0.4.4での発話生成の方法

人狼知能プラットフォーム0.4.4では、ContentクラスとContentBuilderのサブクラスで生成可能な発話のみで会話を行います。生成可能な発話は以下の14種類です。

発話生成の手順：以下の手順で得られたtextがtalk()やwhisper()の返す発話テキストとなります。

```
ContentBuilder builder = 発話の種類に応じたContentBuilder;
Content content = new Content(builder);
String text = content.getText();
```

1. estimate：targetの役職はroleだと思う。
   builder = new EstimateContentBuilder(target, role);
   例：ESTIMATE Agent[01] SEER

2. comingout：targetの役職はroleだ。
   builder = new ComingoutContentBuilder(target, role);
   例：COMINGOUT Agent[01] MEDIUM

3. divination：targetを占う。
   builder = new DivinationContentBuilder(target);
   例：DIVINATION Agent[01]

4. divined：targetを占った結果result（人間or人狼）だった。
   builder = new DivinedResultContentBuilder(target, result);
   例：DIVINED Agent[01] HUMAN

5. identified：targetは霊媒の結果result（人間or人狼）だった。
   builder = new IdentContentBuilder(target, result);
   例：IDENTIFIED Agent[01] WEREWOLF

6. guard：targetを護衛する。
   builder = new GuardCandidateContentBuilder(target);
   例：GUARD Agent[01]

7. guarded：targetを護衛した。
   builder = new GuardedAgentContentBuilder(target):
   例：GUARDED Agent[01]

8. vote：targetに投票する。
   builder = new VoteContentBuilder(target);
   例：VOTE Agent[01]

9. attack：targetに襲撃投票する。
   builder = new AttackContentBuilder(target);

例：ATTACK Agent[01]

10. agree：talkDay日目，種類talkType，talkID番目の発話に同意する。
    builder = new AgreeContentBuilder(talktype, talkDay, talkID);
    例：AGREE TALK day3 ID:20

11. disagree：talkDay日目，種類talkType，talkID番目の発話に反対する。
    builder = new DisagreeContentBuilder(talktype, talkDay, talkID);
    例：DISAGREE WHISPER day2 ID:10

12. request: agentにcontentを要求する。
    requestは特殊な構文で，requestのcontentは具体的な要求の内容を別途contentとして持つことになります.
    agentがnullの場合、要求先は不定となります。
    builder = new RequestContentBuilder(agent, content);
    例：　content = new Content(new DivinationContentBuilder(Agent.getAgent(2));
    builder = new RequestContentBuilder(Agent.getAgent(1), content);
    ↓
    REQUEST(Agent[01] DIVINE Agent[02])
    （Aget[01]にAgent[02]を占うことを要求する）

    builder = new RequestContentBuilder(null, content);
    ↓
    REQUEST(DIVINE Agent[02])
    （Agent[02]を占うことを要求する）
    RequestContentBuilderで作られた要求内容は，Content#getContentList()で獲得できます.
    なお，RequestトピックはOPERATORとなり，Operator変数がREQUESTに設定されます
    content.getTopic() => OPERATOR
    content.getOperator()=>REQUEST
    今後，0.5.x以降では，OPERATORトピックが増える予定です.

13. over：もう話すことは無い（全プレイヤーがoverを返した時点で会話フェーズ終了）。
    builder = new OverContentBuilder();
    以下で定義される定数Content.OVERが用意されています。
    public static final Content OVER = new Content(new OverContentBuilder());

14. skip：様子を見る（ただし規定回数を超えて連続するとoverになる）。
    builder = new SkipContentBuilder();
    以下で定義される定数Content.SKIPが用意されています。
    public static final Content SKIP = new Content(new SkipContentBuilder());

Contentクラス

Contentクラスは，従来のutteranceクラスを拡張した物です．人狼プロトコルのパーサの役割を果たします．

Contentクラスには以下のようなメソッドが含まれています．

- getText：発言内容を取得します．人狼プロトコルそのもの
- getOperator：操作トピックの場合に，どのような操作を表すかを返します．他のエージェントへの要求であれば，Requestが返されます．
- getSubject：このContentの主語となるエージェント．通常は発言者．
- getTopic：このContentの内容を表す列挙型Topicを返します．
- getRole：役割を表します．COなどの時に使われます
- getResult：占い結果，霊媒結果を表します．Speciesが返されます．
- getTalkType：WhisperかTalkかを返します
- getTalkDay：何日目の発話かを返します
- getTalkID：発言IDを返します
- getContentList：他のコンテンツを含む場合，そのListを返します．Requestの場合，Request内容が含まれます．

Topicの変更

Topicに以下のような変更が行われました．

- 従来のINQUESTEDはIDENTIFIEDに変更されました。
- DIVINATION, GUARD, OPERATORが追加されました

Topicの意味は以下の通りです．

- ESTIMATE：推測
- COMINGOUT：カミングアウト
- DIVINATION：占い（宣言）
- DIVINED：占い結果
- IDENTIFIED：霊媒結果
- GUARD：護衛対象（宣言）
- GUARDED：護衛対象（過去の護衛）
- VOTE：投票
- ATTACK：襲撃
- AGREE：（他の発言への）同意
- DISAGREE：（他の発言への）反対
- OVER：もう話すことはない
- SKIP：今は話すことはない
- OPERATOR：操作であることを意味する

その他

- TemplateTalkFactoryとTemplateWhisperFactoryは非推奨になりました。