# 6
# Conclusions and Future Work

This dissertation proposes the use of cloud infrastructures as the target environment for the deployment of distributed component-based applications. To support the deployment of these types of applications the Deployment Infrastructure was extended for SCS components [Junior09]. This deployment infrastructure, composed of *DeployManager*, *Packager*, and *Repository* services, was also deployed on a cloud infrastructure, obtaining an experimental Platform as a Service(PaaS). These services manage a set of deployment activities such as packaging, publishing, planning, installing, activating, deactivating, uninstalling, and retiring. Thus, the deployment infrastructure acts as a *platform* on the cloud allowing *deployment actors* to deploy their applications on the same cloud.

In order to deploy distributed component-based applications we set up and configured a private and a public cloud. We utilized *OpenStack* to set up a private cloud and configured the access to *Amazon Web Services* as public cloud. In doing this, we created a suitable cloud computing environment for our test scenarios. However, to orchestrate the provision of cloud resources for the target environment and the deployment infrastructure, it was necessary to implement a *cloud API*. Hence, we implemented a *cloud API* composed of two layers, *EucaEngine* and *Cloud Deployment Service*. We highlighted the use of the policies approach to simplify the requirement specifications related to the cloud resources following an incremental level of details. For example, we can run new virtual machine instances from a number of instances to a policy containing a customized resource cloud specification. In addition, we took advantage of the policies to define parameters of the *platform*, *users*, and the *applications*. The deployment infrastructure and the cloud infrastructures involve many types of users, thus we proposed a *user management model* to deal with the management of users. Finally, a set of extensions to the DeployManager service was implemented combining the *cloud API*, *policies*, *user management model*, and using the *cloud computing environment*. We successfully tested the adoption of cloud infrastructures for provisioning computing resources as the target environment. Although the cloud API was the connec-

tion point to join the deployment process and cloud infrastructures, we found some difficulties integrating both areas. For example, cloud computing does not have a standard documentation to guide the implementation of a new cloud API. Also, we did not find the necessary libraries already implemented with Lua for cloud infrastructures, therefore this made extra work. We deployed applications into private and public clouds. To combine both could be an interesting research area, however compatibility issues need to be studied. Other research areas like the fault-tolerant approach are being studied because complex cloud computing environments are built using many layers.

We are planning several directions for future work. First, we obtained an experimental Platform as a Service (PaaS) after working with the Deployment Infrastructure deployed on the cloud. Thus, we believe that increasing the level of automation to deploy the deployment infrastructure, and offering a proper API to programmers we can obtaining a more complete PaaS. Second, we have developed a cloud API inspired by compute resources, but there is a plethora of services available in private and public clouds. Therefore, we could add more support to our cloud API in order to include services such storage, databases, monitoring, etc. Third, we used policies to specify a group of parameters, however we did not get a full adoption of the policy approach explained in [Marriot96]. Due to the fact that policies involve a relationship between a subject and a target, and this relationship is specified as an action. Therefore, evolving our policy approach we can offer *deployment actors* a more flexible and complete form in order to specify policies. Fourth, we obtained many metadata associated with the virtual machine instances, when *getting* or *running* virtual machine instances. But the deployment infrastructure does not use these metadata yet. Therefore, we plan to use this information to improve some deployment activities such as planning, configuration, reconfiguration, and adaptation.