

# Chapitre 3:

## Rédaction

# Chapitre 3: Rédaction

a. Structure

b. Contenu

c. Style

d. Présentation

# Introduction

- Outre la bibliographie, présentée ci-dessus, la rédaction s'appuie sur les éléments suivants : structure, contenu, style et présentation. La structure est détaillée dans la Section 1.
- La Section 2 traite de la manière de proposer un contenu rigoureux et scientifique.
- Enfin, des conseils de style et de présentation sont donnés dans les Sections 3 et 4.

# Introduction(Suite)

- Un travail écrit est organisé en chapitres, sections, sous-sections, etc. Pour simplifier la lecture, nous utiliserons dorénavant le terme section sans préciser s'il s'agit d'un chapitre, d'une section ou d'une sous-section.
- Une section peut en effet être vue comme étant une unité de la présentation, qui possède un titre, quelque soit le niveau de profondeur de cette section.

# Structure

- A ce stade, on devrait avoir un plan de rédaction, même si celui-ci peut encore évoluer.
- Il s'agit maintenant de rédiger les sections identifiées et de comprendre le rôle et l'importance de toutes les parties (introduction, conclusion, etc.) qui forment l'organisation du travail écrit.
- Le plan contient déjà les titres des chapitres et des sections. Ces titres sont des clés importantes pour le lecteur.

# Structure(Suite1)

- En écrivant le contenu d'une section, révissez également son titre. Choisissez-le de manière à ce qu'il soit bref mais suffisamment informatif (on comprend ce qui va suivre).
- Conservez le même schéma de structuration et gardez la même forme grammaticale pour l'ensemble des titres.



1. Pour introduire le problème
2. Les solutions précédentes
3. Ma solution
4. Quelques remarques pour conclure



1. Introduction
2. Etat de l'art
3. Résultats
4. Conclusion

# Structure(Suite2)

## Structure interne

- Les sections identifiées dans le plan de rédaction servent à donner une structure globale au travail écrit. Cependant, il faut également structurer localement le texte à l'intérieur de chaque section.
- Les unités de présentation qui servent à structurer le texte dans une section sont les paragraphes. Ceux-ci doivent se suivre de façon logique.
- Un paragraphe doit également être organisé lui-même de façon logique. Il est constitué de quelques phrases.

# Structure(Suite3)

## Structure interne (Suite)

- Il est constitué de quelques phrases. Ces phrases sont liées et communes à un point précis du sujet qui est l'objet de la section qui le contient.
- Comme pour la structure générale du travail, n'organisez pas vos paragraphes de manière chronologique. Structurer la présentation des résultats et éviter de donner la liste chronologique des essais infructueux et des erreurs commises.
- Le premier paragraphe d'une section devrait introduire le sujet de celle-ci.

# Structure(Suite4)

## Introduction

- L'introduction est en général organisée comme suit: contexte, définition du problème, présentation et limitations des solutions existantes (s'il y en a), objectifs du travail et idées principales.
- Elle se termine par une brève description du contenu, chapitre par chapitre .
- L'introduction est une partie importante du texte. Elle doit convaincre le lecteur que le travail vaut la peine d'être lu.

# Structure(Suite5)

## Introduction (Suite)

- Il faut motiver ce lecteur, qui n'est peut-être pas a priori intéressé par votre travail.
- Expliquez pourquoi le problème étudié est important, quelle sera votre contribution et pourquoi les solutions apportées sont appropriées.
- Gardez à l'esprit que le lecteur n'a pas encore lu le travail, qu'il ne connaît pas le sujet et qu'il n'est pas un expert du domaine.
- Dans votre introduction, parlez du sujet de votre travail, et pas du fait que vous réalisez un travail en informatique.

# Structure(Suite6)

## Chapitres et sections

- La rédaction des chapitres et des sections constitue le cœur du travail. On y présente l'objet et le développement du travail selon le plan constitue au préalable.
- Il faut d envelopper les idées principales et les résultats et convaincre le lecteur de leur importance et de leur validité.
- Le contenu des chapitres doit suivre une approche scientifique.

# Structure(Suite7)

## Chapitres et sections(Suite)

- Quand vous rédigez une section, gardez en tête que le titre de celle-ci qui sert à structurer
- le texte et à introduire le sujet de la section ne sera pas toujours lu par le lecteur ! Le premier paragraphe de la section devrait l'introduire en précisant son sujet car seul le titre ne suffit pas.
- Pour des sections de haut niveau (comme un chapitre), il est utile de commencer par une brève description du contenu, en présentant les sous-sections.

# Structure(Suite8)

## Conclusion

- La conclusion est la dernière partie du travail écrit (la bibliographie et les annexes n' étant pas considérées comme faisant partie du texte lui-même). Elle est en général organisée comme suit: résumé du travail et des contributions, rappel des résultats principaux, applications possibles des résultats (s'il y a lieu), limitations de la solution proposée et perspectives (pistes pour d' éventuels travaux futurs).
- Le texte de la conclusion doit rester neutre mais doit mettre en avant l'apport de l'auteur par rapport au sujet.

# Structure(Suite9)

## Conclusion (Suite)

- La conclusion est une partie importante du texte. Elle résume la contribution personnelle du travail et met en avant les principaux résultats. Elle sert à remettre en place tout ce qui précède.
- Contrairement à l'introduction où le lecteur ne connaît encore rien du sujet, ici on suppose que le lecteur a lu l'ensemble du travail. La conclusion permet au lecteur de confirmer son opinion sur l'étendue du travail réalisé.
- Il est conseillé de rédiger la conclusion en dernier lieu.

# Contenu

- Pour convaincre le lecteur, une approche scientifique doit être suivie (hypothèses, mesures, vérification, preuve, etc.).
- Toute affirmation ou donnée chiffrée doit être justifiée ou, si elle n'est pas personnelle, doit être citée en incluant une référence.
- Le lecteur doit pouvoir clairement faire la distinction entre ce qui est un apport personnel et ce qui ne l'est pas.
- Discutez toujours les choix technologiques et les alternatives possibles. Pourquoi utiliser tel outil, langage, algorithme, formalisme ?

# Contenu(Suite1)

 En évolution logicielle, seulement 30 à 40% des classes réellement modifiées sont identifiées comme susceptibles de l'être.

 Nous avons choisi d'utiliser le langage  $X$  car nous l'avons déjà utilisé dans le cadre du cours  $A$ .

 Sur base d'une étude empirique, Lindvall et Sandahl [8] observent que seulement 30 à 40% des classes réellement modifiées avaient été identifiées comme susceptibles de l'être.

 Une des caractéristiques du langage  $X$  est de pouvoir utiliser l'héritage multiple, ce qui n'est pas le cas des langages  $Y$  et  $Z$ . L'héritage multiple est important dans notre cas, car (...).

# Contenu(Suite2)



Pour ma part, je crois que le TCL et le TK sont des langages fantastiques, permettant énormément de choses sans devoir lire dix livres et qui ne nécessitent pas dix ans de pratique.



Entre les deux algorithmes précédemment décrits, ce fut  $A$  qui fut retenu, et ce, pour diverses raisons. Ce fut tout d'abord pour permettre une manipulation plus simple de la structure de donnée. Enfin, l'algorithme  $A$  est plus « objectif » que le  $B$ . Or, il me semblait nécessaire d'être objectif.



Fantastiques ? Enormément de choses ? Dix livres ? Dix ans de pratique ? Quels sont les critères objectifs en faveur de ces langages ? Soyez neutre et quantifiez avec précision.



Les complexités en temps dans le *pire des cas* des deux algorithmes  $A$  et  $B$  sont identiques ( $O(n^2)$ ). Cependant, les temps d'exécution *en moyenne* sont différents :  $\Theta(n \log n)$  pour l'algorithme  $A$  et  $\Theta(n^2)$  pour  $B$ . On peut donc espérer, sur des données aléatoires, de meilleurs temps d'exécution en choisissant l'algorithme  $A$ . Cette espérance théorique est confirmée par les tests présentés ci-dessous. (...)

# Contenu(Suite3)

## Tests sur ordinateurs

- Dans le cas d'une expérience sur ordinateur, il faut toujours mentionner la configuration de la machine (hardware).
- Il faut également, pour tous les logiciels utilisés lors de l'expérimentation, indiquer d'où ils viennent (site web pour les télécharger) et quelle version a été utilisée.
- Ces éléments permettent au lecteur de reproduire lui-même l'expérience.

# Contenu(Suite4)

## Tests sur ordinateurs (Suite)



Le temps d'exécution du programme sur les données  $X$  est de 12 secondes et sur les données  $Y$  de 23 secondes.



Le Tableau  $x$  reprend les temps d'exécution, exprimés en micro-secondes CPU, sur les différents ensembles de données. Ces tests ont été effectués sur une machine ayant les caractéristiques suivantes : Dell Dual Core, 2.66 GHz, 2 Gb RAM, système SuSE Linux 10.0 (kernel 2.4.2), java 1.5.0, etc. Pour calculer le temps CPU, la classe *ThreadMXBean* a été utilisée.

# Contenu(Suite5)

## Algorithmes

- Pour que le lecteur puisse facilement comprendre un algorithme complexe, seul le pseudo-code ne suffit pas.
- Il faut utiliser une approche qui présente les grandes idées, avant de détailler les choses progressivement.
- Par exemple, pour un algorithme important, on peut :
  1. présenter l'objectif de l'algorithme (entrées, sorties) ;
  2. donner les grandes idées de son fonctionnement en français ;
  3. détailler les idées importantes ;
  4. donner le pseudo-code (si celui-ci est trop long, les parties

# Contenu(Suite6)

## Algorithmes(Suite)

4. donner le pseudo-code (si celui-ci est trop long, les parties importantes seulement) ;
  5. appliquer l'algorithme sur un exemple ;
  6. prouver son exactitude ;
  7. donner et prouver sa complexité en temps et en mémoire ;
  8. éventuellement, donner une implémentation en annexe.
- Pour présenter les algorithmes ou les structures de données, inspirez-vous de livres de références en informatique.

# Style

- Vous n'écrivez pas un roman mais un document scientifique. Cela implique un style adapté.
- Le texte doit être compréhensible par un non-spécialiste du sujet.
- Il faut éviter le style prise de notes, faire des phrases et s'inspirer du style d'un livre scientifique.
- Un style propre à un travail scientifique devrait respecter les consignes suivantes :

**1. Précision** . Il faut définir précisément les notions ( formalisme)

# Style(Suite1)

**2. Concision.** Allez à l'essentiel et faites des phrases courtes. Evitez d'utiliser des mots inutiles. Essayez de ne donner qu'une seule idée par phrase. Une phrase complexe peut être coupée en phrases plus courtes.

**3. Neutralité.** Utilisez un style neutre. N'utilisez pas le « je », sauf dans les remerciements.

**4. Conjugaison.** La voix active est plus directe que la voix passive. Utilisez le présent autant que possible pour un style plus dynamique (sauf dans la conclusion où l'on peut utiliser le passé). Les phrases doivent contenir un verbe.

# Style(Suite2)

**5. Orthographe.** Vérifiez l'orthographe et les fautes grammaticales.

**6. Typographie.** Respectez les règles typographiques propres à la langue utilisée.

**7. Exemples.** Illustrez les concepts importants ou complexes par des exemples simples.

**8. Acronymes.** Évitez d'utiliser trop d'abréviations et acronymes si ce n'est pas nécessaire. Si vous utilisez une abréviation il faut d'abord l'introduire.

# Style(Suite3)

[Précision : formalisme et cohérence]

 Le nombre de nœuds d'un graphe est noté  $n$ . Un graphe est un ensemble de sommets  $S$  et d'arêtes entre nœuds. (...) Un graphe pour lequel  $|S| = 0$  est un graphe vide.



Un graphe *non-dirigé*  $G$  est une paire ordonnée  $(S,A)$ , où  $S$  est un ensemble fini d'éléments appelés *sommets* et  $A$  est un ensemble de paires non-ordonnées de sommets distincts de  $S$ . Chaque élément  $\{s,t\} \in A$  est une *arête* reliant les sommets  $s$  et  $t$ . On note  $n$  le *nombre de sommets* de  $G$ . (...) Si  $n = 0$ , alors le graphe est *vide*.

[Précision : cohérence]

 (...) software evolution (...) software change (...) software modification (...)



Ces notions sont-elles les mêmes ? Si seulement la première a été définie formellement dans le texte, le lecteur peut en douter. Evitez d'utiliser des synonymes pour une même notion.

# Style(Suite4)

## [Concision]

 Pour sauvegarder des données on peut le faire sous forme de fichiers textes, qui représentent les données dans une forme lisible par l'être humain, par une séquence de caractères ; ou par des fichiers binaires dans lesquels les données sont représentées par des octets.



Il y a deux manières différentes de stocker des données : dans un format *texte* ou *binaire*. Dans un format *texte*, les données sont représentées par une séquence de *caractères*. Ces données sont lisibles pour un être humain. Dans un format *binaire*, les données sont représentées par une séquence d'*octets*.

## [Neutralité]

 Je commençais à regretter d'avoir voulu programmer les quatre opérations, mais je m'étais fixé un but et je voulais l'atteindre. De toute façon, cela rend le projet encore plus passionnant !



Quel est l'intérêt de savoir cela ? Evitez ce genre de réflexion personnelle.

# Style(Suite5)

*[Concision, précision, neutralité]*



Par la pratique, on remarqua que des couleurs pouvaient être obtenues en mélangeant d'autres couleurs. On est assez vite arrivé à quelques couleurs de base à partir desquelles on pouvait représenter toutes les couleurs. Ces fameuses couleurs sont appelées couleurs de base : le rouge, le vert et finalement le bleu. A partir de ces couleurs, il est donc possible par un savant mélange de représenter toutes les couleurs.



Toute couleur peut s'obtenir par un mélange de trois couleurs de base : le rouge, le vert et le bleu.

# Style(Suite6)

## [Conjugaison]



- L'algorithme sera exécuté en temps  $O(n)$ .
- Nous avons d'abord essayé la méthode  $X$  mais (...).
- La valeur 0 signifiant le noir et la valeur 1 représentant le blanc.



- La complexité en temps de l'algorithme est  $O(n)$ .
- La méthode  $X$  ne convient pas car (...).
- La valeur 0 représente le noir et la valeur 1 représente le blanc.

## [Acronymes, précision]



L'objet de ce travail est de développer un *CMS*.



L'objet de ce travail est de développer un *Contents Management System (CMS)*. Un *CMS* est un système de gestion de contenu, c'est-à-dire (...).

# Présentation

- Pour bien concevoir la mise en page de votre travail, utilisez un bon logiciel de traitement de texte, comme LATEX.
- Choisissez les polices et la taille de caractères avec soin et évitez d'en changer trop. En général, on utilise un texte justifié pour les paragraphes.
- La hiérarchie des titres doit être cohérente visuellement ( e.g., taille des titres).
- Vous pouvez utiliser des notes de bas de page (mais pas trop) pour donner des précisions qui ne sont pas nécessaires à la compréhension du texte.

# Présentation(Suite1)

- Une note de bas de page ne sera pas toujours lue.
- Pour être sûr qu'une précision soit lue, on peut la donner dans le texte entre parenthèses.
- D'autres éléments, présentés ci-dessous, facilitent la lecture et améliorent la présentation : environnements, références croisées, tableaux et figures.

# Présentation(Suite2)

## Environnements

- Utilisez des environnements pour faire ressortir les éléments importants du texte.
- **Définition 1.** Un environnement est une partie du texte qui a un rôle bien défini, comme un théorème, une définition, un exemple, une démonstration, un morceau de code.
- Cette partie du texte est formatée de telle sorte que l'on puisse identifier directement l'environnement et son rôle.

# Présentation(Suite3)

## Environnements(Suite1)

- Certains environnements sont numérotés pour que l'on puisse y faire référence ailleurs dans le texte.
- **Exemple 5.** La définition 1 est présentée dans un environnement utilise pour les définitions importantes.
- Cet exemple est lui-même représenté dans un environnement dont le rôle est d'illustrer certaines notions.

# Présentation(Suite4)

## Environnements(Suite2)

 L'Algorithme A a une complexité  $O(n^2)$  car (longs arguments. . .). Il donne le résultat souhaité car (longs arguments. . .).



**Proposition 1.** *La complexité de l'Algorithme A est en  $O(n^2)$  où  $n$  représente le nombre d'éléments dans le tableau donné en entrée.*

*Démonstration.* (longs arguments. . .)



**Théorème 1** (Exactitude de l'Algorithme A). *Quand l'Algorithme A se termine, le tableau résultat contient les nombres triés par ordre croissant.*

*Démonstration.* (longs arguments. . .)



# Présentation(Suite5)

## Références croisées

- **Définition 2.** Tout élément du texte (section, équation, définition, référence bibliographique, figure, théorème, tableau, etc.) qui possède une étiquette (sous la forme d'un numéro ou de quelques lettres) peut être cité dans le texte en utilisant cette étiquette.
- On appelle cette citation une référence croisée.
- **Exemple 6.** Nous avons introduit la notion d'environnement dans la définition 1.

# Présentation(Suite6)

## Figures et tableaux

- Les tableaux et les figures facilitent la lecture et permettent d'illustrer des résultats ou des observations.
- Une figure peut aider la compréhension mais ne constitue pas un argument en soi.
- Il faut justifier les choix ou les résultats dans le texte. Un tableau permet de résumer des éléments.
- Il peut par exemple être utilisé dans la comparaison de différentes approches, pour donner les résultats de tests ou d'une étude empirique.

# Présentation(Suite7)

## Figures et tableaux (Suite)

- Une figure ou un tableau doivent toujours être numérotés et accompagnés d'une légende.
- En général, la légende d'une figure est placée en dessous de celle-ci. C'est le contraire pour un tableau : la légende se trouve au dessus.
- N'utilisez pas de figures superflues qui ne sont pas référencées dans le texte. Une référence croisée dans le texte devrait apparaître pour chaque figure ou tableau.