**Master « Systèmes d'Information et**
**Analyse Décisionnelle»** **Année 2008/2009**

# Projet d'étude : réalisation d'un outil d'évaluation de la qualité

## 1. Objectif général

L'objectif de ce projet d'étude est de vous permettre, au travers d'un problème concret, de **faire une synthèse de vos connaissances acquises** en génie logiciel, programmation orientée objet et bases de données. Il a également pour objectif de vous **faire découvrir de nouveaux concepts** encore non étudiés en cours.

## 2. Sujet

### *2.1. Descriptif général*

L'objectif de ce projet est de **développer une application permettant de mesurer la qualité des données et de l'indiquer à l'utilisateur lors de la réponse à ses requêtes**.

Plus précisément, cette application devra permettre de :

- Évaluer la qualité des données stockées dans une base de données. L'évaluation consistera en l'exécution d'un ensemble d'algorithmes permettant de mesurer ou d'estimer différentes dimensions ou facettes de la qualité (par exemple l'exactitude des données ou l'existence de valeurs nulles).
- À chaque nouvelle requête utilisateur sur la base de données, les valeurs de qualité sont affichées avec les résultats de la requête afin de permettre à l'utilisateur d'en apprécier la fiabilité.

Dans le cadre du projet, le travail commencera par la lecture de l'article *Data Quality at a Glance*, afin de se familiariser avec les dimensions de qualité et les stratégies pour leur évaluation, dont les intuitions sont présentées ci-dessous. Cet article sera présenté en détail lors du bilan de la première phase. Une veille bibliographique sur les algorithmes d'évaluation sera également présentée lors du bilan de la première phase.

### *2.2. Qualité des données*

Généralement, la qualité des données est exprimée ou caractérisée par un ensemble de *dimensions de qualité* qui décrivent les différentes facettes des données fournies aux utilisateurs (par exemple fraîcheur, précision, complétude) ou les processus qui produisent ces données (par exemple temps de réponse, fiabilité, sécurité). Pour un système donné, les dimensions de qualité à évaluer dépendent des utilisateurs et des domaines d'applications. Par

exemple, certains utilisateurs peuvent s'intéresser au temps de réponse et d'autres à la fraîcheur des données.

Dans le cadre du projet, on se concentrera sur cinq dimensions de qualité :

- Exactitude : Concerne la correction et la précision avec laquelle les objets du monde réel sont représentés dans le système d'information.
- Complétude : Concerne le fait de que tous les objets pertinents du monde réel soient représentés dans le système d'information.
- Fraîcheur : Concerne l'âge et l'actualité des données d'un système d'information.
- Cohérence : Concerne la satisfaction des règles d'intégrité d'un système d'information.
- Unicité : Concerne le fait que les données d'un système d'information ne soient pas dupliquées.

Chaque dimension de qualité peut être raffinée dans un ensemble de *facteurs de qualité* qui représentent des aspects particuliers de la qualité. Par exemple, l'exactitude concerne le référencement des objets corrects du monde réel (correction sémantique), la représentation sans erreurs de typage ou format (correction syntaxique) et le niveau de détail (précision).

Chaque facteur de qualité est évalué selon plusieurs métriques, chacune plus ou moins adaptée à un type spécifique de données ou de domaine d'application. Par exemple, la correction syntaxique peut se mesurer comme un booléen (correcte ou incorrecte) ou comme un degré (valeur entre 0 et 1 exprimant le niveau de correction).

Chaque métrique peut être calculée en utilisant plusieurs algorithmes d'évaluation, chacun mieux adapté à un type de données (pour exemple, différents règles de format sont utilisées selon que l'on contrôle une adresse ou un numéro de téléphone).

La Figure 1 illustre la relation entre les concepts de qualité.

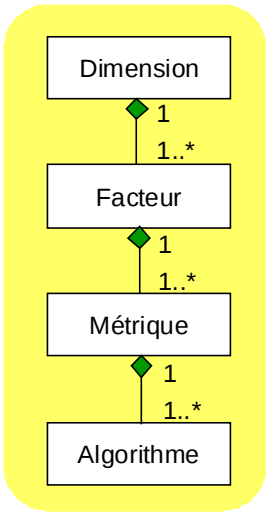


**Figure 1 - Modèle de qualité.**

Les facteurs et métriques à considérer dans le cadre du projet sont décrits en annexe.

### *2.3. Algorithmes d'évaluation*

Les algorithmes d'évaluation sont des routines qui permettent de calculer un facteur de qualité selon une métrique. Ils prennent en entrée une table de la base de données, une collection d'attributs de la table et produisent en sortie un ensemble de mesures de qualité. Certaines méthodes peuvent avoir des paramètres supplémentaires comme des règles de syntaxe à vérifier, des attributs contenant des marques de temps, etc. En général, les méthodes calculent la qualité de chaque donnée de la table (chaque attribut de chaque tuple), mais certaines méthodes calculent la qualité des tuples complets.

Dans le cadre de ce projet, les étudiants devront choisir la manière d'implémenter les algorithmes d'évaluation pour chaque métrique proposée. Il est possible de définir plusieurs algorithmes pour calculer une même métrique. Les bases de données interrogées étant potentiellement volumineuses, une attention particulière devra être portée à la performance des algorithmes. Il sera nécessaire de stocker les mesures de qualité obtenues. Les étudiants devront définir des structures de stockage et des mécanismes de récupération des mesures.

### *2.4. Affichage de la qualité des données*

Lorsqu'un utilisateur pose une requête à la base de données, les résultats devront être affichés avec des indications de qualité. L'utilisateur pourra choisir l'affichage détaillé des valeurs de chaque métrique ou l'affichage d'une valeur globale de qualité calculée en combinant les valeurs des métriques. Il pourra également choisir l'affichage des valeurs de qualité de manière soit détaillée, soit agrégée, et ce pour chaque type d'objet de la base de donnée (valeur d'attribut, tuple, attribut, relation). Il y aura donc 8 modes de visualisation possibles : (1) détaillé – valeur d'attribut, (2) global – valeur d'attribut, (3) détaillé – tuple, (4) global – tuple, (5) détaillé – attribut, (6) global – attribut, (7) détaillé – ensemble des résultats, (8) global – ensemble des résultats.

Pour obtenir une valeur globale de qualité, on combine des valeurs de plusieurs métriques via une somme pondérée. Par exemple, si pour une donnée on a les mesures : <métrique_exactitude=0.8, métrique_fraîcheur=0.9, métrique_complétude=0.6> et on considère comme poids des métriques 0.5, 0.3 et 0.2 respectivement, la valeur globale de qualité est : 0.8 * 0.5 + 0.9 * 0.3 + 0.6 * 0.2 = 0.79. Les poids par défaut sont tous identiques mais ils peuvent être modifiés par l'utilisateur. Noter que pour obtenir des valeurs de qualité comprises entre 0 et 1, la somme des poids doit être égale à 1.

Pour obtenir une valeur de qualité pour un tuple, on combine également des valeurs de qualité des attributs du tuple via une somme pondérée. Les poids par défaut des attributs sont tous identiques mais ils peuvent être modifiés par l'utilisateur. Pour obtenir une valeur de qualité pour un attribut, on calcule la moyenne des valeurs de qualité de l'attribut dans tous les tuples. Pour obtenir une valeur de qualité pour l'ensemble des résultats, on combine les valeurs de qualité des tuples via une moyenne.

La façon d'afficher les valeurs de qualité peut être très variée. On peut afficher des vecteurs de qualité pour chaque donnée (par exemple <métrique_exactitude=0.8, métrique_fraîcheur=0.9, métrique_complétude=0.6>), on peut afficher des colonnes supplémentaires avec des valeurs de la qualité, on peut colorer des données selon leur qualité, etc. Les étudiants sont libres de proposer des façons d'affichage qui leur semblent intuitives, expressives ou bien adaptée à chaque mode de visualisation.

Après avoir visualisé la qualité des données, l'utilisateur peut raffiner les résultats de deux façons différentes : (1) en ordonnant les tuples par rapport à la qualité (valeur d'une métrique ou valeur globale ; valeur pour un attribut ou pour le tuple) ; (2) en filtrant les tuples qui n'ont pas une qualité suffisante, en permettant de spécifier des niveaux de qualité requis (pour chaque métrique ou pour la valeur globale).

### *2.5. Base de données de test*

Afin de tester l'application réalisée, il faut construire une base de données de tests. La base contiendra des données décrivant des films, des personnes participant dans des films (acteurs, directeurs, etc.), des cinémas qui les projettent et des opinions des spectateurs. Les

donnés doivent être téléchargées de différents sites web proposant la programmation ciné. Il est important de collecter des données de plusieurs sites, même en introduisant des doublons ou des données erronées afin de préparer la phase de tests. Le schéma de la base de données à construire sera fourni.

Plusieurs référentiels et dictionnaires seront disponibles afin de servir comme point de comparaison pour évaluer la correction de certaines données de la base de données de tests.

### *2.6. Configuration et extensibilité*

Les algorithmes d'évaluation utilisés pour mesurer chaque métrique ne sont pas connus des utilisateurs. On suppose une phase de configuration dans laquelle l'administrateur du système choisit quel algorithme sera utilisé pour calculer chaque métrique pour chaque attribut ou type de données. Cette configuration pourra être soit statique (par exemple, en utilisant un fichier de configuration) soit dynamique (pouvant être modifié lors de l'exécution de l'application). Les étudiants pourront choisir la manière de configurer leurs applications.

En revanche, l'utilisateur pourra sélectionner les facteurs et métriques qu'il désire visualiser, parmi ceux qui ont été configurés dans l'application. Une interface de sélection lui sera proposée.

Il serait souhaitable de pouvoir ajouter des dimensions, facteurs, métriques et algorithmes d'évaluation à l'application. Pour garantir cela, il faudra définir et documenter des mécanismes pour que des futurs programmeurs puissent réaliser facilement cette extension.

### *2.7. Résumé des fonctionnalités et interface avec l'utilisateur*

L'application doit donc permettre à l'utilisateur:

- de se connecter à une base de données hébergée par un SGBD,
- de sélectionner/désélectionner les facteurs de qualité à visualiser et les métriques correspondantes,
- de sélectionner le mode de visualisation et des poids de combinaison (selon le mode choisi),
- de poser des requêtes sur la base de données,
- de visualiser les résultats des requêtes en incluant de façon intuitive les niveaux de qualité des données (l'application doit fournir à l'utilisateur une explication sur la visualisation de la qualité)
- d'ordonner les résultats des requêtes par rapport à leur qualité,
- de filtrer les tuples du résultat qui n'ont pas une qualité suffisante.

Pour cela, l'application doit donc permettre:

- de configurer les algorithmes d'évaluation à utiliser
- de mesurer plusieurs facteurs de qualité en utilisant plusieurs métriques
- d'évaluer les requêtes sur la base de données et d'obtenir les résultats,
- de calculer des valeurs globales de qualité en fonction des valeurs plus détaillées.

L'interface utilisateur et les schémas de bases de données seront en anglais.

# 3. Organisation du projet

## *3.1. Groupe et encadrement*

Le projet devra être réalisé par groupe de 4 ou 5 étudiants, avec au plus 6 groupes de 5 étudiants. Par ailleurs, un chef de projet devra être désigné pour chaque groupe. Ce chef de projet aura notamment pour rôle de veiller au bon avancement du projet et de servir d'intermédiaire entre son groupe et les enseignants encadrant le projet.

Ce projet sera encadré par trois enseignant-chercheurs intervenant en M1 : Arnaud Giacometti, Patrick Marcel et Verónika Peralta.

Des réunions seront organisées régulièrement entre les encadrants et les étudiants pour faire le point sur l'avancement, des créneaux spécifiques étant réservés à cet effet dans le planning.

## *3.2. Outils informatiques*

La réalisation de ce projet implique l'utilisation d'outils de développement et de système de gestion de bases de données. Les outils à utiliser ne sont pas imposés a priori, et devront être choisis par les étudiants, la seule contrainte étant que tous les outils utilisés soient disponibles gratuitement.

## *3.3. Calendrier*

La réalisation du projet d'étude devra se faire en plusieurs phases, le calendrier étant précisé entre parenthèse :

1. **une phase d'étude bibliographique** sur l'évaluation de la qualité, à l'issue de laquelle une présentation sera réalisée (dans la semaine du 5 janvier). Lors de cette phase, les étudiants devront montrer qu'ils ont compris le sujet et étudié la bibliographie sur les facteurs et métriques de qualité et les algorithmes d'évaluation,
2. **une phase de spécification**, à l'issue de laquelle un cahier des charges général devra être rédigé, présenté et validé et les modalités des tests devront être définies (dans la semaine du 26 janvier). Cette phase correspond à la phase de **préparation** étudiée dans le cadre du cours de gestion de projet. Le cahier des charges inclura donc la capture des besoins et le cadrage, un planning prévisionnel sous la forme d'un diagramme de Gantt, une répartition précise des tâches, et une analyse de risque succincte. Il est également important à ce stade de réfléchir aux indicateurs utilisé pendant le *reporting*, afin de pouvoir mettre en place dès le début de la réalisation le processus de mesure adequat. Pour cela vous pourrez vous appuyer sur le TD3 de gestion de projet (préparation avec 3P) ainsi que sur le TD4 (suivi avec OPPM).

   Durant les phases suivantes, chaque groupe présentera un reporting incluant des indicateurs sur l'avancement. Le chef de projet sera notamment chargé de collecter et de mettre à jour le reporting (technique OPPM). En fonction de ces indicateurs, un réajustement du projet sera peut-être nécessaire. Le chef de projet fera alors les

propositions qui semblent les mieux adaptées. En particulier, la planification sera remise à jour.

3. **une phase d'analyse détaillée**, à l'issue de laquelle un cahier des charges détaillé, avec une modélisation UML du système à développer, devra être présentée et validée (dans la semaine du 9 mars),
4. **une phase de développement**, précédée d'une phase d'analyse et de choix des outils utilisés pour réaliser le système spécifié (présentée dans la semaine du 20 avril),
5. **une phase de test** (présentée dans la semaine du 4 mai)
6. **une phase de présentation,** incluant la remise du rapport final (au plus tard le 22 mai) et une soutenance (se déroulant la semaine du 25 mai). En amont de cette dernière phase, une analyse du déroulement du projet, incluant des recommandations, devra être conduite et présentée dans le rapport sous la forme d'un bilan de projet.

## 4. Évaluation

Chaque phase de déroulement du projet d'étude sera évaluée. Une note différente pourra être attribuée aux différents membres d'un groupe, en fonction de leur rôle (chef de projet ou non) et degré d'implication dans le projet. Les modalités d'évaluation sont présentées dans le tableau ci-dessous :

| | | pourcentage de la note partielle | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| phase | % de la note globale | compréhension | cahier des charges | diagramme de Gantt | analyse | présentation orale | réalisation | démonstration | rapport |
| étude bibliographique | 5 | 60 | | 10 | | 30 | | | |
| spécification | 20 | | 20 | 10 | 50 | 10 | | | 10 |
| analyse | 20 | | 20 | 10 | 50 | 10 | | | 10 |
| développement | 10 | | | 10 | | 10 | 50 | 30 | |
| test | 5 | | | | | 10 | 60 | 30 | |
| présentation | 40 | | | | | 10 | 60 | 10 | 20 |

Quelques précisions :

- **les présentations et rapports à l'issue de chaque phase** devront être en anglais,
- **le rapport final** devra comporter une vingtaine de pages, et devra résumer le problème posé, l'organisation du groupe, les choix effectués, les problèmes rencontrés et solutions proposées, de même que les perspectives possibles. Il devra également présenter une analyse du déroulement débouchant sur des recommandations. Il devra enfin comporter en Annexe un manuel d'utilisation, d'installation et d'extension de l'outil.
- **la présentation orale finale** se déroulera sur 40 minutes, avec 20 minutes de présentation, 10 minutes de démonstration (de l'application développée) et 10 minutes de questions.

- **la réalisation de l'outil** sera évaluée tout au long des phases de développement, test et présentation. Les sources de l'application développée devront être fournis.

## Références


Bibliographie générale (à lire obligatoirement) :

☯□❁Scannapieco, M.; Missier, P.; Batini, C.: Data Quality at a Glance. Datenbank-Spektrum, vol. 14, 2005. http://www.dis.uniroma1.it/~monscan/ResearchActivity/Articoli/DS2005.pdf

☯□❁Wikipedia: Levenshtein distance. http://en.wikipedia.org/wiki/Levenshtein_distance

Bibliographie complémentaire (optionnel) :

Pour approfondir sur la cohérence :

☯□❁Rahm, E.; Do, H.H.: Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, Vol. 23(4): 3-13, 2000. http://homepages.inf.ed.ac.uk/wenfei/tdd/reading/cleaning.pdf (section 2)

Pour approfondir sur la fraîcheur ou l'exactitude :

☯□❁Peralta, V.: Data quality evaluation in data integration systems. PhD Thesis, Université de Versailles (France) and Universidad de la República (Uruguay), 2006. http://www.prism.uvsq.fr/~vepe/pubs/2006/phd-vp.zip (chapitre 2)

Pour approfondir sur la complétude :

☯□❁Naumann, F.; Freytag, J.C.; Leser, U.: Completeness of Information Sources. Proc. of the Workshop on Data Quality in Cooperative Information Systems (DQCIS'03), Siena, Italy, 2003. http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_Naumann/publications/DQCIS03a.pdf (section 4)

Pour approfondir sur l'unicité :

☯□❁Batini, C.; Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Springer-Verlag, ISBN-10 3-540-33172-7, 2006 (chapitre 5)

Pour approfondir sur le matching approximatif :

☯□❁Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys, 33(1):31–88, 2001. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.3112


## Annexe

Cette annexe décrit quelques facteurs et métriques de qualité considérés dans le projet. Les étudiants devront implémenter des algorithmes d'évaluation pour au moins 6 métriques : une de chaque dimension plus une métrique de difficulté élevée (soit correction sémantique booléenne, soit déviation syntaxique, soit dégré d'unicité).

Exactitude (accuracy) :

– Correction sémantique : Indique si les données correspondent aux objets du monde réel. Par exemple si l'adresse stockée pour un cinéma est effectivement l'adresse où se trouve le cinéma. On utilisera comme métrique :

- Correction sémantique booléenne : C'est une valeur booléenne qui indique si la donnée est sémantiquement correcte ou pas. Pour la mesurer, il faut comparer la donnée à un référentiel considéré correct. Noter que les données peuvent être écrites différemment (par exemple « Av. de Paris » et « Avenue de Paris ») ; il faut faire une comparaison approximative. Il existe plusieurs fonctions de matching approximatif (Wikipedia liste quelques unes dans : http://en.wikipedia.org/wiki/Approximate_string_matching), la plus connue étant la distance de Levenshtein.

– Correction syntaxique : Indique si les données sont libres d'erreurs syntaxiques ou de format. Par exemple, si l'adresse d'un cinéma est une adresse valide (la rue existe, le CP est valide, etc.) On utilisera deux métriques :

  - Correction syntaxique booléenne : C'est une valeur booléenne qui indique si la donnée est syntaxiquement correcte ou pas. Pour la mesurer, on peut comparer la donnée à un dictionnaire ou vérifier la satisfaction de règles de format.
  - Déviation syntaxique : C'est une valeur compris entre 0 et 1 qui indique la déviation de la donnée par rapport à une valeur correcte. Elle permet de capturer les différents dégrés de correction (par exemple Pariz est plus proche de Paris que Pariggi). Pour la mesurer, on peut comparer la donnée à la valeur la plus proche dans un dictionnaire, en utilisant des fonctions de matching approximatif.

– Précision : Indique si le niveau de détail des données est suffisant. Par exemple, si les nombres ont assez de décimales ou si on dispose de tous les attributs d'une adresse (numéro de porte, rue, ville, CP). On utilisera comme métrique :

  - Dégrée de précision : Une valeur comprise entre 0 et 1 qu'indique le niveau de détail de la donnée. Par exemple 1 pour 2 chiffres décimaux, 0.5 pour des valeurs sans décimaux, plus petit pour des arrondis. Autre exemple : 1 pour les adresse complètes, 0.5 quand il manque le numéro de porte, etc. Il faut définir les pondérations pour chaque attribut.

Complétude (completeness) :

– Densité : Indique si les données sont nulles ou pas. On utilisera comme métrique :

  - Densité booléenne : C'est une valeur booléenne qui indique si la donnée est non nulle ou pas. Pour la mesurer, il faut tester si la donnée est nulle.

Fraîcheur (freshness) :

– Actualité : Indique le temps passé depuis l'extraction des données. On utilisera comme métrique :

  - Dégrée d'actualité : C'est une valeur compris entre 0 et 1 qui indique l'actualité de la donnée. Elle est calculée en divisant le temps passé depuis l'extraction de la donnée par le temps de validité de la donnée, selon la formule :

    Max { 0, 1- temps_extraction / temps_validité }

Cohérence (consistency) :

– Intégrité de domaine : Indique si les données satisfont des règles de domaine (par exemple, que les numéros de téléphones soient composés de 10 chiffres en commençant par 0). On utilisera comme métrique :

- Intégrité de domaine booléenne : C'est une valeur booléenne qui indique si la donnée satisfait les règles de domaine. Pour la mesurer il suffit de vérifier la satisfaction des règles.

– Intégrité de tuple : Indique si les tuples satisfont des règles inter-attributs (par exemple, que la ville et le code postal soient cohérents entre eux). On utilisera comme métrique :

- Intégrité de tuple booléenne : C'est une valeur booléenne qui indique si le tuple satisfait les règles inter-attributs. Pour la mesurer il suffit de vérifier la satisfaction des règles. Noter qu'on obtient de mesures par tuple et non par donnée unitaire.

– Intégrité référentielle : Indique si les tuples satisfont des règles d'intégrité référentielle (clés étrangères). On utilisera comme métrique :

- Intégrité référentielle booléenne : C'est une valeur booléenne qui indique si le tuple satisfait les règles référentielles. Pour la mesurer il suffit de vérifier la satisfaction des règles. On obtient aussi de mesures par tuple et non par donnée unitaire.

Unicité (uniqueness) :

– Unicité : Indique si les données sont uniques ou dupliquées. On utilisera comme métrique :

- Unicité booléenne : C'est une valeur booléenne qui indique si le tuple est unique ou pas. Pour la mesurer, il faut comparer les tuples, par paire, en regardant l'égalité des clés. On obtient aussi de mesures par tuple et non par donnée unitaire.
- Degré d'unicité : C'est une valeur comprise entre 0 et 1 qui indique le degré d'unicité du tuple. Pour la mesurer, il faut comparer les tuples, par paire, en calculant leur degré de similarité, en utilisant des fonctions de matching approximatif. On obtient aussi de mesures par tuple et non par donnée unitaire.

Noter que pour les métriques qui produisent des mesures par tuple on ne peut pas afficher les valeurs de qualité de chaque donnée. Les étudiants pourront choisir entre répéter la même valeur pour chaque attribut du tuple ou ne pas visualiser les métriques dans les modes « donnée ».