

Étude de l'impact de changement dans les systèmes à objet par les réseaux bayésiens

Chahira CHERIF, Mustapha Kamel ABDI

Département d'Informatique, Université d'ORAN
ORAN, ALGERIE

cherifchahera@yahoo.fr, abdi.mustapha@univ-oran.dz

Résumé

L'étude de l'impact de changement est une activité fondamentale dans le génie logiciel car elle peut servir à planifier des changements, à les mettre en place et à prévoir ou détecter leurs effets sur le système et essayer de les réduire. Plusieurs travaux ont été présentés dans la littérature pour essayer de traiter cette problématique. L'objectif de ce travail est d'améliorer la maintenance des systèmes Orientés Objet (OO), et d'intervenir plus précisément dans la tâche de l'analyse et la prédiction de l'impact du changement. Parmi plusieurs modèles de représentation, les Réseaux Bayésiens (RBs) constituent une approche quantitative particulière qui peut intégrer l'incertitude dans le raisonnement, offrant ainsi des explications proches à la réalité. Dans ce papier, nous proposons une approche probabiliste afin de déterminer l'impact des changements dans les programmes à objets. Nous avons utilisé plusieurs environnements à base des réseaux bayésiens pour confirmer nos résultats obtenus avec l'outil développé dans le cadre de ce travail.

Mots-clés: Analyse d'impact de changement, maintenance, modèle probabiliste, réseaux bayésiens, inférence bayésienne.

Abstract

The study of the change impact is a fundamental activity in software engineering as it can be used to plan changes, implement them and to predict or detect their effects on the system and try to reduce them. Several works have been presented in the literature to try to treat this problem. The

objective of this work is to improve the maintenance of OO systems and to intervene specifically in the task of analyzing and predicting the change impact. Among several models of representation, Bayesian Networks (BNs) constitute a particular quantitative approach that can incorporate uncertainty in reasoning and offering explanations close to reality. In this paper, we propose a probabilistic approach to determine the change impact in object oriented programs. We used multiple environments based Bayesian network to confirm our results obtained with the tool developed as part of this work.

Keywords: *Change impact analysis, maintenance, probability model, Bayesian networks, bayesian inference.*

1. Introduction

La maintenance est la phase la plus longue et ne s'achève qu'avec la fin de vie du logiciel. Dans l'industrie, le coût de maintenance d'un logiciel (par rapport à son budget total) est passé de 40 à 60% dans les années 80, à plus de 75% ou même 80% au début de l'an 2000. On estime également que plus de la moitié de cette maintenance est consacrée à la compréhension du programme lui-même. Ce coût élevé de la maintenance et surtout de la compréhension de programmes pousse les chercheurs et les industriels à se focaliser sur cette phase du cycle de vie d'un système afin d'essayer de comprendre les facteurs qui influent ce coût. D'autre part, la programmation OO a été définie de manière à modéliser des entités du monde réel, c'est-à-dire, la définition d'objets correspondants à des entités (contrairement à la programmation procédurale qui définit une suite de fonctions censées représenter des traitements sur des données). Les effets des changements subis par le système doivent donc être pris en considération. En effet, un petit changement peut avoir des effets considérables et inattendus sur le reste des éléments du système. L'objectif de ce papier est d'améliorer la maintenance des systèmes à objets et d'intervenir plus précisément dans la tâche de l'analyse et de la prédiction de

l'impact du changement. En identifiant l'impact potentiel d'une modification, on réduit le risque d'entamer des changements coûteux et imprévisibles. Pour cela, nous essayons de donner plus d'explications sur les facteurs réels et responsables de cet impact du changement ainsi que son évolution. Il existe plusieurs modèles de représentation et les RBs constituent une approche quantitative particulière qui peut intégrer l'incertitude dans le raisonnement [1], offrant ainsi des explications proches à la réalité. La section suivante parle de l'analyse d'impact de changement et de ses principaux objectifs. La section 3 présente quelques travaux sur l'impact de changement en utilisant les réseaux bayésiens puis rappelle quelques notions de base sur ces derniers. La section 4 a été réservée à l'étude de cas dans laquelle on commence d'abord par présenter le réseau d'impact en question, puis on montre comment se fait l'affectation des paramètres de ce réseau et on décrit par la suite les outils utilisés dans cette étude empirique. Les expérimentations ainsi que les résultats obtenus sont présentés dans la section 5. Enfin, la sixième et dernière section conclut le travail réalisé dans le cadre de cet article.

2. Analyse d'impact de Changement des systèmes Orientés Objet

2.1 Analyse d'impact

L'analyse d'impact décrit comment mener, à un coût efficace, une analyse complète d'impact d'un changement dans un logiciel existant. Les mainteneurs doivent posséder une connaissance intime de la structure et du contenu du logiciel. Ils utilisent ce savoir pour faire l'analyse d'impact.

En orienté objet, les concepts comme l'encapsulation, l'héritage ou le polymorphisme rendent les systèmes plus difficiles à maintenir, de même que l'identification des parties affectées par les changements [2].

Li et Offut [3] analysent la propagation des changements sous trois angles différents : l'encapsulation, l'héritage et le polymorphisme. Ils listent les changements possibles, puis les classifient en changements de données membres, de méthodes ou de classes, et proposent par la suite un ensemble d'algorithmes afin de déterminer les classes influencées par les changements. Enfin, les points suivants illustrent pourquoi les systèmes orientés objet peuvent être difficiles à maintenir [4] :

- Même s'il est relativement facile de comprendre les structures de données et les fonctions membres des classes d'objets, la compréhension des fonctionnalités et des effets combinés n'est pas triviale.
- Les relations complexes entre les classes d'objets rendent difficiles l'anticipation et l'identification de la propagation des effets de changement.
- Les dépendances de données, les dépendances de contrôle et les dépendances de comportement d'état rendent difficiles la préparation des cas de test et la génération des données de test afin de re-tester les composants affectés par les changements.

2.2 Les Objectifs d'analyse d'impact

Parmi les principaux objectifs de l'analyse d'impact, on cite :

- Détermination de la portée d'un changement pour établir un plan et implanter le travail;
- Développement d'estimations justes des ressources nécessaires pour effectuer le travail;
- L'analyse de coûts/bénéfices du changement demandé.

L'analyse d'impact du changement est effectuée principalement lors de la maintenance des programmes, en vue d'évaluer les effets du changement sur le reste du système et de réduire le risque de s'embarquer dans des dépenses coûteuses. Cette évaluation

englobe l'estimation des ressources humaines, financières, temps, efforts et plannings nécessaires pour accomplir le changement.

3. Analyse d'impact de changement et Réseaux Bayésiens

3.1. Travaux connexes

Parmi les principaux travaux réalisés dans ce domaine on cite :

Tang et al. [5] ont utilisé des réseaux bayésiens pour l'analyse de l'impact des changements au niveau architectural. En effet, dans cet article, les auteurs présentent le modèle AREL (Architecture Rationale and Element Linkage) qui est utilisé pour présenter la relation entre les décisions à apporter et les éléments de conception qui dépendent de ces choix de modification d'architecture. Par la suite, ce modèle fut transposé en des réseaux bayésiens où les nœuds sont les éléments du AREL et les arcs sont les relations de dépendances entre ces nœuds. Les réseaux bayésiens ont fait l'objet d'une étude de cas qui consiste en une application du domaine bancaire. Les besoins fonctionnels ont été définis et traduits en AREL et ensuite en réseaux bayésiens dont les tables de probabilités (TPs) sont définies par des spécialistes.

Abdi et al. [6] se sont intéressés à la compréhension des facteurs réels qui sont responsables de l'impact de changement et de son évolution. Des métriques de conception et d'implémentation sont étudiées afin de comprendre leurs effets sur les systèmes. Les auteurs ont proposé une approche probabiliste qui utilise les réseaux bayésiens pour déterminer l'impact des changements.

Mirarab et al. [7] ont proposé une approche qui utilise deux sources d'information: les métriques de dépendances et les données historiques. Les métriques de dépendances reflètent le degré de liaison entre les paquetages du système exprimé en nombre d'appels entre les classes de chaque paquetage. Ces dépendances peuvent correspondre à des appels méthodes ou à des utilisations de variables. Pour la détection de ces dépendances, les auteurs

utilisent des techniques statiques. Le deuxième type d'information qui sont les données historiques, représente des données relatives aux co-changements dans le passé. Une fois ces données collectées, la deuxième étape consiste à construire des réseaux bayésiens pour prédire l'impact des changements. Les réseaux bayésiens utilisés dans cette approche sont de trois types :

- BDM (Bayesian Dependency Model)
- BHM (Bayesian History Model)
- BDHM (Bayesian Dependency and History Model).

Zhou et al. [8] présentent un modèle de réseaux bayésiens qui prédit la probabilité des co-changements entre les entités du système. Les informations servant de point d'entrée aux réseaux sont extraites à partir des données historiques et des dépendances dans le code et sont essentiellement : l'âge du changement, son auteur, la fréquence des changements, l'objectif des changements et d'autres informations qui servent à prédire les classes potentiellement affectées par un changement. Le processus de cette approche est divisé en trois étapes. La première consiste à collecter les données à partir des dépôts de contrôle de versions. Ces données sont transformées en une base de données de changements. La deuxième étape consiste à extraire les facteurs qui peuvent influencer les changements détectés tels que l'auteur qui les a effectués, l'entité du système qui a été modifiée ou la date du changement vu que les auteurs adhèrent à l'idée que les entités changées depuis longtemps deviennent plus stables. Ces facteurs forment ensuite les nœuds des réseaux bayésiens qui vont être, durant la dernière phase du processus, entraînés pour pouvoir prédire l'impact des changements.

Le tableau suivant résume quelques travaux relatifs à l'utilisation des réseaux bayésiens pour l'analyse de l'impact de changement :

Travaux	Techniques	Systèmes (logiciels) utilisés
Tang et al [5]	AREL + réseaux bayésiens	Architecture du système
Mirarab et al [7]	Métriques de couplage + réseaux bayésiens	Code source + historique
Cherif et Abdi [12]	Métriques de couplage + réseaux bayésiens	Code source
Zhou et al [8]	réseaux bayésiens	Données historiques

Tab.1 Synthèse de quelques travaux d'impact de changement

3.2. Rappel sur les réseaux bayésiens

Les réseaux bayésiens, qui doivent leur nom aux travaux de Thomas Bayes au dix-huitième siècle sur "la probabilité des causes", sont le résultat de recherches effectuées dans les années 80, le théorème de Bayes permet de déterminer la probabilité de A sachant B :

$$P(A/B) = \frac{P(B/A) P(A)}{P(B)}$$

Les réseaux bayésiens sont un outil de modélisation à la jonction de la théorie des probabilités et de la théorie des graphes, permettant de décrire les relations régissant un ensemble de variables aléatoires et d'effectuer un raisonnement probabiliste sur celles-ci.

Un *réseau bayésien* est défini par :

- un graphe orienté sans circuit (*DAG*) $G = (V, E)$, où V est l'ensemble des nœuds de G , et E l'ensemble des arcs de G ;
- un *espace probabilisé* fini (Ω, Z, p) ;
- un ensemble de *variables aléatoires* associées aux nœuds du graphe et définies sur (Ω, Z, p) , tel que :

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1..n} [p(V_i | C(V_i))]]$$

où $C(V_i)$ est l'ensemble des causes (parents) de V_i dans le graphe G [1].

4. Étude de cas

Dans [6], les auteurs ont proposé une approche probabiliste qui utilise les réseaux bayésiens pour déterminer l'impact des changements. Les principales étapes sur lesquelles repose l'approche sont les suivantes :

- 1- Construction de la structure du graphe (RB) à partir de connaissances pratiques (études empiriques)
- 2- Affectation des paramètres (tables de probabilités de sommets)
- 3- Inférence bayésienne (algorithmes, outils)
- 4- Résultats.

4.1. Réseau d'impact

La structure du réseau d'impact a été inspirée des travaux de Abdi et al [6] :

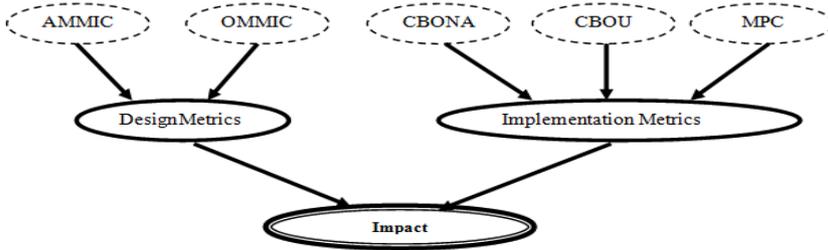


Fig.1. Réseau d’impact du changement

Le réseau d’impact de changement à été construit selon l’approche proposée par Abdi et al [6] sur la compréhension des facteurs réels qui sont responsables de l’impact de changement et de son évolution. Cinq types de métriques influencent effectivement l’impact du changement. Certaines de ces métriques sont considérées comme des métriques de conception (AMMIC et OMMIC), car pouvant être utilisées dès la phase de conception, alors que d’autres, sont des métriques d’implémentation (MPC, CBOU, CBONA), car nécessitant la présence du code source.

Les métriques sélectionnées sont définies dans la table Tab.2 suivante :

Métrique	Définition
CBOU	CBO Using : se réfère aux classes utilisées par la classe cible
CBONA	CBO No Ancestors : CBO sans considérer les classes ancêtres.
MPC	Message Passing Coupling : nombre de messages envoyés par une classe en direction des autres classes du système.
OMMIC	Others Method–Method Import Coupling : nombre de classes (autres que les super-classes et les sous-classes) avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type IC (Import Coupling).
AMMIC	Ancestors Method–Method Import Coupling : nombre de classes parentes avec lesquelles une classe a une interaction de type méthode-méthode et un couplage de type IC (Import Coupling).

Tab.2 Les métriques utilisées dans le réseau d’impact

4.2. Affectation des paramètres

Les nœuds du réseau bayésien sont donc décomposés en deux catégories : les variables d'entrée et les variables intermédiaires. Les probabilités des sommets d'entrée sont déterminées directement à partir des mesures de ces variables données par un système de test. Dans notre cas, il s'agit de BOAP (Boîte à Outils pour l'Analyse de Programmes) développé au Centre de Recherche Informatique de Montréal (CRIM) par Alikacem et Snoussi [9]. C'est un ensemble d'outils logiciels intégrés, qui permet à un expert d'évaluer rapidement le niveau de qualité d'un logiciel (faiblesses conceptuelles ou structurelles, instructions trop complexes, etc.). Le système BOAP (version 1.1.0) contient en tout 394 classes. Les métriques considérées dans ce travail sont extraites de ce système.

- **Les sommets d'entrée :** ces sommets représentent les différentes métriques sélectionnées. Toutes ces variables d'entrée sont des variables quantitatives qui ont des valeurs numériques mesurables. Le nombre de valeurs possibles pour ces variables peut être infini, en fonction du système de test considéré. Afin de faciliter la définition des probabilités des sommets d'entrée, nous avons besoin de transformer ces variables en variables discrètes ayant un nombre limité de valeurs qui sont : « petit » et « grand ». Le sommet MPC est illustré par la table Tab.3, MPC a une probabilité de 20% d'être «Petit», et une probabilité de 80% d'être «Grand ».

Petit	20%
Grand	80%

Tab.3 Table de Probabilité du sommet d'entrée MPC.

- **Les sommets intermédiaires :** ces sommets représentent les deux nœuds « DesignMetrics, ImplementationMetrics », ces

nœuds ne sont pas directement mesurables. Ils sont définis ou influencés par leurs sommets parents. Les sommets intermédiaires ont une table de probabilité associée.

DesignMetrics est définie par ses deux parents AMMIC et OMMIC et le nœud ImplementationMetrics est le nœud fils des trois nœuds d'entrée MPC, CBOU et CBONA.

Considérons le nœud ImplementationMetrics et calculons sa probabilité conditionnelle :

$$P(\text{ImplementationMetrics}/\text{MPC,CBOU,CBONA})$$

Or comme la relation entre les sommets parents MPC, CBOU & CBONA et leur sommet fils ImplementationMetrics est définitionnelle, la forte présence de ces métriques définit également la forte présence des métriques d'implémentation (ImplementationMetrics). Le tableau Tab.4 suivant illustre un scénario pour la TP du sommet ImplementationMetrics :

MPC	Petit				Grand			
CBOU	Petit		Grand		Petit		Grand	
CBONA	Petit	Grand	Petit	Grand	Petit	Grand	Petit	Grand
Oui	20%	30%	30%	40%	30%	40%	40%	80%
Non	80%	70%	70%	60%	70%	60%	60%	20%

Tab.4 Table de Probabilité du sommet intermédiaire ImplementationMetrics.

4.3. Outils utilisés

Les réseaux bayésiens ont été utilisés comme un outil fondamental pour la représentation et la manipulation des croyances en intelligence artificielle. Il y a eu des implémentations de réseaux bayésiens dans une variété de formats.

- IBN « *Impact Bayes Net* »

IBN est un système qui gère les réseaux bayésiens: il calcule les probabilités marginales et les attentes, et permet à l'utilisateur d'importer, créer, modifier et exporter des réseaux [12].

IBN est une implémentation complète des réseaux bayésiens en Java. Une implémentation en Java présente plusieurs avantages. Tout d'abord, Java est encore le meilleur comme langage véritablement portable, ie, un paquet écrit en Java peut être exporté et exécuté sous Unix, Macintosh et Windows.

Java a été adopté par les navigateurs de l'Internet, un programme ou une application écrite en Java peut être intimement liée avec les pages web. Java est un bon langage orienté objet, et dispose d'une fonctionnalité multi-thread, qui peut être très utile pour le parallélisme.

Le système *IBN* est un ensemble d'outils pour la création et la manipulation de réseaux bayésiens. Le système est composé d'un éditeur graphique, qui vous permet de créer et de modifier des réseaux bayésiens dans une interface conviviale. Il permet aussi d'importer des réseaux bayésiens dans une variété de formats. Il peut produire:

- la probabilité marginale pour une variable dans un réseau bayésien.
 - la configuration avec un maximum de probabilités à posteriori.
- BNJ « Bayesian Network tools in Java »

BNJ¹ est un ensemble d'outils Java de recherche et de développement des réseaux bayésiens. Ce projet a été développé au sein du laboratoire KDD de l'université du Kensas. C'est un

¹ : <http://bnj.sourceforge.net>

projet Open source distribué sous la licence GNU (General Public Licence). Il est écrit en Java et est disponible sur le web. Cet outil utilise l'algorithme d'arbre de jonction pour l'inférence bayésienne.

- BAYESIALAB

BayesiaLab² est un laboratoire complet de manipulation de réseaux bayésiens qui permet d'élaborer des modèles décisionnels par recueil d'expertise et automatiquement à partir des données, d'assimiler rapidement des connaissances représentées grâce à une boîte à outils d'analyse originale, d'exploiter des modèles en mode interactif ou par lots et faire l'apprentissage des politiques d'actions.

- HUGIN³

La société danoise Hugin Expert A/S, qui édite ce logiciel, a été créée en 1989 et est située à Aalborg au Danemark. La société a été créée suite à un projet ESPRIT, qui avait pour but de développer des systèmes experts de diagnostic dans le domaine médical. Hugin³ est un outil de construction des réseaux bayésiens, probablement le plus connu et le plus utilisé commercialement. Cet outil présente les fonctions principales suivantes :

- Construction de bases de connaissance fondées sur des réseaux bayésiens ou des diagrammes d'influence ;
- Développement de réseaux bayésiens orientés objets ;
- Apprentissage de structure et de paramètres.

5. Expérimentations et Résultats

5.1 Expérimentations par l'outil IBN

- **Algorithme d'inférence sous IBN**

Nous avons utilisé l'algorithme d'inférence de l'approche exacte, qui exploite les indépendances conditionnelles contenues dans les

² : <http://www.bayesia.com>

³ : <http://www.hugin.com>

réseaux et donnent à chaque inférence les probabilités à posteriori exactes. L'algorithme de l'arbre de jonction est détaillé comme suit :

- **Arbre de jonction**

La méthode de l'arbre de jonction a été introduite par Lauritzen & Spiegelhalter [10] et Jensen et al. [11]. Elle est aussi appelée méthode JLO (pour Jensen, Lauritzen, Olessen). Elle est applicable pour toute structure de DAG contrairement à la méthode des messages locaux.

Néanmoins, s'il y a peu de circuits dans le graphe, il peut être préférable d'utiliser une méthode basée sur un ensemble de coupes. Cette méthode est divisée en cinq étapes qui sont :

- moralisation du graphe,
- triangulation du graphe moral,
- construction de l'arbre de jonction,
- inférence dans l'arbre de jonction en utilisant des messages locaux,
- transformation des potentiels de clique en lois conditionnelles mises à jour.

Une fois la structure du graphe et toutes les TPs définies, nous pouvons procéder à l'inférence bayésienne. Il en résulte une mise à jour de toutes les probabilités conditionnelles de tous les sommets. Après la mise en œuvre du réseau d'impact en utilisant les tables de probabilité obtenues dans l'étude de Abdi et al. [6], nous avons appliqué l'algorithme d'inférence (arbre de jonction) sous l'éditeur bayésien IBN « Impact Bayes Net ».

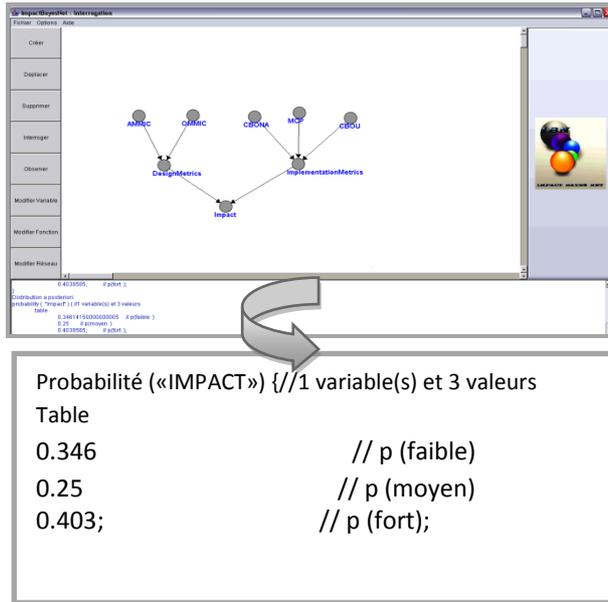


Fig.2. Réseau d'Impact du changement après inférence (Scénario 1)

Nous avons affecté trois états « Faible », « Moyen » et « Fort » au sommet Impact, et nous avons défini les valeurs (de probabilités) des données d'entrée, utilisées dans les quatre scénarios illustrés dans la table Tab.5 suivante :

Scénarios	Scénario 1		Scénario 2		Scénario 3		Scénario 4	
	Petit	Grand	Petit	Grand	Petit	Grand	Petit	Grand
AMMIC	0.35	0.65	0.5	0.5	0.05	0.95	0.05	0.95
OMMIC	0.6	0.4	0.5	0.5	0.5	0.5	0.5	0.5
CBONA	0.75	0.25	0.9	0.1	0.05	0.95	0.9	0.1
CBOU	0.45	0.55	0.9	0.1	0.05	0.95	0.9	0.1
MPC	0.2	0.8	0.5	0.5	0.5	0.5	0.5	0.5

Tab.5 Table de probabilités des nœuds d'entrée [12]

L'exécution du réseau sur ce jeu de données sous IBN a généré ces estimations d'impact (valeurs de probabilités du sommet Impact). Nous pouvons conclure par exemple que l'impact de changement a 40.38% de chances d'être « Fort » au premier scénario (Tab.6).

Dans le scénario 2, en diminuant les valeurs des métriques CBONA et CBOU, l'impact de changement s'affaiblit de plus en plus (sa probabilité d'être «faible» passe de 34.61 % à 37.80 %). Dans le 3^{ème} scénario, en augmentant cette fois-ci les valeurs des métriques CBONA et CBOU (Tab.5), l'impact de changement devient de plus en plus fort. La probabilité de l'état « Fort » passe de 37,19% à 44,41% avec un gain de plus que 7 points (Tab.6).

Enfin, le dernier scénario (scénario 4) exécuté montre qu'en diminuant les valeurs des métriques CBONA et CBOU et en augmentant la valeur de AMMIC, l'impact du changement devient moins fort. La probabilité de l'état « Fort » passe de 44,41% à 41,01%.

IBN			
Scénarios	Impact		
	Faible	Moyen	Fort
Sénarios1	34.61%	25%	40.38%
Sénarios2	37.80%	25%	37.19%
Sénarios3	30.58%	25%	44.41%
Sénarios4	33.99%	25%	41.01%

Tab.6 Résultats d'impact après inférence :
"Les quatre scénarios » sous IBN"

5.2 Expérimentations par les autres outils

Nous avons fait appel aux environnements BNJ, BAYESIALAB et HUGIN pour comparer les résultats obtenus par le biais de ces

outils avec nos résultats obtenus sous IBN. Avant de présenter les résultats de nos expérimentations avec ces outils, nous préférons décrire brièvement comment se fait l'inférence dans chacun de ces derniers :

- **Inférence dans BNJ** : Pour la définition des réseaux bayésiens dans l'environnement de BNJ, l'utilisateur peut utiliser l'interface graphique de ce système. Il est possible de définir deux types de distribution de probabilité pour les nœuds : distribution tabulaire discrète et distribution continue. Les réseaux bayésiens créés sont stockés dans des fichiers XML. BNJ fournit un ensemble d'algorithmes d'inférence pour les réseaux bayésiens. Ces algorithmes se classent en deux catégories : inférence exacte et inférence approchée.

- Les algorithmes d'inférence exacte développés sont : "Arbre de Jonction", "Elimination des variables avec optimisation", "Singly-connected network belief propagation" (Pearl) et "Cutset Conditioning".
- Les algorithmes d'inférence approchée développés sont : certaines méthodes utilisent la notion d'échantillonnage tel que "Adaptive Importance Sampling (AIS)", "Logic Sampling" et "Forward Sampling", d'autres méthodes appliquent les algorithmes d'inférence exacte sur une sélection d'arcs du graphe à traiter tels que "KruskalPolytree", "BCS" et "PTReduction".

- **Inférence dans BayesiaLab** : BayesiaLab gère deux types d'inférence : exacte (basée sur l'algorithme de l'arbre de jonction) et une inférence approchée lorsque les réseaux sont de complexité trop grande. L'approximation peut se faire soit par échantillonnage stochastique (Likelihood Weighting), soit par inférence exacte sur un graphe simplifié (suppression des relations les plus faibles et causant la plus grande complexité). Pour les réseaux de grande taille, un mode d'inférence exacte basé sur les requêtes est également disponible (relevance reasoning). Ce mode permet, par

l'analyse des observations et des nœuds requêtés, de construire l'arbre de jonction minimal. Enfin, L'exploitation nécessite la possibilité d'insérer des observations dans le réseau. BayesiaLab permet d'insérer des évidences certaines ou négatives (un nœud a cette valeur ou n'a pas cette valeur), des vraisemblances (une valeur entre 0 et 100 sur chaque modalité), et des distributions de probabilités.

- **Inférence dans HUGIN** : L'inférence dans Hugin s'effectue grâce au calcul d'un arbre de jonction sur le réseau. Le mode le plus simple d'inférence consiste à entrer des observations dans le réseau, simplement en cliquant sur la valeur observée. Le type d'inférence standard, c'est-à-dire le calcul de la probabilité des nœuds non observés conditionnellement aux observations, s'appelle la propagation Sum normal dans Hugin, qui offre d'autres modes d'inférences. En particulier, la propagation Max normal permet de trouver la configuration du réseau la plus probable, ayant effectué certaines observations.

Les résultats obtenus par les trois outils BNJ, BayesiaLab et Hugin dans les quatre scénarios reproduits de même qu'auparavant (ie, de même qu'avec l'outil IBN (section 5.1)) sont résumés dans le tableau suivant :

Scenarios	Outils Bayésiens	Impact		
		Faible	Moyen	Fort
Scenario 1	BNJ	34.6 %	25%	40.40%
	BAYESIALAB	34.61%	25%	40.39%
	HUGIN	34.61%	25%	40.38%
Scenar no.2	BNJ	37.8 %	25%	37.20%
	BAYESIALAB	37.80%	25%	37.20%

Scenario 3	HUGIN	37.80%	25%	37.19%
	BNJ	30.6 %	25%	44.40%
	BAYESIALAB	30.58%	25%	44.42%
Scenario 4	HUGIN	30.58%	25%	44.41%
	BNJ	34.0 %	25%	41.00%
	BAYESIALAB	33.99%	25%	41.01%
	HUGIN	33.99%	25%	41.00%

Tab.7 Résultats d'impact après inférence :
"Les quatre scénarios sous les autres outils"

Dans les tableaux ci-dessus (Tab.6 et Tab.7), nous remarquons que dans les quatre scénarios, la probabilité d'avoir un impact "moyen" est stable (égale à 25%) pour les quatre outils (IBN, BNJ, BayesiaLab, Hugin). De plus, les résultats de l'inférence obtenus sous IBN et les autres outils sont proches. Par exemple, en scénario 1, la probabilité d'avoir un impact "faible" est en moyenne 34,61% alors qu'en scénario 4, la probabilité pour que l'impact soit "fort" est de 41,01%. Cette convergence dans les résultats obtenus par les quatre outils est à notre avis tout à fait acceptable vu que ces derniers utilisent l'inférence exacte à base de l'algorithme d'arbre de jonction à quelques variantes près.

6. Conclusion

Nous avons proposé dans cet article une approche probabiliste utilisant les réseaux bayésiens afin de répondre à la problématique d'analyse d'impact de changement. Nous avons utilisé un modèle proposé par Abdi et al [6]. Ce modèle est un réseau bayésien composé d'un ensemble de nœuds d'entrée qui sont des métriques de conception et d'implémentation, étudiées afin de comprendre leurs effets sur les systèmes logiciels. Nous avons comparé par la suite les résultats d'inférence sous BNJ vs IBN, Bayesilab et Hugin pour confirmer nos résultats obtenus dans le cadre de cette étude.

Nous cherchons à donner plus d'explications sur les facteurs réels et responsables de cet impact du changement ainsi que de son évolution. Enfin, nous pensons que l'exploitation des capacités d'apprentissage automatique qu'offrent les réseaux bayésiens nous permettra d'avoir dans une perspective à court terme une meilleure précision de la prédiction et ainsi des résultats plus convaincants.

Références

- [1] P. Naïm, P. Wuillemin, P. Leray, O. Pourret, & A. Becker , "Les Réseaux Bayésiens", Edition Eyrolles, 2004.
- [2] M. L. Lee, "Change Impact Analysis for Object-Oriented Software". PhD thesis, George Mason University, Virginia, USA, 1998.
- [3] M. L. Lee and A. J. Offutt. "Algorithmic Analysis of the Change Impact to Object-Oriented Software" in ICSM96, pages 171-184, 1996.
- [4] D. Kung, J. Gao, P. Hsia, F. Wen, Y. Toyoshima and C. Chen, "Change Impact Identification in Object Oriented Software Maintenance" in ICSM94, Victoria, B.C., Canada, pages 202-211, September 1994.
- [5] A. Tang, A. Nicholson, Y. Jin, & J. Han, "Using bayesian belief networks for change impact analysis in architecture design" in Journal of System Software, 80: pages 127-148, January 2007.
- [6] M. K. Abdi, H. Lounis, and H. A. Sahraoui. "Predicting change impact in object-oriented applications with bayesian networks" in the 33rd Annual IEEE International Computer Software and Application Conference, COMPSAC'2009, Seattle, Washington, July 20-24, 2009, pages 234-239.
- [7] S. Mirarab, A. Hassouna, & L. Tahvildari. "Using bayesian belief networks to predict change propagation in software systems" in Proceedings of the 15th IEEE International Conference on Program Comprehension, pages 177-188, 2007.
- [8] Y. Zhou, M. Würsch, E. Giger, H. C. Gall, & J. Lü, "A bayesian network based approach for change coupling prediction" in



Proceedings of the 15th Working Conference on Reverse Engineering, pages 27-36, Washington, DC, USA, 2008, IEEE Computer Society.

[9] E. H. Alikacem, & H. Snoussi, " BOAP 1.1.0. Manuel d'utilisation", CRIM, Montréal, 2002

[10] S. L. Lauritzen & D. J. Spiegelhalter "Local computations with probabilities on graphical structures and their application to expert Systems" in Journal of the Royal Statistical Society B, 50(2), 157-224. 1988.

[11] F. V. Jensen, S. L. Lauritzen & K. G. Olesen, "Bayesian updating in causal probabilistic networks by local computations", in Computational Statistics Quaterly", 4, 269-282. 1990.

[12] C. Cherif , & M.K. Abdi, "Change Impact Study by Bayesian networks" in Modeling Approches and Algorithms for Advanced Computer Applications, Series "Studies in Computational Intelligence", Volume 488, Springer, pp 429-438, 2013.