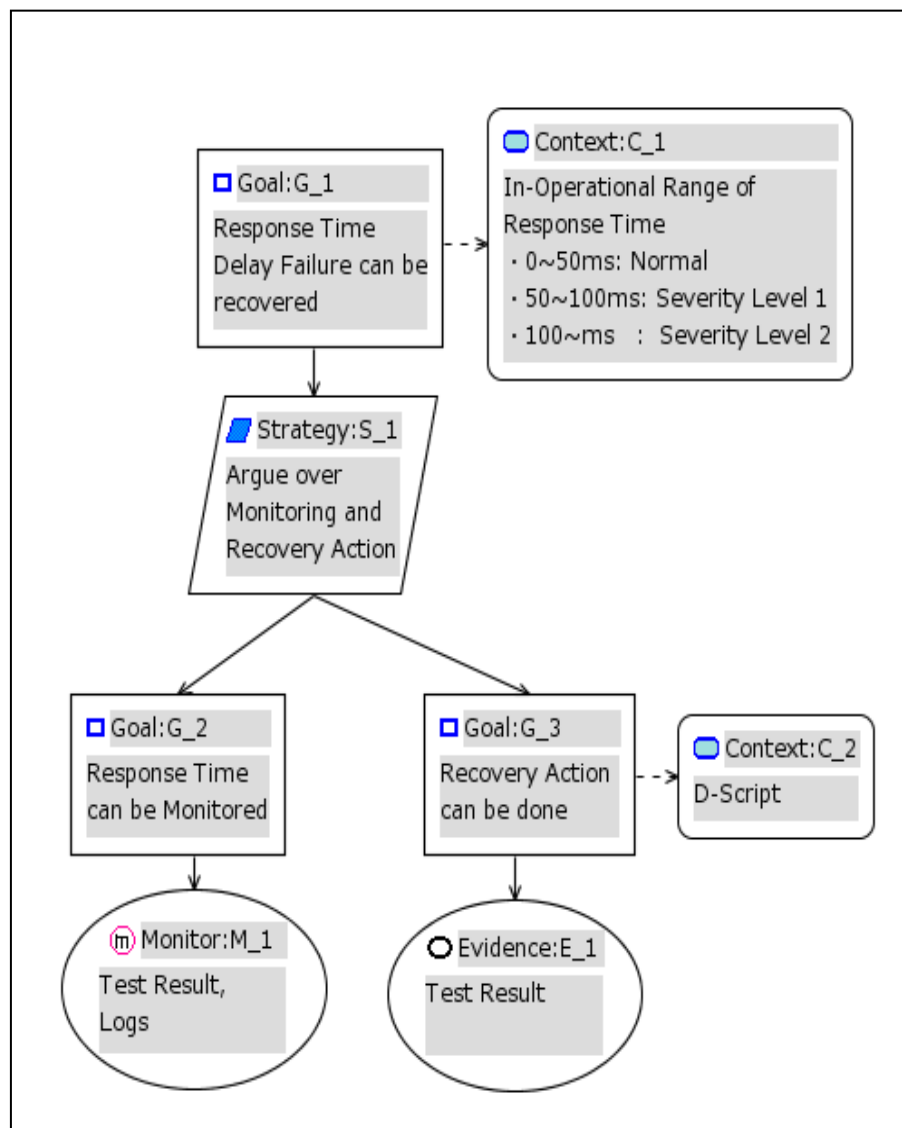


図 2 D-Case の構造

D-Case で記述される情報は主として Goal, Strategy, Context, Evidence といったものです。D-Case ではこれらの情報を GSN (Goal Structuring Notation) を用いて構造化します。

よって D-Case は最終目標である Goal を起点ノードとした木構造になります。

なお GSN は「議論」をモデル化、可視化するもので、トゥルーミンロジックという議論の基礎理論から誕生しています。D-Case の表記スタイルは GSN の応用例の一つですが、他にも OMG(The Object Management Group) による SACM (Structured Assurance Case Metamodel) や ARM (Argument Metamodel) の標準化のベースにも使われています。GNS 自身は議論の論理構造を示す枠組みですが、システム的な視点からは「ここまで議論した」という過程を保持することにより、問題が発生したときに「ここは想定していなかった」という検証に役立てることができます。

2.3. DEOS プロジェクトと D-ADD の関係性

本ホワイトペーパーで説明する D-ADD は、国家プロジェクトである DEOS の中核となる成果物の一つです。なお、DEOS プロジェクト全体では D-ADD (D-Case と D-Script を含む) 以外にも、さまざまなプログラム開発ツール、DEOS ランタイム環境 (D-RE; DEOS Runtime Environment) に関する研究成果を出しており、これらの情報は「DEOS 研究開発センター」¹ から発信されています。

なお DEOS 研究開発センターの成果は 2013 年 10 月に発足した「一般社団法人 ディペンダビリティ技術推進協会 (DEOS 協会)」² が引き継いでいます。D-ADD のオープンソース実装を含む、DEOS プロジェクトのさまざまな研究成果は、DEOS 協会のホームページから入手できます。

¹ <http://www.dependable-os.net/osddeos/index-j.html>

² <http://deos.or.jp/index-j.html>

3. 合意記述データベース D-ADD

本章では DEOS プロジェクトの中核を担う合意記述データベース D-ADD について詳しく説明します。D-ADD は DEOS プロジェクトの成果物の一つであり、DEOS プロセスを支える技術基盤です。ただし D-ADD を活用することは、必ずしもシステムを DEOS プロセス化することではありません。既存システムに D-ADD を組み込みながら、徐々にプロジェクト体制をステップアップし、最終的には DEOS プロセスでの運用を目指すという理念を目指すことができるようになっていきます。その最初の一步が D-ADD の導入です。

3.1. 要求をマネジメントする

企業は、顧客に対して何らかの価値を提供します。この提供をサービスと呼ぶこともあります。サービスの品質や処理能力（量）は時代とともに変化します。昨今はグローバル化の進展により、サービスを支える資材や工場、人的資源、外部との取引先関係は何層にもまたがっています。わずかな変化がサービスの安定性に影響を及ぼすことになり、ディペンダビリティの担保はますます困難になっています。

3.2. D-ADD の必要性

D-Case は表記法ですので、ペンとノートがあればすぐに記述することができます。しかしそのようにノートに書き出した D-Case でディペンダビリティを保つのは実際には困難です。これは次の理由によります。

- ・ 通常、システムの安定稼働のために検討すべき合意事項は多岐におよびます。さらに大規模システムにおいて D-Case は相互に作用しあうネットワーク的な構造を持つことになります。障害発生時に、ある D-Case が他の D-Case に与える影響を人間が洗い出すことは現実的ではありません。さらに障害発生時は切迫した状況で判断を求められるため、担当者のパニックを誘発する恐れがあります。

- ・ D-Case そのものが常に改訂される性質をもっています。改訂記録はすべて保持される必要があり、最新の D-Case だけ持っていればよいということではありません。なぜ、こういう改訂を行ったのかという議論も記録することが重要なためです。頻繁な改訂作業が生じる環境では、ノートに書くだけでは管理不可能なボリュームになります。

- ・ D-Case は合意形成の過程を記録しますが、その過程には「ある合意に対する反論」と、「その反論をどのように納得させたか」という証憑が含まれます。仮に、合意し採用された主張に誤りがあり、一方の主張（その時点では反論）が正しかったということになった場合、その反論に関連する他のすべての D-Case を探索することが求められます。これを人間が適切に行うことは難しいことです。

- ・ D-Case には「こういう状況になったときに、システムに対して行う振る舞い」を与えることが求められます。これが D-Script です。ノートに振る舞いを記述することはできません。例えばシステムに対する操作コマンド手順を記録します。しかし人間による操作はそれ自体がリスク要因であり、操作を間違えた場合を想定した D-Case を用意することは際限がありません。このような操作コマンドの実行は自動運用が適切であり、それゆえに自動運用を支える基盤が求められます。

このような理由から、D-Case（および D-Script）を管理する基盤として D-ADD が開発されました。それでは D-ADD とは具体的にどのような機能を有するのか。以降より、D-ADD の詳細を説明します。

3.3. ネットワーク構造をもった非定型データを一元管理する D-ADD

D-ADD が格納するデータである D-Case と D-Script はいずれも非定型データです。特に D-Case は GSN を基盤としているためそれ自身は木構造であり、かつ D-Case 同士の関係性はネットワーク構造になります。さらに D-Case に対する改訂作業は、過去のデータを消去して上書きするといったトランザクションモデルではなく、時間軸とともにすべての改訂作業が記録されていく時系列モデルであるため、それ自身もまたネットワーク構造です。

つまり D-ADD とは木構造とネットワーク構造の複合体を管理する、まったく新しいデータベースです。これは従来のエンタープライズアプリケーションで活用されてきたデータ整合性を保証するためのトランザクション制御を重視してきたリレーショナルデータベースとは異なるアーキテクチャになります。具体的には “NoSQL” と呼ばれるデータベースアーキテクチャを基盤としますが、「Key/Value Store」「Graph DB」「Document DB」のそれぞれの特徴をもった (D-Case と D-Script のための) 「Repository」として機能します。

少し脱線しますが、D-ADD の能力は GSN そのもののリポジトリデータベースと解釈する

こともできます。GSN を基盤とした表記であれば D-Case に限らず D-ADD を活用することができます。

3.4. ノンストップ運用を実現する D-ADD

D-ADD が管理するもう一つのデータである D-Script に着目してみましょう。D-Script はシステムに対する振る舞いを定義できますが、これにはシステムに対する命令セット（コマンド）を含みます。一種のプログラムコードです。

D-ADD はこの D-Script を保存するだけでなく、D-Case に記述された前提条件（発火条件）によって、予め定められた D-Script を呼び出して実行するための基盤としても動作します。このため D-ADD は D-Script の変更の際しても D-ADD 自身を再起動させることはありません。変更は時系列で記録され、条件によって変更後の D-Script が実行されます。もちろん D-Case と同じく過去の D-Script を保持しているため、変更履歴を辿って戻すこともできます。

3.5. D-ADD からみたモデルの概念

D-Case は「合意」に関する表記ですが、複数の D-Case が関連して「合意グラフ」を形成することもあります。さらに、D-Case を木構造としてみたとき、あるノードを説明するための「ドキュメント」や「ログデータ」などが含まれることもあります。これは証憑という性質を含みます。D-Case では、このようなデータを「モデル」という単位で取り扱い、すべてを一元的に管理することができます。

3.6. D-Case の運用を支援するさまざまなツール

D-ADD そのものは外部からの操作を受け付けるデータベースという位置づけですので、D-ADD が管理する情報を取得して画面に表示（可視化）するためのさまざまなツールが必要です。これを「基本ツール」と呼びます。基本ツール層には、D-Case の Visualization や D-Script のための Monitoring 機構を含みます。この詳細は後述します。

3.7. D-ADD の三層構造

これまで説明したように、D-ADD は D-Case と D-Script を統合管理する「リポジトリ層」と、D-Case を代表とするすべてのデータをモデルとして統一的に扱うための「モデル層」があります。さらにこの上位に D-ADD のためのユーザーインターフェースを提供する「基本ツール層」があります。これを図3に示します。

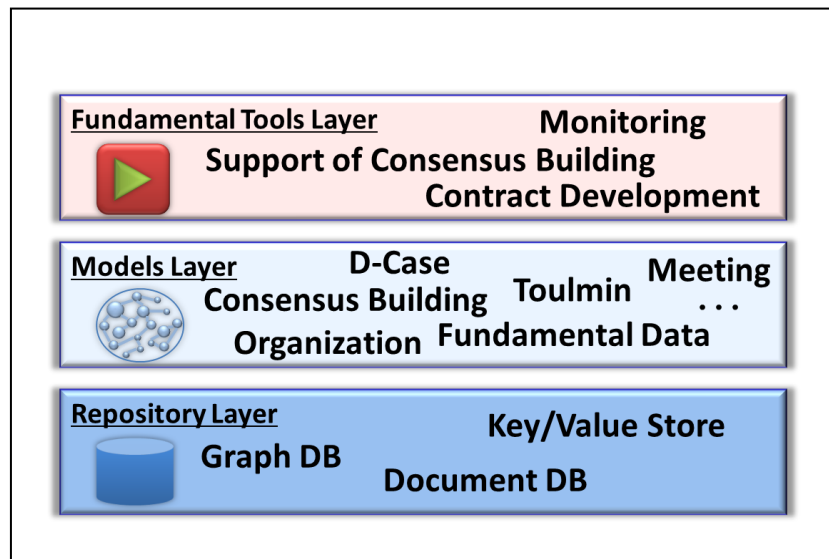


図3 D-ADD の三層構造

3.8. D-ADD 上で動作する基本ツールセット

D-ADD にはさまざまな基本ツールセットが同梱されています。ここでは、各ツールの概要を説明します。

3.8.1. Assure It

Assure It は D-Case と D-Script のための Visual Editor です。Web ブラウザで動作し、グラフィカルに木構造を表示するとともに、ノードの追加や削除、変更を画面上で行うことができます。修正した新しい D-Case ならびに D-Script はシームレスに D-ADD に統合管理されます。Assure It はまた、D-Script を直接、実行することもできます。

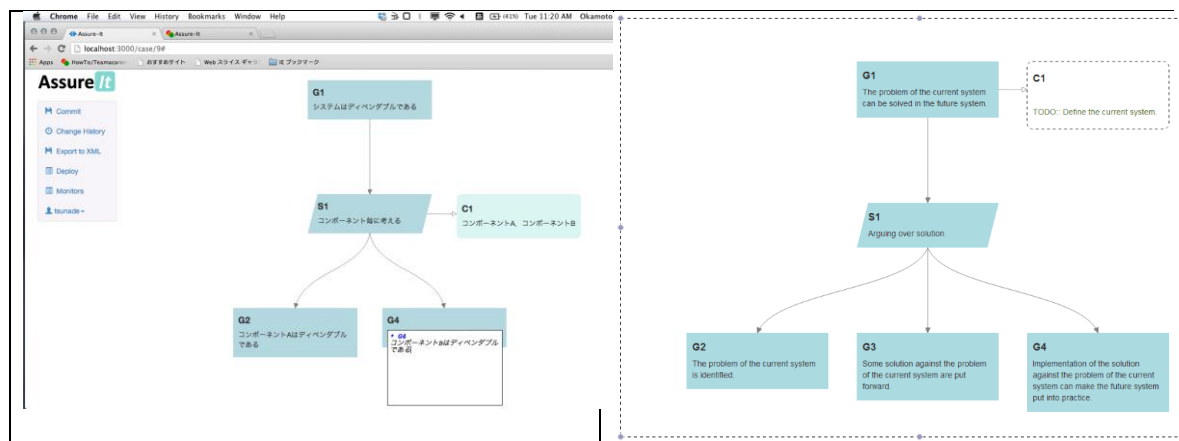


図 4 Assure-It による D-Case の編集

3.8.2. D-Case Weaver

D-Case Weaverは複数の担当者によるD-Caseの共同作業を提供するためのツールです。D-Case WeaverもまたWebブラウザで動作します。ある担当者がD-Caseに編集を加えると、他の担当者がその情報を共有できるようになります。

D-Caseは合意に至るまでにさまざまな主張や反論が重ねられ、適切な証憑が加わっていきます。例えばIT開発の現場を想定すると、要件定義書や基本設計書、さらにはサービス定義書などのシステムに関するドキュメントをどのように管理していくかという視点で捉えることができます。

通常、ドキュメント同士の相関や齟齬を見つける際には、ドキュメントにマークアップ（たとえば赤入れ、フェルペンによるアンダーライン、コメント書き）を行い、担当するメンバー間で議論やマークアップの修正を繰り返すことになるでしょう。

D-Case WeaverはAssure Itと異なり、ドキュメントを構造的に分解する機能を提供しています。さらにそれらをマウスにより選択し、その担当者がどうしてそこに着目したかをコメントとして書き残すといった一連の作業を支援します。D-Case Weaverを通して編集した内容（D-Case）もまた、リアルタイムにD-ADD内部のリポジトリに反映されます。

3.8.3. D-ADD dashboard

ある D-Case に関連する他の D-Case を探し出し、必要な情報を選択して画面に表示するツールです。例えば障害発生時などで、関連性の高い他の D-Case を検索し、表示するといったユーザーインタフェースを提供します。これは説明責任を実行するためのツールとして用いることができるように設計されました。

D-ADD dashboard はコンソールからのコマンド操作に加え、オープンソースのプロジェクト管理ソフトウェアとして著名な Redmine³のプラグインとしても利用できるようになっています。将来の D-ADD dashboard では、用語辞書の管理機能や、組織の権限情報を加味した複雑な検索設定を行えるようにします。まだ、D-Case と組織情報との密連携を行うことで、この D-Case が影響する組織を表示するといった、より利便性の高い機能へと拡張する計画です。

³ <http://redmine.jp/>

3.9. D-ADD のアーキテクチャ

D-ADD は Java Virtual Machine 上で動作するように設計されました。D-ADD のアーキテクチャを図 5 に示します。

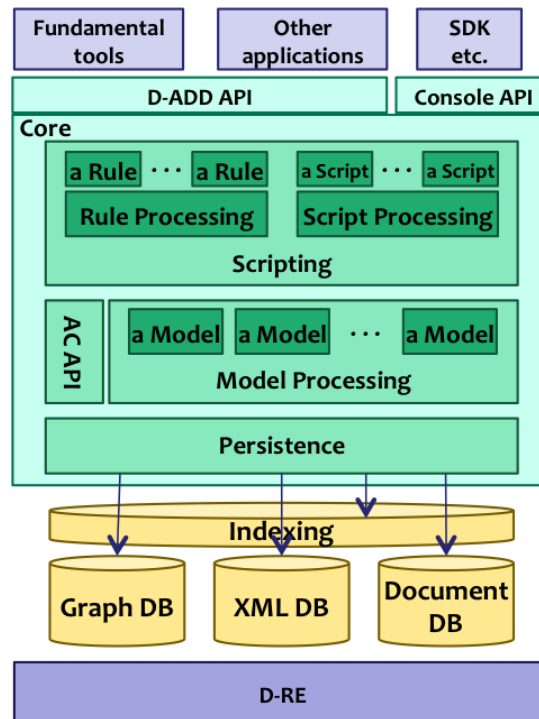


図 5 D-ADD のアーキテクチャ

D-ADD への操作は、Java ベースの API を経由して行います。D-ADD 内部では D-Case ならびに D-Script をそれぞれ管理しており、その永続化を行うリポジトリ層は選択できるようになっています。現在はリポジトリ層の実装として BaseX⁴と Neo4j⁵を用いています。

すべての格納データは相互に関連しており、また合意形成の過程も保持するため時系列での管理も行われます。証拠という形でさまざまな添付ファイルも紐づいており、巨大な「合意グラフ」として管理されるようになっています。

⁴ <http://basex.org/>

⁵ <http://www.neo4j.org/>

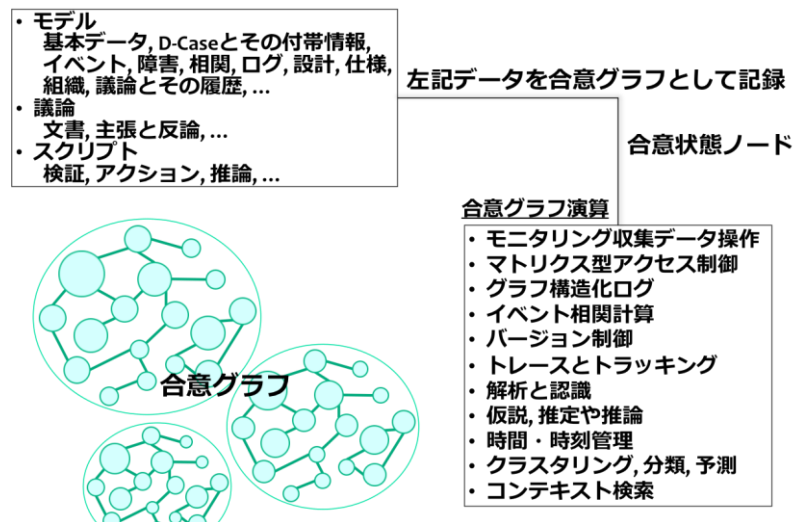


図 6 合意グラフのイメージ

D-ADD が管理する D-Script は実行可能なスクリプトです。D-Script の変更により、D-ADD を再起動することなく（変更された）D-Script が実行されます。すなわち D-ADD からみて D-Script は pluggable な実行モジュールになっています。

D-ADD による D-Case ならびに D-Script の永続化処理のタイミングで、適切な検索用のインデックスが更新されます。このため大量のデータに対する関係性による検索、キーワードによる検索、時系列の検索が高速に実行できるようになっています。

3.10. ライセンスポリシー

D-ADD は OSS 版とエンタープライズ版の二つのライセンスを用意しています。OSS 版はリポジトリ層に Neo4j を用いています。これは小規模な D-Case の管理に向いており、D-ADD を試したいという評価者向けに提供されています。企業での本格的な利用には、エンタープライズ版をご利用ください。エンタープライズ版ではより大規模な D-Case を扱うことができます。

4. D-ADD の活用シナリオ

これまで説明した D-ADD の利活用は遠い将来の話ではありません。ここでは大企業が共通して抱えている課題を通して、今すぐに D-ADD を導入することで得られる効果を明らかにします。

4.1. エンタープライズアプリケーションの上流設計に関わる課題の解決

1950 年代にはじめてコンピュータが企業に導入されたとき、日々の業務を電子化することで効率化を図るという、自明の目的が存在しました。それから 60 年が経過し、ハードウェアが汎用機から PC へ、またネットワーク構成が集中型から分散型へと変わったものの、業務の電子化という目標は今や多くの企業で達成されつつあります。

そして現在、エンタープライズアプリケーションに求められているのはこれまでと異なる目的です。企業が直接、インターネット経由で顧客（対企業、対個人）と向き合い、直接的なサービスを行えるようになったため、企業の存在理由となるビジネスモデルそのものを電子化することが求められるようになりました。これは「何をつくるか」ということから練り上げる必要があり、かつ市場の要求と変化にあわせて迅速に進めることが成功の鍵となっています。これは「超上流」や「源流」という概念で知られており、与えられた課題をどう実現するかといったロジカルな発想を軸とした従来のシステムエンジニアの目線が、何をつくるかといったビジネスモデルの設計に変わっています。ロジカルではなくクリエイティブな視点が重視され、かつ経営センスというものがシステムエンジニアに問われる時代になっています。

クリエイティブな思考を支えるためには、ブレインストーミングに代表される自由な発想と、経営資源という制約を意識した議論が必要です。しかしこれまでの上流設計支援とは議事録の電子化や、または SNS (Social Networking Service) として知られる共有型掲示板のようなものでした。前者は結果しか残さないため議論の過程が不明瞭です。後者は一方向の時間軸で情報が流れており、議論の発散化により全体を把握しづらいという問題がありました。この分野を支えるための取り組みはまだ始まったばかりといえます。

D-ADD が実現する「合意形成の統合管理」は、まさにこの超上流・源流の設計を支えるツールとして活用できます。設計者はコンピュータプログラミングや、オブジェクト指向

といった技術に基づいた特殊なダイアグラム (UML) を学ぶ必要はありません。議論を可視化する D-Case の表記は簡潔で、かつ日常の文章で記述することができます。D-ADD により複数メンバーによる同時設計作業を行うことができ、かつ遠隔地を含む分散拠点での議事進行も可能です。不参加の場合や、新たにチームに加わったメンバーも、視覚化された合意形成の流れを追うことで理解できるようになります。合意の論拠となった各種資料もリンクされているため、全体を一つの知識として体系的に把握することができます。

例えば航空機や船舶、宇宙ステーションといった巨大な建造物の設計支援に、D-ADD 環境がどれほど役に立ちそうかを想像することは難しくありません。対象がそこまで複雑でなくとも、チームによるすべての創造的な活動を支援するために D-ADD は効果を発揮します。

なお DEOS プロセスでは、新規システム開発に際して求められる要求をディペンダビリティという視点で明らかにし、D-Case を記述することを “フォワードサービス” と呼びます。一方で既存システムや、すでに利用している外部サービスに対するディペンダビリティを実現するために D-Case を記述することを “リバースサービス” と呼びます。いずれも要求をマネジメントするために関係者が共通で使える思考の「ものさし」として D-Case を用います。D-ADD はフォワード、リバースいずれの場合にも適用可能です。

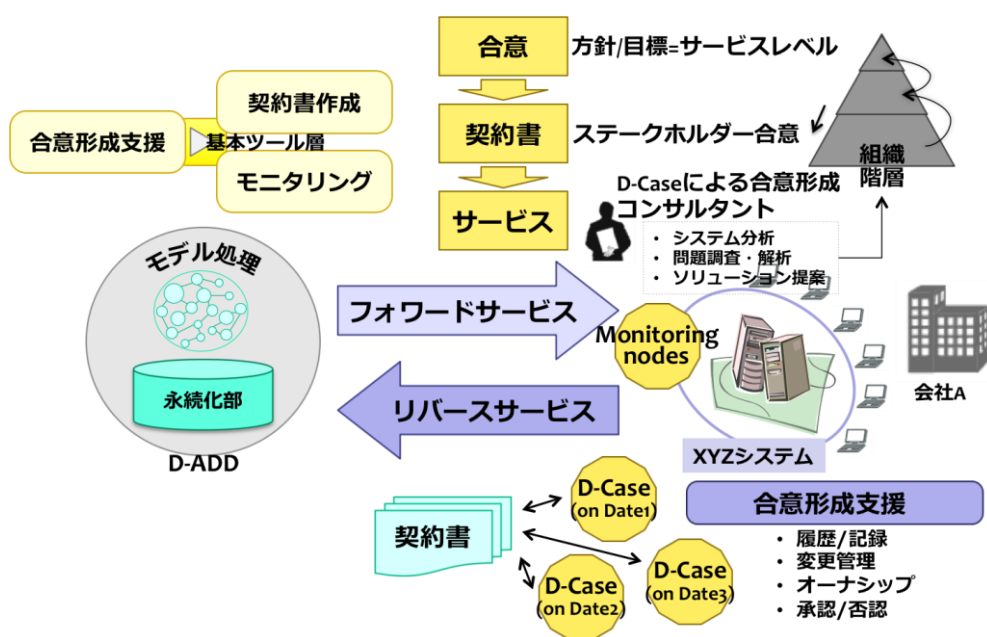


図 7 D-ADD を活用したエンタープライズアプリケーション開発イメージ

4.2. 企業の内部統制管理と業務遂行支援

巨大な組織において、個々のメンバーが与えられた権限の範囲内で、定められた業務分掌に従い、日々の業務を遂行することを監督することは実際には簡単ではありません。業務が高度に専門性を帯び、かつ細分化された組織においては、隣の同僚が行っている業務を把握することも難しいのです。すべてのマネージャが適切な監督を行っていることを保証する基盤が不在であるため、週報や月報などの報告書をとおした管理を行うしかなく、それは見落としの温床になります。さらに問題が適切に上長に報告されるまでにはタイムラグがあります。中間管理職独自の判断で問題が報告されなかった場合、あとで経営層が謝罪会見を行うことになるといった危険もあります。

当社の業務遂行にはどのようなリスクがあり、どうやって運用されているのかを見える化すること。一部の先進的企業はこの重要性に気づき、GSN による表記に注目しています。D-Case は GSN をベースにした表記方法として国際標準化のプロセスに沿っており、企業が安心して採用できるよう設計されています。さらに、D-Case と組織構成をリンクさせ、権限設定と照らし合わせることで気付かなかった業務分掌の適正性も把握できるようになります。ルーチンワークを逸脱した異常時の対応ルールを D-Script で記述することで組織の危機管理意識を高め、いざという場合に迅速に対応できる素地をつくることもできます。そして、大量の D-Case の統合管理と、D-Script の実行基盤として D-ADD が活用できます。D-ADD はこのような環境を提供する、世界でも類をみない先進性を実現しています。

4.3. ユーザ企業とシステム・インテグレータの関係改善

1990 年代にはじまった「オープンシステム」という潮流の中で、多くの開発会社はシステム・インテグレータという形でユーザ企業と関係を構築するようになりました。そのようになった理由の一つは、オープン化によってハードウェア、OS、ミドルウェアの組み合わせが爆発的に増加したため、個々の製品を組み合わせで動作させる責任をどこかが負う必要が生じたためです。もう一つは、オープン化を支える要素技術が多岐に渡るため、ユーザ企業がすべての技術を把握することが困難となり、分野毎の専門化集団が求められたためです。さらにインターネットの普及により、これらの部品はインターネット上のサービスとして再構築され、かつ世界中のサービスを選択できるようになりました。ますます高度化・複雑化するオープンシステムの選択肢に対して、システム・インテグレータがユーザ企業向けの最適な選択肢を提示し、システム開発を成功に導くことは年々、難しくな

っています。

このような流れとは別に、我が国では 1990 年代前半のバブル崩壊をきっかけとした不況が長期に渡って続いたことで、ユーザ企業の関心はシステム開発費の削減に向かいました。システム・インテグレータは、より高度な対応を、より安価に行うことが求められており、その対応策として人件費の安い海外でのプログラム開発（オフショア開発）やプロジェクト管理の厳格化など、さまざまなアプローチを試してきました。しかしその効果は限定的と言わざるをえない状態であり、近年は大型プロジェクトの破綻と訴訟の発生というニュースがますますクローズアップされています。

今やユーザ企業は、どのシステム・インテグレータと契約するかという選定が重要であるにも関わらず、その方法は確立されていません。仮にさまざまな条件をクリアできた会社と契約を締結したとしても、そのあとの要件定義の打ち合わせで少しずつ双方の「ずれ」を認識する事態が生じます。そのずれを修正できないまま開発作業が進んでいくと、最終的には使えないシステムを現場に押し付けられたという被害者意識が植え付けられ、その後の保守や運用にも支障をきたしてしまいます。とはいえシステム・インテグレータ側も当初契約した内容から時間とともにユーザの要件が変化するという前提では見積作成が困難であり、できるだけ変更要求を抑えたいという力学が働きます。結果として両者はパートナーから敵対関係へと変質し、いずれの立場の関係者の満足度が低くなるに伴い、失敗プロジェクトという烙印を押されるに至ります。この問題を改善するためにアジャイル型開発をはじめとするさまざまなアプローチが提案されていますが、いずれも小規模開発案件で成果を挙げているものの、大規模基幹系での成功事例は多くありません。

この困難な課題に対して、DEOS プロセスが有効であることを説明しましょう。ユーザ企業とシステム・インテグレータの間に、DEOS プロセスを理解したコンサルタントを登場させます。そのコンサルタントは要件定義から開発、そして運用保守に至る全行程を DEOS 的視点に立ったオペレーショナルマネジメントとして再定義することができます。

ユーザ企業が当初掲げた要件と、それが時間の経過とともにどのような理由で変遷したかについては、ビジネス環境の外的変化をはじめとして、さまざまな要因があります。これを D-Case で記述することで要件定義が変わった理由を関係者全員が共有できる場を提供します。この場が、D-ADD になります。

すべての関係者が D-ADD を通して双方の意見を出し合う中で、時間軸に沿った着地点が見えやすくします。DEOS コンサルタントは最も追加コスト負担が少なくシステム価値を最大化する方法を、膨大な D-Case を元に見出し出すことができます。もちろん DEOS コンサルタントならびに D-ADD の導入コストはプロジェクトの経費として追加されますが、トラブルの芽を早期に防止することを考えると、大規模プロジェクトほど DEOS プロセスの導入は効果が高いといえるでしょう。

ここからわかることは、DEOS プロセスならびにそれを支える技術基盤である D-ADD は、目に見えないものをつくるという、これまで人類が経験したことのないソフトウェア開発というものが本質的に抱える困難さを可視化し、人（関係者）と技術を上手に結びつけていくための有効な手法になるということです。システム開発はウォーターフォールがよいのかアジャイルがよいのかという議論ではなく、変化することを前提に、関係者間の合意形成を確実に行うという開発プロセスが重要であるという提言でもあります。

このような開発プロセスの実践を掲げる DEOS コンサルタントや、システム・インテグレータは市場において有効な差別化戦略となりえます。ユーザ企業によるシステム・インテグレータ選定も価格勝負ではなく、このようなプロセスをしっかりと行えることを保証するプロフェッショナルなチームを選択するようになるでしょう。

5. D-ADD の将来

これからの社会における IT は、私たちの生活を支えるインフラストラクチャとして、よりいっそうディペンダビリティを重要視するようになります。DEOS の視点に基づいて設計された D-ADD にはさまざまな可能性があります。この章では D-ADD の将来像について説明していきます。

5.1. 次世代のアプリケーションサーバとしての D-ADD

現在、エンタープライズアプリケーションのプラットフォームとして最も使われているのは「Java アプリケーションサーバ」です。この仕様は JCP (Java Community Process) ⁶ によって規格化されており、このホワイトペーパー執筆時点 (2013 年 12 月) での最新仕様は Java EE 7 (Java Enterprise Edition 7) ⁷ となっています。この仕様は業務データの保存と活用のための巨大な API セットであり、次のようなテーマを含みます。

- ・ RDB (Relational Database) に格納されたデータと、Java オブジェクトのマッピングを行うための標準方式 (ORM, Object Relational Mapping)
- ・ 複数テーブルにまたがったデータ更新の整合性を保証するトランザクション管理方式
- ・ 機能毎に役割を与えられた Java オブジェクトの関係性を定義づける方式 (DI: Dependency Injection)
- ・ ユーザーインタフェース部とアプリケーションサーバ部をつなげるためのプログラミングモデル
- ・ 外部システム連携の標準 API の規定
- ・ バッチ処理を行うプログラムの標準方式の規定

これらの仕様はエンタープライズ分野で求められてきた、財務会計および管理会計を中心とする伝票型データの管理方式には有効です。売上や在庫数量といったデータは上書き更新され、集計表となってまとめられます。会計年度で締められたデータは確定され、企業が存続する間、蓄積されていきます。昨今はビッグデータというキーワードで知られている、現場で発生した加工前のデータを大量に収集し分析するという仕組みが注目されて

⁶ <http://www.jcp.org/en/home/index>

⁷ <http://www.oracle.com/technetwork/jp/java/javaee/overview/index.html>

いますが、これも既存アーキテクチャの一部として組み込もうという議論が進んでいます。

しかし現在のアプリケーションサーバが提供する機能は、DEOS プロジェクトが掲げる変化し続けるシステムのためのディペンダビリティを保つという視点には必ずしも対応できていません。DEOS がその研究で明らかにしたことは、外的要因の変化に対して組織が自律的かつ柔軟に対応するためには、組織を構成するホワイトカラー層の業務遂行を支援するための、新しい枠組みが必要だということです。具体的には業務の適切な遂行を目的としたシステム化にとどまらず、なぜそういうルールになったのかという合意形成の過程および変更プロセスからシステム化し、運用とシームレスに結合することで未知の障害への即応的な対応を図りやすくするということです。

私たちのアイデアでは、D-ADD は現在のアプリケーションサーバの機能を包含する、次世代のアプリケーションサーバとなりえます。



図 8 DEOS アプリケーションサーバの位置づけ

図 8 から明らかなように、DEOS アプリケーションサーバ(D-ADD の進化系)はこれまでのアプリケーションサーバを置き換えるものではありません。Java Enterprise Edition の

仕様は 7, 8, 9, ... と改訂されていくことでしょう。これを包含しつつ、合意記述データベースの内容と、開発されるエンタープライズアプリケーションを結合するようなアーキテクチャになります。これにより、開発されたエンタープライズアプリケーションの運用において、D-Case で記述された合意（これはアプリケーションにおける業務要件に相当します）を満たしていることを常にチェックすることができ、異常時には事前に考慮されたしるべき対応ルール（D-Script）で業務を停止させることなく正常処理への復旧を自動で行えるようになります。

このシナリオは、いわゆるハードウェア障害やネットワーク障害からの自動復旧というテーマとは異なるものであることがわかるでしょう。DEOS アーキテクチャにおいては個々の機器で閉じた障害と復旧にとどまらず、業務ルールを加味したシステム全体で一貫したポリシーの下で運用を行います。これは重大な障害で機器の切り離しが発生したとき、人間系への連絡や現場への動員・指示も含めてシステム全体の運用を継続するといった、より高い視点での安定性を実現するものです。DEOS アーキテクチャの革新性は、アナログ的な運用までも包含した、システム全体の安定運用の記述・可視化・運用自動化を行うことにあります。すなわち DEOS の研究成果は、次世代のアプリケーションサーバはどのような能力をもつべきかという指針を提示しています。

5.2. エンタープライズリポジトリとしての D-ADD

すでに明らかにしたように、D-ADD はエンタープライズアプリケーションの仕様策定から開発、運用というサイクルすべてを包含することに加え、アプリケーションの変更要求に対して影響度合いを把握しながら最小限の工数で適切な改修を行うための「変化対応サイクル」を支援します。さらに障害発生時には D-Case 記述をもとに適切な復旧作業を支援する「障害対応サイクル」も含みます。このことから D-ADD は次世代のアプリケーションサーバの可能性を示唆していますが、さらにエンタープライズリポジトリ - 企業活動を支えるリソース - の管理という視点でも活用できることを示します。

エンタープライズリポジトリとは、システム（を支えるエンタープライズアプリケーション）の仕様を表す概念です。これは一般に、業務の流れであるプロセス定義、条件を整理したルール定義、そしてデータの構造を示すモデル定義の 3 種類があります。これに加えて D-ADD は「組織」と「権限」という情報を管理する方針を掲げています。すべてのプロセス、ルール、モデルは組織および権限と直行した概念で統合することができ、実際の

運用時にはこれらを加味した適切な処理を行うことができるようにします。このアーキテクチャにより、昨今の企業活動で問題視されている不正・不法な行為を防止できるのみならず、すべての業務履歴から不適切な処理を洗い出すということもアプリケーションの基本要件として提供できるようになります。これはコンプライアンス遵守を求められている大企業にとって必要不可欠な機能であり、これからのエンタープライズリポジトリの必須要件になると考えています。さらに D-ADD では、組織と権限のマトリックス情報と日々の業務履歴（これは一種のビッグデータに相当するでしょう）から、組織と権限をこう見直す方がよいというアドバイスまで行えるようになることまでを視野に入れています。DEOS プロセスとは、このような発展性までを念頭において議論されています。

5.3. 運用に関する知の集合体をベースとした、自律的自動運用の実現

GSN を基盤とした D-Case の記述が普及することで、大規模なエンタープライズアプリケーションの運用に関するノウハウスキルが蓄積されます。具体的には、こういう障害に対してはこのように対応することといったケーススタディであったり、こういう状況になった場合は注意喚起が必要といった予防ケースであったりします。D-Case そのものは自由文章で表現できますが、記述量の増大によって記述の形式化が進み、D-Case の記述に関する社会全体の知の共有がより一層、容易になるでしょう。これは Wikipedia⁸に代表される知識の共有化と同様、D-Case による巨大な知識ネットワークの実現を意味します。

社会に蓄積された膨大な D-Case の運用経験則を分析することで、安定運用のためのパターンが発見されることが期待できます。いくつかの有用なパターンはベストプラクティスとして規範になります。ベストプラクティスと自らの組織を見比べることで、組織の問題点が自明になるといった組織改革のツールとして使われるようになるかも知れません。最適な組織構成や人員配置問題の解決に、個々のコンサルタントの経験ではなく、D-Case のベストプラクティスが活かされるということです。さらに D-Case は常に変化するため、時代に則した組織改革を行いやすくなることでしょう。

D-Case の記述形式がルール化されることで、最終的には D-Case 自身が「より強固な運用ルールとなるべく」インターネットから他の D-Case を発見し、自らのシステムに組み込むことで自律的にシステムの自動運用を実現できるようになる可能性があります。ちょ

⁸ <http://www.wikipedia.org/>

うど小さな分子が少しずつ手を取りあって大きな分子となり、最終的には「安定運用をゴールとした」一個の生命体となるように、D-Case 同士が有機的に相互作用するというイメージです。このとき D-ADD は組織に閉じたアプリケーションサーバではなく、社会全体に浸透した D-Case の管理ノードとして広く存在することになるでしょう。すなわち D-ADD は企業だけでなく政府、教育機関、さまざまなインフラ、家庭、個人を結ぶ基盤となり、社会の安定運用を 24 時間 365 日、DEOS プロセスのたゆまない実践によって支えていくということです。

インターネットが「場」となり、プログラムの性格を有した D-Case ならびに D-Script が疑似「生命体」で、これらを統合的に管理するのが D-ADD と表現することもできます。

5.4. D-ADD の再考 - 構造化が社会を変える

第 4 章と第 5 章で、D-ADD の活用シナリオと将来の発展イメージを明らかにしました。最後に、D-ADD はなぜこのようなポテンシャルを持ち得るのか、その鍵となる製品思想を説明します。

D-ADD の根底にある思想は、データの構造化技術を採用するということです。ここでいう構造化とは、文字や数字で表現された情報に意味を与えることです。意味そのものをどう定義するかという点はさておき、情報に意味もしくは意味的要素を加えることによって人類の文化、言語的違いを超えた共通の考え方や規格を共有し、普遍化する基盤になります。

インターネットは情報の流通性に革命を起こしました。異機種分散ネットワークでのデータ流通コストは人類史上かつてないほど安価になり、知識のオープン化は国境を越えるまでになりました。経済のグローバリゼーションが加速し、発展途上国が急激に先進国に追いつこうとする中で貧困問題は解消に向かい、ボーダーレスな社会へと向かっています。

しかしインターネットで流通している情報は人間による解釈が必要なものととどまっています。翻訳技術の進展はあっても、文章や音楽、映像はすべて人間の脳によって処理される必要があります。私たちが考える次世代の情報流通基盤とは、情報と意味を一体化させて流通させることにより、コンピュータ同士で情報を処理できるようにすることです。

これによって情報処理技術は次の段階へステップアップすることでしょう。そのためにはデータの構造化技術が不可欠です。

現在、このようなデータ、特に日々発生する膨大なトランザクションデータを扱っている大企業・政府・その他の組織は、データの管理手法に RDB を採用しています。RDB はデータ同士の関連付けを行うことができますが、あくまでの組織内での関連付けにとどまっています。組織と組織を超えた情報に対する構造化技術の一つとして XML によるタグ付けという手法がありますが、これらの XML データは再び組織内に閉じた形で RDB に格納されるというのが現状であり、その変換に要するコストや、意味解釈の違いによる一部情報の損失など多くの課題を抱えています。しかるにエンタープライズアプリケーションの基盤としてもっとも採用されている Java EE がデータの永続化に RDB を使っているため、情報の取り扱いに関して、もはやボトルネックとなっています。組織内での部分最適を進めた結果、必ずしも社会全体の最適にはつながっていないということです。

しかし今、その流れが変わろうとしています。インターネットの普及により企業が直接、顧客と接点を持てるようになったことから、何をサービスするかという本質的なビジネスモデルそのものが見直されています。その中で企業が取り扱うデータも伝票系（売上や在庫など）から、現場で発生した時点のデータ（生データ）を管理することの重要性が理解されはじめています。生データは「ビッグデータ」⁹として社会に認知されつつありますが、ビッグデータの管理に RDB は不向きであることが知られており¹⁰、かつ適切な構造化を行わないと活用が難しいこともわかってきました。さらに政府組織や研究機関では、構造化されたデータを「オープンデータ」¹¹として広く公開し、社会に還元するという動きもあります。これらの動きを支えるためのハードウェアも進歩しており、並列処理やオンメモリデータ処理といった、大量データの扱いに関する基盤技術も進化しています。データの構造化というテーマそのものは古くから語られてきましたが、いよいよその重要性が無視できなくなってきたと私たちは考えています。この実現により、インターネットは人類が扱う情報を適切に提供し、処理するための第二世代に入ることでしょう。D-ADD はこの流れを押し進めるための原動力として、活躍しようとしています。

⁹ <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc121410.html>

¹⁰ <http://itpro.nikkeibp.co.jp/article/COLUMN/20120312/385905/>

¹¹ <http://opendatahandbook.org/ja/what-is-open-data/index.html>

6. よくある質問と回答

ここでは D-ADD に関するよくある質問と回答を紹介します。

6.1. D-ADD はシステム運用監視ツールですか。

D-ADD はヒューレット・パッカード社の OpenView や NTT データの Hinemos、オープンソースの Zabbix のようなシステム運用監視ツールと類似しているように感じられたかも知れません。もちろん、D-ADD をそのような目的で活用することはできるでしょう。D-Script は汎用性があるため、システムの異常値をモニタリングし、閾値によって予め用意した D-Script を実行させることが可能です。

しかし D-ADD はシステム運用監視を目的として設計されたソフトウェアではありません。このホワイトペーパーが一貫して説明しているとおり、D-ADD は DEOS プロセス（変化に対応したサイクルと、障害に対応したサイクル）という二つのサイクルを実現する基盤ソフトウェアとして生み出されたものです。システム運用監視とは、そのサイクルによって提供される具体的なサービスの一つですが、D-ADD 自体は合意記述データベースというアーキテクチャに基づいているため、より多くの目的で活用することができます。具体的な例として、第 4 章「D-ADD の活用シナリオ」をお読み下さい。

6.2. D-Case は仕様記述言語ですか。

システムの要件を定められた形式で記述すると、その記述に従ってシステムを自動生成するものや、システムの動作パラメータとして活用されるような仕組みを備えたソフトウェアが存在します。このようなソフトウェアへの入力となる、業務ルール記述を独自の「仕様記述言語」として表現することがあります。特に、ある業務に特化した仕様記述言語を DSL (Domain Specific Language) と呼びます。

しかし D-Case はこのような仕様記述言語とは異なります。D-Case の基盤は GSN という論理の木構造表現方式であり、かつ自由文章での記述です。よって D-Case を入力としてシステムの挙動を変えらるということには向いていません。人間のためのドキュメント（ただし木構造化された合意形成のためのドキュメントと、その論拠となる各資料）という位置づけです。

このように D-Case は Machine Readable ではありませんが、D-Case とともに用いられる D-Script は拡張タグを利用することでシステムの振る舞いを記述する能力をもっています。

6.3. DEOS は国際標準なのですか。

DEOS プロジェクトが提供する「DEOS プロセス」および「D-Case 記述」は ISO/IEC JTC1 SC7 Systems and software Engineering と IEC TC56 Dependability にて標準化活動を実施しています。また「D-ADD」や「D-Script」は OMG (Object Management Group) の SysA (System Assurance) Task Force と、TOG (The Open Group) にて標準化活動を開始しました。OMG の SysA Task Force ではアシュアランスケース標準に着手しており、D-Case もその中で議論されています。

6.4. DEOS プロセスはウォーターフォール開発とアジャイル開発のどちらに向いていますか。

ウォーターフォール開発やアジャイル開発とは、アプリケーション開発におけるプロジェクトの進め方に関する視点です。一方で DEOS プロセスは、合意形成の過程をすべて記録することを前提としており、運用後のアプリケーション拡張を適切に行えること（変化対応サイクル）ならびに障害発生時の対応を行うこと（障害対応サイクル）を一体として扱うもので、プロジェクトの進め方については触れていません。よっていずれの開発手法にも適用できます。より正確には、どのような開発手法であっても、合意形成の過程を記録し続けるという考え方は重要になります。さらに近年注目されている、開発と運用を分離せずに一体として扱うという DevOps という考え方を包含しているため、さまざまなベストプラクティスと組み合わせて利用することができるといえます。このような開発を支える基盤として D-ADD が活用できます。

6.5. D-ADD はどこから入手できますか。

D-ADD のオープンソース版は一般社団法人ディペンダビリティ技術推進協会（DEOS 協会）の Web サイトからダウンロードできます。D-ADD はオープンソース版とエンタープライズ版のデュアルライセンスで提供されています。サポートを受けるためにはエンタープライズ版をご利用ください。エンタープライズ版は株式会社 Symphony にて取り扱っています。

6. 6. D-ADD を使ってみたいのですが、D-Case の記述方法に馴れていません。

D-Case の実業務への展開においては、導入時に専門家のコンサルティングを受けることを推奨しています。DEOS 研究開発センターでは、D-Case のコンサルタントを紹介しています。

6. 7. D-ADD の機能拡張に当社の希望を伝えたいのですが。

可能です。今後の D-ADD の機能拡張に関して、共同で進めるためのスキームをご提案できます。詳細は株式会社 Symphony D-ADD 事業部担当へお問い合わせ下さい。

7. おわりに

D-ADDは、D-Caseのデータベースとして考案されました。D-Caseは、アシュアランスケースを元にDEOSプロジェクトが新たにモニターノード、アクションノードをD-Scriptとして定義し、ソフトウェアとして実行可能なプログラムとして研究されてきました。従ってプログラムとしてのD-Caseを格納するD-ADDは当初よりリポジトリとしての性能を重視しています。

D-Caseを統治するD-ADDですが、D-Caseにより制定されるディペンダビリティは関係者によって合意される必要があり、D-Caseの制定の経緯とその合意に至る経緯は、二つの状態を正しく変遷管理せねばなりません。合意は、DEOSプロセスの実行タスクになりますので、DEOSプロセスのアプリケーションとして命名され、D-ADD (DEOS Agreement Description Database) として生まれました。かくしてD-ADDは、DEOSプロセスを実行状態にする重要なキーコンポーネントになりました。

このような必要条件を満たすD-ADDの研究は、領域の他の研究チームの成果を統合し、よりよいDEOSプロセスを実行状態にする能力を持たねばならないことから、各研究チームの成果をユースケースとして見立てて、統合開発環境として着実にステップアップ形式で拡張してきました。そして、DEOSプロセスを実現するD-ADDですから、D-ADD自体の開発も合意記述を研究者、開発者の間で絶えず行いながら開発されてきました。D-ADDがDEOSプロセスの開発環境になるであろうことは研究当初からの課題であり、次世代のIT開発環境を目指そうと設計段階からその目的を持たせています。それは、第5章のD-ADDの将来で触れている「非RDB系のWebアプリケーションサーバ」として結実しようとしています。

DEOSプロセスが国際標準化を目指している中で、その開発環境が非RDB系のWebアプリケーションサーバとして世の中に登場しようとしていることは、日本が世界のIT技術に貢献できる無限の可能性を秘めています。なぜならば、今まさにビックデータ時代をむかえ、ディペンダビリティの重要性は極限まで達しているからです。

株式会社Symphonyは、20年以上前から一貫してデータの構造化の重要性と、それが社会に与えるインパクトを考察するとともに、さまざまな構造化技術の開発を行ってきました。今、私たちはDEOSと出会い、D-ADDの開発によってデータ構造化の基盤を提供する準備が整いました。D-ADDの研究開発が、いかに先進的であり、優れたアーキテクチャを有して

いるかは、今後D-ADDを実用化していく過程でそれを証明して、多くの先進ユーザに受入れられるかを模索することになります。そして日々重要性が増しているITシステムのディペンダビリティに具体的に寄与し、D-ADDそのものが人類進化のためのデータベースであることを目指す日が来ることを期待してやみません。そしてD-ADDが人類共通の考え方や規格を管理流通する「場」として使われるようになることが私の願いです。

株式会社Symphony
代表取締役 永山辰巳

D-ADD Technical White Paper

Revision: 1.0

株式会社 Symphony

本書に掲載されている文言、画像等の無断転載はご遠慮下さい。