

**iRobotSma**

**Clastrier & Debat**

**ARCHITECTURE**

Récapitulatif des exigences

|  |  |  |
| --- | --- | --- |
| **CODE** | **DESIGNATION** | **VALIDATION** |
| ENF 1 | De sa création jusqu’à son suicide, un agent répète un cycle d’exécution composé de trois opérations séquentielles Percevoir-Décider-Agir. | **OK** |
| ENF 2 | Un agent peut créer d’autres agents. |  |
| ENF 3 | Un agent peut se suicider mais ne peut détruire un autre agent. |  |
| ENF 4 | Il doit être possible de visualiser l’exécution du système. | **OK** |
| ENF 5 | Il doit être possible de visualiser l’état du système et/ou d’agents sélectionnés. | **OK** |
| ENF 6 | Le système de visualisation doit être le plus découplé possible du reste du système. | **OK** |
| ENF 7 | Le langage de programmation doit être Java. | **OK** |
| ENF 8 | Le langage de description des composants et des assemblages doit être SpeADL. | **OK** |
| ENF 9 | Vous devez quantifier le nombre de classes et de lignes de code écrites (et pas générées). |  |
| ENF 10 | La réutilisabilité/généricité de votre solution doit être évaluable objectivement, mesurée et la plus forte possible. |  |
| ENF 11 | La paramétrisation de votre solution pour le cas d’étude doit être complètement spécifiée. |  |
| ENF 12 | Votre architecture doit être décrite le plus précisément possible (vues C&C, module et allocation) | **OK** |
| ENF 13 | Il doit être possible de contrôler l’exécution du système (mettre en pause, pas à pas , vitesse plus ou moins rapide). |  |
| ENF 14 | Il doit être possible d’obtenir une trace d’une exécution d’un agent sous forme de log. |  |
| ENF 15 | Il doit être possible de rejouer une exécution d’un scenario avec un état initial donné, y compris en présence de phénomènes aléatoires. |  |
| ENF 16 | Il doit être possible de persister l’état du système. |  |
| ENF 17 | Il doit être possible de répartir l’exécution du système. | **KO** |