

Visualisation d'informations

TD 2

Dans ce TD, vous allez voir quelles sont les étapes de conception et de validation d'une visualisation. Le modèle que vous allez découvrir tout au long des exercices est basé sur le modèle imbriqué proposé par Tamara Munzner¹.

1. Définition du problème

La première étape que vous devez effectuer pour concevoir une visualisation consiste à **caractériser le problème du public ciblé**, i.e. pour qui et pourquoi votre visualisation doit être faite. Pour cela, vous devez observer les pratiques des potentiels futurs utilisateurs. Cette étape est cruciale et pourtant souvent négligée, entraînant ainsi la production d'outils visuels qui ne seront pas utilisés ! L'objectif principal est de comprendre les données du public ciblé, les besoins des potentiels utilisateurs de la visualisation et surtout les pratiques courantes (ce qu'ils ont, ce qui marche, ce qui leur manque...).

D'autres éléments doivent aussi être relevés lors cette première phase, par exemple :

- Le public ciblé est-il habitué à l'outil informatique ?
- Le public ciblé est-il habitué à manipuler des outils de visualisation évolués ?
- Combien de temps peuvent/veulent consacrer les individus à l'apprentissage de l'utilisation de l'outil de visualisation ?
- Quel sera l'apport (financier ou en temps) de l'application pour ses utilisateurs ?
- Y-a-t 'il des contraintes technologiques (e.g. application Web vs. desktop, application interactive vs. production d'images sans contrainte de temps de calcul, ...) ?
- ...

Toutes ces questions vous permettront de proposer une application adaptée. Il est clair que si un grand journal vous demande de produire une visualisation grand public, des objectifs de simplicité et d'intuitivité seront à privilégier, quitte à ne pas explorer toute la richesse des données. En revanche, les collaborateurs d'une entreprise travaillant dans la finance seront sans doute plus enclins à passer du temps à se former sur votre outil afin de pouvoir effectuer l'analyse la plus fine possible et vous devrez donc privilégier la finesse des visualisations et la richesse des fonctionnalités sans vous soucier de leur « complexité cognitive ».

***Exemple :** Tout au long de ce TD, nous allons prendre l'exemple de Bacchuseo, une entreprise proposant une application pour smartphone permettant d'échanger, entre utilisateurs, des avis sur les vins. L'application ne se limite pas aux œnologues, elle touche un plus large public (environ 300 personnes). Le directeur marketing que vous rencontrez vous explique qu'après avoir longuement parcouru les échanges entre utilisateurs, il lui semble que ces derniers ont tendance à former des petits groupes d'intérêt communs, comme celui des « amateurs de vin Australien » ou des « fidèles des châteaux de Pomerol » (selon ces propres termes). Les membres des petits groupes semblent échanger beaucoup entre eux, mais peu avec les membres des autres groupes. L'application est gratuite pour les utilisateurs, le chiffre d'affaire de l'entreprise correspond aux annonces publicitaires diffusées dans l'application. Attention, elles sont « exclusivement sélectionnées en fonction de leur rapport avec le vin ! D'ailleurs, il n'y en a que très peu afin de ne pas dégoûter les clients. » Comme tous les ans, les organisateurs du salon du vin de Saint-Emilion ont contacté*

¹ Tamara Munzner. A nested process model for visualization design and validation. IEEE Transactions on Visualization and Computer Graphics 15(6):921–928, 2009.

Bacchuseo pour diffuser auprès de leurs utilisateurs l'annonce du salon. Seulement, cette année, les organisateurs veulent aller plus loin en offrant deux entrées gratuites ainsi qu'une bouteille gratuite aux utilisateurs les plus influents de l'application. Ils ont donc demandé à Bacchuseo de sélectionner ces membres, sans donner de chiffre précis. Comme le nombre n'est pas fixé, l'entreprise désirerait investir dans un outil visuel permettant de voir les utilisateurs les plus influents selon les différents groupes, et ainsi pouvoir discuter avec les organisateurs du salon au sujet de ceux à sélectionner. Par exemple, si 3 personnes sont très influentes dans un groupe, il n'est peut-être par nécessaire de leur donner un cadeau aux 3. En revanche, si une personne est peu influente mais qu'elle est la seule à avoir un minimum d'influence dans un groupe, il serait judicieux de lui donner un cadeau.

2. Abstraction des tâches et des données

Cette étape consiste à « traduire » le problème identifié dans la partie 1 en problème pouvant être résolu à l'aide d'une visualisation. Trois activités constituent cette étape. **(1) Vous devez clairement formuler les tâches que devront accomplir les utilisateurs.** Par exemple, ils peuvent vouloir identifier des valeurs aberrantes, trouver des points maximaux, découvrir des cycles dans des données temporelles, voir des chemins dans des graphes, etc. Comme vous pouvez le voir, ces tâches doivent être exprimées de façon abstraite, elles concernent les données en elles-mêmes et non leur sémantique. **(2) Vous devez clairement définir les structures de vos données :** manipulez-vous des tables, des graphes, des champs, des géométries ? quels sont les types des variables ? etc. **(3) Parfois les données brutes ne suffisent pas à remplir les tâches définies en (1) et vous allez devoir calculer de nouvelles données à partir de celles fournies. Ces données sont dites dérivées.** C'est par exemple le cas lorsque vous calculez une médiane, lorsque vous extrayez des motifs fréquents ou lorsque vous partitionnez vos éléments à l'aide d'un algorithme de *clustering*.

Exercice 1 : Reprenez l'exemple de la partie 1 et définissez les tâches que les individus du public ciblé devront pouvoir accomplir avec votre visualisation, les structures de données que vous allez utiliser et les attributs dérivés que vous allez calculer.

3. Description de la visualisation

Cette étape consiste à **définir les différents idiomes visuels et les méthodes d'interaction** que vous souhaitez mettre en place afin de permettre à un utilisateur d'accomplir les tâches sur les données décrites précédemment. Pour cela, vous pouvez réaliser des maquettes de l'application (dessins des différentes vues), faire des schémas montrant comment les composants interagissent, décrire textuellement des fonctionnalités... Il arrive même parfois que vous ayez à réaliser des preuves de concept (plus connues sous l'acronyme anglais *POC, Proof Of Concept*).

Exercice 2 : Reprenez l'exemple précédent et proposez une visualisation.

4. Conception des algorithmes

La dernière étape de la phase de conception consiste à **trouver les algorithmes** permettant de mettre en place la visualisation proposée lors de l'étape ci-dessus. Ces algorithmes peuvent être recherchés dans la littérature ou vous pouvez les créer. Plusieurs critères doivent entrer en compte dans le choix des algorithmes : qualité du résultat en fonction des attendus, temps de calcul en fonction de la taille des données et du besoin d'interactivité, coût de « production » (= temps de programmation nécessaire à les mettre en place), etc.

Exercice 3 : Trois algorithmes sont nécessaires pour réaliser cette visualisation : celui qui calcule le *clustering*, celui qui calcule la centralité et celui qui calcule le positionnement des sommets du

graphe. Vous verrez les deux premiers dans le cours d'analyse de réseaux sociaux en M2. Quel algorithme de positionnement vu dans ce cours allez-vous utiliser pour dessiner votre graphe ?

5. Risques

Les étapes de conception identifiées ci-dessus comportent des risques qui entraînent fréquemment la production de visualisations non adaptées. Par exemple, le principal risque de la première étape est de **mal comprendre le problème du public ciblé**, et donc de produire une visualisation ne leur servant à rien. Le meilleur moyen de s'en prémunir est de **passer du temps à observer et à discuter avec les futurs utilisateurs** pour se familiariser au mieux à leur domaine afin d'en comprendre les enjeux, et connaître les outils qu'ils utilisent actuellement et leurs limites. Cette pratique peut paraître évidente mais elle est pourtant trop souvent ignorée ou bâclée !

Exercice 4 : Pour les étapes de conception 2 à 4, identifiez les risques et proposez, si vous en trouvez, des moyens pour vous en prémunir.

6. Implémentation

Les étapes de conception sont maintenant terminées, vous pouvez passer à l'implémentation de votre visualisation. Vous devez commencer par choisir une technologie, en fonction de différents critères : temps d'implémentation, familiarité avec le langage, performances nécessaires, contraintes imposées par le ou les utilisateurs, etc. Vous devez ensuite tout coder, et tester.

7. Validation

Dans la section 5, vous avez vu comment vous prémunir de certaines erreurs. Il est cependant assez évident que ces précautions sont insuffisantes pour vous assurer que votre visualisation remplira bien les fonctions pour lesquelles elle a été créée. Pour chacune des étapes, il existe des pratiques permettant de les valider. Ces validations ne peuvent être effectuées qu'après l'implémentation du système. Les étapes sont traitées de la dernière, à la première. Pourquoi ? Prenons par exemple les deux dernières étapes (description de la visualisation et conception des algorithmes). Il faut d'abord vérifier que les algorithmes réalisent bien la visualisation que vous avez imaginé avant de vérifier que celle-ci permet bien d'accomplir les tâches pour laquelle elle a été créée. En effet, si vous faites l'inverse, vous risquez d'invalider les idiomes visuels choisis alors que c'est simplement un défaut des algorithmes qui fait que ces idiomes ne paraissent pas efficaces. Le principe est le même pour les autres étapes : il faut avoir validé l'étape $i+1$ pour pouvoir valider l'étape i .

Exercice 5 : Dans l'exercice 3, vous avez identifié un algorithme intéressant pour placer les sommets de votre graphe. Comment allez-vous l'évaluer ?

La validation de la description de la visualisation (étape 3) est plus délicate. Il faut arriver à montrer que celle-ci permet bien d'accomplir les tâches sur les données structurées issues de l'étape 2. Cette étape nécessite le test de la visualisation par des utilisateurs, pour voir leur capacité à réaliser les tâches. Bonne nouvelle, comme celles-ci sont définies de façon abstraite, il n'y a pas besoin de faire appel pour l'instant à des personnes connaissant le jeu de données. Si l'on reprend l'exemple précédent, il va falloir vérifier qu'un utilisateur lambda arrive visuellement à retrouver les clusters et les sommets centraux pour chacun d'entre eux. Pour vérifier cela, vous allez devoir mettre en place une expérimentation.

Une **expérimentation** se compose d'essais. Chaque **essai** consiste à demander à un **participant** d'accomplir une **tâche** sur un **jeu de données**. Chaque participant doit réaliser un certain nombre d'essais mais vous devez faire en sorte que le temps nécessaire n'excède pas 30

minutes, afin qu'il reste concentré. Vous devez réunir un nombre important de participants afin de pouvoir réaliser une « analyse statistique » de vos résultats (30 au minimum). Vous devez aussi réfléchir à vous prémunir de différents biais en vous assurant au maximum que chaque participant se trouvera dans les mêmes conditions : même ordinateur (peut par exemple jouer sur les temps de calculs de la visualisation...), même luminosité de la pièce (peut jouer sur la perception des couleurs...), même résolution (peut jouer sur la qualité du rendu...), etc. Il est évident que vous n'arriverez pas à vous débarrasser de tous les biais possibles, mais essayez au moins de les limiter au maximum.

Généralement, 2 valeurs sont mesurées à chaque essai : la **rapidité** d'exécution de la tâche et la **justesse** de la réponse donnée. Une **évaluation quantitative** est ensuite réalisée à partir de ces chiffres. A la fin des tests, on demande aussi généralement aux participants de donner leur avis de façon à réaliser une **évaluation qualitative**. Attention, les deux aspects peuvent être importants, nous allons voir pourquoi dans l'exemple suivant, tiré de faits réels !

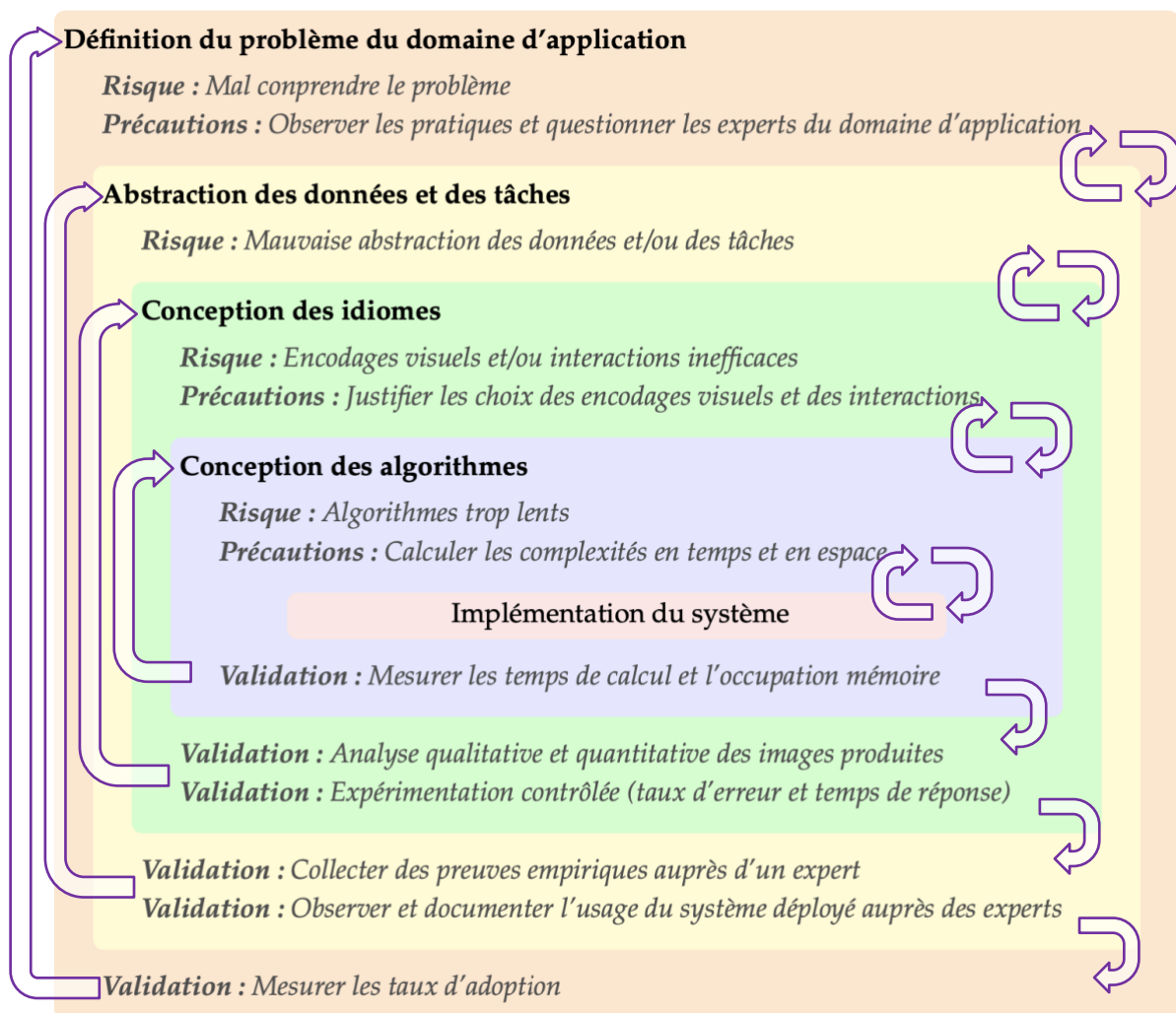
Exercice 6 : J'ai visité une organisation dont l'objectif était de réaliser et vendre des visualisations. L'une d'entre elles avait pour but de montrer un ensemble d'éléments munis d'un attribut quantitatif. Comme nous l'avons vu en cours, un simple diagramme en barres permet parfaitement de visualiser ce type de données. Dans ce cas, la variable visuelle utilisée est la longueur (taille 1D) des barres. Cependant, l'organisation avait opté pour une autre solution : ils représentaient leurs données à l'aide d'un *treemap* sur un niveau, dans lequel chaque rectangle représentait un élément et son aire représentait la valeur de l'attribut quantitatif. Étonné, j'ai demandé pourquoi ils n'avaient pas choisi la simplicité. Ils m'ont répondu que leurs clients préféraient le *treemap* au diagramme en barres (« évaluation qualitative »). Or faites l'expérimentation disponible sur l'espace de cours (« Taille 2D (aires) vs. taille 1D (longueurs) »). Comparez vos résultats avec les autres étudiants. Que nous apprend l'évaluation quantitative ? Est-elle en accord avec l'évaluation qualitative ?

Vous devez maintenant évaluer l'abstraction des tâches et des données, i.e. vérifier que les tâches abstraites que vous avez identifiées, les structures que vous avez définies et les données dérivées que vous avez produites permettent de répondre au problème de l'étape 1. Pour cela, vous allez devoir **rendre accessible votre application à vos commanditaires et observer la façon dont ils l'utilisent**. Vous pouvez leur proposer un questionnaire de satisfaction, en demandant pour chaque fonctionnalité si elle leur est utile (i.e. si elle répond bien à un besoin) et si elle est facile à utiliser. Vous allez aussi leur demander de s'en servir pour voir s'ils arrivent à retrouver des connaissances qu'ils avaient déjà et surtout s'ils arrivent à en extraire de nouvelles.

Pour finir, vous allez devoir vérifier que le problème que vous avez compris lors de l'étape 1 correspond bien à un besoin des utilisateurs. Pour cela, il va falloir **mesurer le taux d'adoption de la visualisation**.

8. Conclusion

Le modèle de Tamara Munzner peut être résumé par le schéma suivant. Il est dit « imbriqué » car les phases de validation sont réalisées dans l'ordre inverse des étapes de conception.



Comme vous pouvez l'observer dans ce schéma, les étapes ne sont pas réalisées de façon linéaire. **(1)** Si la validation d'une étape échoue, vous devez refaire cette étape et potentiellement refaire les suivantes. Par exemple, si une évaluation quantitative montre que l'idiome que vous avez sélectionné n'est pas efficace, vous devez non seulement proposer un nouvel idiome (étape 3) mais aussi proposer les algorithmes permettant de le mettre en place (étape 4). L'invalidation d'une étape i ne remet pas systématiquement en cause tous les éléments des étapes suivantes mais peut cependant y avoir de lourds impacts dont vous devez vous soucier. **(2)** Vous allez souvent être amenés à faire intervenir l'utilisateur sur les différentes étapes de la conception. Au fur et à mesure que vous lui ferez des propositions, il affinera sa demande et vous votre définition du problème, donc votre abstraction, donc... etc.

Pour conclure, ce modèle n'est donc pas figé et vous ne devez pas le suivre au sens strict. Cependant, vous efforcez à distinguer et à séparer les différentes étapes et leur validation vous permettra d'éviter de nombreuses erreurs et perte de temps malheureusement si répandues.