**STG(14)30**

**SEAMCAT model exposure for plugin developers and proposed re-farming**

We are now at a stage where we are exposing relevant model data for all system types (Generic, CDMA UpLink/DownLink, OFDMA UpLink/DownLink) for plugin developers. This allows SEAMCAT to not be a black box in terms of interference calculations, and also allow for extension by plugins for producing additional results.

To make this work a lot of re-factoring was needed, especially for defining an 'interference calculation engine' (the engine class is called InterferenceSimulationEngine) that would have a level of abstraction that was suiting, i.e. not too abstract like a black box 'simulate', nor too detailed as the old version handling all of the system types which lead to an explosion of complexity and virtually made it impossible to extend the system any further.

As mentioned, we are now at a satisfactory level of making all this possible. The SystemResult is exposed to plugin writers and from this SystemResult most of the intermediate results can be inspected. The downside is that we have not reached a completely stable version of the model code. The problem is two folds:

1. Abstractions are not the same.

For Generic and Cellular the results are exposed in different types. For Generic the results are of type InterferenceLinkResult and for Cellular the results are collected in SimulationElement (either a mobile station or a base station).

Looking at the code, however, there is no need for this dichotomy. Everything should be collected in InterferenceLinkResults - because that is the most accurate definition of what they represent.

2. External Interference calculation scattered.

As mentioned in 1. the abstractions are not aligned among the systems. This is only for historic reasons. But it has had the consequence of calculations being scattered throughout the application. Some of newly performed re-factoring has been aligning parts of this calculation, i.e. the InterferenceCalculator handles the unwanted and blocking interference for any system now. From a high level perspective the calculation of the external interference is done by combining the victim system active receivers with the interfering system(s) active transmitters. Looking at the code this intent is quite evident but as mentioned above has been scattered and duplicated multiple places.

The problem we currently face is that fixing the two issues stated above could turn out to be a Pandora's box in time to implement. We believe that a 'good enough' version for now is more beneficial as there are many other significant tasks on our to-do list for this year. This is also in understanding of the certain breakage of existing plugins as this is re-factored to the desired solution.

In conclusion the current state of the plugin model is 'good enough', but as highlighted above not complete. Once the two issues above is addressed the logical place to conduct the external interference calculation is inside the InterferenceSimulationEngine - this would make the intent of the engine more obvious and further down the road, if SEAMCAT would be extended with several other systems, make it much easier to do so.