

**GESTION ASSISTEE PAR ORDINATEUR
DES SITUATIONS DE CRISE
AFFECTANT LES INFRASTRUCTURES TECHNOLOGIQUES**

- NOTE TECHNIQUE NT3-12 -

FAISABILITE DE LA METHODE SHAMAN-X

Nom du projet: "Shaman-X"
Avancement: Phase 2, validation de l'opportunité technologique

Auteur: dominique.chabord@bluedjinn.com
Copyright: Dominique Chabord tous droits réservés
Adresse du projet: <http://www.bluedjinn.com/shaman-x>; shaman-x@bluedjinn.com
Référence: NT3-12
Version: a, datée du 27/04/03, en français.

Liste des destinataires:

La diffusion du présent document est limitée aux personnes et institutions suivantes:

Nom	Titre	Adresse	Pour Action/Info
Membres de l'équipe Shaman-X phases 2 et 3			Action

Sommaire

- 1 - Objectif de la méthode Shaman-X
- 2 - Architecture générale de l'implémentation de référence Shaman-X
- 3.- Concepts de base
- 4.- Les contraintes de l'implémentation de référence
- 5.- Les 6 dimensions de l'expression des besoins et la graduation des axes
- 6.- Les opérateurs de l'implémentation de référence
- 7.- Etude de faisabilité des opérateurs sur un tableur
- 8.- Plan type de l'étude d'une implémentation de Shaman-X

Historique des évolutions

Version	Objet	Auteur
-a	Première diffusion du document	Dominique Chabord

1 - Objectif de la méthode Shaman-X

Le projet Shaman-X a pour objet la définition d'outils de gestion des situations de crise affectant le fonctionnement des infrastructures informatiques et techniques.

Contenu:
Ce document vise à vérifier la faisabilité de l'ensemble du projet Shaman-X en posant tout d'abord les principes théoriques de la modélisation de l'infrastructure par des équations logiques (chap.3). Les contraintes qui pèsent sur les composants informatiques de l'implémentation sont précisées au chapitre 4 et illustrent en quoi la permanence du modèle impose des fonctions différentes des clusters et des produits de gestion centralisée. Au stade actuel, la méthode Shaman-X propose de définir six axes d'utilité des infrastructures et de les considérer comme indépendants. Cette simplification drastique est détaillée au chapitre 5, alors que les opérateurs de modélisation sont listés au chapitre 6 et la simulation des opérateurs est réalisée sur un tableur séparé du présent document. Il s'agit de la reprise du SPD du composant MathsDoctor. Enfin, la liste des étapes d'une étude dans le cadre Shaman-X est proposé au chapitre 8.

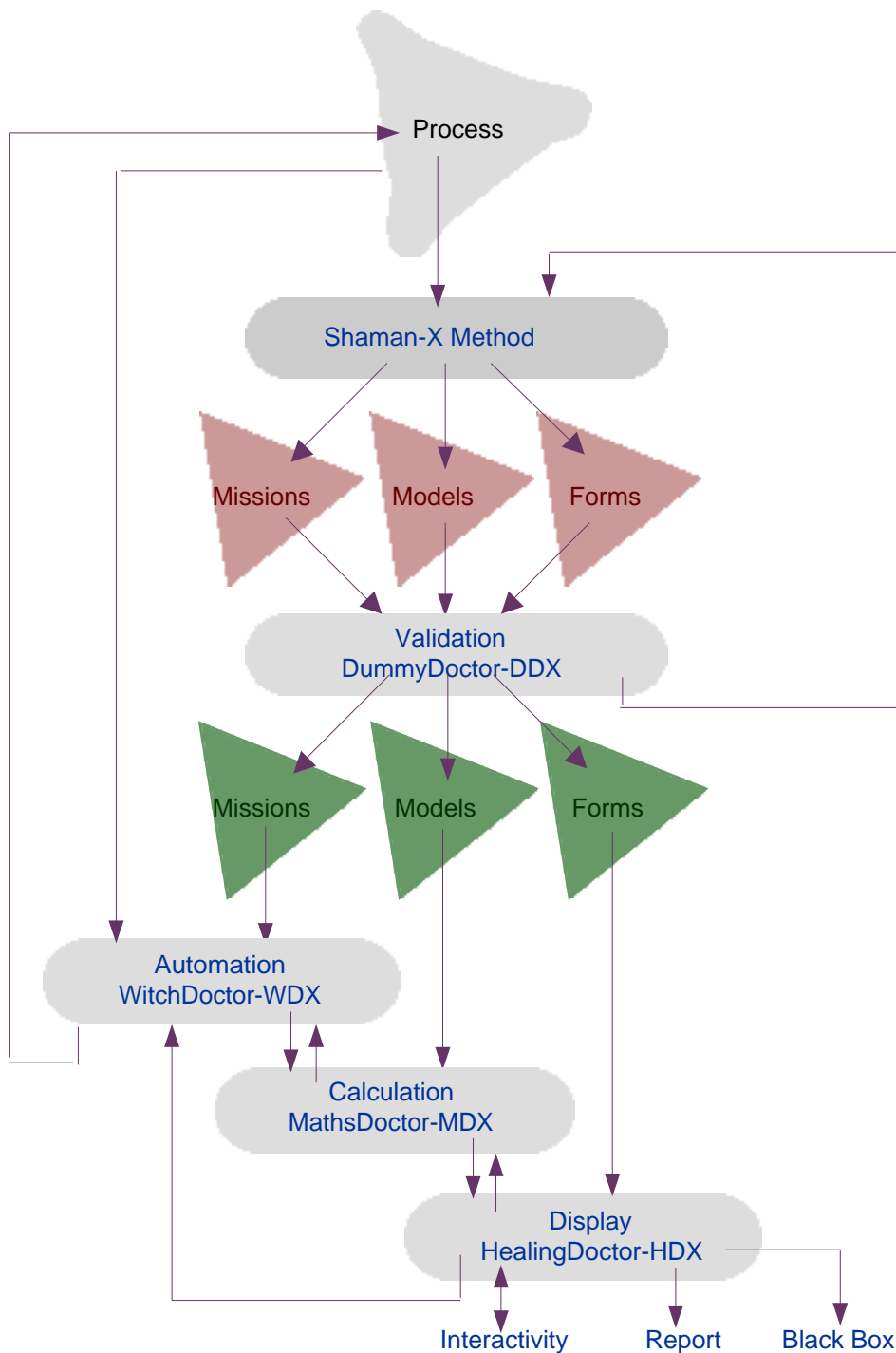
Conclusions:
La faisabilité de la décomposition fonctionnelle de l'infrastructure par des matrices de dépendance est avérée, au moins sur l'axe de la continuité de service en période normale d'exploitation. La spécification d'équations logiques de situation permet des automatisations dans des cas actuellement non couverts par les produits actuels. La pertinence de l'évaluation de gravité doit encore être démontrée sur des cas concrets. La faisabilité des composants d'implémentation, selon les contraintes exprimées ici, fait l'objet du document NT3-11.

Les résultats sont donc suffisamment prometteurs pour poursuivre le projet dans ses objectifs actuels.

2 - Architecture générale de l'implémentation de référence Shaman-X

La méthode Shaman-X permet de prendre connaissance de l'infrastructure et de la documenter. Les résultats du processus d'étude permettent de définir le paramétrage des outils informatiques Shaman-X qui implémentent une surveillance des éléments de l'infrastructure, automatisent des actions de premier niveau, calculent des modèles complexes d'analyse des états de l'infrastructure et animent des écrans et éditent des rapports destinés à une équipe de gestion de crise.

L'environnement d'exécution simulé par le logiciel DummyDoctor-DDX permet une mise en oeuvre fictive des paramètres issus de l'étude pour une validation fonctionnelle des résultats et un entraînement du personnel. La méthode Shaman-X est par conséquent un processus itératif permettant d'aboutir à des modélisations pertinentes de l'infrastructure.



3.- Concepts de base

La méthode Shaman-X étudie les conséquences de dysfonctionnements des infrastructures technologiques sur les utilisateurs.

L'utilisateur Shaman-X est à priori humain en dernier ressort, mais la notion d'utilité de l'infrastructure peut aussi s'appliquer à des choses, si on garde en mémoire les dépendances humaines ultimes. (ex.: un satellite a "besoin" d'être téléguidé, mais en fait la perte d'un satellite a des conséquences directes sur le responsable du vol, les utilisateurs des services et les vendeurs du service)

Le recensement des utilisateurs et la caractérisation de leur dépendance à l'infrastructure est donc le point de départ de l'étude Shaman-X.

N dimensions sont proposées pour caractériser la dépendance d'un utilisateur.

Pour réaliser sa mission, l'infrastructure est constituée d'un ensemble de ressources techniques, tangibles ou non tangibles, organisées de façon complexe.

La méthode Shaman-X pose tout d'abord les principes suivants:

3.1.- Analyse de l'utilité de l'infrastructure par le recours aux équations logiques

Le besoin fonctionnel d'un utilisateur (i) à un instant donné s'exprime par le vecteur Besoin(i) dont les N dimensions caractérisent la dépendance de l'utilisateur à l'infrastructure.

En conséquence, la satisfaction d'un utilisateur (i) à un instant donné (t) s'exprime par la fonction:

$$\text{Service-rendu (i)} = F[\text{service-rendu}](i)(\text{Temps}; \text{Ressource}(n); \dots; \text{Ressource}(p))$$

dans laquelle les ressources (n...p) sont des constituants matériels ou immatériels de l'infrastructure, et la variable temps fait intervenir la durée d'une perturbation ou sa fréquence.

L'utilisateur (i) est touché par un dysfonctionnement de l'infrastructure si

$$\text{Service-rendu (i)} < \text{Besoin (i)}$$

Dans ce cas, la non-satisfaction de l'utilisateur (i) est évaluable par

$$\text{Gravité (i)} = \text{Besoin (i)} - \text{Service-rendu (i)}$$

et on peut en déduire que $\text{Gravité (i)} = F[\text{gravité}](i)(\text{Temps}; \text{Ressource}(r); \dots; \text{Ressource}(s))$

En graduant les axes du besoin par des valeurs discrètes correspondant à des niveaux de gravité, la fonction $F[\text{gravité}](i)$ devient équivalente à un jeu d'expressions logiques $F\text{-logique}[\text{gravité}](j)$, combinant les états des ressources et qui exprime alors si un niveau de gravité est atteint.

Par exemple, dans le cas où l'axe d'évaluation du besoin est gradué en j niveaux:

$$\text{Niveau-de-gravité}(j)(i) = F\text{-logique}[\text{gravité}](i)(j)(\text{Temps}; \text{Ressource}(r); \dots; \text{Ressource}(s)) = \text{vrai ou faux}$$

La méthode Shaman-X vise donc dans un premier temps à déterminer les fonctions de calcul $F\text{-logique}[\text{gravité}]$ pour chaque utilisateur, afin de confronter en temps réel avec les valeurs de graduation en niveau de gravité du vecteur Besoin (i) qui peut être obtenu par exemple par l'interview de l'utilisateur (i).

Le moyen employé repose pour cela sur des matrices de dépendances successives qui permettent d'obtenir la liste des opérandes Ressource(z) pour la graduation (j) de la dimension (n) du besoin de l'utilisateur (i). On déduit alors par analyse du comportement des composants les opérateurs de la fonction $F\text{-logique}[\text{gravité}]$ et on détermine l'influence du temps à prendre en compte dans chaque calcul.

3.2. Etat vrai et situation type

A un instant donné (t), l'état fonctionnel de l'infrastructure peut être exprimé par la fonction

$$\text{Etat (t)} = F[\text{Etat}](t)(\text{Service-rendu}(i), \text{ pour } i = 1..j)$$

Cet état caractérise le niveau de fonctionnalité instantané. Son évolution au cours du temps traduit la dégradation fonctionnelle de l'infrastructure, ou son amélioration. Il peut être utile de l'élaborer comme une information très globale, mais l'état instantané ne permet pas de définir des principes opérationnels de pilotage de l'infrastructure, pour deux raisons essentielles: D'une part, l'état ne fait pas intervenir les règles d'exploitation des ressources impliquées, mais seulement leur état à un instant donné, d'autre part, du fait son évolution continue, il ne permet pas de caractériser un point de départ et un objectif d'exploitation.

C'est pour ces raisons que la méthode Shaman-X introduit des situations type qui s'expriment par

Situation(k)=F-logique[situation](k)(Temps, Ressources(l);...;Ressource(m))=vrai ou faux

Dans cette expression, les fonctions de situation F-logique[situation] sont des fonctions logiques combinant les états des ressources ou des niveaux de gravité. Une situation type doit présenter un caractère de stabilité suffisante pour permettre l'analyse, la décision et la mise en oeuvre d'une stratégie par l'organisation. Il convient de définir des situations qui regroupent des cas similaires et qui restent globalement stables quand des paramètres secondaires évoluent. Les situations doivent également être calculées à l'abri des états de transition éphémères et des paramètres non pertinents.

3.3 Le rôle du temps (durée et fréquence)

Il est facile de démontrer que la durée fait partie de l'expression du besoin des utilisateurs, et qu'elle est donc un élément indispensable des opérateurs des fonctions logiques de niveau de gravité et de situation.

Le temps intervient de trois façons:

Une situation dégradée peut rester acceptable pendant un certain temps pour l'utilisateur. C'est à dire que quand une expression change d'état et se maintient à sa nouvelle valeur pendant un certain temps, une autre expression peut changer d'état et déterminer de nouveaux résultats de niveau de gravité et de situation supérieurs du fait de la persistance des états.

Le temps intervient également pour caractériser la stabilité ou l'instabilité d'une ressource, qui se détermine par le nombre de ses changements d'états sur une période de temps. Cette instabilité peut elle aussi amener à un nouveau niveau de gravité ou une nouvelle situation car d'une part l'utilisateur voit une situation dégradée de temps en temps, d'autre part, l'instabilité d'une ressource peut entraîner l'instabilité d'autres ressources qui en dépendent.

Le temps intervient pour des raisons de conception et d'implémentation propres aux systèmes de détection notification des états des ressources. En effet, les états des ressources qui peuvent être corrélés ne sont pas forcément acquis dans une parfaite synchronisation. Afin d'obtenir des résultats pertinents et d'éviter des résultats intempestifs et fugitifs, il est nécessaire de s'assurer que toutes les informations corrélées sont bien acquises avant de lancer un calcul d'expression.

Enfin la durée est un excellent filtre permettant de masquer des états transitoires qui ne doivent pas déclencher d'action corrective et de prendre en compte des états qui deviennent permanents et qui seuls doivent être pris en compte pour asseoir une stratégie de retour à la normale.

4.- Les contraintes de l'implémentation de référence

L'implémentation des mécanismes d'élaboration des résultats des fonctions logiques employées dans les modèles décrits précédemment nécessite le développement de nouveaux outils spécialisés car elle doit satisfaire à des principes originaux qui vont à l'encontre des fonctionnalités des produits actuels. Ces principes sont au nombre de quatre et sont décrits ci dessous.

4.1 Le symbolisme des ressources et la délocalisation des composants

Les ressources et les états des ressources qui interviennent dans les fonctions logiques ne désignent pas directement des éléments matériels ou logiques de l'infrastructure. Il faut considérer qu'une ressource ou une combinaison de ressources représente l'utilité des fonctions réalisées par l'élément matériel ou logique, pour un utilisateur donné. En ce sens une ressource est représentée par une chaîne de caractère est possède les cinq états suivants: inexistant, libre, réservée, défaut, et inconnu. Seuls des états vrai et faux sont combinés dans les fonctions logiques et une correspondance des états est donc réalisée.

La détection d'un changement d'état significatif d'un élément matériel ou logique doit résulter en une signalisation des changements d'états des ressources descriptives associées. Seules ces ressources symboliques sont prises ensuite en considération.

D'autre part, les composants de l'implémentation de référence doivent pouvoir s'exécuter à n'importe quel endroit de l'infrastructure, voire commencer à s'exécuter quelque part, et poursuivre autre part. L'implémentation est alors délocalisée et déplaçable.

4.2. L'ubiquité

L'évolution des états des ressources, leur prise en compte dans des systèmes d'équation sont basées sur des mécanismes de distribution fiable de l'information, et supportent d'être répliquées dans l'infrastructure.

En conséquence, plusieurs mécanismes peuvent signaler un changement d'état d'une ressource, plusieurs modèles d'équations, identiques ou différents peuvent le prendre en compte, le résultat d'un modèle est rediffusé comme une ressource supplémentaire.

Les résultats obtenus peuvent différer selon les conditions d'exécution des mécanismes car aucun système de contrôle de cohérence n'est possible sans remettre en cause ce principe d'ubiquité. Il appartient donc à l'implémentation de référence de fournir les outils permettant de choisir un ou plusieurs résultats parmi l'ensemble des résultats disponibles.

4.3. L'immortalité

L'implémentation de référence est considérée comme un corps distribué sur l'infrastructure qui maintient l'ensemble de ses fonctions vitales propres quelque soit l'état des éléments qui la supportent. Ceci implique que l'état des ressources et l'élaboration des résultats des fonctions logiques sont toujours gérés, c'est à dire que les mécanismes sont automatiquement recrées dans les parties survivantes d'une infrastructure altérée.

Par exemple, si l'implémentation est coupée en N parties, chaque partie va recréer les mécanismes qui lui font défaut et on obtient aussitôt N implémentations complètes et valides.

En corrolaire, cette division est réversible, et N implémentations remises en contact éliminent automatiquement les doublons indésirables pour former aussitôt une seule implémentation cohérente.

4.4. L'amnésie

Le principe d'amnésie implique que chaque constituant local ne peut réagir qu'en fonction d'une part de son état courant et d'autre part de ses observations directes, donc excluant toute interprétation de données préalablement mémorisées qui ne correspondent pas à l'état instantané.

Si un modèles d'équations logiques est répliqué dans l'infrastructure, les différences de comportement des deux instances correspondent ou bien aux différences de l'état courant local, ou bien à la différence d'observation directe liée à un point de vue différent sur l'infrastructure. A aucun moment, le résultat d'une instance ne dépendra donc de l'historique des états et des observations. Sur un plan formel, les résultats apparemment contradictoires sont également vrais et relatifs aux conditions locales d'exécution et d'observation.

5.- Les six dimensions de l'expression des besoins et la graduation des axes

L'expression des besoins est relative à une classe d'utilisateurs. Exemple de classes d'utilisateurs: utilisateur des applicatifs, utilisateur client connecté, utilisateur periodique et fonctions d'administration de l'organisation, administrateur, help-desk, maintenance matérielle et applicative,...L'implémentation de référence retient à l'heure actuelle 6 dimensions indépendantes d'expression des besoins et des échelles de graduation des axes à trois niveaux. (bon-fonctionnement, tolérable, intolérable) Ces éléments sont explicités ci dessous.

5.1 Contuinité de service

Ce besoin fait référence à la disponibilité et à la réponse de l'infrastructure et de ses missions essentielles. Interviennent les notions de disponibilité et de stabilité.

L'axe comporte trois niveaux: de gravité qui sont illustrés à titre d'exemple dans les cas suivants:

- | | |
|-----------------------|---|
| - bon-fonctionnement, | le système répond à l'attente |
| - tolérable | L'indisponibilité est partielle ou totale depuis peu de temps
La stabilité de système est insuffisante depuis peu de temps |
| - intolérable: | l'indisponibilité dure au dela du seuil acceptable
L'instabilité du système se prolonge |

5.2 Continuité au cours des arrêts planifiés

Ce besoin fait référence au bon fonctionnement de l'infrastructure et de ses missions essentielles, pendant des périodes d'intervention planifiées. Interviennent les mêmes notions que précédemment, de disponibilité et de stabilité.

L'axe comporte trois niveaux: de gravité qui correspondent typiquement aux cas suivants (exemples):

- | | |
|-----------------------|---|
| - bon-fonctionnement, | le système répond à l'attente |
| - tolérable | L'indisponibilité est partielle ou totale depuis peu de temps
La stabilité de système est insuffisante depuis peu de temps |
| - intolérable: | l'indisponibilité dure au dela du seuil acceptable
L'instabilité du système se prolonge |

5.3 Transparence des dispositifs de reprise sur panne

Ce besoin fait référence aux actions que l'utilisateur doit entreprendre pour rétablir la fonctionnalité.

L'axe comporte trois niveaux: de gravité qui correspondent typiquement aux cas suivants (exemples):

- | | |
|-----------------------|--|
| - bon-fonctionnement, | le système répond à l'attente |
| - tolérable | Les actions de réparation impliquent que des instructions soient données à l'utilisateur |
| - intolérable: | L'utilisateur n'est pas en mesure d'effectuer ces opérations |

5.4 Protection des données

- | | |
|----------------------|---|
| - bon-fonctionnement | |
| - tolérable | risque de perte ou d'incohérence |
| - intolérable | faillie de protection de sécurité ou perte ou corruption réelle |

5.5 Les compétences nécessaires à l'utilisation

- bon-fonctionnement
- tolérable
- intolérable

5.6 La disponibilité des plans de secours et répartition géographique

- bon-fonctionnement
- tolérable
- intolérable

6.- Les opérateurs de l'implémentation de référence

Logical equation computing engine

The MathsDoctor-MDX-Engine computes equations represented by RESULT = EXPRESSION

where:

EXPRESSION = operator([TIMER;][PARAMETER;]LIST OF EXPRESSIONS or INPUT or OUTPUT)

operator is in the list of operators below

TIMER = ["Timer"] number of seconds ["s"]

PARAMETER = integer

input = External dynamic value

output = Result of an other equation or result of this equation at previous step

List of operators

Operators are:

and

and (A;B) is TRUE if A is TRUE and B is TRUE

or

or (A;B) is TRUE if A is TRUE or if B is TRUE

xor

xor (A;B) is TRUE if A is TRUE and B is False, or if B is FALSE and A is TRUE

not

not (A) is TRUE if A is FALSE

quorum PARAM

quorum PARAM (A;...;N) is TRUE if "PARAM" values are TRUE simultaneously in the list

Example: quorum3(A;B;C;D) is TRUE if three values or more are TRUE

inhibition

inhibition(A;B) equals A if B is TRUE, else it stays still at the last known value before B changed to FALSE

confirm TIME

confirm TIME s(A) is TRUE if A has been TRUE for "time" seconds consecutively, else it is FALSE. confirm is different from delay because changes of A to TRUE for periods shorter than TIME seconds are ignored and the result is immediately changed to FALSE if A changes to FALSE.

example: CONFIRM 5 s (A) is TRUE, if A has never been FALSE in the last five seconds, else it is FALSE.

delay TIME

delay TIME s(A) is TRUE if A was TRUE "time" seconds ago and FALSE if A was FALSE "time"seconds ago. delay TIME s (A) changes like A, shifted by TIME seconds.

stability PARAM;TIME

stability PARAM;TIME (A) is TRUE is the number of transitions of A during Time seconds is less than PARAM, else it is FALSE.

example: STABILITY 5;3600s (A) is TRUE if A did not transition more than 4 times in the last hour, and is FALSE if A had 5 or more transitions during the last hour.

7.- Etude de faisabilité des opérateurs sur un tableur

L'exercice proposé ici a pour but de contrôler que les opérateurs proposés ci dessus peuvent satisfaire aux principes d'implémentation édictés précédemment.

En particulier on vérifie que l'introduction des timers automatiques à auto-incrémentation permettent bien d'évaluer tout résultat en n'utilisant que l'état actuel de la ressource et l'état actuel du timer automatique.

On se reportera au fichier Excel séparé du présent document.

Cette règle est effectivement vérifiée par le tableur mis en place, à l'exception de l'opérateur inhibition qui pose donc problème. En effet, tel qu'il est spécifié, l'opérateur réutilise la dernière valeur avant inhibition et enfreint la règle d'amnésie. Une possibilité serait de traiter l'opérateur inhibition de façon similaire à l'opérateur delay, c'est à dire introduire un compteur que l'on incrémente à chaque changement d'état de l'entrée pendant la période d'inhibition et qu'on utilise pour déduire la dernière valeur avant inhibition de la valeur courante. Ce problème sera ré-évalué dans la suite du projet.

8.- Plan type de l'étude d'une implémentation de Shaman-X

Rappel du chapitre 3:

La méthode Shaman-X vise dans un premier temps à déterminer les fonctions de calcul F-logique[gravité] pour chaque utilisateur, afin de confronter en temps réel avec les valeurs de graduation en niveau de gravité du vecteur Besoin (i) qui peut être obtenu par exemple par l'interview de l'utilisateur (i).

Le moyen employé repose pour cela sur des matrices de dépendances successives qui permettent d'obtenir la liste des opérands Ressource(z) pour la graduation (j) de la dimension (n) du besoin de l'utilisateur (i). On déduit alors, par analyse du comportement des composants, les opérateurs de la fonction F-logique [gravité] et on détermine l'influence du temps à prendre en compte dans chaque calcul.

Organisation d'une étude et proposition du plan du document de synthèse:

- 1 - Objectif de la méthode Shaman-X
- 2 - Architecture générale de la suite Shaman-X d'implémentation de référence
- 3.- Enclenchement de l'étude
 - 3.1. Délimitations de l'infrastructure
 - 3.2. Phase d'inventaire des applications et des ressources, matérielles, logicielles et données
 - 3.3. Recensement des mécanismes de redondance, des procédures et des outils de gestion
 - 3.4. Analyse des plans de secours et de gestion de crise
 - 3.5. Documentation des évolutions en cours et à venir
- 4 - Définition des étapes de la phase d'analyse
 - 4.1. Utilisateurs et missions
 - 4.2. Ressources
 - 4.3. Fonctions de dépendance
 - 4.4. Spécification de l'utilité de l'infrastructure
 - 4.5. Modélisation de la gravité
 - 4.6. Elaboration des consignes d'exploitation en situation de crise
 - 4.7. Analyse des risques de dégradation supplémentaire
 - 4.8. Modélisation des situations
- 5 - Formalisation des résultats
 - 5.1. Nomenclature fonctionnelle et fiches techniques des ressources, situations gravités et modèles
 - 5.2. Documentation et programmation des modèles
 - 5.3. Spécification des jeux de test
- 6.- Planification et recommandations