

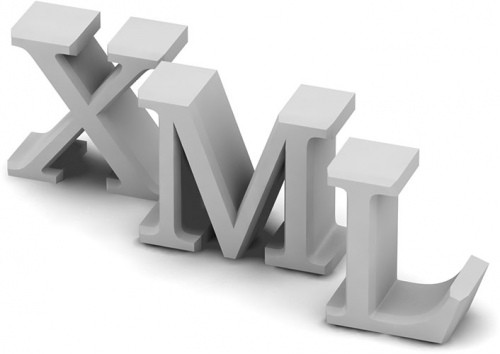
Benoît Julien, Catel Torres A. Gabriel

HEIG-VD

mercredi, 13 mars 2019

SER : labo 01

**DTD et XML**



# INTRODUCTION

Le but de ce premier laboratoire de sérialisation est de comprendre le principe de DTD. Nous allons mettre en place un ensemble de règles qui permettra une rédaction de document HTML qui sera conforme au cahier des charges reçu (ici la donnée du laboratoire).

Dans un premier temps il nous faudra donc réfléchir au niveau d’abstraction et de détail que l’on souhaitera avoir lors du traitement des informations récupérées dans le document XML fourni, il nous faudra nous mettre dans la peau de la personne qui aura ce document et rendre la récupération la plus ergonomique possible.

Pour cela il faudra écrire une DTD qui soit la plus cohérente possible d’après les contraintes que nous allons définir mais il nous faudra malgré tout prendre en compte les possibilités que nous propose le XML. En fin de compte le but est d’avoir un projet final a mi-chemin entre les contraintes du XML et les contraintes que nous avons définies (en cherchant tout de même a être le plus proche possible de notre propre définition du problème).

# DTD REALISEE

**Voici ci-dessous le code de notre DTD :**



**Détails des choix effectués pour l’écriture de la DTD :**

Les choix effectués ci-dessous ont été discuté dans l’optique d’avoir un document le plus facile à traiter au niveau de la structure.

- Pour ce projet nous allons considérer qu’une saison d’échec se déroulant sur un temps non défini est la structure qui permettra de stocker les différents tournois dans lesquels les joueurs auront un ELO défini par leurs performances.

- Un détail important de la modélisation est que nous souhaitions qu’il soit pratique pour l’utilisateur de récupérer des blocs d’informations qui soient le plus compréhensible possible. Ainsi nous avons pensé définir trois grands blocs réunissant les informations nécessaires pour réaliser les tournois.

1. Structure joueurs, qui va répertorier tous les joueurs participant à la saison d’échec

2. Structure arbitres, qui fera la même chose pour les arbitres d’une saison

3. Structure tournois, dans laquelle nous retrouverons tous les tournois d’une saison

Cette structure nous permet ainsi de réutiliser les arbitres et les joueurs d’une saison dans différents tournois sans avoir besoin de les redéclarés.

- Pour le groupe des joueurs, et des arbitres, nous forçons la création d’au moins un joueur et un arbitre car une saison n’a pas de sens d’exister s’il n’y a personne pour y participer. Dans l’absolu, nous aurions aussi pu donner la possibilité de créer une saison sans joueurs ni arbitres. C’est un choix d’implémentation que nous avons fait. En revanche nous permettons qu’une saison n’ai pas de tournoi dans un premier temps et une saison doit pouvoir être créée avec une base de joueurs et d’arbitres qui seront choisis par la suite pour créer des tournois.

- Avoir un ID différent et unique pour chaque joueur permet d’utiliser une IDREFS pour se référer à un joueur lorsque l’on souhaite renseigner quels joueurs ont participés à une partie (puis pour déterminer un vainqueur etc…).

- Plutôt que de mettre dans chaque partie deux joueurs dans lequel nous aurions pu mettre un élément couleur qui renseignerai quel couleur il jouait durant cette partie, nous avons mis deux éléments JoueurBlanc et JoueurNoir qui permet directement d’avoir cette information avec le nom du champ. L’ID du joueur sera ensuite renseigné comme attribut.

- Dans le cas des déplacements, nous avons il n’est pas nécessaire d’avoir l’information concernant la case de départ en revanche nous avons laissé la possibilité d’avoir les coups spéciaux présents tous les deux dans un même élément. En effet de notre point de vue (du fait de ne pas vérifier les entrées d’informations éventuelles d’informations dans un fichier XML) un pion qui procède à une élimination peut aussi être soumis à une promotion. En revanche l’utilisateur devra faire attention à ne pas soumettre de promotion à une autre pièce qu’un pion.

Ces différentes explications justifient suffisamment de notre point de vue tous les cas qui pouvaient se discuter quant à leur implémentation.

# TD REALISEE

**Vo**

Imaginons que vous souhaitez enregistrer le classement ELO que chaque joueur

d’une partie avait au moment où elle a été jouée, qu’est-ce qu’il faudrait modifier

dans votre DTD ?

Il existe plusieurs manières de pallier au problème. Il serait par exemple possible d’ajouter dans l’élément JoueurNoir et JoueurBlanc d’une partie l’ELO qu’il a lorsque la partie est créée.

Est-ce possible dans votre DTD de représenter le fait qu’il ne peut y avoir que 20

parties au maximum dans un tournoi ? Si oui, comment ?

Non c’est impossible

Est-ce possible dans votre DTD de représenter le fait que les 2 joueurs d’une partie

doivent être différents ? Si oui, comment ?

Non c’est impossible