La gestion des incertitudes dans les modèles à base de cas en estimation des coûts de logiciels

Ali Idri* - Alain Abran**

* ENSIAS, BP. 713, Agdal, Rabat
Université Mohamed V-Souissi
idri@ensias.ma

** École de technologie supérieure,
Montréal, Canada
aabran@ele.etsmtl.ca

RÉSUMÉ. Fuzzy Analogy est une approche d'estimation des coûts de logiciels que nous avons développée pour remédier à la problématique d'utilisation des valeurs linguistiques dans la description des projets logiciels. Le modèle Fuzzy Analogy est basé sur la technique CBR (Case-Based Reasoning) et la logique floue pour représenter et traiter convenablement le cas des valeurs linguistiques. Cependant, Fuzzy Analogy ne permet pas encore la gestion des incertitudes au niveau de ses estimations du fait qu'elle génère seulement une seule valeur numérique du coût estimé. Cet article présente une stratégie pour la gestion des incertitudes au niveau des estimations des coûts fournies par Fuzzy Analogy. La validation de cette stratégie est basée sur les projets logiciels du COCOMO'81.

ABSTRACT. In an earlier study, we have proposed an innovative approach referred to as Fuzzy Analogy to estimate the cost of software projects when they are described by linguistic values. Fuzzy Analogy is based on a fuzzy CBR technique to deal adequately with linguistic values. In this paper, we investigate the uncertainty of cost estimates generated by the Fuzzy Analogy approach. The primary aim is to generate a set of possible values for the actual software development cost. The proposed idea is validated by using the COCOMO'81 software projects.

MOTS-CLÉS : Estimation des coûts de logiciels, Raisonnement à base de cas, Logique Floue.

KEYWORDS: Software cost estimation, Case-Based Reasoning, Fuzzy Logique, Uncertainty

1. Introduction

Le raisonnement à base de cas (*Case-Based Reasoning* –CBR) est une version simplifiée du raisonnement par analogie très connu en intelligence artificielle. Il retrouve ses origines dans les travaux de Schank sur la mémoire dynamique et ceux de Gentner sur le raisonnement par analogie [6,11]. Depuis, la technique CBR a été développée simultanément par plusieurs groupes de recherche en Amérique et en Europe [1,9]. L'idée de l'utilisation de la technique CBR en estimation des coûts n'est pas nouvelle. Boehm avait déjà cité, dans son ouvrage *Software Engineering Economics*, l'estimation par analogie comme étant une technique plausible pour la prédiction des coûts de développement de logiciels [3]. Cependant, Boehm n'avait pas proposé une version formelle de l'estimation par analogie. En 1990, Vicinanza et Prietula ont proposé une reformulation du processus de la technique CBR afin d'être appliqué en estimation des coûts. Cette reformulation est basée sur l'affirmation suivante: *similar software projects have similar costs*[12].

L'estimation des coûts de logiciels par la technique CBR est une alternative prometteuse qui s'adapte bien au problème de la prédiction des coûts de projets logiciels. Récemment, nous avons proposé une nouvelle version d'estimation par analogie, nommée Fuzzy Analogy [6]. Fuzzy Analogy incorpore la logique floue pour gérer les imprécisions engendrant l'utilisation des valeurs linguistiques tout le long du processus CBR. Ainsi, notre approche permet de traiter convenablement le cas des projets logiciels décrits par des valeurs linguistiques souvent floues et imprécises.

Jusqu'à présent, Fuzzy Analogy, ainsi que presque tous les modèles d'estimation basés sur la technique CBR, fournissent une seule valeur numérique du coût estimé. Par conséquent, si cette valeur est largement différente du coût nécessaire pour le développement du logiciel, les gestionnaires pourront commettre des erreurs fatales au niveau de la gestion du projet logiciel. Il est donc préférable que ces modèles génèrent un ensemble de valeurs possibles, plutôt qu'une seule valeur du coût estimé. Dans cet article, nous proposons une stratégie pour résoudre la problématique de la gestion des incertitudes dans Fuzzy Analogy.

Cet article est composé de six sections. Dans la deuxième section, nous présentons les étapes composant le processus d'estimation de Fuzzy Analogy. La troisème section porte sur l'identification des sources d'incertitudes dans l'approche Fuzzy Analogy. La quatrième section décrit l'expérimentation que nous avons menée afin d'évaluer la nature de l'affirmation de base de la technique CBR (déterministe ou non-déterministe) en estimation des coûts. La cinquième section présente notre stratégie pour la gestion des incertitudes dans Fuzzy Analogy. La sixième section discute des résultats obtenus ainsi que des perspectives de ce travail.

2. Estimation du coût par Fuzzy Analogy

Depuis que Vicinanza et Prietula. ont proposé une version formelle de l'estimation des coûts par analogie, plusieurs travaux de recherche ont été entrepris afin d'améliorer, reformuler, et /ou appliquer cette technique dans plusieurs environnements de développement de logiciels [11,12]. Récemment, nous avons développé une nouvelle approche d'estimation, Fuzzy Analogy, qui permet la tolérance des imprécisions tout le long du processus CBR [6]. Fuzzy Analogy est une «fuzzification» de la procédure classique d'estimation des coûts par analogie. Son processus d'estimation est composé, comme dans le cas de la procédure classique d'estimation par analogie, de trois étapes:

- Dans l'étape d'Identification, les valeurs linguistiques décrivant les projets logiciels sont représentées et traitées par des éléments de la logique floue contrairement aux cas des modèles CBR déjà mis au point dans le domaine d'estimation des coûts.

- Dans l'étape d'Évaluation de la similarité entre les projets logiciels, Fuzzy Analogy utilise de nouvelles mesures de similarités que nous avons développées pour traiter convenablement le cas des projets logiciels décrits par des valeurs linguistiques. La similarité entre deux projets logiciels P_1 et P_2 est évaluée par:

$$d(P_1,P_2)=\begin{cases} all\ of\ (d_{v_j}(P_1,P_2)) \\ most\ of\ (d_{v_j}(P_1,P_2)) \\ many\ of\ (d_{v_j}(P_1,P_2)) \\ .... \\ there\ exists\ of\ (d_{v_j}(P_1,P_2)) \end{cases} \qquad \text{(Equation 1)}$$

où $d_{v_j}(P_1,P_2)$ est la distance selon la variable v_j entre P_1 et P_2; *all*, *most*, *many* et *there exists* sont des quantificateurs linguistiques.

- Dans l'étape d'Adaptation, Fuzzy Analogy utilise une nouvelle stratégie pour déduire le coût d'un nouveau projet à partir des coûts réels des projets logiciels les plus similaires au nouveau projet.

3. Sources d'incertitude dans le processus de Fuzzy Analogy

La gestion des incertitudes au niveau des estimations de coûts fournies par Fuzzy Analogy signifie qu'elle doit générer un ensemble de valeurs estimées, plutôt qu'une seule, au coût d'un projet avec une distribution de possibilités. Cette fonction de distribution indique les degrés de certitude associés aux différentes valeurs possibles du coût d'un projet logiciel. Kitchenham et Linkman ont examiné quatre sources

d'incertitude dans un modèle d'estimation des coûts: les erreurs de mesurage des attributs affectant le coût, les erreurs du modèle, les erreurs relatives aux hypothèses faites sur le modèle et les erreurs relatives aux caractéristiques de l'environnement à partir duquel le modèle a été mis au point [8]. Les deux premières sources d'incertitudes sont dépendantes respectivement de la précision des mesures des attributs décrivant le projet logiciel et de la précision des estimations du modèle. Les deux autres sources d'incertitude sont attachées aux hypothèses faites sur les entrées inconnues du modèle au moment de la formulation d'une estimation au coût d'un projet. Dans cet article, nous étudions seulement les incertitudes relatives aux erreurs des estimations de Fuzzy Analogy. En effet, nous avons montré que la première source d'incertitude concernant les erreurs de mesurages des attributs n'est pas très conséquente quand les attributs sont évalués par des valeurs linguistiques [8]; quant aux deux dernières sources d'incertitudes, elles sont spécifiques à la gestion des risques associés aux incertitudes des estimations d'un modèle.

4. Gestion des incertitudes relatives aux imprécisions

Ce type d'incertitude est dû aux imprécisions des estimations générées par Fuzzy Analogy. La précision des estimations d'un modèle est souvent évaluée par l'indicateur MMRE (Moyenne des Erreurs Relatives). L'évaluation de la précision du Fuzzy Analogy sur la base de projets COCOMO'81 a donné une MMRE égale à 21% [6]. Par conséquent, pour chaque estimation du Fuzzy Analogy, nous avons une incertitude de ±21%. Kitchenham et Linkman critiquent l'utilisation du MMRE pour la gestion des incertitudes par le fait qu'elle accorde les mêmes degrés d'incertitude aux sur-estimations aussi bien qu'aux sous-estimations. Ils proposent l'utilisation d'une distribution de probabilité telle que la distribution Gamma pour la gestion de ces incertitudes [8]. Nous nous sommes basés sur leur approche pour la gestion des incertitudes au niveau des estimations du *Fuzzy Analogy*. Cependant, la distribution de probabilité que nous adoptons dépendra des caractéristiques de l'environnement étudié. Cette distribution sera déterminée en utilisant la théorie de possibilité de Zadeh [13].

L'affirmation *similar software projects have similar costs*. Cette affirmation comprend deux sources d'incertitude: premièrement, sa conséquence est imprécise; deuxièmement, elle peut être non-déterministe. Dubois et al. ont proposé deux approches différentes pour la reformulation de cette affirmation selon qu'elle est déterministe ou non-déterministe [4]. Ainsi, la question principale que nous devons examiner est: l'affirmation *similar software projects have similar costs* est-elle déterministe ou non-déterministe en estimation des coûts de développement de logiciels?

Intuitivement, il n'y a aucune raison pour que deux projets similaires n'aient pas des coûts similaires. Cependant, dans la pratique, nous pouvons remarquer dans certains cas l'opposé de cette intuition. Au niveau empirique, nous avons mené une expérimentation sur la base de projets logiciels du COCOMO'81 pour évaluer la nature de l'affirmation *similar software projects have similar costs* (déterministe ou non-déterministe). Ainsi, nous évaluons la similarité entre chaque couple de projets (P_i, P_j) du COCOMO'81 et nous comparons $d(P_i, P_j)$ avec $C(c_i, c_j)$, c_i et c_j sont respectivement les coûts réels de P_i et P_j :

- Si $d(P_i, P_j) \leq C(c_i, c_j)$, l'affirmation est déterministe pour le couple (P_i, P_j). Cela signifie que la valeur de vérité de la proposition P_i *et* P_j *sont similaires* doit être inférieure ou égale à celle de la proposition c_i *et* c_j *sont similaires*.

- Si $d(P_i, P_j) > C(c_i, c_j)$, l'affirmation est non-déterministe pour le couple (P_i, P_j).

Pour le coût de développement d'un projet logiciel, nous présentons ci-dessous deux mesures de similarité entre les coûts de logiciels (Fig. 1):

- La mesure C_R qui définit l'ensemble des valeurs similaires au coût C_0 par l'intervalle $[C_0\text{-}pC_0, C_0+pC_0]$, p est un pourcentage à choisir.

- La mesure C_A qui définit l'ensemble des valeurs similaires au coût C_0 par l'intervalle $[C_0\text{-cst}, C_0+\text{cst}]$, cst est constante à choisir.

Le tableau 1 résume les résultats de l'évaluation de l'affirmation *similar software projects have similar costs* sur la base de projets du COCOMO'81. Dans cette expérimentation, nous avons utilisé plusieurs quantificateurs linguistiques pour l'évaluation de la similarité entre les projets logiciels (colonne α-RIM). Pour l'évaluation de la similarité entre leurs coûts, nous avons utilisé les deux mesures C_A et C_R. Pour tous les projets du COCOMO'81, nous calculons les quantités suivantes:

- NB_CBR_ND: le nombre de cas des couples (P_i,P_j) où l'affirmation *similar software projects have similar costs* est non-déterministe ($d(P_i, P_j) > C(c_i, c_j)$). Dans ces cas, E_{ij} dénote la différence entre $d(P_i, P_j)$ et $C(c_i, c_j)$.

- Le minimum et le maximum des E_{ij}.

L'analyse des résultats de cette évaluation montre que le nombre NB_CBR_ND dépend du quantificateur linguistique utilisé dans l'évaluation de la similarité entre les projets. Autrement dit, NB_CBR_ND dépend de α. Dans les deux cas des mesures C_A et C_R, NB_CBR_ND est monotone croissant en fonction de α. Ceci est dû au fait que nos mesures de similarité sont monotones décroissantes en fonction de α ($d_\alpha(P_i, P_j) \geq d_{\alpha'}(P_i, P_j)$ $\alpha \leq \alpha'$). Si nous considérons le cas du quantificateur linguistique *all* (la ligne *min* du tableau 1) pour évaluer la similarité globale entre les projets et la mesure C_A pour évaluer celle entre leurs coûts, nous constatons que dans 77 cas (seulement 1,94%) l'affirmation est non-déterministe. Par conséquent, nous

avons opté, dans un premier temps, pour l'utilisation de la version déterministe de l'affirmation *similar project have similar costs* dans Fuzzy Analogy.

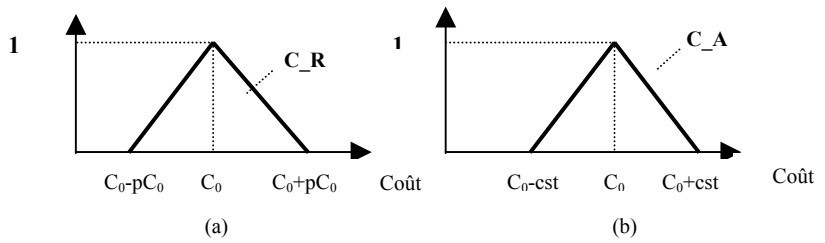

Figure 1. Deux exemples de mesures de similarité entre les coûts de logiciels: (a) C_R, (b) C_A.

	C_A: cts=6			C_R: p=25		
α-RIM	NB_CBR_ND	Min E_{ij}	Max E_{ij}	NB_CBR_ND	Min E_{ij}	Max E_{ij}
Max	3899	0,1666	1	3899	0,0446	1
1/10	3899	0,0839	0,9956	3897	0,0083	0,9924
1	3857	0,0041	0,9606	3799	0,0085	0,9313
100	3765	9,4E-77	0,7276	3623	9,4E-77	0,5769
1000	3487	5,3E-310	0,7276	3355	5,3E-310	0,5769
4000	977	2,6E-309	0,7276	930	2,6E-309	0,5769
10000	239	3,7E-310	0,7276	166	3,7E-310	0,5769
20000	77	0,0121	0,7276	72	0,0121	0,5769
Min	77	0,0121	0,7276	72	0,0121	0,5769

Tableau 1. Résultats de l'évaluation de l'affirmation *Similar software projects have similar costs* sur la base des projets du COCOMO'81.

La formulation d'une estimation au coût d'un projet P dans la version déterministe consiste tout d'abord à chercher tous les projets historiques P_i qui sont étroitement similaires à P, ensuite, déterminer pour chaque P_i, l'ensemble des valeurs similaires au coût de P_i avec un degré de similarité supérieur ou égal à $d(P,P_i)$, $E_i(P)$:

$$E_i(P) = \{c / C(c, c_i) \geq d(P, P_i)\} \qquad \text{(Équation 2)}$$

où c_i est le coût de P_i. Les valeurs possibles au coût du P sont donc données par l'ensemble E(P):

$$E(P) = \cap E_i(P) \qquad \text{(Équation 3)}$$

En appliquant cette procédure de la version déterministe au cas de la base COCOMO'81, nous avons constaté que, dans la plupart des cas, l'ensemble E(P) était vide. En effet, un projet P peut être similaire à plusieurs projets P_i mais les coûts des P_i sont totalement différents. Pour remédier à cette situation, Dubois et al. suggèrent que les mesures de similarités, utilisées dans la version déterministe, doivent satisfaire deux propriétés relatives à la cohérence des deux parties (prémisse et conséquence) des règles floues suivantes [4]: Ces deux propriétés garantissent que l'ensemble E(P), associé à un projet P, ne peut être vide. Nous ne sommes pas convaincu par leur suggestion puisqu'elle peut mener à des mesures de similarité (entre projets ou entre leurs coûts) qui contredisent notre intuition concernant l'attribut *similarité*. Ainsi, nous avons opté pour la version non-déterministe afin de gérer les incertitudes au niveau des estimations fournies par Fuzzy Analogy.

5. Formulation de la version non-deterministe du raisonnement par analogie en estimation des coûts

La version non-déterministe de l'affirmation *similar software projects have similar costs* peut être reformulée comme suit: *similar oftware projects have **possibly** similar costs*, c'est-à-dire, si un projet P est similaire à P_i, il est possible que le coût de P soit similaire à celui de P_i. Dubois et al. proposent de représenter cette affirmation de la version non-déterministe par la règle de possibilité suivante [5]: *Autant P est similaire à P_i, autant il est possible que le coût de P soit similaire à celui de P_i*

Cette règle exprime qu'il est possible, avec un degré au moins égal à $d(P,P_i)$, que le coût de P soit similaire à celui de P_i. En plus, toute autre valeur numérique similaire au coût de P_i est aussi possible, avec au moins le même degré, qu'elle soit similaire au coût du projet P. La modélisation de la règle de possibilité ci-dessus, en utilisant la conjonction pour l'implication, donne l'ensemble flou de toutes les valeurs possibles au coût du P:

$$\pi_{cost(P_i)}(c) = min(d(P,P_i), C\_R_{cost(P_i)}(c)) \qquad \text{(Équation 4)}$$

Dans le cas où la base de données contient N projets logiciels, chaque projet P_i génère un ensemble de valeurs possibles au coût du P selon l'équation 4. Ces ensembles flous sont combinés par l'opérateur *max* (opération de disjonction) afin d'obtenir l'ensemble flou C_p contenant toutes les valeurs possibles du coût du projet P:

$$C_P(c) = \max_i(\pi_{cost(P_i)}(c)) \qquad \text{(Équation 5)}$$

La fonction d'appartenance à l'ensemble C_p représente la distribution de possibilités du coût de P. Il sera utilisée par Fuzzy Analogy, pour évaluer le degré d'incertitude associé à une estimation du coût du projet P. Par exemple, la figure 2 montre la distribution de possibilités associée au coût du projet P_{45} de la base COCOMO'81. P_{45} est similaire aux projets P_{42}, P_{43}, P_{44}, et P_{46} avec des degrés respectivement égaux à 0,38, 0,40, 0,38 et 0,36. La distribution de possibilités est définie seulement pour les valeurs réelles similaires à au moins une des quatre valeurs représentant les coûts de P_{42}, P_{43}, P_{44}, et P_{46}; pour les autres valeurs, la distribution de possibilité est inconnue.

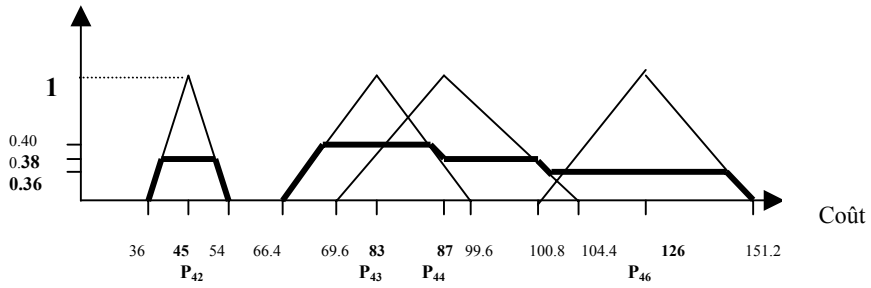

Figure 2. Exemple d'une distribution de possibilités associée au coût du projet P_{45} du COCOMO'81.

7. Discussion et conclusion

Dans cet article, nous avons présenté une nouvelle stratégie pour la gestion des incertitudes au niveau des estimations fournies par notre approche Fuzzy Analogy. Cette stratégie est basée essentiellement sur le fait que l'affirmation *similar software projects have similar costs* est non-déterministe en estimation des coûts. Ainsi, nous avons utilisé la théorie des possibilités pour la modélisation de cette affirmation. Fuzzy Analogy fournit donc un ensemble de valeurs estimées au coût d'un projet avec une fonction de distribution des possibilités indiquant pour chaque valeur son degré de possibilité pour qu'elle soit la valeur réelle du coût. Cette fonction de distribution peut être aussi utilisée pour la gestion des risques attachées aux incertitudes des estimations. La validation de la stratégie de gestion des incertitudes est faite sur la base de projets COCOMO'81; plusieurs expérimentations de cette stratégie sur d'autres bases de projets logiciels sont actuellement en cours.

8. Bibliographie

[1] Aamodt, Agnar, et Enric Plaza. 1994. «Case-Based Reasoning: Foundational Issues, Methodological Variations. and System Approaches». *AI Communications*, IOS Press, vol. 7:1, p. 39-59.

[2] Angelis, L., et I. Stamelos. 2000. «A Simulation Tool for Efficient Analogy Based Cost Estimation». *Empirical Software Engineering*, vol. 5, no. 1, p. 35-68.

[3] Boehm, Barry W. 1981. *Software Engineering Economics*, Prentice-Hall.

[4] Dubois, Didier, F. Esteva, P. Garcia, L. Godo, R. L. deMantaras et H. Prade. 1999. «Case-based Reasoning: A Fuzzy Approach». *Workshop on Fuzzy Logic in Artificial Intelligence fonctions . Lecture Notes in Atificial Intelligence*, Vol. 1566, Springer, Berlin, p. 79-90.

[5] Gentner, G. 1983. Structure Mapping: «A Theorotical Framework of Analogy». *Cognitive Science*, Vol. 7, p. 155-170.

[6] Idri, Ali, Alain Abran et Taghi Khoshgoftaar. 2002a. «Estimating Software Project Effort by Analogy based on Linguistic Values». *8th International Symposium on Software Metrics*, IEEE computer Society, June, Ottawa, p.21-30.

[7] Idri, Ali, Taghi Khoshgoftaar et Alain Abran. 2002b. «Investigating Soft Computing in Case-Based Reasoning for Software Cost Estimation». *International Journal of Engineering Intelligent Systems*, Vol. 10, No. 3, September. p. 147-157.

[8] Kitchenham, Barbara, et S. Linkman. 1997. «Estimates, Uncertainty and Risks». *IEEE Software*, 14(3), p. 69-74.

[9] Kolodner, J. L. 1993. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[10] Schank, Roger. 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computer and People*. Cambridge University Press, 1982

[11] Shepperd, Martin, et C. Schofield. 1997. «Estimating Software Project Effort Using Analogies». *Transactions on Software Engineering*, vol. 23, no. 12, November, p. 736-743.

[12] Vicinanza, S. et M.J. Prietulla. 1990. «Case-Based Reasoning in Software Effort Estimation». *Proceedings of the 11th International Conference on Information Systems*.

[13] Zadeh L.. 1979. «Fuzzy Sets as a basis for a theory of possibility». *Fuzzy Sets and Systems,* Vol 1, p. 3-28.