

RoboCup Rescue Simulation
秋キャンプ講習
第二部：初心者向けエージェント開発

立命館大学 Ri-one
(情報理工学部情報システム学科 2回生)

渡邊 華奈子

アジェンダ

- チームの作成
 - とりあえず「動く」チームを作る
- チームの開発
 - とりあえず「動く」チームで使えるもの
- チームの拡張
 - Ri-oneチームの構成を簡単に紹介

チームの作成@Eclipse

- 新プロジェクト作成
- jarファイルの追加
- サンプルチームの切り離し
- チーム起動
 - start.shの編集
 - Eclipseでの実行
- コンパイル作業の短縮

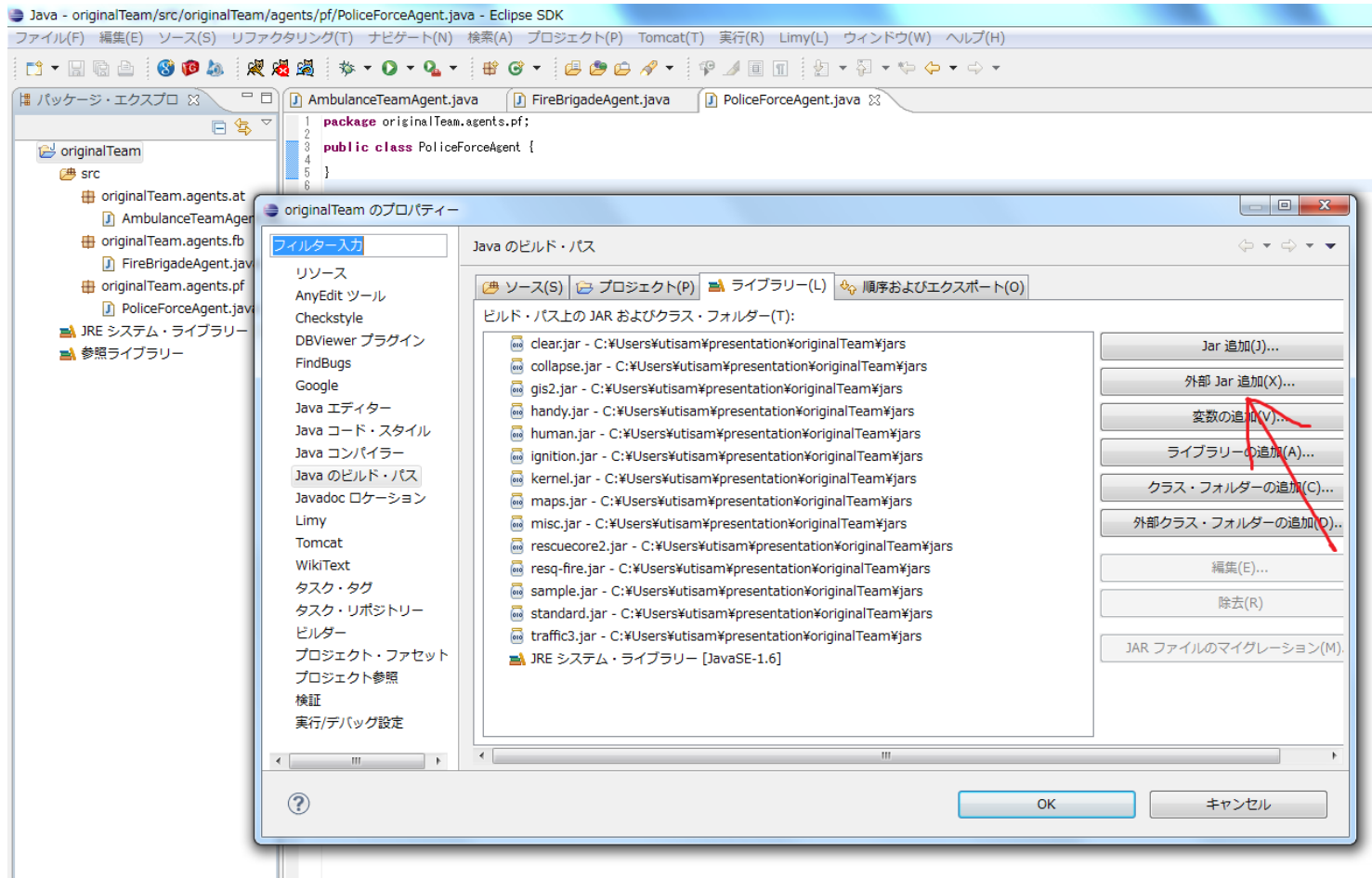
新プロジェクトの作成

- Eclipseで作成

jarファイルの追加

- rescueサーバに入っている
"jars","lib"の中にある~.jarファイル
- とりあえず全て追加すれば動く

jarファイルの追加



jarファイル補足

- Ri-one内で現在使用しているjarファイル
 - rescuecore2.jar
 - standard.jar
 - commons-logging-1.1.1.jar
 - dom4j.jar
 - jaxen-1.1.1.jar
 - jcommon-1.0.16.jar
 - jfreechart-1.0.13.jar
 - jscience-4.3.jar
 - jsi-1.0b2p1.jar
 - jts-1.11.jar
 - junit-4.5.jar
 - log4j-1.2.15.jar
 - trove-0.1.8.jar
 - uncommons-maths-1.2.jar
 - xml-0.0.6.jar

チームの切り離し

- サンプルエージェントを切り離してみる
 - AbstractSampleAgent.java
 - DistanceSorter.java
 - SampleAmbulanceTeam.java
 - SampleCentre.java
 - SampleFireBrigade.java
 - SamplePoliceForce.java
 - SampleSearch.java

start.shについて

- チーム起動のためのシェルスクリプト
- 世界大会で配布

- 今回はコメント付きのものを用意
- Windowsでは使えない

start.shの編集

- TEAM_CP:jarファイルとbin
- -Xmx:使用メモリMAX指定
- 引数の仕様はサーバにより規定
- 同じパソコン内で実行する場合は
./start.sh 127.0.0.1 7000 n n n n n n

Eclipseからの起動1

1.チームを右クリック

2.Run As

3.Run Configurations

4.Main

1.Projectにチーム名入力

2.Main classにrescuecore2.LaunchComponents

Eclipseからの起動2

- Arguments
 - program arguments
 - FB,AT,PF,FS,AC,POの順に
 - VM arguments

コンパイル作業の短縮

- 例 : antでコンパイルできるbuild.xmlを作成
 1. チームを右クリック
 2. Export
 3. General
 4. Ant build files
 5. プロジェクト(チーム)を選択
 6. Finish

チームの起動(実行)

- `java -Xms512m -Xmx1024m`
(`-cp $TEAM_CP`)
`rescuecore2.LaunchComponents`
`rione.agent.fb.FireBrigadeAgent*n`
`rione.agent.at.AmbulanceTeamAgent*n`
`rione.agent.pf.PoliceForceAgent*n`
`rione.agent.fb.FireStationAgent*n`
`rione.agent.at.AmbulanceCentreAgent*n`
`rione.agent.pf.PoliceOfficeAgent*n`
`-h` (サーバのIPアドレス)
`--loadabletypes.inspect.dir=jars` (jarファイル)
`--random.seed=1`

チームの開発

- 最初から使えるもの(サーバから提供)
 - 世界観(WorldModel)
 - Entity(実体)を取得
 - エージェント
 - サーバに投げるコマンド

エージェントのコマンド

- 共通
 - sendRest(int time)
 - sendSay(int time, byte[] data)
 - sendSubscribe(int time, int... channels)
- FireBrigade
 - sendExtinguish(int time, EntityID target, int water)
- PoliceForce
 - sendClear(int time, EntityID target)
- AmbulanceTeam
 - sendRescue(int time, EntityID target)
 - sendLoad(int time, EntityID target)
 - sendUnload(int time)

世界観の利用

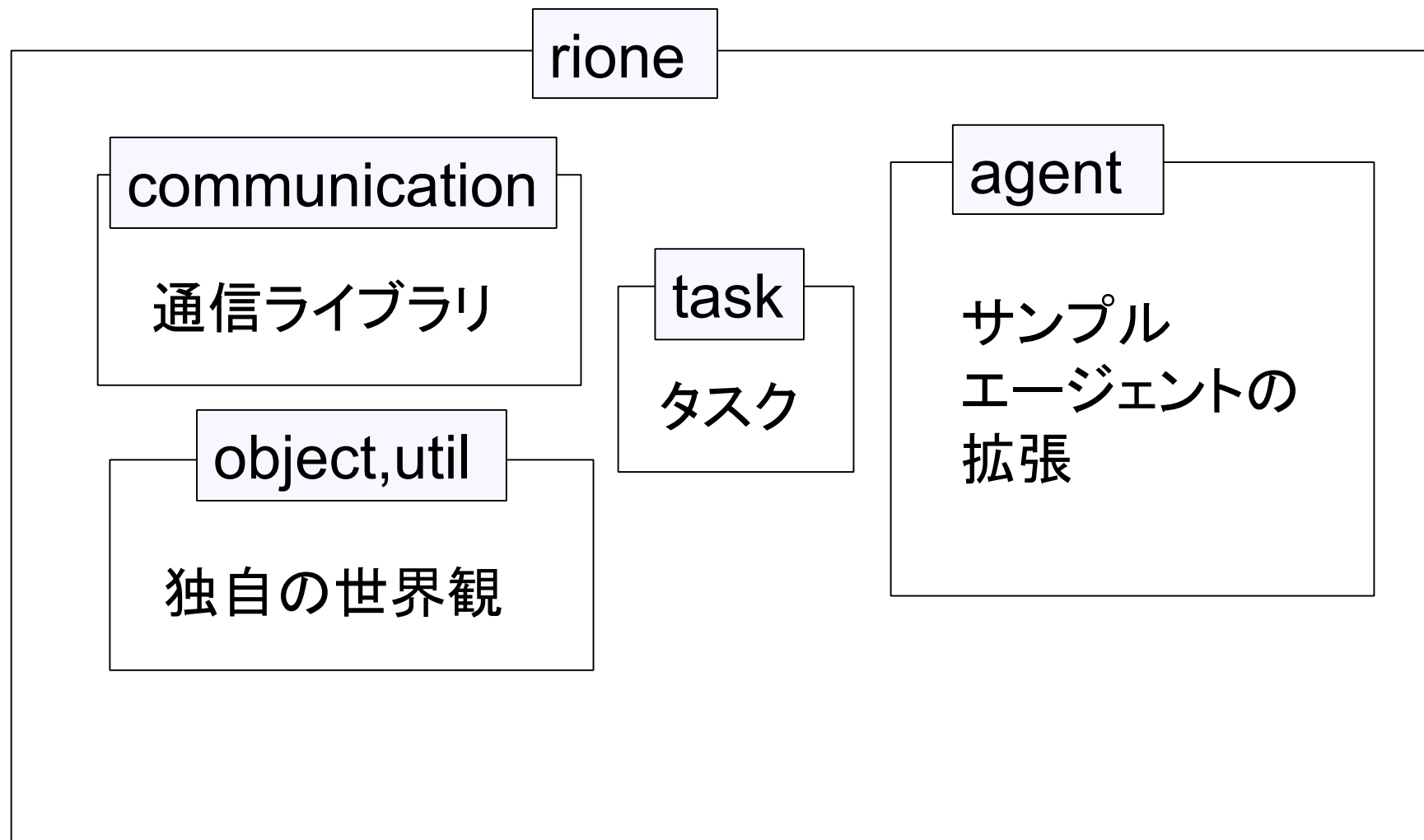
- StandardAgentはWorldModelを
"model"として持っている
- できること
 - Entity(実体)の取得
 - 動く
 - 瓦礫除去
 - 消火 ...etc

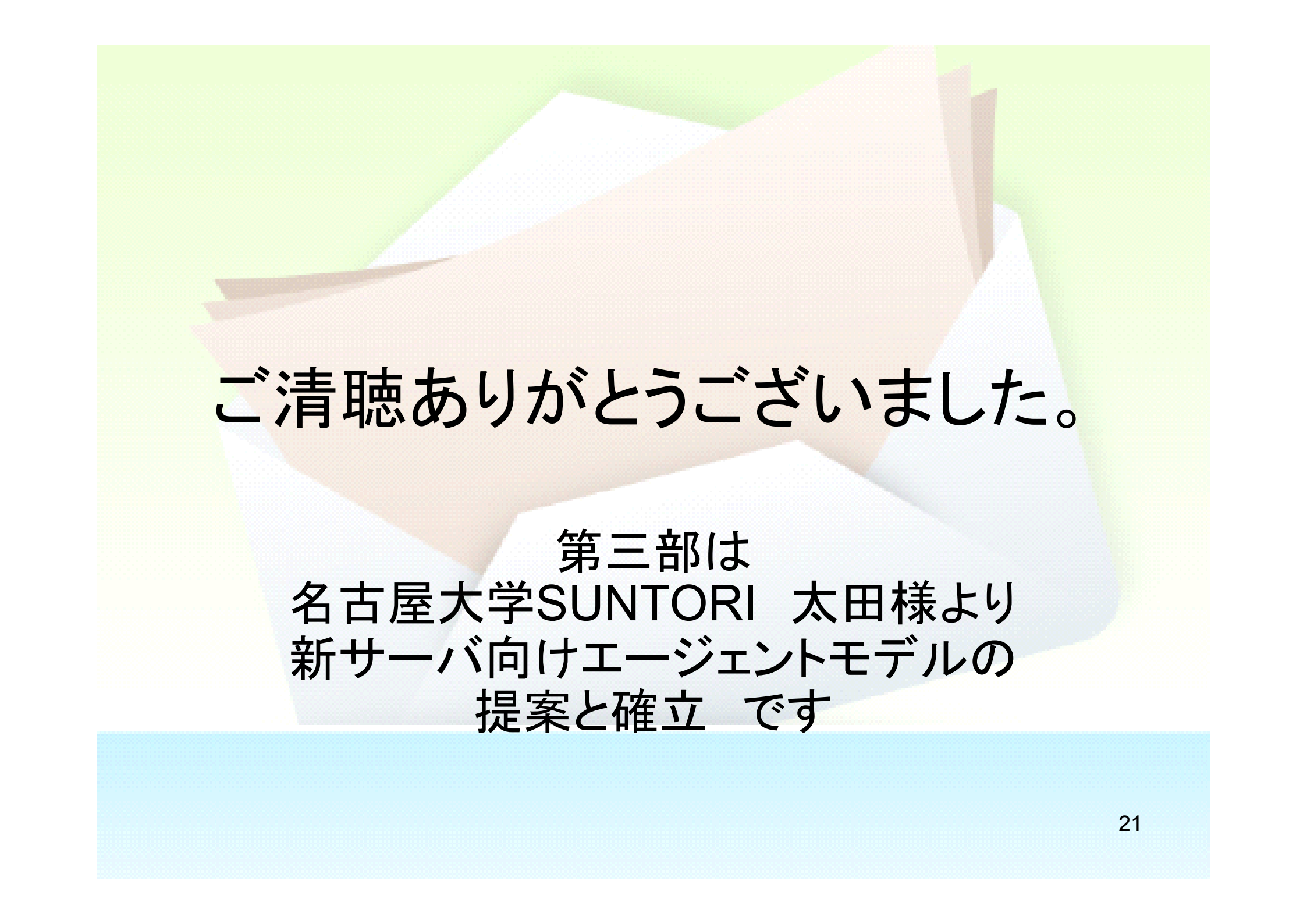
実体の取得

- `model.getEntitiesOfType(StandardEntityURN.*****)`
- `StandardEntityURN`
 - Entity(実体)を識別するためのもの
 - ROAD, BUILDING, REFUGEなど

チームの拡張

- Ri-oneの拡張例
 - 世界観 (WorldModel)
 - Agent
 - 通信ライブラリ





ご清聴ありがとうございました。

第三部は
名古屋大学SUNTORI 太田様より
新サーバ向けエージェントモデルの
提案と確立 です