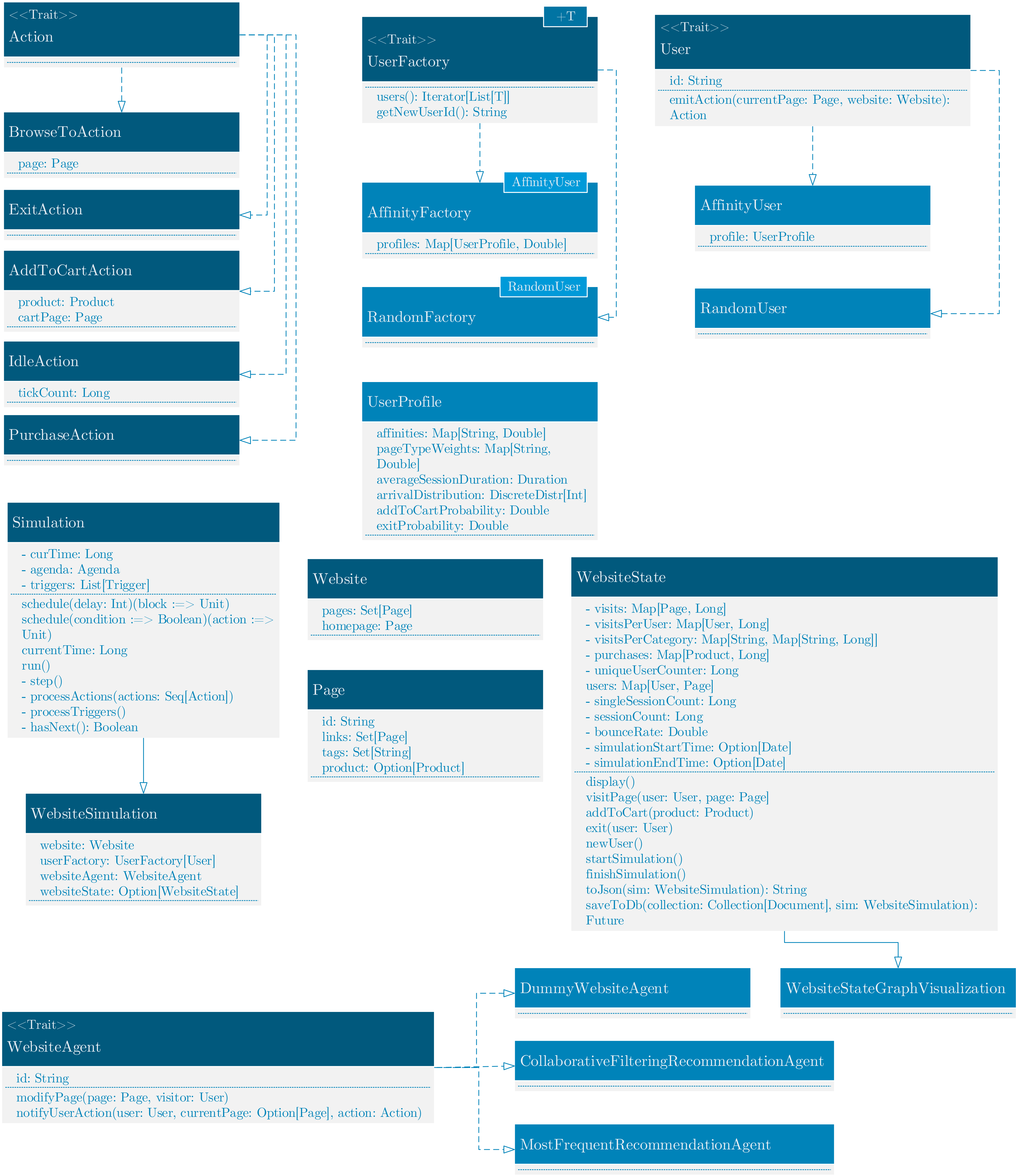
<<Trait>>
Action
BrowseToAction
page: Page
ExitAction
AddToCartAction
product: Product
cartPage: Page
IdleAction
tickCount: Long
PurchaseAction
+T
<<Trait>>
UserFactory
users(): Iterator[List[T]]
getNewUserId(): String
AffinityUser
AffinityFactory
profiles: Map[UserProfile, Double]
RandomUser
RandomFactory
<<Trait>>
User
id: String
emitAction(currentPage: Page, website: Website): Action
AffinityUser
profile: UserProfile
RandomUser
UserProfile
affinities: Map[String, Double]
pageTypeWeights: Map[String, Double]
averageSessionDuration: Duration
arrivalDistribution: DiscreteDistr[Int]
addToCartProbability: Double
exitProbability: Double
Simulation
- curTime: Long
- agenda: Agenda
- triggers: List[Trigger]
schedule(delay: Int)(block :=> Unit)
schedule(condition :=> Boolean)(action :=> Unit)
currentTime: Long
run()
- step()
- processActions(actions: Seq[Action])
- processTriggers()
- hasNext(): Boolean
WebsiteSimulation
website: Website
userFactory: UserFactory[User]
websiteAgent: WebsiteAgent
websiteState: Option[WebsiteState]
Website
pages: Set[Page]
homepage: Page
Page
id: String
links: Set[Page]
tags: Set[String]
product: Option[Product]
WebsiteState
- visits: Map[Page, Long]
- visitsPerUser: Map[User, Long]
- visitsPerCategory: Map[String, Map[String, Long]]
- purchases: Map[Product, Long]
- uniqueUserCounter: Long
users: Map[User, Page]
- singleSessionCount: Long
- sessionCount: Long
- bounceRate: Double
- simulationStartTime: Option[Date]
- simulationEndTime: Option[Date]
display()
visitPage(user: User, page: Page]
addToCart(product: Product)
exit(user: User)
newUser()
startSimulation()
finishSimulation()
toJson(sim: WebsiteSimulation): String
saveToDb(collection: Collection[Document], sim: WebsiteSimulation): Future
DummyWebsiteAgent
WebsiteStateGraphVisualization
<<Trait>>
WebsiteAgent
id: String
modifyPage(page: Page, visitor: User)
notifyUserAction(user: User, currentPage: Option[Page], action: Action)
CollaborativeFilteringRecommendationAgent
MostFrequentRecommendationAgent