Proposal for OntologyManager interface extension

Alfredo Pérez & David Rozas

Introduction

In order to continue with the implementation of the ApplicationManager, and according to the use cases defined in the document "Design of the EUT: Application Manager", we have to define the communication interface between ApplicationManager and OntologyManager.

The involved use cases are: UC001_, D42_UC-REP01, D42_UC-REP05, UC003 and UC005.

Java interface proposal

Share Application

When the user shares an application we send information about that application to the OntologyManager.

The information that the OntologyManager needs has to be analyzed more deeply, but a first approach can be the following:

```
public String[]  getApplicationParameters();
public String getParameterType(String parameter);
public String getParameterDefaultValue(String parameter);
public String getParameterDescription(String parameter);
public void setParameterValue(String appID, Object o);
public boolean validateParameter(String appID, Object o);
```

This interface specifies the communication between the ApplicationManager and the OntologyManager.

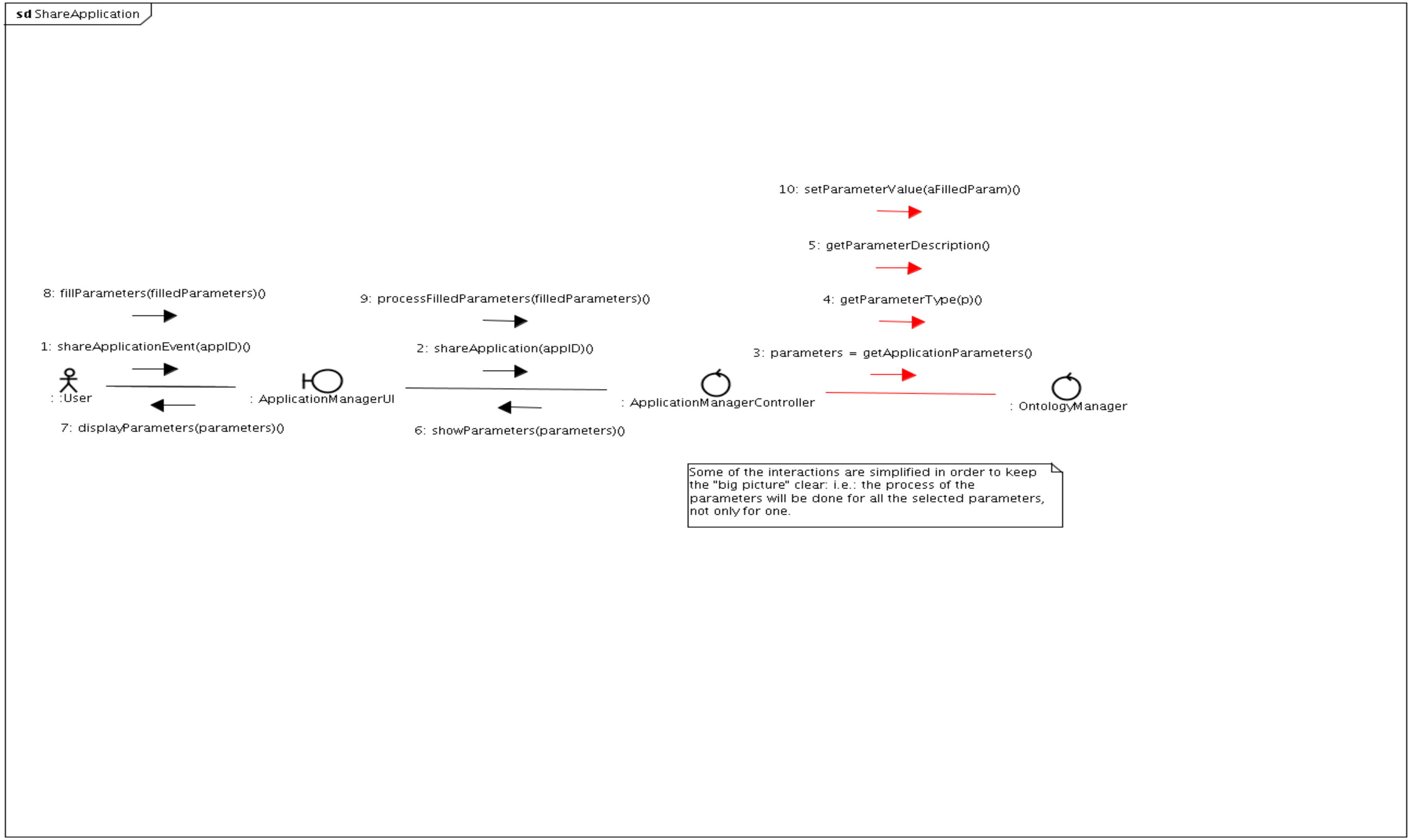
The *getApplicationParameters* method requests the *OntologyManager* for a list of the possible parameters that can be used in order to categorize an application. After retrieving the list of parameters it's necessary to get the type of the parameter in each case (integer, string, list ...).

The user will be asked to fill a set of fields with every parameter that specifies the application.

It will be possible to validate the value of every parameter by using the validateParameter method which will inform whether the value of the parameter is well formed or not.

After the validation process the ApplicationManager would transfer the information to the OntologyManager iterating over the setParameterValue.

An example of the interactions between ApplicationManager and OntologyManager is shown in the following diagram (marked in red).



sd ShareApplication
10: setParameterValue(aFilledParam)()
5: getParameterDescription()
8: fillParameters(filledParameters)()
9: processFilledParameters(filledParameters)()
4: getParameterType(p)()
1: shareApplicationEvent(appID)()
2: shareApplication(appID)()
3: parameters = getApplicationParameters()
: :User
: ApplicationManagerUI
: ApplicationManagerController
: OntologyManager
7: displayParameters(parameters)()
6: showParameters(parameters)()

Some of the interactions are simplified in order to keep the "big picture" clear: i.e.: the process of the parameters will be done for all the selected parameters, not only for one.

The original image can be found at: http://basar.idi.ntnu.no/svn/astra/WP3/Y3/SOA.2.0/ApplicationManager/doc/design/img/ShareApplication.png if needed)

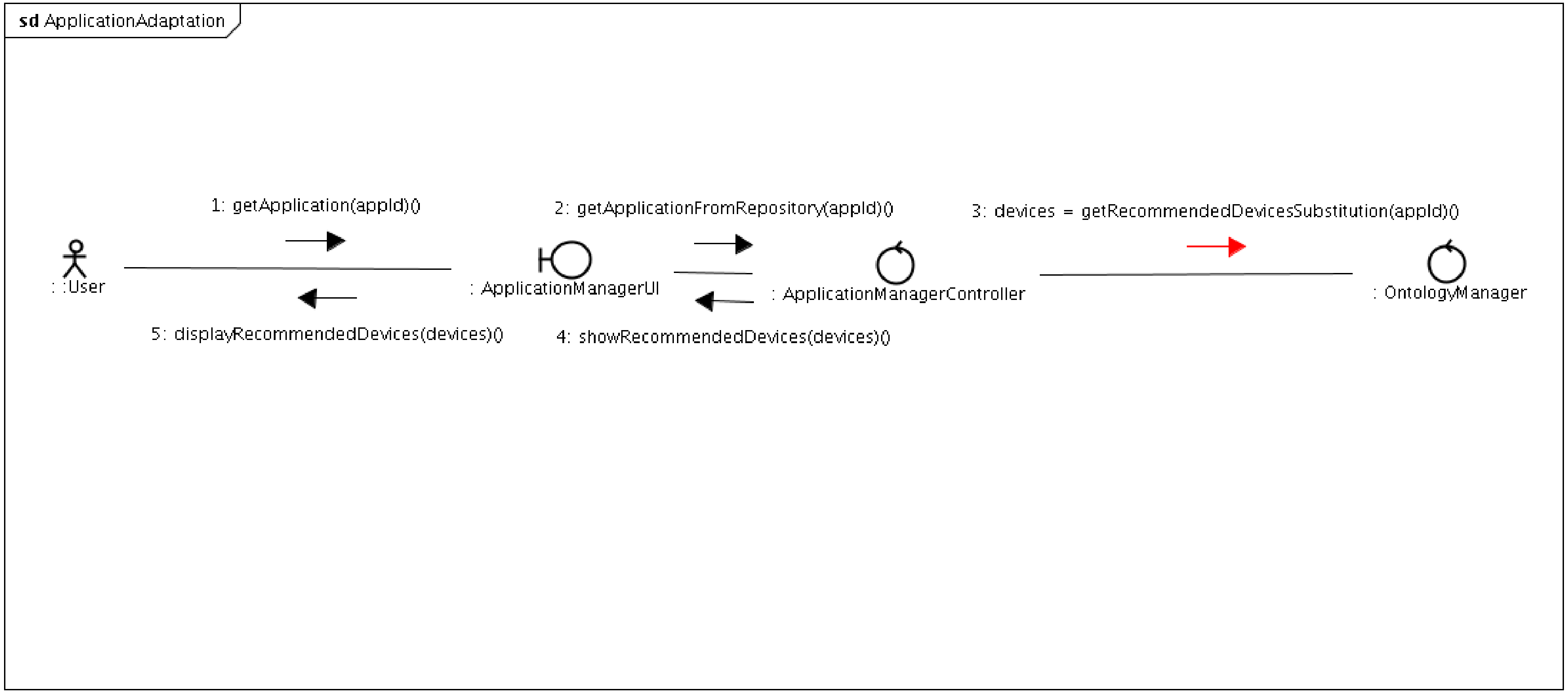
Application Adaptation

When the user retrieves an application from the repository, it will ask the OntologyManager for information in order to help it to adapt that application (ex.: which kind of devices are being used, ...).

The information that the OntologyManager offers has to be analyzed more deeply, but a first approach can be the following:

```
public String[] getRecommendedDevicesSubstitution(String userId, String
appId,String device);
```

An example of the interactions between ApplicationManager and OntologyManager is shown in the following diagram (marked in red).


sd ApplicationAdaptation
1: getApplication(appId)()
2: getApplicationFromRepository(appId)()
3: devices = getRecommendedDevicesSubstitution(appId)()
: :User
: ApplicationManagerUI
: ApplicationManagerController
: OntologyManager
5: displayRecommendedDevices(devices)()
4: showRecommendedDevices(devices)()

The original image can be found at:
http://basar.idi.ntnu.no/svn/astra/WP3/Y3/SOA.2.0/ApplicationManager/doc/design/img/ApplicationAdaptation.png if needed)

Search Application

As it's specified in the Use Cases document, it's necessary to provide a way to search for applications by using the Ontology.

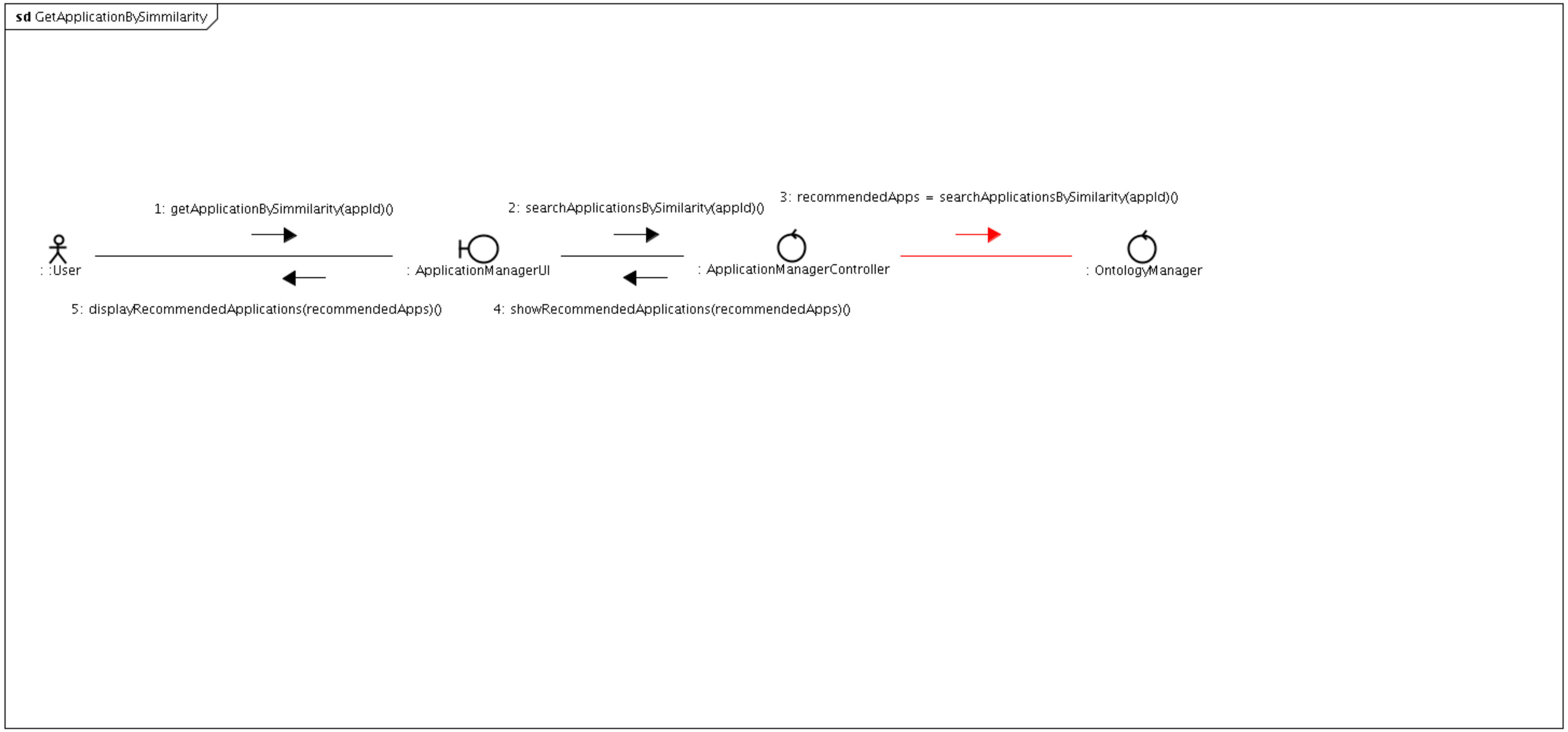
One approach would be to implement the following functionality.

```
      public String[]  searchApplication(String  [][] parameters);
      public String[] searchBySimilarity(String appID);
```

```
Where the parameters for the first method would have the following structure:
{[ {[ Parameter_name, Parameter_Value ]} | ... ]}

      String [][] parameters = { { “Location” , “Patras”} , { “Time”,
“Weekly”} , { “Category” , “Social” } ... };
```

An example of the interactions between ApplicationManager and OntologyManager for the searchBySimmilarity case is shown in the following diagram (marked in red).


sd GetApplicationBySimmilarity
1: getApplicationBySimmilarity(appId)()
2: searchApplicationsBySimilarity(appId)()
3: recommendedApps = searchApplicationsBySimilarity(appId)()
: :User
: ApplicationManagerUI
: ApplicationManagerController
: OntologyManager
5: displayRecommendedApplications(recommendedApps)()
4: showRecommendedApplications(recommendedApps)()

The original image can be found at:
http://basar.idi.ntnu.no/svn/astra/WP3/Y3/SOA.2.0/ApplicationManager/doc/design/img/GetAppliationBySimmilarity.png if needed)

The last version of this document (in different formats) can be found at:
http://basar.idi.ntnu.no/svn/astra/WP3/Y3/SOA.2.0/ApplicationManager/doc/ontologies-proposal/