関係者外秘

展示会用カスタムプロファイル　及び　サンプル実装の説明

evolveITWorks
エボルブアイティワークス株式会社

カスタムプロファイルサンプル　プロファイル概要

関係者外秘

RobotExhibitionService（Service）

RobotExhibitionService2（Service兼Robot）

実装プロファイル
・EnqueteProfile_Service
・ApproachingNotification_Service
・SnsOauthProfile_Service

CLtoSV
・transfer_enquete_data_profile

SVtoCL
・transfer_enquete_data_profile

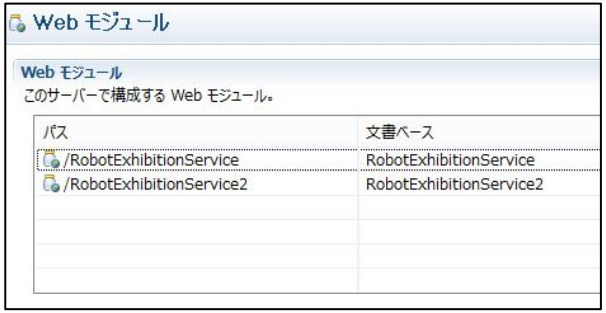
実装プロファイル
・EnqueteProfile_client

CLtoSV
・get_enquete_data
・answer_enquete

実装プロファイル
・EnqueteProfile_client
・ApproachingNotification_client
・CutomProfile_SnsOauthProfile_client（おまけ）

RobotExhibitionMain（Robot）

サンプル実装　起動方法その1



1. RobotExhibitionServiceを開始
   a. Tomcatにて起動
   b. パス設定例は右記参照

2. RobotExhibitionService2を開始
   a. Tomcatにて起動
   b. パス設定例は右記参照
   c. 起動後、自動でRobotExhibitionServiceに接続し、EnqueteProfileのスタートが行われ、transferEnqueteDataの準備が完了する

3. transferEnqueteDataの実行
   a. CLtoSV(RobotExhibitionService2からRobotExhibitionServiceへ)の実行
      i. 下記URLを開くと自動で実行され、Service側へデータが送信される
         1. http://localhost:8080/RobotExhibitionService2/transferEnquete
      ii. Service側での実装により、コンソールに出力される

```
2015-10-14 14:23:31,185 INFO  [MessageDispatcher-34] (EnqueteProfile_Impl.java:93) - responder_id=[responder_id_y] enqueteData=[[question[設問1] an
2015-10-14 14:23:31,236 INFO  [MessageDispatcher-34] (AbstractDataPushProfileCustomProfileSorter.java:143) - push_data end
```

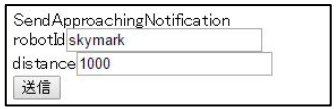
   b. SVtoCL(RobotExhibitionServiceからRobotExhibitionService2へ)の実行
      i. 下記URLを開くと自動で実行され、Robot側へデータが送信される
         1. http://localhost:8080/RobotExhibitionService/transferEnquete
      ii. Robot側での実装により、コンソールに出力される

```
情報: responder_id=[responder_id_x] enqueteData=[[question[設問1] answers=[[回答1][回答2][回答3]]][question[設問2] answers=[[2回答1][2回答2
10 14, 2015 2:30:50 午後 org.robotservices.custom_profile.lib.client.acceptor.AbstractDataPushProfileCustomProfileSorter push_data
情報: push_data end
```

サンプル実装　起動方法その2

1. RobotExhibitionMainの実行
    a. Main.javaの実行を行うことにより以下のプロファイルが実行される
        i. snsOauthProfile
            1. startProfile
            2. setTwitterOauth
                a. 認証情報の送信
            3. setFacebookOauth
                a. 認証情報の送信
            4. setgooglePlusOauth
                a. 認証情報の送信
        ii. ApproachingNotificationProfile
            1. startProfile
            2. sendApproachingNotification_profile
                a. ロボット接近の通知
        iii. EnqueteData
            1. startProfile
            2. getEnquete
                a. 設問の取得
            3. answerEnquete
                a. 回答をServiceへ送信

2. ApproachingNotificationProfileのSVtoCLの実行
    a. 以下のURLを表示し、SVtoCLのreceiveApproachingNotificationの実行が可能
    b. http://localhost:8080/RobotExhibitionService/sendApproachingNotification



- アンケートの設問と回答を、転送するプロファイル
- CLtoSV
  - startProfile()
    - プロファイルの開始
  - endProfile()
    - プロファイルの終了
  - getEnquete(String enquete_id,String responder_id)
    - enquete_id(アンケート固有のIDを想定)とresponder_id(回答者固有のID)から、アンケートをServiceから取得する
    - 戻り値は「EnqueteData」詳細は後述
  - answerEnquete(String enquete_id, String responder_id,AnswerData answerData)
    - 回答を行ったenquete_id、回答者responder_id、回答内容AnswerDataをServiceへ送信する
  - transferEnqueteData(String responder_id,EnqueteData enqueteData, AnswerData answerData)
    - 回答者が回答した設問EnqueteDataと回答AnswerData を同時にServiceへ転送する
- SVtoCL
  - transferEnqueteData(String responder_id,EnqueteData enqueteData, AnswerData answerData)
    - 回答者が回答した設問EnqueteDataと回答AnswerData を同時にRobotへ転送する
    - CLtoSVの逆

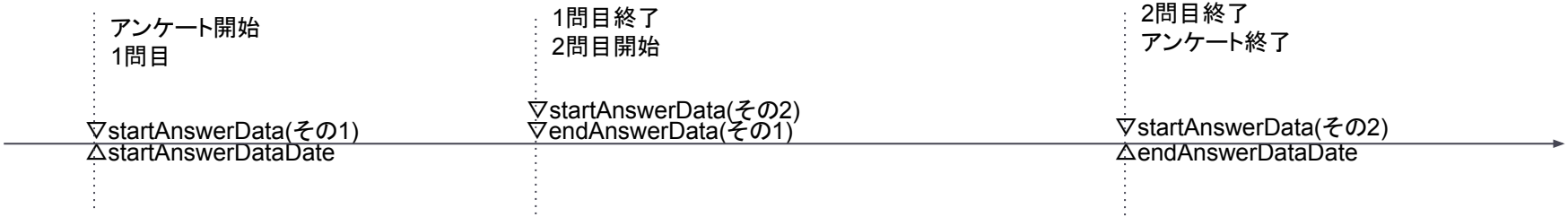
EnqueteProfile概要その2

- IDの概要
  - enquete_id
    - 設問群の固有のID
    - 設問ごとのIDは、question_id
  - question_id
    - 設問ごとのID
  - responder_id
    - アンケートに回答した人(robot_idを想定だが、別のIDでも問題ない)

- Data群の概要
  - EnqueteData
    - enquete_idに対する設問群
  - Enqute
    - EnquteDataに属する設問
    - 設問は、question_idの1つに対応する
    - 設問は、複数の選択肢を保持する
  - AnswerData
    - EnqueteDataに対する回答群
    -
  - Answer
    - AnswerDataに属する回答
    - 回答は、question_idの1つに対応する

- AnswerDataと、それに付随するAnswerには、それぞれStartDateとEndDateがある
  - AnswerData
    - startAnswerDataDate
      - アンケート全体の設問開始時間
    - startAnswerDataDate
      - アンケート全体の設問終了時間
  - Answer
    - startAnswerDate
      - 設問単位の設問開始時間
    - endAnswerDate
      - 設問単位の設問終了時間

実装例:



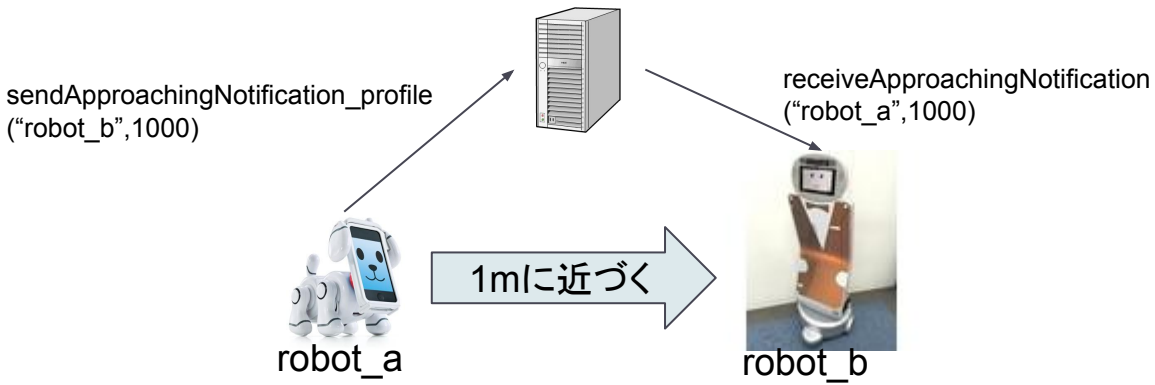
EnqueteDataのXML構造(参考)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:enquete_profile xmlns:ns2="http://www.robotservices.org/schemas/EnqueteQuestionData">
    <euquete_id>enquete_id_1</euquete_id>
    <beforeEnqueteAnnotation>アンケート前注釈</beforeEnqueteAnnotation>
    <afterEnqueteAnnotation>アンケート後注釈</afterEnqueteAnnotation>
    <enqueteList>
        <enquete>
            <question_id>ID1</question_id>
            <beforeQuestionAnnotation>設問1前注釈</beforeQuestionAnnotation>
            <afterQuestionAnnotation>設問1後注釈</afterQuestionAnnotation>
            <question>設問1</question>
            <answers>回答1</answers>
            <answers>回答2</answers>
            <answers>回答3</answers>
        </enquete>
        <enquete>
            <question_id>ID2</question_id>
            <beforeQuestionAnnotation>設問2前注釈</beforeQuestionAnnotation>
            <afterQuestionAnnotation>設問2後注釈</afterQuestionAnnotation>
            <question>設問2</question>
            <answers>2回答1</answers>
            <answers>2回答2</answers>
            <answers>2回答3</answers>
        </enquete>
    </enqueteList>
</ns2:enquete_profile>
```

```
<answerData>
  <enquete_id>enquete_id_1</enquete_id>
  <answerList class="java.util.ArrayList">
    <answer>
      <question_id>ID1</question_id>
      <answer>回答1 </answer>
      <startAnswerDate>2015-11-12 19:43:24.736 JST</startAnswerDate>
      <endAnswerDate>2015-11-12 19:43:24.845 JST</endAnswerDate>
    </answer>
    <answer>
      <question_id>ID2</question_id>
      <answer>2回答1 </answer>
      <startAnswerDate>2015-11-12 19:43:24.845 JST</startAnswerDate>
      <endAnswerDate>2015-11-12 19:43:24.949 JST</endAnswerDate>
    </answer>
  </answerList>
  <startAnswerDataDate>2015-11-12 19:43:24.736 JST</startAnswerDataDate>
  <endAnswerDataDate>2015-11-12 19:43:25.58 JST</endAnswerDataDate>
</answerData>
```

ApproachingNotification_Profileの概要

- 端末(robot)の接近をサーバに通知するプロファイル
- CLtoSV
  - startProfile()
    - 開始
  - endProfile()
    - 終了
  - sendApproachingNotification_profile(String to_robot_id, long distance_mm)
    - 端末がto_robot_idに対して近づいていることを、距離で通知する
- SVtoCL
  - RetValue receiveApproachingNotification(String from_robot_id, long distance_mm);
    - from_robot_idが、通知先へ近づいてることを、距離と共に通知する



- 端末側の認証情報をサーバに送信するプロファイル
- CLtoSV
  - startProfile()
    - 開始　Robo_idを貰えるので、それをユーザーの IDとして使う
  - endProfile()
    - 終了

  - RetValue setTwitterOauth(String consumer_key,String consumer_secret, String token_secret, String access_token,String screen_name)
    - TwitterのOauth情報のサーバ送信

  - RetValue setFacebookOauth(String application_id, String access_token, Sring expire_in)
    - FacebookのOauth情報のサーバ送信

  - RetValue setgooglePlusOauth(String application_id,String client_id,String client_secret, String access_token,String token_type,String expires_in,String refresh_token)
    - GooglePlusのOauth情報サーバ送信

SNS認証情報

- Twitter
  - 開発側固有Key
    - CONSUMER_KEY
    - CONSUMER_SECRET
  - ユーザー側固有Key
    - TOKEN_SECRET
    - ACCESS_TOKEN
- Facebook
  - 開発側固有Key
    - APPLICATION_ID
  - ユーザー側固有Key
    - ACCESS_TOKEN
    - EXPIRE_IN
- Google+
  - 開発側固有Key
    - CLIENT_ID
    - client_secret
  - ユーザー側固有Key
    - access_token
    - token_type
    - expires_in
    - refresh_token