

JST-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」

DEOS プロジェクト



DEOS Programming Reference

Version 1.0

2013/05/01

DEOS 研究開発センター



科学技術振興機構
Japan Science and Technology Agency

目次

1. はじめに	3
2. DEOS プログラミングの考え方	4
2.1. DEOS プログラミングの目標	4
2.2. DEOS プログラミングの概要	6
2.3. DEOS プログラミングの考慮点	8
3. D-Case/D-RE のデモ・システムにおける実装事例	10
3.1. デモ・システムの概要	10
3.2. デモ・システムのソフトウェア構造	12
3.3. DEOS プログラムの開発から実行までの流れ	12
3.4. D-Case の監視系と分析系ノードの実行の仕組み	13
3.5. D-Case の監視系ノードの作成	16
3.6. D-Case の分析系ノードの作成	19
3.7. Action 系のコマンド例	23
4. おわりに	23
Appendix	24
A1.監視系と分析系モジュール・テンプレートの例	24
A2.デモ・システムの D-Case	41

本書に記載されているシステム名、製品名、サービス名などは一般に各社の商標または登録商標です。

本書は次の DEOS プロジェクト文書を事前に読まれている事を前提とする。

DEOS-FY2012-PU-01J : DEOS Project Update 2012

また、以下の DEOS ホームページ、DEOS プロジェクト文書を参考資料とする。

DEOS ホームページ URL : <http://www.dependable-os.net/>

DEOS-FY2013-RE-01J : D-RE 仕様書、D-RE 導入ガイド付き

DEOS-FY2013-EA-01J : D-RE API 仕様書

DEOS-FY2013-EC-01J : D-RE コマンド仕様書

DEOS-FY2013-SP-01J : D-RE API サンプルプログラム解説書

1. はじめに

現代のコンピュータシステムは、

- システムの複雑化・巨大化
- システムのネットワーク化
- 個別システムの集合体（寄せ集め）化
 - ▶ 個別システム・構成要素のブラックボックス化
 - ▶ システムの増改築・レガシーコードへの依存度増大

が進み、全体を把握し、詳細を理解し、完璧に構築する事は至難の業で、常に不完全である。

その上、

- サービス利用者の要求の変容、社会のルールや仕組みの変容
- 事業目的の変容、サービスの多様化
- ネットワークで接続された他システムの仕様の変容

などシステム外部との相互作用が常に存在し、動作環境は不確実である。

すなわち、現代のコンピュータシステムは機能、構造、システム境界が変化しつづけるオープンシステム（開放系）であり、これに起因する不完全さと不確実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。

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果たし、およびそれらを継続的に行う能力が必要である。

この能力を「OSD：オープンシステムディペンダビリティ（Open Systems Dependability）」と呼び、この OSD を実用化し社会に生かすための、プロセスや開発・実行のための環境・ツールやプログラミング技法などの知識・技術を扱う分野を「DEOS：開放系のためのディペンダビリティ工学（Dependability Engineering for Open Systems）」と名付ける。

DEOS の実行による OSD の実現、すなわち「変化しつづけるシステムのサービス継続と説明責任の全う」のためには以下が必須である。

- 継続的な改良改善のための開発と運用を連携する反復的なプロセス（DEOS プロセス）
- DEOS プロセスを仕組みとして支える合意形成・確認手法・ツール（D-Case）と実行環境（D-RE：DEOS Runtime Environment）
- D-Case に基づいて、OSD 実現を具体化した、スクリプト（D-Script）およびアプリケーション・プログラム（D-Aware Application Program）

本 Programming Reference では、この 3 番目の必須要素であるスクリプトおよびアプリケーション・プログラムを書く上での考え方（DEOS プログラミング）を第 2 章で、デモ・システムにおける事例を第 3 章で紹介する。本書は次の DEOS プロジェクト文書を事前に読まれている事を前提とする。

DEOS-FY2012-PU-01J : DEOS Project Update 2012

2. DEOS プログラミングの考え方

2.1. DEOS プログラミングの目標

前節でも述べられているように、現代のコンピュータシステムでは障害は避けられない。DEOS プロセスでは **D-Case** に次のような対応が記述される事を求めている。

A) 通常運用として

- 1) 重要な判断情報を得るための監視 (Monitoring)
- 2) 監視した情報の適切な記録 (Logging)

B) 事前対応として、

- 1) できる限り**予兆を検出**する
 - 1-1) 予兆を検出したら、できる限り障害を D-Case に基づいて**未然に回避** (バックアップへの切り替えを含む) する
 - 1-2) 回避できない場合、D-Case に基づいて**縮退運転** (部分リセット、性能の制限を含む) が可能か判断する
 - 1-2-1) 可能であれば、縮退運転をする
 - 1-3) しかるべき部署・責任者へ連絡する

C) 事後対応として

- 1) できる限り早く**障害を検知**する
 - 1-1) D-Case に基づいてバックアップへ切り替えか、障害部位の部分リセットを行う
 - 1-2) 上記ができない場合、**縮退運転** (障害部位の切り離し、稼働性能の制限を含む) が可能か D-Case に基づいて判断する
 - 1-2-1) 可能であれば、縮退運転をする
 - 1-2-2) 不可能であれば、**システムリセット**が可能か判断する
 - 1-2-2-1) 可能であれば、**最短リセット**が可能か判断する
 - 1-2-2-1-1) 可能であれば、最短リセットを実施する
 - 1-2-2-1-2) 不可能であれば、通常リセットを実施する
 - 1-3) しかるべき部署・責任者へ連絡する

これらを実現するためには、System Designer、System Programmer、Application Designer、Application Programmer は、次の6つのステップを常に考えなくてはならない。

監視・検出 (Monitoring・Sensing)
記録 (Logging)
報告・通知・発信 (Reporting・Notification・Signaling)
分析 (Analysis)
診断/判断 (Diagnosis)
反応/実行 (Action)

これらのステップを考慮し、DEOS プロセスの実現を支援するプログラミングを **DEOS プログラミング** と呼ぶ。これは Software のみで対応するものではなく、サーバー群や冗長性のあるネットワークなどの Hardware 系、Operator や Programmer を含む人間系を含んだものであってもよい。

- ✓ 必要な Monitoring・Sensing、Logging、Reporting・Notification・Signaling を可能な限り実現する
- ✓ 必要な Analysis、Diagnosis を可能な限り実現する
- ✓ 必要な Action を可能な限り実現する

Action を支えるシステムの機能として

- Migration
- Isolation
- Quota
- Undo
- Reboot
- Debug

が求められる。

図 1 はこれらの関係を整理したものである。

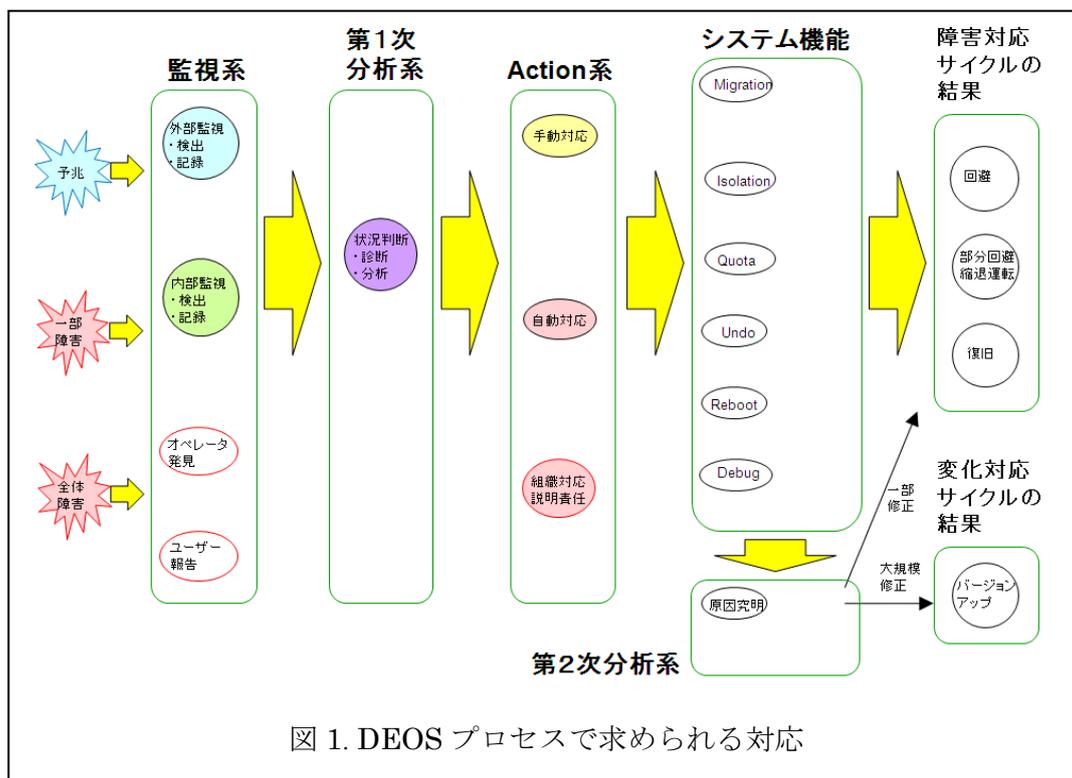


図 1. DEOS プロセスで求められる対応

以上を簡潔にまとめると、

DEOS プログラミング

=D-Case に基づくプログラミング

=D-RE で実行される D-Script および D-Aware Application のプログラミング

=DEOS プロセスを有効に回すプログラミング

と言う事になる。

2.2. DEOS プログラミングの概要

何を Monitor するかは合意形成プロセスで議論され、D-Case に記述される。Monitor 結果を診断した結果の Action も、D-Case に記述される。

Script 言語としては、市販もしくは OSS のスクリプト言語が存在している。DEOS プロジェクトでは、軽量・小型・高性能の **D-Script** を研究開発した（詳細は、以下の Document 参照）。

- D-Script — スクリプトによる障害対応の実現 （文書番号 DEOS-FY2012-DS-01JA）

Monitoring、Logging、Reporting は次のように作られる。

- Application の仕事の内容(条件)を Application の中で Monitoring する場合は、Application の中に Monitoring Point が設定される
- Application の動作を Application の外から Monitoring する場合は、D-Application Monitor を通じて行う
- System あるいは Container の動作を System Program の中で Monitoring する場合は、System Program の中に Monitoring Point が設定される
- System あるいは Container の動作を外から Monitoring する場合は、D-System Monitor を通じて行う

Diagnosis、Action は次のように作られる。

- Application の仕事の内容(条件)を Application の中で分析、対応する場合は、Diagnosis、Action が Application の中でプログラムされる
- Application の動作を Application の外から分析、対応する場合は、Diagnosis、Action が外部 Program あるいはスクリプトでプログラムされる
- System あるいは Container レベルで分析、対応する場合は、Diagnosis、Action が System Program の中でプログラムされる
- System あるいは Container の動作を外部から分析、対応する場合は、Diagnosis、Action が D-System Monitor の中でプログラムされる

基本的な System の動作状況 (CPU 負荷、メモリ使用量) の Monitoring には D-RE が提供する C の API を使う事もできる。

また継続的な動作状況 (統計状況) を Monitoring する場合は、市販プログラムもしくは OSS (Nagios、Zabbix 等) を利用する事もできる。

DEOS プログラミングは VM、Hypervisor、Container と言った基盤技術を活用する。これらには市販もしくは OSS の Middleware (VMware や OpenStack 等) がある。DEOS プロジェクトでは、Reference Platform として KVM や LXC を使った D-RE (DEOS Runtime Environment) を開発し、コードを公開している（詳細は、以下の Document 参照）。

- D-RE 仕様書 （文書番号 DEOS-FY2013-RE-01J）
- D-RE API 仕様書 （文書番号 DEOS-FY2013-EA-01J）
- D-RE コマンド仕様書 （文書番号 DEOS-FY2013-EC-01J）
- D-RE API サンプルプログラム解説書 （文書番号 DEOS-FY2013-SP-01J）

図 2 はこれらの技術が障害対応サイクルの実現にどう使われるかを示す。

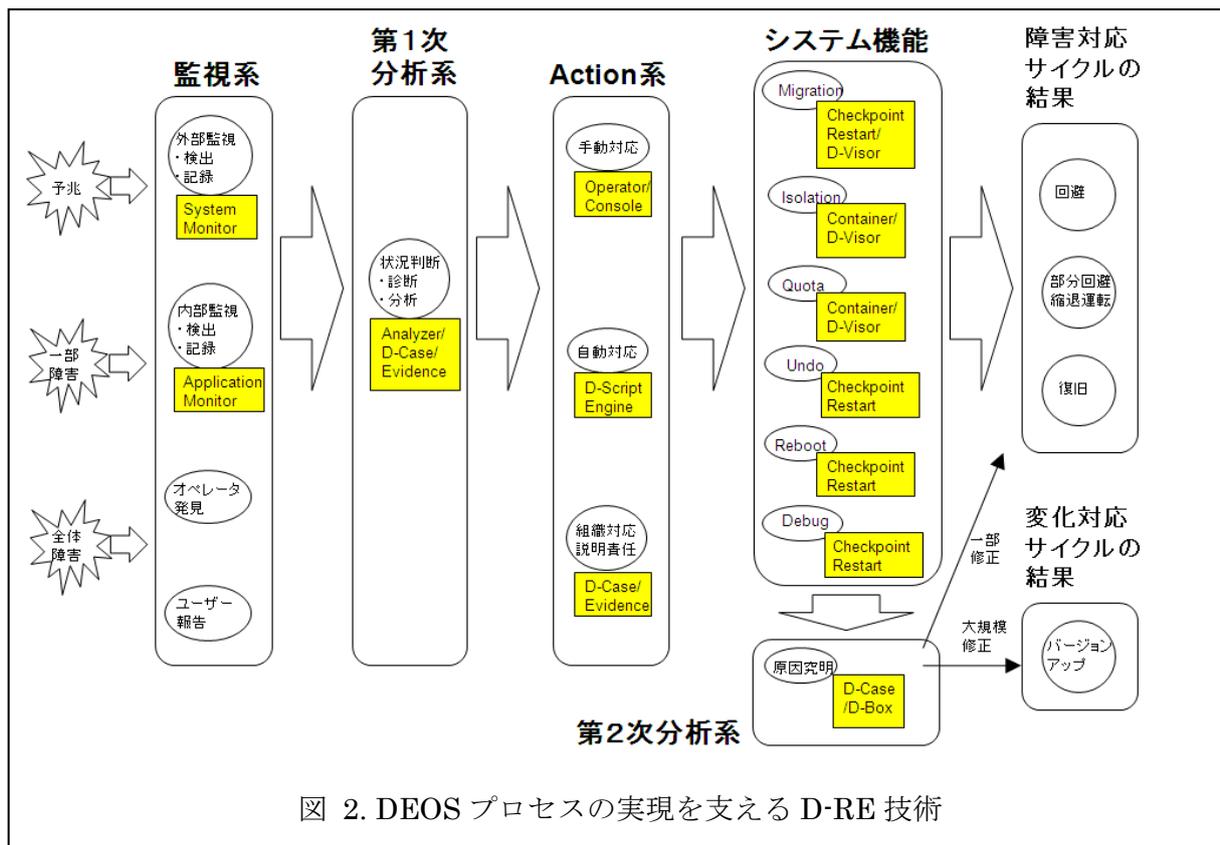


図 2. DEOS プロセスの実現を支える D-RE 技術

2.3. DEOS プログラミングの考慮点

1) 監視系

何を Monitoring するかは合意形成プロセスで議論され、D-Case に記述される。HW 障害の予兆検出は、System に適切なセンサーが実装されていて、検出 SW が生きている場合は比較的实现の可能性が高い。これに比して SW 障害の予兆検出は難しい。

- 長時間統計を取っていて、過去の障害時のパターンと似たパターンが現れる時は、障害の可能性はあるが、確実に障害になるとは言い切れない。
- メモリ使用量が徐々に増加する場合は、メモリ・リークの疑いがあり、障害に至る前に、可能なタイミングでのリブート等の若化操作は推奨できる。
- システム全体が Down する場合、検知が不可能である。これに対応するためには、以下の方法が考えられる。
 - ✓ 検出系を別システムにする
 - ✓ Operator が常駐している場合は Operator が異常を検知する
 - ✓ Operator がいない場合、End User からの通報による
- Monitor Program、あるいは Monitoring Point を有する Application Program が動作していて、他の部分が Down している場合、検知できる可能性は高い

D-RE には、OS の異常、あるいは OS への Malware 攻撃などを検出するための D-System Monitor を実装する事ができる。

次に Logging、Reporting であるが、何をいかに Logging、Reporting するかは D-Case に記述がある。

- 検知すると同時に、障害が起きた時の System 環境情報を Log しておく事は、原因究明に役立つ
- 検知したら、いかにその事実を System および Operator に通知するかが次の重要な Step となる
 - ✓ Diagnosis Program がある場合には、そこへ通知する
 - ✓ Operator には必ず通知する

2) 分析系

何をいかに Diagnosis するかは、D-Case に記述されるが、常に明示的に書かれるとは限らない。分析系の記述に含まれる事もあるし、Action 系の記述に包含されている場合もある。

- System Programmer もしくは Application Programmer は、D-Case から Diagnosis の Logic を組み立てる
 - 多くの場合、Log を Diagnosis する
- Log が重複する情報でパンク (Log 破産) しないように、情報の整理が必要である
 - デモ・システムでは SEC (Simple Event Correlator、<http://simple-evcorr.sourceforge.net/>) を使用した
- Log 内容の整合性 (記録の正統性、改竄等がなされていない事など) を保証する必要がある

3) Action 系

何をいかに Action するかは、D-Case に記述されるが、常に明示的に書かれるとは限らない。監視系の記述に含まれる事もあるし、分析系の記述に包含されている場合もある。

- 多くの場合、Application に関係した障害は Application の中で、Action がプログラムされる
 - ✓ あらかじめ用意されたいくつかの対応策がある場合
 - ◆ Application もしくはスクリプトが自動的に選択する
 - ◆ Operator に提示し、Operator が選択する
- System の動作の場合、D-Case にしたがって対応策があらかじめ用意される
 - ✓ スクリプトにより自動的に実行される
 - ✓ Operator に提示され、Operator により実行される

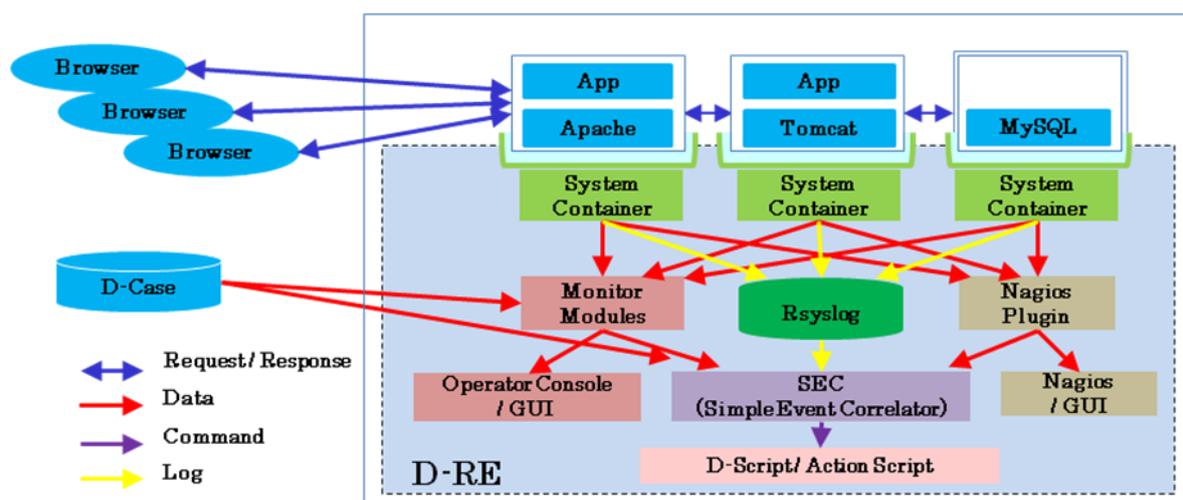
3. D-Case/D-RE のデモ・システムにおける実装事例

3.1. デモ・システムの概要

本デモ・システムは、D-Case/D-RE の機能を紹介するためのものである。

ソフトウェア・モジュール構成

このデモ・システムは、Apache と Tomcat、MySQL を使った、Web 上での CD 販売サービス (CD Online Shopping システム) をシミュレートしたもので、Apache と Tomcat、MySQL はそれぞれ独立した System Container 上で稼働している。また、このデモ・システムでは、コンピュータシステム及びネットワークの監視のためのアプリケーション Nagios (<http://www.nagios.org/>) も使っている。



また、ネットワーク上の複数の Browser からの複数の同時リクエストをシミュレートするために、ab (Apache HTTP server benchmarking tool) を使っている。

デモ・シナリオ

このシステムは、典型的な Web/Application/DB Server で構成されている。また、これらのサーバーを Operator Console で常に監視している。ソフトウェア構成としては、各サーバーは、D-RE (DEOS Runtime Environment) のシステムコンテナ上に実装されている。Operator Console では、System Container からのログを Rsyslog に集約し、モニター・モジュールと Nagios Plugins で、各サーバーのリソース及びパフォーマンスを監視する。モニター・モジュールは障害を検出すると、GUI に表示し、同時に SEC に情報を渡す。SEC はそれに従って D-Script を実行する。以下のシナリオは、D-Case と D-RE がシステムの障害にどのように対処するかを示している。

デモ・シナリオ 1)

新サービスを追加した結果、アクセス数が想定数を超過した。そこで、あらかじめ D-Case に記述されていた対処方法に従い、サービスの追加を Undo し、システムを正常な状態に戻す。

デモ・シナリオ 2)

サーバーのレスポンスが急激に遅れた。そこで、あらかじめ D-Case に記述されていた対処方法に従い、不適切なバッチ・ジョブを停止し、システムを正常な状態に戻す。

デモ・シナリオ 3)

メモリ使用量がシステムの許容値を超えた。そこで、あらかじめ D-Case に記述されていた対処方法に従い、サーバーを再起動し、診断モジュールを投入して原因を特定する。

デモ・シナリオ 4)

日常点検の一環として、翌日のサービスが問題ないか確認したところ、サービスが実行出来なかった。原因究明の結果、ライセンス切れであることがわかり、ライセンスを更新する。

以上のデモ・シナリオの動画は以下の URL のサイトで公開されている。

<http://www.dependable-os.net/>

3.2. デモ・システムのソフトウェア構造

D-RE を使った場合のソフトウェア構造を下図に示す。

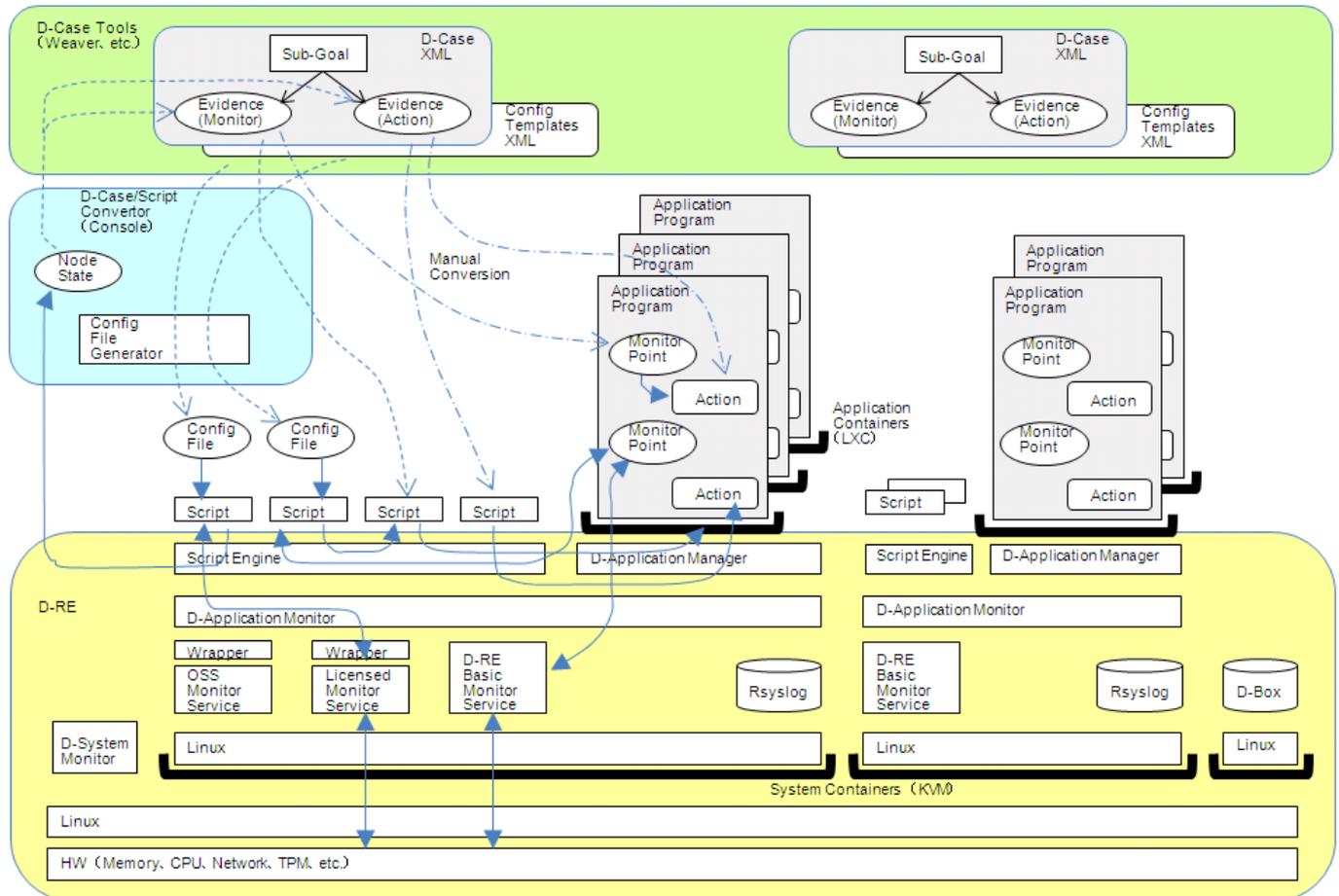


図 3. D-RE を使ったソフトウェア構造

監視系の実装は、D-Case の監視系（モニター）ノードに対応するモニター・モジュールのインスタンスにより行う。

分析系の実装は、SEC（Simple Event Correlator）を使った状態マシンに監視系のモニター・インスタンスが検知したイベントを送り、そこで複数イベントの相互関係を考慮したイベントの分析を行う。

3.3. DEOS プログラムの開発から実行までの流れ

DEOS プロセスを仕組みとして支える合意形成・確認手法・ツール (D-Case) と実行環境 (D-RE: DEOS Runtime Environment) を活用したデモ・システムは、以下のような流れで開発され、実行される。

- 1) サービス要求仕様の合意と確定 (D-Case の確定)
 - 会社の経営理念やビジネス・モデルの確認
 - BCP (Business Continuity Plan) の決定と合意
 - SLA パラメータの決定と合意
 - SLA に関する監視系（モニター）ノードとアクション・ノードも作成

- 2) 監視系 (モニター) ノードのモニター対象やパラメータの決定
既存のモニター・モジュールからの選択
新規のモニター・モジュールの仕様を決定

監視用の D-Script Description File (外部仕様)を作成

- 3) 分析系・ノードの詳細決定
Analyzers を状態マシンで実現する
Events の相互関連を状態と状態遷移で表現
状態マシン記述用の文法の選択

分析用の D-Script Description File (外部仕様1、SEC 用の Config File のテンプレート)を作成
既存の Actions の呼び出し (引用)
ユーザ固有の Actions の仕様を決定

- 4) アプリケーションの開発
- 5) アプリケーション開発と並行して、ユーザ固有のモニター・モジュールの開発
- 6) アプリケーション開発と並行して、状態マシンの定義とユーザ固有の Actions の開発
- 7) D-RE 上でのアプリケーションの実行
D-Case の監視系 (モニター) ノードとアクション・ノードからインスタンスが作成されて、実行される。

以上の流れで、DEOS プロセスが支援される。

3.4. D-Case の監視系と分析系ノードの実行の仕組み

下図は、Web Browser 上で稼動する JavaScript で書かれたツールである D-Case Weaver を用いた D-Case からモニター・モジュールのインスタンスと SEC の構成ファイルが作成される仕組みを表している。まず、D-Case Weaver は D-Case 内の D_Case_Node テーブル(表)に、モニター・モジュールを作成する。その時、D-Case Weaver は、ユーザに、監視系 (モニター) ノードに対応するモニター・モジュールを D_Script テーブルから選択させ、そのインスタンス作成のためのパラメータ群を監視系 (モニター) ノードに追加する。すなわち、D-Case Weaver は、D-Case の監視系 (モニター) ノードと D-Script (モニター・モジュール) とを結び付ける。その後、D-Case Weaver は、監視系 Config Generator と分析系 Config Generator を呼び出す。

分析系 Config Generator は、D_Case_Node 表と D_Script 表とを結合 (Join) して、各監視系 (モニター) ノードに対して、それに対応するモニター・モジュールを見つけ出し、そのモニター・モジュールの雛型に監視系 (モニター) ノード内のパラメータ群を適用して、監視系 (モニター) インスタンスを作成する。分析系 Config Generator も同様に、監視系 (モニター) ノードと D-Script (SEC の構成ファイルの雛型) とを組み合わせることで、SEC の構成ファイルの雛型に、監視系 (モニター) ノード内の具体的なパラメータ値を適用した、SEC の構成ファイルを生成する。

監視系 (モニター) ノードと分析系ノードの作成については、次節以降に記述する。

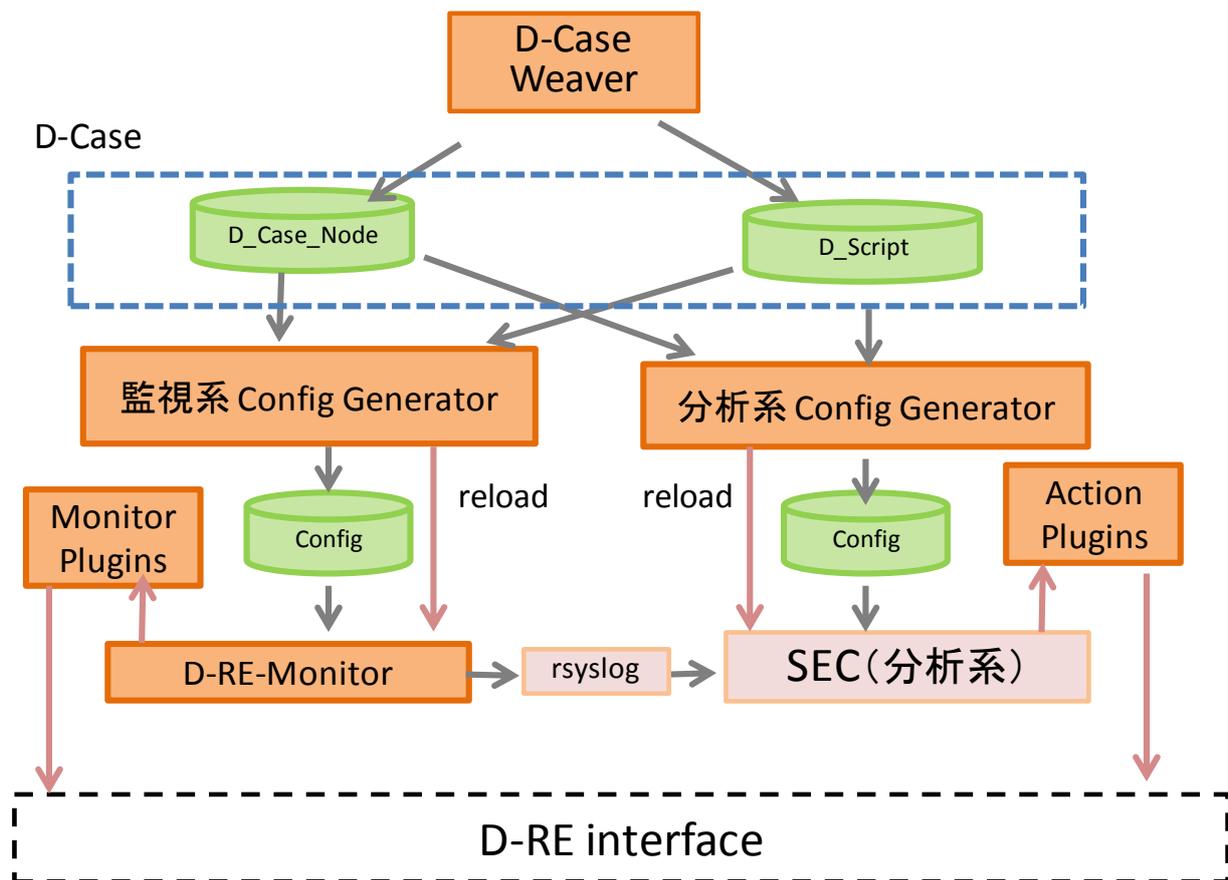


図 4. D-Case から監視系と分析系のインスタンスの生成

監視系（モニター）モジュールの実装

各モニターは、あるモニター・モジュールを特定のパラメータ群を用いてカスタマイズして生成したモニター・インスタンスである。

デモ・システムでは、モニター・モジュールは D-RE の提供する、デモのために Python で書かれた PluginBase クラスのサブクラスとして実装している。従って、モニター・インスタンスとは、そのサブクラスのインスタンスを意味していて、モニター・モジュールを表すサブクラスのコンストラクタに特定のパラメータ群を与えて生成したそのサブクラスのインスタンスである。

下図は、置換可能なパラメータ群を含む D-Script として定義されたモニター・モジュールと、D-Case 内の監視系（モニター）ノードに記述されたそれらパラメータ群の具体的値から、特定のモニター・インスタンスが生成される例を示す。

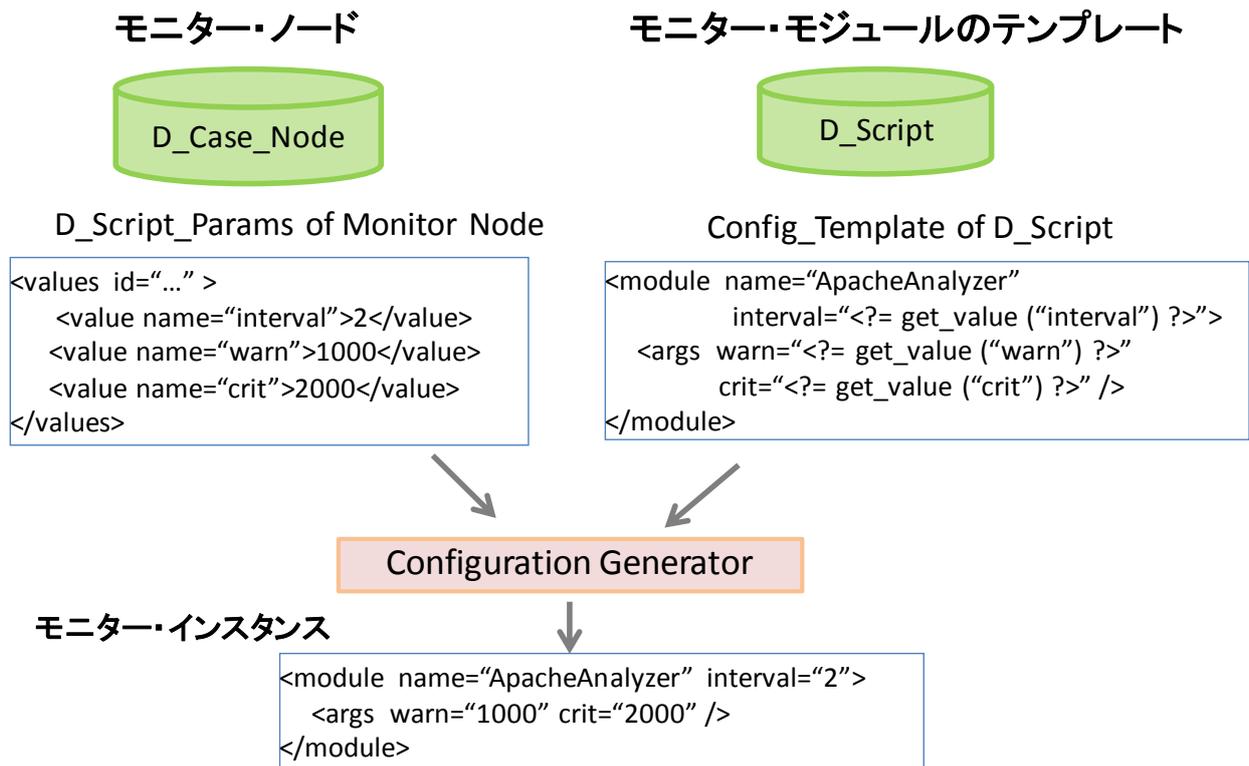


図 5. モニター・インスタンスの生成

分析系モジュールの実装

SEC を用いた状態マシンの実装において、各状態マシンと Actions も同様に、ある状態マシンや Actions のテンプレートを特定のパラメータ群を用いてカスタマイズして生成したものである。

監視系（モニター）モジュールと分析系モジュールのテンプレート

モジュール・テンプレートは、以下を含む XML 形式のファイルである。

- ・ モニター・モジュールや状態マシンのテンプレートを指定する識別子
- ・ 機能の説明、著作者、設定可能なパラメータ群及びその説明
- ・ パラメータを置換してモニターや状態マシン・Action のインスタンスを生成する基となるテンプレート

監視系（モニター）モジュールと分析系モジュールのテンプレートの一例としては、付録を参照。

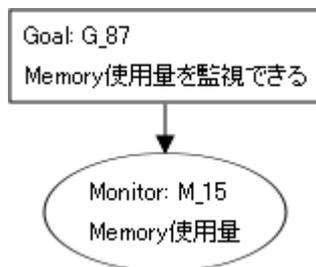
3.5. D-Case の監視系ノードの作成

D-RE 上で実行するプログラムに対する監視は、D-Case にモニターに関するエビデンス・ノードに記述される。通常は、プログラムの設計時の D-Case の作成過程で、サービス要求仕様を満足するために必要なモニターに関するエビデンス・ノードが特定されていく。

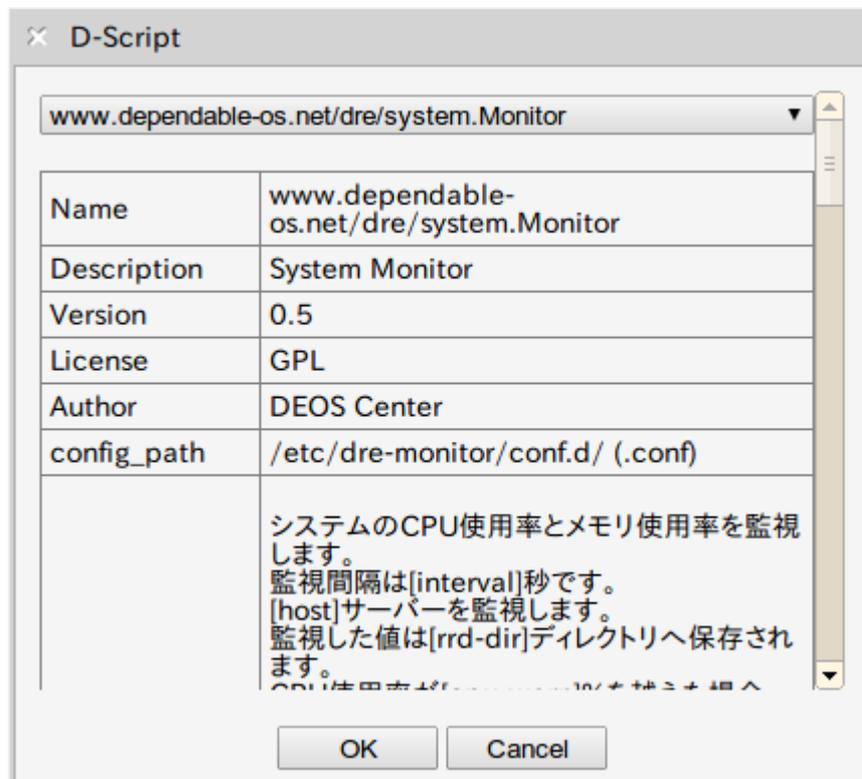
各監視系（モニター）ノードには、モニター・モジュールの特定やそのモニター・モジュールが実行時に必要とするパラメータ群の値を指定する。ここでは、D-Case Weaver を使ったパラメータ群の指定の方法を説明する。D-Case Weaver は、Web Browser 上で稼働する D-Case を編集するツールである (DEOS-FY2013-CW-01J: 「D-Case Weaver 仕様書・導入・使い方ガイド」を参照)。

1) CPU とメモリ Usage の監視

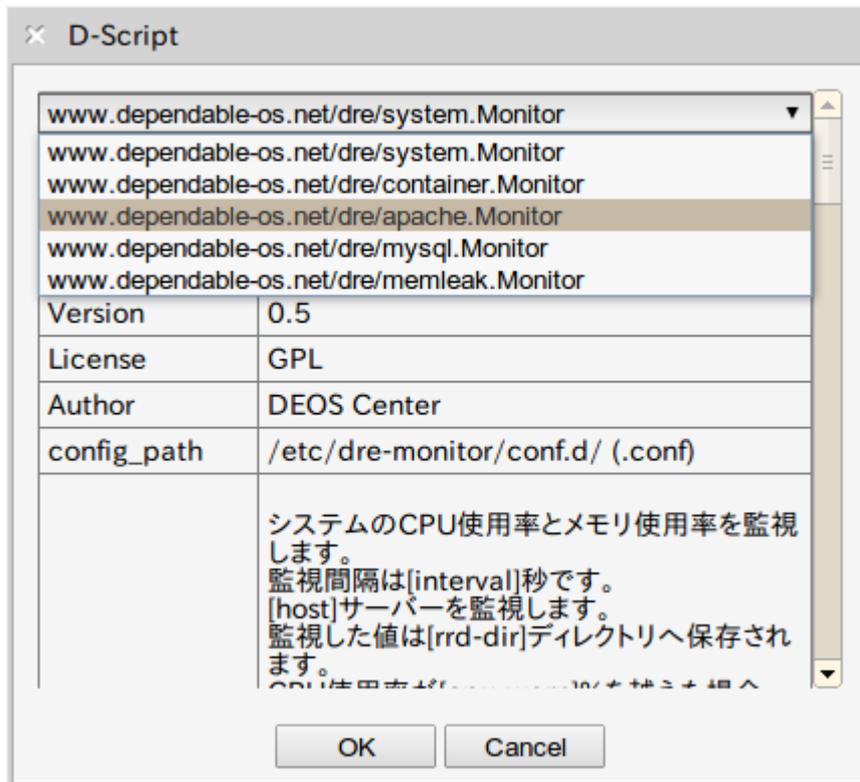
D-Case の D-Case Weaver 上での表現：



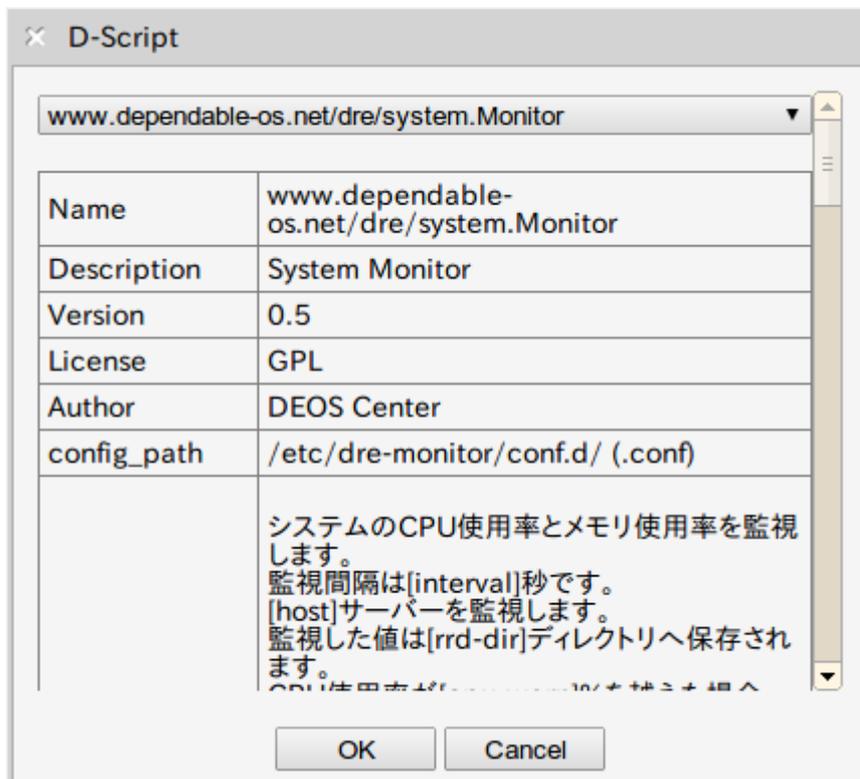
監視系（モニター）ノード M_15 の作成後、ノード上で右クリックして、Edit D-Script を選択し、モニター・モジュールの選択とパラメータ群の指定を行う。



リストボックスからモニター・モジュールを選択する。
新規のモニター・モジュールの導入方法は Appendix 参照。



モニター・モジュールの種類毎に表示される必要なパラメータ群に対して、適切な値を設定する。



The screenshot shows a dialog box titled "D-Script" with a close button (X). It contains two sections, each with a table of configuration parameters. The first section has parameters: name (interval), type (integer), description (監視間隔[秒]), default (2), and value (2). The second section has parameters: name (host), type (string), description (監視対象のホスト名), default (localhost), and value (sys-tom). At the bottom are "OK" and "Cancel" buttons.

name	interval
type	integer
description	監視間隔[秒]
default	2
value	2

name	host
type	string
description	監視対象のホスト名
default	localhost
value	sys-tom

この例では、以下のような監視用パラメータ群を指定する D-Case 内の XML ノードが記録される。

XML :

```
<dcase:node type="Monitor" id="_JHchkKurEeGUa93z1Rd6MQ" name="M_15">
<dcase:description>Memory 使用量</dcase:description>
<dcase:properties><dcase:property name="Attachment" value=""/>
</dcase:properties>

<dcase:d-script>
<dcase:full-name>www.dependable-os.net/dre/system.Monitor</dcase:full-name>
<dcase:values>
<dcase:value name="interval">2</dcase:value>
<dcase:value name="host">sys-tom</dcase:value>
<dcase:value name="user">dre</dcase:value>
<dcase:value name="cpu-crit">90</dcase:value>
<dcase:value name="cpu-warn">80</dcase:value>
<dcase:value name="mem-crit">90</dcase:value>
<dcase:value name="mem-warn">80</dcase:value>
<dcase:value name="related-node"/>
<dcase:value name="rrd-dir">/var/lib/dre-monitor/rrd/sys-tom/system</dcase:value>
<dcase:value name="graph-dir">/var/lib/dre-monitor/chart/sys-tom/system</dcase:value>
</dcase:values>
</dcase:d-script>
</dcase:node>
```

Script : D-RE builtin monitor (ID: www.dependable-os.net/dre/system.Monitor)

3.6. D-Case の分析系ノードの作成

監視系のモニター・インスタンスは特定の監視対象の状態を監視し、障害・異常やその予兆等を検出する。一般的に、同一の障害原因について、複数のモニター・インスタンスがそれぞれの観点から障害を検出し、独立に報告する。また、あるモニター・インスタンスが複数の種類の障害を報告する場合に、分析系の実装が、複数のモニター・インスタンスからの障害報告と組み合わせることにより、障害原因の範囲を狭めることができる。従って、分析系の実装においては、複数のモニター・インスタンスからの障害報告(イベント)を総合的に解釈して、最も蓋然性のある障害原因を推定することが必要となる。

デモ・システムで実装した分析系は、(拡張された)有限状態オートマトンを用いて複数のイベントの相互間の関係を考慮した分析を行う。複数のイベントを総合的に解釈する方法として、イベントの検出に対して条件分岐(if文)を行い、それらの条件分岐を組み合わせることも考えられるが、複数イベントの発生順序の組合せに対応して複数の条件分岐の実行順序を並び変えた条件分岐列の数が指数関数的に増大する問題がある。さらに、複数イベントがある一定時間内に発生した場合の検出することも、条件分岐文の組合せでは簡単ではない。

下図は、初期状態において Event A が発生すると、Action A を実行して、状態 A に遷移し、状態 A で Event B が発生すると Action B を実行して、状態 B に遷移する状態遷移を含む有限状態オートマトンを示している。

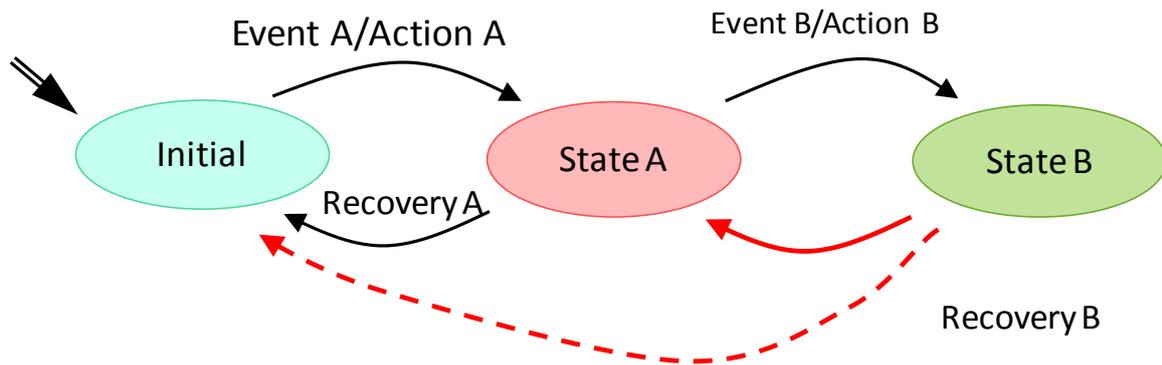


図 6. 状態遷移と Actions

次図は Event A と B の発生順序に関係なく、どちらも発生すると状態 C に遷移する有限状態オートマトンを表している。

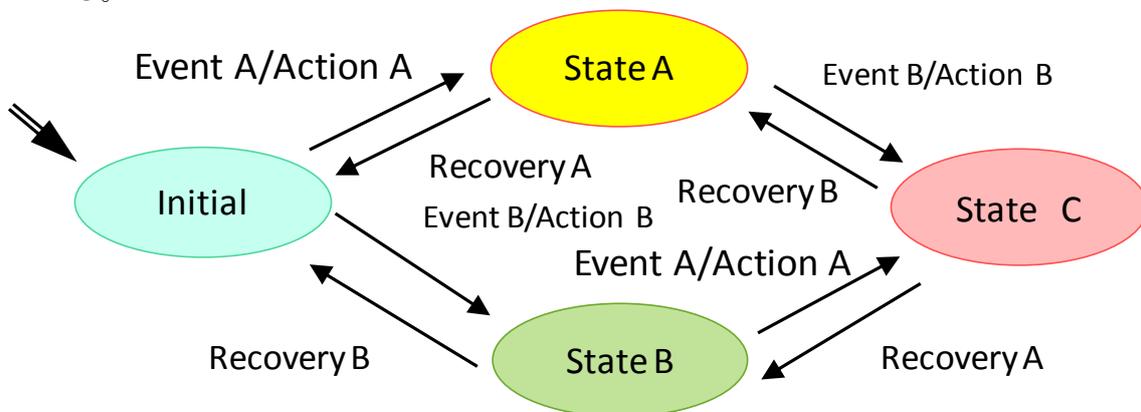


図 7. 発生順序に依存しない Events の処理

また、各状態に、その状態に留まることのできる時間の最大値を指定することにより、複数のイベントがある一定の期間内に発生したことを検出することもできる。例えば、状態Aでの最大滞在時間を30秒とすることにより、Event Aの発生後、30秒以内にEvent Bが発生すると、状態Cに遷移して、ある処理を行うように指定することもできる。

有限状態オートマトンを使う分析系の実装において、各有限状態オートマトンは関連するイベント列を受け付け、対応する状態遷移を行うことにより、複数のイベントにより表わされる障害原因を絞り込んで行く。さらに、各D-Case Nodeに対応した有限状態オートマトンをD-Case全体として複数個を疑似的に同時並行的に動作させることにより、互いに独立したイベント列の分析は同時並行的に行うことができると同時に、同じイベント列を複数の有限状態オートマトンが受け付けることにより、イベント列を複数の観点から同時に分析することができる。これは、イベント列の分析を条件文の組合せにより行うよりも容易であると考えられる。

以下の図は、ネットワーク上の障害のように、通常の運用時においても単発的に発生するイベントと、障害発生時に生ずる連続したイベントを処理する例を示している。最初にイベントに対しては、記録のためにログに書き出す等の処理をした後Burst状態に遷移し、最大Timeout時間だけその状態に留まる。その状態でイベントが発生した場合、N個発生するまでは、何もしないで、N個目のイベントに対して、恒久的な障害と認識して、管理者への通知等を行うとともに、Dormant(休眠)状態に遷移する。Dormant状態では、同イベントは無視される。

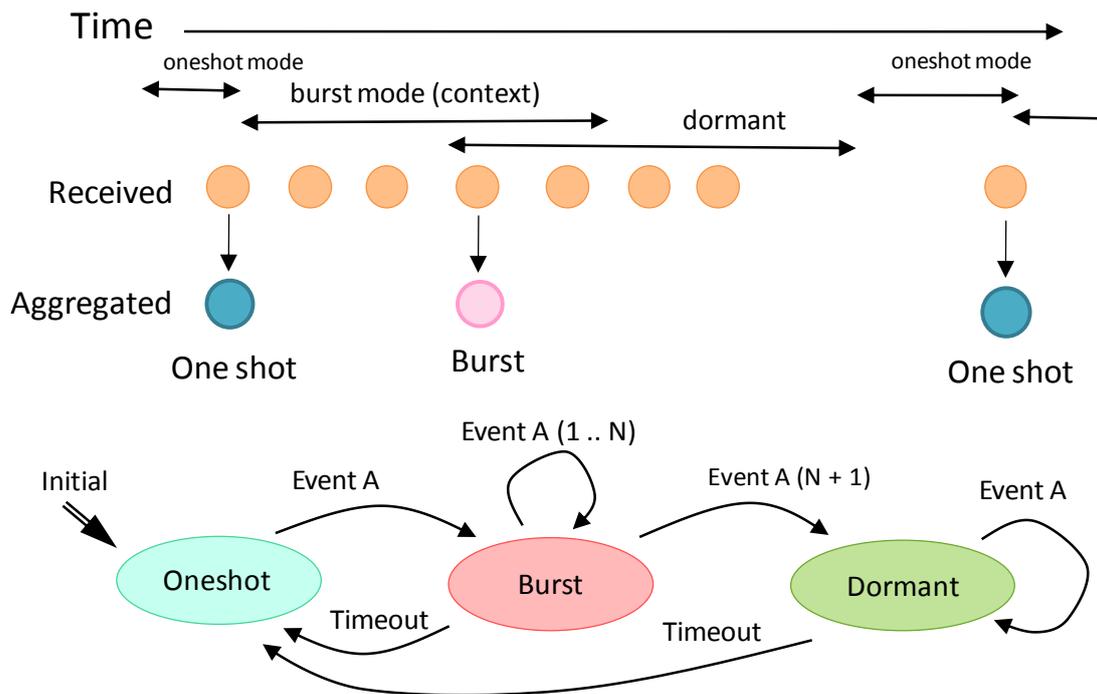


図 8. 連続した Events の処理

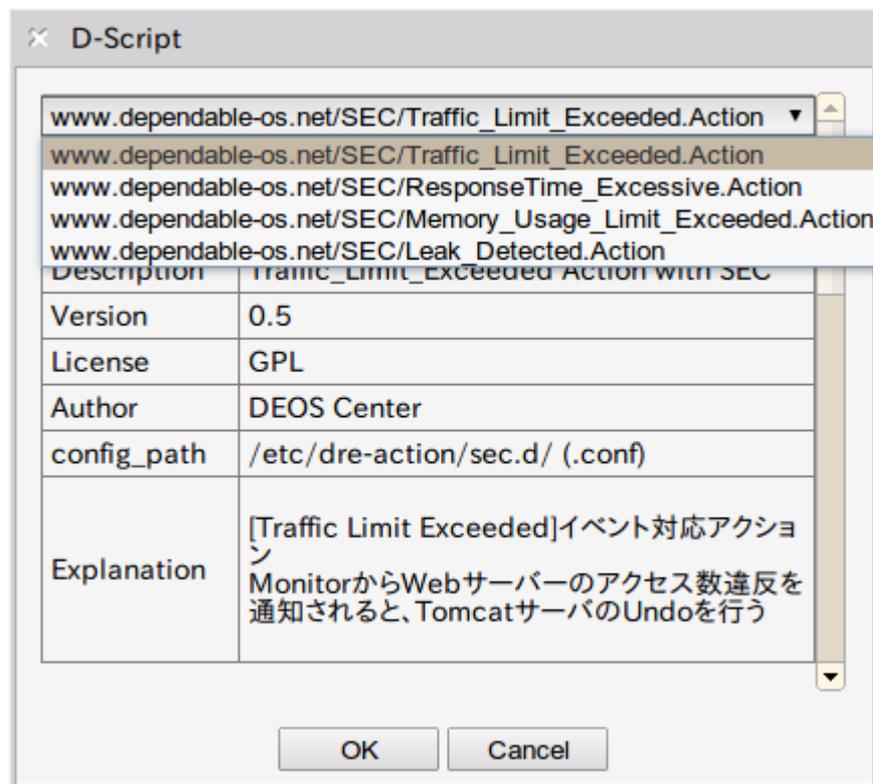
具体的には、SEC (Simple Event Correlator)を用い、D-Case の Action Nodes から SEC 用の構成ファイルを生成する。

- 1) Web サーバーへのアクセス数の増大に対して、プロモーションプログラムを停止
D-Case Node の D-Case Weaver 上での表現：

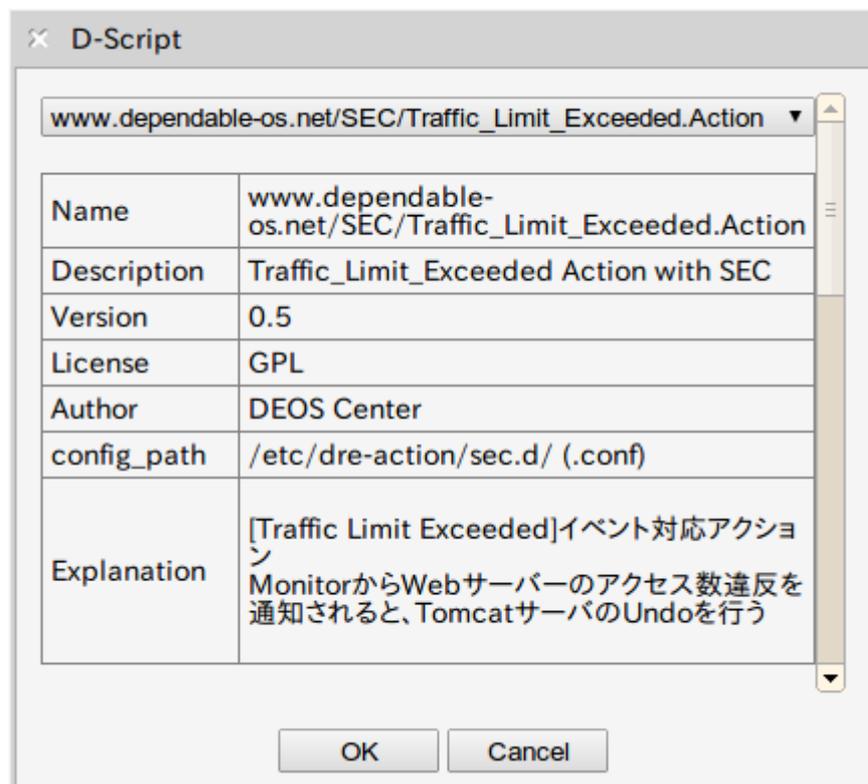
Goal: G_69 プロモーションプログラムを 停止できる

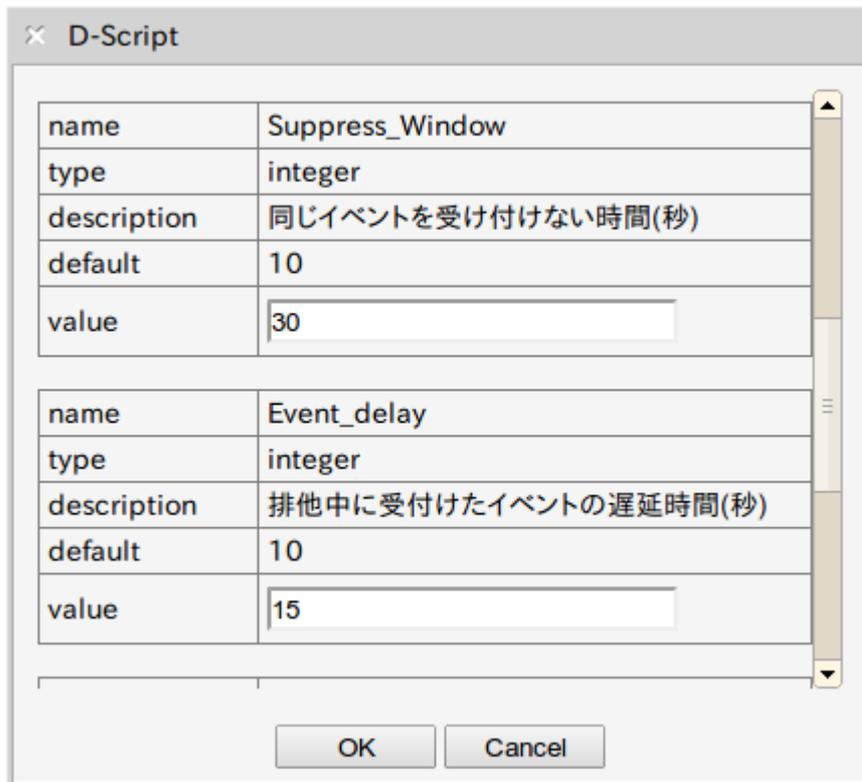
ゴール・ノード G_69 の作成後、ノード上で右クリックして、Edit D-Script を選択し、分析・モジュールの選択とパラメータ群の指定を行う。

分析・モジュールの設定を行う。



分析モジュールの種類毎に表示される必要なパラメータ群に対して、適切な値を設定する。





The screenshot shows a dialog box titled "D-Script" with a close button (X) in the top-left corner. It contains two sections, each with a table of properties and a text input field for the value.

name	Suppress_Window
type	integer
description	同じイベントを受け付けない時間(秒)
default	10
value	<input type="text" value="30"/>

name	Event_delay
type	integer
description	排他中に受付けたイベントの遅延時間(秒)
default	10
value	<input type="text" value="15"/>

At the bottom of the dialog are two buttons: "OK" and "Cancel".

この例では、D-Case 内に以下のような XML ノードが記録される。

XML:

```
<dcase:node type="Goal" id="_y1_a8MQhEeGBP79tzkVYg" name="G_69">
<dcase:description>プロモーションプログラムを停止できる</dcase:description>
<dcase:properties/>

<dcase:d-script>
<dcase:full-name>www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action</dcase:full-
name>
<dcase:values>
<dcase:value name="Suppress_Window">30</dcase:value>
<dcase:value name="Event_delay">15</dcase:value>
<dcase:value name="Action_delay">1</dcase:value>
<dcase:value name="Apply_Snapshot">ss1</dcase:value>
</dcase:values>
</dcase:d-script>
</dcase:node>
```

3.7. Action 系のコマンド例

このデモ・システムでは、SEC を用いて分析系を実装した。SEC からの Actions の呼び出しは、SEC の構成ファイルの”action”識別子の値として、D-RE コマンド列を記述されている。SEC からの呼び出した Actions のリストは以下の通りである。

Actions	意味
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh	解析結果のログを書き出す
shellcmd usr/share/dre-demo/d-script/action/act_reboot_sys_container.sh	System Container の再起動する
shellcmd /usr/share/dre-demo/d-script/action/act_undo_sys_container.sh	System Container への変更の Undo する
shellcmd /usr/share/dre-demo/d-script/action/act_kill_batch_sys.sh	バッチ・ジョブの停止する
shellcmd /usr/share/dre-demo/bin/dcase-status	D-Case の監視系（モニター）ノードの状態表示を変更する
event	SEC 内部イベントを発生する
create,delete	SEC の状態を変更する

4. おわりに

DEOS の実行による OSD の実現、すなわち「変化しつづけるシステムのサービス継続と説明責任の全う」のためには以下が必須である。

- 継続的な改良改善のための開発と運用を連携する反復的なプロセス (**DEOS プロセス**)
- DEOS プロセスを仕組みとして支える合意形成・確認手法・ツール (**D-Case**) と実行環境 (**D-RE: DEOS Runtime Environment**)
- D-Case に基づいて、OSD 実現を具体化した、スクリプト (**D-Script**) およびアプリケーション・プログラム (**D-Aware Application Program**)

本 Programming Reference では、この3番目の必須要素であるスクリプトおよびアプリケーション・プログラムを書く上での考え方 (**DEOS プログラミング**) を第2章で、デモ・システムにおける事例を第3章で紹介した。

Appendix

A1.監視系と分析系モジュール・テンプレートの例

以下に、モニター・モジュールと SEC を使った分析系モジュールの外部仕様を表した XML ファイルを記述する。

モニター・モジュール・テンプレート

1) システムの CPU 使用率とメモリ使用率の監視

```

<monitor>
  <name>www.dependable-os.net/dre/system.Monitor</name>
  <description>System Monitor</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
  <config_template><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<plug-ins>

  <plug-in name="plugins.system.Monitor"
    interval="<?= get_value('interval') ?>">
    <args host="<?= get_value('host') ?>"
      user="<?= get_value('user') ?>"
      rrd-dir="<?= get_value('rrd-dir') ?>"/>
  </plug-in>

  <plug-in name="plugins.system.Analyzer"
    interval="<?= get_value('interval') ?>">
    <args node-name="<?= get_value('d_case_node_name') ?>"
      host="<?= get_value('host') ?>"
      rrd-dir="<?= get_value('rrd-dir') ?>"
      cpu-warn="<?= get_value('cpu-warn') ?>"
      cpu-crit="<?= get_value('cpu-crit') ?>"
      mem-warn="<?= get_value('mem-warn') ?>"
      mem-crit="<?= get_value('mem-crit') ?>"
      related-node="<?= get_value('related-node') ?>"/>
  </plug-in>

  <plug-in name="plugins.system.Graph"
    interval="<?= get_value('interval') ?>">
    <args rrd-dir="<?= get_value('rrd-dir') ?>"
      graph-dir="<?= get_value('graph-dir') ?>"/>
  </plug-in>

</plug-ins>
]]>
  </config_template>
  <config_path extension=".conf">/etc/dre-monitor/conf.d</config_path>
  <explanations>
    <explanation lang="ja_JP"><![CDATA[
      システムの CPU 使用率とメモリ使用率を監視する。
      監視間隔は[interval]秒。
      [host]サーバーを監視する。
    ]]>
  </explanations>

```

```

監視した値は[rrd-dir]ディレクトリへ保存される。
CPU 使用率が[cpu-warn]%を越えた場合、WARNING メッセージを syslog へ出力する。
CPU 使用率が[cpu-crit]%を越えた場合、CRITICAL メッセージを syslog へ出力する。
Memory 使用率が[mem-warn]を越えた場合、WARNING メッセージを syslog へ出力する。
Memory 使用率が[mem-crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
[graph-dir]ディレクトリへグラフが保存される。
]]>
</explanation>
</explanations>
<parameters>
<!-- Details must be determined.  Similar to XML schema? -->
<parameter name="interval" type="integer">
  <description>監視間隔[秒]</description>
  <default>2</default>
</parameter>
<parameter name="host" type="string">
  <description>監視対象のホスト名</description>
  <default>localhost</default>
</parameter>
<parameter name="cpu-crit" type="integer" min="1" max="100">
  <description>CPU 使用率 CRITICAL メッセージの閾値[%]</description>
  <default>90</default>
</parameter>
<parameter name="cpu-warn" type="integer" min="1" max="100">
  <description>CPU 使用率 WARNING メッセージの閾値[%]</description>
  <default>80</default>
</parameter>
<parameter name="mem-crit" type="integer" min="1" max="100">
  <description>メモリ使用率 CRITICAL メッセージの閾値[%]</description>
  <default>90</default>
</parameter>
<parameter name="mem-warn" type="integer" min="1" max="100">
  <description>メモリ使用率 WARNING メッセージの閾値[%]</description>
  <default>80</default>
</parameter>
<parameter name="related-node" type="string">
  <description>関連ノード(判定機能)</description>
</parameter>
<parameter name="rrd-dir" type="string">
  <description>監視データの保存ディレクトリ</description>
  <default>/tmp/dre-monitor/rrd/localhost/system</default>
</parameter>
<parameter name="graph-dir" type="string">
  <description>グラフの保存ディレクトリ</description>
  <default>/tmp/dre-monitor/graph/localhost/system</default>
</parameter>
</parameters>
</monitor>

```

2) コンテナの CPU 使用率とメモリ使用率の監視

```
<monitor>
```

```

<name>www.dependable-os.net/dre/container.Monitor</name>
<description>Container Monitor</description>
<version>0.5</version>
<license>GPL</license>
<author>DEOS Center</author>
<config_template><![CDATA[<<?>?xml version="1.0" encoding="utf-8"?>
<plug-ins>

<plug-in name="plugins.container.Monitor"
  interval="<?> get_value('interval') ?>">
  <args container="<?> get_value('container') ?>"
    rrd-dir="<?> get_value('rrd-dir') ?>">
</plug-in>

<plug-in name="plugins.container.Analyzer"
  interval="<?> get_value('interval') ?>">
  <args node-name="<?> get_value('d_case_node_name') ?>"
    container="<?> get_value('container') ?>"
    rrd-dir="<?> get_value('rrd-dir') ?>"
    cpu-warn="<?> get_value('cpu-warn') ?>"
    cpu-crit="<?> get_value('cpu-crit') ?>"
    mem-warn="<?> get_value('mem-warn') ?>"
    mem-crit="<?> get_value('mem-crit') ?>"
    related-node="<?> get_value('related-node') ?>">
</plug-in>

<plug-in name="plugins.container.Graph"
  interval="<?> get_value('interval') ?>">
  <args rrd-dir="<?> get_value('rrd-dir') ?>"
    graph-dir="<?> get_value('graph-dir') ?>">
</plug-in>
</plug-ins>
]]>
</config_template>
<config_path extension=".conf">/etc/dre-monitor/conf.d</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    コンテナの CPU 使用率とメモリ使用率を監視する。
    監視間隔は[interval]。
    [container]コンテナを監視する。
    監視した値は[rrd-dir]ディレクトリへ保存される。
    CPU 使用率が[cpu-warn]を越えた場合、WARNING メッセージを syslog へ出力する。
    CPU 使用率が[cpu-crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
    Memory 使用率が[mem-warn]を越えた場合、WARNING メッセージを syslog へ出力する。
    Memory 使用率が[mem-crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
    [graph-dir]ディレクトリへグラフが保存される。
  ]]>
</explanation>
</explanations>
<parameters>
  <parameter name="interval" type="integer">
    <description>監視間隔[秒]</description>
    <default>2</default>
  </parameter>

```

```

<parameter name="container" type="string">
  <description>監視対象のコンテナ名</description>
  <default>container</default>
</parameter>
<parameter name="cpu-crit" type="integer" min="1" max="100">
  <description>CPU 使用率 CRITICAL メッセージの閾値[%]</description>
  <default>90</default>
</parameter>
<parameter name="cpu-warn" type="integer" min="1" max="100">
  <description>CPU 使用率 WARNING メッセージの閾値[%]</description>
  <default>80</default>
</parameter>
<parameter name="mem-crit" type="integer" min="1" max="100">
  <description>メモリ使用率 CRITICAL メッセージの閾値[%]</description>
  <default>90</default>
</parameter>
<parameter name="mem-warn" type="integer" min="1" max="100">
  <description>メモリ使用率 WARNING メッセージの閾値[%]</description>
  <default>80</default>
</parameter>
<parameter name="related-node" type="string">
  <description>関連ノード(判定機能)</description>
</parameter>
<parameter name="rrd-dir" type="string">
  <description>監視データの保存ディレクトリ</description>
  <default>/tmp/dre-monitor/rrd/localhost/container</default>
</parameter>
<parameter name="graph-dir" type="string">
  <description>グラフの保存ディレクトリ</description>
  <default>/tmp/dre-monitor/graph/localhost/container</default>
</parameter>
</parameters>
</monitor>

```

3) Apache サーバーのアクセス数の監視

```

<monitor>
  <name>www.dependable-os.net/dre/apache.Monitor</name>
  <description>Apache Server Access Count Monitor</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
  <config_template><![CDATA[<<?>?xml version="1.0" encoding="utf-8"?>
<plug-ins>

<plug-in name="plugins.apache.Monitor"
  interval="<?= get_value('interval') ?>">
  <args host="<?= get_value('host') ?>"
    rrd-dir="<?= get_value('rrd-dir') ?>"/>
</plug-in>

<plug-in name="plugins.apache.Analyzer"
  interval="<?= get_value('interval') ?>">

```

```
<args node-name="<?= get_value ('d_case_node_name') ?>"
  host="<?= get_value('host') ?>"
  rrd-dir="<?= get_value('rrd-dir') ?>"
  warn="<?= get_value('warn') ?>"
  crit="<?= get_value('crit') ?>"
  related-node1="<?= get_value('related-node1') ?>"
  related-node2="<?= get_value('related-node2') ?>"/>
</plug-in>

<plug-in name="plugins.apache.Graph"
  interval="<?= get_value('interval') ?>">
  <args rrd-dir="<?= get_value('rrd-dir') ?>"
    graph-dir="<?= get_value('graph-dir') ?>"/>
</plug-in>

</plug-ins>
]]>
</config_template>
<config_path extension=".conf">/etc/dre-monitor/conf.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    Apache サーバーのアクセス数を監視する。
    監視間隔は[interval]。
    [host]サーバーを監視する。
    監視した値は[rrd-dir]ディレクトリへ保存される。
    アクセス数が[warn]を越えた場合、WARNING メッセージを syslog へ出力する。
    アクセス数が[crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
    [graph-dir]ディレクトリへグラフが保存される。
  ]]>
</explanation>
</explanations>
<parameters>
  <parameter name="interval" type="integer">
    <description>監視間隔[秒]</description>
    <default>2</default>
  </parameter>
  <parameter name="host" type="string">
    <description>監視対象のホスト名</description>
    <default>localhost</default>
  </parameter>
  <parameter name="crit" type="integer" min="0">
    <description>アクセス数 CRITICAL メッセージの閾値[%]</description>
    <default>2500</default>
  </parameter>
  <parameter name="warn" type="integer" min="0">
    <description>アクセス数 WARNING メッセージの閾値[%]</description>
    <default>2000</default>
  </parameter>
  <parameter name="related-node1" type="string">
    <description>関連ノード 1(判定機能)</description>
  </parameter>
  <parameter name="related-node2" type="string">
    <description>関連ノード 2(通知機能)</description>
  </parameter>
  <parameter name="rrd-dir" type="string">
```

```

    <description>監視データの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/rrd/localhost/apache</default>
  </parameter>
  <parameter name="graph-dir" type="string">
    <description>グラフの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/graph/localhost/apache</default>
  </parameter>
</parameters>
</monitor>

```

4) MySQL サーバーの応答時間の監視

```

<monitor>
  <name>www.dependable-os.net/dre/mysql.Monitor</name>
  <description>MySQL Responce Time Monitor</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
  <config_template><![CDATA[<<??>?xml version="1.0" encoding="utf-8"?>
<plug-ins>

  <plug-in name="plugins.mysql.Monitor"
    interval="<?= get_value('interval') ?>">
    <args host="<?= get_value('host') ?>"
      user="<?= get_value('user') ?>"
      passwd="<?= get_value('passwd') ?>"
      db="<?= get_value('db') ?>"
      table="<?= get_value('table') ?>"
      rrd-dir="<?= get_value('rrd-dir') ?>"/>
  </plug-in>

  <plug-in name="plugins.mysql.Analyzer"
    interval="<?= get_value('interval') ?>">
    <args node-name="<?= get_value('d_case_node_name') ?>"
      host="<?= get_value('host') ?>"
      rrd-dir="<?= get_value('rrd-dir') ?>"
      warn="<?= get_value('warn') ?>"
      crit="<?= get_value('crit') ?>"
      related-node="<?= get_value('related-node') ?>"/>
  </plug-in>

  <plug-in name="plugins.mysql.Graph"
    interval="<?= get_value('interval') ?>">
    <args rrd-dir="<?= get_value('rrd-dir') ?>"
      graph-dir="<?= get_value('graph-dir') ?>"/>
  </plug-in>
</plug-ins>
]]>
  </config_template>
  <config_path extension=".conf">/etc/dre-monitor/conf.d/</config_path>
  <explanations>
    <explanation lang="ja_JP"><![CDATA[

```

```
MySQL サーバーの応答時間を監視する。
監視間隔は[interval]。
[host]サーバーを監視する。
データベースのユーザは[user]で、パスワードは[passwd]。
データベース[db]のテーブル[table]の応答時間を監視する。
監視した値は[rrd-dir]ディレクトリへ保存される。
応答時間が[warn]を越えた場合、WARNING メッセージを syslog へ出力する。
応答時間が[crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
[graph-dir]ディレクトリへグラフが保存される。
]]>
</explanation>
</explanations>
<parameters>
  <parameter name="interval" type="integer">
    <description>監視間隔[秒]</description>
    <default>2</default>
  </parameter>
  <parameter name="host" type="string">
    <description>監視対象のホスト名</description>
    <default>localhost</default>
  </parameter>
  <parameter name="user" type="string">
    <description>ユーザ名</description>
  </parameter>
  <parameter name="passwd" type="password">
    <description>パスワード</description>
  </parameter>
  <parameter name="db" type="string">
    <description>データベース名</description>
  </parameter>
  <parameter name="table" type="string">
    <description>テーブル名</description>
  </parameter>
  <parameter name="crit" type="integer" min="0" max="1">
    <description>アクセス数 CRITICAL メッセージの閾値[%]</description>
    <default>1</default>
  </parameter>
  <parameter name="warn" type="integer" min="0" max="1">
    <description>アクセス数 WARNING メッセージの閾値[%]</description>
    <default>0.8</default>
  </parameter>
  <parameter name="related-node" type="string">
    <description>関連ノード(判定機能)</description>
  </parameter>
  <parameter name="rrd-dir" type="string">
    <description>監視データの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/rrd/localhost/mysql</default>
  </parameter>
  <parameter name="graph-dir" type="string">
    <description>グラフの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/graph/localhost/mysql</default>
  </parameter>
</parameters>
</monitor>
```

5) メモリ診断モジュール診断結果の監視

```

<monitor>
  <name>www.dependable-os.net/dre/memleak.Monitor</name>
  <description>Memory Leak Monitor</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
  <config_template><![CDATA[<?xml version="1.0" encoding="utf-8"?>
<plug-ins>

<plug-in name="plugins.memleak.Monitor"
  interval="<?= get_value('interval') ?>">
  <args host="<?= get_value('host') ?>"
    rrd-dir="<?= get_value('rrd-dir') ?>"/>
</plug-in>

<plug-in name="plugins.memleak.Analyzer"
  interval="<?= get_value('interval') ?>">
  <args node-name="<?= get_value ('d_case_node_name') ?>"
    host="<?= get_value('host') ?>"
    rrd-dir="<?= get_value('rrd-dir') ?>"
    warn="<?= get_value('warn') ?>"
    crit="<?= get_value('crit') ?>"
    related-node="<?= get_value('related-node') ?>"/>
</plug-in>

<plug-in name="plugins.memleak.Graph"
  interval="<?= get_value('interval') ?>">
  <args rrd-dir="<?= get_value('rrd-dir') ?>"
    graph-dir="<?= get_value('graph-dir') ?>"/>
</plug-in>

</plug-ins>
]]>
  </config_template>
  <config_path extension=".conf">/etc/dre-monitor/conf.d/</config_path>
  <explanations>
    <explanation lang="ja_JP"><![CDATA[
      メモリ診断モジュール診断結果を監視する。
      監視間隔は[interval]。
      [host]サーバーを監視する。
      監視した値は[rrd-dir]ディレクトリへ保存される。
      診断値が[warn]を越えた場合、WARNING メッセージを syslog へ出力する。
      診断値が[crit]を越えた場合、CRITICAL メッセージを syslog へ出力する。
      [graph-dir]ディレクトリへグラフが保存される。
    ]]>
  </explanation>
</explanations>
<parameters>
  <parameter name="interval" type="integer">
    <description>監視間隔[秒]</description>

```

```

    <default>2</default>
  </parameter>
  <parameter name="host" type="string">
    <description>監視対象のホスト名</description>
    <default>localhost</default>
  </parameter>
  <parameter name="crit" type="integer" min="0" max="1">
    <description>CRITICAL メッセージの閾値</description>
    <default>200</default>
  </parameter>
  <parameter name="warn" type="integer" min="0" max="1">
    <description>WARNING メッセージの閾値</description>
    <default>150</default>
  </parameter>
  <parameter name="related-node" type="string">
    <description>関連ノード(判定機能)</description>
  </parameter>
  <parameter name="rrd-dir" type="string">
    <description>監視データの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/rrd/localhost/mysql</default>
  </parameter>
  <parameter name="graph-dir" type="string">
    <description>グラフの保存ディレクトリ</description>
    <default>/tmp/dre-monitor/graph/localhost/mysql</default>
  </parameter>
</parameters>
</monitor>

```

SEC を使った分析系モジュール・テンプレート

1) SEC (Simple Event Correlator)の構成ファイル生成用汎用モジュール・テンプレート

```

<action>
  <name>www.dependable-os.net/SEC/generic.Action</name>
  <description>Generic Action with SEC</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
  <config_template><![CDATA[#
# Generic SEC configuration file generated at <?= date (DateTime::RFC3339) ?>
# For D-Case Node (<?= get_value ('d_case_node_id') ?> | <?= get_value ('d_case_node_name') ?> |
<?= get_value ('d_case_node_id_by_editor') ?>) with D-Script (<?= get_value ('d_script_id') ?>)
  <?= get_value ('config_file_content') ?>
]]>
</config_template>
<config_path extension=".conf">/etc/dre-action/sec.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    Simple Event Correlator (SEC)の構成ファイルの中身を指定する。
    詳細は、sec の man page や Simple Event Correlator FAQ を参照。
  ]]>
</explanation>

```

```

</explanations>
<parameters>
  <parameter name="config_file_content" type="string">
    <description>Content of the entire configuration file</description>
  </parameter>
</parameters>
</action>

```

2) Apache のアクセス数超過にたいしてアクセスの制限

```

<action>
  <name>www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action</name>
  <description>Traffic_Limit_Exceeded Action with SEC</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>
<config_template><![CDATA[#####
#####
# Receives an external event [Traffic_Limit_Exceeded]
#####
type=SingleWithSuppress
ptype=RegExp
pattern=sys-apa TrafficLimitExceeded (¥S+)
desc=[SEC Action]
action=shelcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.err SEC
"LogFormat=DRE1.0>Event=TrafficLimitExceeded>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/SysContainer:sys-apa/apache>¥
EventDiscoverer=/Host:deosc-laptop/Monitor>Message=Traffic Limit Exceeded$1>";
window=<?= get_value ('Suppress_Window') ?>

type=SingleWithSuppress
ptype=RegExp
pattern=Start TrafficLimitExceeded
desc=[SEC Action]
action=event <?= get_value ('Action_delay') ?> "Traffic_Limit_Exceeded_process1";
window=<?= get_value ('Suppress_Window') ?>

type=Single
ptype=RegExp
pattern=(¥S+)Message=Installed new-service.deb(¥S+)
context=! NEW_SERVICE_INSTALLED
desc=[SEC Action]
action=create NEW_SERVICE_INSTALLED

#####
# Receives an internal event [Traffic_Limit_Exceeded]
#####
type=Single
ptype=RegExp
continue=TakeNext
pattern=Traffic_Limit_Exceeded_process1
context=PROCESSING && NEW_SERVICE_INSTALLED
desc=[SEC Action]

```

```

action=event <?= get_value ('Event_delay') ?> $0

type=Single
ptype=RegExp
continue=DontCont
pattern=Traffic_Limit_Exceeded_process1
context=(! PROCESSING) && NEW_SERVICE_INSTALLED
desc=[SEC Action]
action=create PROCESSING; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionStarted>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/usr/share/dre-demo/d-script/action/
act_undo_sys_container.sh>¥
Message=Undo Container : sys-tom. [TrafficLimitExceeded] Recovery Action>; ¥
event <?= get_value ('Action_delay') ?> "Traffic_Limit_Exceeded_process2";shellcmd
/usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s running && ¥
  /usr/share/dre-demo/bin/dcase-status -n G_27 -s running && /usr/share/dre-demo/bin/dcase-status
-n G_27 -s normal;

type=SingleWithScript
ptype=RegExp
pattern=Traffic_Limit_Exceeded_process2
script=/bin/bash /usr/share/dre-demo/d-script/action/act_undo_sys_container.sh sys-tom <?=
get_value ('Apply_Snapshot') ?> :2 >> /var/log/sec.log 2>&1
desc=[SEC Action]
action=delete PROCESSING; delete NEW_SERVICE_INSTALLED; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionCompleted>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/usr/share/dre-demo/d-script/action/
act_undo_sys_container.sh>¥
Message=Undo Container : sys-tom. [TrafficLimitExceeded] Recovery Action>;shellcmd
/usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s normal;
action2=shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionFailed>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/usr/share/dre-demo/d-script/action/
act_undo_sys_container.sh>¥
Message=Undo Container : sys-tom. [TrafficLimitExceeded] Recovery Action>;
]]>
</config_template>
<config_path extension=".conf">/etc/dre-action/sec.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    [Traffic Limit Exceeded] イベント対応アクション
    モニターから Web サーバーのアクセス数違反を通知されると、Tomcat サーバーの Undo を
    行う
  ]>
</explanation>
</explanations>
<parameters>
  <!-- Details must be determined. Similar to XML schema? -->
  <parameter name="Suppress_Window" type="integer">
    <description>同じイベントを受け付けない時間(秒)</description>
    <default>10</default>
  </parameter>
  <parameter name="Event_delay" type="integer" min="1" max="100">

```

```

    <description>排他中に受付けたイベントの遅延時間(秒)</description>
    <default>10</default>
  </parameter>
  <parameter name="Action_delay" type="integer" min="1" max="100">
    <description>アクション実行までの遅延時間(秒)</description>
    <default>0</default>
  </parameter>
  <parameter name="Apply_Snapshot" type="string">
    <description>Undo で適用するスナップショット</description>
    <default>ss1</default>
  </parameter>
</parameters>
</action>

```

3) MySQL からの応答時間の超過に対して MySQL Batch Job のアポート

```

<action>
  <name>www.dependable-os.net/SEC/ResponseTime_Excessive.Action</name>
  <description>ResponseTime_Excessive Action with SEC</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>

<config_template><![CDATA[#####
#####
# Receives an external event [ResponseTime_Excessive]
#####
type=SingleWithSuppress
ptype=RegExp
pattern=sys-mysql ResponseTimeExcessive (¥S+)
desc=[SEC Action]
action=shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.err SEC
"LogFormat=DRE1.0>Event=ResponseTimeExcessive>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/SysContainer:sys-mysql/mysql>¥
EventDiscoverer=/Host:deosc-laptop/Monitor>Message=ResponseTime Excessive$1>";
window=<? get_value ('Suppress_Window') ?>

type=SingleWithSuppress
ptype=RegExp
pattern=Start ResponseTimeExcessive
desc=[SEC Action]
action=event <? get_value ('Action_delay') ?> "ResponseTime_Excessive_process1";
window=<? get_value ('Suppress_Window') ?>

#####
# Receives an internal event [ResponseTime_Excessive]
#####
type=Single
ptype=RegExp
continue=TakeNext
pattern=ResponseTime_Excessive_process1
context=PROCESSING
desc=[SEC Action]

```

```

action=event <?= get_value ('Event_delay') ?> $0

type=Single
ptype=RegExp
continue=DontCont
pattern=ResponseTime_Excessive_process1
context=! PROCESSING
desc=[SEC Action]
action=create PROCESSING; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionStarted>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_kill_batch_sys.sh>¥
Message=Kill BatchJob : sys-mysql. [ResponseTimeExcessive] Recovery Action>; ¥
event <?= get_value ('Action_delay') ?> "ResponseTime_Excessive_process2";shellcmd
/usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s running;

type=SingleWithScript
ptype=RegExp
pattern=ResponseTime_Excessive_process2
script=/bin/bash /usr/share/dre-demo/d-script/action/act_kill_batch_sys.sh <?= get_value
('Kill_Param') ?> >> /var/log/sec.log 2>&1
desc=[SEC Action]
action=delete PROCESSING; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionCompleted>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_kill_batch_sys.sh>¥
Message=Kill BatchJob : sys-mysql. [ResponseTimeExcessive] Recovery Action>; ¥
shellcmd /usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s normal;
action2=shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionFailed>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_kill_batch_sys.sh>¥
Message=Kill BatchJob : sys-mysql. [ResponseTimeExcessive] Recovery Action>;
]]>
</config_template>
<config_path extension=".conf">/etc/dre-action/sec.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    [ResponseTime_Excessive]イベント対応アクション
    モニターから DB サーバーの応答時間違反を通知されると、DB サーバプログラムを停止を
    行う
  ]>
]]>
</explanation>
</explanations>
<parameters>
  <!-- Details must be determined. Similar to XML schema? -->
  <parameter name="Suppress_Window" type="integer">
    <description>同じイベントを受け付けない時間(秒)</description>
    <default>10</default>
  </parameter>
  <parameter name="Event_delay" type="integer" min="1" max="100">
    <description>排他中に受付けたイベントの遅延時間(秒)</description>
    <default>10</default>

```

```

    </parameter>
    <parameter name="Action_delay" type="integer" min="1" max="100">
      <description>アクション実行までの遅延時間(秒)</description>
      <default>0</default>
    </parameter>
    <parameter name="Kill_Param" type="string">
      <description>act_kill_batch.sh へのパラメータ</description>
      <default>sql</default>
    </parameter>
  </parameters>
</action>

```

4) メモリ使用量超過に対するの緊急対応 (システム・コンテナの再起動)

```

<action>
  <name>www.dependable-os.net/SEC/Memory_Usage_Limit_Exceeded.Action</name>
  <description>Memory_Usage_Limit_Exceeded Action with SEC</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>

<config_template><![CDATA[#####
#####
# Receives an external event [Memory_Usage_Limit_Exceeded]
#####
type=SingleWithSuppress
ptype=RegExp
pattern=sys-tom MemoryUsageLimitExceeded (¥S+)
desc=[SEC Action]
action=shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.err SEC
"LogFormat=DRE1.0>Event=MemoryUsageLimitExceeded>¥
TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/SysContainer:sys-tom/tomcat>¥
EventDiscoverer=/Host:deosc-laptop/Monitor>Message=Memory Usage Limit Exceeded$1>";
window=<?= get_value ('Suppress_Window') ?>

type=SingleWithSuppress
ptype=RegExp
pattern=Start MemoryUsageLimitExceeded
desc=[SEC Action]
action=event <?= get_value ('Action_delay') ?> "Memory_Usage_Limit_Exceeded1";
window=<?= get_value ('Suppress_Window') ?>

type=Single
ptype=RegExp
pattern=(¥S+)Message=Installed new-service(¥S+)
context=! NEW_SERVICE_INSTALLED
desc=[SEC Action]
action=create NEW_SERVICE_INSTALLED

#####
# Receives an internal event [Memory_Usage_Limit_Exceeded]
#####
type=Single

```

```

ptype=RegExp
continue=TakeNext
pattern=Memory_Usage_Limit_Exceeded1
context=PROCESSING && NEW_SERVICE_INSTALLED
desc=[SEC Action]
action=event <?= get_value ('Event_delay') ?> $0

type=Single
ptype=RegExp
continue=DontCont
pattern=Memory_Usage_Limit_Exceeded1
context=(! PROCESSING) && NEW_SERVICE_INSTALLED
desc=[SEC Action]
action=create PROCESSING; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionStarted>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_reboot_sys_container.sh>¥
Message=Reboot Container : sys-tom. [MemoryUsageLimitExceeded] Recovery Action>; ¥
event <?= get_value ('Action_delay') ?> "Memory_Usage_Limit_Exceeded2";shellcmd
/usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s running;

type=SingleWithScript
ptype=RegExp
pattern=Memory_Usage_Limit_Exceeded2
script=/bin/bash /usr/share/dre-demo/d-script/action/act_reboot_sys_container.sh sys-tom sys-apa >>
/var/log/sec.log 2>&1
desc=[SEC Action]
action=delete PROCESSING; delete NEW_SERVICE_INSTALLED; ¥
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionCompleted>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_reboot_sys_container.sh>¥
Message=Reboot Container : sys-tom. [MemoryUsageLimitExceeded] Recovery Action>; ¥
shellcmd /usr/share/dre-demo/bin/dcase-status -n <?= get_value ('d_case_node_name') ?> -s normal;
action2=shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh local5.info SEC
"LogFormat=DRE1.0>Event=RecoveryActionFailed>¥
TagSource=/Host:deos-laptop>EventSource=/Host:deos-laptop/usr/share/dre-demo/d-script/action/
act_reboot_sys_container.sh>¥
Message=Reboot Container : sys-tom. [MemoryUsageLimitExceeded] Recovery Action>;
]]>
</config_template>
<config_path extension=".conf">/etc/dre-action/sec.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    [Memory_Usage_Limit_Exceeded] イベント対応アクション
    モニターから Application サーバーのメモリ使用量違反を通知されると、
    Web/Application サーバーの再起動を行う
  ]>
</explanation>
</explanations>
<parameters>
  <!-- Details must be determined. Similar to XML schema? -->
  <parameter name="Suppress_Window" type="integer">
    <description>同じイベントを受け付けない時間(秒)</description>

```

```

    <default>10</default>
  </parameter>
  <parameter name="Event_delay" type="integer" min="1" max="100">
    <description>排他中に受付けたイベントの遅延時間(秒)</description>
    <default>10</default>
  </parameter>
  <parameter name="Action_delay" type="integer" min="1" max="100">
<description>アクション実行までの遅延時間(秒)</description>
    <default>0</default>
  </parameter>
</parameters>
</action>

```

5) メモリ・リーク時の診断モジュールの導入

```

<action>
  <name>www.dependable-os.net/SEC/Leak_Detected.Action</name>
  <description>Leak_Detected Action with SEC</description>
  <version>0.5</version>
  <license>GPL</license>
  <author>DEOS Center</author>

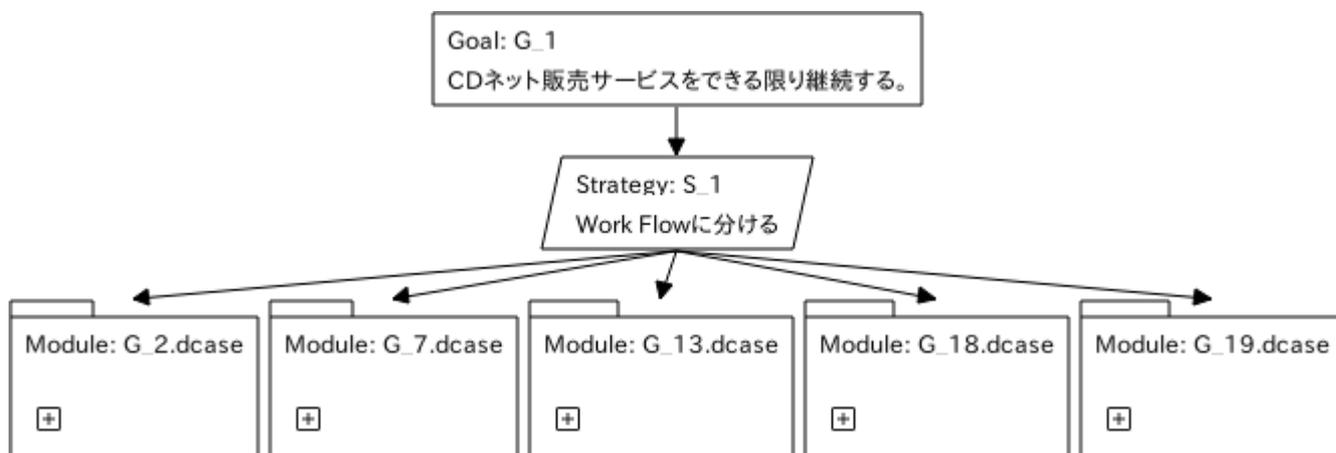
<config_template><![CDATA[#####
#####
# Receives an external event [Leak_Detected]
#####
type=SingleWithSuppress
ptype=RegExp
pattern=sys-tom LeakDetected (¥S+)
desc=[SEC Action]
action=delete DIAGNOSYS_INSTALLED; shellcmd
/usr/share/dre-demo/d-script/action/act_logging.sh local5.err SEC ¥
"LogFormat=DRE1.0>Event=LeakDetected>TagSource=/Host:deosc-laptop>EventSource=/Host:deosc-laptop/SysContainer:sys-tom/tomcat>¥
EventDiscoverer=/Host:deosc-laptop/Monitor>Message=Leak Detected$1>";¥
shellcmd /usr/share/dre-demo/bin/dcbase-status -n G_34 -s normal; shellcmd
/home/dre/work/ET2012/demo_3/demo3end.sh;
window=<?%= get_value ('Suppress_Window') ?>
type=Single
ptype=RegExp
pattern=(¥S+)Message=Installed diagnosys.deb(¥S+)
context=! DIAGNOSYS_INSTALLED
desc=[SEC Action]
action=create DIAGNOSYS_INSTALLED
  ]>
</config_template>
<config_path extension=".conf"/>/etc/dre-action/sec.d/</config_path>
<explanations>
  <explanation lang="ja_JP"><![CDATA[
    [Leak_Detected] イベント対応アクション
    モニターからメモリ・リークの可能性のあるアプリ検出を通知されると、対象アプリをログ
    に書込む
  ]>
]]>

```

```
</explanation>
</explanations>
<parameters>
  <!-- Details must be determined. Similar to XML schema? -->
  <parameter name="Suppress_Window" type="integer">
    <description>同じイベントを受け付けない時間(秒)</description>
    <default>10</default>
  </parameter>
</parameters>
</action>
</actions>
```

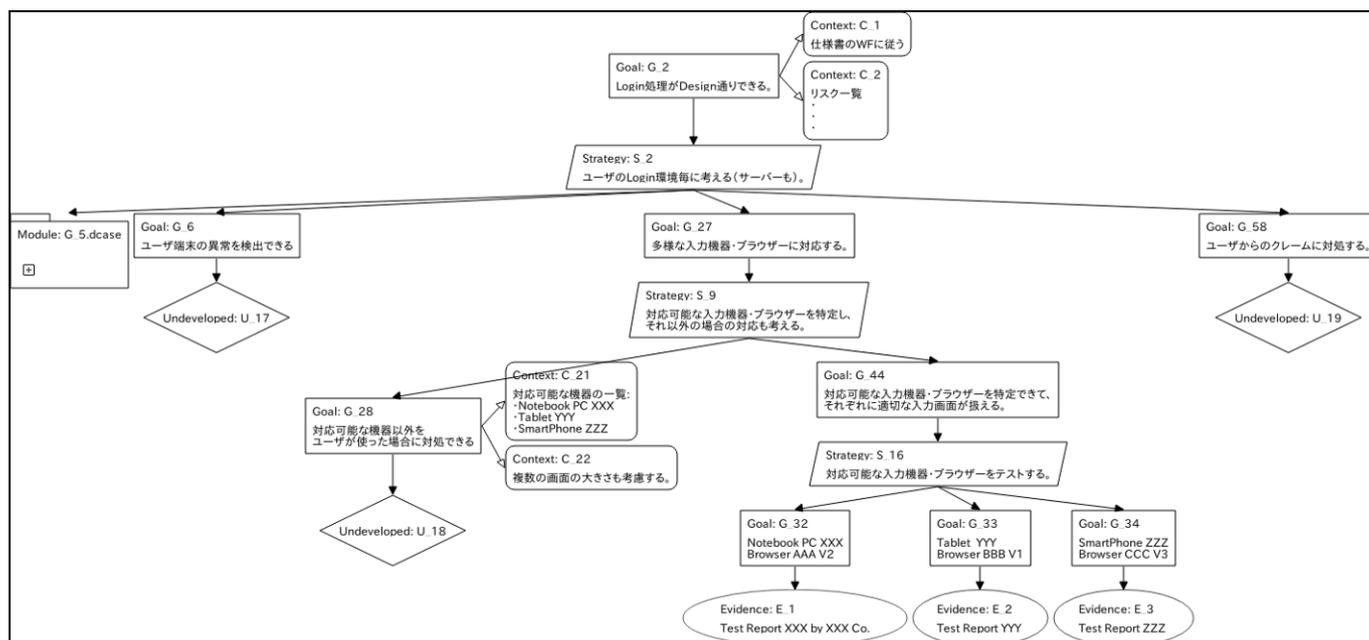
A2. デモ・システムの D-Case

D-Case の全体図

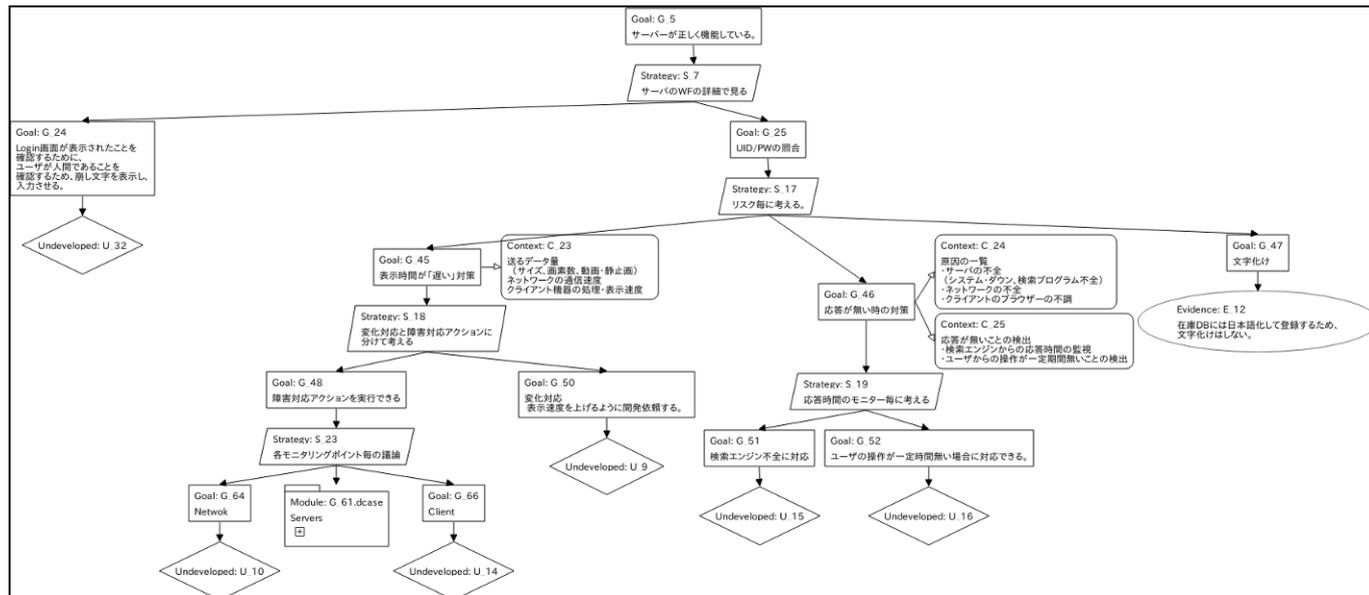


D-Case モジュール図 (部分木図)

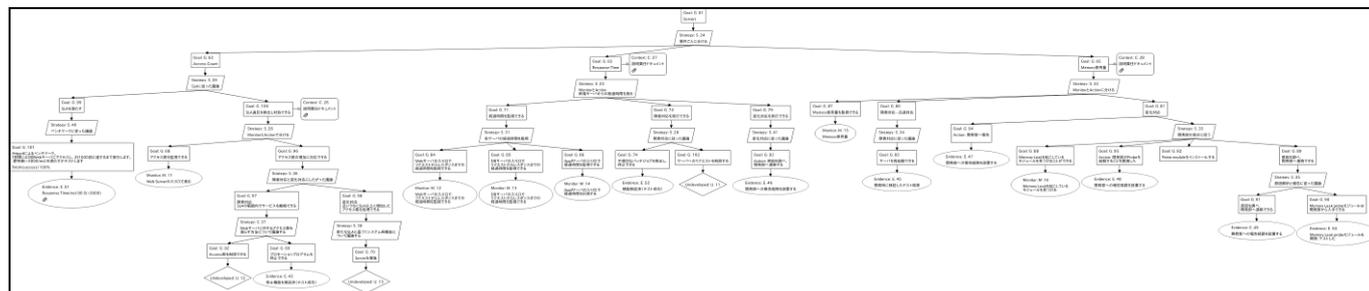
Module: G_2.dcase



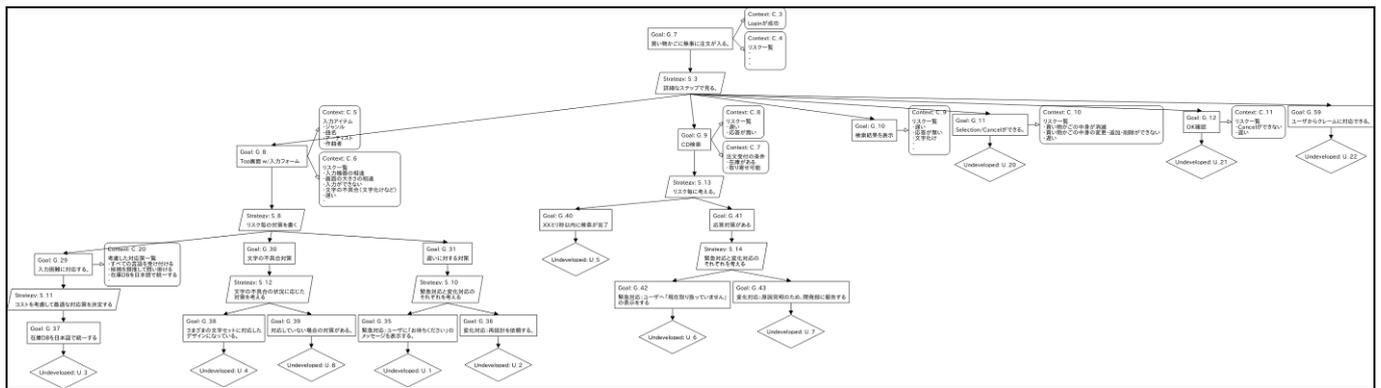
Module: G5.dcase



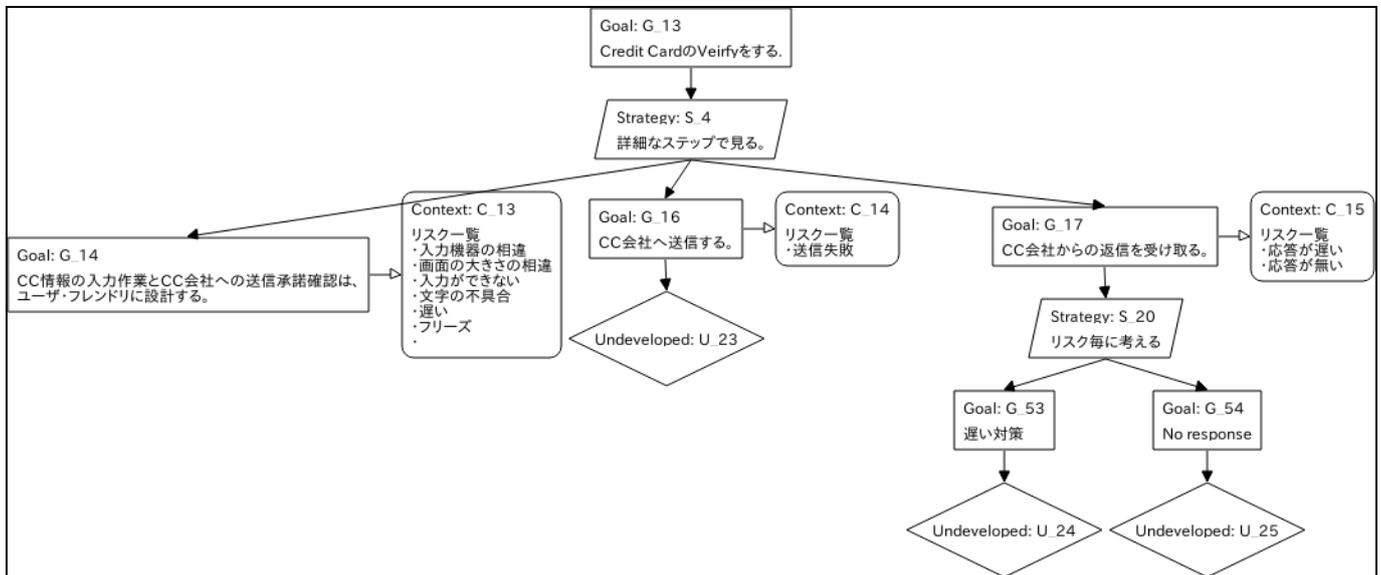
Module: G_61.dcase



Module: G_7.dcase



Module: G_13.dcase





DEOS プロジェクト