

# UE MOBJ : Compte-rendu TME6

DUPUY, LEGOUEIX

25 novembre 2020

## 1 `Term::fromXml()`

Dans `Term::fromXml()` on crée un `Term`.

Objet simple : donc pas besoin de boucle `while(true)`.

On instancie les attributs les uns après les autres en fait appelle à la primitive `xmlTextReaderGetAttribute()`.

## 2 `Instance::fromXml()`

Idem que pour `Term::fromXml()`.

## 3 `Net::fromXml()`

Nous pourrions écrire un roman sur les raisons qui nous ont amené à trouver une solution pour le cas du `Net`. Mais nous dirons seulement que de vouloir faire la même boucle que dans `Cell::fromXml()` était une mauvaise idée, et que la veille au soir à très exactement 23h30 nous trouvâmes la solution à ce problème.

En effet, la solution était bien plus simple ! Il suffisait simplement de reprendre la structure de la fonction `::fromXml()` comme définie dans `Term`, `Instance` ou bien `Node`, puis d'ajouter une boucle après avoir créé le `Net`. Maintenant que nous savons la vérité, notre regard sur notre survie dans l'UE MOBJ a changé. Tout est possible !

## 4 `Node::fromXml()`

La primitive de `Node::fromXml()` est différente de celle des autres objet de la `Netlist` car l'objet `Node` lors de sa création doit effectuer la liaison du `Net` auquel il est rattaché avec le `Term` qui l'encapsule.

De plus, comme les `Term` ont automatiquement créé leur `Node`, il s'agit alors de mettre à jour les attributs du `Node` encapsulé.