DEOS-FY2013-PR-01E

DEOS Programming Reference

Version E1.0

2013/07/15

Edited by DEOS R&D Center

DEOS Project

JST-CREST

Research Area

“Dependable Operating Systems for Embedded Systems Aiming at Practical Applications”

Japan Science and Technology Agency

Contents

The following DEOS project document is recommended:
DEOS-FY2012-PU-01J : DEOS Project Update 2012 (in Japanese)

Also, the DEOS home page and other DEOS project documents must be referred to.
DEOS Home Page URL : **http://www.dependable-os.net/**
DEOS-FY2013-RE-01E : D-RE Specification, with D-RE Installation Guide
DEOS-FY2013-EA-01E : D-RE APIs Specification
DEOS-FY2013-EC-01E : D-RE Commands Specification
DEOS-FY2013-SP-01E : D-RE API Sample Program Guide

1. Introduction

Today's large systems are

- more complex, larger
- connected with communication networks
- mixtures of independent subsystems with
    - functions that are known incompletely like black boxes
    - inadequate expansion of systems
    - much dependency on legacy codes

It is very hard to grasp the whole picture of these systems, to understand them in detail, and to build upon them correctly. In short, the systems are always **incomplete**.

Moreover,

- requirements from service users are always changing
- rules or mechanisms of society are always changing
- objectives of business are always changing
- service requirements and specifications are diversified
- specifications of other systems which are connected via communication networks are changing independently

There is always interaction with outer world of systems. In short, the operating environments of the systems are always **uncertain**.

In other words, today's large systems are **Open Systems** whose functions, structures, and boundaries are always changing, and always will have **incompleteness** and **uncertainty** which may result in failures in the future (**factors in open systems failure**).

We define **Open Systems Dependability (OSD)** as follows: the ability of a system to continuously prevent these problem factors from causing failure, to take appropriate and quick action when failures occur, to minimize damage, to provide the services expected by users as safely and continuously as possible, and to maintain accountability for the system operations and processes.

And, we call knowledge and technologies which realize OSD for society, such as process definitions, tools, programming techniques, and environments for development and operations, **Dependability Engineering for Open Systems (DEOS)** .

Realization of OSD, in other words, Service Continuity and Accountability for ever-changing systems, through DEOS is achieved by the following:

- An iterative process (**DEOS Process**) for continuous improvement by collaboration of development and operations
- Mechanisms which support the DEOS Process such as tools and methodology (**D-Case**) for consensus building and confirmation, and execution environments (**D-RE : DEOS Runtime Environment**)
- **D-Script** and **D-Aware Application Program**, which realize OSD based on D-Case

This Programming Reference describes **DEOS Programming** by providing hints for **D-Script** and **D-Aware Application Programs** in Chapter 2, and programming examples in Chapter 3. Reading the following documents is recommended:

DEOS-FY2012-PU-01J : DEOS Project Update 2012 (in Japanese)
ISBN 978-1-46657-751-0 "Open Systems Dependability" (CRC Press)

2. Concepts of DEOS Programming

2.1. DEOS Programming Objectives

As described in former chapter, failures of today's large systems are **inevitable**. **DEOS Process** requires that the following descriptions be written in **D-Case**.

A) **ordinary operations**:

- 1) monitoring to get important information for judgment (Monitoring)
- 2) keeping adequately logged information (Logging)

B) Failure Preventive **actions**

- 1) **detecting signals of failure** as soon as possible
  - 1-1) in case a failure signal is detected, **preventive action** (including switching to back-up system, or partial reset) must be taken as soon as possible
  - 1-2) in case preventive action cannot be taken, the possibility of **scaled-down operation** (including partial function isolation or maintenance of performance quota) must be examined
    - 1-2-1) if possible, do the scaled-down operation
  - 1-3) the fact that these failure preventive actions were taken must be reported to the responsible department or person

C) Failure Responsive **actions**

- 1) **detecting failure** as soon as possible
  - 1-1) switching to the back-up system or partial reset must be executed as soon as possible
  - 1-2) if impossible, the possibility of **scaled-down operation** (including failed function isolation or maintenance of performance quota) must be examined
    - 1-2-1) if possible, do scaled-down operation
    - 1-2-2) if impossible, the possibility of **system reset** must be examined
      - 1-2-2-1) if this is possible, the possibility of **quick reset** must be examined
        - 1-2-2-1-1) if possible, do quick reset
        - 1-2-2-1-2) if impossible, do normal reset
  - 1-3) the fact that these failure responsive actions were taken must be reported to the responsible department or person

To achieve the goals described above, the System Designer, the System Programmer, the Application Designer, and the Application Programmer must always strive to carry out the following 6 steps:

Monitoring, Sensing
Logging
Reporting, Notification, Signaling
Analysis
Diagnosis
Actions

Considering those steps, we call the programming scheme which supports the realization of the DEOS Process **DEOS Programming**. This scheme consists of not only software solutions but also hardware solutions such as a redundant server system or network, or human solutions including operators and programmers.

- ✓ implement necessary Monitoring, Sensing, Logging, Reporting, Notification, Signaling as much as possible

- ✓ implement necessary Analysis, Diagnosis as much as possible
- ✓ implement necessary Actions as much as possible

To support Actions, the following system functions are required:

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

Figure 1 shows the relation between steps and functions:

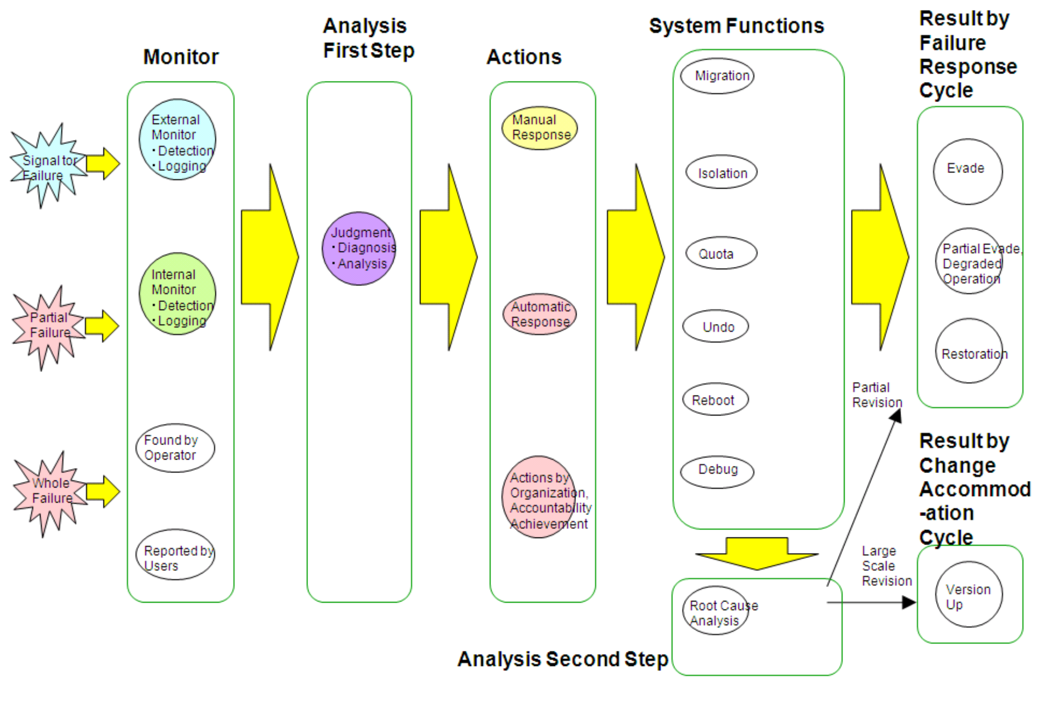


Figure 1. Steps and Functions in DEOS Process

In summary, we can say that DEOS Programming
＝Programming based on D-Case
＝Programming for D-Script and D-Aware Applications which run on D-RE
＝Programming which drives the DEOS Process efficiently

2.2. DEOS Programming Overview

What should be monitored is argued in the consensus building process, and is described in D-Case. Actions taken based on the diagnosis made based on monitoring results are also described in D-Case.

There are many Script Languages, both marketed software and OSS . In the DEOS Project, we have been developing **D-Script,** aiming for light weight, small size, and high performance (refer to the following document).

- D-Script - Realizing Failure Response by Script (Doc # : DEOS-FY2012-DS-01JA) (in Japanese)

Monitoring, Logging, and Reporting might be implemented as below:

- In the case where Application task contents (conditions) are monitored in the Application program, Monitoring Points are set in the Application program
- In the case where Application movement is monitored from outside the Application program, Monitoring is done via D-Application Monitor
- In the case where System or Container movement is monitored in the System program, Monitoring Points are set in the System program
- In the case where System or Container movement is monitored from outside the System program, Monitoring is done via D-System Monitor

Analysis, Diagnosis, and Actions might be implemented as below:

- In the case where Application task contents (conditions) are analyzed and actions are taken in the Application program, Diagnosis and Actions are programmed in the Application program
- In the case where Application movement is analyzed and actions are taken outside the Application program, Diagnosis and Actions are programmed as an outside program or script
- In the case where System or Container movement is analyzed and actions are taken in the System program, Diagnosis and Actions are programmed in the System program
- In the case where System or Container movement is analyzed and actions are taken outside the System program, Diagnosis and Actions are programmed in D-System Monitor

C APIs provided by D-RE can be used for monitoring basic system movement such as CPU load or memory consumption.

In case of monitoring continuous system status (statistical status), marketed software products or OSS such as Nagios, Zabbix, etc. can be used.

DEOS Programming needs software framework technologies such as VM, Hypervisor, and Container. There are several marketed middleware products or OSS such as VMware and OpenStack. In the DEOS Project, we have been developing D-RE（DEOS Runtime Environment）using KVM and LXC as Reference Platforms. The binary and source codes are open in the DEOS home pages (refer to the following documents).

- D-RE Specifications (Doc # : DEOS-FY2013-RE-01E)
- D-RE API Specifications (Doc # : DEOS-FY2013-EA-01E)
- D-RE Commands Specifications (Doc # : DEOS-FY2013-EC-01E)
- D-RE API Sample Program Guide (Doc # : DEOS-FY2013-SP-01E)

Figure 2 shows how the technologies of D-RE are used to realize the DEOS Process:

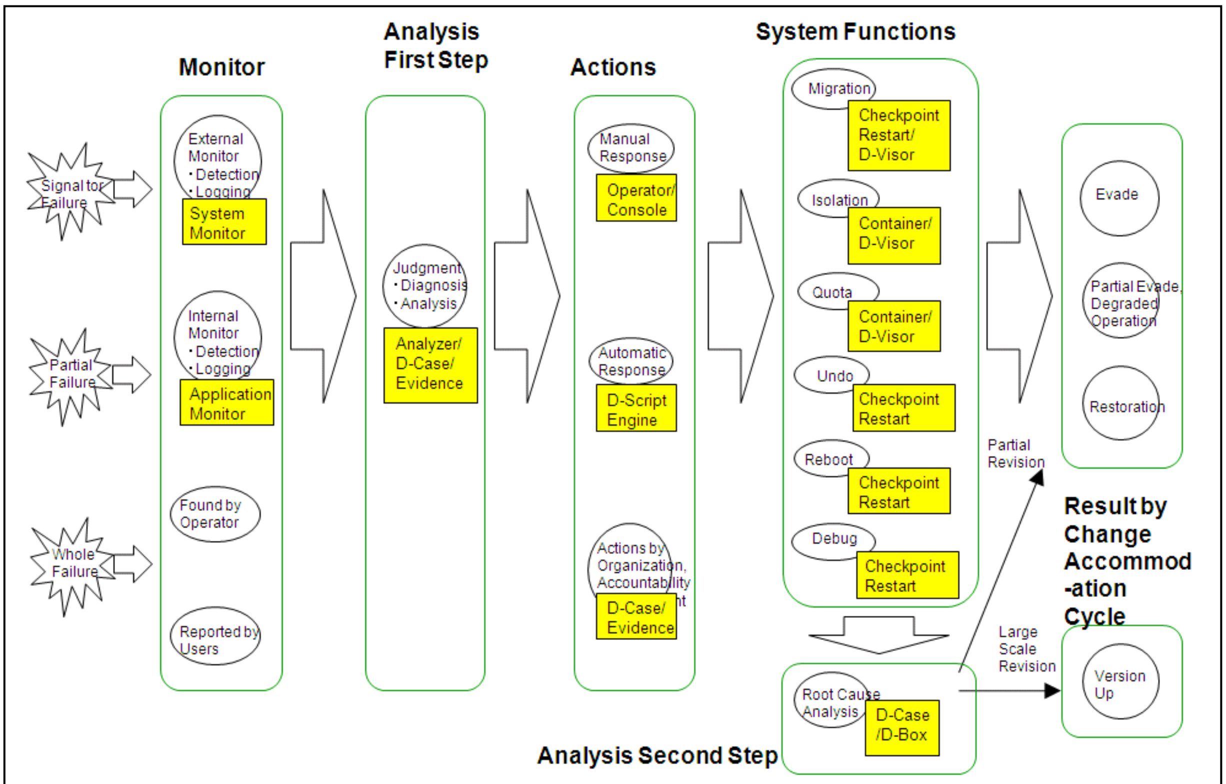


Figure 2. D-RE technologies which support realization of DEOS Process

2.3. DEOS Programming Remarks

1) Monitoring, Logging, Reporting

What should be monitored is argued in the consensus building process, and is described in D-Case. The possibility of detecting signals of HW failure is relatively high if appropriate sensors are implemented in the system and detecting software is working. In comparison, detecting signals of SW failure is difficult.

- There is a possibility of failure when a pattern which looks like the pattern of a failure in the past is monitored, but it is very hard to determine surely whether it will become a failure.

- There is a greater danger of memory leak as memory utilization gradually grows. A rejuvenating action such as rebooting the system at appropriate times is strongly recommended.

- Detecting causes of failure is impossible when the whole system goes down. To cope with this situation, the following should be considered:
  - ✓ Implement an independent monitoring mechanism
  - ✓ Have an operator watch for failure when the operator is on site
  - ✓ Implement a reporting mechanism by end users when an operator is not on site

- In case a portion is down other than the Monitor Program or the Application Program which includes Monitoring Points, the possibility of detecting failure is high

On D-RE, D-System Monitor for detecting abnormal movement of the OS or a malware attack on the OS could be mounted.

What should be logged or reported, or how logging or reporting is achieved is described in D-Case.

- It is very helpful in analyzing the root cause if there was logging of information of the system environment when the system failure was detected
- Reporting the fact of failure to the system or to the operator is the next important step
  - ✓ Report to Diagnosis Program when it is implemented
  - ✓ Report to Operator always

2) Analysis and Diagnosis

What should be analyzed, or how analysis or diagnosis is done is not always described explicitly in D-Case. It may be included implicitly in the description of Logging or Actions

- The System or Application Programmer should extract the algorithm or logic detailing how to analyze or diagnose from D-Case.

  - ➢ In many cases, a log is analyzed

- Log information must be adjusted or arranged so as not to exhaust the Log database by repetition of data (Log insolvency)

  - ➢ SEC（Simple Event Correlator、http://simple-evcorr.sourceforge.net/) is used in the demo system

- Integrity (no data modification etc.) of Log data must be assured

3) Actions

What should be acted upon, or how actions are done is not always described explicitly in D-Case. These may be included implicitly in the description of Logging or Analysis

- In many cases, Actions are implemented in Application programs to deal with failures related to Application tasks.
  - ✓ In the case where many actions are designed and implemented:
    - ◆ An application program or script might select appropriate action automatically
    - ◆ Appropriate actions are proposed to the Operator, and then the Operator selects from them

- To deal with system failures, actions are designed and implemented based on D-Case in advance
  - ✓ Script might select the appropriate action automatically
  - ✓ Appropriate actions are proposed to the Operator, and the Operator selects from them

3. Implementation Example: D-Case/D-RE Demo System

3.1. Overview of Demo system

This demonstration system will show some features of D-Case/D-RE.

Structure of Software Modules

This demonstration system is a Web-based application simulating a simplified CD Online Shopping system consisting of Apache, Tomcat and MySQL. Apache, Tomcat and MySQL are running in separated System Containers dedicated to them. Furthermore, this system uses Nagios management software (http://www.nagios.org), which monitors the hardware resources and network in the demonstration system.

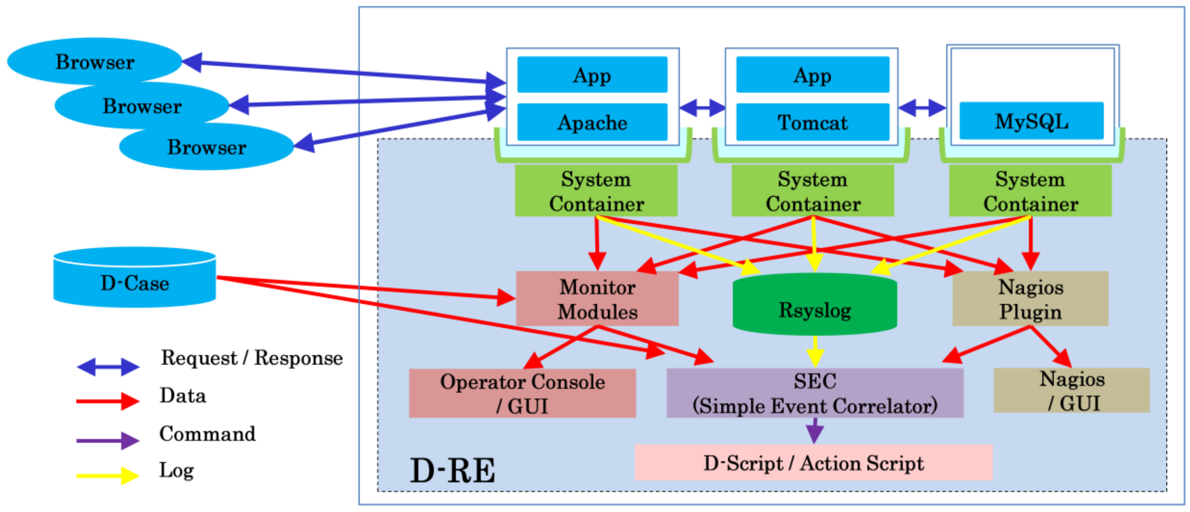


Figure 3. Structure of Software Modules

"ab" (Apache HTTP server benchmarking tool) is also used to simulate multiple, simultaneous requests from multiple Web Browsers in the Internet.

Demonstration Scenarios

This demonstration system is a typical Web-based application consisting of a Web server, an application server, and a DB Server, which are continuously monitored with an Operator Console. Each server is running in an individual System Container implemented in D-RE (DEOS Runtime Environment). At the Operator Console, Rsyslog daemon collects log records from the System Containers, and monitors modules, Nagios plugins and resource usage and performance of each server. When the monitor modules detect a failure, they indicate the failure on a GUI window, and pass the information on the failure to SEC at the same time. SEC will act on the failure by executing D-Scripts. The following four scenarios will explain how D-Case and D-RE collaborate to react to system failures:

Demonstration Scenario 1)
When a new service is deployed, the number of accesses per second to the system exceeds the expected number. At that time, following the prescription described in D-Case, the system performs an "undo" action on the deployment operation, and the system will return to the former operating state.

Demonstration Scenario 2)
The response time from the server increases suddenly. At that time, according to the prescription described in D-Case, the system forcibly stops some batch jobs which are not supposed to run during this time of day, and the system will return to the normal operation state.

Demonstration Scenario 3)
The memory usage exceeds the limit allowed by the system capacity. At that time, in accordance with the prescription described in D-Case in advance, the system restarts servers, deploys diagnostic modules and identifies the root cause.

Demonstration Scenario 4)
As part of daily checking operations, some pseudo operations are performed after the time of day is advanced to "tomorrow" temporarily, and some services are found to be unavailable. As the result of locating and identifying the root cause, it is found that a license will expire tomorrow, and the license is renewed in advance.

Videos of the above demonstration scenarios can be viewed at the site with the following URL:

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

3.2. Software Structure of Demonstration System

The following figure shows the structure of software components of a system using D-RE:

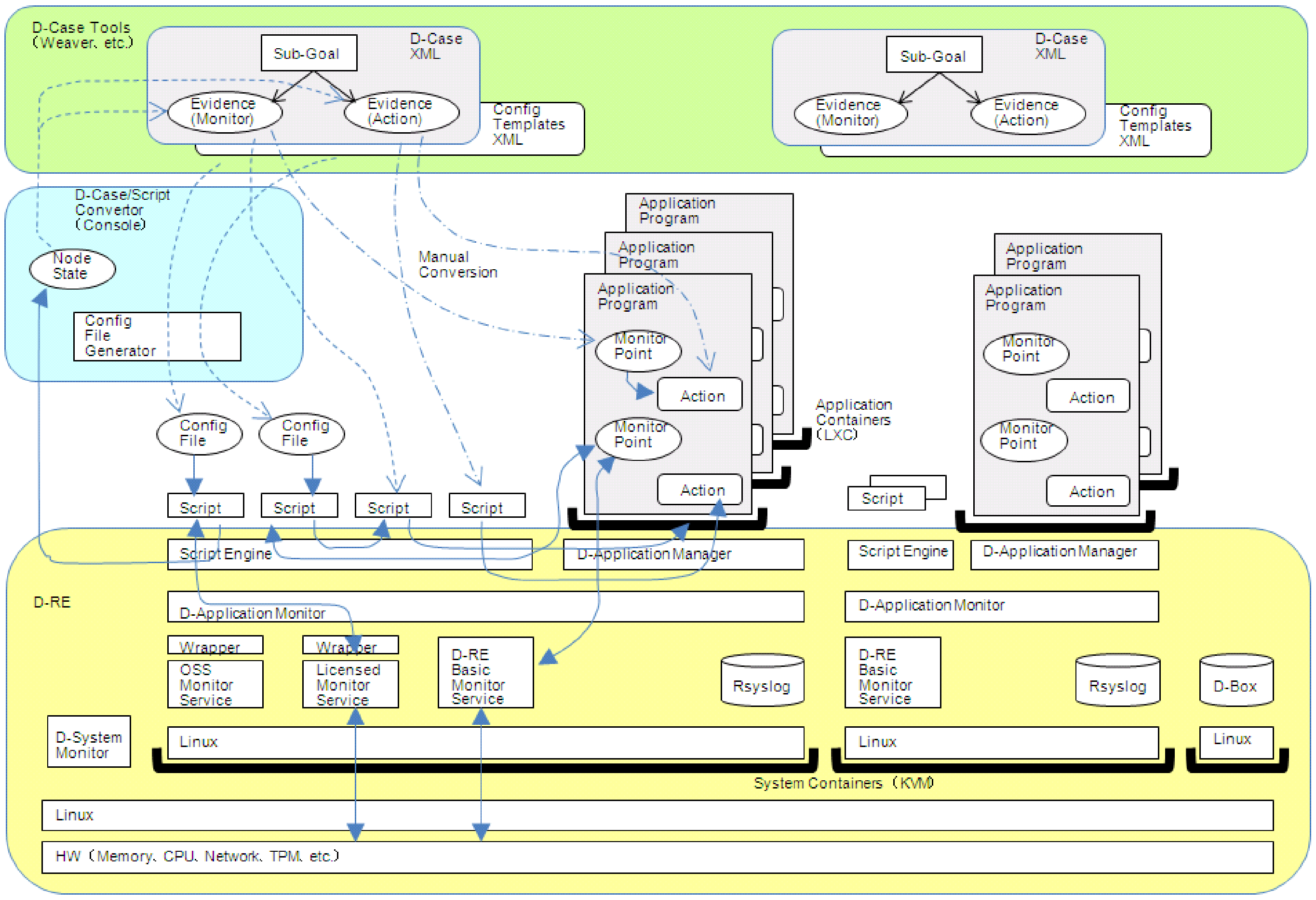


Figure 4. Software Structure of a system using D-RE

The monitoring subsystem is implemented by associating a specific instance of a monitor module to each monitoring node of a D-Case.
The event analyzing subsystem is implemented by feeding events to state machines realized with SEC (Simple Event Correlator), which takes into account mutual relationships among multiple events.

3.3. Steps from System Development to Execution

DEOS process is supported with Agreement Achievement/Confirmation Methodology and/or Tools **(D-Case)** together with a **DEOS Runtime Environment (D-RE)**. Utilizing D-Case and D-RE, the demonstration system is developed and executed through the following steps:

1) Agree on Service Requirements (Consequently, D-Case is constructed and fixed)
   Clarify management concepts and the business model of the company
   Decide and Agree on BCP (Business Continuity Plan)
   Determine and Agree on SLA-related parameters
   Create Monitor Nodes and Goal Nodes related to SLA

2) Determine monitoring targets and parameters of corresponding Monitor Nodes
   - Select a Monitor Node from the set of existing Monitor Modules
   - Specify behavior of new Monitor Modules if none of the existing ones are applicable

   - Specify external behavior of Monitoring Nodes in a D-Script Description File (template of configuration file)

3) Design behavior of Analysis Nodes, which drive Analyzers
   - Provide Analyzers with State Machines
     - Represent a sequence of multiple Events as state transitions
     - Have appropriate actions performed during state transitions and entries into new states

     - Create a D-Script Description File, which is a template of a configuration file
     - In this demonstration, SEC is used as an Analysis engine
     - Call up available Action scripts
     - Create and call up Action scripts specific to each user

4) Develop Application Logic

5) Develop Monitor Modules designed for users' specific monitoring needs in parallel with the development of Application Logic

6) Develop State Machines for analyzing events and actions specific to users' needs in parallel with the development of Application Logic

7) Execute the Application in D-RE
   The Application is run in accordance with created D-Script instances, from Monitor Nodes and Analysis Nodes in D-Case.

All the above steps will support DEOS Process.

3.4. Mechanism for Monitor Nodes and Analysis Nodes in D-Case

The following diagram shows how a D-Case Weaver, which is a tool written in JavaScript and which runs in a Web Brower, generates instances of Monitor Modules and configuration files for SEC from a D-Case. First, when a user creates a Monitor Node, the D-Case Weaver inserts a record (table row) into a D_Case_Node table in a relation database which is used to store a D-Case. At that time, the D-Case Weaver asks the user to select a Monitor Module for the Monitor Node from the list of the available Monitor Modules which are stored in the D_Script table in the database, and specify monitoring parameters for the node. The users' selections and associated parameters are stored in the table row corresponding to the Monitor Node. In other words, the D-Case Weaver associates D-Case Monitor Nodes with D-Script（Monitoring Modules）. After inserting rows for Monitor Nodes, D-Case Weaver invokes Config Generator for Monitoring and Config Generator for Analyzers.

The Config Generator for Monitoring joins the D_Case_Node table and D_Script table to gather all the information in a single instance of a Monitor in a single row of the joined table. Then, the generator selects only rows which originate from a Monitor Node, and repeatedly applies the parameters stored in a row to a template of a Monitor Module stored also in the same row. Here, to apply the parameters to a template means that the value of a parameter is inserted into the placeholder in the template corresponding to the name of the parameter. The Config Generator for Analyzers woks in the same way to generate configuration files for SEC by joining the D_Case_Node table and D_Script table, and generating the configuration files by applying the parameters in a row to a template of a SEC configuration file in the row.

The following sections explain how to create Monitor Nodes and Analysis Nodes with D-Case Weaver.

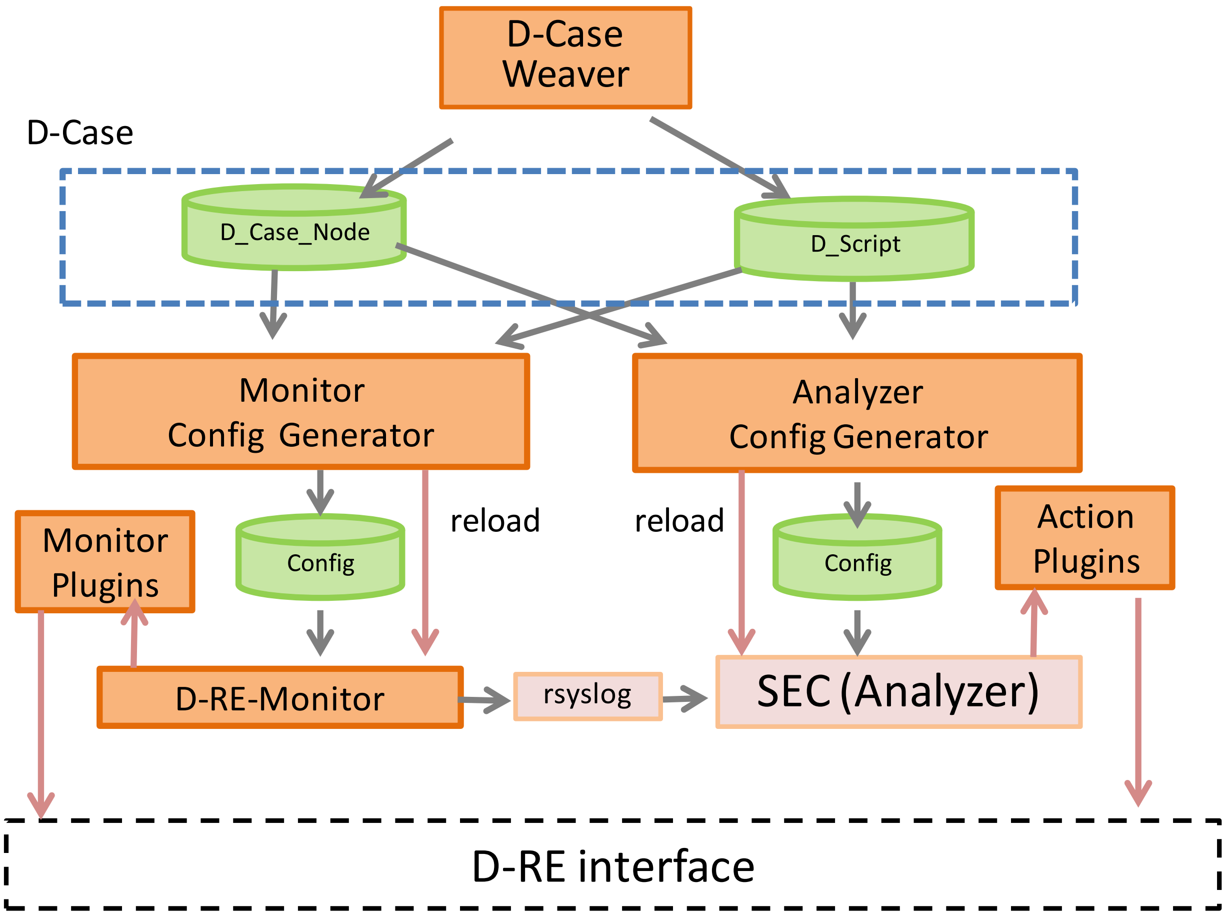
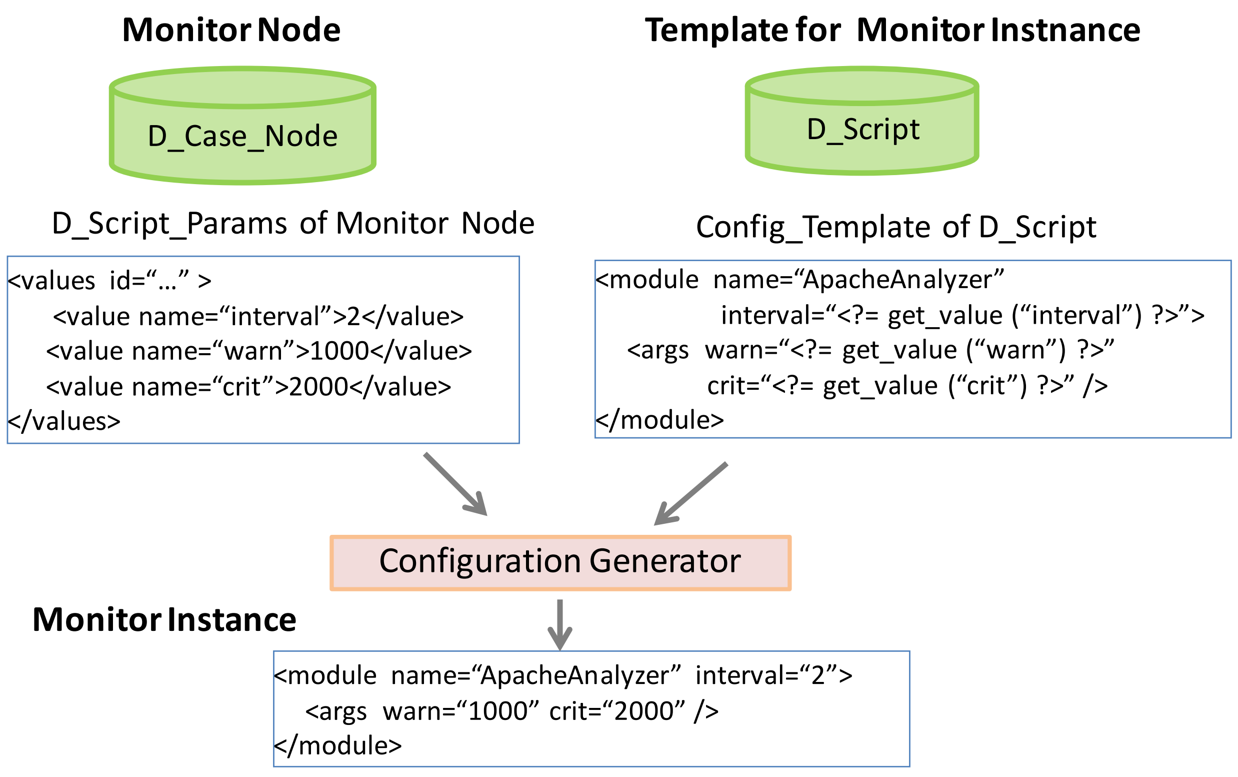


Figure 5. Creating specific instances of Monitor and Analysis Modules

Implementation of Monitor Modules

Each monitor is a specific Monitor Module instance, customized with a set of configuration parameters. In the demonstration system, Monitor Modules provided by D-RE are written in Python programming language, and are subclasses of the “PluginBase” class prototyped for this demonstration system. Consequently, a Monitor Instance means one of those Python subclasses, and is constructed with values based on the set of the configuration parameters.

The following diagram shows how a particular Monitor instance is constructed by combining a D-Script defined as a template and a set of parameter values for the instance. The template contains several placeholders in the format (<?= get_value (name) ?>), and the Monitor instance is created by replacing the placeholders with the values corresponding to the names in the placeholders. A set of pairs of names and values are stored in a Monitor Node in D-Case.


Monitor Node
D_Case_Node
Template for Monitor Instnance
D_Script
D_Script_Params of Monitor Node
Config_Template of D_Script
Configuration Generator
Monitor Instance

```
<values id="…" >
   <value name="interval">2</value>
   <value name="warn">1000</value>
   <value name="crit">2000</value>
</values>
```

```
<module name="ApacheAnalyzer"
        interval="<?= get_value ("interval") ?>">
   <args warn="<?= get_value ("warn") ?>"
        crit="<?= get_value ("crit") ?>" />
</module>
```

```
<module name="ApacheAnalyzer" interval="2">
   <args warn="1000" crit="2000" />
</module>
```

Figure 6. Generating a Monitor instance

Implementation of Analyzer Modules

Analyzer modules are implemented as state machines realized with SEC. State machines in SEC are defined with configuration files of SEC. The configuration files are generated from template files with placeholders to represent a group of state machines and a set of parameter values which are substituted for the placeholders.

Templates of Monitor Module and Analyzer Module

Templates for Module instances are written in an XML file containing multiple templates, each of which contain the following elements:

- Identifier to identify a template for the Monitor module and configuration information for a state machine
- Description and author of the template
- Description and data type of each parameter of template parameters
- Template with placeholders which are replaced with actual values to generate a Monitor instance or a configuration file for state machines in SEC

Refer to Appendix A1 for examples of templates for Monitor modules and Analysis modules。

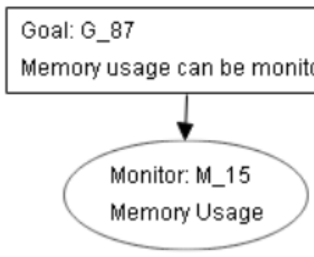
3.5. How to implement Monitor Nodes in D-Case

Evidence Nodes in D-Case specify what aspects of programs' behavior are monitored while running on D-RE. Normally, while constructing D-Case during development of a program, monitoring requirements are gathered and Monitor Nodes are constructed.

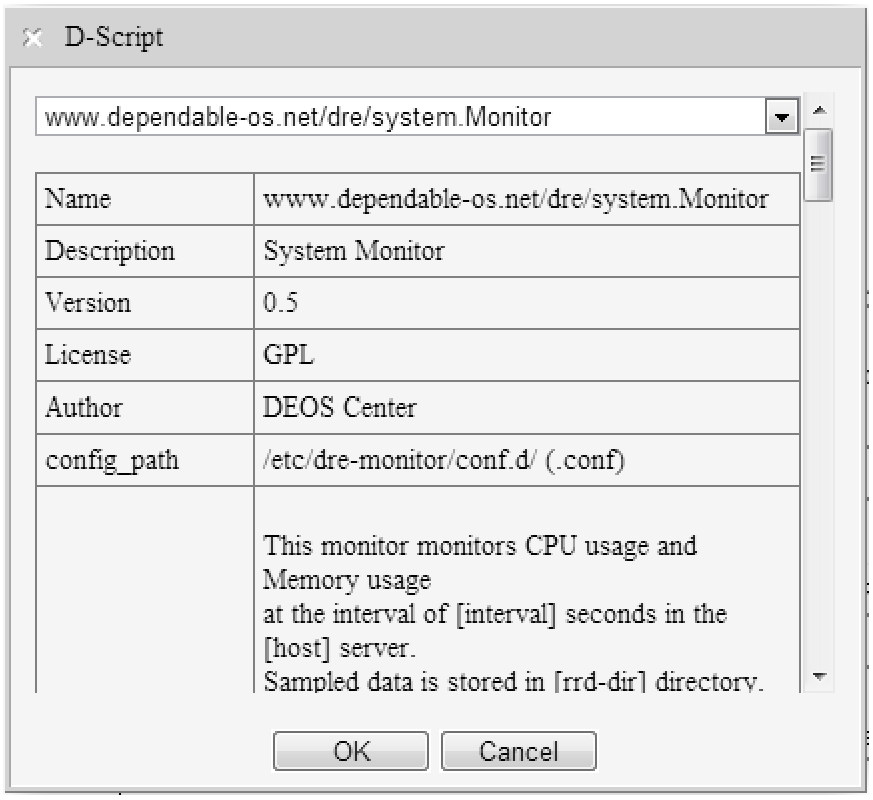
Each Monitor Node has data which identifies a specific Monitor Module and a set of parameter values which need to be generated to run an instance of the Monitor Module. The following briefly explains how to specify those parameter values with D-Case Weaver, which is a JavaScript application running on a Web Browser to create and edit D-Case (See DEOS-FY2013-CW-01E : "D-Case Weaver Specification, and Introduction and Users' Guide" for details).

1) Monitoring CPU and Memory Usage

Graphical Presentation of a Goal in D-Case Weaver:

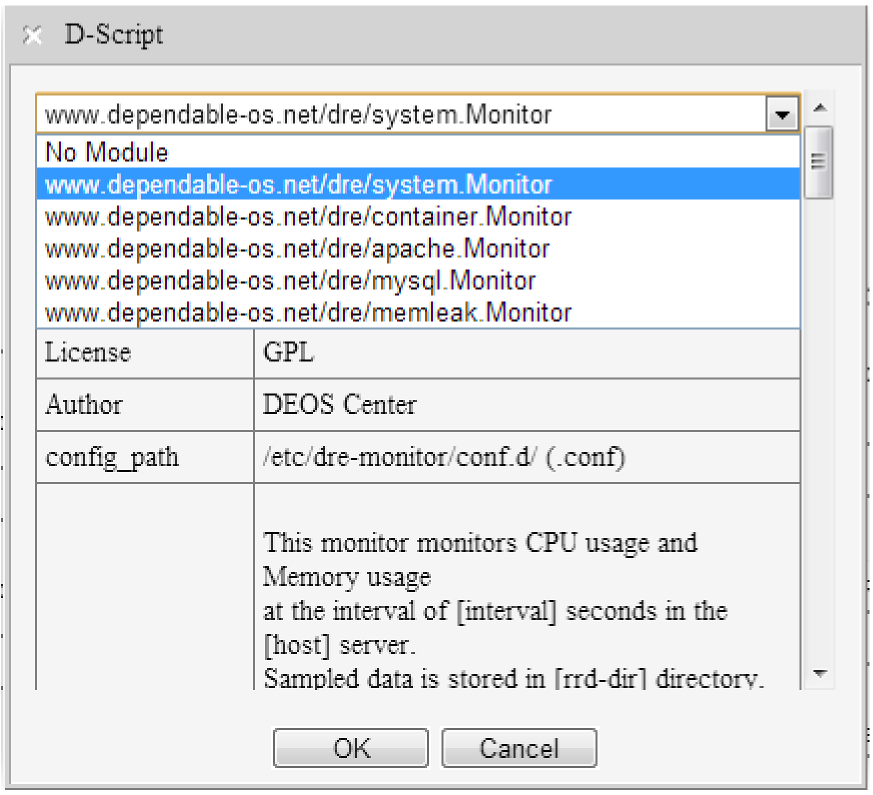


After creating a Monitor Node (M_15), right-click on the Node, select "Edit D-Script", and open the D-Script edit panel.



Name	www.dependable-os.net/dre/system.Monitor
Description	System Monitor
Version	0.5
License	GPL
Author	DEOS Center
config_path	/etc/dre-monitor/conf.d/ (.conf)
	This monitor monitors CPU usage and Memory usage at the interval of [interval] seconds in the [host] server. Sampled data is stored in [rrd-dir] directory.

Click a pull-down list of Monitor Modules, and choose an appropriate Monitor Module from the list.

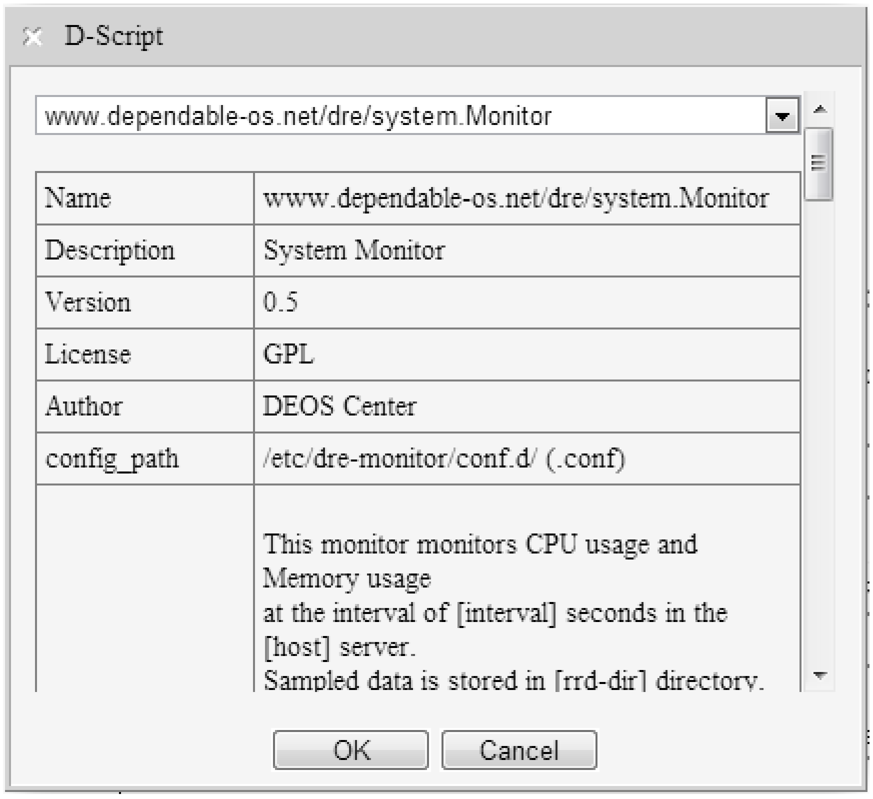


www.dependable-os.net/dre/system.Monitor

No Module
www.dependable-os.net/dre/system.Monitor
www.dependable-os.net/dre/container.Monitor
www.dependable-os.net/dre/apache.Monitor
www.dependable-os.net/dre/mysql.Monitor
www.dependable-os.net/dre/memleak.Monitor

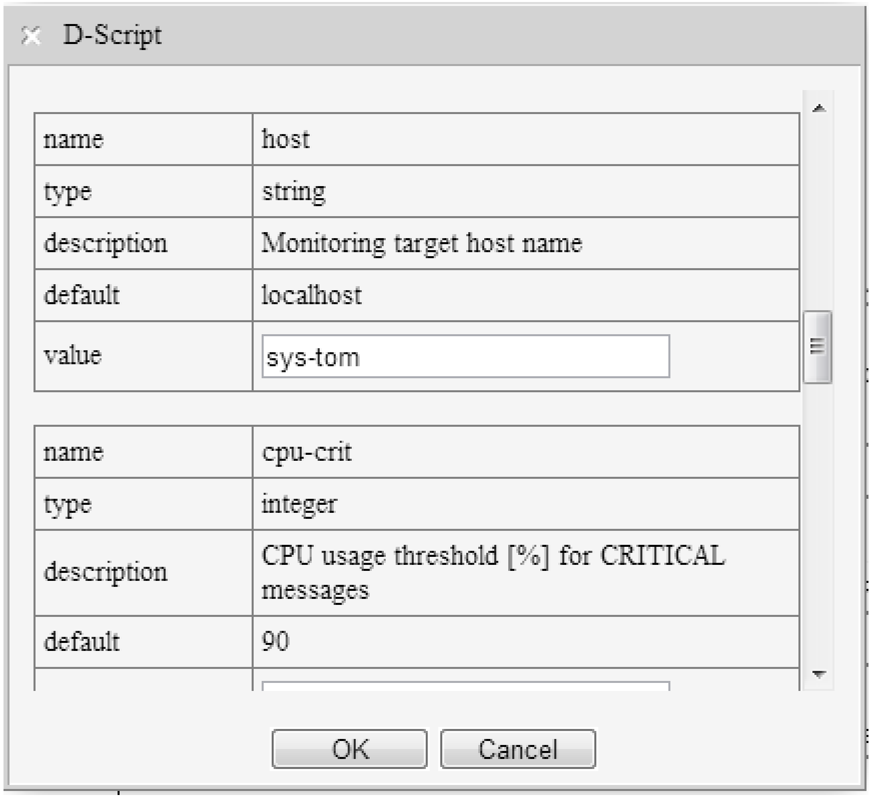
License	GPL
Author	DEOS Center
config_path	/etc/dre-monitor/conf.d/ (.conf)
	This monitor monitors CPU usage and Memory usage at the interval of [interval] seconds in the [host] server. Sampled data is stored in [rrd-dir] directory.

For the selected Monitor Module, a set of applicable parameters is shown. Specify the appropriate value for each parameter.



www.dependable-os.net/dre/system.Monitor

Name	www.dependable-os.net/dre/system.Monitor
Description	System Monitor
Version	0.5
License	GPL
Author	DEOS Center
config_path	/etc/dre-monitor/conf.d/ (.conf)
	This monitor monitors CPU usage and Memory usage at the interval of [interval] seconds in the [host] server. Sampled data is stored in [rrd-dir] directory.



name	host
type	string
description	Monitoring target host name
default	localhost
value	sys-tom

name	cpu-crit
type	integer
description	CPU usage threshold [%] for CRITICAL messages
default	90

In this example, the following XML fragment is recorded in a XML file representing the D-Case:

XML :

```
<dcase:node type="Monitor" id="_JHchkKurEeGUa93z1Rd6MQ" name="M_15">
<dcase:description>Memory Usage</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 built-in monitor (ID: www.dependable-os.net/dre/system.Monitor)

3.6. How to implement Analysis Nodes in D-Case

A Monitor instance usually monitors occurrences of some events related to some resources, and reports a failure and a fault, or an early sign of them. In general, several Monitor instances detect different symptoms caused by a single failure from their own point of view, and report them independently. Furthermore, a Monitor instance may report several kinds of failures caused by different resources, and an Analyzing instance may be able to narrow down the root cause of the failures by combining several reports from multiple Monitor instances. Consequently, an Analyzer subsystem needs to receive reports from multiple Monitor instances and consider relations among them to determine the most feasible root cause based on all of the reports.

In the implementation of the Analyzer subsystem, (enhanced) finite state machines are used to consider mutual relations among multiple events and select the most plausible course of actions for the events. To select an appropriate course of actions regarding the multiple events, we could use multiple conditional statements ("if" statements) by combining them in a sequential manner and/or nested manner. However, the number of these combinations usually increases exponentially or factorially as the number of events increase when the order of occurrences of the events are considered. Furthermore, it would not be a trivial task to detect two specific events occurring within a certain period of time by using combinations of conditional statements.

The following figure shows an automaton, which stays in the Initial state until an Event A occurs, and then Action A is taken. The state changes to State A when an Event A occurs. Once in State A, when an Event B occurs, Action B is taken and the state changes to State B. In the State A, after a certain time passes, Recovery A action is carried out and the state returns to the Initial State.

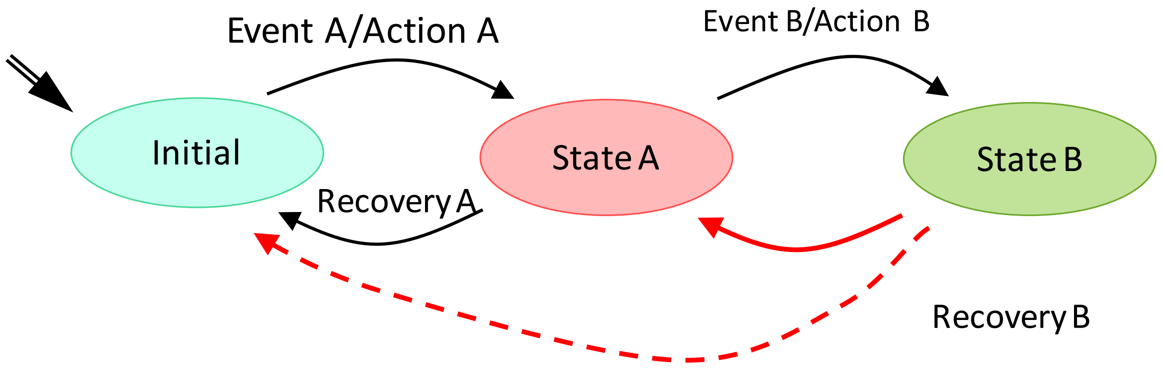


Figure 7. State Transitions and Actions

The following figure shows an automaton which goes to the State C when an Event A and an Event B occur in any order:

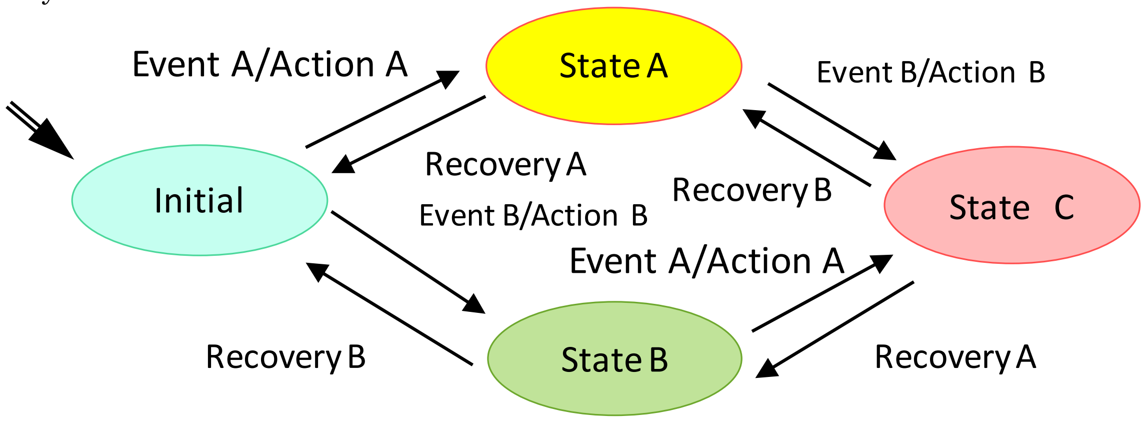


Figure 8. Processing of Events independently of order of occurrence

In addition, by specifying a maximum duration in which a state machine may stay in each state, the state machine can detect multiple events occurring in the specified interval. For instance, we can specify that the state machine goes to the State A upon Event A, and may stay in the State A for at most 30 seconds. When we further specify that the state machine is affected by Event B while in the State A to change to State C. However, the state machine can be affected by Event B only if that event occurs within 30 seconds after Event A.

In an implementation of an Analyzer subsystem utilizing finite state machines, each state corresponds to a sequence of related events. By making transitions to states and taking appropriate action upon the transitions, the state reached after the transitions will indicate some feasible causes of the events. Furthermore, we can simulate running multiple state machines concurrently to handle multiple sequences of events which are independent of each other, as well as to handle a single sequence to view that sequence of the events from different perspectives and so perform a better root cause analysis. Using state machines to perform these operations would be easier than combinations of multiple conditional statements.

The following figure shows an example of handling of a single failure event which may occur during normal operation together with multiple events which occur upon an actual failure in a network system. For a single event, which may be the first event in the sequence, the state machine writes a log record and goes to the Burst State, and will stay in that state until a Timeout. If the same event occurs in the Burst State, the state machine waits without taking any actions, until the event occurs N times. At the occurrence of N-th event, the state machine considers the failure as a permanent failure, reports the failure to a manager, and goes to the Dormant State, in which the events are ignored.

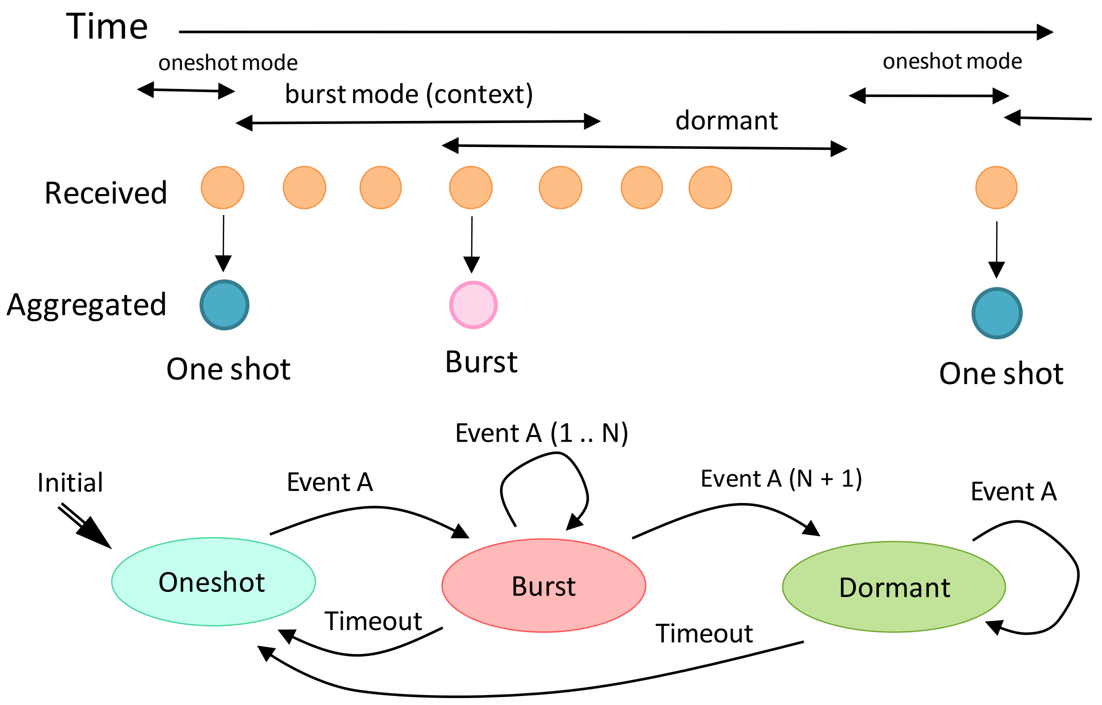
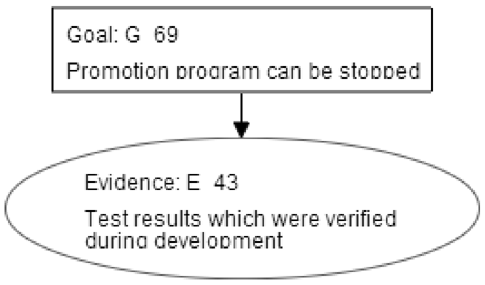


Figure 9. Processing of consecutive events

Implementation of state machines using SEC (Simple Event Correlator). State machines are represented with configuration files, which are generated from Goal Nodes in the D-Case.

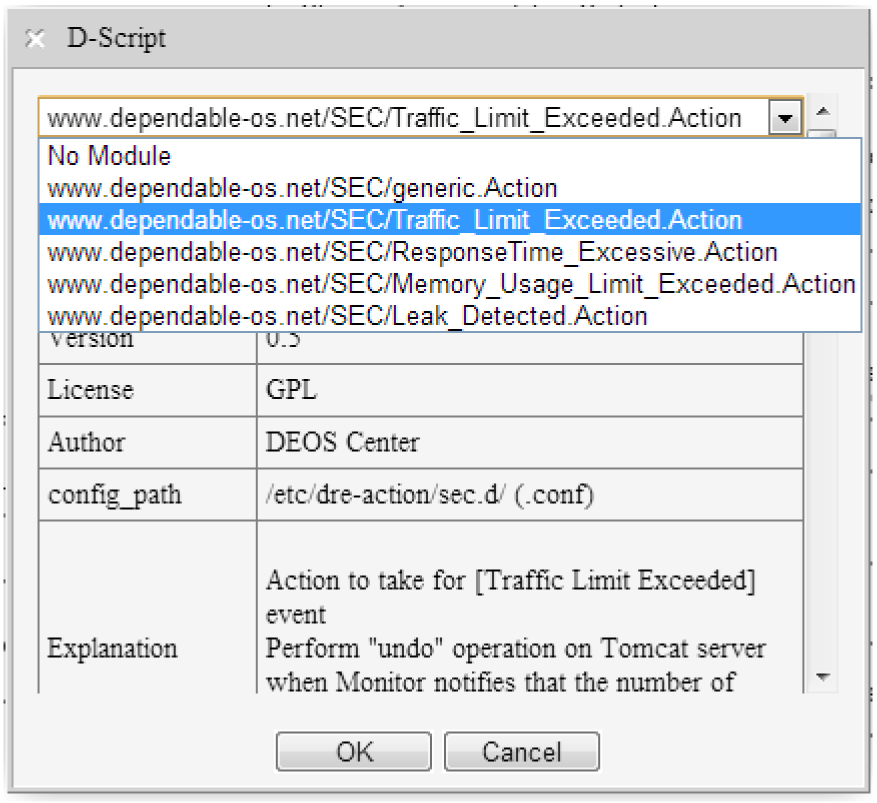
The following shows how D-Case Weaver is used to create a Goal Node for handling an increased number of accesses to a Web Server by stopping a promotion program:

Graphical Representation of D-Case Node in Case Weaver:

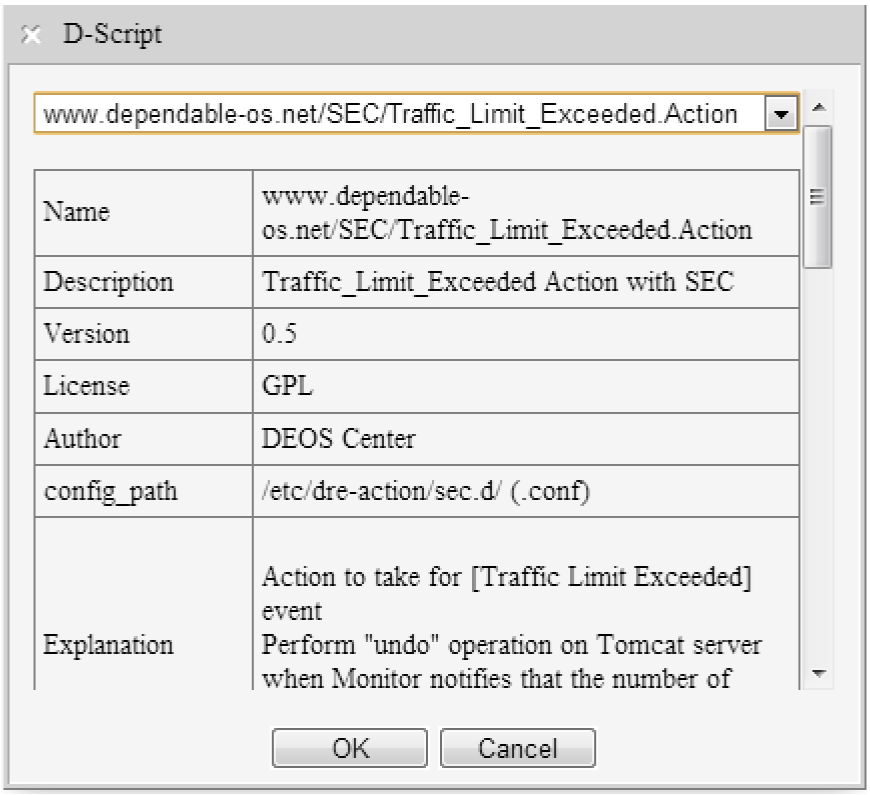

Goal: G 69
Promotion program can be stopped
Evidence: E 43
Test results which were verified during development

After creating Goal Node G_69, right-click on the node, select "Edit D-Script", choose Analyzer Module, and specify values of parameters.

Select an Analyzer Module


D-Script
www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action
No Module
www.dependable-os.net/SEC/generic.Action
www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action
www.dependable-os.net/SEC/ResponseTime_Excessive.Action
www.dependable-os.net/SEC/Memory_Usage_Limit_Exceeded.Action
www.dependable-os.net/SEC/Leak_Detected.Action
OK
Cancel

Version	0.5
License	GPL
Author	DEOS Center
config_path	/etc/dre-action/sec.d/ (.conf)
Explanation	Action to take for [Traffic Limit Exceeded] event Perform "undo" operation on Tomcat server when Monitor notifies that the number of

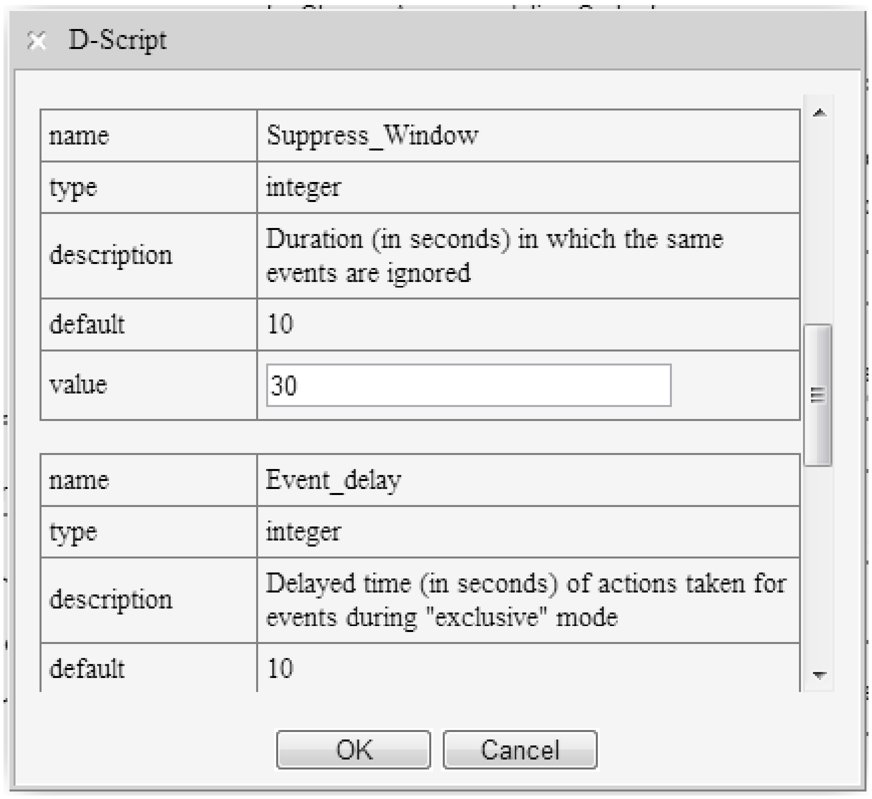

D-Script

www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action

Name	www.dependable-os.net/SEC/Traffic_Limit_Exceeded.Action
Description	Traffic_Limit_Exceeded Action with SEC
Version	0.5
License	GPL
Author	DEOS Center
config_path	/etc/dre-action/sec.d/ (.conf)
Explanation	Action to take for [Traffic Limit Exceeded] event Perform "undo" operation on Tomcat server when Monitor notifies that the number of

OK Cancel

Specify appropriate values for parameters required by the Analyzer Module


D-Script

name	Suppress_Window
type	integer
description	Duration (in seconds) in which the same events are ignored
default	10
value	30

name	Event_delay
type	integer
description	Delayed time (in seconds) of actions taken for events during "exclusive" mode
default	10

OK Cancel

In this example, the following XML node is recorded in the D-Case:

XML:

```
<dcase:node type="Goal" id="_yl_a8MQhEeGBPY79tzkVYg" name="G_69">
<dcase:description>Stop Promotion Program</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. Examples of Action Commands

In this demonstration system, SEC is used to implement the Analyzer Subsystem. Actions taken by state machines implemented with SEC are specified in configuration files of SEC as the string values of D-RE commands for "action" identifiers. The following table shows D-RE commands called by SEC:

Actions	Description
shellcmd /usr/share/dre-demo/d-script/action/act_logging.sh	Write results of analysis
shellcmd usr/share/dre-demo/d-script/action/act_reboot_sys_container.sh	Restart System Container
shellcmd /usr/share/dre-demo/d-script/action/act_undo_sys_container.sh	Undo the changes applied to System Container
shellcmd /usr/share/dre-demo/d-script/action/act_kill_batch_sys.sh	Stop batch jobs
shellcmd /usr/share/dre-demo/bin/dcase-status	Update the status of Monitor Nodes in D-CaseWeaver
event	Raise events internal to SEC
create,delete	Change states of SEC

4. Conclusion

The following items are essential to accomplish OSD in DEOS, namely, continuous service availability and accountability in ever changing environments:

- Repeatable, agile process for continuous improvement encompassing development and operational phases （**DEOS Process**）
- Tools and methodology for agreement establishment and confirmation (**D-Case**) and runtime environment (**D-RE：DEOS Runtime Environment**) to support DEOS Process systematically
- Scripting language （**D-Script**） and **D-Aware Application Programs**, which achieve OSD based on D-Case

In this Programming Reference, Chapter 2 explained an approach which might be useful to write scripts and application programs which are the third element in the above list of essential elements of OSD, and Chapter 3 described a demonstration system which is an example of the approach.

Appendix

A1. Examples of Templates for Monitor and Analyzer Modules

The following sample XML file shows templates for Monitor Modules and Analyzer Modules using SEC.

Templates for Monitor Modules

1) Monitoring CPU and Memory Usage in Host System

```
 <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="en_US"><![CDATA[
        This monitor monitors CPU usage and Memory usage
        at the interval of [interval] seconds in the [host] server.
        Sampled data is stored in [rrd-dir] directory.
        When the CPU usage exceeds [cpu-warn]%, a WRNING message is written to syslog.
        When the CPU usage exceeds [cpu-crit]%, a CRITICAL message is written to syslog.
        When the Memory usage exceeds [mem-warn]%, a WARNING message is written to syslog.
        When Memory usage exceeds [mem-crit]%, a CRITICAL message is written to syslog.
        Graph data is stored in [graph-dir] directory.

      ]]></explanation>
    </explanations>
    <parameters>
      <!-- Details must be determined.   Similar to XML schema? -->
      <parameter name="interval" type="integer">
          <description>Monitoring Interval [seconds]</description>
          <default>2</default>
      </parameter>
      <parameter name="host" type="string">
          <description>Monitoring target host name</description>
          <default>localhost</default>
      </parameter>
      <parameter name="cpu-crit" type="integer" min="1" max="100">
          <description>CPU usage threshold [%] for CRITICAL messages</description>
          <default>90</default>
      </parameter>
      <parameter name="cpu-warn" type="integer" min="1" max="100">
           <description>CPU usage threshold [%] for WARNING messages</description>
           <default>80</default>
      </parameter>
       <parameter name="mem-crit" type="integer" min="1" max="100">
          <description>Memory usage threshold [%] for CRITICAL messages</description>
           <default>90</default>
      </parameter>
      <parameter name="mem-warn" type="integer" min="1" max="100">
           <description>Memory usage threshold [%] for WARNING messages</description>
           <default>80</default>
      </parameter>
      <parameter name="related-node" type="string">
           <description>Related D-Case Node</description>
      </parameter>
      <parameter name="rrd-dir" type="string">
           <description>Directory path to store sampled data</description>
           <default>/tmp/dre-monitor/rrd/localhost/system</default>
      </parameter>
      <parameter name="graph-dir" type="string">
           <description>Directory path to store graph data</description>
           <default>/tmp/dre-monitor/graph/localhost/system</default>
      </parameter>
      </parameters>
</monitor>
```

2) Monitoring CPU and Memory Usage in Containers

```
  <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="en_US"><![CDATA[
               This monitor monitors CPU usage and Memory usage
               at the interval of [interval] seconds in the [container] container.
               Sampled data is stored in [rrd-dir] directory.
               When the CPU ussage exceeds [cpu-warn]%, a WRNING message is written to syslog.
               When the CPU usage exceeds [cpu-crit]%, a CRITICAL message is written to syslog.
               When the Memory usage exceeds [mem-warn]%, a WARNING message is written to
syslog.
               When Memory usage exceeds [mem-crit]%, a CRITICAL message is written to syslog.
               Graph data is stored in [graph-dir] directory.

            ]]></explanation>
      </explanations>
      <parameters>
         <parameter name="interval" type="integer">
               <description>Monitoring interval [seconds]</description>
            <default>2</default>
        </parameter>
        <parameter name="container" type="string">
            <description>Monitoring target container name</description>
            <default>container</default>
        </parameter>
        <parameter name="cpu-crit" type="integer" min="1" max="100">
            <description>CPU usage threshold [%] for CRITICAL messages</description>
            <default>90</default>
        </parameter>
        <parameter name="cpu-warn" type="integer" min="1" max="100">
            <description>CPU usage threshold [%] for WARNING messages</description>
            <default>80</default>
        </parameter>
        <parameter name="mem-crit" type="integer" min="1" max="100">
            <description>Memory usage threshold [%] for CRITICAL messages</description>
            <default>90</default>
        </parameter>
        <parameter name="mem-warn" type="integer" min="1" max="100">
            <description>Memory usage threshold [%] for WARNING messages</description>
            <default>80</default>
        </parameter>
        <parameter name="related-node" type="string">
            <description>Related D-Case Node</description>
        </parameter>
        <parameter name="rrd-dir" type="string">
            <description>Directory path to store sampled data</description>
            <default>/tmp/dre-monitor/rrd/localhost/container</default>
        </parameter>
        <parameter name="graph-dir" type="string">
            <description>Directory path to store graph data</description>
            <default>/tmp/dre-monitor/graph/localhost/container</default>
        </parameter>
      </parameters>
   </monitor>
```

3) Monitoring Access Counts in Apache Server

```
   <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="en_US"><![CDATA[
               This Monitor monitors the number of accesses to Apache Server
               at the interval of [interval] seconds
               in [host] server.
               Sampled data is stored in [rrd-dir] directory.
               When the number of accesses exceeds [warn], a WARNING message is written to syslog.
               When the number of accesses exceeds [crit], a CRITICAL message is written to syslog.
               Graph data is stored in [graph-dir] directory.

           ]]></explanation>
      </explanations>
      <parameters>
         <parameter name="interval" type="integer">
               <description>Monitoring Interval [seconds]</description>
               <default>2</default>
         </parameter>
         <parameter name="host" type="string">
               <description>Apache server host name</description>
               <default>localhost</default>
         </parameter>
         <parameter name="crit" type="integer" min="0">
               <description>Access counts threshold for CRITICAL messages</description>
               <default>2500</default>
         </parameter>
         <parameter name="warn" type="integer" min="0">
               <description>Access counts threshold for WARNING messages</description>
               <default>2000</default>
         </parameter>
         <parameter name="related-node1" type="string">
               <description>Related D-Case Node</description>
         </parameter>
         <parameter name="related-node2" type="string">
               <description>Related D-Case Node 2 (Notification)</description>
         </parameter>
```

```
        <parameter name="rrd-dir" type="string">
            <description>Directory path to store sampled data</description>
            <default>/tmp/dre-monitor/rrd/localhost/apache</default>
        </parameter>
        <parameter name="graph-dir" type="string">
            <description>Directory path to store graph data</description>
            <default>/tmp/dre-monitor/graph/localhost/apache</default>
        </parameter>
    </parameters>
 </monitor>
```

4) Monitoring Response Times from MySQL Server

```
 <monitor>
        <name>www.dependable-os.net/dre/mysql.Monitor</name>
        <description>MySQL Response 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="en_US"><![CDATA[
                This Monitor monitors response times from MySQL Server
                at the interval of [interval] seconds in [host] server.
                This monitor connects to database [db]L as user [user] and password [passwd],
                and monitors response time when accessing a table [table] in database [db].
                Sampled data is stored in [rrd-dir] directory.
                When the response time exceeds [warn] seconds, a WRNING message is written to syslog.
                When the response time exceeds [crit] seconds, a CRITICAL message is written to syslog.
                Graph data is stored in [graph-dir] directory.

          ]]></explanation>
    </explanations>
    <parameters>
        <parameter name="interval" type="integer">
            <description>Monitoring Interval [seconds]</description>
            <default>2</default>
        </parameter>
        <parameter name="host" type="string">
            <description>MySQL server host name</description>
            <default>localhost</default>
        </parameter>
        <parameter name="user" type="string">
            <description>MySQL user name</description>
        </parameter>
        <parameter name="passwd" type="password">
            <description>MySQL password</description>
        </parameter>
        <parameter name="db" type="string">
            <description>MySQL database name</description>
        </parameter>
        <parameter name="table" type="string">
            <description>MySQL table name</description>
        </parameter>
        <parameter name="crit" type="integer" min="0" max="1">
            <description>Response time threshold [seconds] for CRITICAL messages</description>
            <default>1</default>
        </parameter>
        <parameter name="warn" type="integer" min="0" max="1">
            <description>Response time threshold [seconds] for WARNING messages</description>
            <default>0.8</default>
        </parameter>
        <parameter name="related-node" type="string">
            <description>Related D-Case Node</description>
        </parameter>
        <parameter name="rrd-dir" type="string">
            <description>Directory path to store sampled data</description>
            <default>/tmp/dre-monitor/rrd/localhost/mysql</default>
        </parameter>
        <parameter name="graph-dir" type="string">
            <description>Directory path to store graph data</description>
            <default>/tmp/dre-monitor/graph/localhost/mysql</default>
        </parameter>
     </parameters>
</monitor>
```

5) Monitoring Diagnosis by Memory Leak Analyzer

```
<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="en_US"><![CDATA[
          This Monitor watches diagnosis from Memory Leak Analyzer
          at the interval of [interval] seconds in [host] server.
          Sampled data is stored in [rrd-dir] directory.
          When the value of the diagnosis exceeds [warn], a WRNING message is written to syslog.
          When the value of the diagnosis exceeds [crit], a CRITICAL message is written to
syslog.
          Graph data is stored in [graph-dir] directory.

      ]]></explanation>
    </explanations>
    <parameters>
      <parameter name="interval" type="integer">
          <description>Monitoring Interval [seconds]</description>
          <default>2</default>
      </parameter>
```

```
        <parameter name="host" type="string">
            <description>Monitoring target host name</description>
            <default>localhost</default>
        </parameter>
        <parameter name="crit" type="integer" min="0" max="1">
            <description>Threshold for CRITICAL messages</description>
            <default>200</default>
        </parameter>
        <parameter name="warn" type="integer" min="0" max="1">
            <description>Threshold for WARNING messages</description>
            <default>150</default>
        </parameter>
        <parameter name="related-node" type="string">
            <description>Related D-Case Node</description>
        </parameter>
        <parameter name="rrd-dir" type="string">
            <description>Directory path to store sampled data</description>
            <default>/tmp/dre-monitor/rrd/localhost/mysql</default>
        </parameter>
        <parameter name="graph-dir" type="string">
            <description>Directory path to store graph data</description>
            <default>/tmp/dre-monitor/graph/localhost/mysql</default>
        </parameter>
      </parameters>
  </monitor>
```

Templates for Analyzer Modules using SEC

1) A generic template of configuration files for SEC (Simple Event Correlator)

```
  <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="en_US"><![CDATA[
            Specify the content of a configuration file for Simple Event Correlator (SEC).
                        See man page of "sec" command or FAQ on Simple Event Correlator.

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

2) Limit Access Counts (per minute) to Apache Server

```
 <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=shellcmd /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="en_US"><![CDATA[
          Action to take for [Traffic Limit Exceeded] event
          Perform "undo" operation on Tomcat server when Monitor notifies that the number of
accesses exceeds the predefined limit.
      ]]></explanation>
    </explanations>
    <parameters>
       <parameter name="Suppress_Window" type="integer">
          <description>Duration (in seconds) in which the same events are ignored</description>
          <default>10</default>
      </parameter>
      <parameter name="Event_delay" type="integer" min="1" max="100">
          <description>Delayed time (in seconds) of actions taken for events during "exclusive"
mode</description>
          <default>10</default>
      </parameter>
      <parameter name="Action_delay" type="integer" min="1" max="100">
          <description>Delayed time (in seconds) of actions</description>
```

```
            <default>0</default>
        </parameter>
        <parameter name="Apply_Snapshot" type="string">
            <description>Snapshot which will be used for Undo operation</description>
            <default>ss1</default>
        </parameter>
    </parameters>
</action>
```

3) Abort a MySQL Batch Job when response time from MySQL exceeds the limit

```
 <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:deosc-laptop>EventSource=/Host:deosc-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:deosc-laptop>EventSource=/Host:deosc-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:deosc-laptop>EventSource=/Host:deosc-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="en_US"><![CDATA[
          Action to take for [ResponseTime_Excessive] event
          Stop batch jobs using DB Server when Monitor notifies that the response time from DB
exceeds the limit.

      ]]></explanation>
    </explanations>
    <parameters>
       <parameter name="Suppress_Window" type="integer">
          <description>Duration (in seconds) in which the same events are ignored</description>
          <default>10</default>
      </parameter>
      <parameter name="Event_delay" type="integer" min="1" max="100">
          <description>Delayed time (in seconds) of actions taken for events during "exclusive"
mode</description>
          <default>10</default>
      </parameter>
      <parameter name="Action_delay" type="integer" min="1" max="100">
          <description>Delayed time (in seconds) of actions</description>
          <default>0</default>
      </parameter>
      <parameter name="Kill_Param" type="string">
```

```
            <description>Parameters passed to act_kill_batch.sh script</description>
            <default>sql</default>
        </parameter>
    </parameters>
 </action>
```

4) Restart System Container when memory usage exceeds the limit

```
 <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:deosc-laptop>EventSource=/Host:deosc-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:deosc-laptop>EventSource=/Host:deosc-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:deosc-laptop>EventSource=/Host:deosc-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="en_US"><![CDATA[
            Action taken for [Memory_Usage_Limit_Exceeded] event
            Restart Web/Application servers when Monitor notifies that Application server consumes
more memory than the limit.

        ]]></explanation>
    </explanations>
    <parameters>
        <parameter name="Suppress_Window" type="integer">
            <description>Duration (in seconds) in which the same events are ignored</description>
            <default>10</default>
        </parameter>
        <parameter name="Event_delay" type="integer" min="1" max="100">
            <description>Delayed time (in seconds) of actions taken for events during "exclusive"
mode</description>
            <default>10</default>
        </parameter>
        <parameter name="Action_delay" type="integer" min="1" max="100">
```

```
                <description>Delayed time (in seconds) of actions</description>
                <default>0</default>
            </parameter>
        </parameters>
 </action>
```

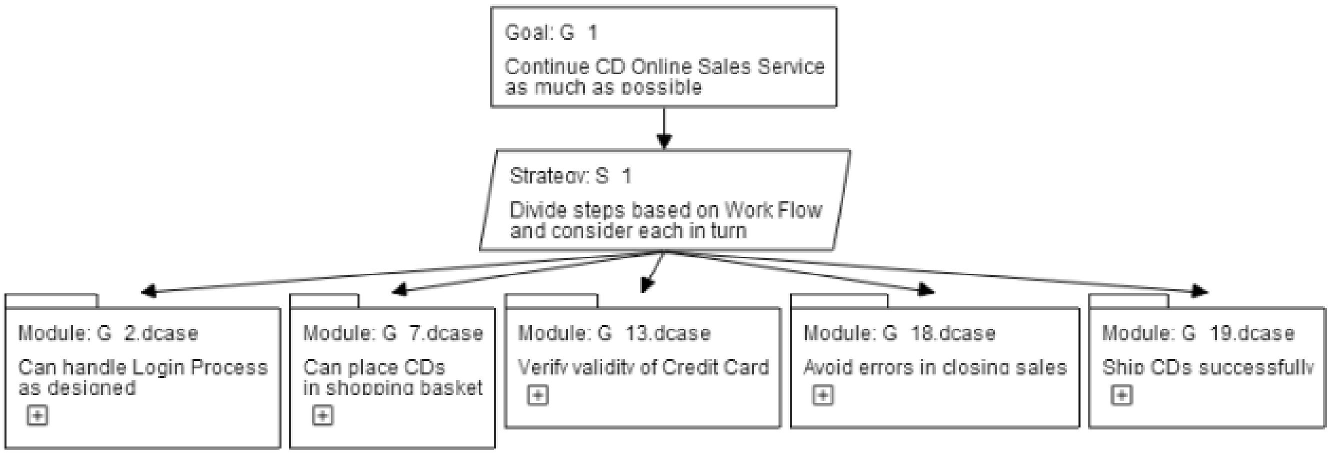
5) Deploy Memory Leak Diagnostic Module when Memory Usage exceeds the limit

```
 <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:deo
sc-laptop/SysContainer:sys-tom/tomcat>¥
EventDiscoverer=/Host:deosc-laptop/Monitor>Message=Leak Detected$1>";¥
shellcmd /usr/share/dre-demo/bin/dcase-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="en_US"><![CDATA[
               Action take for [Leak_Detected] event
               Record the name of an application on which Monitor reports a possible memory leak
problem

         ]]></explanation>
      </explanations>
      <parameters>
         <parameter name="Suppress_Window" type="integer">
               <description>Duration (in seconds) in which the same events are ignored</description>
               <default>10</default>
```

```
        </parameter>
    </parameters>
</action>
```

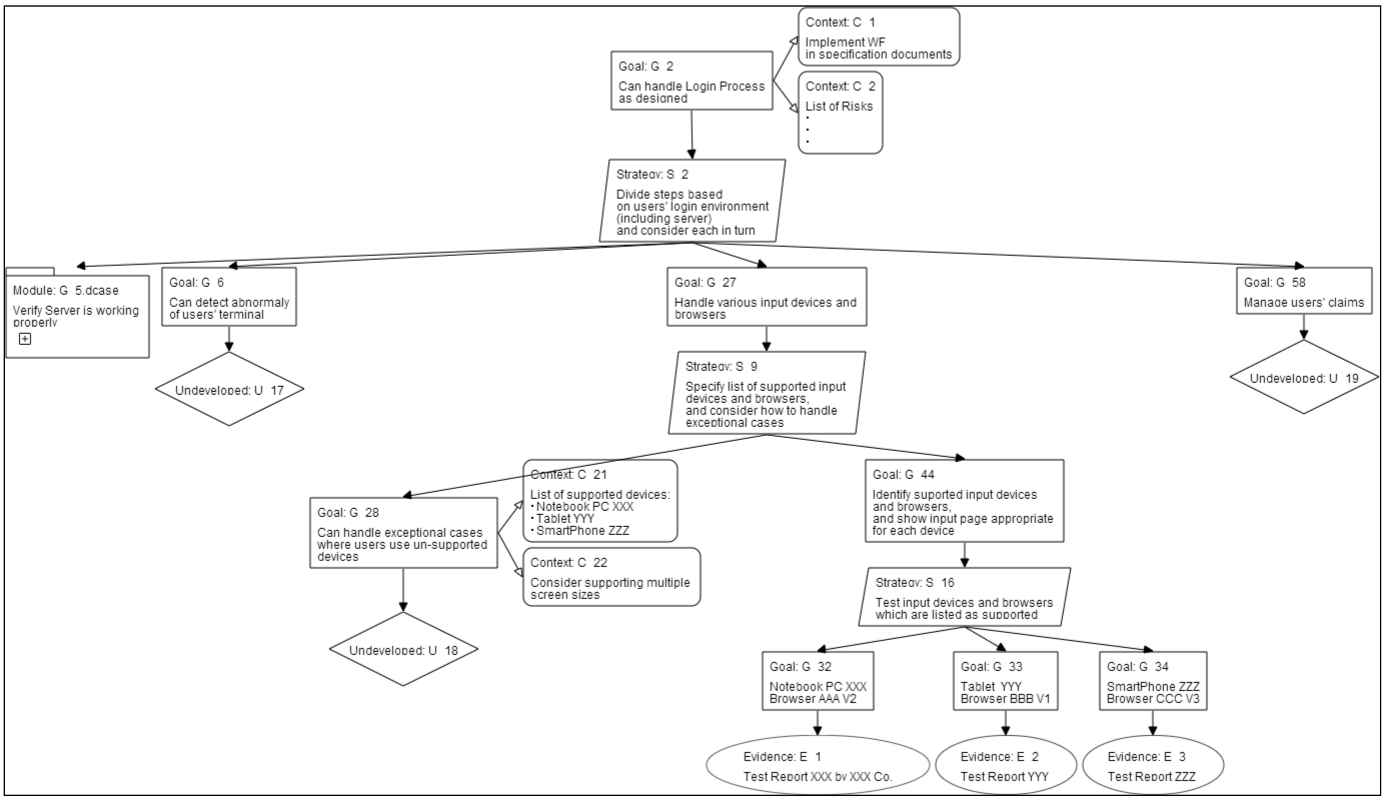
A2. D-Case of Demonstration System

Entire D-Case

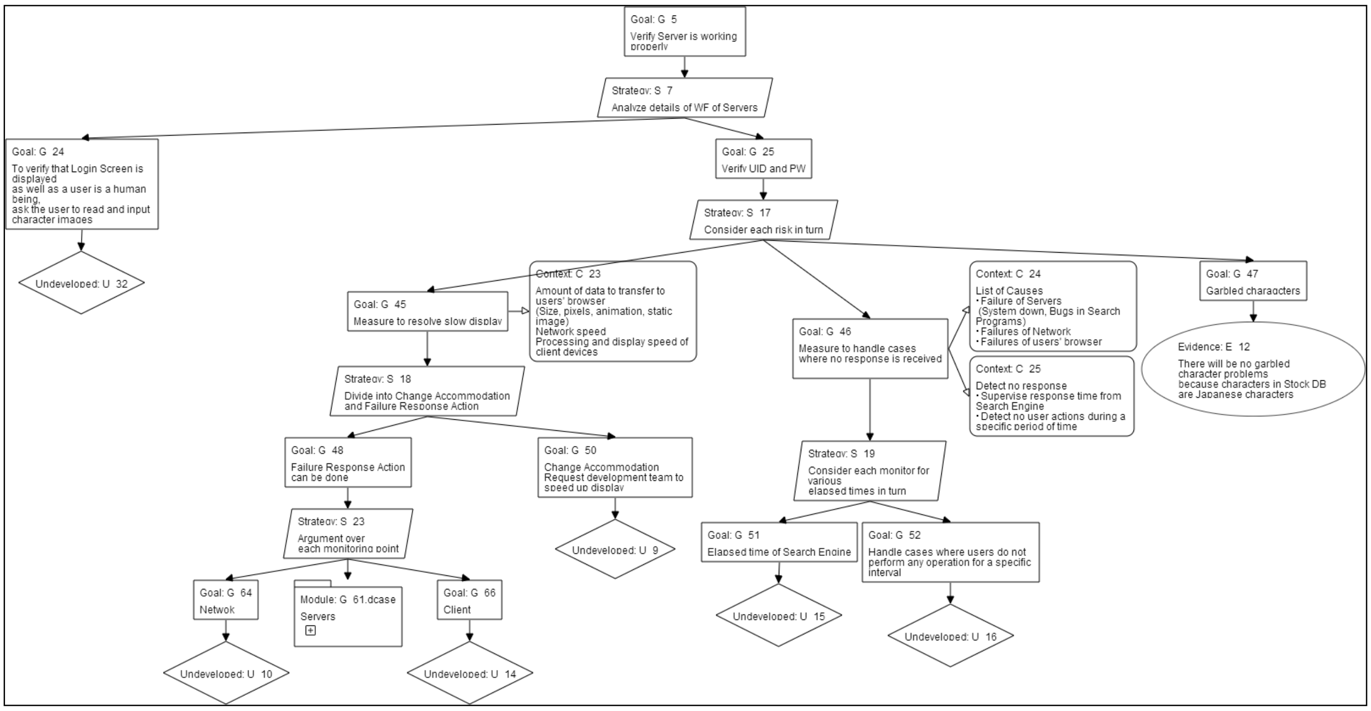

Goal: G_1
Continue CD Online Sales Service as much as possible
Strategy: S_1
Divide steps based on Work Flow and consider each in turn
Module: G_2.dcase
Can handle Login Process as designed
Module: G_7.dcase
Can place CDs in shopping basket
Module: G_13.dcase
Verify validity of Credit Card
Module: G_18.dcase
Avoid errors in closing sales
Module: G_19.dcase
Ship CDs successfully

Modules of D-Case (Partial Tree)

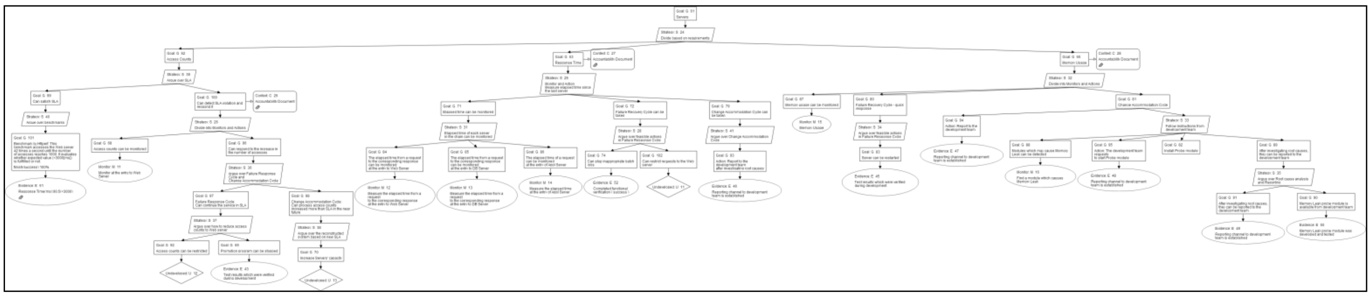
Module: G_2.dcase


Goal: G_2
Can handle Login Process as designed
Context: C_1
Implement WF in specification documents
Context: C_2
List of Risks
Strategy: S_2
Divide steps based on users' login environment (including server) and consider each in turn
Module: G_5.dcase
Verify Server is working properly
Goal: G_6
Can detect abnormaly of users' terminal
Undeveloped: U_17
Goal: G_27
Handle various input devices and browsers
Goal: G_58
Manage users' claims
Undeveloped: U_19
Strategy: S_9
Specify list of supported input devices and browsers, and consider how to handle exceptional cases
Context: C_21
List of supported devices:
・Notebook PC XXX
・Tablet YYY
・SmartPhone ZZZ
Context: C_22
Consider supporting multiple screen sizes
Goal: G_28
Can handle exceptional cases where users use un-supported devices
Undeveloped: U_18
Goal: G_44
Identify suported input devices and browsers, and show input page appropriate for each device
Strategy: S_16
Test input devices and browsers which are listed as supported
Goal: G_32
Notebook PC XXX Browser AAA V2
Goal: G_33
Tablet YYY Browser BBB V1
Goal: G_34
SmartPhone ZZZ Browser CCC V3
Evidence: E_1
Test Report XXX by XXX Co.
Evidence: E_2
Test Report YYY
Evidence: E_3
Test Report ZZZ

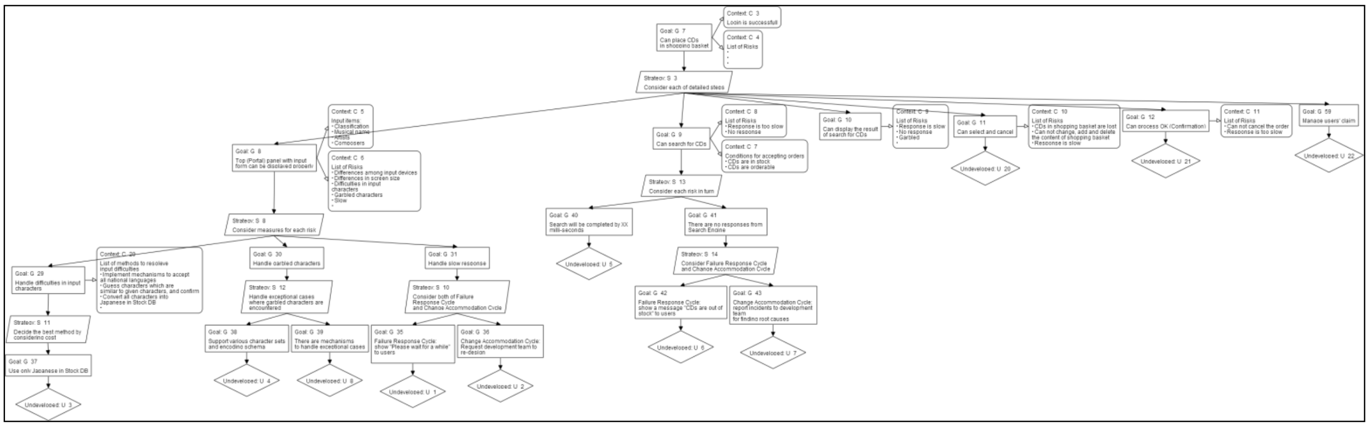
Module: G5.dcase


Goal: G_5 Verify Server is working properly
Strategy: S_7 Analyze details of WF of Servers
Goal: G_24 To verify that Login Screen is displayed as well as a user is a human being, ask the user to read and input character images
Undeveloped: U_32
Goal: G_25 Verify UID and PW
Strategy: S_17 Consider each risk in turn
Goal: G_45 Measure to resolve slow display
Context: C_23 Amount of data to transfer to users' browser (Size, pixels, animation, static image) Network speed Processing and display speed of client devices
Strategy: S_18 Divide into Change Accommodation and Failure Response Action
Goal: G_48 Failure Response Action can be done
Goal: G_50 Change Accommodation Request development team to speed up display
Undeveloped: U_9
Strategy: S_23 Argument over each monitoring point
Goal: G_64 Netwok
Module: G_61.dcase Servers
Goal: G_66 Client
Undeveloped: U_10
Undeveloped: U_14
Goal: G_46 Measure to handle cases where no response is received
Context: C_24 List of Causes • Failure of Servers (System down, Bugs in Search Programs) • Failures of Network • Failures of users' browser
Context: C_25 Detect no response • Supervise response time from Search Engine • Detect no user actions during a specific period of time
Strategy: S_19 Consider each monitor for various elapsed times in turn
Goal: G_51 Elapsed time of Search Engine
Goal: G_52 Handle cases where users do not perform any operation for a specific interval
Undeveloped: U_15
Undeveloped: U_16
Goal: G_47 Garbled characters
Evidence: E_12 There will be no garbled character problems because characters in Stock DB are Japanese characters

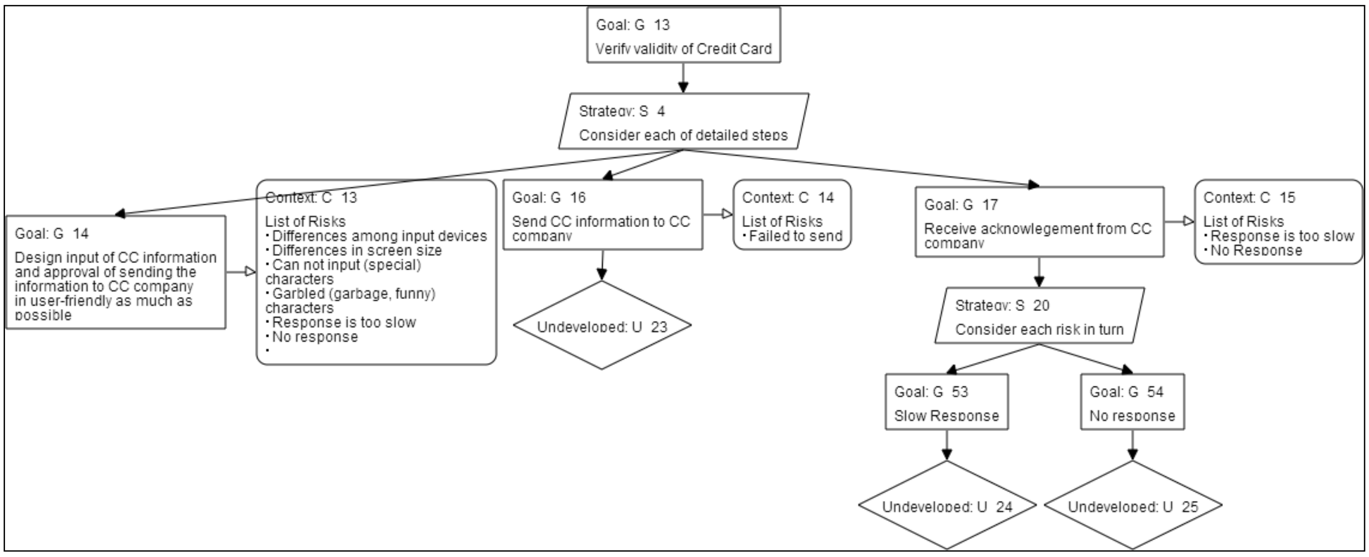
Module: G_61.dcase



Module: G_7.dcase


Goal: G 7
Can place CDs in shopping basket
Context: C 3
Login is successfull
Context: C 4
List of Risks
Strategy: S 3
Consider each of detailed steps
Goal: G 8
Top (Portal) panel with input form can be displayed properly
Context: C 5
Input items:
・Classification
・Musical name
・Artists
・Composers
Context: C 6
List of Risks
・Differences among input devices
・Differences in screen size
・Difficulties in input characters
・Garbled characters
・Slow
Strategy: S 8
Consider measures for each risk
Goal: G 29
Handle difficulties in input characters
Context: C 20
List of methods to resoleve input difficulties
・Implement mechanisms to accept all national languages
・Guess characters which are similar to given characters, and confirm
・Convert all characters into Japanese in Stock DB
Strategy: S 11
Decide the best method by considering cost
Goal: G 37
Use only Japanese in Stock DB
Undeveloped: U 3
Goal: G 30
Handle garbled characters
Strategy: S 12
Handle exceptional cases where garbled characters are encountered
Goal: G 38
Support various character sets and encoding schema
Undeveloped: U 4
Goal: G 39
There are mechanisms to handle exceptional cases
Undeveloped: U 8
Goal: G 31
Handle slow response
Strategy: S 10
Consider both of Failure Response Cycle and Change Accommodation Cycle
Goal: G 35
Failure Response Cycle: show "Please wait for a while" to users
Undeveloped: U 1
Goal: G 36
Change Accommodation Cycle: Request development team to re-design
Undeveloped: U 2
Goal: G 9
Can search for CDs
Context: C 8
List of Risks
・Response is too slow
・No response
Context: C 7
Conditions for accepting orders
・CDs are in stock
・CDs are orderable
Strategy: S 13
Consider each risk in turn
Goal: G 40
Search will be completed by XX milli-seconds
Undeveloped: U 5
Goal: G 41
There are no responses from Search Engine
Strategy: S 14
Consider Failure Response Cycle and Change Accommodation Cycle
Goal: G 42
Failure Response Cycle: show a message "CDs are out of stock" to users
Undeveloped: U 6
Goal: G 43
Change Accommodation Cycle: report incidents to development team for finding root causes
Undeveloped: U 7
Goal: G 10
Can display the result of search for CDs
Context: C 9
List of Risks
・Response is slow
・No response
・Garbled
Goal: G 11
Can select and cancel
Undeveloped: U 20
Context: C 10
List of Risks
・CDs in shopping basket are lost
・Can not change, add and delete the content of shopping basket
・Response is slow
Goal: G 12
Can process OK (Confirmation)
Undeveloped: U 21
Context: C 11
List of Risks
・Can not cancel the order
・Response is too slow
Goal: G 59
Manage users' claim
Undeveloped: U 22

Module: G_13.dcase


Goal: G 13
Verify validity of Credit Card
Strategy: S 4
Consider each of detailed steps
Goal: G 14
Design input of CC information and approval of sending the information to CC company in user-friendly as much as possible
Context: C 13
List of Risks
・Differences among input devices
・Differences in screen size
・Can not input (special) characters
・Garbled (garbage, funny) characters
・Response is too slow
・No response
Goal: G 16
Send CC information to CC company
Undeveloped: U 23
Context: C 14
List of Risks
・Failed to send
Goal: G 17
Receive acknowlegement from CC company
Context: C 15
List of Risks
・Response is too slow
・No Response
Strategy: S 20
Consider each risk in turn
Goal: G 53
Slow Response
Undeveloped: U 24
Goal: G 54
No response
Undeveloped: U 25

DEOS Project