1 Les actions A

1.1 Les actions relatives aux objets

1.2 Les actions relatives aux aménagements

1.2.1 A_DepositObjectInAmenagementLargeQuantity

L'action permet de déposer N objets dans un aménagement d'un agent du groupe si l'on se trouve sur l'aménagement en question. La signature de l'action est A_DepositObjectInAmenagementLargeQuantity(Modified Object, N) :

— Modified Object, type de l'objet à déposer dans l'aménagement.
— N, le nombre d'objets à déposer.

1.2.2 A_withdrawObjectInAmenagementLargeQuantity

L'action permet de retirer N objets dans un aménagement d'un agent du groupe si l'on se trouve sur l'aménagement en question. La signature de l'action est A_withdrawObjectInAmenagementLargeQuantity(Modified Object, N) :

— Modified Object, type de l'objet à retirer de l'aménagement.
— N, le nombre d'objets à retirer.

1.3 Les actions relatives au cognitons

1.3.1 A_ChangeCognitonWeightInteger

Cette action permet d'ajouter N au poids d'un cogniton dans l'esprit de l'agent. La signature de l'action est A_ChangeCognitonWeightInteger(Modified Cogniton, N) :

— Modified Cogniton, le cogniton dont le poids est à modifier.
— N, le nombre à ajouter au poids du cogniton dans l'esprit de l'agent.

1.4 Les actions relatives aux déplacements

1.4.1 A_GoToGroupFaicility

Cette action ramène l'agent à un aménagement d'un des membres de son groupe. La signature de l'action est A_GoToGroupFaicility(Facility) :

— Facility, le type d'aménagement vers lequel l'agent doit se diriger.

1.5 Les actions relatives aux agents

1.5.1 A_HireForRole

Cette action recrute un agent sans groupe se trouvant sur le même patch que l'agent et lui donne un rôle dans son groupe. La signature de l'action est A_HireForRole(Role) :

— Role, le rôle dans lequel le nouvel agent recruté sera affecté dans le groupe.

1.5.2 A_AskRandomMemberToChangeRoleForAnother

Cette action change le rôle d'un des membre du groupe de l'agent pour un autre rôle spécifié (s'il n'est pas déjà dans ce rôle). La signature de l'action est A_AskRandomMemberToChangeRoleForAnother(Role) :

— Role, le rôle vers lequel un membre du groupe se convertira.

1.5.3 A_BirthGroupAndRole

Cette action ressemble à l'action A_Birth, elle créer un nouvel agent en appelant le BirthPlan puis ajoute le nouveau membre au groupe en lui affectant un rôle. La signature de l'action est A_BirthGroupAndRole(Role) :

— Role, le rôle vers lequel le nouvel agent sera affecté dans le groupe.

1.5.4 A_ChangeAttributeDouble

Cette action ressemble à l'action A_ChangeAttribute, elle permet de modifier l'attribut d'un agent en ajoutant un N (double) à la valeur de l'attribut. La signature de l'action est A_ChangeAttributeDouble(Modified Attribute, N) :

— Modified Attribute, attribut de l'agent à modifier.

— N, la valeur à ajouter à la valeur de l'attribut.

1.5.5 A_ChangeRoleForAnother

Cette action change le rôle de l'agent pour un autre rôle du groupe. La signature de l'action est A_ChangeRoleForAnother(Role) :

— Role, le rôle vers lequel l'agent se convertira.

1.5.6 A_DieAndRemoveFacilities

Cette action fait mourir l'agent et efface tous ses aménagements. Cette action ne possède pas d'arguments.

1.5.7 A_DieAndRemoveSpecificFacility

Cette action fait mourir l'agent et n'efface qu'un type d'aménagement. La signature de l'action est A_DieAndRemoveSpecificFacility(Facility) :

— Facility, le type d'aménagements supprimés à la mort de l'agent.

1.5.8 A_setPoissonLaw

Cette action modifie un attribut et lui donne la valeur d'un nombre suivant un loi de poisson de paramètre $\lambda = 10$ et $n = 20$ et multiplié par 5. La signature de l'action est A_setPoissonLaw(Modified Attribute) :

— Modified Attribute, est l'attribut modifié.

2 Les action L

2.1 Les Tests

2.1.1 L_CompareAttributeToAttribute

La condition porte sur les valeurs de deux attributs de l'agent, elles sont comparés par un comparateur. La signature de cette action est L_CompareAttributeToAttribute(Attribute1, comparator, Attribute2) :

— Attribute1 est le premier attribut concerné.
— comparator est l'opérateur de comparaison $<, >, \leq, \geq, ==$.
— Attribute2 est le deuxième attribut concerné.

2.1.2 L_CompareAttributeToConstant

La condition porte sur la valeur d'un attribut par rapport à une constante. La signature de cette action est L_CompareAttributeToConstant(Attribute, comparator, Constant) :

— Attribute est l'attribut concerné.
— comparator est l'opérateur de comparaison $<, >, \leq, \geq, ==$.
— Constant est la constante concernée.

2.1.3 L_CompareGroupAttributeToPopulation

La condition porte sur un la somme des attributs des membres d'un groupe par rapport au nombre d'individus de ce groupe. La signature de cette action est L_CompareGroupAttributeToPopulation(Attribute, comparator) :

— Attribute, est l'attribut de groupe à comparer.
— comparator est l'opérateur de comparaison $<, >, \leq, \geq, ==$.

2.1.4 L_CompareRoleMembers

La condition porte sur le nombre d'individus du groupe par rapport à un nombre N. La signature de cette action est L_CompareRoleMembers(N, comparator) :

— N est le nombre à comparer à la population du groupe.
— comparator est l'opérateur de comparaison $<, >, \leq, \geq, ==$.

2.1.5 L_IsInThatGropInThatRole

La condition porte sur le rôle de l'agent dans son groupe. La signature de cette action est L_IsInThatGropInThatRole(Role) :

— Role, le rôle à comparer avec le rôle de l'agent.

2.2 Les algorithmiques