

Cours de Système d'Information

Chapitre 3 - Codification et contrôle des données

Tarek Boutefara

Version 0.1, 2020-12-28

Table of Contents

1. Introduction	1
2. L'importance de la codification	1
3. Définition	1
4. Types de codifications	4
5. Les contrôles	8
6. Concluion	10

1. Introduction

Dans le chapitre précédent, nous avons modélisé la circulation de l'information. Dans ce cadre, nous avons utilisé le Diagramme des Flux d'Informations qui modélise les acteurs et les flux d'information entre eux.

Ce diagramme ne modélise pas l'information en elle-même. Il est conçu pour focaliser sur les échanges des informations sans analyser ou modéliser le contenu de ces flux. Il est aussi possible de référer au "support de l'information" comme un flux d'informations. A titre d'exemple, il est commun de référer à la "facture" comme étant un flux d'informations dans un diagramme des flux d'information. Néanmoins, une "facture" est un support de l'information qui contient plusieurs informations.

Dans ce chapitre, nous allons faire notre premier pas vers la manipulation de l'information. Pour ce faire, il est nécessaire de proposer une représentation claire pour chaque information du système. En d'autres termes, il est nécessaire de proposer "un codage" pour chaque "information" qui existe dans le système pour pouvoir la manipuler.

2. L'importance de la codification

Il y a une différence entre les types de données offerts par l'outil informatique et les concepts (notions) manipulés par le Système d'Information de l'organisation. La codification enlève cette différence par associer des codes aux différents concepts manipulés.

Associer un concept issu du monde réel (système existant) à un code (issu du monde informatique) veut dire qu'il est désormais possible de le stocker, le traiter et le partager. Cela est vrai aussi pour les objets liés au concept.

Exemple

La notion d'Etudiant est un "concept" issu de l'université; une organisation qui fait partie du monde réel. Une fois codifié, il devient possible de le stocker, le lier à d'autres concepts codifiés, et le partager.

Il devient aussi possible de stocker les données des différents étudiants (objets ou instances du concept). A leur tour, ces données peuvent être traitées et partagées.

3. Définition

3.1. Un code

Un nom abrégé ou une représentation de l'information permettant de désigner un objet ou un concept de manière unique.

Exemples

- Codes des concepts

Code d'Etudiant → Code_E

- Codes des objets

Un étudiant (bac 2008) → 2018/0049

Un autre étudiant (bac 2017) → 2017/0033

3.2. Codification

La codification est l'opération qui consiste à remplacer une information sous sa forme naturelle par un code claire qui serait mieux adapté aux besoins de l'utilisateur.

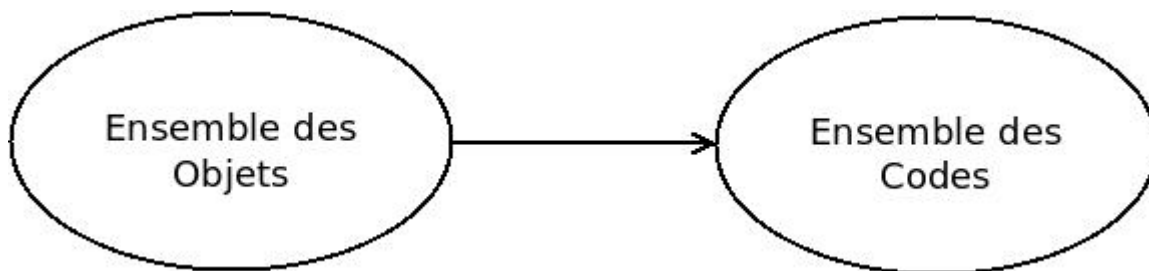


Figure 1. Codification

En programmation, la codification fait référence :

- Au choix des noms des variables,
- Aux valeurs distinctives de ces variables.

3.3. Caractéristiques d'une codification

1. **Non ambiguë** : une codification doit représenter d'une manière claire, précise et unique le concept ou l'objet en question.
2. **Facilité d'utilisation** : il est possible de proposer plusieurs codifications pour le même concept ou bien pour le même objet. Il faut opter pour la codification la plus simple et la plus facile à utiliser.
3. **Concision** : la codification proposée doit être plus courte que les informations codifiées. Autrement, le code sera plus difficile à manipuler par rapport à l'information originale.
4. **Possibilité d'extension et d'insertion** : le caractère dynamique des organisations implique un changement constant des concepts et des objets manipulés. Ainsi, la codification proposée doit supporter :
 - a. L'extension : elle doit permettre d'ajouter de nouveaux codes (de concepts et d'objets).
 - b. L'insertion : si la codification présente un certain ordre, elle doit permettre l'insertion des éléments dans leurs bonnes positions entre les objets existants (suivant l'ordre défini).

Exemples

Exemple 01

Dans le cadre de réalisation d'un système de gestion pour l'université, un premier concepteur modélisait la partie concernant les étudiants. Il a proposé :

Code Etudiant → Code_E

Dans le cadre du même projet, un autre concepteur travaillait sur la partie qui concerne les employés de l'université. Il a proposé :

Code Employé → Code_E

Dans ce cas, Code_E est un code ambigu.

Exemple 02

Soit les deux codes suivants proposés pour Code Etudiant :

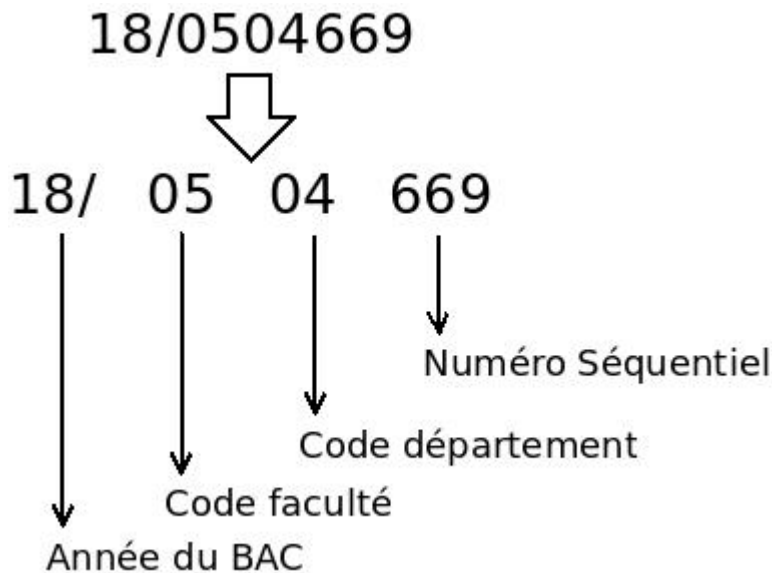


Figure 2. Première proposition pour le code d'étudiant

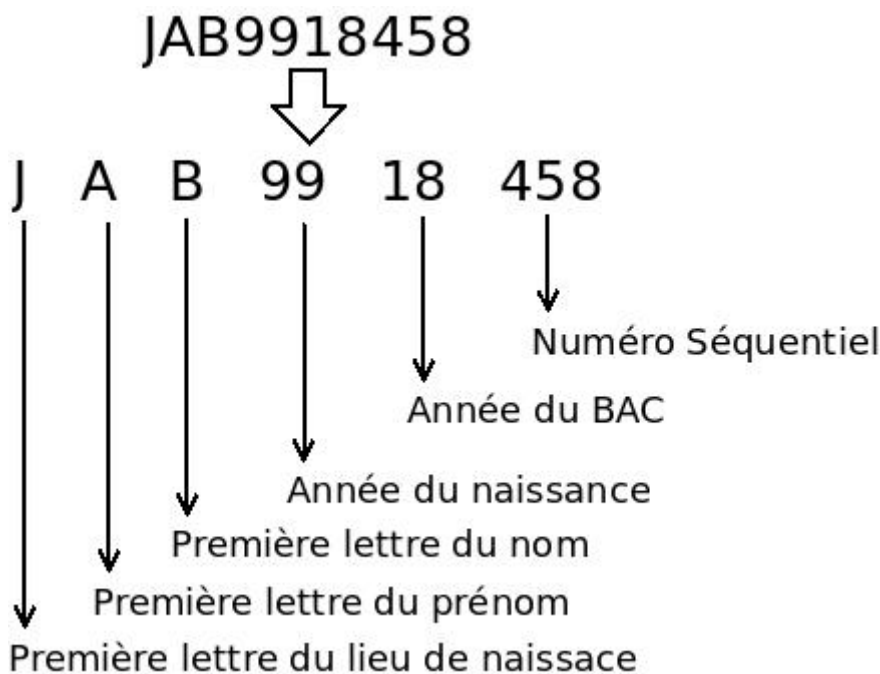


Figure 3. Deuxième proposition pour le code d'étudiant

Malgré que les deux codes sont "corrects" vu qu'ils permettent de définir d'une manière précise et unique un étudiant. La première proposition est plus simple et plus facile à manipuler.

4. Types de codifications

4.1. Coditification séquentielle

Associer des numéros séquentiels (consécutifs) aux informations.

Avantages	Inconvénients
<ul style="list-style-type: none">• Non ambiguë : chaque code (numéro) est unique,• Simple à mettre en oeuvre : il suffit de mettre oeuvre un compteur,• Extension possible : il est toujours possible d'ajouter des codes.	<ul style="list-style-type: none">• Insertion impossible : il est impossible de créer un nouveau code entre deux codes (deux numéros) consécutifs déjà utilisés,• Non significative : le code n'a aucune signification et sert uniquement à identifier d'une manière unique un objet.

4.2. Codification par tranche

Associer des tranches de codes aux catégories. Chaque tranche contient des codes séquentiels.

Avantages	Inconvénients
<ul style="list-style-type: none">• Non ambiguë : chaque code (numéro) est unique,• Insertion et extension possibles : il est toujours possible d'insérer de nouveaux codes dans une catégories même si les catégories suivantes contiennent déjà des codes,• Possibilité de contrôle : il est possible de mettre en oeuvre qui contrôle est-ce que la valeur saisie est valide ou pas.	<ul style="list-style-type: none">• Possibilité de saturation d'une tranche : l'insertion ne devient plus possible,• Un seul regroupement est possible.

Exemple

Les codes de réponse du protocole HTTP utilisent une codification par tranche :

1xx Réponse informationnelle	<ul style="list-style-type: none">• 100 Continue• 101 Switching Protocols• 102 Processing• 103 Early Hints
-------------------------------------	---

2xx Succès	<ul style="list-style-type: none"> • 200 OK • 201 Created • 202 Accepted • 203 Non-Authoritative Information • 204 No Content • ...
3xx Redirection	<ul style="list-style-type: none"> • 300 Multiple Choices • 301 Moved Permanently • 302 Found • 303 See Other • 304 Not Modified • ...
4xx Erreur (par le client)	<ul style="list-style-type: none"> • 400 Bad Request • 401 Unauthorized • 402 Payment Required • 403 Forbidden • 404 Not Found • ...
5xx Erreur (par le serveur)	<ul style="list-style-type: none"> • 500 Internal Server Error • 501 Not Implemented • 502 Bad Gateway • 503 Service Unavailable • 504 Gateway Timeout • ...

4.3. Codification articulée

Utilisation des codes découpés en zones (descripteurs). Chaque descripteur a une signification particulière.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Non ambiguë : chaque code (numéro) est unique, • Insertion et extension possibles : il est toujours possible d'insérer de nouveaux codes dans une catégorie même si les catégories suivantes contiennent déjà des codes, • Possibilité de regrouper les objets selon un critère donné : chaque descripteur offre une possibilité de regroupement, • Possibilité de contrôle : il est possible de mettre en oeuvre qui contrôle est-ce que la valeur saisie est valide ou pas. 	<ul style="list-style-type: none"> • Codes difficiles à manipuler : vu qu'il est composé de plusieurs 'codes', • Possibilité de saturation d'un descripteur : l'insertion ne devient plus possible, • Instabilité : si un descripteur change, tout le code doit être changé.

Exemple

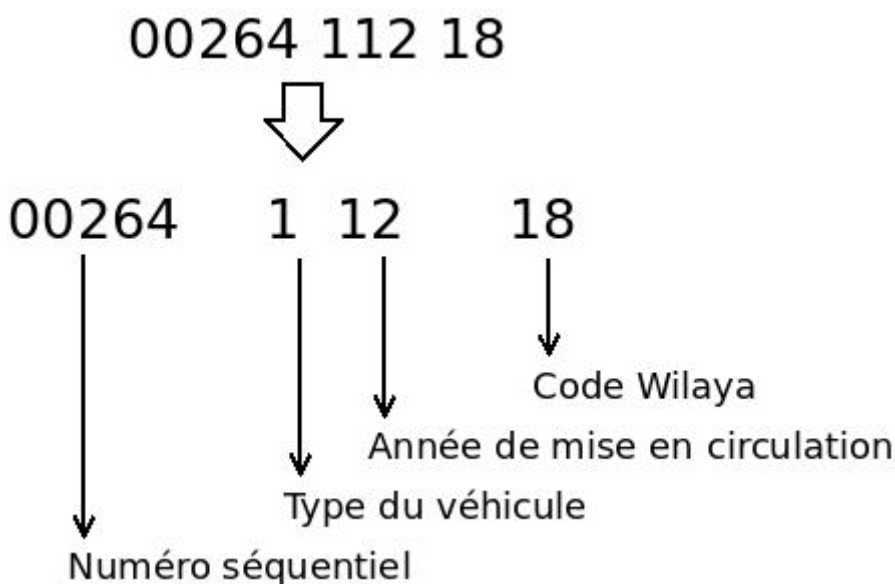


Figure 4. Matricule d'un véhicule

4.4. Codification par niveau

Cas particulier de la codification articulée où les descripteurs représentent des niveaux.

Exemple

Le code postale peut être vu comme une codification par niveau parce que la ville (ou la région) est une partie de (appartient à) la Daïra qui fait, à son tour, partie de la Wilaya.

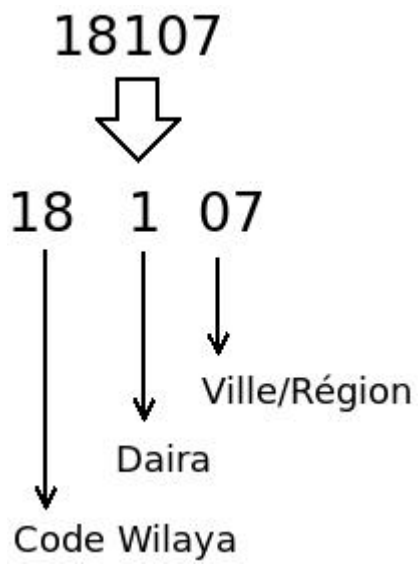


Figure 5. CodePostal

4.5. Codification mnémonique

Abréviation de la désignation en utilisant un ensemble réduit de caractères.

Consonants

Total → ttl

Abréviatifs

Total → tot

5. Les contrôles

Dans un système d'information, plusieurs contrôles sont effectués :

- Contrôle des traitements,
- Contrôle des résultats,
- Contrôle des informations.

Dans cette partie, nous nous intéressons à ce dernier type.

5.1. Définition

Un contrôle est la vérification que les caractéristiques d'une information sont conformes à ce qu'elles doivent être.

5.2. Types des contrôles

Contrôles directs

Ces contrôles se rapportent à l'information elle-même :

Contrôle de présence

L'information figure bien sur le support.

Exemple

Champs obligatoires dans un formulaire.

Contrôle de type

(numérique, alphanumérique, date)

Exemple

L'âge ne peut pas contenir des caractères.

Contrôle du cadrage

(position dans la zone de saisie)

Exemple

Les types alphanumériques sont alignés à gauche tandis que les types numériques sont alignés à droite.

	A
1	Alphanumérique
2	25
3	

Figure 6. Exemple du cadrage sous Excel

Contrôle de vraisemblance

S'assurer que la valeur de l'information est semblable à une valeur correcte (par comparaison à des règles, des valeurs précédentes, ou des valeurs issues du secteur professionnel).

Exemple

La date doit être valide (30/02/2020, par exemple, a une forme correcte mais ce n'est pas une date valide).

Contrôles indirects

Des contrôles qui vérifient une information par comparaison à d'autres informations.

Contrôle de cohérence interne

(Codification articulée)

Exemple

Si la Wilaya 18 compte 6 Daïras, alors le code postale "18721" ne peut pas être correct.

Contrôle de cohérence externe

Comparaison avec d'autres informations.

Exemple

La date d'obtention du diplôme doit être supérieure (après) la date d'inscription.

5.3. Ordre d'exécution des contrôles

Les différents contrôles sont exécutés dans l'ordre suivant :

1. Présence,
2. Type,
3. Cadrage,
4. Vraisemblance,
5. Cohérence interne,
6. Cohérence externe.

6. Concluision

Dans ce chapitre, nous avons vu la notion de codification. Cette notion permet de ramener les notions et concepts manipulés par l'organisation vers le monde informatique en proposant une représentation adéquate. Cette représentation permettra l'utilisation de l'outil informatique de stocker, traiter et communiquer ces concepts.

La codification nécessite des actions du contrôle pour une utilisation correcte et pertinente de cette représentation. Ainsi, nous avons évoqué les différents contrôles possibles sur une information.