**1** The text I selected is a wikipedia article on footballer Fernando Torres <https://en.wikipedia.org/wiki/Fernando_Torres> . The format followed by wikipedia for majority of footballers is same and there are few variations but the basic structure remains somewhat similar. I have taken the page in the mentioned link as reference in our case.  This document is a structured text document.

**2** The document has more than 10 elements  and have more than 5 attributes 2 of which take a controlled list of value. Position attribute and award type take a controlled list of values. The attributes are either required or implied/optional depending on the data/text present in the page. The nesting of the elements has also been shown in the DTD depending on the content of the document.

**3** The document has been attached in the solution

**4Write prose documentation for each element, attribute, and attribute value.**

On the top level document has mainly 9 elements. Introduction of the player. Personal information of the player.,Early career,Club Career,

International Career,Style of Play,personal life,career statistics  and honours.

Introduction is basically the introduction of player. This element has three attributes name of the player,playing position of the player and country for which player played. The value of the element contains the introduction about the player and playing career. All the attributes here are required which are mentioned in the DTD.

Personal information of the player contains full name,place of birth,date of birth and a brief overview of statistics of the player.

This element contains the nested elements ((fullname,dateofbirth,placeofbirth,statistics\*). Full name date of birth and place of birth contains the information and statistics again contain nested elements with details of year,team,apps and goals. Statistics with an \* shows that 0 or more of this could be present in the XML document as a player may play for different clubs.

Early career of the player contains textual data of the details of the youth career of the player

Club career contains the details of the the players club career. A player can play for multiple clubs and we could have data either season wise or combined. Therefore, the team attribute is required and season is optional in case the consolidated text data for all the seasons for one club is present.

International career of the player contains textual data about the international career of the player and since a player only plays for one country hence no attribute is chosen for this like the club career where a player can play for multiple clubs.

Style of play contains the data about the playing style of the player and is text data only.

Personal\_life also contains the data about the personal life of the player and contains text data.

Career\_statistics contain the goals scored per season by player and deep dives into each season. Career statistics contains set of statistics which is reused from the personal\_information but contains more detailed data by populating the inner elements of statistics accordingly.

Honours contain the awards/medals won by the player. We have chosen attribute team which is optional because they are not required for individual awards and decorations. The awardtype contains a set of 3 values(team,individual and decorations) and is mandatory. It then contains the list of awards on the basis of combination of awardtype and team.

**5** The markup text has been attached and internal DTD has been used .

**6**

* **How did you decide to represent the data in the way that you did? Why did you choose the elements and attributes that you did?**

            The outermost structure of the document was somewhat already defined as it was a well structured document. Introduction of the player contains attributes name,country and position. There are taken as attributes because essentially they are meta data for introduction of the player and not the actual textual data. Position takes fixed set of values which is standard in football. The introduction contains basic details about the player.

           Personal information of the player contains details about the players and statistics of the player. This contains goals scored for clubs and country and hence there could be more than one statistics element depending on the number of clubs a player has played for. So multiple statistics elements are present to incorporate multiple rows. The year team appearance and goals translated into elements inside statistics as they are data points.

Early career of the player is mainly text and hence there are not attributes required for this element and no nested elements required too.

Club career of a player is again a text element. This element could have been nested and could have contained every season's data but some of the data is missing and for some of the clubs the consolidated data of certain years have been provided hence the idea was to choose attributes for the season and team. While team is a required/mandatory field ,season is not. So for teams where consolidated data is there we will not have a season attribute and for teams where we have every season's data we will have season attribute. For eg the data for teams Liverpool and Chelsea have all the seasons but for Atletico Madrid we have consolidated data.

International career of the player contains the details about international career. To include a team attribute here would have been redundant because we already have a country attribute for player and international team is same as country. Also since a player doesn't change international teams a consolidated textual data representing the entire text corresponding to international career is acceptable.

Style of play and personal life mainly contains text data and doesn't need nesting of elements or any attributes.

Career statistics contains the statistics of the player for each season and the same has been populated from the table present in the document. Only the valid data has been taken from the table. The club season goals and appearance gives a clear picture of the player's season. The xml document only contains few entries of this table for sample.

Honours contain the awards/medals won by the player. The idea to use attributes was to minimize the duplication as team name and type of award would be duplicated for a lot of entries. Therefore an attribute for award type was chosen which contains a set of values(team,individual,decoration) . For team awards there could be duplication of teams if we had team as a nested attribute. The idea is to list down awards in a single element (with common awardtype and team). i.e. trophies won at chelsea would come inside a single xml element and similarly trophies won at Atletico Madrid would be inside a single element and same goes for individual awards.

* **What were the hardest decisions you had to make in this design process?**

The hardest decisions were to choose between attributes and elements in certain places which has been discussed in above paragraph. The approach followed to resolve it was to use attributes for meta-data and elements for the data. For eg to have multiple elements for each award with team and type as nested elements could have been chosen but it would have lead to duplication of data. Also the actual data is award name and the other data like team and type are meta data for the awards. So instead of having team and type as elements we had it as attributes.

Similarly to structure elements and how to nest elements was also one of the focal points for designing the structure of the document. How to avoid duplication of attributes and how to reuse elements if they can be used as nested elements in more than one elements. For eg statistics element has been used as child element for two other elements in the schema.

* **How does your DTD design support data independence?**

The schema is not impacted by how the data is stored. The same schema can be stored in multiple types of physical storage that is take care by file management software. The schema can be used by multiple applications just by validating the structure and the actual data can be stored accordingly. Our application can use this standard DTD to verify data irrespective of how and where it is stored.

Adding new data to schema can be done easily by adding elements and attributes. For eg as of now we only have goals and appearance data even for players other than those playing in forward position. Eg in future if we add other statistics like clean sheets, assists to the document, we will be able to incorporate that by few changes in our schema by adding new elements or attributes depending on how new data is represented in the text document.

* **How may your DTD design support the overarching goals of data curation (revisit objectives and activities of Week 1)?**
* Collection : The data can be collected from various such wikipedia pages to populate the xml document.
* Organization : We are using tree data/DTD model to organize the data
* Storage: This XML document can be stored on any OS and validation of the data can be done as well.
* Preservation The attributes and elements will help in understanding the data. We have a DTD schema and that will help our application in identifying data present inside xml elements
* Discoverability: The data can be searched using the elements. There are various frameworks with which we can access our data using keys in our application.
* Access The access can be taken care by the File System where the files are stored.
* Workflow Support the ability to systematize data workflows. This will also be take care by publishing software. A utility can be written on the top of the apis to automate various workflows and that can be documented. Eg we can document how data will be populated from text document to XML schema and vice versa.
* Identification We have various constraints and attributes and keys which will help in authenticating and validating the data. The sample text has also been validated using the online validator
* Integration The data can be integrated from various text documents of such type using the XML schema to represent data.
* Security: This is also managed by File System but we need to ensure that the permissions and ACLs are managed accordingly
* Compliance This will be ensured by the legal team of the organization. Any data that we need to populate in the xml document should be first approved by the legal team.
* **What are the pros and cons of your DTD design?**

**Pros of design**

* The design is independent of platform and programming language and hence will work on every system and technology.
* Our design can represent human readable text in XML which can be processed by the application
* The schema is designed in such a way that we want to add data/elements and remove elements from our schema it will be easy and will not affect other elements
* The validation ensures that our data is validated and it’s free from any discrepancies
* Our design works for all the languages i.e. even if the text data is not in English language it will still be able to represent the data. i.e. schema is independent of the contents of the data and relies on the structure of the data,

**Cons of design**

* There are few redundancies in the schema. While it is a tradeoff between easily understanding the tag structure and optimizing the storage, if there are redundancies in the structure some of the data will be repeated and this will lead to bloating of the xml document which in turn will slow down the validation and processing of the document.
* Our design will not be efficiently able to represent tables as relational model is better suited for such data. There are few tables in our document and hence some of the data is not normalized because of our DTG schema.