***Metamodel and Model***

* A new version of the metamodel has been defined, trying to reduce the number of different WTC types.
  + Ecore file: WTSpec4M50.ecore
  + Ns URI: http://WTSpec4M/5.0
  + Only 34 different WTC types are considered now
* A new example model has been prepared, based on the previous one but conforming to this *new* metamodel.
  + WindPower4MONDO.wtspec4m

***User roles***

There will be six different user roles working with the collaborative modeling tool. They are explained below.

***User 1: WT Principal Engineer***

* Can add, remove and modify all types of item in a *Wind Turbine Control System (aka WTCS)* model.
* It is a kind of “*super user”* for editing *WTCS* models.

***User 2:*** ***WT IO manager***

* Can add, remove or modify *SystemInput* and *SystemOutput* items.
* Cannot edit, nor even view any other type of item.
* He/she will only work with *SystemInput* and *SystemOutput*, and has no access to any other element in the model.
* *SystemInput* and *SystemOutput* entities could be grouped creating a new entity in the metamodel as container of them.

***User 3:*** ***WT manager***

* Can add and remove *MainSubsystem* items for a WT
* Can modify the properties of any *MainSubsystem* item
* Can add SystemVariable, SystemFault, SystemTimer items
* Can and SystemParam items and set values to all its attributes
* The difference of ***WT Manager*** role compared to ***WT Principal Engineer*** role is that the latter can add or remove SystemInput and SystemOutput items as well as modify them, while the first onecan only view them and reference them from WTC items. So, SystemInput and SystemOutput items are read only items for a ***WT manager***.

***User4: Subsystem manager***

* Can only edit the content of a MainSubsystem item
* Can add and remove Subsystem items contained by the MainSubsystem item
* Can add WTC items in a Subsystem or MainSubsystem
* Can remove WTC items contained by a Subsystem or MainSubsystem
* Can reference *SystemInput*, *SystemOutput*, *SystemParam, SystemFault, SystemTimer, and SystemVariable* items from WTCs

***User5: “Specific” subsystem manager***

As a more fine grained rule than the one specified for User4, would it be possible to define a control access rule depending on some property of a domain concept?

Eg. would it be possible to define rules like

* “Peter (or *Generator control engineer*) can edit a MainSubsystem when it is named (value of sysId property) as ‘Generator’”
* “Mary (or *Converter control engineer*) can edit a MainSubsystem when it is named (value of sysId property) as *Converter*”
* “John (or *Pitch control engineer*) can edit a MainSubsystem when it is named (value of sysId property) as *Pitch*”

***User6: WT Service Engineer***

* Can **only** modify *Enabled* attribute value for any WTC
* Can modify any WTC’s references to its inputs (can modify referenced items only for WTCInput type items)
* Can see SystemParam items, but only when their *Settable* attribute’s value is **true**. Under such condition
  + Can see SysId and Description attributes for the SystemParam
  + Can **only** modify the value of Value attribute

If the access rules for this user are too tricky, we can only consider the first condition, modification of enabled attribute.

***Locking***

* A ***WT Principal Engineer*** can lock the whole model or a fragment of a model (e.g. a MainSubsystem).
* A ***WT Manager*** can lock the same items of a ***WT Principal Engineer*** as well.
* A ***WT IO manager*** can lock the set of SystemInput and SystemOutput items to prevent other users (a *WT Principal Engineer*) from removing them.
* A ***Subsystem Manager*** can lock the MainSubsystem he/she is working with.
* A ***Subsystem Manager*** can lock any Subsystem (and all the contained Subsystems recursively) contained by the MainSubsystem.

***Obfuscation***

***Fragmentation strategy***

* A *reasonable* fragmentation strategy could be to split a model into
  + one file for SystemInput and SystemOutput items
  + one file for SystemParam items
  + one file for the WT. This file will contain the root element (WT) and also SystemVariable, SystemFault and SystemTimer items.
  + one file for each ***MainSubsystem*** item contained by WT. All Subsystem and WTC items contained by the MainSubsystem should also be stored in the same file.