

Documento de desenvolvimento da aplicação OMAS TECPAR-CSCW

Allan Trevisan 11/09/2012

1. Agente de serviço

O agente TECPAR-DP-MONITOR é um agente de serviço a cargo de monitorar a base de dados da ferramenta de gestão de projetos dotproject, afim de captar ações realizadas na ferramenta.

O agente possui um conjunto de skills que realizam buscas na base de dados (mysql) da ferramenta dotproject. O comportamento de monitoramento da base de dados é definido pelos objetivos do agente, inicialmente ele esta responsável por monitorar apenas um projeto com o id 1 e nome "CSCW-TECPAR", posteriormente mais projetos podem ser adicionados com a Skill ADD-MONITORED-PROJECT, onde dado o nome do projeto que se quer adicionar ela buscará o código do mesmo armazenado na ferramenta e irá adicionar à lista de projetos monitorados.

O comportamento autônomo de monitoramento é definido com o uso de objetivos, um objetivo é definido com segue:

```
(defun goal-search-what-activities-to-do-now (agent)
  (list (make-instance 'omas::message :type :request
    :from (omas::key agent) :to (omas::key agent)
    :action :search-what-activities-to-do-now
    :args nil
    :task-id :T0)))
```

```
(make-goal 'MONITOR-PROJECT-TASKS :TECPAR-DP-MONITOR
  :type :cyclic
  :period 30 ; seconds
  :activation-date (+ (get-universal-time) 3))
```

```
:script 'goal-search-what-activities-to-do-now)
```

O objetivo definido acima é do tipo periódico rígido. Um objetivo cria uma lista de mensagens (no caso a função goal-search-what-activities-to-do-now) que o agente envia para si mesmo, em um intervalo de tempo (30 segundos nesse exemplo) afim de ativar uma skill ciclicamente.

As funções de acesso ao banco de dados estão definidas no arquivo database-access. Uma serie de strings foram definidas, cada uma contendo uma consulta sql. As funções que realizam as consultas na base de dados fazem uso dessas strings e de uma macro with-database-connection, essa macro cria uma abstração ao livrar o usuário da tarefa de liberar recursos alocados na conexão com o banco de dados. Ela é utilizada da seguinte forma:

```
(with-database-connection (registers fields)
                           (:selector *string-consulta*
                           :args (list arg1-consulta arg2-consulta ... argn-
consulta))

                           retorno)
```

onde registers corresponde aos registros retornados pela consulta e fields são os campos. Esse arquivo esporta as funções de consulta que são utilizadas pelo agente de serviço.

2. Agente Assistente

EUZEBIO é o agente assistente a cargo de interagir com o usuário, é através da interação com o usuário que o agente assistente irá enviar requisições ao agente de serviço afim dele realizar consultas na base de dados.

Inicialmente para fins de teste foram definidos dois diálogos cada um responsável por interagir com o usuário e permitir que ele obtenha informações relativas ao andamento das atividades nos projetos.

Uma das tarefas definidas no arquivo EUZEBIO-TASKS é a tarefa WHAT-ACTIVITIES-SHOULD-A-PROJECT-MEMBER-TO-DO-NOW responsável por obter informações relativas às atividades que alguém deve realizar no momento. O diálogo relacionado a essa tarefa possui os seguintes estados:

- `_gpma-entry-state`: (get project member activities entry state), nesse estado busca-se um padrão na frase inserida pelo usuário, se a frase corresponder ao padrão passa-se para o próximo estado do dialogo (`_gpma-ask-monitor-agent`) caso contrario passa-se para o estado `_gpma-sorry`, que mostra uma mensagem se desculpando por não ter encontrado nenhuma informação.
- `_gpma-ask-monitor-agent`: (get project member activities ask monitor agent), nesse estado do dialogo o agente assistente envia uma mensagem ao agente de serviço para que ele execute sua skill, os argumentos são a frase entrada pelo usuário, e a informação que se trata de um dialogo em português no brasil. O agente de serviço irá receber uma mensagem com argumentos no formato:

```
(:data "Atividades" "de" "Allan" "Trevisan")
```

Para processar a mensagem o agente de serviço irá então fazer uso da sua ontologia de domínio onde a função:

```
(moss::locate-objects data "pessoa")
```

será utilizada para buscar por indivíduos pessoa definidos na ontologia que corresponda aos dados da mensagem. Dentro da skill o objeto pessoa é desmembrado em seus componentes (nome e sobrenome) que serão utilizados pela função que realiza a consulta das atividades que um membro de projeto deve realizar.

- `_gpma-sorry`: (get project member activities sorry), imprime uma mensagem de desculpas dizendo que não achou nenhuma informação.

